ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,

ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Από τα Στοιχειοσύνολα και τις Τροχιές Κινούμενων Αντικειμένων στις Υπηρεσίες Θέσης: Μια Προσέγγιση Προστασίας της Ιδιωτικότητας μέσω Απόκρυψης Γνώσης

Μια εργασία που εκπονήθηκε από

τον Άρη Γκουλαλά-Διβάνη για τις απαιτήσεις

του Διδακτορικού Διπλώματος

Επιβλέπων Καθηγητής: κ. Ηλίας Ν. Χούστης

Συνεπιβλέπων Καθηγητής: κ. Βασίλειος Σ. Βερύκιος

Βόλος, Ιανουάριος 2009

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

ΕΙΣΑΓΩΓΗ

Η ραγδαία ανάπτυξη της τεχνολογίας στους τομείς των βάσεων δεδομένων, των αποθηκευτικών μέσων και των συσκευών συλλογής δεδομένων, έχει επιτρέψει την καταγραφή ενός τεράστιου όγκου πληροφοριών σε εταιρικές βάσεις δεδομένων. Ανεξάρτητα από τη χρήση των πληροφοριών αυτών για την υποστήριξη των καθημερινών δραστηριοτήτων της εταιρίας, η αυτοματοποιημένη ανάλυση των πληροφοριών από την εταιρία, με τη χρήση μεθοδολογιών εξόρυξης γνώσης, μπορεί να συντελέσει στη λήψη καθοριστικών αποφάσεων για μελλοντικές της κινήσεις. Από την άλλη πλευρά, συχνά οι πληροφορίες που καταγράφονται στις εταιρικές βάσεις δεδομένων εμπεριέχουν ευαίσθητα προσωπικά δεδομένα (π.χ. συναλλαγές ατόμων με την εταιρία) ή/και εταιρικά μυστικά (π.χ. τεχνογνωσία για νέα προϊόντα ή μελλοντικές υπηρεσίες). Η γνώση των πληροφοριών αυτών από μη εξουσιοδοτημένα άτομα μπορεί να θέσει σε κίνδυνο την ιδιωτικότητα των ατόμων των οποίων τα στοιχεία έχουν καταγραφεί στη βάση δεδομένων ή και να δώσει συγκριτικό πλεονέκτημα σε κάποια εταιρία έναντι των ανταγωνιστών της. Η ανάγκη προστασίας των ευαίσθητων προσωπικών δεδομένων και/ή της γνώσης που βρίσκεται «κρυμμένη» σε βάσεις δεδομένων που υποστηρίζουν καθημερινές λειτουργίες, οδήγησε στη δημιουργία ενός νέου τομέα έρευνας που είναι γνωστός με το όνομα «Προστασία της Ιδιωτικότητας κατά την Εξόρυξη Γνώσης» (Privacy Preserving Data Mining). Στόχος του συγκεκριμένου τομέα είναι η κατασκευή αλγορίθμων που να εγγυώνται την προστασία της ιδιωτικότητας των ατόμων των οποίων τα στοιχεία καταγράφονται σε βάσεις δεδομένων, καθώς επίσης και την προστασία εταιρικών μυστικών που μπορούν να ανακαλυφθούν κατά την εφαρμογή αλγορίθμων εξόρυξης γνώσης.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Ο τομέας της Προστασίας της Ιδιωτικότητας κατά την Εξόρυξη Γνώσης ξεκίνησε το 2000 με δύο ερευνητικές εργασίες που αφορούσαν τη δημιουργία κατηγοριοποιητών από ευαίσθητα δεδομένα, τα οποία και έπρεπε να παραμείνουν απόρρητα. Από τότε, πολλές ήταν οι κατευθύνσεις του τομέα οι οποίες ερευνήθηκαν, ενώ επίσης πολλές μεθοδολογίες προτάθηκαν σε κάθε κατεύθυνση. Ορισμένες από τις σπουδαιότερες κατευθύνσεις που έχουν ερευνηθεί, είναι οι ακόλουθες: (α) προστασία της ιδιωτικότητας των ατόμων κατά τη δημοσιοποίηση συλλογών δεδομένων, (β) προστασία των ευαίσθητων προτύπων που μπορεί να ανακαλυφθούν κατά τη διαδικασία εξόρυξης γνώσης, (γ) έλεγχος και τροποποίηση ή απόκρυψη αποτελεσμάτων επερωτήσεων προς βάσεις δεδομένων, τα οποία αφορούν ευαίσθητα δεδομένα, (δ) προστασία των δεδομένων κατά την από κοινού ανάλυσή τους από συνεργαζόμενους φορείς, και (ε) θεωρητικά ζητήματα που αφορούν τη δυνατότητα κλιμάκωσης των αλγορίθμων απόκρυψης σε πολυδιάστατα δεδομένα. Τα τελευταία χρόνια, έχει αναπτυχθεί ένα σύνολο αλγορίθμων για την προστασία της ιδιωτικότητας κατά την εξόρυξη γνώσης από ένα σύνολο διαφορετικών μορφών δεδομένων από ότι τα παραδοσιακά δεδομένα συναλλαγών. Χαρακτηριστικά παραδείγματα αποτελούν μεθοδολογίες που έχουν προταθεί για την προστασία της ιδιωτικότητας σε χωροχρονικά δεδομένα, σε δεδομένα κίνησης και τροχιών κινούμενων αντικειμένων, καθώς επίσης και σε «ροές» δεδομένων, όπου η πληροφορία γίνεται διαθέσιμη αυξητικά και με μη προκαθορισμένο ρυθμό.

Η παρούσα διατριβή προτείνει ένα σύνολο μεθοδολογιών για την προστασία της ευαίσθητης γνώσης κατά την εφαρμογή τεχνικών εξόρυξης σε διαφορετικά είδη δεδομένων (δεδομένα συναλλαγών και δεδομένα τροχιών κινούμενων αντικειμένων) και σε διαφορετικές τεχνικές εξόρυξης δεδομένων (εξόρυξη συχνών στοιχειοσυνόλων και κανόνων συσχέτισης, εξόρυξη ακολουθιακών προτύπων και εξόρυξη χωροχρονικών προτύπων που αφορούν τροχιές κινούμενων αντικειμένων). Η διατριβή είναι χωρισμένη σε τρία μέρη. Το πρώτο μέρος

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

επικεντρώνεται στην προστασία της ευαίσθητης γνώσης κατά την εξόρυξη συχνών στοιχειοσυνόλων και κανόνων συσχέτισης από δεδομένα συναλλαγών (σχεσιακά δεδομένα). Συγκεκριμένα, προτείνεται ένα σύνολο τεχνικών οι οποίες επιτυγχάνουν την απόκρυψη όλης της ευαίσθητης γνώσης στα δεδομένα συναλλαγών αποφεύγοντας τυχόν παρενέργειες και επιφέροντας με αυτόν τον τρόπο τη μικρότερη δυνατή τροποποίηση της αρχικής βάσης. Επίσης, προτείνεται ένα πλαίσιο για την παραλληλοποίηση των τεχνικών απόκρυψης το οποίο εγγυάται την ικανότητα κλιμάκωσης και την ελαχιστοποίηση του χρόνου εκτέλεσης των τεχνικών προστασίας.

Στο δεύτερο μέρος της διατριβής, γενικεύουμε τους αλγορίθμους απόκρυψης που αναπτύξαμε στο πρώτο μέρος, εφαρμόζοντάς τους σε σύγχρονες μορφές δεδομένων, όπως είναι οι τροχιές κινούμενων αντικειμένων. Οι προκλήσεις που καλούμαστε να αντιμετωπίσουμε σε αυτό το μέρος έχουν να κάνουν με την χρονική και την χωρική φύση των δεδομένων. Σε αντίθεση με τα δεδομένα συναλλαγών, οι νέες τεχνικές καλούνται να ανιχνεύσουν περιοχές κινδύνου μέσα σε τροχιές κινούμενων αντικειμένων και να τις καταστήσουν ανώνυμες, δηλαδή να αποτρέψουν τη σύνδεση της τροχιάς με κάποιο συγκεκριμένο χρήστη. Στα πλαίσια του μέρους αυτού στηριζόμαστε στο σύστημα διαχείρισης τροχιών κινούμενων αντικειμένων HERMES (υλοποιημένο πάνω στο ΣΔΒΔ της Oracle) και προτείνουμε μια μηχανή επερωτήσεων (HERMES++) μέσω της οποίας οι διάφοροι (μη έμπιστοι) χρήστες ενός ΣΔΒΔ μπορούν να υποβάλλουν επερωτήσεις σχετικά με ένα σύνολο από καταχωρημένες τροχιές κινούμενων αντικειμένων. Στόχος του HERMES++ είναι να εξασφαλίσει ότι οι απαντήσεις στις υποβαλλόμενες επερωτήσεις θα αποτρέπουν επιθέσεις ανακάλυψης-ταυτοποίησης και επιθέσεις παρακολούθησης συγκεκριμένης, καταχωρημένης, τροχιάς-χρήστη.

Στο τρίτο μέρος της διατριβής γενικεύουμε το πρόβλημα της απόκρυψης τροχιών εξετάζοντας την εφαρμογή του σε ένα δυναμικό σύστημα στο οποίο τα δεδομένα παράγονται σε πραγματικό χρόνο. Συγκεκριμένα, οι τροχιές των

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

κινούμενων αντικειμένων παύουν να είναι στατικά αποθηκευμένες σε ένα ΣΔΒΔ, αλλά αποθηκεύονται σημείο προς σημείο τη στιγμή της κίνησης των αντικειμένων. Επίσης, υπεισέρχεται η έννοια του «συνωστισμού» (traffic) καθώς και του δικτύου κίνησης (π.χ. οδικό δίκτυο) των αντικειμένων, ενώ και η προστασία της ιδιωτικότητας αφορά πλέον αιτήσεις που λαμβάνουν χώρα σε πραγματικό χρόνο. Μια τυπική εφαρμογή που πληροί όλες τις παραπάνω προϋποθέσεις, αποτελούν τα συστήματα παροχής υπηρεσιών θέσης. Τα συστήματα αυτά έχουν αυξημένες απαιτήσεις σε υπολογιστικό κόστος, ικανότητα κλιμάκωσης (υποστήριξη πολλών χρηστών και ταυτόχρονων αιτήσεων για υπηρεσίες θέσης), ταχύτητα απόκρισης, ανοχή σε σφάλματα και διαθεσιμότητα. Ως μέρος της παρούσας διατριβής προτείνουμε ένα σύνολο μεθοδολογιών για την προστασία της ιδιωτικότητας (ως καταγραφή θέσης από την οποία έλαβε χώρα μια αίτηση για υπηρεσία θέσης ή/και τροχιά του κινούμενου αντικειμένου μέχρι την παροχή της ζητούμενης υπηρεσίας) κατά την ελεύθερη κίνηση των αντικειμένων στο σύστημα, καθώς επίσης και κατά την κίνηση των αντικειμένων πάνω σε πραγματικά οδικά δίκτυα. Προκειμένου να καλύψουμε τις προαναφερθείσες ανάγκες ενός τέτοιου συστήματος, χρησιμοποιούμε τις χωρικές δυνατότητες των σύγχρονων ΣΔΒΔ όπως αυτό της Oracle, της DB2 και του SQLServer. Τέλος, προτείνουμε ένα ολοκληρωμένο σύστημα, το PLOT, το οποίο αποτελεί την πρώτη εργαλειοθήκη που έχει προταθεί ως σήμερα για την υλοποίηση νέων αλγορίθμων προστασίας θέσης/τροχιάς σε υπηρεσίες θέσης, καθώς επίσης και για τη σύγκριση τεχνικών προστασίας θέσης/τροχιάς ως προς ένα σύνολο παραμέτρων.

Οι βασικότερες συνεισφορές της παρούσας διατριβής είναι οι ακόλουθες:

- Προτείνουμε τις πρώτες τεχνικές στον τομέα για την «ακριβή» απόκρυψη ευαίσθητης γνώσης κατά την εξαγωγή συχνών στοιχειοσυνόλων και κανόνων συσχέτισης από δεδομένα συναλλαγών. Οι ως σήμερα προτεινόμενες τεχνικές ήταν ευρεστικής φύσεως με αποτέλεσμα αρκετά συχνά να αποτυγχάνουν να

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

ανακαλύψουν μια λύση χωρίς παρενέργειες και με μικρή τροποποίηση της αρχικής βάσης. Οι αλγόριθμοι που προτείνουμε μοντελοποιούν τη διαδικασία απόκρυψης γνώσης ως ένα πρόβλημα βελτιστοποίησης, όπου οι περιορισμοί διαχωρίζουν τα συχνά από τα μη συχνά στοιχειοσύνολα στη νέα βάση προκειμένου να μην υπάρχουν παρενέργειες κατά την απόκρυψη. Επιπλέον, το κριτήριο βελτιστοποίησης του παραγόμενου προβλήματος περιορισμών διασφαλίζει ότι η τροποποίηση των συναλλαγών της αρχικής βάσης θα είναι ελάχιστη.

- Προτείνουμε ένα ολοκληρωμένο σύστημα ελέγχου επερωτήσεων το οποίο επιτρέπει σε ένα σύνολο χρηστών να θέσουν επερωτήσεις αναφορικά με τροχιές που βρίσκονται αποθηκευμένες σε κάποια βάση δεδομένων. Οι ως σήμερα τεχνικές προσέφεραν μηχανισμούς ασφαλείας για την προστασία των βάσεων δεδομένων σε στατιστικές βάσεις με δεδομένα συναλλαγών. Συγκεκριμένα, οι υπάρχουσες τεχνικές στον τομέα αυτό μπορούν να ταξινομηθούν στις ακόλουθες ομάδες: (α) εννοιολογικές, (β) περιορισμού επερωτήσεων, (γ) εισαγωγής θορύβου στα δεδομένα, και (δ) εισαγωγής θορύβου στην απάντηση. Όλες οι προαναφερθείσες κατηγορίες αφορούν αποκλειστικά αριθμητικές ή/και συναθροιστικές επερωτήσεις σε στατιστικά δεδομένα και επομένως καμιά άλλη πληροφορία από τη βάση δεδομένων δε γνωστοποιείται στον ερωτώντα. Το προτεινόμενο σύστημα επεκτείνει την υπάρχουσα έρευνα στον τομέα των περιοριστικών τεχνικών προσφέροντας μια πληθώρα επιτρεπόμενων τύπων επερωτήσεων, τόσο για τροχιές κινούμενων αντικειμένων, όσο και για συμβατικά είδη δεδομένων. Παράλληλα, αποτρέπει ένα σύνολο επιθέσεων στο σύστημα για την παρακολούθηση των τροχιών-χρηστών.

- Προτείνουμε ένα σύνολο αλγορίθμων με στόχο την παροχή υπηρεσιών θέσης τόσο στα πλαίσια ελεύθερης κίνησης των αντικειμένων όσο και στα πλαίσια δικτυακών υποδομών (όπως είναι π.χ. ένα οδικό δίκτυο), όπου ορίζεται ένα σύνολο από επιτρεπόμενες (δυνατές) και μη επιτρεπόμενες διαδρομές. Οι ως σήμερα τεχνικές

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

αφορούσαν ελεύθερη κίνηση των αντικειμένων και αγνοούσαν την ύπαρξη κάποιας δικτυακής υποδομής η οποία επιφέρει ένα σύνολο ευεργετικών αποφάσεων αναφορικά με το επίπεδο προστασίας της ιδιωτικότητας του κάθε χρήστη στο σύστημα. Επιπροσθέτως, εξετάζουμε την εύρεση συχνών (χωροχρονικών) προτύπων κίνησης για κάθε χρήστη στα πλαίσια μιας δικτυακής υποδομής και το διαχωρισμό των προτύπων αυτών σε ασφαλείς και ανασφαλείς διαδρομές για κάθε χρήστη. Η γνώση των ανασφαλών διαδρομών κάθε χρήστη στο σύστημα επιτρέπει στους αλγορίθμους μας να εγγυηθούν την καλύτερη προστασία του χρήστη κατά τη χρήση υπηρεσιών θέσης.

- Τέλος, προτείνουμε το πρώτο ολοκληρωμένο σύστημα που παρέχει την κατάλληλη υποδομή για τη δημιουργία νέων (κεντρικοποιημένων) αλγορίθμων προστασίας θέσης και τροχιάς κινούμενων αντικειμένων, καθώς επίσης και για την εις βάθος σύγκριση ενός συνόλου μεθοδολογιών προστασίας θέσης/τροχιάς με τη βοήθεια πολλαπλών μέτρων σύγκρισης. Οι έως σήμερα προτεινόμενες τεχνικές για την προστασία θέσης/τροχιάς κινούμενων αντικειμένων δε βασίζονταν σε κάποιο κοινό πλαίσιο το οποίο θα μπορούσε να χρησιμοποιήσει κανείς προκειμένου να προχωρήσει στη σύγκρισή τους. Επιπλέον, η δημιουργία νέων αλγορίθμων στον τομέα αυτό ήταν δύσκολη λόγω της έλλειψης κάποιας υποδομής που θα παρείχε τις βασικές δομές και τις πρωταρχικές συναρτήσεις τις οποίες θα μπορούσε να χρησιμοποιήσει κανείς προκειμένου να διευκολυνθεί στην κατασκευή νέων αλγορίθμων. Το σύστημα που προτείνουμε προσφέρει μια ολοκληρωμένη λύση στα παραπάνω ζητήματα.

Στο Παράρτημα που ακολουθεί παρέχεται λεπτομερής ανάλυση όλων όσων αναφέρθηκαν παραπάνω.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

ΠΑΡΑΡΤΗΜΑ

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

From Itemsets Through Trajectories to Location Based Services: A Knowledge Hiding Privacy Approach

by

Aris Gkoulalas-Divanis, BSc,MS

Dissertation

Presented to the Faculty of the Graduate School of the

Department of Computer & Communication Engineering

in the University of Thessaly in Greece

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

University of Thessaly

January 2009

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Dedicated to my family

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Acknowledgments

First of all, my enormous debt of gratitude goes to my thesis supervisor, Professor Elias N. Houstis, without whom this research effort would not have begun. I am indebted to him on many levels and for many reasons. Throughout the period of my PhD studies, Professor Houstis was there for me to actively support and guide me towards taking the best possible decisions with respect to my future. I am especially thankful to him for all the help he provided me throughout these years.

This dissertation would not have been possible without the expert and constant guidance of Professor Vassilios S. Verykios to whom I am deeply indebted. His excellent supervision and guidance throughout the period of my PhD studies, helped me to become fascinated about the research area of privacy preserving data mining and to develop to a much better researcher. It has been a great privilege for me to work under his guidance and I am really looking forward to our continuing collaboration in the future.

It would be an important omission not to explicitly state how grateful I am to Professor Panayiotis Bozanis both for his contribution towards the preparation of our joint publications, as well as for the time that he devoted to help and advise me all the years with respect to the procedural aspects of the PhD.

I also want to express my deep gratitude to my thesis committee members, Professors Panayiotis Bozanis, Aspassia Daskalopulu, Catherine Houstis, Yannis Theodoridis and Yannis Manolopoulos for their meticulous reading of my dissertation and for providing a lot of constructive comments that helped me to improve its quality.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

I am deeply thankful to Professor Leandros Tassiulas for offering me financial support during the last five months of my PhD studies in the University of Thessaly. Moreover, my sincere gratitude goes to Professor Panos Kalnis who helped me with the preparation of the survey in location based services (provided in the third part of this dissertation), as well as to Professor Mohamed Mokbel for offering to review my dissertation.

The research that is presented in the second and the third part of this dissertation was supported in part by grants from the EU project GeoPKDD (IST–6FP–014915). I would like to thank the project coordinators, Professors Dino Pedreschi and Fosca Giannotti, as well as my fellow researchers Dr. Francesco Bonchi and Dr. Nikos Pelekis, and the Professors Yücel Saygin and Maria Luisa Damiani, for our great collaboration that led to the exchange of lots of interesting and useful ideas, as well as some joint publications.

I also wish to thank my fellow PhD students and friends: Dimitris Syrivelis ($\mathcal{J}$), George Giannikis, Nikos Tziritas, Manos Koutsoubelias, George Georkakoudis and Apostolos Apostolaras, as well as Dr. Alexandros Karypidis. They each helped make my time in the PhD program more fun and interesting. I wish them the best.

Finally, I thank my parents for instilling in me confidence and a drive for pursuing my PhD. Their constant ethical support and encouragement was invaluable throughout the period of these years. The least I can do is to dedicate this dissertation to them.

ARIS GKOULALAS-DIVANIS

University of Thessaly

January 2009

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

From Itemsets Through Trajectories to Location Based Services: A Knowledge Hiding Privacy Approach

Aris Gkoulalas-Divanis, Ph.D.
University of Thessaly, 2009

Supervisor: E. N. Houstis
Co-supervisor: V. S. Verykios

Since its inception in 2000, privacy preserving data mining has gained increasing popularity in the data mining community. This line of research can be primarily attributed to the growing concern of individuals and organizations regarding the violation of privacy in the mining of their data by the existing data mining technology. As a result, a whole new body of research emerged, providing novel approaches for the mining of data, while prohibiting the leakage of private and sensitive information.

In this dissertation, we investigate a set of methodologies for the preservation of privacy in different contexts, data domains, and application scenarios. The dissertation is divided into three parts: In the first part, we investigate methodologies for the hiding of sensitive knowledge in the form of association rules, extracted from large transactional databases. Our research in this area led to the proposal of a new direction of approaches that guarantee optimality in the hiding solution by introducing the least side–effects and by causing minimal distortion to the original data.

In the second part of the dissertation, we extend the applicability area of association rule hiding (and, generally, semantic knowledge hiding) by applying similar techniques for the hiding of various types of temporally and spatially annotated data. Trajectory data leads to a more powerful kind of knowledge than the one that is produced by association rules and

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

transactional data. First, there exists an ordering of the elements in the trajectories, thus we are faced with sequential data. Second, since mobility data is of spatiotemporal nature, both the spatial and the temporal dimension of the data require special handling. Our contribution in this part of the thesis is a privacy aware trajectory tracking query engine, which enables a set of untrusted end users to query trajectory data that is stored in a database. The proposed engine guarantees that the information that is returned to the end users as part of a query, does not violate the privacy of the users whose movement in recorded in the database. Moreover, special care is taken to shield the database against different types of attacks.

In the last part of the dissertation, we go even further by investigating trajectory hiding in a dynamic, real–time environment. In particular, we consider real–time (rather than historical) traffic data produced by a number of users who are on the move. Each user is equipped with a mobile device that regularly transmits his/her location to a given station. Users subscribe to a set of services that depend on user location and are free to request any service at any time. The research problem we investigate deals with the offering of the requested services in a privacy aware manner, so that the identity of the requester is adequately protected. We have contributed a set of methodologies that protect the trajectory of the requester from the time of request, until the provision of the requested location based service. To shield the identity of the requester, we use his/her historical movement traces, as collected by the system, to build mobility patterns that capture the locations and times where his/her privacy is in danger. Then, we derive a set of privacy methodologies both for unconstrained and network–constrained user movement, which generalize (anonymize) the actual location of request as well as the subsequent locations of the user until the completion of the service. Moreover, we present PLOT, the first contributed open–ended toolbox for the offering of privacy in location based services.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Contents

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

List of Tables

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

List of Figures

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

List of Algorithms

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Part I

Association Rule Hiding in Transactional Data

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 1

Introduction

The significant advances in data collection and data storage technologies have provided the means for the inexpensive storage of enormous amounts of transactional data in data warehouses that reside in companies and public organizations. Apart from the benefit of using this data per se (e.g., for keeping up to date profiles of the customers and their purchases, maintaining a list of the available products, their quantities and price, etc), the mining of these datasets with the existing data mining tools can reveal invaluable knowledge that was unknown to the data holder beforehand. The extracted knowledge patterns can provide insight to the data holders as well as be invaluable in tasks such as decision making and strategic planning. Moreover, companies are often willing to collaborate with other entities who conduct similar business, towards the mutual benefit of their businesses. Significant knowledge patterns can be derived and shared among the collaborative partners through the aggregate mining of their datasets. Furthermore, public sector organizations and civilian federal agencies usually have to share a portion of their collected data or knowledge with other organizations having a similar purpose, or even make this data and knowledge public. For example, in the United States, the National Institutes of Health (NIH) [1] endorses research that leads to significant findings which improve human health and provides a set of guidelines which sanction the timely dissemination of NIH–supported research findings for use by other researchers.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

As it becomes evident, there exists an extended set of application scenarios in which data or knowledge derived from the data has to be shared with other (possibly untrusted) entities. The sharing of data and/or knowledge may come at a cost to privacy, primarily due to two reasons: (i) if the data refers to individuals (e.g., as in customers' market basket data) then the disclosure of this data or any knowledge extracted from the data can potentially violate the privacy of the individuals if their identity is revealed to untrusted third parties, and (ii) if the data regards business (or organizational) information, then the disclosure of this data or any knowledge extracted from the data may potentially reveal sensitive trade secrets, whose knowledge can provide a significant advantage to business competitors and thus can cause the data holder to lose business over his/her peers. The aforementioned privacy issues in the course of data mining are amplified due to the fact that untrusted entities (adversaries) may utilize other external and publicly available sources of information (e.g., the yellow pages, public reports) in conjunction with the released data or knowledge, in order to reveal any protected sensitive information.

Since the pioneering work of Agrawal and Srikant [15], and Lindell and Pinkas [113], several approaches have been proposed for the offering of privacy in data mining. Most existing approaches can be classified along two broad categories: (i) *data hiding* methodologies that aim at the removal of confidential or private information from the data prior to its disclosure, and (ii) *knowledge hiding* methodologies that protect the sensitive data mining results (i.e. the extracted knowledge patterns) that were produced by the application of data mining. The first category refers to methodologies that apply perturbation, sampling, generalization/suppression, transformation, etc. techniques to the original datasets in order to generate their sanitized counterparts that can be safely disclosed to untrusted third parties. The goal of this category of approaches is to enable the data miner to get accurate data mining results when is not provided with the real data. On the other hand, the second category deals with distortion and blocking techniques that prohibit the disclosure of sensitive knowledge patterns derived through the application of data mining algorithms, as well as techniques for downgrading the effectiveness of the classifiers in

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

classification tasks, such that they do not reveal any sensitive knowledge.

In this part of the dissertation, we focus on the knowledge hiding thread and study a specific class of methods in the knowledge hiding area, known as *frequent itemset* and *association rule hiding*[1]. “Association rule hiding” (a term used for brevity instead of the longer title “frequent itemset and association rule hiding”) has been mentioned for the first time in 1999 in a workshop paper by Atallah et al. [20]. The authors in [20], tried to apply general ideas regarding the implications of data mining in security and privacy of information – first presented by Clifton and Marks in [43] – to the association rule mining [14] framework. Clifton and Marks following the suggestions of D.E. O’Leary [132] – who was the very first to point out the security and privacy breaches that originate from data mining algorithms – indicated the need to consider different data mining approaches under the prism of preserving the privacy of information. Along these lines, they proposed a number of solutions like fuzification of the source database, limiting access to the source database, as well as releasing of samples instead of the entire database.

1.1 Privacy Issues and Research Challenges

The following scenario, borrowed from the work of Verykios et al. [172], exemplifies the necessity of applying association rule hiding algorithms to protect sensitive knowledge. Let us suppose that we are negotiating with Dedtrees Paper Company, as purchasing directors of BigMart, a large supermarket chain. They offer their products in reduced prices, provided that we agree to give them access to our database of customer purchases. We accept the deal and Dedtrees starts mining our data. By using an association rule mining tool, they find that people who purchase skim milk also purchase Green Paper. Dedtrees now runs a coupon marketing campaign offering a 50 cents discount on skim milk with every purchase of a Dedtrees product. The campaign cuts heavily into the sales of Green Paper, which increases the prices to us, based on the lower sales. During our next negotiation with Dedtrees, we find

[1]Other classes of methods under the same area, include classification rule hiding, clustering model hiding, sequence hiding, and so on and so forth. An overview of some of these methodologies can be found in [11, 76, 168]. Ref. [173] provides an extended survey of association rule hiding methods for privacy.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

out that with reduced competition they are unwilling to offer to us a low price. Finally, we start losing business to our competitors, who were able to negotiate a better deal with Green Paper. In other words, the aforementioned scenario indicates that BigMart should sanitize competitive information (and other important corporate secrets of course) before delivering their database to Dedtrees, so that Dedtrees does not monopolize the paper market.

We should emphasize here that the association rule hiding problem can be considered as a variation of the well known *database inference control* problem [50] in statistical and multilevel databases. The primary goal in database inference control is to protect access to sensitive information that can be obtained through nonsensitive data and inference rules. In association rule hiding we consider that it is not the data but the sensitive rules that create a breach to privacy. Given a set of sensitive association rules, which are specified by the security administrator, the task of the association rule hiding algorithms is to sanitize the data so that the association rule mining algorithms that will be applied to this data (i) will be incapable of discovering the sensitive rules under certain parameter settings, and (ii) will be able to mine all the nonsensitive rules and no other rules. The challenge that arises in association rule hiding can be stated as follows: *How can we expect reasonable data mining results that will allow for correct decision making when, at the same time, prohibit the disclosure of all the sensitive information?*

Association rule hiding algorithms are fundamental in providing a solution to this problem. They accomplish this by causing a small deviation to the original database which blocks the production of the sensitive rules. What differentiates the quality of one approach from that of another is the actual harm that is introduced to the dataset (in terms of distortion) as a result of the hiding process. Ideally, the hiding process should be accomplished in such a way that the nonsensitive knowledge remains, to the highest possible degree, intact. Another problem has been investigated recently, which even though it is not targeted to addressing privacy issues per se, it does give a special solution to the association rule hiding problem. The problem is known as *inverse frequent itemset mining* [124].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

1.2 Organization of the Rest of Part I

The rest of this part of the dissertation is organized as follows: In Chapter 2 we formalize the problem and provide the necessary background for its proper understanding. Then, in Chapter 3 we survey the state–of–the–art methodologies that have been proposed for association rule hiding. Chapter 4 summarizes our contributions in this area of research, which consist of three novel algorithms for the identification of optimal hiding solutions, as well as a parallelization framework for reducing the time that is necessary for the computation of the hiding solutions. In Chapter 5, we highlight the border revision process which is essential for the identification of optimal hiding solutions and is employed in all our proposed hiding methodologies. Chapters 6, 7 and 8 present in detail the three methodologies that we have contributed for optimal association rule hiding. Following that, in Chapter 9 we provide a novel framework that can be employed for the parallelization of all the three hiding methodologies, allowing the identification of optimal hiding solutions in a reduced time. Chapter 10 presents a strategy that enables the data holder to quantify the level of privacy that he/she wishes to achieve when hiding specific sensitive rules by using any of our proposed methodologies. Finally, Chapter 11 concludes this part of the dissertation by presenting a summary along with some potential future work.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 2

Background and Problem Formulation

In this chapter we provide all the necessary background which is essential for the proper understanding of the related work, as well as our contributions (i.e. [64, 65, 67, 68, 70, 72]) provided in detail in the chapters that follow. Following that, we discuss the properties of an optimal hiding solution and formally set out the problem at hand.

2.1 Terminology and Preliminaries

Association rule mining is the process involving the discovery of sets of items (also known as *itemsets*) that frequently co–occur in a transactional database so as to produce association rules that hold for the data. Each association rule is defined as an implication of the form $A \Rightarrow B$, where A, B are frequent itemsets in the transactional database, such that $A \cap B = \varnothing$ (i.e. A and B are disjoint). The itemset $A \cup B$ that leads to the generation of an association rule is known as the *generating* itemset and consists of two parts, the *Left Hand Side* (LHS) or antecedent, which is the part on the left of the arrow of the rule (here A), and the *Right Hand Side* (RHS) or consequent, which is the part on the right of the arrow of the rule (here B). Two metrics, *support* and *confidence*, are incorporated to the task of association

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

rule mining to drive the generation of association rules and expose only the ones that are expected to be of interest to the owner of the data. In particular, the measure of support eliminates rules that are not adequately backed up by the transactions of the dataset and thus are expected to be uninteresting, i.e. occurring simply by chance. On the other hand, confidence measures the reliability of the inference that is made by the rule [163] — a low value of confidence in a rule $A \Rightarrow B$ shows that it is not very likely for itemset B to be present in transactions containing itemset A.

Frequent itemset mining (as well as association rule mining) was introduced by Agrawal, et al. [12, 14] in 1993 and is defined as follows: Let $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$ be a finite set of literals, called *items*, where M denotes the cardinality of the set. Any subset $I \subseteq \mathcal{I}$ is an *itemset* over $\mathcal{I}$. A k–itemset is an itemset of *length* (equiv. of *size*) k, i.e. an itemset consisting of k items. A transaction T over $\mathcal{I}$ is a pair $T = (tid, I)$, where I is the itemset and tid is a unique identifier, used to distinguish among transactions that correspond to the same itemset. A transaction database $D = \{T_1, T_2, \ldots, T_N\}$ over $\mathcal{I}$ is a $N \times M$ table consisting of N transactions over $\mathcal{I}$ carrying different identifiers, where entry $T_{nm} = 1$ if and only if the m-th item ($m \in [1, M]$) appears in the n-th transaction ($n \in [1, N]$). Otherwise, $T_{nm} = 0$. A transaction $T = (tid, J)$ supports an itemset I over $\mathcal{I}$, if $I \subseteq J$. Let D_I denote the supporting transactions of itemset I in database D. Furthermore, let S be a set of items. Notation $\wp(S)$ denotes the powerset of S, which is the set of all subsets of S.

Given an itemset I over $\mathcal{I}$ in D, $\sup(I, D)$ denotes the number of transactions $T \in D$ that support I and $\text{freq}(I, D)$ denotes the fraction of transactions in D that support I. An itemset I is called *large* or *frequent* in database D, if and only if, its frequency in D is at least equal to a minimum threshold *mfreq*. Equivalently, I is large in D, if and only, $\sup(I, D) \geq msup$, where $msup = mfreq \times N$. All the itemsets having a frequency lower than *mfreq* are called *infrequent*.

The theory of border is also important in our discussion. Let $\mathcal{F}_D = \{I \subseteq \mathcal{I} : \text{freq}(I, D) \geq mfreq\}$ be the set of all frequent itemsets in D, and $\mathcal{P} = \wp(\mathcal{I})$ be the set of all patterns in the lattice of D. The *positive border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d^+(\mathcal{F}_D)$, consists of all the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

maximally frequent patterns in $\mathcal{P}$, i.e. all the patterns in $\mathcal{F}_D$, whose all proper supersets are infrequent. Formally, $\mathcal{B}d^+(\mathcal{F}_D) = \{I \in \mathcal{F}_D |$ for all $J \in \mathcal{P}$ with $I \subset J$ we have that $J \notin \mathcal{F}_D\}$. Respectively, the *negative border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d^-(\mathcal{F}_D)$, consists of all the minimally infrequent patterns in $\mathcal{P}$, i.e. all the patterns in $\mathcal{P} \backslash \mathcal{F}_D$, whose all proper subsets are frequent. Formally, $\mathcal{B}d^-(\mathcal{F}_D) = \{I \in \mathcal{P} \backslash \mathcal{F}_\mathcal{D} |$ for all $J \subset I$ we have that $J \in \mathcal{F}_D\}$. Finally, the *border* of $\mathcal{F}_D$, denoted as $\mathcal{B}d(\mathcal{F}_D)$, is the union of these two sets: $\mathcal{B}d(\mathcal{F}_D) = \mathcal{B}d^+(\mathcal{F}_D) \cup \mathcal{B}d^-(\mathcal{F}_D)$. Borders allow for a condense representation of the itemsets lattice, identifying the key itemsets which separate all frequent patterns from their infrequent counterparts. More details regarding the border theory and its underlying concepts can be found in the work of Mannila and Toivonen [117]. The need to revise the border so as to hide certain sensitive association rules is paramount to all our proposed hiding algorithms. A presentation of the theory of border revision is given in the work of Moustakides and Verykios [128], while efficient algorithms for the computation of the borders can be found in [64, 117]. Chapter 5 also highlights the process of border revision as well as it provides a set of algorithms for the efficient computation of both the original and the revised borders.

2.2 Problem Formulation and Statement

Knowledge hiding, in the context of association rule mining, aims at sanitizing the original database in a way that at least one of the following goals is accomplished: (i) no rule that is considered as sensitive from the owner's perspective, can be revealed from the sanitized database when this is mined at pre–specified thresholds of confidence and support (or at any value higher than these thresholds), (ii) all the nonsensitive rules can be successfully mined in the sanitized database at pre–specified thresholds of confidence and support (or higher), and (iii) no rule that was not found in the original database can be found its sanitized counterpart when mining this database at pre–specified thresholds of confidence and support (or higher). The first goal requires all the sensitive rules to disappear in the sanitized database. A hiding solution that achieves this goal is called *feasible*. The second goal simply states that there should be no *lost rules* in the sanitized database. The third goal says that no *false*

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

rules (also known as *ghost rules*) should be produced when mining the sanitized database as a side–effect of the sanitization process. Generally speaking, in the typical case hiding scenario, the sanitization process has to be accomplished in a way that minimally affects the original database, preserves the general patterns and trends, and achieves to conceal the sensitive knowledge. A solution that addresses all the above mentioned goals (i.e. it is feasible and introduces no side–effects in the sanitization process) is called *exact*. All the association rule hiding approaches that we have contributed and are presented in detail in Chapters 6, 7 and 8 aim at identifying exact hiding solutions. We should point out that we address the problem of sensitive knowledge hiding by properly covering up a user–defined set of itemsets that lead to the generation of sensitive rules. From here on, we refer to these itemsets as *sensitive* and our goal is to ensure that they cannot be mined from the released database at the same (or higher) thresholds of support and confidence. As we demonstrate in Chapter 3 the hiding of sensitive association rules through the hiding of their generating itemsets is a very common strategy that is adopted by the overwhelming majority of researchers. By ensuring that the itemsets that lead to the generation of a sensitive rule become insignificant in the released dataset, the data owner can be certain that his/her sensitive knowledge is adequately protected from untrusted third parties.

Based on the notation presented in Section 2.1, we proceed to our problem's statement. In what follows, we assume that we are provided with a database $\mathcal{D}_O$, consisting of N transactions, and a threshold *mfreq* set by the owner of the data. After performing frequent itemset mining in $\mathcal{D}_O$ with *mfreq*, we yield a set of frequent patterns, denoted as $\mathcal{F}_{\mathcal{D}_O}$, among which a subset $\mathcal{S}$ contains patterns which are considered to be *sensitive* from the owner's perspective.

Given the set of sensitive itemsets $\mathcal{S}$, we define the set $\mathcal{S}_{min} = \{I \in \mathcal{S} | \text{ for all } J \subset I, \quad J \notin \mathcal{S}\}$ that contains all the *minimal* sensitive itemsets from $\mathcal{S}$, and the set $\mathcal{S}_{max} = \{I \in \mathcal{F}_{\mathcal{D}_O} | \exists J \in \mathcal{S}_{min}, J \subseteq I\}$ that contains all the itemsets of $\mathcal{S}_{min}$ along with their frequent supersets. Our goal is to construct a new, sanitized database $\mathcal{D}$, which achieves to protect the sensitive itemsets from disclosure, while leaving intact the nonsensitive itemsets existing in

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$\mathcal{F}_{\mathcal{D}_O}$. The hiding of a sensitive itemset corresponds to a lowering of its statistical significance, depicted in terms of support, in the resulting database. To be more specific, what we want to achieve is to minimally modify the original dataset $\mathcal{D}_O$ in such a way that when the sanitized dataset $\mathcal{D}$ is mined at the same (or a higher) level of support, the frequent patterns that are discovered are *exactly* those in $\mathcal{F}'_{\mathcal{D}} = \mathcal{F}_{\mathcal{D}_O} - \mathcal{S}_{max}$. We call this set *ideal* as it pertains to an exact hiding solution. To formulate this set we use $\mathcal{S}_{max}$ instead of $\mathcal{S}$ since, due to the *apriori principle* [14], hiding of the itemsets in $\mathcal{S}$ has as a consequence the hiding of all itemsets in $\mathcal{S}_{max}$. When constructed, database $\mathcal{D}$ can be safely released since it protects the sensitive knowledge. The key step in the association rule hiding methodologies lies on the *minimum impact*[1] of $\mathcal{D}_O$ to construct $\mathcal{D}$. All the solutions that are presented in this part of the dissertation are based on this principle. Furthermore, in all our proposed methodologies we are interested in computing exact hiding solutions that cause the least possible distortion to the original database; we term these solutions as *ideal* or *optimal*. Finally, in the case of hiding problems where an exact solution cannot be attained we need to result to the best possible non–exact feasible solution. We call such a solution *approximate*.

[1]The impact is minimum when from the frequent itemsets in $\mathcal{D}_O$, only the itemsets in $\mathcal{S}_{max}$ become infrequent in the released database $\mathcal{D}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 3

Related Work

In this chapter, we present a taxonomy of frequent itemset and association rule hiding algorithms after having reviewed a large collection of independent works in the area. In order to be able to classify the various algorithms, we propose a set of orthogonal dimensions based on which we will present the existing approaches. As a first dimension, we consider whether the hiding algorithm uses the support or the confidence of the rule to driving the hiding process. In this way, we separate the hiding algorithms into *support*–based and *confidence*–based. The second dimension in our classification is related to the modification in the raw data that is caused by the hiding algorithm. The two forms of the modification comprise the *distortion* and the *blocking* of the original values. Distortion is the process of replacing 1's by 0's and 0's by 1's (i.e. excluding or including items in selected transactions), while blocking refers to replacing original values by question marks (unknowns). The third dimension, refers to whether a single rule or a set of rules can be hidden in every iteration of the hiding algorithms. Based on this criterion we differentiate hiding algorithms into *single rule* and *multiple rule* schemes. The fourth dimension has to do with the nature of the hiding algorithm, which can be either *heuristic* or *exact*. Heuristic techniques rely on optimizing certain sub–goals in the hiding process, while they do not guarantee optimality in the hiding solution. The formulation of the association rule hiding problem, presented in Chapter 2, implies that there are two specific sub–goals that need to be attained by every

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

association rule hiding algorithm. The first sub–goal, which is basically the most important, is to try to hide as many sensitive rules as possible. The second sub–goal is to manage to hide the sensitive rules by minimizing the possible side–effects. As side–effects in the hiding process, we consider (i) the number of data items affected by the hiding process, (ii) the number of nonsensitive rules which were accidentally hidden during the hiding process, and (iii) the number of rules which were created by the hiding process. Different hiding algorithms give different priorities to the satisfaction of the sub–goals presented, producing in this way a list of hiding primitives. Exact techniques rely on formulating the association rule hiding problem in such a way that a solution can be found that satisfies all the sub–goals. Of course there is a possibility that an exact approach fails to give a solution, and for this reason, some of the sub–goals need to be relaxed. However, this relaxation process is still part of the exact approach, which makes it different from the heuristic approaches.

3.1 Classes of Association Rule Hiding Algorithms

Association rule hiding algorithms can be divided into three distinct classes, namely *heuristic* approaches, *border–based* approaches and *exact* approaches. The first class of approaches involves efficient, fast algorithms that selectively sanitize a set of transactions from the database to hide the sensitive knowledge. Due to their efficiency and scalability, the heuristic approaches have been the focus of attention for the vast majority of researchers in the data mining field. However, there are several circumstances in which they suffer from undesirable side–effects that lead them to approximate hiding solutions.

The second class of approaches considers the task of sensitive rule hiding through modification of the original borders in the lattice of the frequent and the infrequent patterns in the database. In these schemes, the sensitive knowledge is hidden by enforcing the revised borders (which accommodate the hiding of the sensitive itemsets) in the sanitized database. The algorithms in this class differ both in the borders that they track and use for the hiding strategy, and in the methodology that they follow to enforce the revised borders in the sanitized database. The theory behind the revision of the borders is also critical to

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

our proposed association rule hiding methodologies and is presented in detail in Chapter 5.

Finally, the third class of approaches contains non–heuristic algorithms which conceive the hiding process as a constraints satisfaction problem that they solve by using integer or linear programming. The main difference of these approaches, compared to the previous ones, is the fact that the sanitization process guarantees optimality in the hiding solution, provided that an optimal hiding solution exists. On the negative side, these approaches are usually several orders of magnitude slower than the heuristic ones, especially due to the runtime of the integer or the linear programming solver.

3.1.1 Heuristic Approaches

In this section, we review support–based and confidence–based heuristic approaches, which are based on either distortion or blocking of the original values. Between these two categories of approaches, the distortion–based are the ones commonly adopted by the overwhelming majority of researchers.

Support–Based and Confidence–Based Distortion Schemes

Atallah, et al. [20] were the first to propose an algorithm for the hiding of sensitive association rules through the reduction in the support of their generating itemsets. The authors propose the construction of a lattice–like graph [155] in the database. Through this graph, the hiding of a large itemset, related to the existence of a sensitive rule, is achieved by a greedy iterative traversal of its immediate subsets, selection of the subset that has the maximum support among all candidates (therefore is less probable to be hidden) and setting of this itemset as the new candidate to be hidden. By iteratively following these steps, the algorithm identifies the 1–itemset ancestor of the initial sensitive itemset, having the highest support. Then, by identifying the supporting transactions for both the initial candidate and the currently identified 1–itemset, the algorithm removes the 1–itemset from the supporting transaction which affects the least number of 2–itemsets. In sequel, the algorithm propagates the results of this action to the affected itemsets in the graph. When hiding a

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

set of sensitive rules, the algorithm first sorts the corresponding large itemsets based on their support and then proceeds to hide them in a one–by–one fashion, using the methodology presented above. One of the most significant contributions of this work is the proof regarding the NP–hardness of finding an optimal sanitization of a dataset. On the negative side, the proposed approach is not interested in the extent of the loss of support for a large itemset, as long as it remains frequent in the sanitized outcome.

Dasseni, et al. [46] generalize the problem in the sense that they consider the hiding of both sensitive frequent itemsets and sensitive rules. The authors propose three single rule heuristic hiding approaches that are based on the reduction of either the support or the confidence of the sensitive rules, but not both. In all three approaches, the goal is to hide the sensitive rules while minimally affecting the support of the nonsensitive itemsets. The first two strategies reduce the confidence of the sensitive rule either (i) by increasing the support of the rule antecedent, through transactions that partially support it, until the rule confidence drops below the minimum confidence threshold, or (ii) by decreasing the frequency of the rule consequent through transactions that support the rule, until the rule confidence is below the minimum threshold. The third strategy decreases the frequency of a sensitive rule, by decreasing the support of either the antecedent or the rule consequent, until either the confidence or the support lies below the corresponding minimum threshold. A basic drawback of the proposed methodologies is the strong assumption that all the items appearing in a sensitive rule do not appear in any other sensitive rule. Under this assumption, hiding of the rules one at a time or altogether makes no difference. Moreover, since this work aims at hiding all the sensitive knowledge appearing in the dataset, it fails to avoid undesired side–effects, such as lost and false rules.

Verykios, et al. [172] extend the work of Dasseni, et al. [46] by improving and evaluating the hiding algorithms for their performance under different sizes of input datasets and different sets of sensitive rules. Moreover, the authors propose two heuristic algorithms that incorporate the third strategy presented earlier. The first of these algorithms protects the sensitive knowledge by hiding the item having the maximum support from the minimum

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

length transaction (i.e. the one with the least supporting items). The hiding of the generating itemsets of the sensitive rules is performed in a decreasing order of size and support and in a one–by–one fashion. Similarly to the first algorithm, the second algorithm first sorts the generating itemsets with respect to their size and support, and then hides them in a round–robin fashion as follows: First, for each generating itemset, a random ordering of its items and of its supporting transactions is attained. Then, the algorithm proceeds to remove the items from the corresponding transactions in a round–robin fashion, until the support of the sensitive itemset drops below the minimum support threshold. The intuition of hiding in a round–robin fashion is fairness and the proposed algorithm (although rather naïve) serves as a baseline for conducting a series of experiments.

Oliveira and Zaïane [133] were the first to introduce multiple rule hiding approaches. The proposed algorithms are efficient and require two scans of the database, regardless of the number of sensitive itemsets to hide. During the first scan, an index file is created to speed up the process of finding the sensitive transactions and to allow for an efficient retrieval of the data. In the second scan, the algorithms sanitize the database by selectively removing the least amount of individual items that accommodate the hiding of the sensitive knowledge. An interesting novelty of this work is the fact that the proposed methodology takes into account not only the impact of the sanitization on hiding the sensitive patterns, but also the impact related to the hiding of nonsensitive knowledge. Three item restriction–based algorithms (known as MinFIA, MaxFIA, and IGA) are proposed that selectively remove items from sensitive transactions. The first algorithm, MinFIA, proceeds as follows: For each restrictive pattern it identifies the supporting transactions and the item having the smallest support in the pattern (called the *victim item*). Then, by using a user–supplied disclosure threshold, it first sorts the identified transactions in ascending order of degree of conflict and then selects the number of transactions (among them) that need to be sanitized. Finally, from each selected transaction the algorithm removes the victim item. The MaxFIA algorithm proceeds exactly as the MinFIA with the only difference of selecting as the victim item the one that has the maximum support in the sensitive rule. Finally, IGA aims

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

at clustering the restricted patterns into groups that share the same itemsets. By identifying overlapping clusters, the algorithm proceeds to hide the corresponding sensitive patterns at once (based on the sensitive itemsets they share) and consequently reduces the impact on the released database.

A more efficient approach than the one of [133] and the work of [46, 152, 153] was proposed by Oliveira and Zaïane [134]. The proposed algorithm, called SWA, is an efficient, scalable, one–scan heuristic which aims at providing a balance between the needs for privacy and knowledge discovery in association rule hiding. It achieves to hide multiple rules in only one pass through the dataset, regardless of its size or the number of sensitive rules that need to be protected. The algorithm proceeds in five steps that are applied to every group of K transactions (thus formulating a window of size K) read from the original database. Firstly, the nonsensitive transactions are separated from the sensitive ones and copied directly to the sanitized database. For each sensitive rule, the item having the highest frequency is selected and its supporting transactions are identified. Then, a disclosure threshold ψ, potentially different for each sensitive rule, is used to capture the severity characterizing the release of the rule. Based on this threshold, SWA computes the number of supporting transactions that need to be sanitized for each rule and then sorts them in ascending order of length. For each selected transaction, the corresponding item is removed and then the transaction is copied to the sanitized database. SWA is experimentally shown to outperform the state–of–the–art heuristic approaches in terms of concealing all the sensitive rules, while maintaining high data utility of the released database.

Amiri [17] proposes three effective, multiple rule hiding heuristics that outperform SWA by offering higher data utility and lower distortion, at the expense of increased computational speed. Although similar in the philosophy to the previous approaches, the proposed methodologies do a better job in modeling the overall objective of a rule hiding algorithm. The first approach, called *Aggregate*, computes the union of the supporting transactions for all sensitive itemsets. Among them, the transaction that supports the most sensitive and the least nonsensitive itemsets is selected and expelled from the database. The same process is

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

repeated until all the sensitive itemsets are hidden. Similarly to this approach, the *Disaggregate* approach aims at removing individual items from transactions, rather than removing the entire transaction. It achieves that by computing the union of all transactions supporting sensitive itemsets and then, for each transaction and supporting item, by calculating the number of sensitive and nonsensitive itemsets that will be affected if this item is removed from the transaction. Finally, it chooses to remove the item from the transaction that will affect the most sensitive and the least nonsensitive itemsets. The third approach, called *Hybrid*, is a combination of the previous two; it uses the *Aggregate* approach to identify the sensitive transactions and the *Disaggregate* approach to selectively delete items of these transactions, until the sensitive knowledge is hidden.

Wu, et al. [178] propose a sophisticated methodology that removes the assumption of [46] regarding the disjoint relation among the items of the various sensitive rules. Using set theory, the authors formalize a set of constraints related to the possible side–effects of the hiding process and allow item modifications to enforce these constraints. However, the existing correlation among the rules can make impossible the hiding of the sensitive knowledge, without the violation of any constraints. For this reason, the user is permitted to specify which of the constraints he/she considers more significant and relax the rest. A drawback of the approach is the simultaneous relaxation (without the users' consent) of the constraint regarding the hiding of all the sensitive itemsets. To accommodate for rule hiding, the new scheme defines a class of allowable modifications that are represented as templates and are selected in a one–by–one fashion. A template contains the item to be modified, the applied operation, the items to be preserved or removed from the transaction and coverage information regarding the number of rules that are affected. Based on this information the algorithm can select and apply only the templates that are considered as beneficial, since they cause the least amount of side–effects in the sanitized database.

Pontikakis, et al. [144] propose two distortion–based heuristics to selectively hide the sensitive rules. On the positive side, the proposed schemes use effective data structures for the representation of the rules and effectively prioritize the selection of transactions

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

for sanitization. However, in both algorithms the proposed hiding process may introduce a number of side–effects, either by generating rules which were previously unknown, or by eliminating existing nonsensitive rules. The first algorithm, called *Priority–based Distortion Algorithm* (PDA), reduces the confidence of a rule by reversing 1's to 0's in items belonging in its consequent. On the other hand, the second algorithm, called *Weight–based Sorting Distortion Algorithm* (WDA), concentrates on the optimization of the hiding process in an attempt to achieve the least side–effects and the minimum complexity. This is achieved through the use of priority values assigned to transactions based on weights. Regarding the quality of the hiding solution, the proposed schemes tend to produce solutions of comparable or slightly higher quality than the algorithms in [153] by generally introducing less side–effects. However, both algorithms are computationally demanding, with PDA requiring typically twice the time of the schemes in [153] to facilitate knowledge hiding.

Support–Based and Confidence–Based Blocking Schemes

Saygin, et al. [152,153] are the first to propose the use of *unknowns* (represented as question marks in the database) instead of transforming 1's to 0's and the opposite, for the hiding of sensitive association rules. As demonstrated in [152], the use of unknowns provides a safer alternative especially in critical real life applications where the distinction between "false" and "unknown" is vital. In their work, the authors introduce three simple heuristic approaches. The first approach, relies on the reduction in the support of the generating itemsets of the rule, while the other two rely on the reduction of the confidence of the rule, below the minimum thresholds. The definitions of both the support and the confidence measures are extended to capture the notion of an interval instead of being crisp values, while the algorithms consider both 0 and 1 values to use for hiding (in some proportion) so that it is difficult for an adversary to conclude upon the value hidden behind a question mark. A universal safety margin is applied to capture how much below the minimum thresholds should the new support and confidence of a sensitive rule lie, in order to consider that the rule is safely hidden. An important contribution of this work, apart from the methodology

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

itself, is a discussion regarding the effect of the algorithms towards the hiding of the sensitive knowledge, the possibility of reconstruction of the hidden patterns by an adversary, and the importance of choosing an adequate safety margin when hiding the sensitive rules.

Wang and Jafari [176] propose two modification schemes that incorporate unknowns and aim at the hiding of predictive association rules, i.e. rules containing the sensitive items on their LHS (rule antecedent). Both algorithms rely on the distortion of a portion of the database transactions to lower the confidence of the sensitive association rules. Compared to the work of Saygin, et al. [152, 153], the algorithms presented in [176] require a reduced number of database scans and have an efficient pruning strategy. However, by construction, they are assigned the task of hiding all the rules containing the sensitive items on their LHS, while the algorithms in the work of Saygin, et al. [152, 153] can hide any specific rule. The first strategy, called ISL, decreases the confidence of a rule by increasing the support of the itemset in its LHS. The second approach, called DSR, reduces the confidence of the rule by decreasing the support of the itemset in its RHS (rule consequent). Both algorithms experience the *item ordering effect* under which, based on the order that the sensitive items are hidden, the produced sanitized databases can be different. Moreover, the DSR algorithm seems to be more effective when the sensitive items have a high support.

Pontikakis, et al. [143] argue that the main disadvantage of a blocking algorithm is the fact that the dataset, apart from the blocked values (i.e. the unknowns), is not distorted. Thus, an adversary can disclose the hidden rules simply by identifying those generating itemsets that contain question marks and lead to rules with a maximum confidence that lies above the minimum confidence threshold. If the number of these rules is small then the probability of identifying the sensitive ones among them becomes high. To avoid this shortcoming, the authors propose a blocking algorithm that purposely creates rules that were not existent in the original dataset (i.e. ghost rules) and that their generating itemsets contain unknowns. Thus, the identification of the sensitive rules becomes harder since the adversary is unable to tell which of the rules that have a maximum confidence above the minimum threshold are the sensitive and which are the ghost ones. However, the introduction of ghost

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

rules leads to a decrement in the data quality of the sanitized outcome. In order to balance the trade–off between privacy and data loss, the proposed algorithm incorporates a safety margin that corresponds to the extent of sanitization that is performed to the database. The higher the safety margin the better the protection of the sensitive rules and the worse the data quality of the resulting sanitized database.

3.1.2 Border–Based Approaches

In this section, we review two border–based approaches that have been recently proposed for the hiding of sensitive association rules. The work of Sun and Yu [160] was the first to introduce the process of *border revision* for the hiding of sensitive association rules. In this paper, the authors propose a heuristic approach that uses the notion of the *border* (further analyzed in [117]) of the nonsensitive frequent itemsets to track the impact of altering transactions in the database. The proposed methodology, first computes the positive and the negative borders in the lattice of all the itemsets and then focuses on preserving the quality of the computed borders during the hiding process. The quality of the borders directly affects the quality of the sanitized database that is produced, which can be maintained by greedily selecting those modifications that lead to minimal side–effects. In the proposed heuristic strategy, a weight is assigned to each element of the expected positive border (which is the original positive border after it has been shaped up with the removal of the sensitive itemsets) in an attempt to quantify its vulnerability of being affected by item deletion. These weights are dynamically computed (during the sanitization process) as a function of the current support of the corresponding itemsets in the database. To reduce the support of a sensitive itemset from the negative border, the algorithm calculates the impact of the possible item deletions by computing the sum of the weights of the positive border elements that will be affected. Then, it proceeds to delete the candidate item that will have minimal impact on the positive border.

Moustakides and Verykios [128] follow a similar approach to [160] by proposing two heuristics that use both the revised positive and the revised negative borders, produced

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

by the removal of the sensitive itemsets and their supersets from the old frequent itemset lattice. The proposed algorithms try to remove from the database all the sensitive itemsets that belong to the revised negative border, while maintaining frequent all the itemsets of the revised positive border. For every item of a sensitive itemset, the algorithms list the set of positive border itemsets which depend on it. Then, from among all minimum border itemsets, the one with the highest support is selected as it is the one with the maximum distance from the border. This itemset, called the *max–min* itemset, determines the item through which the hiding of the sensitive itemset will incur. The proposed algorithms try to modify this item in such a way that the support of the max–min itemset is minimally affected. When hiding multiple itemsets, the algorithms perform the sanitization in a one–by–one fashion, starting from the itemsets that have lower supports. Through experimental evaluation, the proposed methodologies are shown to provide better hiding solutions when compared to those of [160], in the majority of the tested cases.

3.1.3 Exact Approaches

In this section, we review the exact approaches that have been proposed so far for the hiding of sensitive association rules. Exact approaches are typically capable of providing superior solutions when compared to heuristic methodologies, however at a high computational cost. They achieve this by formulating the hiding process as a *Constraints Satisfaction Problem* (CSP) and solving it by using an integer/linear programming solver. Thus, the sanitization of the dataset is performed as an atomic operation, which avoids the local minima issues experienced by the heuristic algorithms.

Menon, et al. [122] propose a methodology that consists of an exact and a heuristic part for the hiding of sensitive frequent itemsets. The exact part formulates a CSP with the objective of identifying the minimum number of transactions that need to be sanitized for the proper hiding of all the sensitive knowledge that appears in the original database. To avoid the NP–hardness issue, the authors reduce the problem size considering only the sensitive itemsets, requesting that their support remains below the minimum support threshold

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

in the sanitized database. The optimization process is driven through a criterion function that is inspired by the measure of accuracy [111]. Moreover, the constraints imposed in the integer programming formulation capture the number of supporting transactions that need to be sanitized for the hiding of each sensitive itemset. An integer programming solver is then applied to identify the best solution of the CSP and to derive the objective. In turn, this objective is provided as input to a heuristic sanitization algorithm that is assigned the task of identifying the corresponding number of transactions from the original database and performing their sanitization. An important contribution of the authors, apart from the algorithm itself, is a discussion over the possibility of parallelization of the exact part. As demonstrated, based on the underlying properties of the dataset to be sanitized, it is possible for the produced CSP to be decomposed into parts that are solved independently. Bearing in mind the exponential complexity involving the solution of a CSP, this process can drastically reduce the required computational time for the hiding of the sensitive knowledge.

Gkoulalas–Divanis and Verykios [64] propose the first exact (non–heuristic) approach for the hiding of sensitive rules that uses the itemsets belonging in the revised positive and the revised negative borders to identify the candidate itemsets for sanitization. Through a set of theorems, involving existing relations among itemsets, the authors achieve to further reduce the set of candidates to a small fraction of its original size. The hiding process is then performed by formulating a CSP in which the status (frequent vs infrequent) of each of the itemsets in the reduced set is controlled though a set of constraints. By using a process of constraints degree reduction, all the participating constraints in the CSP become linear and have no coefficients. Moreover, all the variables involved in the CSP are of binary nature. These facts allow for an efficient solution of the CSP. The provided solution is proved to lead to an exact (without any side–effects) hiding of the sensitive patterns. Furthermore, the proposed methodology is capable of identifying the exact solution that is also optimal (i.e. it causes the least distortion to the original database). A heuristic approach that relaxes the initial CSP to allow for the identification of a good approximate solution, is applied only when the initial CSP is infeasible and thus an exact solution cannot

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

be attained. An important property of the proposed methodology is the fact that although the problem formulation leads to a CSP with a size that is typically larger than the one of [122], (i) the hiding algorithm achieves good efficiency, and (ii) there is no need to result to heuristics — i.e. potentially suboptimal solutions — if the CSP is feasible. We feel that this can be attributed both to the binary nature of the variables and to the linear (and without coefficients) constraints involved in the formulation of the CSP.

In [70] the authors introduce a two–phase iterative methodology that augments the functionality of [64] to allow for the identification of exact hiding solutions to a wider range of problems than the ones that can be handled by [64]. The first phase of the algorithm uses [64] to hide the sensitive knowledge. If the algorithm succeeds to find an exact solution then the process is terminated and the corresponding sanitized database is returned to the data owner. On the other hand, if the first phase is unable to identify an exact solution (i.e. if the solution that is found suffers from side–effects), then the second phase takes over. The purpose of the second phase is to reduce (or completely remove) the side–effects of the hiding solution that was found by the first phase of the algorithm. The two phases of the proposed algorithm iterate until either (i) an exact solution is found, in which case the solution is returned to the data holder, or (ii) a pre–specified number of iterations have taken place. In both cases the quality of the identified hiding solution is bound to be superior or (in the worst case) equal to the one attained by [64].

A similar approach to [64] that operates by applying an extension to the original database instead of modifying existing transactions to accommodate knowledge hiding, is proposed by Gkoulalas–Divanis and Verykios in [65, 68]. The extended portion of the dataset contains a set of carefully crafted transactions that achieve to lower the importance of the sensitive patterns to a degree that they become uninteresting from the perspective of the data mining algorithm, while minimally affecting the importance of the nonsensitive ones. Following the work of [64], the proposed algorithm is exact in nature, while guaranteeing minimal extension of the original database. The solution methodology relies on the same underlying principles as [64] to (i) model the hiding process as a CSP that controls

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the status (frequent vs infrequent) of the itemsets that belong to the revised positive and the revised negative borders, (ii) linearize all the constraints of the corresponding CSP, and (iii) solve the resulting CSP by using *Binary Integer Programming* (BIP). Moreover, the hiding algorithm takes special care to ensure the validity of the transactions in the database extension, prior to releasing the whole (sanitized) database (i.e. the original database augmented with the computed extension) to untrusted third parties. When compared to [64], this approach is experimentally proven (see Chapter 8) to better preserve the quality of the border and to identify hiding solutions of superior quality. On the other hand, this approach is less scalable than [64] primarily due to the large number of variables and associated constraints that participate to the CSP. To improve the runtime cost of the proposed methodology, the authors introduce a partitioning approach that decomposes the CSP into smaller parts which can be solved separately, yielding the same hiding solution as the one of the original CSP. Moreover, in [72], Gkoulalas–Divanis and Verykios provide a complete framework that is suitable for the decomposition and the parallelization of the CSPs that are produced by the exact hiding methodologies of [64,68,70].

3.2 Other Hiding Approaches

Association rule hiding algorithms aim at protecting sensitive knowledge depicted in the form of frequent patterns and the related association rules. However, sensitive knowledge may appear in various forms directly related to the applied data mining algorithm that achieved to expose it. As a consequence, a set of hiding approaches have been devised recently to allow for the safeguarding of sensitive knowledge exposed by data mining tasks such as *clustering*, *classification* and *sequence* mining. In what follows, we briefly cover some state–of–the–art research work involving the hiding of sensitive knowledge depicted in the aforementioned formats.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

3.2.1 Classification Rule Hiding

Classification rule hiding algorithms consider a set of classification rules as sensitive and proceed to protect them by using either *suppression*–based or *reconstruction*–based techniques. The suppression–based techniques aim at reducing the *confidence* of a sensitive classification rule (measured in terms of the owner's belief regarding the rule's holding given the data), by distorting some attributes in the dataset belonging to transactions related to its existence. Chang and Moskowitz [34], were the first to address the inference problem caused by the downgrading of the data in the context of decision rules. Through a blocking technique, called *parsimonious downgrading*, the authors block the inference channels that lead to the identification of the sensitive rules by selectively sanitizing transactions so that missing values appear in the released dataset. This has as an immediate consequence the lowering of the confidence for the holding of the sensitive rules. Wang, et al. [175] propose a heuristic approach that achieves to fully eliminate all the sensitive inferences, while effectively handling overlapping rules. The algorithm identifies the set of attributes that influence the existence of each sensitive rule the most and removes them from those supporting transactions that affect the nonsensitive rules the least.

On the other hand, reconstruction–based approaches, inspired by the work in [36, 146] and introduced by Natwichai, et al. [129], target at reconstructing the dataset by using only supporting transactions of the nonsensitive rules. These approaches are advantageous over the heuristic data modification approaches, since they are shown to hardly introduce any side–effects [170]. First, they perform a rule–based classification of the original dataset to enable the owner of the data to identify the sensitive rules. Then, they construct a decision tree that is constituted only of nonsensitive rules approved by the data owner. The constructed dataset remains similar to the original one, except from the sensitive part, while the difference between the two datasets is proved to reduce as the number of rules increases. In [130] the authors achieve to further improve the quality of the reconstructed dataset. This is accomplished by extracting additional characteristic information from the original dataset with regard to the classification issue and by improving the decision tree building process.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Furthermore, with the aid of information gain, the usability of the released dataset is substantially ameliorated even in the case of hiding many sensitive rules with high discernibility in records classification. A similar approach can be found in [102]. The proposed methodology modifies transactions supporting both sensitive and nonsensitive classification rules in the original dataset and then uses the supporting transactions of the nonsensitive rules in order to produce its sanitized counterpart.

3.2.2 Privacy Preserving Clustering

The field of privacy preserving clustering collects techniques that aim at protecting the underlying attribute values and thus assure the privacy of individuals when sharing data for clustering. Achieving privacy preservation when sharing data for clustering is a challenging problem since the privacy requirements should be met, while the clustering results remain valid. The methodologies that have been proposed can be separated into two broad categories: the *transformation*–based approaches and the *protocol*–based approaches.

The transformation–based approaches are directly related to the distortion–based approaches of association rule hiding. They operate by performing a data transformation of the original dataset that maintains the similarity among the various pairs of attributes. In most of the cases, these methodologies are independent of the clustering algorithm that is used. In the transformed space, the similarity between the distorted attribute pairs can still provide accurate results that allow for the correct clustering of the various objects. Some interesting approaches in this category involve the work of Oliveira and Zaïane [135, 136].

On the other hand, protocol–based approaches assume a distributed scenario where a set of data owners want to share their data for clustering, without compromising the privacy of their data by revealing any secrets. The algorithms of this category make an assumption regarding the partitioning of the data among the interested, collaborating parties and are typically the privacy–aware versions of commonly used clustering algorithms, such as K–means [163]. The proposed protocols control the information that is communicated among the data owners and guarantee that no sensitive knowledge can be learned from the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

model. Approaches in this category include the work of Jha, et al. [99] and the work of Jagannathan, et al. [97]. A somewhat different kind of approach that targets on density–based clustering is presented in [44]. The authors propose a kernel–based distributed clustering algorithm that uses an approximation of density estimation in an attempt to harden the reconstruction process for the original database. Each site computes a local density estimate for the data it holds and transmits it to a trusted third party. In sequel, the trusted party builds a global density estimate and returns it to the collaborating peers. By making use of this estimate, the sites can locally execute density–based clustering.

3.2.3 Sequence Hiding

The hiding of sensitive sequences is one of the most recent research directions in privacy preserving data mining, particularly due to the tight relation that exists between sequences and trajectories. The given problem has the same underlying principles as rule hiding in the sense that a set of sensitive sequential patterns need to be hidden from a database, while causing the least side–effects on their nonsensitive counterparts. In Chapter 13 of the dissertation we survey some methodologies that belong to this category.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 4

Contributions

In Chapter 3 we surveyed the different research directions along with the state–of–the–art methodologies that have been proposed for association rule hiding in transactional databases. Prior to our first exact hiding algorithm, presented in the 15th International Conference on Information and Knowledge Management in 2006 [64], the approaches that had been proposed in this area of research were based on heuristics to perform the actual sanitization of the original database. The benefit of using heuristics to guide the knowledge hiding process has primarily to do with the efficiency and the scalability of these methodologies to very large problem sizes, rather than with the actual quality of the produced hiding solutions. This is due to the fact that heuristics aim at taking locally best decisions with respect to knowledge hiding which, however, are not necessarily globally best. As an effect, after conducting several experiments with state–of–the–art heuristic methodologies for association rule hiding, we found that in many cases these methodologies were unable to identify an optimal hiding solution, although such a solution existed.

In order to find an optimal hiding solution that introduces no side–effects to the hiding process, as well as it minimally distorts the original database to facilitate knowledge hiding, one needs to have a way to model the optimal solution and a methodology to enforce it. The introduction of the border revision process by Sun and Yu [160] in 2005 was a milestone with respect to the modeling part of an exact hiding solution (i.e. a solution

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

that introduces no side–effects in the hiding process). Border revision, given in detail in Chapter 5, provided us with the means to compute a small portion of itemsets from the original database, which are proven to play a key role in the hiding process. The control of the status of these itemsets (i.e. frequent vs infrequent) in the sanitized database leads to the elimination (when possible) of the side–effects in the computed hiding solution. To control the status of an itemset in the sanitized database, we introduced a set of unknowns that capture items' inclusion or exclusion in selected transactions. Then, we generated a set of inequalities involving these unknowns. As is proven in [64], the holding of all these inequalities in the sanitized database guarantees the exactness of the hiding solution. To attain an optimal hiding solution, one needs to solve the set of produced inequalities in a way that guarantees the minimum distortion of the original database. To do so, we proposed the transformation of the hiding process to an optimization problem where the goal is to identify a solution of the produced CSP that minimizes the database distortion. As a result, the identified solution is guaranteed to be optimal in the sense that it both eliminates all the side–effects of the hiding process, as well as it modifies the original database the least.

Our proposed exact hiding methodologies achieve to compute superior solutions when compared to state–of–the–art heuristics for association rule hiding, since in our case the sanitization process guarantees the simultaneous hiding of all the sensitive itemsets and thus allows for the identification of globally optimal solutions. In this part of the dissertation we present three exact hiding methodologies that we have proposed to facilitate optimal association rule hiding. The first methodology, called *inline* (see Chapter 6), is based on the modification of selected transactions from the original database in order to hide the sensitive knowledge. Certain limitations of this approach, which prevented the identification of optimal hiding solutions in some tested cases, led us to propose a new exact algorithm that augments the capabilities of the inline methodology by applying a two–phase iterative process. The details of this algorithm are presented in Chapter 7. Finally, in Chapter 8, we introduce an exact hiding methodology that instead of modifying existing transactions from the original database, it extends the database to facilitate knowledge hiding. The extended

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

part of the original database contains a set of carefully crafted transactions that aim at regulating the support of the itemsets in the sanitized database, such that an optimal hiding solution is attained.

Apart from providing a set of exact hiding methodologies, in this part of the dissertation we have also contributed along two other directions. First, in order to improve the scalability of the proposed algorithms, in Chapter 9, we propose a parallelization framework that breaks large optimization problems into smaller ones, which can be solved independently, without however compromising the quality of the computed hiding solution. Second, in Chapter 10, we propose a methodology for the quantification of the privacy that is offered by the exact hiding algorithms. By using this methodology, the database owner can regulate the privacy that is offered by the applied exact hiding algorithm versus the utility of the sanitized database. As is expected, the higher the privacy that is offered in the sanitized database by the hiding algorithm, the lower the utility (equiv. the higher the distortion) of the sanitized database, and vice versa. Thus, by using the methodology that is presented in Chapter 10, the owner of the data can decide the extend to which his/her sensitive knowledge will be protected in the sanitized database.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 5

Border Revision

In this chapter, we highlight the process of border revision which plays a key role on our proposed exact association rule hiding methodologies, as it captures the itemsets that need to remain frequent and those that need to become infrequent in the sanitized database to allow for an exact hiding solution. Moreover, we present a set of algorithms for the efficient computation of both the original and the revised borders of a transactional database.

Borders allow for a condense representation of the itemsets' lattice, identifying the key itemsets which separate all the frequent patterns from their infrequent counterparts. The process of border revision was originally proposed in [160] to facilitate minimum harm in the hiding of frequent itemsets. In what follows, we use an example to demonstrate how this process works. Consider database $\mathcal{D}_O$ of Table 8.1. Applying frequent itemset mining in $\mathcal{D}_O$ using $mfreq = 0.3$ leads to the set of frequent itemsets shown in the upper part of Table 8.2. Among these itemsets, assume that $\mathcal{S} = \{e, ae, bc\}$ are the sensitive ones that have to be protected. Figure 5.1 demonstrates the process of border revision for the problem at hand. In this figure, near each itemset we depict its support in the original (Figure 5.1(i)) and the revised (Figure 5.1(ii)) database. As one can observe, there are four possible scenarios involving the status of an itemset I prior and after the application of border revision:

C1 Itemset I was frequent in $\mathcal{D}_O$ and remains frequent in $\mathcal{D}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

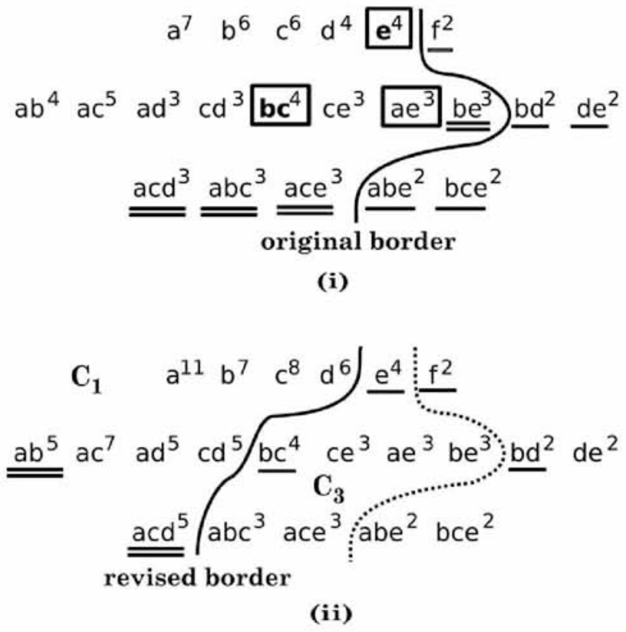


Figure 5.1: An itemsets' lattice demonstrating (i) the original border and the sensitive itemsets, and (ii) the revised border.

C2 Itemset I was infrequent in $\mathcal{D}_O$ and remains infrequent in $\mathcal{D}$.

C3 Itemset I was frequent in $\mathcal{D}_O$ and became infrequent in $\mathcal{D}$.

C4 Itemset I was infrequent in $\mathcal{D}_O$ and became frequent in $\mathcal{D}$.

The *original border* (Figure 5.1(i)) corresponds to the hyperplane that partitions the universe of itemsets into two groups: the frequent itemsets of $\mathcal{F}_{\mathcal{D}_O}$ (depicted on the left of the original borderline) and their infrequent counterparts in $\mathcal{P} - \mathcal{F}_{\mathcal{D}_O}$ (shown on the right of the borderline). The hiding process is then defined as the revision of the original border so that the revised border excludes from the frequent itemsets the sensitive ones and their supersets, i.e. all the itemsets in $\mathcal{S}_{max}$. Given the four possible scenarios for an itemset I prior and after the application of border revision, we deduce that in the ideal scenario where no nonsensitive itemsets are harmed, C2 should always hold, while C4 must never hold. On the contrary, C1 must hold for all the itemsets in $\mathcal{P} - \mathcal{S}_{max}$, while C3 must hold for all the itemsets in $\mathcal{S}_{max}$. Thus, the hiding of the sensitive itemsets corresponds to a movement of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the original borderline in the lattice to a new position that adheres to the ideal set $\mathcal{F}'_{\mathcal{D}}$.

Continuing our example, given that $\mathcal{S} = \{e, ae, bc\}$ (shown in squares in Figure 5.1(i)), we have that $\mathcal{S}_{min} = \{e, bc\}$ (shown in bold in Figure 5.1(i)), and $\mathcal{S}_{max} = \{e, ae, be, bc, ce, abc, ace\}$. Based on $\mathcal{D}_O$ in Figure 5.1(i) we present the itemsets of the original positive border (double underlined) and the original negative border (single underlined). Ideally, the frequent itemsets in the sanitized database $\mathcal{D}$ will be exactly those in $\mathcal{F}'_{\mathcal{D}} = \{a, b, c, d, ab, ac, ad, cd, acd\}$. The (ideally) revised borderline along with the corresponding borders for $\mathcal{D}$ and scenarios C_1 and C_3 that must hold for an exact solution, are depicted in Figure 5.1(ii). Specifically, in this figure, the itemsets of the revised positive border are double underlined while those of the revised negative border are single underlined. To enhance the clarity of the figure only a small portion of the itemsets involved in C_2 are shown. What is then needed, is a way to modify the transactions of the original database $\mathcal{D}_O$ in order to support the revised border in database $\mathcal{D}$.

To capture the ideal borderline, which is the revised border for the original database $\mathcal{D}_O$ based on the problem at hand, one can use the borders theory and identify the key itemsets which separate all the frequent patterns from their infrequent counterparts. As it is proved in [64, 65], these key itemsets correspond to the union of the revised positive border $\mathcal{B}\mathbf{d}^+(\mathcal{F}_{\mathcal{D}})$ and the revised negative border $\mathcal{B}\mathbf{d}^-(\mathcal{F}_{\mathcal{D}})$. Since in all the algorithms presented in the following chapters we treat these two sets differently, we provide separate algorithms for computing the positive and the negative borders, both for capturing the original and the revised borderline.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 5.1 Computation of the large itemsets and the original negative border.

```
1: procedure NB-APRIORI($\mathcal{D}_O$, $msup$)
2:     $F_1 \leftarrow$ GETLARGEITEMS($\mathcal{D}_O$, $msup$)
3:     for $k = 2; F_{k-1} \neq \varnothing; k$++ do
4:         $C_k \leftarrow$ APRIORIGEN($F_{k-1}, msup$)
5:         for each $t \in T_i$ do                              ▷ for all transactions in $\mathcal{D}_O$
6:             $C_t$ = subset($C_k$, $t$)                       ▷ get candidate subsets of $t$
7:             for each candidate $c \in C_t$ do
8:                 $c.count$++
9:         $F_k = \{c \in C_k | c.count \geq msup\}$
10:        $\mathcal{B}\mathbf{d}^-(\mathcal{F}_{\mathcal{D}_O}) = \{c \in C_k | c.count < msup\}$
11: procedure GETLARGEITEMS($\mathcal{D}_O$, $msup$)
12:     for $T_i \in \mathcal{D}_O$ do                          ▷ traverse all transactions
13:         for $x \in T_i$ do                                  ▷ traverse all items in the transaction
14:             $x.count$++
15:     for each item $x$ do
16:         if $x.count \geq msup$ then                         ▷ item $x$ is frequent
17:             $x \in F_k$
18:         else
19:             $x \in \mathcal{B}\mathbf{d}^-(\mathcal{F}_{\mathcal{D}_O})$      ▷ add this item to the negative border
```

Algorithm 5.1 provides a straight–forward way to compute the negative border. It achieves this, by incorporating the computation to the Apriori algorithm [14]. The new Apriori algorithm has extra code in the *NB–Apriori* and the *GetLargeItems* procedures to compute the border. On the other hand, the *Apriori–Gen* procedure is the same as in the original version of Apriori. The proposed method for the computation of the negative border is based on Apriori's candidate generation scheme, which uses $(k-1)$ frequent itemsets to produce candidate k large itemsets. Thus, it achieves to identify the first infrequent candidates in the lattice, whose all subsets are found to be frequent. In this algorithm, we use F_n to denote large n–itemsets that belong to $\mathcal{D}_O$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 5.2 Computation of the positive border (original and revised) $\mathcal{B}\mathbf{d}^{+}(\mathcal{F})$.

```
1: procedure PB-COMPUTATION(F)
2:     $count\{0...|F|\} \leftarrow 0$                          ▷ initialize counters
3:     $F_{sort}$ = reverse-sort($F$)
4:     for each $k$-itemset $f \in F_{sort}$ do
5:         for all $(k-1)$-itemsets $q \in F_{sort}$ do
6:             if $q \subset f$ then
7:                 $q.count$++
8:     for each $f \in F_{sort}$ do
9:         if $f.count = 0$ then
10:            $f \in \mathcal{B}\mathbf{d}^{+}(\mathcal{F})$      ▷ add itemset to $\mathcal{B}\mathbf{d}^{+}(\mathcal{F})$
```

Having identified the original negative border, the next step is to compute the original positive border $\mathcal{B}\mathbf{d}^{+}(\mathcal{F}_{\mathcal{D}_O})$. Algorithm 5.2, presents a level–wise (in the size of large itemsets) approach to achieve this computation. Assume $\mathcal{F}_{\mathcal{D}_O}$ is the set of frequent itemsets identified using Apriori. For each itemset in $\mathcal{F}_{\mathcal{D}_O}$ we associate a counter, initialized to zero. The algorithm first sorts these itemsets in decreasing length and then for all itemsets of the same length, say k, it identifies their (k–1) large subsets and increases their counters by one. The value of k iterates from the length of the largest identified frequent itemset down to 1. Finally, the algorithm performs a one–pass through all the counters of the itemsets and collects the large itemsets having a value of zero in the associated counter. These, constitute the positive border $\mathcal{B}\mathbf{d}^{+}(\mathcal{F}_{\mathcal{D}_O})$. Due to its very nature, this algorithm is suitable for computing both the original and the revised positive borders. For this reason, we use the notation $\mathcal{B}\mathbf{d}^{+}(\mathcal{F})$ to abstract the reference to the exact border that is computed (namely the original or the revised border).

A way of computing the negative border of the exact solution, in which $\mathcal{F}'_{\mathcal{D}} = \mathcal{F}_{\mathcal{D}_O} - \mathcal{S}_{max}$, is presented in Algorithm 5.3. In this algorithm we move top–down in the lattice to identify infrequent itemsets whose all proper subsets are frequent in $\mathcal{F}_{\mathcal{D}}$. First, we examine all 1–item itemsets. If any of these itemsets is infrequent, it should be included in the negative border of $\mathcal{F}_{\mathcal{D}}$. Then, we examine all 2–item itemsets by properly joining (symbol $\bowtie$ denotes a join) the frequent 1–item itemsets. Again, if the produced 2–item itemset does

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 5.3 Computation of the revised negative border $\mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$ of $\mathcal{D}$.

```
1: procedure INB-COMPUTATION(F)
2:     for $k = 1; F_k \neq \varnothing; k{+}{+}$ do
3:         if $k = 1$ then
4:             for each item $x \in F_1$ do
5:                 if $x \notin \mathcal{F}_\mathcal{D}$ then                    ▷ $x$ is infrequent
6:                     $x \in \mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$
7:         else if $k = 2$ then
8:             for $x \in F_1$ do
9:                 for $y \in F_1$ do
10:                    if $(x < y) \wedge (x \bowtie y \notin F_2)$ then
11:                        $(x \bowtie y) \in \mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$
12:        else                                                   ▷ for $k > 2$
13:            for $x \in F_{k-1}$ do
14:                for $y \in F_{k-1}$ do
15:                    if $(x_1 = y_1) \wedge \ldots \wedge (x_{k-1} < y_{k-1})$ then
16:                        $z = x \bowtie y$                       ▷ $z$ is the join of $x$ and $y$
17:                        if $z \notin \mathcal{F}_\mathcal{D}, \nexists r_{k-1} \subset z : r_{k-1} \notin F'$ then
18:                            $z \in \mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$
```

not exist in $\mathcal{F}_\mathcal{D}$ we include it in $\mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$. To examine k–itemsets (where $k > 3$), we first construct them by properly joining frequent (k–1)-itemsets (as in Apriori) and then check to see if the produced itemset is large in $\mathcal{F}_\mathcal{D}$. If it is reported as infrequent, then we examine all its (k–1) proper subsets. If none of these is infrequent, then the itemset belongs to the negative border, so we include it in $\mathcal{B}\mathbf{d}^-(\mathcal{F}_\mathcal{D})$.

Algorithm 5.4 Hiding of all the sensitive itemsets and their supersets.

```
1: procedure HIDESS($\mathcal{F}_{\mathcal{D}_O}$, $\mathcal{S}_{max}$)
2:     for each $s \in \mathcal{S}_{max}$ do                    ▷ for all sensitive itemsets
3:         for each $f \in \mathcal{F}_{\mathcal{D}_O}$ do                ▷ for all large itemsets
4:             if $s \subseteq f$ then                    ▷ the large itemset is sensitive
5:                 $\mathcal{F}_{\mathcal{D}_O} = \mathcal{F}_{\mathcal{D}_O} - f$                    ▷ remove itemset $f$
```

Finally, Algorithm 5.4 presents the hiding process in which we identify $\mathcal{F}'_\mathcal{D}$ by removing from $\mathcal{F}_{\mathcal{D}_O}$ all the sensitive itemsets and their supersets. To do so, we iterate over all the sensitive itemsets and their supersets (i.e. set $\mathcal{S}_{max}$), and the large itemsets in $\mathcal{F}_{\mathcal{D}_O}$, and we identify all those large itemsets that are supersets of the sensitive. Then, we remove these itemsets from the list of large itemsets, thus construct a new set $\mathcal{F}'_\mathcal{D}$ with the remaining large itemsets in $\mathcal{F}_{\mathcal{D}_O}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 6

An Integer Programming Approach for Itemset Hiding

In this chapter, we present the first non–heuristic algorithm that has been proposed for the securing of sensitive knowledge in patterns extracted during association rule mining. Instead of hiding the produced rules directly, we decide to hide the sensitive frequent itemsets that may lead to the production of these rules. As a first step, we introduce the notion of distance between two databases and a measure for quantifying it. The quantification of distance allows us to gain knowledge regarding the minimum modification that is necessary to be made to the original database in order to facilitate the hiding of the sensitive, while minimally affecting the nonsensitive itemsets. By trying to minimize the distance between the original database and its sanitized counterpart (that can safely be released), we propose a novel, exact algorithm for association rule hiding and evaluate it on real world datasets demonstrating its effectiveness towards solving the problem. The proposed algorithm (called *inline* [64]) formulates the hiding process as a CSP in which distance is the optimization criterion that has to be minimized, and subsequently solves the CSP by using BIP. It is important to mention that since our approach is non–heuristic, it does not suffer from local minima issues that would lead the hiding algorithm to suboptimal (i.e. locally — but not globally — best) solutions. Thus, this methodology is guaranteed to identify hiding

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

solutions of superior quality when compared to state–of–the–art heuristics.

The remainder of this chapter is organized as follows. Section 6.1 present the proposed privacy model for the identification of exact and optimal hiding solutions. In Section 6.2, we demonstrate the solution methodology that we followed. Finally, Section 6.3 presents the experiments that we conducted to prove the effectiveness of our approach. More experiments regarding the inline algorithm can be found in Chapters 7 and 8.

6.1 Privacy Model

In Chapter 5, we highlighted the process of border revision and we presented a set of algorithms which allow us to compute the revised borders that pertain to an exact hiding solution. The inline methodology that is presented in this chapter tries to enforce the revised borders when constructing the sanitized version $\mathcal{D}$ of the original database $\mathcal{D}_O$. The sanitization approach that we follow is based on the modification of selected T_{nm} values from one to zero ($1 \rightarrow 0$), thus excluding certain items from some transactions of $\mathcal{D}_O$. Since we do not add or remove any transactions, the sanitized version $\mathcal{D}$ of the database will have the same size (i.e. number of transactions) as the original one. Therefore, $|\mathcal{D}| = |\mathcal{D}_O|$. Using $\mathcal{F}_\mathcal{D}$ to denote the set of frequent itemsets in the sanitized database $\mathcal{D}$, our target for the hiding of sensitive itemsets can be expressed as follows: $\forall S \in \mathcal{S}_{min} : S \notin \mathcal{F}_\mathcal{D}$.

It is trivial to prove that the above property can be easily achieved for any database $\mathcal{D}_O$. The simplest solution will probably be to identify all items appearing in the frequent itemsets in S and remove them from all transactions in $\mathcal{D}_O$. However, this is certainly a naïve approach towards solving the problem. What one would like to do in reality is to hide all sensitive itemsets by minimally affecting the original database. As stated in Chapter 2, this minimum harm can be quantified as $\mathcal{F}'_\mathcal{D} = \mathcal{F}_{\mathcal{D}_O} - \mathcal{S}_{max}$ meaning that in the *ideal* case, we expect the sanitized database $\mathcal{D}$ to contain all the frequent itemsets of $\mathcal{D}_O$ except the sensitive ones[1]. By only removing items from transactions in $\mathcal{D}_O$, we ensure that $\mathcal{D}$ will

[1]We should point out that we use $\mathcal{F}'_\mathcal{D}$ to refer to the ideal set of frequent itemsets that we wish to hold in the sanitized database $\mathcal{D}$, while $\mathcal{F}_\mathcal{D}$ denotes the actual set of frequent itemsets in $\mathcal{D}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

contain no ghost itemsets. However, we still need to minimize the loss of nonsensitive frequent itemsets in $\mathcal{D}$ as a side–effect of the sanitization process.

Menon, et. al. in [122] use the notion of *accuracy* as the optimization criterion for deriving the hiding solutions. Maximizing accuracy is equivalent to minimizing the number of transactions from $\mathcal{D}_O$ that need to be modified. Although we consider this measure to lead to an interesting property, we argue that it is the number of actual *item modifications* rather than the number of sanitized transactions that needs to be minimized. Indeed, an algorithm may alter less transactions, but to a much higher degree than another one, that probably alters more transactions to a much lower degree. A basic drawback of the accuracy measure is that it penalizes all potential modifications of a frequent itemset, say for instance, both *'abc'* $\rightarrow \varnothing$ and *'abc'* $\rightarrow$ *'ab'*, to the same extent. Based on this mishap, we propose a novel global measure of quality, called *distance*, that aims at capturing the proximity between the original and the sanitized database. Formally, we define distance as follows:

$$dist(\mathcal{D}_O, \mathcal{D}) = \left| \sum_{n\in[1,N],m\in[1,M]} T_{nm} - \sum_{n\in[1,N],m\in[1,M]} T'_{nm} \right| = |\{T_{nm} \neq T'_{nm}\}| \qquad (6.1)$$

where T_{nm} refers to the original database $\mathcal{D}_O$ and T'_{nm} to its sanitized version $\mathcal{D}$, N corresponds to the number of transactions in $\mathcal{D}_O, \mathcal{D}$, and M to the number of items. Due to the fact that in our proposed methodology we allow *only* exclusion of items from transactions, minimizing the outcome of the above formula is exactly the same as maximizing the number of ones left in $\mathcal{D}$. Thus, we can write

$$min\{dist(\mathcal{D}_O, \mathcal{D})\} = \max \left(\sum_{n\in[1,N],m\in[1,M]} T'_{nm} \right) \qquad (6.2)$$

Formula (6.2) captures the optimization criterion that is used in our integer program formulation, presented in detail in Section 6.2. The distance metric allows us to select the optimal solution among the exact ones (in the case that many exact hiding solutions exist for the problem at hand) or to approximate the optimal solution as much as possible (in the case that an exact hiding solution cannot be found). Since we only want to reduce the support of the minimum sensitive itemsets, i.e. the itemsets in $\mathcal{S}_{min}$, the T_{nm} values in $\mathcal{D}_O$

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

that will be zeroed out in $\mathcal{D}$ will have to correspond to items within transactions supporting itemsets in $\mathcal{S}_{min}$.

Having defined the optimization criterion that we need to minimize, our next target is to identify the set of transactions in which we will apply it. Solving the entire problem is well known to be NP–hard (c.f. [20, 122]). Therefore, we will have to use only a subset of the transactions in our formulation. We proceed by demonstrating the two possible constraints for a frequent itemset appearing in $\mathcal{D}_O$.

An itemset I will continue to be frequent in $\mathcal{D}$ if and only if

$$\sup(I,\mathcal{D}) \geq msup \Rightarrow \sum_{n=1}^{N} \prod_{i_m \in I} T'_{nm} \geq msup \tag{6.3}$$

and will be infrequent otherwise, i.e. when

$$\sup(I,\mathcal{D}) < msup \Rightarrow \sum_{n=1}^{N} \prod_{i_m \in I} T'_{nm} < msup \tag{6.4}$$

Based on these inequalities we argue that any exact solution will necessarily satisfy them for the entire problem (i.e. for all the itemsets in the lattice of $\mathcal{D}$). Specifically, for all frequent itemsets appearing in $\mathcal{F}'_{\mathcal{D}}$, we want them to remain frequent in $\mathcal{D}$ and therefore they should satisfy inequality (6.3). On the contrary, all frequent itemsets in $\mathcal{S}_{max}$ should be hidden in $\mathcal{D}$ and therefore they must satisfy inequality (6.4). If at least one inequality does not hold for an itemset in $\mathcal{F}'_{\mathcal{D}}$, then the identified solution is non–exact but approximate. Solving this system of inequalities for all possible itemsets is NP–hard since in the general case there are $|\wp(I)| - 1 = 2^M - 1$ such inequalities.

We should mention at this point that the two inequalities, namely (6.3) and (6.4), are not of the same importance. While it is crucial to ensure that inequality (6.4) holds for all the sensitive itemsets in $\mathcal{D}$ such that they are properly hidden, inequality (6.3) just allows us to revise the border of the frequent vs infrequent patterns in the itemsets' lattice, thus ensuring minimization of the side–effects in the hiding process. As we present in Section 6.2, we exploit the difference in the significance of these two inequalities in order to obtain the best possible suboptimal (approximate) solutions, when optimal ones do not exist.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

6.2 Solution Methodology

Hopefully, it turns out that the NP–hard problem of the $2^M - 1$ inequalities can be reduced to an extent that is solvable while yielding the exact same set of solutions. In what follows, we discuss how this is possible. Ref. [64] provides the formal proof of how this reduction is achieved. Based on the reduced set of produced inequalities, we formulate the hiding process as a CSP and solve it by using BIP.

6.2.1 Problem Size Minimization

Although we are unable to solve the entire problem, due to its exponential growth, we argue that we can minimize its size up to a point, without any loss on the attained solution. Specifically, we argue that we can remove certain inequalities of the system (referring to the status of itemsets in $\mathcal{D}$) that do not affect its set of solutions. The rational behind this claim is that in a typical case there exist many overlapping itemsets in $\mathcal{D}_O$ and therefore a large number of overlapping inequalities are produced. Identifying such inequalities helps us to drastically reduce the size of the problem up to a point that it becomes solvable.

As we presented earlier, an one–to–one correspondence exists between itemsets and produced inequalities in our system. Given that $\mathcal{C}$ is the total set of affected itemsets, such that $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{2^M-1}\}$, we denote by $\mathcal{L}_\mathcal{C}$ the set of solutions of the corresponding inequalities from $\mathcal{C}$. The set of solutions for the system of inequalities will be the intersection among the solution produced for each individual inequality. Thus, we can write $\mathcal{L}_\mathcal{C} = \bigcap_{i=1}^{|\mathcal{C}|} \mathcal{L}_{\mathcal{C}_i}$.

Based on the above notation, the set of solutions that corresponds to all itemsets in $\mathcal{D}_O$ is denoted as $\mathcal{L}_{\wp(I)\setminus\{\varnothing\}}$. Given an inequality (produced, for example, by an itemset $\mathcal{C}_2$) with a solution set being a proper subset of the solution set of another inequality, say $\mathcal{C}_1$, we can deduce that the latter inequality can be removed from any system containing the first inequality, without affecting the global solution of this system. In this case we have that $\mathcal{L}_{\mathcal{C}_2} \subset \mathcal{L}_{\mathcal{C}_1}$ and we state that $\mathcal{C}_1$ is a (generalized) *cover* of $\mathcal{C}_2$. By exploiting cover relations, we achieve to reduce the problem of satisfying all the inequalities produced by all

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the itemsets in $\mathcal{D}_O$, to the examination of a much smaller set

$$\mathcal{C} = \{I \in \mathcal{B}\mathbf{d}^{+}(\mathcal{F}'_{\mathcal{D}}) : I \cap I^{\mathcal{S}_{min}} \neq \varnothing\} \cup \mathcal{S}_{min} \tag{6.5}$$

where notation $I^{\mathcal{S}_{min}}$ depicts the items of the itemsets in $\mathcal{S}_{min}$. Thus, $I \cap I^{\mathcal{S}_{min}}$ refers to those itemsets of the revised positive border that do not contain any items from those appearing in the minimum sensitive itemsets (i.e. in $\mathcal{S}_{min}$). Set $\mathcal{C}$ provides us the optimal hiding solution $\mathcal{L}_{\mathcal{C}}$, if one exists. The reduction of the set of all inequalities to those involving the itemsets in $\mathcal{C}$, is proven as part of [64].

6.2.2 Reduction to CSP and the BIP solution

After our formalization of the sensitive itemsets' hiding procedure, the whole problem can been regarded as a CSP. A CSP [147] is defined by a set of variables and a set of constraints, where each variable has a non–empty domain of potential values. The constraints, on the other hand, involve a subset of the variables and specify the allowable combinations of valid values that these variables can attain. An assignment that does not violate the set of constraints is called *consistent*. A solution of a CSP is a complete assignment of values to the variables that satisfies all the constraints. As a final remark, in CSPs we usually wish to maximize or minimize an objective function subject to a number of constraints.

CSPs can be solved by using various techniques such as linear and non–linear programming [114]. In our context all variables are binary as they refer to items participating to specific transactions; this fact provides us with an important advantage as we will see later on. To solve our CSP, we first transform it to an optimization problem and then apply a technique called BIP [83]. Our formulation enables us to solve the sanitization problem in $\mathcal{D}_O$ and is capable of identifying the optimal solution (if one exists). In the case of problems where exact solutions are infeasible, we provide a relaxation of this algorithm (by using a heuristic targeted for inequalities selection and removal) that allows the identification of a good approximate solution.

Figure 6.1 presents the CSP formulation as an optimization problem. As we can notice, the degree of the constraints participating in the problem formulation is a–priori

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{maximize}\left(\textstyle\sum_{u_{nm}\in U} u_{nm}\right)$$

$$\textit{subject to}\begin{cases}\sum_{T_n\in\mathcal{D}_I}\prod_{i_m\in I} u_{nm} < msup, \forall I \in \mathcal{S}_{min}\\ \sum_{T_n\in\mathcal{D}_I}\prod_{i_m\in I} u_{nm} \geq msup, \forall I \in V\end{cases}$$

$$\text{where } V = \{I \in \mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) : I \cap I^{\mathcal{S}_{min}} \neq \varnothing\}$$

Figure 6.1: The CSP formulation as an optimization process.

$$\textbf{Replace}$$

$$\left(\textstyle\sum_{T_n\in D_F}\Psi_i \lesseqqgtr msup, \Psi_i = \prod_{i_m\in F} u_{nm} = u_{nF_1}\times\ldots\times u_{nF_{|F|}}\right)$$

$$\textbf{with}$$

$$\forall i\begin{cases}\Psi_i \leq u_{nF_1}\\ \Psi_i \leq u_{nF_2}\\ \vdots\\ \Psi_i \leq u_{nF_{|F|}}\\ \Psi_i \geq u_{nF_1}+u_{nF_2}+\ldots+u_{nF_{|F|}}-|F|+1\end{cases}$$

$$\textbf{and}$$

$$\textstyle\sum_i \Psi_i \lesseqqgtr msup$$

$$\text{where } \Psi_i \in \{0,1\}$$

Figure 6.2: The Constraints Degree Reduction approach.

unknown, and can be as large as the number of items in the given dataset $\mathcal{D}_O$. This fact prohibits us from solving the CSP in the format that is presented in Figure 6.1. Fortunately though, due to the binary nature of the variables, we are capable of handling this issue. To do so, we replace any inequality that contains at least one product of two or more u_{nm} variables, with a number of inequalities that each contains no product of variables and, when all solved, we attain exactly the same solution as that of the initial inequality. The side–effect of this methodology is that it increases the number of constraints in the system. On the other hand, the resulting inequalities are very simple and allow for fast solutions, thus adhere for an efficient solution of the CSP. A number of temporary binary values Ψ_i need to be introduced, as shown in Figure 6.2. After applying the *Constraints Degree Reduction* (CDR) approach all constraints are linear with no coefficients. A linear optimization solver

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 6.1 Relaxation Procedure in **V**.

```
1: procedure SELECTREMOVE(Constraints $\mathbf{C_R}$, $\mathbf{V}$, $\mathbf{D}$)
2:     $C_{R_{maxlen}} \leftarrow \bigcup \mathbf{argmax_i}\{|R_i|\}$                ▷ $C_{R_i} \leftrightarrow V_i$
3:     $cr_{msup} \leftarrow \mathbf{min}_{C_{R_{maxlen}},i}(\sup(R_i \in V, D))$                ▷ $R_i \in \mathbf{V}$
4:     for each $c \in C_{R_{maxlen}}$ do
5:         if $\sup(R_i, D) = cr_{msup}$ then
6:             Remove (c)                ▷ remove constraint from the CSP.
```

can thus be applied to provide the optimal solution $\mathcal{L}_C$.

The final case that needs to be examined is what happens if the optimization problem yields no solution, i.e. the CSP is unsolvable. This means, that an exact hiding solution cannot be found based on our distance criterion. What allows us to proceed in such a case and identify a good approximate solution is the fact that the integer programming problem that contains only the constraints imposed by the sensitive itemsets in $\mathcal{S}_{min}$, will always have a solution. Thus, we can always identify a good hiding solution if we remove a portion of the constraints imposed by the itemsets in V (see Figure 6.1). In a realistic situation, only a small fraction of constraints will have to be discarded. The question that is now raised is how to select which inequalities to remove. Algorithm 6.1 provides a simple heuristic for selection and removal of inequalities from the CSP of Figure 6.1. It applies a relaxation process by removing all constraints that correspond to maximal size and minimum support itemsets in V. The rationale behind this heuristic is that the hidden itemsets in $\mathcal{D}$ (due to the side–effects of hiding itemsets in $\mathcal{S}_{max}$) will be the first that would be hidden in $\mathcal{D}_O$ if the support threshold was increased, since their current support is also low. After removing a small subset of the constraints, we expect the remaining problem to become solvable. As a final point, we need to mention that we iteratively apply this relaxation process until the CSP has a solution, each time potentially removing more than one constraints from the CSP.

6.2.3 An Example

To demonstrate our core approach we will use an example. Consider the transactional database shown in Table 6.1. Using a minimum frequency threshold *mfreq*$= 0.2$, the set of frequent itemsets identified by Apriori is: $\mathcal{F}_{\mathcal{D}_O} = \{A, B, C, D, AB, AC, AD, CD, ACD\}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 6.1: The original database $\mathcal{D}_O$ used in the example.

A	B	C	D
1	0	1	0
1	0	1	1
0	0	1	1
0	1	0	0
1	1	1	1
0	0	0	1
0	0	1	0
1	1	0	0

Suppose we want to hide sensitive itemset $\{AB\}$, therefore $\mathcal{S} = \{AB\}$. As a first step we compute the ideal set of frequent itemsets for $\mathcal{D}$ by using Algorithm 5.4. That is: $\mathcal{F}'_{\mathcal{D}} = \{A,B,C,D,AC,AD,CD,ACD\}$. The revised positive border for $\mathcal{D}$ will then be: $\mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) = \{B,ACD\}$. To produce the needed constraints for the CSP, we consider all itemsets appearing in $\mathcal{C}$ (see (6.5)). In this example we have that: $V = \mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) = \{B,ACD\}$. First, we substitute in all transactions supporting the sensitive itemsets, their current T_{nm} values of the sensitive items with u_{nm} binary variables. This constitutes the intermediate form of database $\mathcal{D}_O$, shown in Table 6.2. Then, for each itemset in $(V \cup \mathcal{S}_{min})$, we create a constraint:

$$1 + u_{52} + u_{82} \geq msup \Rightarrow u_{52} + u_{82} \geq 0.6 \tag{6.6}$$

$$1 + u_{51} \geq msup \Rightarrow u_{51} \geq 0.6 \tag{6.7}$$

$$u_{51}u_{52} + u_{81}u_{82} < msup \Rightarrow u_{51}u_{52} + u_{81}u_{82} < 1.6 \tag{6.8}$$

The first two inequalities correspond to the itemsets in V, whereas the third one is for the sensitive itemset in $\mathcal{S}$ (here $\mathcal{S}_{min} = \mathcal{S}$). From these constraints, only the third one contains products of u_{nm} variables and therefore needs to be replaced. Based on Figure 6.2 we replace it, while introducing two new temporary binary variables, Ψ_1 and Ψ_2, as follows:

$$\overbrace{u_{51}u_{52}}^{\Psi_1} + \overbrace{u_{81}u_{82}}^{\Psi_2} < 1.6 \Rightarrow \begin{cases} \Psi_1 \leq u_{51}, & \Psi_2 \leq u_{81} \\ \Psi_1 \leq u_{52}, & \Psi_2 \leq u_{82} \\ \Psi_1 \geq u_{51} + u_{52} - 1, & \Psi_2 \geq u_{81} + u_{82} - 1 \end{cases}$$

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 6.2: The intermediate form of database $\mathcal{D}_O$ used in the example.

A	**B**	**C**	**D**
1	0	1	0
1	0	1	1
0	0	1	1
0	1	0	0
u_{51}	u_{52}	1	1
0	0	0	1
0	0	1	0
u_{81}	u_{82}	0	0

$$\textbf{maximize}\,(u_{51}+u_{52}+u_{81}+u_{82})$$

$$\textit{subject to}\begin{cases} u_{52}+u_{82}\geq 0.6\\ u_{51}\geq 0.6\\ \Psi_1\leq u_{51}\\ \Psi_1\leq u_{52}\\ \Psi_2\leq u_{81}\\ \Psi_2\leq u_{82}\\ \Psi_1\geq u_{51}+u_{52}-1\\ \Psi_2\geq u_{81}+u_{82}-1\\ \Psi_1+\Psi_2<1.6 \end{cases}$$

$$\text{where } \{u_{51},u_{52},u_{81},u_{82},\Psi_1,\Psi_2\}\in\{0,1\}$$

Figure 6.3: The CSP formulation for dataset $\mathcal{D}_O$.

By using this process we achieve to linearize all the constraints of the CSP that involve products of binary variables. The resulting CSP is presented in Figure 6.3. We solve it by using BIP and we get one among the three optimal solutions presented in Table 6.3.

6.3 Experiments and Results

We tested the inline algorithm on real world datasets using different parameters such as minimum support threshold and number/size of sensitive itemsets to hide. In this section, we provide the list of datasets we used, their special characteristics, the selected parameters

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 6.3: The three exact (and optimal) solutions for the CSP of the example.

Solution	u_{51}	u_{52}	u_{81}	u_{82}
l_1	1	0	1	1
l_2	1	1	0	1
l_3	1	1	1	0

Table 6.4: The characteristics of the three datasets.

Dataset	***n***	***m***	**Avg tlen**
BMS–WebView–1	59,602	497	2.50
BMS–WebView–2	77,512	3,340	5.00
Mushroom	8,124	119	23.00

and the attained experimental results.

6.3.1 The Datasets

All the real datasets that we used to evaluate our methodology are publicly available through the FIMI repository (http://fimi.cs.helsinki.fi/). Datasets BMS–WebView–1 and BMS–WebView–2 both contain click stream data from the Blue Martini Software, Inc. and were used for the KDD Cup 2000 [106]. The mushroom dataset was prepared by Roberto Bayardo (University of California, Irvine) [22]. These datasets demonstrate varying characteristics in terms of the number of transactions and items and the average transaction lengths. Table 6.4 summarizes them.

The primary bottleneck of our approach was the time taken to run the frequent itemset mining algorithm. The thresholds of minimum support were properly selected to ensure a large amount of frequent itemsets. Among them, we randomly selected a portion and characterized them as sensitive. We conducted several experiments trying to hide 1, 2, 4 and 5–sensitive itemsets. Since our sanitization methodology is item–based, the higher the number of items in a sensitive itemset, the more the u_{nm} variables involved and therefore the more constraints need to be solved. To avoid the explosion in the number of constraints we enforced two pruning techniques. Firstly, we removed tautologies, i.e. inequalities that always hold. Secondly, we partitioned inequalities, identified overlapping sets and kept only

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the most specific inequality from each set. Both pruning techniques can be easily applied and relieve the solver from a good portion of unnecessary work. Our code was implemented in Perl and C and we conducted our experiments on a PC running Linux on an Intel Pentium D, 3.2 Ghz processor. The integer programs were solved using ILOG CPLEX 9.0 [95].

6.3.2 Evaluation Methodology

We evaluated our algorithm by the metric of distance between databases $\mathcal{D}_O$ and $\mathcal{D}$. The lower the distance, the better the quality of the hiding solution. Since even in non–exact solutions we ensure that the sensitive itemsets have been successfully hidden, we argue that in any case the sanitized database $\mathcal{D}$ can be safely released to untrusted third parties.

As a final comment on the evaluation procedure we should mention that it is not sensible to compare the attained experimental results against the widely known metric of accuracy (as used in other state–of–the–art algorithms). The reason is that in our case what we try to minimize is not the number of transactions from $\mathcal{D}_O$ that will be sanitized (which is what the metric of accuracy captures), but the total number of individual items that will be removed from transactions in $\mathcal{D}_O$. These two goals are highly distinct. According to our perception a good sanitization procedure should try to minimize not the number of affected transactions but the total number of affected items. Thus, the metric of accuracy would be inappropriate to evaluate the outcome of the presented methodology.

6.3.3 Experimental Results

In order to test our approach we considered the following hiding scenarios: hiding 1 1–itemset (*HS1*), hiding 2 1–itemsets (*HS2*), hiding 1 2–itemset (*HS3*), hiding 2 2–itemsets (*HS4*), hiding 1 4–itemset (*HS5*), hiding 2 4-itemsets (*HS6*), and hiding 1 5–itemset (*HS7*). Table 6.5 summarizes the attained experimental results. The number of u_{nm} variables participating in the CSP provides an estimate of the worst–case scenario in the context of our algorithm; it is equivalent to the maximum distance between $\mathcal{D}_O$ and $\mathcal{D}$. The fourth column shows the actual distance of the two databases as reported by the solver. It corresponds to

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 6.5: The experimental results for the three datasets.

Dataset	***Hiding Scenario***	u_{nm} ***vars***	**Distance**
BMS–1	HS1	157	128
BMS–1	HS2	359	301
BMS–1	HS3	60	19
BMS–1	HS4	120	12
BMS–1	HS5	120	8
BMS–1	HS6	228	21
BMS–1	HS7	150	24
BMS–2	HS1	12	3
BMS–2	HS2	36	18
BMS–2	HS3	24	3
BMS–2	HS4	40	8
BMS–2	HS5	40	4
BMS–2	HS6	80	12
Mushroom	HS1	32	8
Mushroom	HS2	68	20
Mushroom	HS3	64	12
Mushroom	HS4	128	16
Mushroom	HS5	128	8
Mushroom	HS6	256	24
Mushroom	HS7	160	16

the actual number of items that were hidden from individual transactions of $\mathcal{D}_O$.

6.3.4 Some Remarks on the Efficiency of the Inline Algorithm

At a first glance, a shortcoming of our proposed methodology seems to be the time required for solving the produced CSPs, especially due to the number of constraints introduced by the CDR approach. However, due to the fact that the produced constraints involve binary variables and no products among these variables exist, it turns out that the solution time is acceptable. To support our claim, we created a dataset of 10,000 transactions with 10 items each, where all items participate in all transactions. We set *msup* to 1.0 and extracted all itemsets from the lattice. Then, we hide 2 1–itemsets and 2 4–itemsets, all consisting of disjoint items. To ensure our problem has a feasible solution, we relaxed it by using only the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

constraints of set $\mathcal{S}_{min}$ that always lead to a satisfiable CSP. Based on our formulation, the total number of u_{nm} variables participating in the problem are 100,000. To split their products for the 2 4–itemsets, another 20,000 Ψ_i variables were introduced. The total number of constraints were 100,004 and the optimal solution is 99,996 (since 4 items, one belonging in each sensitive itemset, need to be zeroed in a transaction). CPLEX was capable of identifying this solution within 889 seconds (approx. 15 min), which we consider low for such a demanding problem. Of course, adding the constraints from set V increases time complexity. However, in all our experiments so far the solver was able to rapidly decide if the optimization problem was infeasible. Thus, in such cases, the relaxation procedure (presented in Algorithm 6.1) was soon applied to allow for the production of a solvable CSP and the identification of a good approximate hiding solution.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

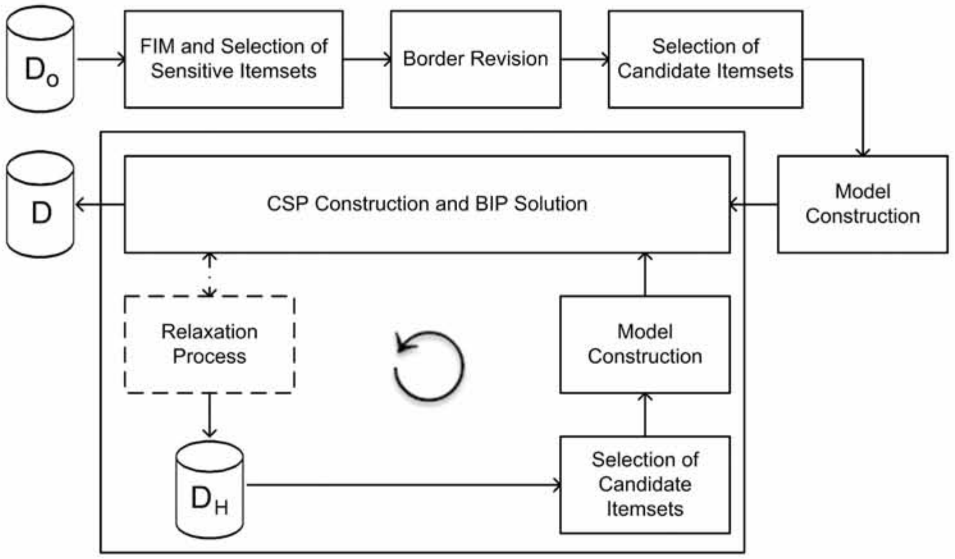
Chapter 7

A Two–Phase Iterative Algorithm for Itemset Hiding

In this chapter, we present a two–phase iterative methodology (originally proposed in [70]) that extends the functionality of the inline algorithm of [64] (Chapter 6) to allow for the identification of exact hiding solutions for a wider spectrum of problem instances. We define a *problem instance* as the set of (i) the original dataset $\mathcal{D}_O$, (ii) the minimum frequency threshold *mfreq* that is considered for its mining, and (iii) the set of sensitive itemsets that have to be protected. Since the inline algorithm allows only supported items in $\mathcal{D}_O$ to become unsupported in $\mathcal{D}$, we argue that there exist problem instances that although they allow for an exact solution, the inline approach is incapable of finding it. A proof of this statement is provided in Section 7.4, as well as in the experimental evaluation of [68].

The remainder of this chapter is organized as follows. In Section 7.1 we present the theoretical background behind the proposed two–phase iterative process (see Figure 7.1; large box shows the iteration). Section 7.2 highlights some important aspects of the proposed methodology that influence the quality of the hiding solution that is found. In Section 7.3, we present a specific problem instance for which the inline algorithm fails to provide an exact hiding solution and demonstrate how the new approach achieves to identify it. Finally, Section 7.4 presents the experimental evaluation of the proposed methodology.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



D_o
FIM and Selection of Sensitive Itemsets
Border Revision
Selection of Candidate Itemsets
Model Construction
CSP Construction and BIP Solution
D
Relaxation Process
Model Construction
D_H
Selection of Candidate Itemsets

Figure 7.1: The architectural layout for the two–phase iterative approach.

7.1 Theory Behind the Two–Phase Iterative Approach

The proposed methodology for the identification of exact hiding solutions consists of two phases that iterate until either (i) an exact solution of the given problem instance is identified, or (ii) a pre–specified number of subsequent iterations ℓ have taken place. Threshold ℓ is called the *limiting factor* and must remain low enough to allow for a computationally efficient solution. The first phase of the algorithm utilizes the inline approach to hide the sensitive knowledge. If it succeeds, then the process is terminated and database $\mathcal{D}$ is returned. This phase causes the retreat of the positive border of $\mathcal{D}_O$ in the lattice, thus excluding from $\mathcal{F}_{\mathcal{D}}$ the sensitive itemsets and their supersets. If the first phase is unable to identify an exact solution, the algorithm proceeds to the second phase, which implements the dual counterpart of the inline algorithm. Specifically, from the produced (infeasible) CSP of the first phase, the proposed algorithm proceeds to remove selected inequalities as in Algorithm 6.1 but in a "one–to–one" rather than in a "batch" fashion[1], until the underlying CSP becomes feasible. The second phase results to the expansion of the positive border,

[1]The selection among inequalities corresponding to the same "batch" is performed arbitrarily.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

and the iteration of the two phases aims at the approximation (to the highest possible extent) of the ideal borderline in the sanitized database.

In what follows, let $\mathcal{H}$ denote the set of itemsets for which the corresponding inequalities were removed from the CSP to allow for a feasible solution. Obviously, $\mathcal{H} \subseteq \mathcal{B}\mathbf{d}^{+}(\mathcal{F}'_{\mathcal{D}})$. Since the produced CSP is now feasible, the algorithm proceeds to identify the hiding solution and to compose the corresponding sanitized database $\mathcal{D}$. This database is bound to suffer from the side–effect of hidden (nonsensitive) frequent patterns. The purpose of the second phase is to try to constitute these lost (due to the side–effects) itemsets of set $\mathcal{H}$ frequent again by increasing their support in $\mathcal{D}$. However, the support increment should be accomplished in a careful manner to ensure that both the sensitive itemsets as well as all the itemsets that do not participate in the ideal set $\mathcal{F}'_{\mathcal{D}}$, remain infrequent.

Let $\mathcal{D}_{\mathcal{H}}$ denote this intermediate (for the purposes of the second phase) view of the sanitized database $\mathcal{D}$. In this database we want the itemsets of $\mathcal{H}$ to become frequent and the outcome database to adhere as much as possible to the properties of the ideal set $\mathcal{F}'_{\mathcal{D}}$. As one can observe, the second phase of the approach has as a consequence the movement of the revised positive border downwards in the lattice to include an extended set of itemsets, constituting them frequent. As a result, this two–phase iterative process, when viewed vertically in the itemset lattice, resembles the oscillation of a pendulum, where one stage follows the other with the hope that the computed revised positive border will converge at some point to the ideal one.

We now proceed to analyze the mechanism that takes place as part of the second phase of this "oscillation". Since the goal of this phase is to constitute the itemsets of $\mathcal{H}$ frequent in $\mathcal{D}_{\mathcal{H}}$, the modification of this database will be based only on item inclusions on a selected subset of transactions. Following the inline approach, the candidate items for inclusion are only those that appear among the itemsets in $\mathcal{H}$, i.e. those in the universe of $I^{\mathcal{H}}$. All these items will be substituted in the transactions of $\mathcal{D}_{\mathcal{H}}$ where they are unsupported, by the corresponding u_{nm} variables. This will produce an intermediate form of database $\mathcal{D}_{\mathcal{H}}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{minimize}\left(\sum_{u_{nm}\in U} u_{nm}\right)$$

$$\textit{subject to}\begin{cases} \sum_{T_n\in D_{H,\{I\}}} \prod_{i_m\in X} u_{nm} < \text{msup}, \forall I \in \mathcal{B}\mathbf{d}^-(\mathcal{F}'_{\mathcal{D}}) \\ \sum_{T_n\in D_{H,\{I\}}} \prod_{i_m\in R} u_{nm} \geq \text{msup}, \forall I \in \mathcal{H} \end{cases}$$

Figure 7.2: The CSP for the second stage of the two–phase iterative approach.

Following that, a CSP is constructed in which the itemsets that are controlled belong in set

$$\mathcal{C} = \mathcal{H} \cup \mathcal{B}\mathbf{d}^-(\mathcal{F}'_{\mathcal{D}}) \tag{7.1}$$

Finally, the optimization criterion is altered to denote that a minimization (i.e. least amount of 1s), rather than a maximization of the binary variables u_{nm} will lead to the best possible solution. Figure 7.2 presents the form of the CSP created in the second phase of this two–phase iterative hiding process. As mentioned earlier, these two phases are executed in an iterative fashion until convergence to the exact hiding solution is achieved or a pre–specified number of oscillations ℓ take place. In this figure, $D_{H,\{I\}}$ refers to the portion of D_H (i.e. the transactions) that supports itemset I.

7.2 Removal of Constraints from the Infeasible CSP

An aspect of the two–phase iterative approach that requires further investigation regards the constraints selection and removal process that turns an infeasible CSP into a feasible one. In the workings of the proposed approach we consider the eviction process of Algorithm 6.1 in an attempt to maximize the probability of yielding a feasible CSP after the removal of only a few number of constraints (inequalities). This is achieved by removing the most strict constraints, i.e. those that involve the maximum number of binary variables (equivalently, the maximum length itemsets from $\mathcal{D}_O$). To ensure that the removal of these constraints will cause minimal loss of nonsensitive knowledge to the database, we select among them the ones that involve itemsets of low support in $\mathcal{D}_O$, as those itemsets would be the first to be hidden if the support threshold was increased. Although we consider this to be a

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

reasonable heuristic, it may not always lead to the optimal selection, i.e. the one where the minimum number of constraints are selected for eviction and their removal from the CSP causes the least distortion to the database. Thus, in what follows, we discuss the properties of a mechanism that can be used for the identification of the best possible set of inequalities to formulate the constraints of the feasible CSP.

Let a *constraint set* be a set of inequalities corresponding to the itemsets of the positive border, as taken from the original (infeasible) CSP. We argue that there exists a “1–1” correspondence between constraint sets and itemsets, where each item is mapped to a constraint (and vice versa). A frequent itemset is equivalent to a *feasible constraint set*, i.e. a set of constraints that has a solution satisfying all the constraints in the set. The downwards closure property of the frequent itemsets also holds in the case of constraint sets, since all the subsets (taken by removing one or more inequalities) of a feasible constraint set are also feasible constraint sets and all the supersets of an infeasible constraint set are also infeasible constraint sets. Furthermore, a *maximal feasible constraint set* (in relation to a maximal frequent itemset) is a feasible constraint set where all its supersets are infeasible constraint sets. The constraints removal process, explained earlier, can be considered as the identification of a maximal feasible constraint set among the inequalities of the revised positive border. These constraints will be the ones that will participate to the (feasible) CSP. Due to the correspondence that exists between itemsets and constraint sets, we argue that one could use techniques that are currently applied on frequent itemset mining algorithms (such as pruning approaches) to efficiently identify the maximal constraint sets and possibly further select one among them (if we consider the existence of some metric of quality among the different constraint sets). However, the decision of whether a constraint set is feasible or not is a computationally demanding process and for this reason we consider the proposed heuristic to provide a more efficient (although in some cases suboptimal) alternative.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 7.1: The original database $\mathcal{D}_O$.

A	B	C	D	E
1	0	1	0	0
1	0	1	1	1
0	0	1	1	0
0	1	0	0	1
1	0	1	1	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	1	1	0

Table 7.2: The intermediate database of $\mathcal{D}_O$.

A	B	C	D	E
1	0	1	0	0
1	0	u_{23}	u_{24}	1
0	0	u_{33}	u_{34}	0
0	1	0	0	1
1	0	u_{53}	u_{54}	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	u_{103}	u_{104}	0

7.3 An Example of the Two–Phase Iterative Process

Consider database $\mathcal{D}_O$ depicted in Table 7.1. When mining this database for frequent itemsets, using $mfreq = 0.2$, we have that $\mathcal{F}_{\mathcal{D}_O} = \{A, B, C, D, E, AC, AD, AE, CD, CE, DE, ACD, ACE, ADE, CDE, ACDE\}$. Suppose that the sensitive knowledge corresponds to the frequent itemset $\mathcal{S} = \{CD\}$. Given $\mathcal{S}$ we compute $\mathcal{S}_{max} = \{CD, ACD, ACDE\}$ that corresponds to the sensitive itemsets and their frequent supersets. The revised positive border will then contain the itemsets of $\mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) = \{B, ACE, ADE\}$. The intermediate form of database $\mathcal{D}_O$ (generated using the inline approach) is shown in Table 7.2. From this database, using set $V = \{I \in \mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) : I \cap I^{\mathcal{S}} \neq \varnothing\} = \{ACE, ADE\}$, we produce the following set of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 7.3: The database $\mathcal{D}_{\mathcal{H}}$.

A	**B**	**C**	**D**	**E**
1	0	1	0	0
1	0	1	1	1
0	0	1	0	0
0	1	0	0	1
1	0	0	1	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	1	0	0

inequalities for $\mathcal{C} = V \cup \mathcal{S} = \{ACE, ADE, CD\}$ that are incorporated to the CSP:

$$\text{ACE} : u_{23} + u_{53} \geq 2 \tag{7.2}$$

$$\text{ADE} : u_{24} + u_{54} \geq 2 \tag{7.3}$$

$$\text{CD} : u_{23}u_{24} + u_{33}u_{34} + u_{53}u_{54} + u_{103}u_{104} < 2 \tag{7.4}$$

The first two inequalities correspond to itemsets ACE and ADE, which must remain frequent in $\mathcal{D}$, while the last one reflects the status of the sensitive itemset CD, which must become infrequent in the sanitized outcome. It is easy to prove that if these inequalities are incorporated to the CSP of Figure 6.1, then the produced CSP is unsolvable. Thus, we apply Algorithm 6.1 to alleviate the CSP from inequalities of set $V = \{ACE, ADE\}$. However, as mentioned earlier, we do not use the "batch" mode of operation of this algorithm (which would remove both itemsets in V) but instead, we use the "one-by-one" mode of operation that selects to remove one among the inequalities returned in the same "batch". As already stated, at this point the selection among the inequalities of the same batch is arbitrary. Suppose, we select to remove from the CSP the inequality that corresponds to itemset ACE of the revised positive border. The resulting CSP is then solvable. Among the alternative possible solutions assume that we select the one having $u_{34} = u_{53} = u_{104} = 0$ and $u_{23} = u_{24} = u_{33} = u_{54} = u_{103} = 1$. Since the criterion function of the CSP requires the maximization of the number of binary variables that are set to '1', we attain a solution

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 7.4: Intermediate form of database $\mathcal{D}_{\mathcal{H}}$.

A	**B**	**C**	**D**	**E**
1	0	1	0	u_{15}
1	0	1	1	1
u_{31}	0	1	0	u_{35}
u_{41}	1	u_{43}	0	1
1	0	u_{53}	1	1
u_{61}	0	u_{63}	1	1
u_{71}	0	1	0	u_{75}
1	1	u_{83}	0	u_{85}
1	0	1	0	u_{95}
u_{101}	0	1	0	u_{105}

that has the minimum possible variables set to '0'. This solution (having a distance of 3) is presented in Table 7.3; since it is not exact, we name our database $\mathcal{D}_{\mathcal{H}}$ and proceed to the second phase of the proposed methodology.

As we observe, in database $\mathcal{D}_{\mathcal{H}}$ of Table 7.3, itemset $\mathcal{H} = \{ACE\}$ is hidden, since its support has dropped to one. In an attempt to constitute this itemset frequent again, we create an intermediate form of database $\mathcal{D}_{\mathcal{H}}$ in which in all transactions that do not support items A, C, or E, we substitute the corresponding zero entries with binary variables u_{nm}. This, leads to the database presented in Table 7.4.

As a next step, we create the corresponding inequalities for the itemsets of set $\mathcal{C}$, by taking into account the itemsets of sets $\mathcal{H}$ and $\mathcal{B}\mathbf{d}^{-}(\mathcal{F}'_{\mathcal{H}}) = \{AB, BC, BD, BE, CD\}$ which are not foreign to $I^{\mathcal{H}} = \{A, C, E\}$. Thus, we have the following set of produced inequalities:

$$ACE : u_{15} + u_{31}u_{35} + u_{41}u_{43} + u_{53} + u_{61}u_{63} + u_{71}u_{75} + u_{83}u_{85} + u_{95} + u_{101}u_{105} \geq 1 \qquad (7.5)$$

$$AB : u_{41} < 1 \Rightarrow u_{41} = 0 \qquad (7.6)$$

$$BC : u_{43} + u_{83} < 2 \Rightarrow u_{43} = 0 \vee u_{83} = 0 \qquad (7.7)$$

$$BE : u_{85} < 1 \Rightarrow u_{85} = 0 \qquad (7.8)$$

$$CD : u_{53} + u_{63} < 1 \Rightarrow u_{53} = u_{63} = 0 \qquad (7.9)$$

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 7.5: Database $\mathcal{D}$ produced by the two–phase iterative approach.

A	B	C	D	E
1	0	1	0	1
1	0	1	1	1
0	0	1	0	0
0	1	0	0	1
1	0	0	1	1
0	0	0	1	1
0	0	1	0	0
1	1	0	0	0
1	0	1	0	0
0	0	1	0	0

Notice that no inequality is produced from itemset BD since it is foreign to set $I^{\mathcal{H}}$. Moreover, it is easy to prove that this set of inequalities is solvable, yielding two exact solutions, each of which requires only one binary variable to become '1': either u_{15} or u_{95}. These two solutions are actually the same since the involved transactions (1st or 9th) do not differ to one another and both solutions apply the same modification to either of them. Table 7.5 presents the solution where $u_{15} = 1$.

Figure 7.3 shows the original borderline along with the borderlines of the databases produced as an output of the two phases of the algorithm. Notice that the output of the second phase in the first iteration of the algorithm yields an exact solution. Thus, for $\ell = 1$ the two–phase iterative algorithm has provided an exact solution that was missed by the inline algorithm of [64].

7.4 Experimental Evaluation

In this section, we provide the results of the experiments we conducted to test the two–phase iterative approach against the inline algorithm. The two algorithms were tested on three real world datasets using different parameters such as minimum support threshold and number/size of sensitive itemsets to hide. All these datasets are publicly available through the FIMI repository [80] and are summarized in Table 6.4.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

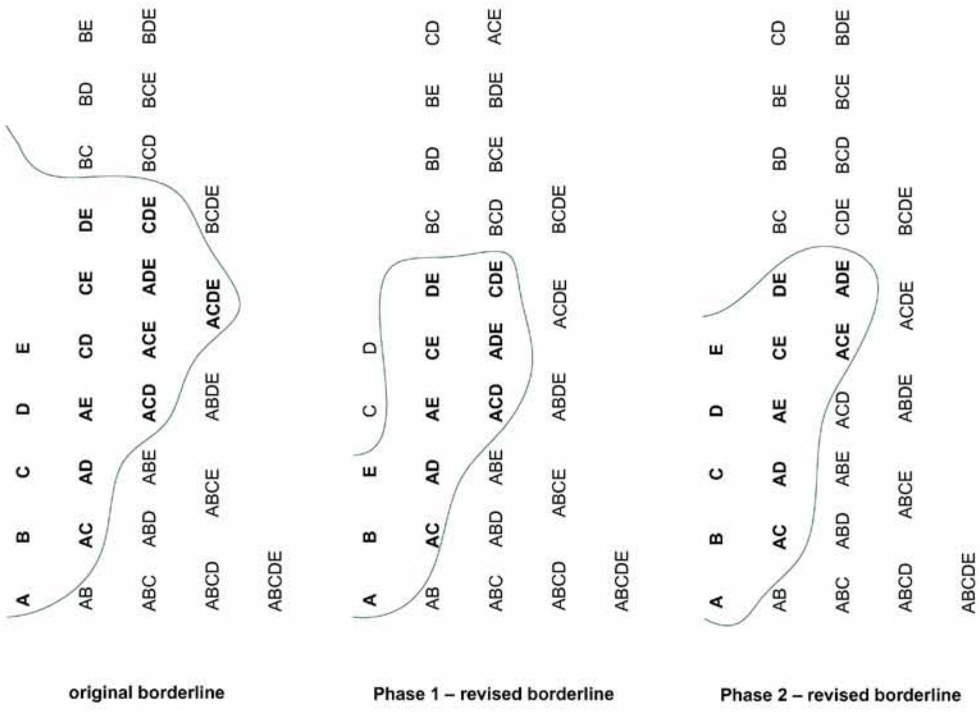


Figure 7.3: The two phases of iteration for the considered example.

The primary bottleneck that we experienced in most of our experiments was the time taken to run the frequent itemset mining algorithm, as well as the time needed to solve the formulated CSPs through the application of BIP. In all tested settings, the thresholds of minimum support were properly selected to ensure an adequate amount of frequent itemsets and the sensitive itemsets to be hidden were selected randomly among the frequent ones. Our source code was implemented in Perl and C and all the experiments were conducted on a PC running Linux on an Intel Pentium D, 3.2 Ghz processor equipped with 4 GB of main memory. All integer programs were solved using ILOG CPLEX 9.0 [95].

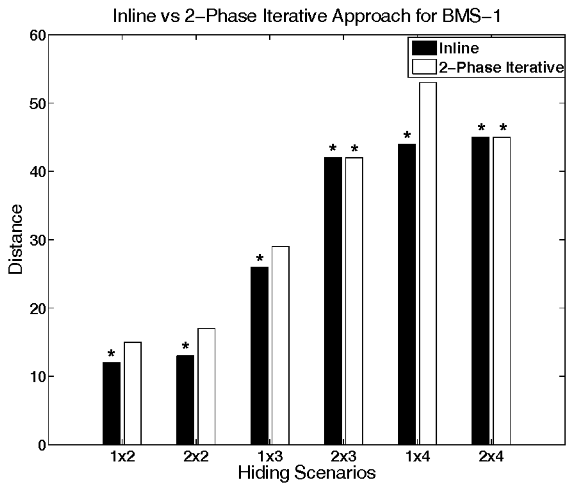
When evaluating the two–phase iterative approach, we used a limiting factor of $\ell = 5$ to control the number of iterations that the algorithm is allowed to execute. After the execution of the first iteration of the two–phase iterative algorithm, we keep track of the attained solution and the respective impact on the dataset. Then, we allow the algorithm to

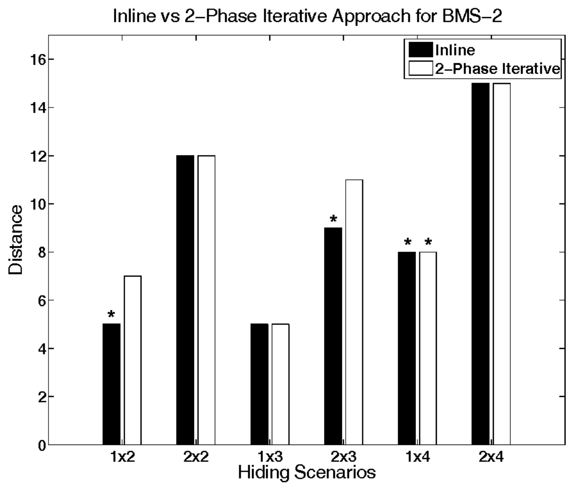
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

proceed to up to ℓ subsequent iterations. If the algorithm fails to identify an exact solution after the ℓ runs, we refer to the stored solution of the first iteration of the algorithm to attain the sanitized database $\mathcal{D}$. This way, we achieve both to limit the runtime of execution of the algorithm to a level that it remains tractable (through the use of ℓ) and to ensure that this algorithm will constantly outperform the inline scheme and provide superior hiding solutions. In what follows, let notation $a \times b$ denote the hiding of a itemsets of length b.

Figure 7.4 presents the performance comparison of the two algorithms. A star above a column in the graphs indicates a suboptimal hiding solution. Based on the attained results, we can make the following observations. First, by construction, the two–phase iterative scheme is constantly superior to the inline algorithm, since its worst performance equals the performance of the inline scheme. Second, as the experiments indicate, there are several settings in which the two–phase iterative algorithm finds an exact hiding solution with a small increment in the distance when compared to that of the inline approach. Third, by construction, the two–phase iterative algorithm can capture all the exact solutions that were also identified by the inline approach. This fact, constitutes the two–phase iterative algorithm superior when compared to the inline methodology.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136





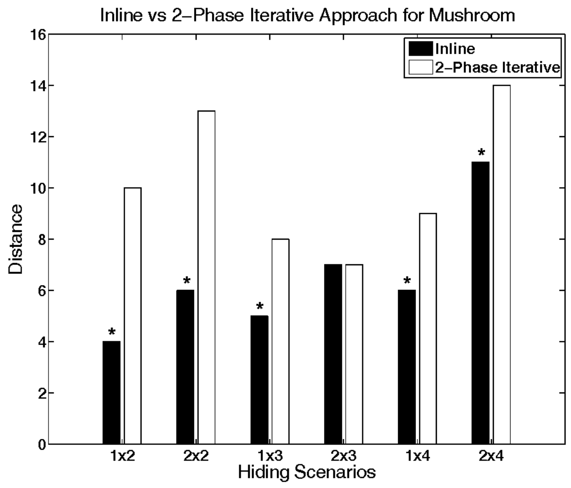


Figure 7.4: The distance between the inline and the two–phase iterative algorithm.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 8

Exact Knowledge Hiding Through Database Extension

In this chapter, we propose a novel, exact border–based approach that provides an optimal solution for the hiding of sensitive frequent itemsets by (i) minimally extending the original database by a synthetically generated database part — the database extension, (ii) formulating the creation of the database extension as a constraint satisfaction problem, (iii) mapping the constraint satisfaction problem to an equivalent binary integer programming problem, (iv) exploiting underutilized synthetic transactions to proportionally increase the support of nonsensitive itemsets, (v) minimally relaxing the constraint satisfaction problem to provide an approximate solution close to the optimal one when an ideal solution does not exist, and (vi) by using a partitioning in the universe of the items to increase the efficiency of the proposed hiding algorithm. Extending the original database for sensitive itemset hiding is proved to provide optimal solutions to an extended set of hiding problems compared to previous approaches and to provide hiding solutions of typically higher quality. Moreover, the application of binary integer programming enables the simultaneous hiding of all the sensitive itemsets and thus allows for the identification of globally optimal solutions.

The remainder of this chapter is organized as follows. Section 8.1 introduces the proposed methodology. In Section 8.2 we provide the basic steps of the methodology, while

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Section 8.3 sheds light on the main issues that pertain to its successful application. Section 8.4 presents the various aspects of the hybrid solution methodology, while Section 8.5 introduces a partitioning approach which substantially improves the scalability of the algorithm. Finally, Section 8.6 contains the experimental evaluation contrasting this approach against the exact algorithm of Chapter 6 as well as two state–of–the–art heuristic methodologies.

8.1 Introduction

In this chapter, we present a novel approach that strategically performs sensitive frequent itemset hiding based on a new notion of hybrid database generation. This approach broadens the regular process of data sanitization (as introduced in [20] and adopted by the overwhelming majority of researchers [20, 122, 134]) by applying an extension to the original database instead of either modifying existing transactions (directly or through the application of transformations), or rebuilding the dataset from scratch to accommodate sensitive knowledge hiding. The extended portion of the database contains a set of carefully crafted transactions that achieve to lower the importance of the sensitive patterns to a degree that they become uninteresting from the perspective of the data mining algorithm, while minimally affecting the importance of the nonsensitive ones. The hiding process is guided by the need to maximize the data utility of the sanitized database by introducing the least possible amount of side–effects, such as (i) the hiding of nonsensitive patterns, or (ii) the production of frequent patterns that were not existent in the initial dataset (ghost itemsets). The released database, which consists of the initial part (original database) and the extended part (database extension), can guarantee the protection of the sensitive knowledge, when mined at the same or higher support as the one used in the original database. Therefore, to protect the sensitive knowledge, the security administrator should appropriately set the minimum support threshold to a value that is lower than the usually used threshold in a typical mining scenario concerning the data at hand. In accordance to the majority of the knowledge hiding approaches, this work aims at creating a sanitized version of the original database that can be safely released to the public.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Following previous work [64, 70], the approach presented in this chapter is exact in nature; provided that a hiding solution that causes no side–effects in the sanitized database exists, the proposed algorithm is guaranteed to find it. On the contrary, when an exact solution is impossible, the algorithm identifies an approximate solution that is close to the optimal one. To accomplish the hiding task, the proposed approach administers the sanitization part by formulating a CSP and by solving it through BIP. The measure of distance [64] is used to formulate the criterion that will drive the optimization process to the optimal solution. Through a set of experiments we demonstrate the effectiveness of this approach towards identifying optimal hiding solutions bearing no side–effects.

8.2 Knowledge Hiding Formulation

This section sets out the proposed hiding methodology, along with a running example which helps towards its proper understanding.

8.2.1 The Hiding Methodology

To properly introduce the hiding methodology one needs to consider the existence of three databases, all depicted in binary format. They are defined as follows:

- Database $\mathcal{D}_O$ is the original transaction database which, when mined at a certain support threshold *msup*, leads to the disclosure of some sensitive knowledge in the form of sensitive frequent patterns. This sensitive knowledge needs to be protected.
- Database $\mathcal{D}_X$ is a minimal extension of $\mathcal{D}_O$ that is created by the hiding algorithm during the sanitization process, in order to facilitate knowledge hiding.
- Database $\mathcal{D}$ is the union of database $\mathcal{D}_O$ and the applied extension $\mathcal{D}_X$ and corresponds to the sanitized outcome that can be safely released.

Suppose that database $\mathcal{D}_O$ consists of N transactions. By preforming frequent itemset mining in $\mathcal{D}_O$, using a support threshold *msup* set by the owner of the data, a set of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.1: The sanitized database $\mathcal{D}$ as a mixture of the original database $\mathcal{D}_O$ and the applied extension $\mathcal{D}_X$.

	a	**b**	**c**	**d**	**e**	**f**
$\mathcal{D}_O$	1	1	0	0	0	1
	1	1	1	1	0	0
	1	0	1	0	0	1
	1	0	0	0	0	0
	0	1	0	0	1	0
	1	1	1	1	1	0
	0	0	0	1	0	0
	1	1	1	0	1	0
	0	1	1	0	0	0
	1	0	1	1	1	0
$\mathcal{D}_X$	1	0	0	0	0	0
	1	0	1	1	0	0
	1	0	1	1	0	0
	1	1	0	0	0	0

frequent patterns $\mathcal{F}_{\mathcal{D}_O}$ are discovered, among which, a subset $\mathcal{S}$ contains patterns which are sensitive from the owner's perspective. The goal of the proposed hiding algorithm is to create a minimal extension to the original database $\mathcal{D}_O$ in a way that the final, sanitized database $\mathcal{D}$ protects the sensitive itemsets from disclosure. The database extension can by itself be considered as a new database $\mathcal{D}_X$, since it consists of a set of transactions in the same space of items I as the ones of $\mathcal{D}_O$. Among alternative hiding solutions that may exist, the target of the proposed algorithm is to protect the sensitive knowledge, while minimally affecting the nonsensitive itemsets appearing in $\mathcal{F}_{\mathcal{D}_O}$. This means that all the nonsensitive itemsets in $\mathcal{F}_{\mathcal{D}_O}$ should continue to appear as frequent among the mined patterns from $\mathcal{D}$, when performing frequent itemset mining using the same or a higher support threshold. The hiding of a sensitive itemset is equivalent to a degradation of its statistical significance, in terms of support, in the result database.

The proposed algorithm first applies border revision to identify the revised borders for $\mathcal{D}$, then computes the minimal size for the extension $\mathcal{D}_X$ and, by using the itemsets of the revised borders, defines a CSP that is solved using BIP. In this way, all possible assignments of itemsets to the transactions of $\mathcal{D}_X$ are examined and the optimal assignment is bound to

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.2: Frequent itemsets for $\mathcal{D}_O$ and $\mathcal{D}_X$ at $msup = 3$.

Frequent itemset in $\mathcal{D}_O$	Support
{a}	7
{b}, {c}	6
{ac}	5
{d}, {e}, {ab}, {bc}	4
{ad}, {ae}, {be}, {cd}, {ce}, {abc}, {acd}, {ace}	3

Frequent itemset in $\mathcal{D}$	Support
{a}	11
{c}	8
{b}, {ac}	7
{d}	6
{ab}, {ad}, {cd}, {acd}	5
~~{e}~~, ~~{bc}~~	4
~~{ae}~~, ~~{be}~~, ~~{ce}~~, ~~{abc}~~, ~~{ace}~~	3

be found. Although the properties of the produced CSP allow for an acceptable runtime of the hiding algorithm, there are cases in which the partitioning approach of Section 8.5 becomes useful to accommodate for very large problem sizes.

8.2.2 A Running Example

Suppose we are provided with database $\mathcal{D}_O$ of Table 8.1. Applying frequent itemset mining in $\mathcal{D}_O$ using $mfreq = 0.3$, leads to the set of large itemsets $\mathcal{F}_{\mathcal{D}_O}$ appearing in the upper part of Table 8.2. Among these itemsets, let $\mathcal{S} = \{e, ae, bc\}$ denote the sensitive knowledge that has to be protected. The proposed hiding algorithm aims at the creation of a database extension $\mathcal{D}_X$ to $\mathcal{D}_O$ (see Table 8.1) that allows the hiding of the sensitive knowledge, while keeping the nonsensitive patterns frequent in the sanitized outcome.

Table 8.1 summarizes the target of the hiding algorithm. The union of the two datasets $\mathcal{D}_O$ and $\mathcal{D}_X$ corresponds to the sanitized outcome $\mathcal{D}$ that can be safely released. Thus, the primary goal of the hiding algorithm is to construct the privacy aware extension $\mathcal{D}_X$ of $\mathcal{D}_O$ such that (i) it contains the least amount of transactions needed to ensure the proper hiding of all the sensitive knowledge in $\mathcal{D}_O$, and (ii) it introduces no side–effects

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

to the hiding process. As we can observe in the lower part of Table 8.2, all the sensitive itemsets of $\mathcal{D}_O$ along with their supersets are infrequent in $\mathcal{D}$ (shown under the dashed line), while all the nonsensitive itemsets of $\mathcal{D}_O$ remain frequent. Since $\mathcal{D}_O$ is extended, in order to ensure that the nonsensitive patterns will remain frequent in $\mathcal{D}$, the hiding algorithm needs to appropriately increase their support in the sanitized database.

8.3 Main Issues Pertaining to the Hiding Methodology

The proposed hiding methodology creates a sanitized database $\mathcal{D}$ that corresponds to a mixture of the original transactions in $\mathcal{D}_O$ and a set of synthetic transactions, artificially created to prohibit the leakage of the sensitive knowledge. For security reasons, all the transactions in $\mathcal{D}$ are assumed to be randomly ordered so that it is difficult for an adversary to distinguish between the real ones and those that were added by the hiding algorithm to secure the sensitive knowledge. There are several issues of major importance, involving the hiding methodology, that need to be examined. To continue, let P denote the size of database $\mathcal{D}$, N is the size of database $\mathcal{D}_O$, and Q is the size of the extension $\mathcal{D}_X$.

8.3.1 Size of the Database Extension

Since database $\mathcal{D}_O$ is extended by $\mathcal{D}_X$ to construct database $\mathcal{D}$, an initial and very important step in the hiding process is the computation of the size of $\mathcal{D}_X$ (i.e. the necessary amount of transactions that need to be added to those of $\mathcal{D}_O$ to facilitate the hiding of the sensitive knowledge). A lower bound on this value can be established based on the sensitive itemset in $\mathcal{S}$ which has the highest support (breaking ties arbitrarily). The rationale here is as follows; by identifying the sensitive itemset with the highest support, one can safely decide upon the minimum number of transactions that must not support this itemset in $\mathcal{D}_X$, so that it becomes infrequent in $\mathcal{D}$. This number, theoretically, is sufficient to allow the hiding of all the other itemsets participating in $\mathcal{S}$ and all its supersets, and corresponds to the minimum number of transactions that $\mathcal{D}_X$ must have to properly secure the sensitive knowledge.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Theorem 1 demonstrates how this lower bound Q is established.

Theorem 1. *Let $I_M \in \mathcal{S}$ such that for all $I \in \mathcal{S}$ it holds that $sup(I_M, \mathcal{D}_O) \geq sup(I, \mathcal{D}_O)$. Then, the minimum size of $\mathcal{D}_X$ to allow the hiding of the sensitive itemsets from $\mathcal{S}$ in $\mathcal{D}$ equals:*

$$Q = \left\lfloor \frac{sup(I_M, \mathcal{D}_O)}{mfreq} - N \right\rfloor + 1 \tag{8.1}$$

Proof. We only need to prove that any itemset $I \in \mathcal{S}$ will become hidden in $\mathcal{D}$ if and only if $Q > \text{sup}(I, \mathcal{D}_O)/mfreq - N$, provided that I is not supported in $\mathcal{D}_X$. Since itemset I must be infrequent in database $\mathcal{D}$, the following condition holds:

$$sup(I, \mathcal{D}) < msup \Longrightarrow sup(I, \mathcal{D}_O) + sup(I, \mathcal{D}_X) < mfreq \cdot P$$

Moreover, since $sup(I, \mathcal{D}_X) \geq 0$ and (by construction) $P = N + Q$, we have that

$$sup(I, \mathcal{D}_O) < mfreq \cdot (N + Q) \Longrightarrow sup(I, \mathcal{D}_O) < N + Q$$

The last inequality was relaxed by removing term $sup(I, \mathcal{D}_X)$. Since it consists of the summation of non–negative terms and a non–negative term was removed, the inequality will continue to hold. Its holding proves the holding of (8.1) since the sensitive itemset with the highest support will require the largest amount of transactions (not supporting it) in the extension $\mathcal{D}_X$ in order to be properly hidden. The lower bound of the number of the necessary transactions for $\mathcal{D}_X$ will thus equal the floor value of $\text{sup}(I_M, \mathcal{D}_O)/mfreq - N$ plus one. Moreover, as expected, itemsets having lower support than I_M may be supported by some transactions of database $\mathcal{D}_X$, as long as they are infrequent in $\mathcal{D}$. □

Equation (8.1) provides the absolute minimum size of $\mathcal{D}_X$ to accommodate for the sensitive knowledge hiding. However, as is presented later on, this lower bound may, under certain circumstances, be insufficient to allow for the identification of an exact hiding solution, even if such a solution exists. This situation may occur if, for instance, the number of transactions returned by (8.1) is too small to allow for consistency among the different requirements imposed upon the status (frequent vs infrequent) of the various itemsets appearing in $\mathcal{D}$. Later on, we propose a way to overcome this limitation.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

8.3.2 Exact and Ideal Solutions in the Hybrid Methodology

Having identified the size Q of database $\mathcal{D}_X$, the next step is to properly construct these transactions to facilitate knowledge hiding. Since the actual values of all the items in the database extension are unknown at this point, the hiding algorithm represents them with binary variables that will be instantiated later on in the process. In what follows, let u_{qm} be the binary variable corresponding to the m–th item of transaction $T_q \in \mathcal{D}_X$ ($q \in [1,Q], m \in [1,M]$), when $\mathcal{D}_O$ is in the sanitization process. Under this formulation, the goal of the hiding algorithm becomes to optimally adjust all the binary variables involved in all the transactions of $\mathcal{D}_X$ in order to hide the sensitive itemsets, while minimally affecting the nonsensitive ones in a way that they remain frequent in the sanitized outcome. As already presented in Chapter 2, this is the notion of an *exact* hiding solution.

In a typical hiding scenario, distinct feasible solutions are of different quality. Thus, an optimization criterion needs to be incorporated into the hiding strategy to guide the algorithm to the best possible among all the feasible solutions. The metric of *distance*, introduced in [64], is applied to quantify the notion of "harm" caused to the original dataset by the sanitization process. In the context of this work, the distance between database $\mathcal{D}_O$ and its sanitized version $\mathcal{D}$ is measured based on the extension $\mathcal{D}_X$ as follows:

$$dist(\mathcal{D}_O, \mathcal{D}) = \sum_{q \in [1,Q], m \in [1,M]} u_{qm} \tag{8.2}$$

As one can observe, the minimum impact of $\mathcal{D}$ can be quantified as the minimum distance between $\mathcal{D}_O$ and $\mathcal{D}$. Thus, the objective of the hiding algorithm becomes to appropriately set the u_{qm} variables such that all the sensitive knowledge is hidden, while the distance is minimized. An interesting property of the distance measure is that it allows the hiding algorithm to ensure high quality in the sanitized database $\mathcal{D}$ and to identify the ideal solution, if one exists. The notion of an ideal solution in the context of this work is presented in Definition 2. Based on the notion of distance and the size of the produced extension $\mathcal{D}_X$ of database $\mathcal{D}_O$, the database quality is defined as follows:

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Definition 1. *(**Database quality**) Given the sanitized database $\mathcal{D}$, its original version $\mathcal{D}_O$ and the produced extension $\mathcal{D}_X$, the **quality** of database $\mathcal{D}$ is measured both in the size of $\mathcal{D}_X$ and in the number of binary variables set to '1' in the transactions of $\mathcal{D}_X$ (i.e. the distance metric). In both cases, lower values correspond to better solutions.*

Through (8.1) the hiding algorithm is capable of identifying the lower bound in the size of $\mathcal{D}_X$ that is necessary to accommodate the hiding of the sensitive knowledge in $\mathcal{D}_O$. As we demonstrate in Sections 8.4.2 and 8.4.4, in any situation, a specific process can be followed to ensure that the size of the extension $\mathcal{D}_X$ is minimized. Furthermore, through the use of (8.2) as an optimization criterion, the proposed hiding approach is guaranteed to identify a feasible solution, having the minimum impact on database $\mathcal{D}$. Given the previous definitions, we now proceed to define the notion of an ideal solution.

Definition 2. *(**Ideal solution**) A solution to the hiding of the sensitive itemsets is considered as **ideal** if it has the minimum distance among all the existing exact solutions and is obtained through the minimum expansion of $\mathcal{D}_O$. In that sense, ideal is a solution that is both minimal (with respect to the distance and the size of the extension) and exact.*

8.3.3 The Revision of the Borders

The concept of border revision, introduced by Sun and Yu [160] and presented in detail in Chapter 5, provides the underlying mechanism for the specification of the values of the u_{qm} variables to 1s or 0s, in a way that minimizes the impact on $\mathcal{D}$. The use of border revision allows the computation of the revised borders that lead to an exact solution and assists in the tracking of the impact of the hiding process on the result database. Thus, in order to maintain high quality on the result database, the hiding algorithm should select such item modifications in $\mathcal{D}_X$ that have minimal impact on the border of the nonsensitive itemsets.

The first step in the proposed hiding methodology rests on the identification of the revised borders for $\mathcal{D}$. The hiding algorithm relies on both the revised positive and the negative borders, denoted as $\mathcal{B}d^+(\mathcal{F}'_\mathcal{D})$ and $\mathcal{B}d^-(\mathcal{F}'_\mathcal{D})$, respectively. These borders can be efficiently computed using the algorithms presented in Chapter 5. After identifying the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

revised borders, the hiding process has to perform all the required minimal adjustments of the transactions in $\mathcal{D}_X$ to enforce the existence of the new borderline in the result database.

Continuing the example of Section 8.2.2, consider the lattice of Figure 5.1, corresponding to the database $\mathcal{D}_O$ of Table 8.1, when applying frequent itemset mining at $mfreq = 0.3$. Near each itemset we depict its support in the original (Figure 5.1(i)) and the sanitized (Figure 5.1(ii)) database. Based on the original hyperplane, the following borders can be identified for $\mathcal{D}_O$: $\mathcal{B}d^+(\mathcal{F}_{\mathcal{D}_O}) = \{abc, be, acd, ace\}$ and $\mathcal{B}d^-(\mathcal{F}_{\mathcal{D}_O}) = \{f, bd, de, abe, bce\}$. The original borders are presented in Figure 5.1(i), where the itemsets belonging to the positive border are double underlined and the ones belonging to the negative border are single underlined. As one can notice, in the lattice of this figure all frequent itemsets lie at the left of the respective border, while their infrequent counterparts are on the right. Given that $\mathcal{S} = \{e, ae, bc\}$ (shown in squares in Figure 5.1(i)), we have that $\mathcal{S}_{min} = \{e, bc\}$ (shown in bold in Figure 5.1(i)) and $\mathcal{S}_{max} = \{e, ae, be, bc, ce, abc, ace\}$. Ideally, the frequent itemsets in the sanitized database $\mathcal{D}$ will be exactly those in $\mathcal{F}'_{\mathcal{D}} = \{a, b, c, d, ab, ac, ad, cd, acd\}$. The revised borderline along with the corresponding borders for $\mathcal{D}$ and scenarios C_1 and C_3 that must hold for an exact solution, are depicted in Figure 5.1(ii). Scenario C_2 corresponds to the itemsets lying on the right of the original border. To enhance the clarity of the figure only a small portion of the itemsets involved in C_2 are shown in Figure 5.1. The revised borders that pertain to an optimal solution are: $\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}}) = \{ab, acd\}$ (double underlined in Figure 5.1(ii)) and $\mathcal{B}d^-(\mathcal{F}'_{\mathcal{D}}) = \{e, f, bc, bd\}$ (single underlined in Figure 5.1(ii)). What is thus needed is a way to enforce the existence of the revised borders in $\mathcal{D}$.

8.3.4 Problem Size Reduction

To enforce the computed revised border and identify the exact hiding solution, a mechanism is needed to regulate the status (frequent vs infrequent) of all the itemsets in $\mathcal{D}$. Let $\mathcal{C}$ be the minimal set of border itemsets used to regulate the values of the various u_{qm} variables in $\mathcal{D}_X$. Moreover, suppose that $I \in \mathcal{C}$ is an itemset, whose behavior we want to regulate in $\mathcal{D}$. Then, itemset I will be frequent in $\mathcal{D}$ if and only if $\sup(I, \mathcal{D}_O) + \sup(I, \mathcal{D}_X) \geq mfreq \times (N + Q)$,

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

or equivalently if:

$$\sup(I, \mathcal{D}_O) + \sum_{q=1}^{Q} \prod_{i_m \in I} u_{qm} \geq mfreq \times (N + Q) \tag{8.3}$$

and will be infrequent otherwise, when

$$\sup(I, \mathcal{D}_O) + \sum_{q=1}^{Q} \prod_{i_m \in I} u_{qm} < mfreq \times (N + Q) \tag{8.4}$$

Inequality (8.3) corresponds to the minimum number of times that an itemset I has to appear in the extension $\mathcal{D}_X$ to remain frequent in $\mathcal{D}$. On the other hand, inequality (8.4) provides the maximum number of times that an itemset I can appear in $\mathcal{D}_X$ in order to remain infrequent in database $\mathcal{D}$.

To identify an exact solution to the hiding problem, every possible itemset in $\mathcal{P}$, according to its position in the lattice — with respect to the revised border — must satisfy either (8.3) or (8.4). However, the complexity of solving the entire system of the $2^M - 1$ inequalities is well known to be *NP*–hard [20, 122]. Therefore, one should restrict the problem to capture only a small subset of these inequalities, thus leading to a problem size that is computationally manageable. The proposed problem formulation achieves this by reducing the number of the participating inequalities that need to be satisfied. Even more, by carefully selecting the itemsets of set $\mathcal{C}$, the hiding algorithm ensures that the exact same solution to the one of solving the entire system of inequalities, is attained. This is accomplished by exploiting cover relations existing among the itemsets in the lattice due to the monotonicity of support (the reader can refer to [14]).

Definition 3. ***(Cover/Generalized cover)*** *Given itemsets $I, J, K \in \mathcal{P}$, itemset I is defined as a **cover** of K if and only if $I \supset K$ and there exists no proper itemset J such that $I \supset J \supset K$. Itemset I is a **generalized cover** of K if and only if $I \supset K$. Given two itemsets $I, J \in \mathcal{P}$, J **covers** I if and only if $J \supset I$, i.e. if itemset J is a (generalized) cover of itemset I.*

The basic premises for formulating set $\mathcal{C}$ are the following. All frequent itemsets in $\mathcal{F}'_{\mathcal{D}}$ should remain frequent in $\mathcal{D}$, thus inequality (8.3) must hold. On the contrary, all frequent itemsets in $\mathcal{S}_{min}$ should satisfy inequality (8.4) in order to be hidden in $\mathcal{D}$. By

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

definition, the hiding of the itemsets in $\mathcal{S}_{min}$ will cause the hiding of the itemsets in $\mathcal{S}_{max}$. Additionally, all infrequent itemsets in $\mathcal{D}_O$ should remain infrequent in $\mathcal{D}$. For these reasons, set $\mathcal{C}$ is chosen appropriately to consist of all the itemsets of the revised border. The proposed hiding algorithm is capable of ensuring that if the inequalities (8.3) and (8.4) are satisfied for all the itemsets in set $\mathcal{C}$, then the produced solution is exact and is identical to the solution involving the whole system of the $2^M - 1$ inequalities.

8.3.5 Handling of Suboptimality

Since an exact solution may not always be feasible, the hiding algorithm should be capable of identifying good approximate solutions. There are two possible scenarios that may lead to non–existence of an exact solution. Under the first scenario, $\mathcal{D}_O$ itself does not allow for an optimal solution due to the various supports of the participating itemsets. Under the second scenario, database $\mathcal{D}_O$ is capable of providing an exact solution but the size of the database extension is insufficient to satisfy all the required inequalities of this solution.

To tackle the first case, the hiding algorithm assigns different degrees of importance to different inequalities. To be more precise, while it is crucial to ensure that (8.4) holds for all sensitive itemsets in $\mathcal{D}$, thus they are properly protected from disclosure, satisfaction of (8.3) for an itemset rests in the discretion of ensuring the minimal possible impact of the sanitization process to $\mathcal{D}_O$. This inherent difference in the significance of the two inequalities, along with the fact that solving the system of all inequalities of the form (8.4) always leads to a feasible solution (i.e. for any database $\mathcal{D}_O$), allows the relaxation of the problem when needed and the identification of a good approximate solution.

To overcome the second issue, the hiding algorithm incorporates the use of a safety margin threshold, which produces a further expansion of $\mathcal{D}_X$ by a certain number of transactions. These transactions must be added to the ones computed by using (8.1). The introduction of a safety margin can be justified as follows. Since equation (8.1) provides the lower bound on the size of database $\mathcal{D}_X$, it is possible that the artificially created transactions are too few to accommodate for the proper hiding of knowledge. This situation

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

may occur due to conflicting constraints imposed by the various itemsets regarding their status in $\mathcal{D}$. These constraints require more transactions (or to be more precise, more item modifications) in order to be met. Thus, a proper safety margin will allow the algorithm to identify an exact solution if such a solution exists. Moreover, as is demonstrated in Section 8.4.4, the additional extension of $\mathcal{D}_X$, due to the incorporation of the safety margin, can be restricted to the necessary degree. A portion of the transactions in $\mathcal{D}_X$ are selected and removed at a later point, thus reducing its size and allowing an exact solution. Therefore, the only side–effect of using a safety margin in the hiding process, is an inflation in the number of constraints and associated binary variables in the problem formulation, leading to a minuscule overhead in the runtime of the hiding algorithm.

8.4 Hybrid Solution Methodology

In the following sections, we present a way to minimize the problem size by regulating the status of only an essential portion of the itemsets from $\mathcal{P}$. Moreover, we propose a solution to the size of the extension $\mathcal{D}_X$ and formulate the hiding process as a CSP that is solved by using BIP. Finally, the critical issues of how to ensure validity of transactions in $\mathcal{D}$ and how to handle suboptimality in the approximate solutions, are appropriately addressed.

8.4.1 Problem Size Minimization

Cover relations governing the itemsets in the lattice of $\mathcal{D}_O$, ensure that the formulated set of itemsets $\mathcal{C}$ has an identical solution to the one of solving the system of all $2^M - 1$ inequalities for $\mathcal{D}$. Since the status of each itemset in $\mathcal{C}$ needs to be regulated in $\mathcal{D}_X$, a one–to–one correspondence exists between itemsets and produced inequalities in the system.

Definition 4. ***(Solution of a system of inequalities from*** $\mathcal{C}$***)*** *Given set* $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{2^M-1}\}$*, let* $\mathcal{L}_\mathcal{C}$ *be the set of solutions of the corresponding inequalities from* $\mathcal{C}$*. The set of solutions for the system of inequalities will be the intersection of the solutions produced by each inequality. Thus,* $\mathcal{L}_\mathcal{C} = \bigcap_{r=1}^{R} \mathcal{L}_{\mathcal{C}_r}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Based on this notation, the set of solutions that corresponds to all itemsets in $\mathcal{D}$ is denoted as $\mathcal{L}_{\mathcal{P}\setminus\{\varnothing\}}$. Given an inequality, for instance $\mathcal{C}_2$, produced by an itemset I with a solution set being a proper subset of the solution set of another inequality, say $\mathcal{C}_1$, one can deduce that the latter inequality can be removed from any system containing the first inequality, without affecting the global solution of this system. In this case $\mathcal{L}_{\mathcal{C}_2} \subset \mathcal{L}_{\mathcal{C}_1}$ and $\mathcal{C}_1$ is a (generalized) cover of $\mathcal{C}_2$.

The following theorems identify itemsets that should be represented in the CSP formulation and prove that the optimal hiding solution $\mathcal{L}_{\mathcal{C}}$ can be found based on the union of the borders $\mathcal{B}d^{+}(\mathcal{F}'_{\mathcal{D}})$ and $\mathcal{B}d^{-}(\mathcal{F}'_{\mathcal{D}})$.

Theorem 2. *If $I \in \mathcal{F}'_{\mathcal{D}}$ then $\forall J \subset I$ we have $\mathcal{L}_I \subset \mathcal{L}_J$.*

Proof. Consider the inequality produced by I, which should be frequent in $\mathcal{D}$, i.e.

$$\sum_{q=1}^{Q} \prod_{i_m \in I} u_{qm} \geq msup - \sup(I, \mathcal{D}_O) \tag{8.5}$$

Suppose that this inequality holds for a combination of u_{qm} values, corresponding to a solution $l \in \mathcal{L}_I$. If all the u_{qm} variables in the extension $\mathcal{D}_X$ are substituted with their values from l, then every bit will gain a specific value. Let $\mathcal{D}_I$ denote the supporting transactions of itemset I in database $\mathcal{D}$ and index $p \in [1, P]$ be used to capture transactions in $\mathcal{D}$. Since $J \subset I$, it holds that $\mathcal{D}_I \subset \mathcal{D}_J$. Then, provided that $A = \sum_{T_q \in (\mathcal{D}_J \setminus \mathcal{D}_I)} \prod_{i_m \in J} u_{qm}$ the following must hold for database $\mathcal{D}$:

$$A + \sup(J, \mathcal{D}_O) - \sup(I, \mathcal{D}_O) \geq 0 \tag{8.6}$$

From (8.5) and (8.6) we have that:

$$A + \sum_{q=1}^{Q} \prod_{i_m \in I} u_{qm} \geq msup - \sup(J, \mathcal{D}_O)$$

and therefore

$$\sum_{q=1}^{Q} \prod_{i_m \in J} u_{qm} \geq msup - \sup(J, \mathcal{D}_O)$$

□

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Theorem 3. *If $\mathcal{F}_{\mathcal{D}}$ is the set of all frequent itemsets in $\mathcal{D}$, then $\mathcal{F}_{\mathcal{D}} = \cup_{I \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})} \wp(I)$.*

Proof. Due to the definition of the positive border $\mathcal{B}d^+(\mathcal{F}_{\mathcal{D}}) \subseteq \mathcal{F}_{\mathcal{D}}$. Based on the downward closure property of the frequent itemsets, all subsets of a frequent itemset are also frequent. Therefore, all subsets of $\mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})$ must be frequent, which means that $\cup_{I \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})} \wp(I) \subseteq \mathcal{F}_{\mathcal{D}}$. To prove the theorem, we need to show that $\mathcal{F}_{\mathcal{D}} \subseteq \cup_{I \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})} \wp(I)$ or (equivalently) that if $I \in \mathcal{F}_{\mathcal{D}}$, then $I \in \cup_{K \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})} \wp(K) \Rightarrow \exists J \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}}) : I \subseteq J$. Assume that this condition does not hold. Then, it must hold that $I \in \mathcal{F}_{\mathcal{D}}$ and that $\nexists J \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}}) : I \subseteq J$. From the last two conditions we conclude that $I \notin \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})$ since, if $I \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})$ then the second condition does not hold for $J = I$. Therefore, and due to the definition of the positive border, we have that $\exists J_1 \in \mathcal{F}_{\mathcal{D}} : I \subset J_1$. Thus, $|J_1| > |I| \Rightarrow |J_1| \geq |I| + 1 \Rightarrow |J_1| \geq 1$. Moreover, since $\nexists J \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}}) : I \subseteq J$ it means that $J_1 \notin \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})$, and since $J_1 \in \mathcal{F}_{\mathcal{D}}$, $\exists J_2 \in \mathcal{F}_{\mathcal{D}} : J_1 \subset J_2$, therefore $|J_2| > |J_1| \Rightarrow |J_2| \geq |J_1| + 1$. Finally, since $|J_1| \geq 1$ we conclude that $|J_2| \geq 2$.

As it can be noticed, using induction it is easy to prove that $\forall k \in \aleph^* : \exists J_k \in \mathcal{F}_{\mathcal{D}} : |J_k| \geq k$. This means that there is no upper bound in the size of frequent itemsets, a conclusion that is obviously wrong. Therefore, the initial assumption that $I \in \mathcal{F}_{\mathcal{D}}$ and $\nexists J \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}}) : I \subseteq J$ does not hold which means that $\mathcal{F}_{\mathcal{D}} \subseteq \cup_{I \in \mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})} \wp(I)$. □

Theorems 2 and 3 prove the cover relations that exist between the itemsets of $\mathcal{B}d^+(\mathcal{F}_{\mathcal{D}})$ and those of $\mathcal{F}'_{\mathcal{D}}$. In the same manner one can prove that the itemsets of $\mathcal{B}d^-(\mathcal{F}_{\mathcal{D}})$ are generalized covers for all the itemsets of $\mathcal{P} \backslash (\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}}) \cup \{\varnothing\})$. Therefore, the itemsets of the positive and the negative borders cover all the itemsets in $\mathcal{P}$. As a result, the following Corollary to Theorems 2 and 3 holds.

Corollary 1. ***(Optimal solution set** $\mathcal{C}$**)*** *The exact hiding solution, which is identical to the solution of the entire system of the $2^M - 1$ inequalities, can be attained based on the itemsets of set*

$$\mathcal{C} = \mathcal{B}\mathbf{d}^+(\mathcal{F}'_{\mathcal{D}}) \cup \mathcal{B}\mathbf{d}^-(\mathcal{F}'_{\mathcal{D}}) \tag{8.7}$$

Based on (8.7) the itemsets of the revised borders $\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$ and $\mathcal{B}d^-(\mathcal{F}'_{\mathcal{D}})$ can be used to produce the inequalities, which will lead to an exact hiding solution for $\mathcal{D}_O$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{minimize}\left(\sum_{q\in[1,Q+\mathrm{SM}],m\in[1,M]} u_{qm}\right)$$

$$\textit{subject to}\begin{cases}\sum_{q=1}^{Q+\mathrm{SM}}\prod_{i_m\in I}u_{qm} < thr, \forall I \in \mathcal{B}d^{-}(\mathcal{F}'_{\mathcal{D}})\\ \\ \sum_{q=1}^{Q+\mathrm{SM}}\prod_{i_m\in I}u_{qm} \geq thr, \forall I \in \mathcal{B}d^{+}(\mathcal{F}'_{\mathcal{D}})\end{cases}$$

$$\text{where}\qquad thr = mfreq\cdot(N+Q+\mathrm{SM}) - \sup(I,\mathcal{D}_O)$$

Figure 8.1: The CSP formulation as an optimization process.

8.4.2 Adjusting the Size of the Extension

Equation (8.1) provides the absolute minimum number of transactions that need to be added in $\mathcal{D}_X$, to allow for the proper hiding of the sensitive itemsets of $\mathcal{D}_O$. However, this lower bound can, under certain circumstances, be insufficient to allow for the identification of an exact solution[1], even if one exists. To circumvent this problem, one needs to expand the size Q of $\mathcal{D}_X$ as determined by (8.1), by a certain number of transactions. A threshold, called *Safety Margin* (SM) is incorporated for this purpose. Safety margins can be either predefined or be computed dynamically, based on particular properties of database $\mathcal{D}_O$ and/or other parameters regarding the hiding process. In any case, the target of using a safety margin is to ensure that an adequate number of transactions participate in $\mathcal{D}_X$, thus an exact solution (if one exists) will not be lost due to the small size of the produced extension.

Since for each transaction in $\mathcal{D}_X$, M new binary variables are introduced that need to be tuned when solving the system of inequalities from $\mathcal{C}$, one would ideally want to identify a sufficiently large number of transactions for $\mathcal{D}_X$ (that allow for an exact solution), while this number be as low as possible to avoid unnecessary variables and constraints participating in the hiding process. Supposing that the value of Q is adjusted based on (8.1) and a sufficiently large safety margin is used, the methodology of Section 8.4.4 minimizes the size of $\mathcal{D}_X$ after the sanitization process to allow for an ideal solution.

[1]On the contrary, the lower bound is always sufficient to allow for an approximate solution of the set of inequalities produced by the itemsets in $\mathcal{C}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Replace All

$$\left(\sum_{T_q \in \mathcal{D}_X} \Psi_s \lesseqgtr thr, \Psi_s = \prod_{i_m \in T_q} u_{qm}\right)$$

With

$$\forall i \begin{cases} c_1: & \Psi_s \leq u_{q1} \\ c_2: & \Psi_s \leq u_{q2} \\ \vdots & \\ c_Z: & \Psi_s \leq u_{qm} \\ \Psi_s \geq u_{q1} + u_{q2} + \ldots + u_{qm} - |Z| + 1 \end{cases}$$

And

$$\sum_s \Psi_s \lesseqgtr thr$$

$$\text{where} \quad \Psi_s \in \{0, 1\}$$

Figure 8.2: The employed CDR approach.

8.4.3 Formulation and Solution of the CSP

The proposed CSP formulation is presented in Figure 8.1, while Figure 8.2 demonstrates the CDR approach that is applied. Assuming that $\mathcal{D}_X$ is large enough, and that $\mathcal{D}_O$ allows for an exact solution, the above hiding formulation is capable of identifying it. However, $\mathcal{D}_O$ may not always allow for an exact solution. Section 8.4.5 introduces an approach for dealing with suboptimality, while the following section examines the issue of validity in the transactions of $\mathcal{D}_X$ and presents an algorithm for removing unnecessary transactions.

8.4.4 Minimum Extension and Validity of Transactions

The incorporation of the safety margin threshold in the hiding process may lead to an unnecessary extension of $\mathcal{D}_X$. Fortunately, as is demonstrated here, it is possible to identify and remove the extra portion of $\mathcal{D}_X$ that is not needed, thus minimize the size of database $\mathcal{D}$ to the necessary limit. To achieve that, one needs to rely on the notion of null transactions, appearing in database $\mathcal{D}_X$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Definition 5. *(**Null transaction**) A transaction T_q is defined as **null** or **empty** if it does not support any valid itemset in the lattice. Null transactions do not support any pattern from $\mathcal{P}\backslash\{\varnothing\}$.*

Apart from the lack of providing any useful information, null transactions are easily identifiable, thus produce a privacy breach in the hiding methodology. They may exist due to two reasons: (i) an unnecessarily large safety margin, or (ii) a large value of Q essential for proper hiding. In the first case, these transactions need to be removed from $\mathcal{D}_X$, while in the second case the null transactions need to be validated, since Q denotes the lower bound in the number of transactions to ensure proper hiding.

The proposed methodology proceeds as follows. After solving the CSP of Figure 8.1, all the null transactions appearing in $\mathcal{D}_X$ are identified. Suppose that Q_{inv} such transactions exist. The size of database $\mathcal{D}_X$ will then equal the value of Q plus the safety margin SM. This means that the valid transactions in $\mathcal{D}_X$ will be equal to $v = Q + \mathrm{SM} - Q_{inv}$. To ensure minimum size of $\mathcal{D}_X$, the hiding algorithm keeps only k null transactions, such that:

$$k = \max(Q - v, 0) \Rightarrow k = \max(Q_{inv} - \mathrm{SM}, 0) \qquad (8.8)$$

As a second step, the hiding algorithm needs to ensure that the k empty transactions that remain in $\mathcal{D}_X$ become valid prior to releasing database $\mathcal{D}$ to public. A heuristic is applied for this purpose, which effectively replaces null transactions of $\mathcal{D}_X$ with transactions supporting itemsets of the revised positive border. After solving the CSP of Figure 8.1, the outcome is examined to identify null transactions. Then, Algorithm 8.1 is applied to replace the null transactions with valid ones, supporting itemsets of $\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$. Notice, that a round robin heuristic is applied based on the relative frequencies of the itemsets in the revised positive border. Algorithm 8.1 has a computational complexity of $O(|\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})| \cdot (|\mathcal{I}| + \log|\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})|))$, where $|\mathcal{I}|$ is the number of null transactions in $\mathcal{D}_X$ and $|\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})|$ is the size of the revised positive border. The heuristic approach for the validation of transactions is preferable in the general case, when $\mathrm{SM} > 0$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 8.1 Validation of Transactions in $\mathcal{D}_X$.

```
 1: procedure VALIDATE($\mathcal{D}_O$, $\mathcal{D}_X$, $\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$)
 2:     $\mathcal{J} \leftarrow \{T_q \in \mathcal{D}_X | \quad T_q$ is null$\}$
 3:     $min \leftarrow 1$
 4:     for each ($I \in \mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$) do
 5:         $ds(I) \leftarrow sup(I, \mathcal{D}) - sup(I, \mathcal{D}_O)$
 6:         if ($ds(I) < min$) then
 7:             $min \leftarrow ds(I)$
 8:     for ($i = 0$; $i < |\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})|$; $i$++) do
 9:         $ds(I) \leftarrow int(ds(I)/min)$
10:     $Sds \leftarrow reverse\_sort(ds)$
11:     $Sds(|\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})|) \leftarrow 1$
12:     $k \leftarrow |\mathcal{J}|$
13:     while ($k > 0$) do
14:         for ($i = 0$; $i < |\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})|$; $i$++) do
15:             for ($j = Sds(i)$; $j \geq Sds(i+1)$; $j$–) do
16:                 if ($k > 0$) then
17:                     $k \leftarrow k - 1$
18:                     $Sds(i) \leftarrow Sds(i) - 1$
19:                     REPLACE($\mathcal{J}(k)$, $T_{\{Sds_i\}}$)
20:                 else   $return$
```

$$\textbf{minimize}\left(\sum_{q\in[1,Q+\text{SM}],m\in[1,M]} u_{qm}\right)$$

$$\textit{subject to}\begin{cases}\sum_{q=1}^{Q+\text{SM}}\prod_{i_m\in I}u_{qm} < thr, \forall I \in \mathcal{B}d^-(\mathcal{F}'_{\mathcal{D}})\\[2ex] \sum_{q=1}^{Q+\text{SM}}\prod_{i_m\in I}u_{qm} \geq thr, \forall I \in \mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})\\[2ex] \forall T_q \in \mathcal{D}_X: \qquad \sum_{i_m\in I} u_{qm} \geq 1\end{cases}$$

$$\text{where} \qquad \text{thr} = mfreq\cdot(N+Q+\text{SM}) - \text{sup}(I,\mathcal{D}_O)$$

Figure 8.3: Expansion of the CSP to ensure validity of transactions.

A different approach is preferable when the minimization of distance between the $\mathcal{D}_O$ and $\mathcal{D}$ is of crucial importance and the requested safety margin is small or zero. In such cases, an exact solution consisting of valid transactions can be attained if, for each transaction in database $\mathcal{D}_X$, a constraint is added to the CSP to enforce its validity in the final solution. The new CSP formulation is depicted in Figure 8.3. An immediate disadvantage of this approach is the increment in the number of constraints of the produced CSP.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.3: The intermediate form of database $\mathcal{D}_X$.

a	b	c	d	e	f
u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}
u_{21}	u_{22}	u_{23}	u_{24}	u_{25}	u_{26}
u_{31}	u_{32}	u_{33}	u_{34}	u_{35}	u_{36}
u_{41}	u_{42}	u_{43}	u_{44}	u_{45}	u_{46}

8.4.5 Treatment of Suboptimality in Hiding Solutions

There are cases where exact hiding solutions do not exist and one must seek for a good approximate solution. To identify such a solution, the difference of importance existing between (8.3) and (8.4) is of crucial importance. Since the target of a hiding methodology is to secure sensitive knowledge, the holding of (8.4) is of major importance. The inherent difference in the significance of these two inequalities, along with the fact that solving the system of all inequalities of the form (8.4) always has a feasible solution, enables the relaxation of the problem when needed and the identification of good approximate solutions.

Algorithm 8.2 Relaxation Procedure in $V = \mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$.

```
1: procedure SELECTREMOVE(Constraints $\mathbf{C_R}$, $V$, $\mathcal{D}_O$)
2:     $C_{R_{\text{maxlen}}} \leftarrow \bigcup \text{argmax}_{\mathbf{i}}\{|R_i|\}$                ▷ $C_{R_i} \leftrightarrow V_i$
3:     $crmsup \leftarrow \mathbf{min}_{C_{R_{\text{maxlen}}},i}(\text{sup}(R_i \in V, \mathcal{D}_O))$
4:     for each $c \in C_{R_{\text{maxlen}}}$ do
5:         if $\text{sup}(R_i, \mathcal{D}_O) = crmsup$ then
6:             Remove (c)                                  ▷ remove constraint
```

Algorithm 8.2 applies a relaxation process that selectively removes inequalities of type (8.3) up to a point that the resulting CSP is solvable. In this algorithm, C_R is the set of constraints imposed by the itemsets of $\mathcal{B}d^+(\mathcal{F}'_{\mathcal{D}})$ and $|R_i|$ denotes the size of itemset R_i in C_R. In a realistic situation, only a small portion of constraints will have to be discarded. Thus, emphasis is given on constraints involving maximal size and minimum support itemsets, appearing in database $\mathcal{D}_O$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\mathcal{B}d^{+}(\mathcal{F}'_{\mathcal{D}}) \begin{cases} \{ab\} \ : u_{11}u_{12} + u_{21}u_{22} + u_{31}u_{32} + u_{41}u_{42} \geq 0.2 \\ \{acd\} : u_{11}u_{13}u_{14} + u_{21}u_{23}u_{24} + \\ \qquad + u_{31}u_{33}u_{34} + u_{41}u_{43}u_{44} \geq 1.2 \end{cases}$$

$$\mathcal{B}d^{-}(\mathcal{F}'_{\mathcal{D}}) \begin{cases} \{e\} \ : u_{15} + u_{25} + u_{35} + u_{45} < 0.2 \\ \{f\} \ : u_{16} + u_{26} + u_{36} + u_{46} < 2.2 \\ \{bc\} : u_{12}u_{13} + u_{22}u_{23} + u_{32}u_{33} + u_{42}u_{43} < 0.2 \\ \{bd\} : u_{12}u_{14} + u_{22}u_{24} + u_{32}u_{34} + u_{42}u_{44} < 2.2 \end{cases}$$

$$T_q \in \mathcal{D}_X \begin{cases} u_{11} + u_{12} + u_{13} + u_{14} + u_{15} + u_{16} \geq 1 \\ u_{21} + u_{22} + u_{23} + u_{24} + u_{25} + u_{26} \geq 1 \\ u_{31} + u_{32} + u_{33} + u_{34} + u_{35} + u_{36} \geq 1 \\ u_{41} + u_{42} + u_{43} + u_{44} + u_{45} + u_{46} \geq 1 \end{cases}$$

Figure 8.4: The constraints in the CSP of the running example.

8.4.6 A Running Example: The Hybrid Approach

Consider database $\mathcal{D}_O$ of Table 8.1. As part of Section 8.3.3 the original and the revised borders for the hiding of the sensitive itemsets in $\mathcal{S} = \{e, ae, bc\}$, when $mfreq = 0.3$, were computed. To proceed, one needs to identify the minimum extension of $\mathcal{D}_O$, namely the minimum size of $\mathcal{D}_X$ that facilitates sensitive knowledge hiding. Using (8.1) for the sensitive itemset with the highest support (here either $\{e\}$ or $\{bc\}$), we have that $Q = 4$. Excluding, for brevity, the use of a safety margin, $\mathcal{D}_O$ needs to be extended by 4 transactions. Table 8.3 depicts database $\mathcal{D}_X$ prior to the adjustment of the various u_{qm} variables involved. To produce the needed constraints, consider all the itemsets appearing in $\mathcal{C}$, where $\mathcal{C} = \mathcal{B}d^{+}(\mathcal{F}'_{\mathcal{D}}) \cup \mathcal{B}d^{-}(\mathcal{F}'_{\mathcal{D}}) = \{e, f, bc, bd, ab, acd\}$. Based on Figure 8.1 the CSP depicted in Figure 8.4 is constructed, where the validity of all transactions of $\mathcal{D}_X$ has to be ensured. Notice that if a safety margin was used, the last set of constraints (regarding the validity of transactions in $\mathcal{D}_X$) would have been excluded from the CSP to allow for the minimization of the size of $\mathcal{D}_X$ at a later point. After formulating the CSP the CDR approach of Figure 8.2 is applied. Solving the produced CSP using BIP creates the transactions of database $\mathcal{D}_X$ presented in the Table 8.1. The hiding solution of Table 8.1 is exact. Notice, that with a

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.4: Database $\mathcal{D}_X$ after the solution of the CSP.

a	**b**	**c**	**d**	**e**	**f**
1	1	1	1	1	0
0	1	0	0	1	0
1	0	1	1	1	0
0	**0**	**0**	**0**	**0**	**0**

minimal extension of 4 transactions, all the sensitive knowledge was hidden, leaving other patterns unaffected in $\mathcal{D}$. This solution is ideal with a distance of $dist(\mathcal{D}_O, \mathcal{D}) = 9$.

To highlight some other aspects of the proposed hiding methodology, consider the hiding of $\mathcal{S} = \{abc\}$ in the same database $\mathcal{D}_O$. In this case, using (8.1) we have that $Q = 1$. However, as it turns out, only one transaction is insufficient to provide an exact solution to the formulated CSP. Suppose that a safety margin of 3 was used, leading to the construction of 4 transactions in database $\mathcal{D}_X$. Due to the use of the safety margin in the CSP formulation, the set of constraints corresponding to the validity of the transactions in database $\mathcal{D}_X$ are excluded, thus, the formulation of Figure 8.1 is used instead. Table 8.4 shows database $\mathcal{D}_X$ after the solution of the CSP. As one can notice, the last transaction of $\mathcal{D}_X$ is null. Based on (8.8), $k = \max(1-3, 0)$ so the empty transaction from $\mathcal{D}_X$ can be safely removed, without jeopardizing the hiding of the sensitive patterns. The result database $\mathcal{D}_X$ contains the 3 valid transactions and pertains to an exact hiding solution.

As one can notice in the presented examples, the solver adjusts the u_{qm} variables in a way that all the inequalities pertaining to an exact solution are satisfied, while the number of zero entries in $\mathcal{D}_X$ is maximized. Consider for instance the database $\mathcal{D}_X$ of Table 8.1. To hide itemset $\{e\}$ this itemset must not be supported by any transaction in $\mathcal{D}_X$. Thus, all the respective variables become zero. Now, consider the 1–itemset $\{f\}$ of the revised negative border. Since no other itemset in $\mathcal{C}$ supports it, all its entries in $\mathcal{D}_X$ can be safely turned into zero. Finally, itemset $\{ab\}$ has to be supported by at least one transaction in $\mathcal{D}_X$ to remain frequent in $\mathcal{D}$. In any case, since the optimization criterion of the CSP requires the adjustment of the minimum number of u_{qm} variables to '1', the solver ensures that the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

sanitized database $\mathcal{D}$ lies as close as possible to $\mathcal{D}_O$, while providing an exact solution.

8.5 A Partitioning Approach to Improve Scalability

To achieve an exact hiding solution the hiding process needs to regulate the values of all items in every transaction of $\mathcal{D}_X$. This number of variables may, under certain circumstances, be very large leading to a large increase in the runtime of the BIP algorithm. Fortunately, the scalability of the hiding process can be improved without relaxing the requirement for an exact hiding solution. To achieve this, the hiding process is properly decomposed into two parts of approximately equal size, then each part is solved separately and the individual solutions are coupled together.

8.5.1 Partitioning Methodology

Let $I = I_P \cup I_{\bar{P}}$, be a decomposition of the universe of all items into two disjoint subsets, such that I_P contains all items that appear in the sensitive itemsets along with possibly other items from I, and $I_{\bar{P}}$ contains the rest of the items in I. First, the revised borders for database $\mathcal{D}$ are identified, as in Section 8.3.3, and set $\mathcal{C}$ is formulated (as in (8.7)). Each itemset in $\mathcal{C}$, is then uniquely assigned to one of the sets $I_P, I_{\bar{P}}$, based on the items it supports. For set I_P all the assigned itemsets from $\mathcal{C}$ must not support any items of $I_{\bar{P}}$. However, itemsets assigned to $I_{\bar{P}}$ may support items of I_P. The ideal case scenario is a partitioning of the items in I_P, $I_{\bar{P}}$ such that (i) the two partitions are of approximately equal size, and (ii) as few as possible itemsets assigned in $I_{\bar{P}}$ support items from I_P.

After formulating the two partitions and assigning the itemsets from $\mathcal{C}$, relation (8.1) and the safety margin are applied to decide on the size of $\mathcal{D}_X$. The goal is to properly construct $\mathcal{D}_X$ based on the principles presented earlier. Starting from set I_P, consider each transaction in $\mathcal{D}$ to be translated in this subspace; I_P regulates the assignment of the u_{qm} variables for all transactions $T_q \in \mathcal{D}_X$ and all items $m \in I_P$. Given the proper itemsets from $\mathcal{C}$ that support items in this subspace, one can adjust these binary variables in a way

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.5: Database $\mathcal{D}_X$ as the union of $\mathcal{D}_1$ and $\mathcal{D}_2$ produced by the partitioning approach.

	a	**b**	**c**	**d**	**e**	**f**
$\mathcal{D}_1$	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	1	1	0	0	0	0
$\mathcal{D}_2$	1	0	1	1	0	0
	1	0	1	1	0	0

that the revised border is preserved for the corresponding items. In the same manner the borderline in the subspace of $I_{\bar{P}}$ is preserved. Especially for the itemsets in $I_{\bar{P}}$ that also support items from I_P, by considering the previously assigned values of the u_{qm} variables in the subspace of I_P one can proceed to properly assign the u_{qm} variables in the subspace of $I_{\bar{P}}$, so that the newly formulated BIP has a solution. In the case that the current transactions in $\mathcal{D}_X$ are insufficient to accommodate for the holding of the new constraints imposed by the itemsets in $I_{\bar{P}}$, database $\mathcal{D}_X$ has to be appropriately expanded. However, this expansion can be negated at a later point with the removal of the null transactions (as in Section 8.4.4) after the overall hiding solution. Since the formulated CSP sets the minimum possible u_{qm} variables to '1', the borderline will be adjusted through this process without causing any harm to the exact solution, apart from potentially using a larger part of the extension. Due to the exponential execution time of the integer programming solver the benefit of this decomposition is substantial since it allows for reasonably low execution times, even when performing knowledge hiding in very large databases.

8.5.2 Running Example: The Partitioning Approach

Using the partitioning approach for the example of Table 8.1, we have that $I_P = \{a, b, c, e\}$ and $I_{\bar{P}} = I \backslash I_P$. The itemsets of $\mathcal{C} = \{e, f, ab, bc, bd, acd\}$ are partitioned as follows: $\mathcal{C} =$

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$\{e, ab, bc\} \cup \{f, bd, acd\}$. Solving the first CSP creates database $\mathcal{D}_1$ of Table 8.5, which allows the satisfaction of the constraint regarding itemset $\{acd\}$ participating in the second CSP. Thus, after the solution of the first CSP, database $\mathcal{D}_X$ has to be extended. Suppose that we use $SM = 2$. The second CSP leads to database $\mathcal{D}_2$, as shown in Table 8.5. Overall, the produced extension contains three null transactions. Using (8.8), $k = \max(3-2, 0) = 1$, which means that only one null transaction must be kept. Thus, after removing the two null transactions of $\mathcal{D}_1$, we proceed to validate the third one by using Algorithm 8.1. As one can notice, database $\mathcal{D}_X$ is exact but not ideal, since there exists no 1–itemset in $\mathcal{B}^+(\mathcal{F}'_{\mathcal{D}})$.

8.5.3 Alternative Approaches for Handling Large CSPs

The proposed partitioning methodology can efficiently handle a wide variety of problem instances by substantially reducing the required runtime for the solution of the CSP. However, there exist some cases when the decomposition of the CSP leads to parts that are still very large to be undertaken by a single processor. When such a situation occurs, one can result in distributed techniques for the solution of the CSP. In distributed approaches, the variables that participate in the CSP are distributed among a set of collaborative agents. Since the variables are connected by constraints, their distribution builds a network of constraints among the agents. The distributed algorithms operate on this network by applying a parallel search technique to identify the solution of the CSP. The interested reader is encouraged to refer to the work of Yokoo, et al. [182,183] for a review of distributed constraint satisfaction algorithms. Furthermore, a comparison of structural decomposition techniques for CSPs is presented in the work of Gottlob, et al. [53].

8.6 Experimental Evaluation

The proposed algorithm was tested on real world datasets using different parameters such as minimum support threshold and number/size of sensitive itemsets to hide. All these datasets are publicly available through the FIMI repository [80].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 8.6: The characteristics of the four datasets.

Dataset	**N**	**M**	**Avg tlen**
BMS–WebView–1	59,602	497	2.5
BMS–WebView–2	77,512	3,340	5.0
Mushroom	8,124	119	23.0
Chess	3,196	76	37.0

Datasets BMS–WebView–1 and BMS–WebView–2 contain click stream data from the Blue Martini Software, Inc. and were used for the KDD Cup 2000 [106]. The mushroom and the chess datasets were prepared by Roberto Bayardo (University of California, Irvine) from the UCI datasets and PUMSB [22]. All these datasets demonstrate varying characteristics in terms of the number of transactions and items and the average transaction lengths. Table 8.6 summarizes them.

The thresholds of minimum support were appropriately set to ensure an adequate amount of frequent itemsets. Several experiments were conducted for hiding up to 20 sensitive 10–itemsets. To avoid an unnecessary explosion in the number of constraints, two pruning techniques were enforced. The first technique removed tautologies, i.e. inequalities that always hold (such as $u_{qm} \geq 0$). The second one partitioned the inequalities, identified overlapping sets and kept the most specific inequality from each set (such as $u_{qm} \geq 1, u_{qm} + u_{qn} \geq 1 \rightarrow u_{qm} \geq 1$). Both pruning techniques can be easily applied and relieve the solver from a good portion of unnecessary work. The hybrid algorithm was implemented in Perl and C and the experiments were conducted on a PC running 64–bit Linux on an Intel Pentium D, 3.2 Ghz processor equipped with 4GB of RAM. All integer programs were solved using ILOG CPLEX 9.0 [95].

In all conducted experiments, the CSP was formulated based on Figure 8.3 and a safety margin of 10 transactions was used. The sensitive itemsets for sanitization were randomly selected among the frequent ones. CPLEX provides the option of pre–solving the BIP, a feature that becomes very useful since it allows the reduction of the BIP's size, the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

improvement of its numeric properties (for example, by removing some inactive constraints or by fixing some non–basic variables), and enables the early detection of infeasibility in the hiding solution.

Let notation $a \times b$ denote the hiding of a itemsets of length b. Figure 8.5 provides a comparison of the hiding solutions identified by the hybrid algorithm against three state-of-the-art border-based approaches: the Border Based Approach (BBA) of Sun and Yu [160, 161], the Max–Min 2 algorithm of Moustakides and Verykios [128] and the inline algorithm of Gkoulalas-Divanis and Verykios [64] (Chapter 6), in terms of side–effects introduced by the hiding process. As one can notice the hybrid algorithm consistently outperforms the three other schemes, with the inline approach being the second best. In most tested cases, the heuristic algorithms failed to identify a solution bearing minimum side–effects, while the inline approach demonstrated in several occasions that an exact solution could not be attained without extending the dataset. Figure 8.6 presents a comparison of the four algorithms at terms of runtime cost. As one can observe, the scalability of the heuristic approaches is better when compared to the exact approaches, since the runtime of the latter ones is primarily determined by the time required by the BIP solver to solve the CSP. However, as shown in Figure 8.5, the runtime cost of solving the CSPs of the exact methodologies is worthwhile, since the quality of the attained solutions is bound to be high.

Figure 8.7 presents the dependence that exists between the size of the extension and the sensitive itemsets. In this figure the hiding scenarios are separated into groups of four. In each group, the hiding of more itemsets of the same length includes all the sensitive itemsets that were selected for the previous hiding scenario. For instance, the 5 10–itemsets of the 10×10 hiding scenario, are the same as in the 5×10 hiding scenario. In the chess dataset, the $a \times 7$ itemsets are selected to have lower supports than their counterparts participating in the $a \times 10$ hiding scenarios. On the other hand, in the mushroom dataset, the group of $a \times 10$ itemsets reflects itemsets that lie near the border, whereas the $a \times 5$ itemsets are highly supported in the dataset. Figure 8.8 follows the layout of Figure 8.7 and presents the relation between the number of constraints in the CSP and the number and the size of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

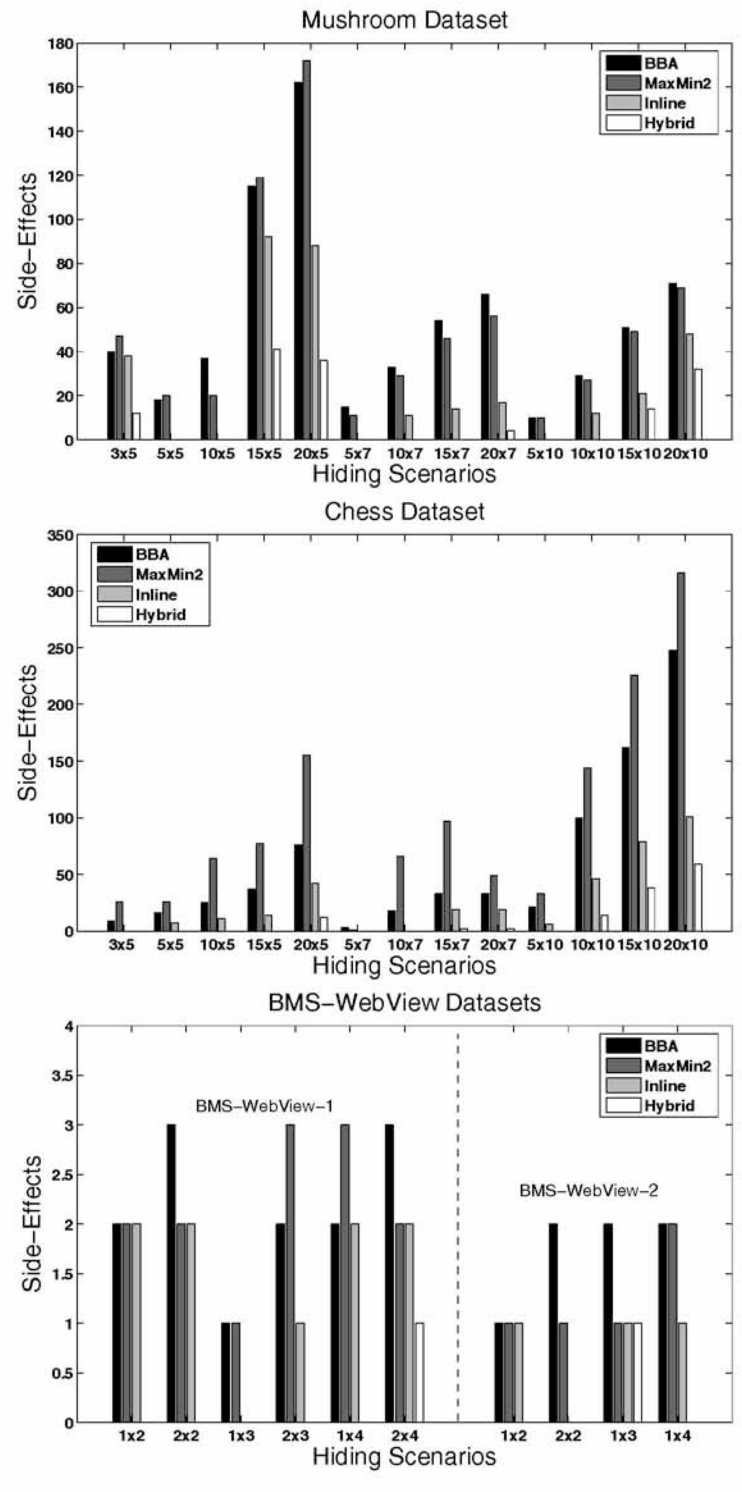


Figure 8.5: The quality of the produced hiding solutions.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

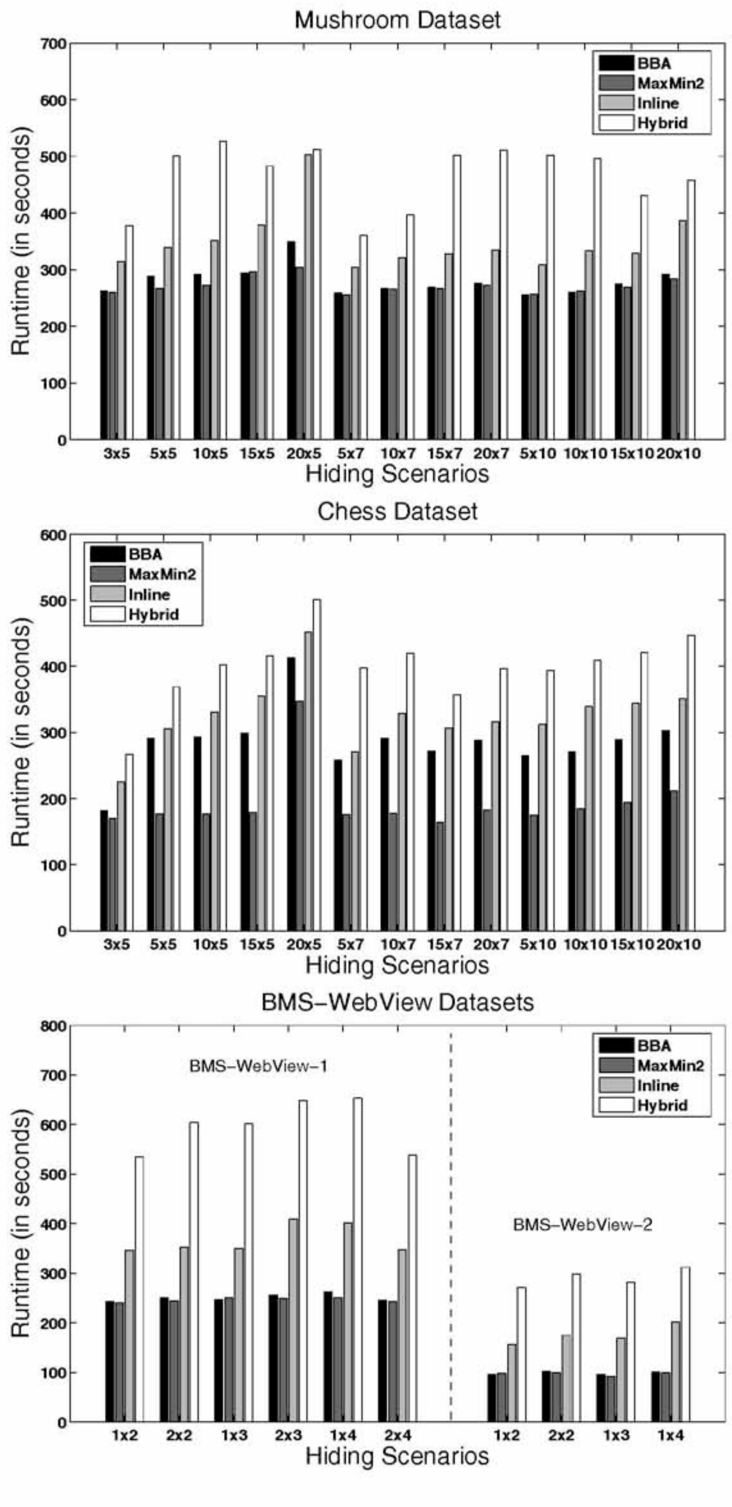


Figure 8.6: The scalability of the hybrid approach.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

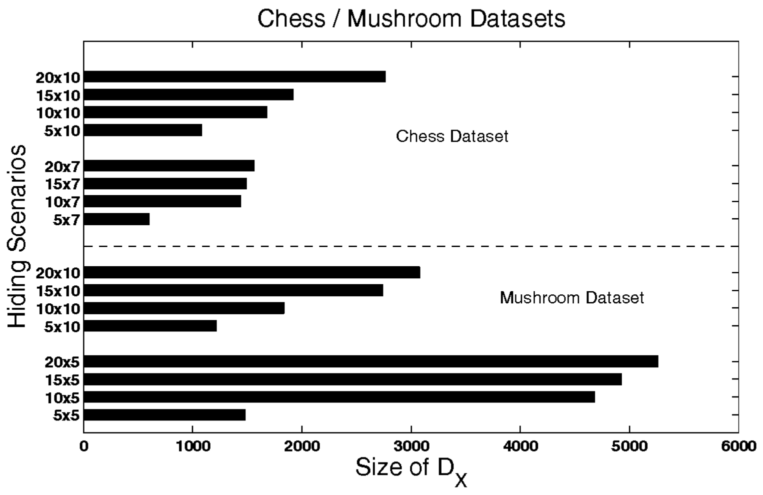


Figure 8.7: The size of extension $\mathcal{D}_\mathcal{X}$.

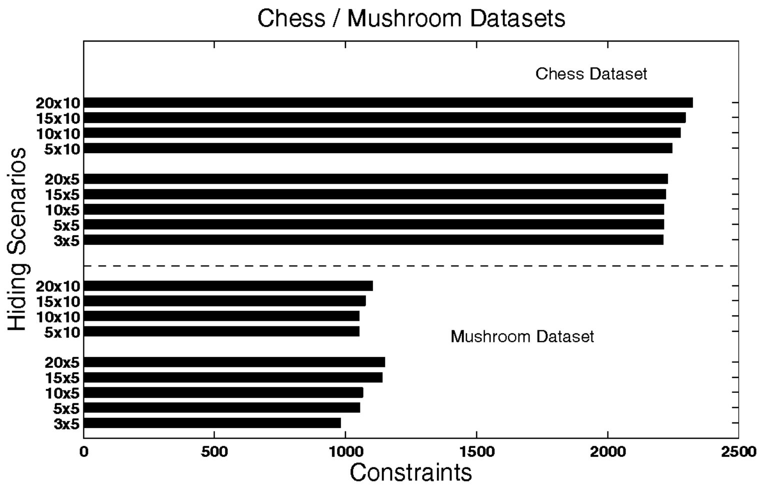


Figure 8.8: The number of constraints in the CSP.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

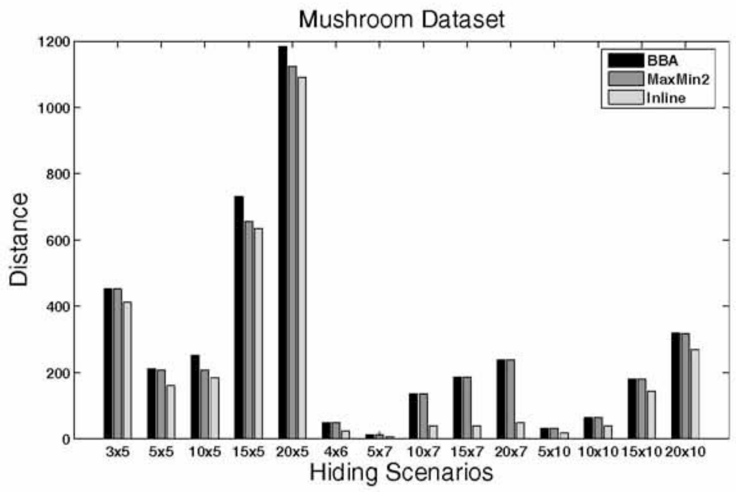


Figure 8.9: The distance of the three hiding schemes.

the sensitive itemsets. As is shown, the hiding of more itemsets of the same size leads to the production of more inequalities for the CSP, since in the typical case, the size of the negative border is augmented. A similar relation exists between the minimum support threshold of the dataset and the sensitive itemsets to be hidden. The lower the minimum support threshold of the dataset, the more itemsets become frequent. Supposing that we wish to hide the exact same itemsets, if we reduce the minimum support threshold then in the typical case more transactions are necessary in $\mathcal{D}_X$ to ensure that the revised positive border will be preserved in $\mathcal{D}$. Thus, more inequalities have to be included in the CSP to accomplish this goal. On the other hand, increment of the minimum support threshold typically leads to smaller problems and thus to a better performance of the hiding algorithm.

Figure 8.9 presents the distance (i.e. number of item changes) between the original and the sanitized database that is required by each algorithm to facilitate knowledge hiding. Since the two heuristic approaches and the inline algorithm operate in a similar fashion (i.e. by selecting transactions of the original database and excluding some items), it makes sense to compare them in terms of the produced distances. From the comparison it is evident that the inline approach achieves to minimize the number of item modifications, a result that can be attributed to the optimization criterion of the generated CSPs. On the contrary, the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

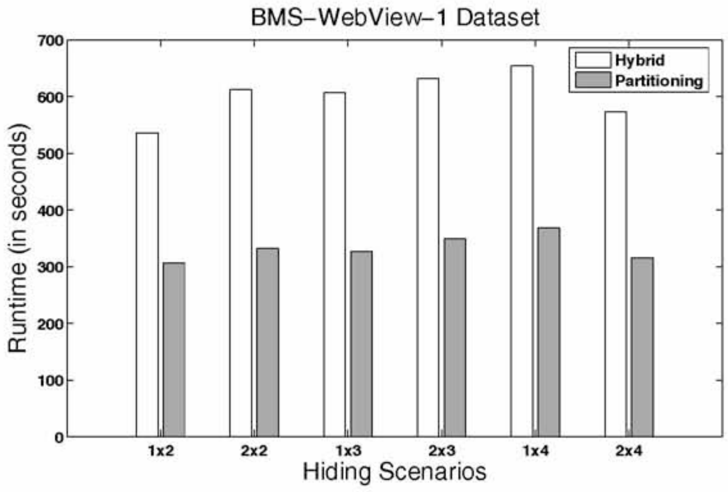


Figure 8.10: The performance of the partitioning approach.

hybrid algorithm does not alter the original dataset but instead uses a database extension to (i) leave unsupported the sensitive itemsets so as to be hidden in $\mathcal{D}$, and (ii) adequately support the itemsets of the revised positive border in order to remain frequent in $\mathcal{D}$. For this reason, we argue that the item changes (0s $\rightarrow$ 1s) that are introduced by the hybrid algorithm in $\mathcal{D}_X$ should not be attributed to the hiding task of the algorithm but rather to its power to preserve the revised positive border and thus eliminate the side–effects. This important difference between the hybrid algorithm and the other three approaches hardens their comparison in terms of item modifications. However, due to the common way that both the inline and the hybrid approaches model the CSPs, we argue that the property of minimum distortion of the original database is bound to hold for the hybrid algorithm.

Figure 8.10 presents the performance gain that is accomplished when using the partitioning approach instead of the hybrid algorithm. As one can notice, the split of the CSP into two parts has a significant benefit in the performance of the hiding algorithm.

An interesting insight from the conducted experiments is the fact that the hybrid approach, when compared to the inline algorithm [64] and the heuristic approaches of [128, 160], can better preserve the quality of the border and produce superior solutions. Indeed, the hybrid approach introduces the least amount of side–effects among the four

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

tested algorithms. On the other hand, the hybrid approach is less scalable than the inline approach and the two heuristics, due to the large number of the u_{qm} variables and the associated constraints of the produced CSP. Moreover, depending on the properties of the used dataset, there are cases where a substantial amount of transactions has to be added to the original dataset to facilitate knowledge hiding. This situation usually occurs when hiding itemsets with a very high support and/or when using a very low minimum support threshold. In most of these cases, approaches like the ones presented in Section 8.5 become essential to allow for a solution of the hiding problem. It is important however to mention that for as long as the given hiding problem decomposes to a CSP that remains computationally manageable and a sufficiently large safety margin is used, the hybrid approach is bound to identify a solution that will bear the least amount of side–effects to the original dataset. Thus, contrary to state–of–the–art approaches, the power of the proposed methodology is that it guarantees the least amount of side-effects to an extended set of hiding problems, when compared to the inline algorithm of [64].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 9

A Parallelization Framework for Exact Knowledge Hiding

In this chapter, we introduce a novel framework that is suitable for decomposition and parallelization of the exact hiding algorithms that we presented in Chapters 6, 7 and 8. The proposed framework (introduced in [72]) decomposes the CSP that is produced by the exact hiding algorithm, into a set of smaller CSPs that can be solved in parallel. First, the original CSP is structurally decomposed into a set of independent CSPs, each of which is assigned to a processor. Second, each independent CSP can be further decomposed into a set of dependent CSPs. In each step of the framework, a function is applied to question the gain of any further decomposition and allow the algorithm to take the appropriate action. Furthermore, the solutions of the various CSPs, produced as part of the decomposition process, can be appropriately combined to provide the solution of the initial CSP (prior to the decomposition). The generality of the proposed framework allows it to efficiently handle any CSP that consists of linear constraints involving binary variables and whose objective is to maximize (or minimize) the summation of these variables. Together with existing approaches for the parallel mining of association rules, as in [13, 86, 186], our framework can be applied to parallelize the most time consuming steps of the exact hiding algorithms. Through experiments, we demonstrate the effectiveness of our solution on

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

handling a large variety of hiding problem instances.

The remainder of this chapter is structured as follows: In Section 9.1 we present our proposed framework for decomposition and parallelization of exact hiding algorithms. Then, Section 9.2 contains the experimental evaluation and demonstrates the benefit of using this framework towards speeding up the hiding process.

9.1 The Parallelization Framework

Performing knowledge hiding by using the inline [64], the two–phase iterative [70] or the hybrid approach [68] allows for the identification of exact hiding solutions, whenever such solutions exist. However, the cost of identifying an exact hiding solution is high due to the solving of the involved CSP.

In this section, we propose a framework for decomposition and parallel solving that can be applied as part of the sanitization process of exact hiding algorithms. Our proposed framework operates in three phases: (i) the *structural decomposition* phase, (ii) the *decomposition of large individual components* phase, and (iii) the *parallel solving* of the produced CSPs. In what follows, we present in detail each phase of the framework.

9.1.1 Phase I: Structural Decomposition of the CSP

The number of constraints in a CSP can be very large depending on the database properties, the minimum support/frequency threshold that is used, and the number of sensitive itemsets to be hidden. Moreover, the fact that various initial constraints may incorporate products of u_{nm} variables, thus have a need to be replaced by numerous linear inequalities (using the CDR approach), makes the whole BIP problem tougher to solve. There is, however, a nice property in the CSPs that we can use to our benefit. That is, decomposition.

Based on the divide and conquer paradigm, a decomposition approach allows us to divide a large problem into numerous smaller ones, solve these new subproblems independently, and combine the partial solutions to attain the exact same overall solution. The

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

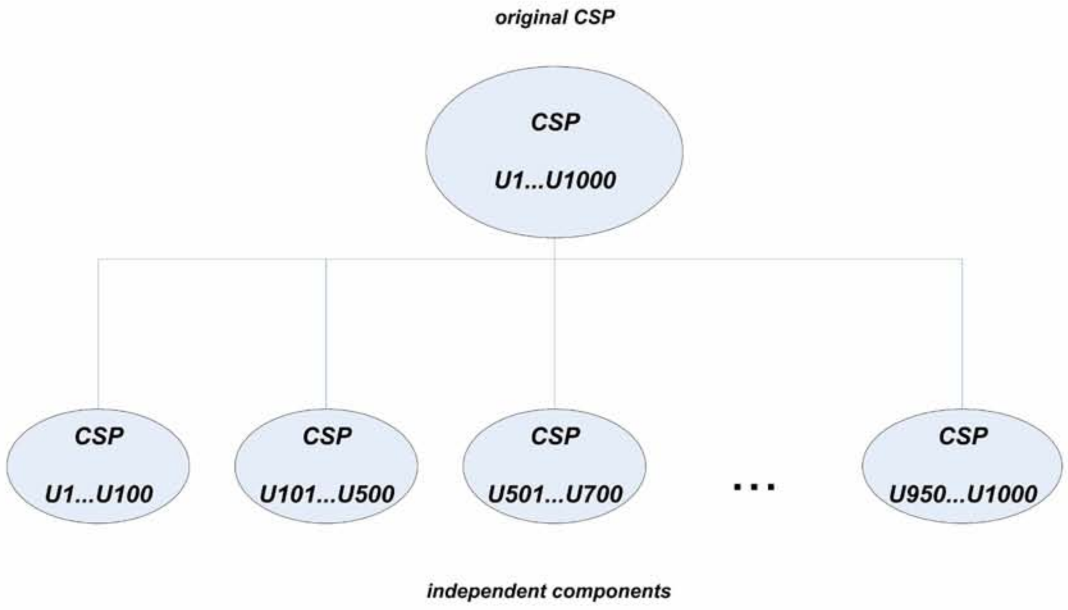


Figure 9.1: Decomposing large CSPs to smaller ones.

property of the CSPs which allows us to consider such a strategy lies behind the optimization criterion that is used. Indeed, one can easily notice that the criterion of maximizing (or equivalently minimizing) the summation of the binary u_{nm} variables is satisfied when as many u_{nm} variables as possible are set to one (equivalently to zero). This, can be established independently, provided that the constraints that participate in the CSP allow for an appropriate decomposition. The approach we follow for the initial decomposition of the CSP is similar to the decomposition structure identification algorithm presented in [122], although applied in a "constraints" rather than a "transactions" level. As demonstrated in Figure 9.1, the output of structural decomposition, when applied on the original CSP, is a set of smaller CSPs that can be solved independently. An example will allow us to better demonstrate how this process works.

Consider database $\mathcal{D}_O$ presented in Table 9.1. By performing frequent itemset mining in $\mathcal{D}_O$ using frequency threshold $mfreq = 0.3$, we compute the following set of frequent itemsets: $\mathcal{F}_{\mathcal{D}_O} = \{A, B, C, D, AB, CD\}$. Suppose that we want to hide the sensitive itemsets

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 9.1: The original database $\mathcal{D}_O$.

A	B	C	D
1	1	0	0
1	1	0	0
1	0	0	0
1	1	0	0
0	1	0	1
1	0	1	1
0	0	1	1
0	0	1	1
1	0	0	0
0	0	0	1

Table 9.2: The intermediate form of database $\mathcal{D}_O$.

A	B	C	D
1	u_{12}	0	0
1	u_{22}	0	0
1	0	0	0
1	u_{42}	0	0
0	u_{52}	0	1
1	0	u_{63}	u_{64}
0	0	u_{73}	u_{74}
0	0	u_{83}	u_{84}
1	0	0	0
0	0	0	1

in $\mathcal{S} = \{B, CD\}$, using for instance the inline approach[1]. Then, we have that:

$$\mathcal{S}_{max} = \{B, AB, CD\} \tag{9.1}$$

$$\mathcal{B}\mathbf{d}^{+}(\mathcal{F}'_{\mathcal{D}}) = \{A, C, D\} \tag{9.2}$$

$$V = \{C, D\} \subset \mathcal{B}\mathbf{d}^{+}(\mathcal{F}'_{\mathcal{D}}) \tag{9.3}$$

The intermediate form of this database is shown in Table 9.2 and the CSP formulation based on the inline algorithm is presented in Figure 9.2. Table 9.3 highlights the various

[1]We need to mention that it is of no importance which methodology will be used to produce the CSP, apart from the obvious fact that some methodologies may produce CSPs that are better decomposable than those constructed by other approaches. However, the structure of the CSP also depends on the problem instance and thus it is difficult to know in advance which algorithm is bound to produce a better decomposable CSP.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{maximize} \; (u_{12}+u_{22}+u_{42}+u_{52}+u_{63}+ u_{64}+u_{73}+u_{74}+u_{83}+u_{84})$$

$$\textit{subject to} \begin{cases} u_{12}+u_{22}+u_{42}+u_{52} < 3 \\ u_{63}u_{64}+u_{73}u_{74}+u_{83}u_{84} < 3 \\ u_{63}+u_{73}+u_{83} \geq 3 \\ u_{64}+u_{74}+u_{84} \geq 1 \end{cases}$$

Figure 9.2: The CSP formulation for the presented example.

Table 9.3: The constraints matrix for the CSP.

	c_1	c_2	c_3	c_4
u_{12}	X			
u_{22}	X			
u_{42}	X			
u_{52}	X			
u_{63}		Y	Y	
u_{64}		Y		Y
u_{73}		Y	Y	
u_{74}		Y		Y
u_{83}		Y	Y	
u_{84}		Y		Y

constraints c_r along with the variables that they control. As we can observe, we can cluster the various constraints into disjoint sets based on the variables that they involve. In our example, we can identify two such clusters of constraints, namely $\mathbf{M_1} = \{c_1\}$, and $\mathbf{M_2} = \{c_2, c_3, c_4\}$. Notice that none of the variables in each cluster of constraints is contained in any other cluster. Thus, instead of solving the entire problem of Figure 9.2, we can solve the two subproblems shown in Figure 9.3, yielding, when combined, the same solution as the one of the initial *CSP*: $u_{12} = u_{22} = u_{42} = u_{63} = u_{64} = u_{73} = u_{74} = u_{83} = 1$ and $u_{52} = u_{84} = 0$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{maximize}(u_{12}+u_{22}+u_{42}+u_{52})$$

$$\textit{subject to}\quad u_{12}+u_{22}+u_{42}+u_{52}<3$$

$$\text{where } \{u_{12},u_{22},u_{42},u_{52}\}\in\{0,1\}$$

and

$$\textbf{maximize}(u_{63}+u_{64}+u_{73}+u_{74}+u_{83}+u_{84})$$

$$\textit{subject to}\begin{cases} u_{63}u_{64}+u_{73}u_{74}+u_{83}u_{84}<3 \\ u_{63}+u_{73}+u_{83}\geq 3 \\ u_{64}+u_{74}+u_{84}\geq 1 \end{cases}$$

$$\text{where } \{u_{63},u_{64},u_{73},u_{74},u_{83},u_{84}\}\in\{0,1\}$$

Figure 9.3: The equivalent CSPs for the provided example.

9.1.2 Phase II: Decomposition of Large Independent Components

The structural decomposition of the original CSP allows one to divide the original large problem into a number of smaller subproblems which can be solved independently, thus highly reduce the runtime needed to attain the overall solution. However, as it can be noticed, both (i) the number of subproblems, and (ii) the size of each subproblem, are totally dependent on the underlying CSP and the structure of the constraints matrix. This fact means that there exist problem instances which are not decomposable and other instances which experience a notable imbalance in the size of the produced components. Thus, in what follows, we present two methodologies which allow us to decompose large individual components that are nonseparable through the structural decomposition approach. In both methods, our goal is to minimize the number of variables that are shared among the newly produced components, which are now *dependent*. What allows us to proceed in such a decomposition is the binary nature of the variables involved in the CSPs, a fact that we can use to our benefit to minimize the different problem instances that need to be solved to produce the overall solution of the initial problem.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Method I: Decomposition Using Articulation Points

To further decompose an independent component we need to identify the least amount of u_{nm} variables which, when discarded from the various inequalities of this CSP, produce a CSP that is structurally decomposable. To find these u_{nm} variables we proceed as follows. First, we create an undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ in which each vertex $v \in \mathcal{V}$ corresponds to a u_{nm} variable, and each edge $e \in \mathcal{E}$ connects vertexes that participate to the same constraint. Graph $\mathcal{G}$ can be built in linear time and provides us with an easy way to model the network of constraints and involved variables in our input CSP. Since we assume that our input CSP is not structurally decomposable, graph $\mathcal{G}$ will be connected.

After creating the constraints graph $\mathcal{G}$, we identify all its articulation points (a.k.a. cut–vertexes). The rationale here is that the removal of a cut–vertex will disconnect graph $\mathcal{G}$ and the best cut–vertex u_{nm} will be the one that leads to the largest number of connected components in $\mathcal{G}$. Each of these components will then itself constitute a new subproblem to be solved independently from the others. To identify the best articulation point we proceed as follows. As is already known, a fast way to compute the articulation points of a graph is to traverse it by using DFS. By adding a counter to a table of vertexes each time we visit a node, we can easily keep track of the number of components that were identified so far. In the end of the algorithm, along with the identified articulation points we can have knowledge of the number of components that each of these articulation points decomposes the initial graph. This operation can proceed in linear time O($\mathcal{V}+\mathcal{E}$).

After identifying the best articulation point, our next step is to remove the corresponding u_{nm} variable from graph $\mathcal{G}$. Then, each component of the resulting graph corresponds to a new subproblem (i.e. is a new CSP) that can be derived in linear time and be solved independently. To provide the same solution as the original CSP, the solutions of the various created subproblems need to be cross–examined, a process that takes place in Phase III and is further explained in Section 9.1.3.

A final step to be addressed involves the situation in which no single cut–vertex can be identified in the graph. If such a case appears, we choose to proceed heuristically

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

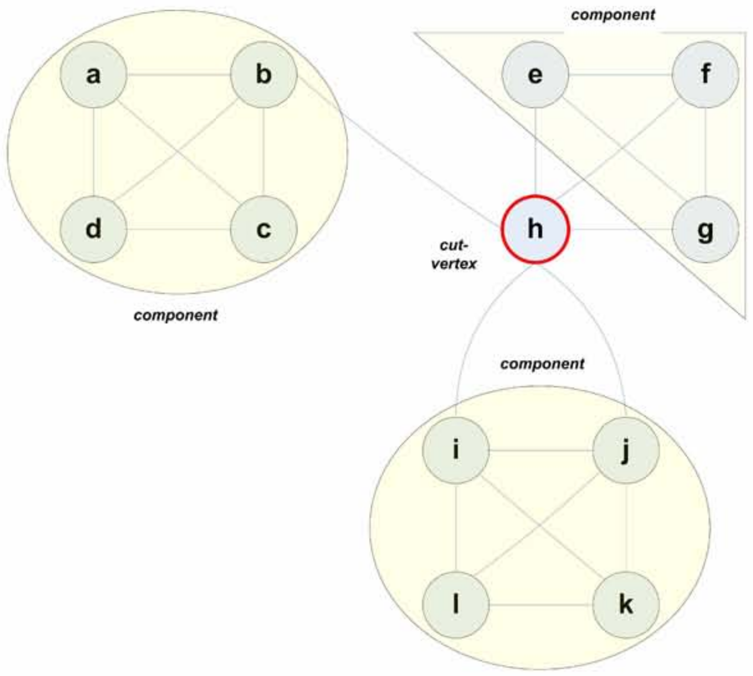


Figure 9.4: An example of decomposition using articulation points.

in order to experience low runtime of the algorithm. Our empirical approach is based on the premises that nodes having high degrees in graph $\mathcal{G}$ are more likely than others to correspond to cut–vertexes. For this reason, we choose to compute the degree of each vertex $u \in \mathcal{V}$ in graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ and identify the one having the maximum degree. Let $v = max_{u \in \mathcal{V}}(degree(u))$ be the vertex whose degree is the maximum among all other vertexes in the graph. Then, among all neighbors of v we identify the one having the maximum degree and proceed to remove both vertexes from the graph. As a final step we use DFS to traverse the resultant graph to examine if it is disconnected. The runtime of this approach is linear in the number of vertexes and edges of graph $\mathcal{G}$. If the resulting graph remains connected, we choose to leave the original CSP as–is and make no further attempt to decompose it. Figure 9.4 demonstrates an example of decomposition using articulation points. In this graph, we denote as "cut–vertex", the vertex which, when removed, leads to a disconnected graph having the maximum number of connected components (here 3).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

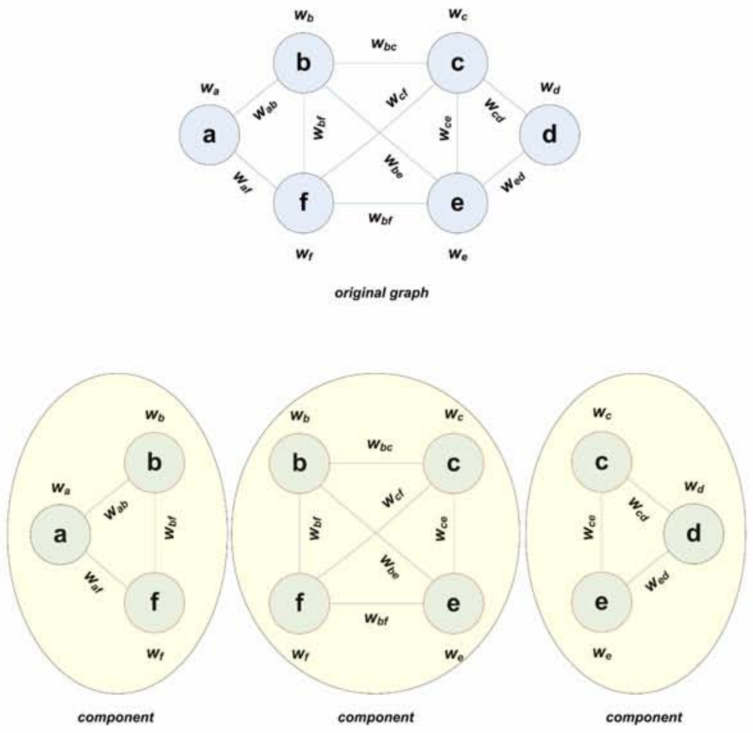


Figure 9.5: A three–way decomposition using weighted graph partitioning.

Method II: Decomposition Using Weighted Graph Partitioning

One of the primary disadvantages of decomposition using articulation points is the fact that we have limited control over (i) the number of components in which our initial CSP will eventually split, and (ii) the size of each of these components. This fact may lead to a low CPUs utilization in a parallel solving environment. For this reason, we present an alternative decomposition strategy which can break the original problem into as many subproblems as we can concurrently solve, based on our underlying system architecture. The problem formulation is once more tightly related to the graph modeling paradigm but instead of using articulation points, we rely on modern graph partitioning algorithms which can provide us with the optimal split.

By assigning each u_{nm} variable of the initial CSP to a vertex in our undirected graph, and each constraint c to a number of edges e_c formulating a clique in the graph

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

(while denoting the dependence of the u_{nm} variables involved), we proceed to construct a weighted version of graph $\mathcal{G}$ (presented in the previous section). This weighted graph, hereon denoted as $\mathcal{G}^W$, has two types of weights: one associated with each vertex $u \in \mathcal{V}^W$, and one associated with each edge $e \in \mathcal{E}^W$. The weight of each vertex corresponds to the number of constraints in which it participates in the CSP formulation. On the other hand, the weight of each edge in the graph denotes the number of constraints in which the two vertexes (it connects) appear together. Using a weighted graph partitioning algorithm, such as the one provided by METIS [101], one can decompose the graph into as many parts as the number of available processors that can be used to concurrently solve them. The rationale behind the applied weighting scheme is to ensure that the connectivity between vertexes belonging in different parts is minimal. Figure 9.5 demonstrates a three–way decomposition of the original CSP, using weighted graph partitioning.

9.1.3 Phase III: Parallel Solving of the Produced CSPs

Breaking a dependent CSP into a number of components (using one of the strategies mentioned earlier) is a procedure that should incur only if the CSP's size is large enough to worth the cost of the decomposition. For this reason, it is necessary to define a function $\mathcal{F}_\mathcal{S}$ to calculate the size of a CSP and a threshold above which the CSP should be decomposed. We choose function $\mathcal{F}_\mathcal{S}$ to be a weighted sum of the number of u_{nm} variables involved in the CSP and the associated constraints $\mathcal{C}$. The weights are problem–dependent. Thus,

$$\mathcal{F}_\mathcal{S} = w_1 \cdot |u_{nm}| + w_2 \cdot |\mathcal{C}| \tag{9.4}$$

Our problem solving strategy proceeds as follows. First, we apply structural decomposition on the original CSP and we distribute each component to an available processor. These components can be solved independently of each other. The final solution (i.e. the value of the objective for the original CSP) will equal the sum of the values of the individual objectives; thus, the master node that holds the original CSP should wait to accumulate the solutions returned by the servicing nodes.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Each node in our system is allowed to choose other nodes and assign them some computation. Whenever a node assigns job to other nodes, it waits to receive the results of the processing and then enforces the needed function to create the overall outcome (as if it had solved the entire job itself). After receiving an independent component, each processor applies the function $\mathcal{F}_S$ in its assigned CSP and decides whether it is essential to proceed with further decomposition. If this is the case, then it proceeds to its decomposition using one of the two schemes presented earlier (i.e. decomposition using articulation points or weighted graph partitioning) and assigns the newly created CSPs, each to an available processor. A mechanism that keeps track of the jobs distribution to processors and their status (i.e. idle vs occupied) is applied to allow for the best possible CPUs utilization. The same procedure continues until all constructed CSPs are below the user–defined size threshold and therefore do not need to be further decomposed.

At any given time, the processors contain a number of independent components and a number of mutually dependent components. In the case of the independent components, as mentioned earlier, the value of the objective function for the overall CSP is attained by summing up the values of the individual objectives. However, in the case of dependent CSPs the situation is more complex. To handle such circumstances, let *border* u_{nm} be a variable that appears in two or more dependent CSPs. This means that this variable was either the best articulation point selected by the first strategy, or a vertex that was at the boundary of two different components, identified using the graph partitioning algorithm. Border variables need to be checked for all possible values they can attain in order to provide us with the exact same solution as the one of solving the original CSP. Let at a given problem instance decomposition exist p such variables. Then, we have 2^p possible value assignments. For each possible assignment we solve the corresponding produced CSPs, in which the objective functions apart from the u_{nm} variables for the non–border cases they also contain (in the objectives and the constraints) the values of the currently tested assignment for the p variables. After solving the CSPs for each assignment we proceed to sum up the resulting objective values. As one can observe, the final solution will correspond to the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

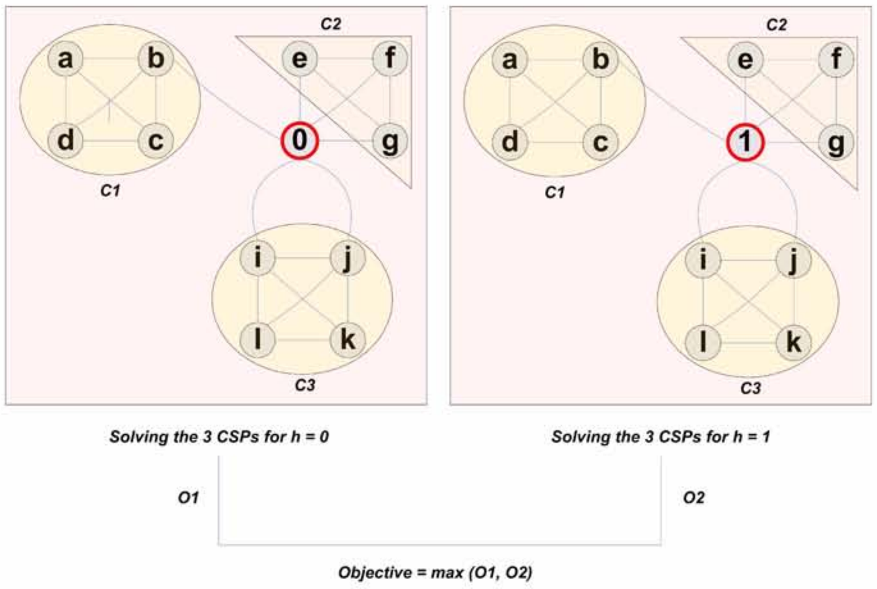


Figure 9.6: An example of parallel solving after decomposition.

maximum value among the different summations produced by the possible assignments.

To make matters clearer, assume that at some point time a processor receives a CSP that needs to be solved and finds that its size is greater than the minimum size threshold; thus the CSP has to be decomposed. Suppose that applying the first strategy yields a decomposition involving two border variables u_1 and u_2 and leads to two components: C_A and C_B. Each component is then assigned to a processor to solve it; all possible assignments should be tested. Thus, we have four different problem instances, namely: $C_{00}, C_{01}, C_{10}, C_{11}$, where C_{xy} means that the problem instance where $u_1 = x$ and $u_2 = y$ is solved; the rest variables' assignments remain unknown. Given two processors, the first one needs to solve these 4 instances for C_A whereas the second one needs to solve them for C_B. Depending on the load balancing strategy, each processor is capable of sending part of the computation (i.e. some instances) to other processors to fine–grain the whole process. Now suppose that the objective values for $C_{A,00}$ and $C_{B,00}$ were found. The objective value for problem instance C_{00} will then be the summation of these two objectives. To calculate the overall objective value and identify the solution of the initial CSP we need to identify the maximum among

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the objective values of all problem instances. An example of parallel solving after the application of a decomposition strategy is depicted in Figure 9.6, where one can notice that the solution of the initial CSP is provided by examining, for all involved CSPs, the two potential values of the selected cut–vertex h (i.e. solving each CSP for $h = 0$ and $h = 1$). The overall objective corresponds to the maximum of the two objectives, an argument that is justified by the binary nature of variable h.

9.2 Computational Experiments and Results

In this section, we provide the results of a set of experiments that we conducted to test our proposed framework. In what follows, we present the datasets we used and the different parameters involved in the testing process, and we provide experimental results involving the structural decomposition process, where we demonstrate the major gain that is achieved in the runtime of the employed knowledge hiding algorithm.

To test our framework, we used the inline approach to hide the sensitive knowledge in three real datasets. All these datasets are publicly available through the FIMI repository [80] and are summarized in Table 6.4. In all tested settings, the thresholds of minimum support were properly selected to ensure an adequate amount of frequent itemsets and the sensitive itemsets to be hidden were selected randomly among the frequent ones. We conducted several experiments trying to hide sensitive 2–itemsets, 3–itemsets, and 4–itemsets. Our source code was implemented in Perl and C and we conducted all our experiments on a PC running Linux on an Intel Pentium D, 3.2 Ghz processor equipped with 4 GB of main memory. All integer programs were solved using ILOG CPLEX 9.0 [95].

To conduct the experiments, we assume that we have all the necessary resources to proceed to a full–scale parallelization of the initial CSP. This means that if our original CSP can potentially break into P independent parts, then we assume the existence of P available processors that can run independently, each one solving one resultant CSP. Thus, the overall runtime of the hiding algorithm will equal the summation of (i) the runtime of the serial algorithm that produced the original CSP, (ii) the runtime of the *Structure Identification*

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

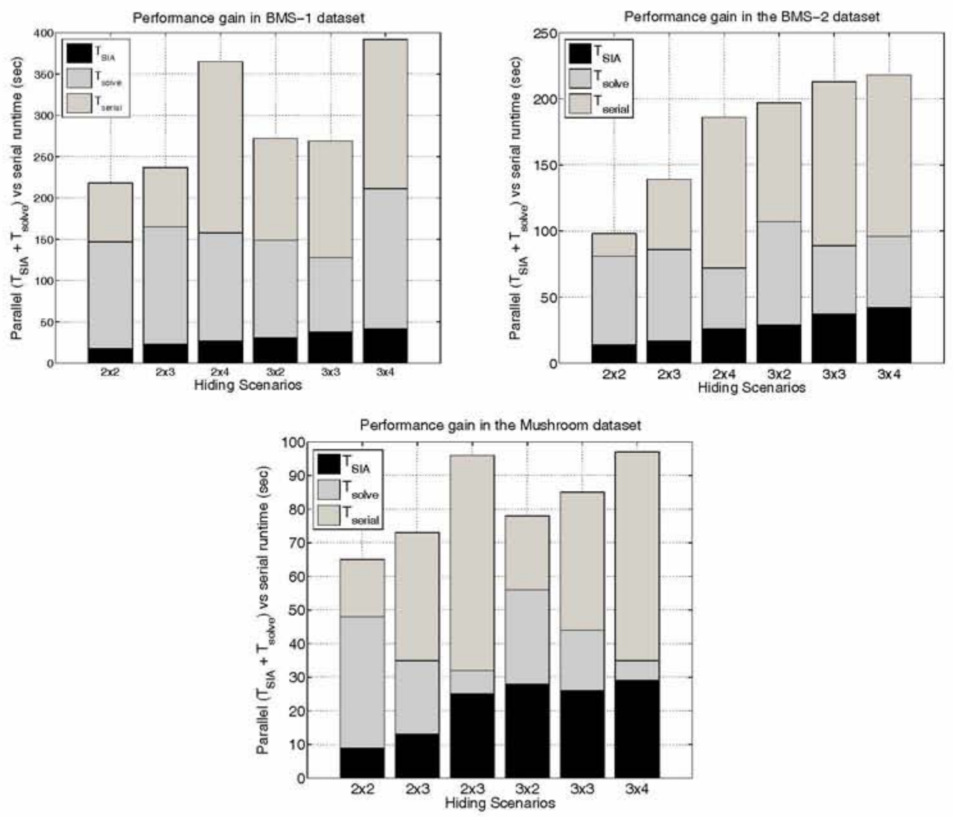


Figure 9.7: Performance gain through parallel solving, when omitting the V part of the CSP.

Algorithm (SIA) that decomposed the original CSP into numerous independent parts, (iii) the time that is needed to communicate each of the resulting CSPs to an available processor, (iv) the time needed to solve the largest of these CSPs, (v) the communication time needed to return the attained solutions to the original processor (hereon called "master") that held the whole problem, and finally (vi) the time needed by the master processor to calculate the summation of the objective values returned in order to compute the overall solution of the problem. That is:

$$T_{\text{overall}} = T_{\text{HA}} + T_{\text{SIA}} + T_{\text{spread}} + T_{\text{solve}} + T_{\text{gather}} + T_{\text{aggregate}} \qquad (9.5)$$

In the following experiments, we capture the runtime of (ii) and (iv), namely T_{SIA} and T_{solve}, since we consider both the communication overhead ($T_{\text{spread}} + T_{\text{gather}}$) and the overall solution calculation overhead ($T_{\text{aggregate}}$) to be negligible when compared to these

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

runtimes. Moreover, the runtime of (i) does not change in the case of parallelization and therefore its measurement in these experiments is of no importance. To allow us compute the benefit of parallelization, we include in the results the runtime T_{serial} of solving the entire CSP without prior decomposition.

In our first set of experiments (presented in Figure 9.7), we ensure the breaking of the original CSP into a controllable number of components by excluding all the constraints involving itemsets from set V (see Figure 6.1). Thus, to break the original CSP into P parts, one needs to identify P mutually exclusive (in the universe of items) itemsets to hide. However, based on the number of supporting transactions for each of these itemsets in $\mathcal{D}_O$, the size of each produced component may vary significantly. As one can observe in Figure 9.7, the time that was needed for the execution of the SIA algorithm and the identification of the independent components is low when compared to the time needed for solving the largest of the resulting CPSs. Moreover, by comparing the time needed for the serial and the one needed for the parallel solving of the CSP, one can notice how beneficial is the decomposition strategy in reducing the runtime that is required by the hiding algorithm. For example, in the 2×2 hiding scenario for BMS–1, serial solving of the CSP requires 218 seconds, while parallel solving requires 165 seconds. This means that by solving the CSP in parallel using two processors, we reduce the solution time by 53 seconds.

In our second set of experiments, presented in Figure 9.8, we included the V part of the CSP, as produced by the inline algorithm. As one can observe, there are certain situations (e.g., 3×3 in BMS–WebView–1 or 3×2 in BMS–WebView–2) in which the original CSP cannot be decomposed (i.e. $T_{\text{solve}} = 0$). In all such cases, one has to apply either the decomposition approach using articulation points or the weighed graph partitioning algorithm, in order to parallelize the hiding process.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

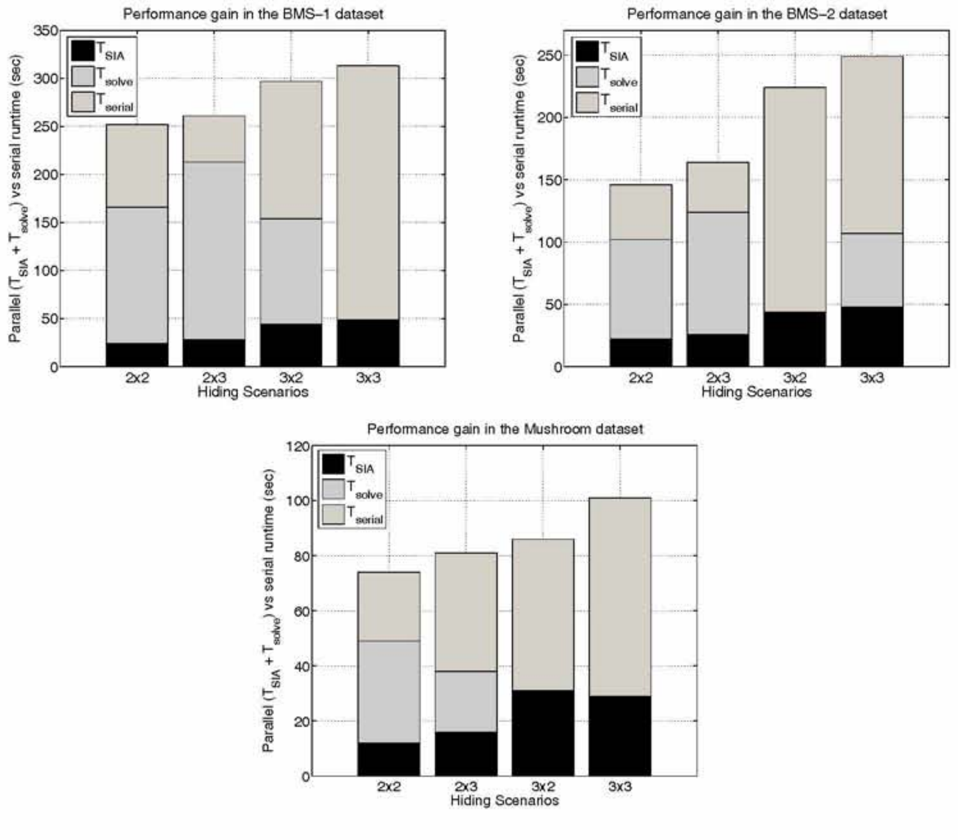


Figure 9.8: Performance gain through parallel solving of the entire CSP.

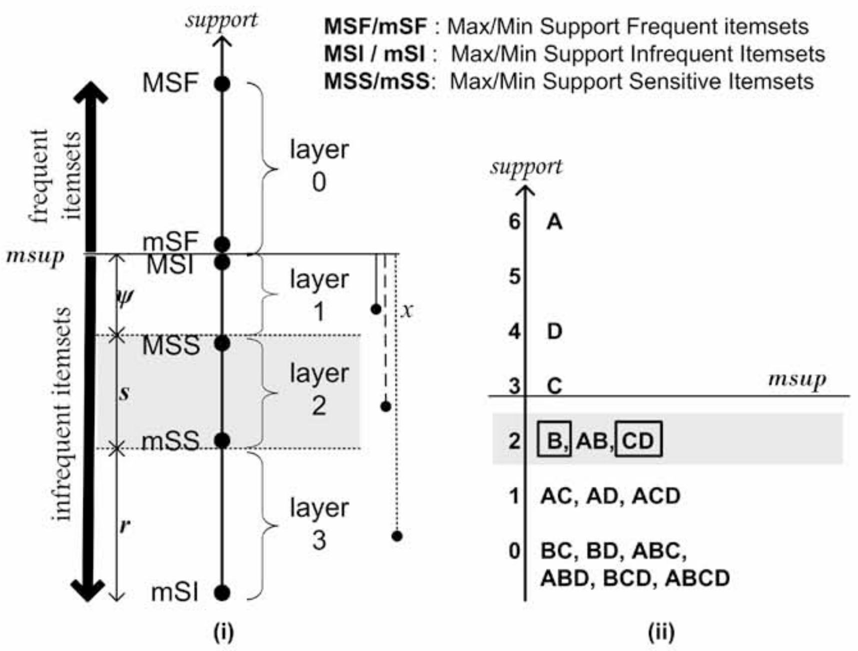
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 10

Quantifying the Privacy of Exact Hiding Algorithms

The exact hiding approaches that were presented in Chapters 6, 7, and 8 are all based on the principle of minimum harm (distortion), which requires the minimum amount of modifications to be made to the original database to facilitate sensitive knowledge hiding. As an effect, in most cases (depending on the problem at hand), the sensitive itemsets are expected to be positioned just below the revised borderline in the computed sanitized database $\mathcal{D}$. However, the selection of the minimum support threshold based on which the hiding is performed can lead to radically different solutions, some of which are bound to be superior to others in terms of offered privacy. In what follows, we present a layered approach that enables the owner of the data to quantify the privacy that is offered on a given database by the employed exact hiding algorithm. Assuming that an adversary has no knowledge regarding which of the infrequent itemsets in $\mathcal{D}$ are the sensitive ones, this approach can compute the disclosure probability for the hidden sensitive knowledge, given the sanitized database $\mathcal{D}$ and a minimum support/frequency threshold *msup*/*mfreq*.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



MSF/mSF : Max/Min Support Frequent itemsets
MSI / mSI : Max/Min Support Infrequent Itemsets
MSS/mSS: Max/Min Support Sensitive Itemsets
support
MSF
frequent itemsets
layer 0
mSF
msup
MSI
layer 1
ψ
x
MSS
layer 2
s
mSS
infrequent itemsets
layer 3
r
mSI
(i)
support
6 A
5
4 D
3 C
msup
2 B, AB, CD
1 AC, AD, ACD
0 BC, BD, ABC, ABD, BCD, ABCD
(ii)

Figure 10.1: A layered approach to quantifying the privacy that is offered by the exact hiding algorithms.

Figure 10.1(i) demonstrates the proposed layered approach as applied on a sanitized database $\mathcal{D}$. The support–axis is partitioned into two regions with respect to the minimum support threshold *msup* that is used for the mining of the frequent itemsets in $\mathcal{D}$. In the upper region (above *msup*), Layer 0 contains all the frequent itemsets found in $\mathcal{D}$ after the application of a frequent itemset mining algorithm like Apriori [14]. The value of MSF indicates the maximum support of a frequent itemset in $\mathcal{D}$. The region starting just below *msup* contains all the infrequent itemsets, including the sensitive ones, provided that they were appropriately covered up by the applied hiding algorithm. In what follows, we proceed to further partition the region below *msup* into three layers:

Layer 1 This layer spans from the infrequent itemsets having a maximum support (MSI) to the sensitive itemsets with a maximum support (MSS), excluding the latter ones. It models the "gap" that may exist below the borderline, either due to the use of a margin of safety to better protect the sensitive knowledge (as is the typical case in various

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

hiding approaches, e.g. [152]), or due to the properties of the original database $\mathcal{D}_O$ and the sensitive itemsets that were selected to be hidden. We consider this layer to contain ψ itemsets.

Layer 2 This layer spans from the sensitive itemsets having a maximum support (MSS) to the sensitive itemsets with the minimum support (mSS), inclusive. It contains all the sensitive knowledge that the owner wishes to protect, possibly along with some nonsensitive infrequent itemsets. We consider this layer to contain s itemsets out of which S are the sensitive ones.

Layer 3 This layer collects the rest of the infrequent itemsets, starting from the one having the maximum support just below mSS and ending at the infrequent itemset with the minimum support (mSI), inclusive. This layer is assumed to contain a total of r itemsets.

Given the layered partitioning of the itemsets in $\mathcal{D}$ with respect to their support values, we argue that the quality of a hiding algorithm depends on the position of the various infrequent itemsets in the Layers 1–3. Specifically, let x denote the distance (from *msup*) below the borderline where an adversary tries to locate the sensitive knowledge (e.g., by mining database $\mathcal{D}$ using support threshold $msup - x$). Then, estimator $\tilde{\mathrm{E}}$ provides the mean probability of sensitive knowledge disclosure, defined as follows:

$$\tilde{\mathrm{E}} = \begin{cases} 0 & x \in [0 \ldots \psi] \\ \dfrac{S \cdot \frac{msup-x}{\psi+MSS-mSS+1}}{\psi + s \cdot \frac{msup-x}{\psi+MSS-mSS+1}} & x \in (\psi \ldots (\psi + \mathrm{MSS} - \mathrm{mSS} + 1)] \\ \dfrac{S}{\psi + s + r \cdot \frac{msup-x}{msup-mSI+1}} & x \in ((\mathrm{MSS} - \mathrm{mSS} + 1) \ldots (\mathrm{msup} - \mathrm{mSI} + 1)] \end{cases} \tag{10.1}$$

By computing $\tilde{\mathrm{E}}$ for the sanitized database $\mathcal{D}$, the owner of $\mathcal{D}_O$ can gain in–depth understanding regarding the degree of protection that is offered on the sensitive knowledge in $\mathcal{D}$. Furthermore, he/she may decide on how much lower (with respect to the support) should the sensitive itemsets be located in $\mathcal{D}$, such that they are adequately covered up. As a result, a hiding methodology can be applied to the original database $\mathcal{D}_O$ to produce a sanitized

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

$$\textbf{maximize}\left(\sum_{u_{nm}\in U} u_{nm}\right)$$

$$\textit{subject to}\begin{cases} \sum_{T_n\in D_X}\prod_{i_m\in X} u_{nm} < \text{msup} - x, \forall X \in \mathcal{S}_{min} \\ \sum_{T_n\in D_R}\prod_{i_m\in R} u_{nm} \geq \text{msup}, \forall R \in V \end{cases}$$

Figure 10.2: The modified CSP for the inline algorithm that guarantees increased safety for the hiding of sensitive knowledge.

version $\mathcal{D}$ that meets the newly imposed privacy requirements. Given the presented exact approaches to sensitive knowledge hiding, such a methodology can be implemented in two steps, as follows:

1. The database owner uses the probability estimator $\tilde{E}$ to compute the value of x that guarantees maximum safety of the sensitive knowledge.

2. An exact knowledge hiding approach is selected and extra constraints are added to the formulated CSP to ensure that the support of the sensitive knowledge in the generated sanitized database will become at most x.

For example, in the case of the inline approach, the CSP of Figure 10.2 guarantees the holding of these requirements[1]. Another possibility is to apply a post–processing algorithm that will increase the support of the infrequent itemsets of Layer 3 in the sanitized database $\mathcal{D}$, such that they move to Layer 2 (thus increase the concentration of itemsets in the layer that contains the sensitive ones). On the negative side, it is important to mention that all these methodologies for increasing the safety of the sensitive knowledge have as an effect the decrement of the quality of the sanitized database, with respect to its original counterpart. This brings up one of the most commonly discussed topics in knowledge hiding: *hiding quality vs. usability of the sanitized database*, offered by the hiding algorithm.

[1]We should also point out that the owner of the data can decide to hide different sensitive itemsets at a different degree, thus consider some of these itemsets as more sensitive than the others. To achieve that he/she can properly adjust the support threshold (i.e. the right side of the inequalities) in the corresponding constraints of the CSP of Figure 10.2, involving these sensitive itemsets.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 10.1: An example of a sanitized database $\mathcal{D}$ produced by the inline algorithm [64], which conceals the sensitive itemsets $\mathcal{S} = \{B, CD\}$ at a frequency threshold of *mfreq* = 0.3.

A	**B**	**C**	**D**
1	1	0	0
1	1	0	0
1	0	0	0
1	0	0	0
0	0	0	1
1	0	1	1
0	0	1	1
0	0	1	0
1	0	0	0
0	0	0	1

Figure 10.1(ii) demonstrates the proposed approach for the example database of Table 10.1. As expected, due to the minimum harm that is introduced by the exact hiding algorithms, both sensitive itemsets B and CD are located just under the borderline. In this example, the size of Layer 1 is zero (i.e. $\psi = 0$). Based on the estimator $\tilde{\text{E}}$ we compute that the probability of an adversary to identify the sensitive knowledge equals 2/3 when using $x = 1$ (equivalently when mining the database using $msup = 2$). Since the probability of sensitive knowledge disclosure is high, the owner of the data could either (i) use the CSP formulation of Figure 10.2 to constraint the support of the sensitive itemsets to at most 1, or (ii) apply a methodology that increases the support of some of the itemsets in Layer 3 so as to move to Layer 2 (i.e. have a support of 2). Both approaches are bound to introduce extra distortion to the original database $\mathcal{D}_O$ from which $\mathcal{D}$ was produced, but will also provide better protection of the sensitive knowledge.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 11

Summary and Future Work

In this first part of the dissertation, we studied the problem of association rule hiding in large transactional databases. In Chapter 1, we introduced the problem and shed light on the research challenges in this area of research. Chapter 2 presented all the necessary background for the proper understanding of the problem, as well as provided the problem formulation and the problem statement. In Chapter 3, we surveyed the recent advances in association rule hiding, along three principal directions: (i) heuristic–based approaches, (ii) border–based approaches, and (iii) exact knowledge hiding approaches. Chapter 4 discussed our research contributions in this area. Since 2006, we have proposed three exact hiding algorithms that are experimentally proven to lead to superior hiding solutions, when compared to state–of–the–art heuristics. All our proposed approaches are based on the process of border revision, which was highlighted in Chapter 5. In Chapters 6, 7 and 8 we elaborated on the workings of our contributed methodologies. As we demonstrated through experimentation, the optimality in the hiding solution that is computed by the exact hiding algorithms, comes at a cost to the computational complexity of the sanitization process. This shortcoming led us to investigate ways to improve the poor scalability of the exact hiding algorithms, which is primarily influenced by the increased complexity of solving the produced CSPs. In [68], we proposed a partitioning approach that improves the scalability of the hybrid algorithm. Then, in Chapter 9, we proposed a framework which is suitable for decomposition and par-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

allel solving of the CSPs that are produced by the exact hiding algorithms. Through the use of this framework, the exact algorithms can scale to very large problem instances, provided that the necessary number of processors are available. Finally, in Chapter 10, we proposed a novel layered approach to the quantification of the privacy that is offered by the exact hiding methodologies. By using this approach, the data owner can direct the exact hiding algorithms to produce a database that adequately protects the sensitive itemsets against disclosure, thus achieving the desired level of privacy at a minimal distortion of his/her data. Moreover, the data owner is capable of deciding the extent to which each sensitive itemset will be hidden in the sanitized database by modifying the corresponding constraints in the CSP that is produced by the employed exact methodology.

As a future work in the area of association rule hiding, we intend to shed light on the two following research problems. First, we want to investigate the possibility of further reducing the size of the CSPs that are generated by the exact hiding methodologies, while still guaranteeing the optimality of the hiding solution. We envision the reduction of the size of the CSP not in terms of how many itemsets will be controlled (i.e. each itemset should continue to be represented in the CSP to allow for an exact hiding solution) but in the form of the inequalities and the unknowns that would be necessary for controlling the status (frequent vs infrequent) of these itemsets in the sanitized database. Second, we would like to propose a methodology for the computation of the exact number of transactions that need to be appended to the original database by the hybrid algorithm [68], in order to facilitate exact knowledge hiding. This methodology will allow us to reduce the size of the CSP that is produced by the hybrid algorithm, as well as to eliminate the post–processing phase regarding the validity of the transactions in the produced database extension.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Part II

Hiding Sensitive Trajectory Patterns

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 12

Introduction

The remarkable progress that is evidenced nowadays in the areas of telecommunications and location tracking technologies, such as GPS, GSM and UMTS, has made possible the tracking of mobile devices (and subsequently their human companions) at an accuracy that spans from a few hundred meters to only a few meters. As ubiquitous computing pervades our society, the available wireless network infrastructures can be used to sense and record the movement of people, animals, and vehicles, thus generate large volumes of geo–spatial, mobility data. From this perspective, we have nowadays the means of collecting, storing and processing mobility data of an unprecedented quantity, quality and timeliness.

A mobility dataset is composed of a series of space–time recordings which capture the changes in the physical location of a set of moving entities throughout a certain amount of time, usually discretized in a set of time instants. People, as well as animals, are entities that move purposely in order to accomplish certain goals. For example, as well stated in [18], "the movements of people may greatly depend on their occupation, age, health condition, marital status, and other properties". In all cases, the movement traces that are left behind by the mobile devices of the users, are an excellent source of information which can aid traffic engineers, city managers and environmentalists towards decision making in a wide spectrum of contexts, such as urban planning, traffic engineering, forecasting of traffic–related phenomena, fleet management, wildlife rescuing, and environmental pollu-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

tion. As an effect, the analysis of the user movement traces by means of appropriate patterns and models extracted through the application of modern data mining technologies, can potentially lead to a source of knowledge of extreme usefulness. The existing large volumes of mobility data, as exemplified by trajectory data collected during the course of various mobile applications (primarily attributed to the usually high update rates that are used for their collection), makes the manual analysis of this data impossible.

Mobility data is of spatiotemporal nature, and as an effect its mining has to account for both its spatial and its temporal characteristics. Spatiotemporal data mining is nowadays emerging as a novel and challenging area of research, targeting at the in–depth analysis of large spatiotemporal datasets, such as datasets of user mobility. However, we should realize that spatiotemporal data mining can become a double–edged sword. As it becomes evident, on the wrong hands this type of emergent knowledge may lead the way to an abuse scenario, where highly sensitive personal information is revealed to unauthorized, potentially malevolent, entities. In the case of mobility data, some examples of misuse include user tailing, surveillance, and user profiling. Generally speaking, the personal mobility data, as sensed through the available telecommunication infrastructures, represents an extremely sensitive source of information; its potential disclosure to untrusted third parties may severely jeopardize the privacy of the individuals to whom this data refers [27,75]. Dobson and Fisher [47] describe the severity of the unauthorized disclosure of personal mobility data or emergent knowledge patterns mined from this data, as "a new form of slavery characterized by location control". To effectively address the privacy concerns regarding the mining of mobility (and, generally, of spatiotemporal) data, a novel set of specially–crafted privacy preserving data mining methodologies have been recently proposed.

In the following sections, we elaborate on the research challenges that pertain to the mining of spatiotemporal data as well as the privacy aware mining of mobility data (Section 12.1) and then proceed to discuss our research contributions (Section 12.2), presented in detail in the chapters that follow. Finally, in Section 12.3, we present the organization of the rest of the chapters in this part of the dissertation.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

12.1 Research Challenges

Mobility data raises a series of important research challenges with respect to its mining, primarily due to the fact that it is referenced both in space and time. The existing techniques for the mining of mobility (and generally of spatiotemporal) data can be classified along the following principal dimensions: (i) the type of extracted knowledge patterns, (ii) the population in reference (i.e. the coverage of the patterns), and (iii) the data dimension that is in focus. With respect to (i), methodologies have been proposed that extract different types of patterns; clustering, association rule mining, classification rule mining, sequential pattern mining, and trend discovery are some among the data mining tasks that have been investigated in the context of spatiotemporal data. Some of the algorithmic approaches that have been proposed for the mining of spatiotemporal data, extend existing techniques for knowledge discovery in classical relational or transactional data to take into consideration the spatiotemporal annotations in the course of mining. With respect to (ii), *local* methodologies have been proposed for the mining of patterns that characterize only a portion of the population that is recorded in the dataset (e.g., a subset of individuals or a limited spatial/temporal or spatiotemporal region), as well as *global* patterns that summarize the trends in the complete data space.

We consider the third dimension of the proposed classification to be the most prevalent. Due to the complexity of spatiotemporal data mining, few approaches exist that handle both the space and the time dimension of the data equally well. Indeed, most of the proposed approaches focus only on one of the dimensions of the data, by either suppressing or devaluing the information that is offered by the other dimension. Specifically, in spatial data mining approaches [154] the focus is given on the spatial characteristics of the dataset, while any temporal characteristics of the data are suppressed. Similarly, temporal data mining approaches [110] give emphasis on the temporal dimension of the data, while the spatial dimension is suppressed. Within the domain of temporal data mining, sequential pattern mining [188] has received particular attention and has influenced most of the approaches that have been proposed so far for the privacy aware mining of mobility data. The

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

sequential pattern mining framework in the context of spatiotemporal data requires that the spatial dimension of the data is simplified to a number of items denoting places of interest, while the temporal dimension provides the ordering of these items in the transactions. The goal of these methodologies is to discover ordered sequences of items (i.e. sequential patterns) that occur in a significant portion of the data.

The challenges in the mining of spatiotemporal data have highly influenced the methodologies that have been proposed so far for the privacy aware mining of mobility data. As it becomes apparent, the removal of just the obvious identifiers from the user trajectories (e.g., the user id, the user name, etc) is insufficient to protect the privacy of the users once the dataset is released. This is due to the spatiotemporal nature of the data which enables malevolent users to use publicly available information, such as the white pages, along with geocoded information, in order to search for certain places in the released user trajectories. For example, an adversary may use publicly available data sources to locate the houses of a set of users and collocate this information along with the released user trajectories. Since users typically start their day by leaving their house to go to work, this information can easily expose the identity of the owner of at least some of the publicized trajectories. On the other hand, the knowledge patterns that can be mined from historical datasets of user mobility may also infringe user privacy by enabling intrusive inferences regarding the habits or the sexual preferences of a portion of the population, or even providing the means for unsolicited advertisement and user profiling. As it becomes evident, both the protection of the sensitive mobility data prior to the course of data mining, as well as the hiding of sensitive knowledge patterns that can be found from the mining of historical datasets of user mobility, are very challenging problems that postulate extensive research.

12.2 Contributions

In this part of the dissertation, we extend the application domain of privacy aware methodologies for knowledge hiding from traditional relational data to historical data of user mobility. The trajectories of the users, generated in the course of their movement and recorded

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

in a dataset, lead to a far more powerful type of knowledge than the one that is offered by frequent itemsets. First of all, in mobility datasets we acknowledge an ordering of the elements per user trajectory, defining the sequence of traces that represents the trajectory that was followed by the user. Furthermore, unlike traditional relational data, mobility data is referenced both in space and time, which significantly adds to the complexity of the privacy aware mining of this data.

Our contributions is this part of the dissertation are the following. First, we provide an up–to–date survey of the existing methodologies for the privacy aware mining of user mobility data. Second, we propose a novel privacy aware trajectory tracking query engine that operates on top of a popular trajectory database engine (HERMES [138, 139]) to enable a set of untrusted end users to query trajectory data stored in a database. The proposed engine (HERMES++ [73, 174]), ensures that the answers that are returned to the end users meet specific privacy standards, as well as it effectively blocks certain types of attacks regarding user identification and user tracking.

12.3 Organization of the Rest of Part II

The rest of this part of the dissertation is organized as follows. In Chapter 13, we provide a survey of the existing methodologies for the privacy aware mining of mobility data. Chapter 14 presents our proposed trajectory tracking query engine, which offers strict guarantees about what can be observed by untrusted third parties. Finally, Chapter 15 provides a short summary along with some potential future directions in this challenging area of research.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 13

Related Work

In the last few years, a set of privacy preserving methodologies have been proposed for the protection of sensitive data and/or sensitive knowledge related to user mobility. The existing so far methodologies can be partitioned into two broad categories [76]: (i) methodologies that protect the sensitive data related to user mobility prior to the course of data mining, and (ii) methodologies that hide sensitive knowledge patterns that summarize user mobility, identified as a result of the application of data mining. The first category of approaches collects data perturbation and obfuscation methodologies that distort the original dataset to facilitate privacy aware publication of mobility data (see Section 13.1), as well as distributed privacy aware methodologies for secure multiparty computation (see Section 13.2). On the other hand, the second category of approaches treats the mobility data as sequential data and applies sequential pattern hiding techniques to prevent the disclosure of the sensitive patterns in the course of sequential pattern mining (see Section 13.3). After the application of these approaches, only the nonsensitive patterns, summarizing user' movement behavior, survive the mining process, while the sensitive ones are suppressed in the data mining result. In the sections that follow, we investigate the methodologies that have been proposed along each of these research directions.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

13.1 Data Perturbation and Obfuscation

Data perturbation and obfuscation approaches aim at sanitizing a dataset containing user mobility data, in such a way that an adversary can no longer match the recorded movement of each user to a particular individual (thus reveal the identity of the user based on his/her recorded movement in the sanitized dataset). In what follows, we consider that user mobility is captured as a set of trajectories that depict the locations and times in the course of the user's history of movement. We assume that these location–time recordings occur at a reasonably high rate that allows the tracking of user movement in the original dataset. For example, an adversary could use these recordings to track the user down to his/her house or place of work, even if the user trajectory was not accompanied by an explicit identifier, such as the user id, the social security number or the name of the user.

Hoh and Gruteser [93] present a data perturbation algorithm that is based on the idea of path crossing. The proposed approach identifies when two nonintersecting trajectories that belong to different users are reasonably close to each other and generates a fake crossing of these two trajectories in the sanitized dataset. The goal of this approach is to prevent an adversary from successfully tracking a complete user trajectory in the sanitized dataset, and thus identifying the corresponding user. Provided that many crossings of trajectories exist in the sanitized dataset, the probability that an adversary succeeds in following the same individual prior and after a crossing, sufficiently deteriorates. As the authors demonstrate, path confusion can be formalized as a constrained nonlinear optimization problem which, when given the trajectories of two users within a bounded area where a crossing has to occur, it estimates the perturbed locations for each user such that their trajectories meet within a pre–specified time period. At each generated fake user location towards the meeting of the two trajectories, the algorithm takes special care to keep the enforced perturbation of the exact user location within reasonable bounds. In order to achieve this, each perturbed (fake) location has to reside within a given perturbation radius (indicating the maximum allowable perturbation) from the original user location. Figure 13.1 presents an example of path perturbation for two users who move in parallel. As is expected, a larger

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

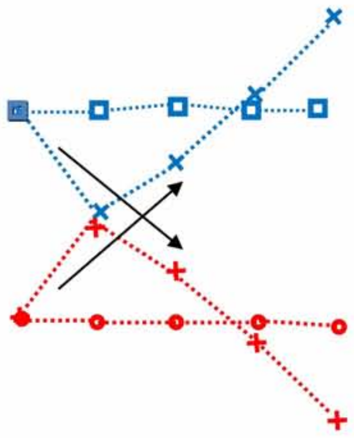


Figure 13.1: An example of path crossing between two users that move in parallel (the black arrows indicate the perturbed paths and their artificial crossing).

radius increases the degree of privacy that is offered to the users but also deteriorates the utility of the sanitized dataset. Equivalently, a smaller radius offers less privacy to the users but achieves a better utility of the publicized data. Through experiments the authors prove that the proposed algorithm limits the duration in which an adversary can successfully track the same individual in the sanitized dataset.

Hoh, et al. [94] introduce a new empirical measure for the quantification of the privacy that is offered in a dataset of publicized user trajectories. The proposed measure calculates the time that a user can be successfully tracked (by an adversary) based on the knowledge of his/her user trajectory. To achieve that, it calculates the time that elapsed between two consecutive occasions where the adversary could not determine (at least with sufficient certainty) the next location–time recording in the trajectory of the user. By using this measure, the authors propose a path perturbation strategy that relies on data coarsening to exclude a limited amount of location–time recordings from a user trajectory. The applied coarsening strategy ensures that the corresponding user cannot be tracked (at least with sufficient certainty) by the adversary, for a time that exceeds a pre–specified time threshold (i.e. the maximum time to confusion). To achieve this goal, the perturbation algorithm discloses a location–time recording of a user trajectory (as it appears in the original dataset) to the sanitized dataset, only if the time that has passed since the last point of confusion is

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

below the pre–specified time threshold. An extension of this algorithm that offers increased privacy guarantees about what can be learned by an adversary, is also examined.

Terrovitis and Mamoulis [164] consider datasets that depict user mobility in the form of sequences of places that each user has visited in the course of his/her movement. For each user, the authors assume the existence of a transaction in the dataset that contains the list of places that this particular user has visited (e.g., based on his/her card transactions), set out in the order of visit. No other information of spatial or temporal nature (e.g., the exact time of each visit) is assumed to be provided. Based on this type of data, the authors propose a suppression technique that removes some of the places that were visited by specific users, in order to protect their identity from adversaries who hold partial information on the user trajectories. Specifically, an adversary is considered to be any individual who has knowledge of certain places that were visited by particular users, for whom he/she knows their identity. To exemplify, consider a bank which has many branches in a city. Each branch of the bank has some ATM machines that people can use to perform regular money transactions. Whenever a person uses the ATM of the bank his/her transaction is recorded. Now assume that the bank manager possesses the original dataset of user mobility, where he/she identifies that some of the users that appear in the dataset have visited certain branches of the bank. By using this information it is possible that the manager can figure out the identity of some of the users who are recorded in the dataset and then learn the other places that they have visited during their movement. To protect the privacy of the users when publicizing their movement data, the proposed methodology assumes that the data holder has knowledge of the sets of places (i.e. the projection of the dataset) that are known to each individual adversary. In our example, the data holder knows the branches of the bank that the bank manager controls. By using this information, the data holder can compute the probability by which the corresponding adversary can infer the identity of a user in the publicized dataset, based on the projection of the data that the adversary holds.

The proposed suppression strategy operates in an iterative fashion to minimize the probability of a given adversary to associate (based on his/her data projection) a place that

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

id	trajectory
t_1	$b_1 \rightarrow a_1 \rightarrow b_2$
t_2	$b_1 \rightarrow a_1 \rightarrow b_2 \rightarrow a_3$
t_3	$b_1 \rightarrow a_2 \rightarrow b_2$
t_4	$b_1 \rightarrow b_2 \rightarrow a_2$
t_5	$b_1 \rightarrow b_3 \rightarrow a_1$
t_6	$b_3 \rightarrow a_1$
t_7	$b_3 \rightarrow a_2$
t_8	$b_3 \rightarrow a_2 \rightarrow a_3$

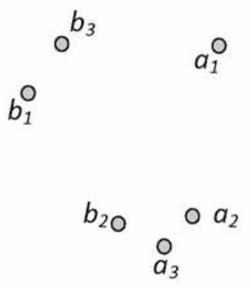
(a) ***exact data (T)***

id	trajectory
t_1^B	$b_1 \rightarrow b_2$
t_2^B	$b_1 \rightarrow b_2$
t_3^B	$b_1 \rightarrow b_2$
t_4^B	$b_1 \rightarrow b_2$
t_5^B	$b_1 \rightarrow b_3$
t_6^B	b_3
t_7^B	b_3
t_8^B	b_3

(b) ***B's knowledge (T_B)***

id	trajectory
t'_1	$b_1 \rightarrow a_1 \rightarrow b_2$
t'_2	$b_1 \rightarrow a_1 \rightarrow b_2$
t'_3	$b_1 \rightarrow a_2 \rightarrow b_2$
t'_4	$b_1 \rightarrow b_2 \rightarrow a_2$
t'_5	$b_3 \rightarrow a_1$
t'_6	$b_3 \rightarrow a_1$
t'_7	$b_3 \rightarrow a_2$
t'_8	$b_3 \rightarrow a_2$

(c) ***transformed database (T')***



(d) ***the map of locations***

Figure 13.2: An example of data suppression to generate a safe projection T_B of the exact data T, with respect to the partial knowledge of an adversary B.

appears in the publicized data to the identity of a particular person. Figure 13.2 presents an example of the proposed algorithm where the original dataset T (see Figure 13.2(a)) is transformed into its sanitized counterpart T' (see Figure 13.2(c)), which can be safely released to adversary B. As is shown in Figure 13.2(b), B's knowledge consists of all the places b_i from which he/she can retrieve information that can lead him/her to the identity of specific users. When selecting which locations to suppress from the original dataset, the proposed algorithm tries to minimize the side–effects that are introduced to the dataset. For example, location a_3 (suppressed from transactions t_2 and t_8) and location b_1 (suppressed from sequence t_5) lie close to their neighboring points (i.e. $\{b_2, a_2\}$, and $\{b_3\}$ respectively) in the original sequences (as demonstrated in Figure 13.2(d)).

Nergiz, et al. [131] also rely on the sequential nature of mobility data and propose a coarsening strategy that generates a sanitized dataset that consists of $\mathcal{K}$–anonymous se-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

quences. The principle of $\mathcal{K}$–anonymity (originally proposed by Samarati and Sweeney [149, 150, 162] in the context of relational data and recently adopted to different types of data) requires that each released data record must be indistinguishable from at least $\mathcal{K}$–1 other records with respect to a certain set of identifying variables. In the case of spatiotemporal data the identifying variables are space and time. Thus, the proposed algorithm aims at generating a dataset from the original data such that every trajectory (sequence) in the released dataset becomes indistinguishable from at least $\mathcal{K}$–1 other trajectories (sequences). Furthermore, to enhance the privacy that is offered in the sanitized data, the authors perform a sampling of the anonymized trajectories to generate representatives of users' movements and then publish the representatives instead of the anonymized trajectories.

The proposed anonymization process operates in two phases. First, the trajectories are consolidated into groups of $\mathcal{K}$ based on a similarity measure that quantifies the cost–optimal anonymization. The similarity measure that is used is inspired from the research on string alignment [84]. The employed strategy iteratively selects the most similar trajectory to the (representative of the) already co–clustered ones, uses it to compute the new representative (trajectory) of the cluster and then locates the trajectory from the dataset that lies closest to the representative. The same process continues until $\mathcal{K}$ trajectories are grouped together. Then, the co–clustered trajectories are removed from the dataset and the same operation reiterates among the remaining trajectories. In the second phase, the trajectories in each cluster are anonymized. The anonymization process operates as follows: It computes a matching between the points of each pair of co–clustered trajectories. The matched points in a pair of trajectories are replaced by their Minimum Bounding Rectangle (MBR) enclosure, while the unmatched ones are suppressed. The resulting coarsened trajectory is then used instead of its origins to participate in the following matches. As a result of this phase, every cluster of trajectories is simplified to one representative (coarsened) trajectory. Figure 13.3 shows how the anonymization is accomplished in the case of three trajectories tr_1,tr_2, and tr_3. First, the trajectories are aligned and then pairwise anonymized. The last sketch of Figure 13.3 demonstrates the employed point matching.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

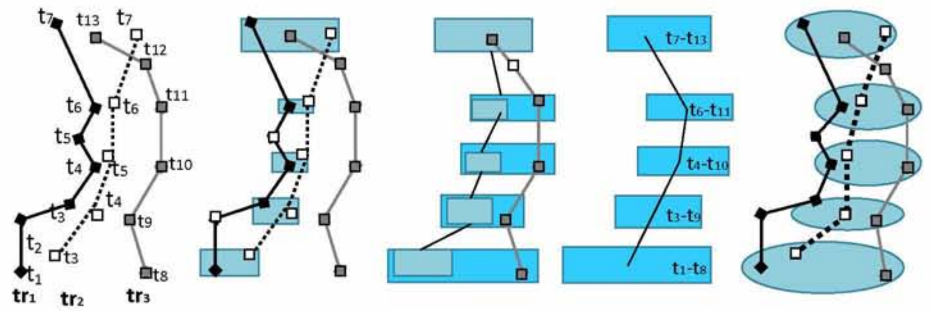


Figure 13.3: An example of the two–phase trajectory anonymization process proposed in [131], presented in a series of steps.

After the trajectory anonymization process, each representative trajectory is a $\mathcal{K}$–anonymization of the co–clustered trajectories from which it was constructed. The last step in the proposed approach regards the release of a representative sample coming from the anonymized trajectories (sequences) to the end user. The employed reconstruction process probabilistically generates a set of sample trajectories such that the reconstructed data does well in explaining the original data.

Abul, et al. [7] propose a $\mathcal{K}$–anonymity strategy that relies on the inherent uncertainty with respect to the whereabouts of user trajectories in historical datasets of user mobility [165]. As the authors mention, due to the imprecision of sampling and positioning systems, trajectories of moving objects can no longer be considered as 3D polylines, consisting of (x,y,t) elements. Instead, the authors represent the uncertainty of a trajectory by using a cylindrical volume that consists of a series of disks, defined for the different time intervals of user movement (see Figure 13.4). The sampling time is assumed to be the same for all trajectories and within each time interval the corresponding user is assumed to move linearly and at a constant speed. The proposed anonymity algorithm identifies trajectories that are close to each other in time and enforces space translation to put them also close to each other in space (in space translation, some of the trajectory points of a user trajectory are moved from their original location to another location). The algorithm operates as follows: First, it generates clusters of at least $\mathcal{K}$ members (trajectories) by employing a clustering technique that constraints the radius of the identified clusters. Then, it applies

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

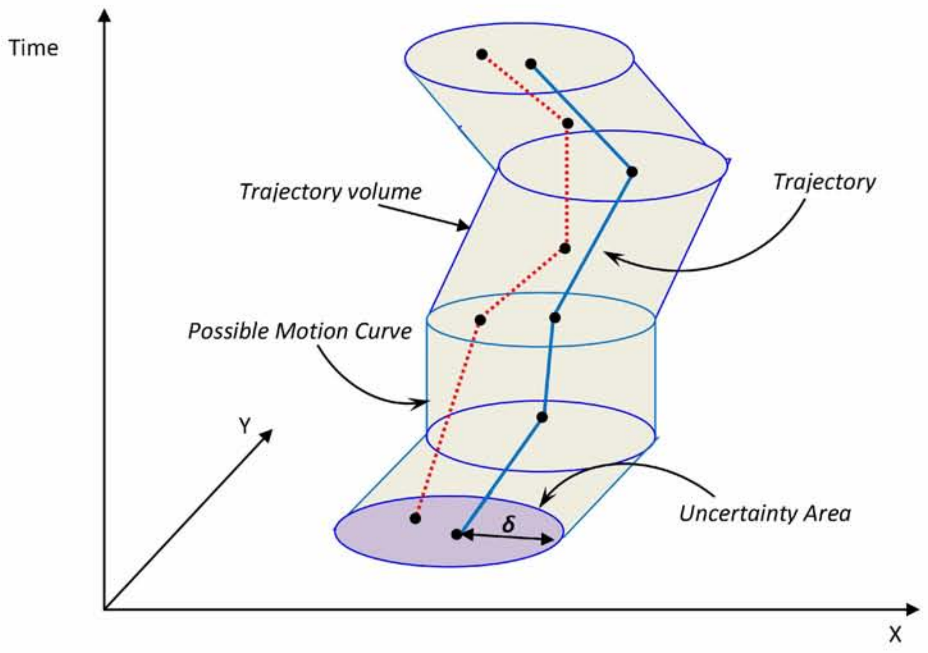


Figure 13.4: Modeling the uncertainty regarding the whereabouts of a user trajectory.

space translation in order to transform each cluster to a $\mathcal{K}$–anonymity region by relocating the trajectories that participate to this cluster to a cylinder of a certain diameter δ. An example of a 2–anonymity region (cylinder) of diameter δ is presented in Figure 13.5.

13.2 Privacy Aware Knowledge Sharing

The previous category of approaches aims at generating a sanitized dataset from the original one, which can be safely shared with untrusted third parties as it contains only nonsensitive data. Secure Multiparty Computation (SMC) provides an alternative family of approaches that can safely protect the sensitive data that appears in a dataset. The SMC approaches consider a set of collaborators who wish to collectively mine their data but are unwilling to disclose their own datasets to each other. As it turns out, this distributed privacy preserving data mining problem can be reduced to the secure computation of a function based on distributed inputs and it is thus solved by using cryptographic approaches. Pinkas [142] elaborates on this close relation that exists between privacy aware data mining and cryp-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

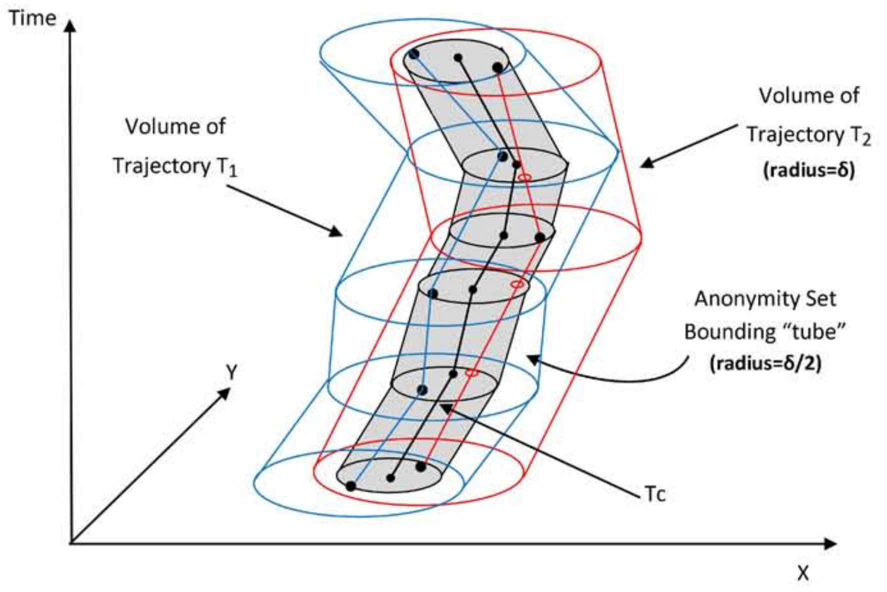

Time
Volume of Trajectory T_1
Volume of Trajectory T_2 (radius=δ)
Anonymity Set Bounding "tube" (radius=δ/2)
Y
Tc
X

Figure 13.5: An example of a 2–anonymity set of diameter δ.

tography. In SMC, each party contributes to the computation of the secure function by providing its private input. A secure cryptographic protocol that is executed among the collaborating parties ensures that the private input that is contributed by each party is not disclosed to the others. Most of the applied cryptographic protocols for multiparty computation result to some primitive operations that have to be securely performed: secure sum, secure set union, and secure scalar product. Clifton, et al. [42] discuss these operations.

SMC has been studied in the context of user mobility (and more generally on spatiotemporal) data. Inan and Saygin [96] were the first to propose a privacy aware methodology for the clustering of a set of spatiotemporal datasets, owned by different parties. To perform clustering, a similarity measure is necessary to quantify the proximity between two objects (e.g., the user trajectories), such that in the computed clustering solution the co–clustered objects are more similar to one another than to objects belonging in different clusters. As part of this work, the authors propose a secure protocol that can be employed to enable the pairwise secure computation of trajectory similarity among all the trajectories of the different parties, thus building a global matrix of trajectory similarity. By using this

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

matrix, a trusted third party can perform the clustering on behalf of the users and communicate the computed clustering results back to the collaborating parties. The proposed privacy preserving protocol supports all the necessary basic operations for the computation of trajectory similarity based on widely adopted trajectory comparison functions: (i) Euclidean distance, (ii) longest common subsequence, (iii) dynamic time warping, and (iv) edit distance. The protocol makes the following assumptions: (i) it assumes the semi–honest model in which all the parties follow the protocol but may also store any information that they receive from other parties in order to infer private data, (ii) the parties do not mutually share any other kind of information, and (iii) the mobility data that is to be clustered follows a horizontal partitioning. The proposed methodology operates as follows: (i) every involved party, including the trusted party, generates pairwise keys which are used to disguise the exchanged messages, (ii) each party locally computes the trajectory similarity matrix (based on the commonly accepted trajectory comparison function) for its own trajectories and securely transmits it to the trusted party, (iii) for every pair of trajectories that belong to the datasets of different parties, the two parties execute the protocol to compute the similarity of their trajectories, build a similarity matrix based on their trajectories, and subsequently transmit it to the trusted party, and (iv) the trusted party uses the computed matrix of trajectory similarity based on the trajectories of all the collaborating parties, in order to perform trajectory clustering. An interesting observation is that by using this technique, the trusted party is free to choose any clustering algorithm, depending on the requirements of the data holders, in order to perform the clustering of the trajectories.

13.3 Sequential Pattern Hiding

The extraction of frequent patterns from mobility data has primarily focused on the sequential nature of the datasets by extracting frequent subsequences of user mobility (e.g., Cao, et al. [32], Giannotti, et al. [60]). Giannotti, et al. [61] proposed the integration of spatial and temporal information in the extracted mobility patterns by temporally annotating the sequences with transition times from one element (place of interest) to another. The re-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

sulting patterns (called $\mathcal{T}$–patterns) depict the frequent movement behavior of the objects whose trajectories are recorded in the dataset. An example of a $\mathcal{T}$–pattern is Home $\xrightarrow{15min}$ Park $\xrightarrow{30min}$ Work. In a similar manner, the approaches that have been proposed for the hiding of frequent mobility patterns consider knowledge hiding in the form of sequential pattern hiding. In what follows, we present some approaches that belong to this category.

Abul, et al. [9] model the problem of trajectory hiding to that of sequential pattern hiding. The authors consider that pertinent to every sensitive sequence is a disclosure threshold that defines the maximum number of sequences in the sanitized database that are allowed to support the sensitive sequence. The sequence sanitization operation is based on the use of unknowns to mask selected elements in the sequences of the original dataset. As the authors prove, the problem of sanitizing a sequence from the original dataset, while introducing the least amount of unknowns, is NP–hard and thus one needs to resort to heuristics to identify an efficient solution. The proposed heuristic operates as follows: For each sensitive sequence, the algorithm searches all the sequences of the original database to identify those in which the sensitive sequence is a subsequence (a sequence S1 is a subsequence of another sequence S2 if it can be obtained by deleting some elements from S2). For every such sequence of the original dataset, the algorithm examines in how many different ways this sequence becomes a subsequence of the sensitive one. Each "different way" (also called a matching) is counted based on the position of each element in the sequence that participates to the generation of the sensitive sequence. As an effect, for each element of the sequence coming from the original dataset, the algorithm maintains a counter depicting the number of matchings in which it is involved. To sanitize the sequence, the algorithm iteratively identifies the element of the sequence which has the highest counter (i.e. it is involved in most matchings) and replaces it by an unknown, until the sensitive sequence is no longer a subsequence of the sanitized one. As a result of this operation, the sensitive sequence becomes unsupported by the sanitized sequence. In order to enforce the requested disclosure threshold the algorithm applies this sanitization operation in the following manner: For each sensitive sequence, all the sequences of the original dataset

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

are sorted in ascending order based on the number of different matchings that they have with the sensitive sequence. Then, the algorithm sanitizes the sequences in this order, until the required disclosure threshold is met in the privacy aware version of the original dataset. The authors have developed extensions of this approach for the handling of time constraints, such as min/max gap and max window.

A similar approach to [9], which operates by removing (instead of masking) elements from temporally annotated sequences [60] representing user mobility in an underlying network of user movement, is presented in the work of Abul, et al. [8]. The employed network topology confines the movement of the users into a set of allowable routes. Each edge of the network is assumed to be a directed straight line which is annotated to denote the minimum time that is required for its traversal (i.e. the traversal of the corresponding road part). Given the underlying network, a trajectory pattern can be represented as a time labeled sequence $u_0 \xrightarrow{a_1} u_1 \xrightarrow{a_2} \dots \xrightarrow{a_m} u_m$, where u_i denotes a vertex of the network and a_j is the minimum time that is required for the transition from one vertex to the next one. A user trajectory supports a trajectory pattern if it is a time–consistent (based on the annotations) subsequence of the pattern. The proposed hiding approach identifies all the user trajectories that support a sensitive trajectory pattern and sanitizes some of them through coarsening, such that the pattern becomes insignificant. The employed sanitization strategy ensures that the sequences that participate to the sanitized dataset remain consistent to the underlying network. It operates in exactly the same manner as [9] to select which spatiotemporal points will be removed from which trajectories, and to perform the necessary sanitization. The authors propose an extension of this strategy which protects the publicized data from inference attacks regarding the lack of alternative paths. In this type of attack, a malevolent user can infer some of the suppressed points in a sanitized user trajectory if there is only one path between two consecutive vertices in the user trajectory, which is time–consistent to the network. The solution that is proposed requires the suppression of spatiotemporal points in the original user trajectories such that between two released points there exist at least k alternative time–consistent paths that the user may have followed in the meanwhile.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Pensa, et al. [140] propose a sequential pattern hiding methodology that aims at removing all the infrequent subsequences from a dataset of user mobility. The proposed algorithm operates in four steps: (i) a prefix tree is generated based on the sequences of the original dataset, (ii) the infrequent subsequences are pruned away from the tree, thus the prefix tree is anonymized, (iii) part of the removed frequent subsequences (lost as a side–effect of the brute–force pruning operation) are re–appended to the prefix tree, and (iv) the sequences of the prefix tree are used to generate the sanitized dataset, which can be safely released. In the first step, the prefix tree is constructed by inserting each sequence of the original dataset as a path of the tree (each element of the sequence is a node and two nodes are joined together by an edge). For the insertion, the algorithm locates the longest prefix path in the tree that is common to the sequence and then the sequence is accordingly inserted. Following the insertion operation, the support of each node in the common prefix path is updated based on the support that the sequence has in the original dataset. After all the sequences are inserted to the prefix tree, the tree is pruned based on the user–specified minimum support threshold. The applied pruning operation aims at the removal of all the infrequent subtrees from the tree; however, it also loses some of the frequent sequences. The support of each path (sequence) in the pruned tree is accordingly updated. Following that, a portion of the lost frequent subsequences are re–appended to the pruned tree. This is achieved through the increment of the support of certain nodes in the tree based on the support of the corresponding subsequences. The final step of the proposed methodology regards the generation of the sanitized dataset based on the sequences that are currently represented in the prefix tree. To generate the sanitized counterpart of the original dataset, the proposed algorithm visits the sequences (paths) of the prefix tree and for each visited path it appends the corresponding sequence (based on its support) to the sanitized dataset.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 14

A Privacy Aware Trajectory Tracking Query Engine

In this chapter, we present a privacy aware trajectory tracking query engine that provides on–site, restricted access to in–house data to facilitate privacy aware data publishing. The proposed engine provides all the necessary mechanisms to secure the database against disclosure of confidential information, as well as it offers strict guarantees about what can be observed by untrusted third parties. Contrary to existing research on security control for statistical databases (c.f. [10]) where all the enforced security approaches are based solely on count and/or sum queries, since no other information can be made available to the inquirer, the proposed query engine supports a large variety of queries, involving both trajectory and non–trajectory data. Specifically, in addition to regular queries involving non–spatial non–temporal attributes, the engine supports a variety of spatiotemporal queries, including range queries, nearest neighbor queries and queries for aggregate statistics. The query results are augmented with fake trajectory data (dummies) to fulfil the requirements of $\mathcal{K}$–anonymity. Through qualitative analysis, we prove the effectiveness of our approach towards blocking certain types of attacks, while minimally distorting the original dataset.

The rest of this chapter is organized as follows: In Section 14.1 we introduce and motivate the proposed methodology. Section 14.2 presents our methodology for building

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the query engine and demonstrates its effectiveness towards tackling two types of attacks. In Section 14.3 we detail over the algorithms that are implemented as part of the query engine to accomplish the various tasks. Following that, Section 14.4 proposes a query specification mechanism that relies on a spatial and a temporal hierarchy to ease the composition of complex queries. Finally, Section 14.5 provides a qualitative analysis of the proposed approach providing insight on the capabilities of the query engine.

14.1 Introduction

Knowledge of user movement is essential to a wide variety of applications, ranging from car navigation systems and traffic control, to urban planing and location–based advertisements. However, potential disclosure of accurate trajectory information of an individual to untrusted third parties can severely compromise his/her privacy. For this reason, privacy preservation techniques are essential to control the accuracy at which user trajectory information is communicated to untrusted entities.

In this work, we consider a data holder, such as a governmental agency or a telecom operator, that collects movement information for a community of people. This information is used to approximate the real user movement and to create user trajectories, subsequently stored in a database. Apart from the analysis this data is undergone within the premises of the data holder, we assume that at least a portion of the data (if not all) has to become available to other, possibly untrusted, parties for analysis and querying purposes. As an example, consider the Octopus (more information is available at `http://www.octopuscards.com/`) smart RFID card, commonly used by Hong Kong residents both for transportation and for selected point–of–sale payments at convenience stores, supermarkets, on–street parking meters, etc. Currently, the Octopus company accumulates a large amount of movement data on a daily basis. Publication of this data would benefit several other parties, such as transportation engineers, insurance companies, and advertisers. For instance, advertisement companies would benefit by identifying locations (or routes) that are usually crowded at certain times during the day, in order to place advertisements for their products.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

It is evident that direct publishing of this information, even after removing the obvious identifiers from the dataset (e.g., name, ID number) is insufficient to protect the privacy of the individuals, whose movement is recorded. Indeed, a malevolent user who wants to compromise the dataset (commonly referred to as a *snooper*) can still learn the movement habits of all users in the database and track them down to their house or place of work. A mechanism is thus needed to regulate the answer data that is made available to the inquirer as a result of his/her query. However, such a task is impossible to be accomplished for all types of possible queries, without severely sacrificing the quality of the data. Contrary to the research in privacy aware data publishing (e.g., [7,8,164]), where authors make the implicit assumption that most of the knowledge in a dataset can become publicly available without causing privacy breaches, in our work we assume that the majority of the information that is stored in the database must remain private.

To protect user privacy, we propose a query engine that restricts querying of the trajectory database to a set of pre–defined types of queries: (i) range queries (both in time and space), (ii) queries for landmarks (e.g., "all trajectories starting from A, ending at E and passing from B, C, D"), (iii) queries for given routes (e.g., "all trajectories crossing the Park street"), (iv) nearest neighbor queries (e.g., "the trajectories of the 3 users that are currently nearest to the park"), and (v) queries for aggregate statistics (e.g., "number of people visiting the city center from 8am to 10am"). Queries involving non–spatial non–temporal attributes are also supported (e.g., "average income of people who are recorded in the dataset"). The engine retrieves real user data from the database and mixes it (if necessary) with carefully crafted dummy data. As a result, an honest user can still attain invaluable knowledge from the database, while this knowledge is insufficient for a snooper to identify or track down a specific subject. Our proposed system blocks two types of attack:

User identification attack In this type of attack the identity of the user can be exposed by a set of ad–hoc queries involving overlapping sets of spatial attributes. Each new query involves the attributes–values pairs of previous queries along with new pairs that constitute it more specific.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Sequential tracking attack In this type of attack the user can be tracked down through his/her trajectories by a set of focused queries on regions that either (partially) overlap or are adjacent to each other, in terms of space and time. Each new query attempts to "follow" the movement of the user by capturing a new part of the user trajectory.

In both types of attack, the malevolent user can increase his/her confidence regarding the identity of a user by segregating the dummies from the real user trajectories.

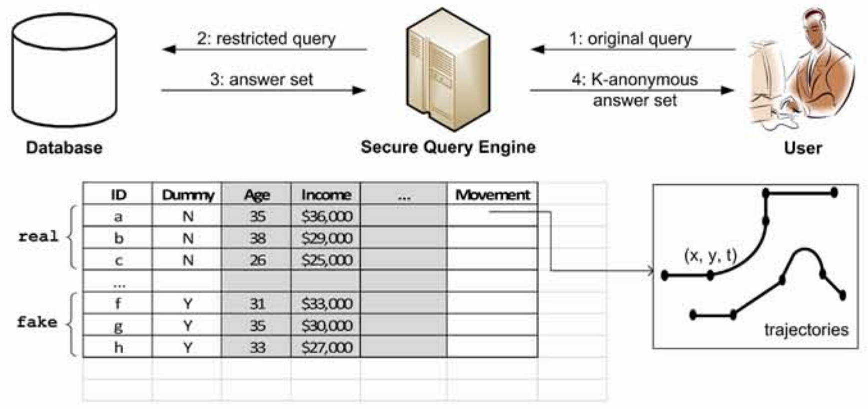
14.2 The Proposed System

In what follows, we provide a big picture of the proposed system, along with the types of the supported queries and the implemented interpolation strategies. Furthermore, we shed light on the applied mechanism for the blocking of possible attacks. Specifically, in Section 14.2.1 we provide an overview of the system and its operation. Section 14.2.2 presents the different types of queries that are handled by the query engine, while Section 14.2.3 explains the interpolation techniques that are used for the generation of fake records. Finally, Section 14.2.4 provides an example of use and Section 14.2.5 presents the defense of the system against the two types of attack.

14.2.1 System Overview

Consider a system that collects movement data for a population of users. Each user is equipped with a positioning device that regularly transmits his/her location (commonly referred to as a *location update*) to a trusted server. The server applies a trajectory reconstruction approach (such as [184]) to approximate the real route that the user has followed from the last location update to the current one. Then, it updates the movement information of the user in the database. Apart from movement data, the system is also in possession of other information for the tracked users (e.g., name, age, income, etc). Due to some mutual agreement, the data holder wants to provide access to his/her data to other interested parties. Thus, the data holder generates a view of the database that contains both the attributes and

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



ID	Dummy	Age	Income	...	Movement
a	N	35	$36,000		
b	N	38	$29,000		
c	N	26	$25,000		
...					
f	Y	31	$33,000		
g	Y	35	$30,000		
h	Y	33	$27,000		

Figure 14.1: A big picture of the proposed system architecture.

the tuples that he/she wants to release. For example, if access to the data is to be offered for an advertisement company, information about the income of people should be kept private. On the other hand, age information may be invaluable to allow for decisions regarding the types of products that will be advertised. Such a view of the database is presented in Figure 14.1. Each record of the database contains information for a single user, identified though a unique ID. The movement of the user is recorded as the collection of his/her trajectories and is stored in a special–type attribute of the record[1].

A user of the system has to register to the query engine to access and query the corresponding database view. Upon registration, the database table is augmented by a column **`Dummy`** (initialized to 'N' for all records) that holds information regarding the real and the fake trajectories. Furthermore, the data holder assigns a threshold $\mathcal{K}$ to the user that indicates the minimum number of records that should formulate the answer set of a user query. A set of procedures are made available to the user to query the database (see Section 14.2.2). When a user submits a query to the query engine, the query is checked for conformance to a set of rules whose purpose is to shield the system against snoopers (see Section 14.2.5). In case of rule violation, the system denies execution of the query. On the other hand, if no rule is violated, the database is searched to identify the answer to the query.

[1]This way of storing trajectory data is in accordance to HERMES [138], a data cartridge for moving objects, implemented as an extension of the Oracle DBMS [2].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Count queries and queries referring to non–spatial non–temporal data (e.g., "find the average age of the recorded population") are handled by common strategies for query restriction and $\mathcal{K}$–anonymity for relational data [10, 150]. However, in this paper, we focus our attention on spatial/temporal and spatiotemporal queries. Provided with such a query, the query engine identifies the records in the database that formulate the correct answer. If the size of the answer set is $R < \mathcal{K}$, the system needs to produce $\mathcal{K}$–R fake records of dummy trajectories to ensure trajectory $\mathcal{K}$–anonymity[2], unless these fake records have been already placed in the database from a previous application of the proposed technique. The answer set is formulated by the union of all the real records along with the fakes so as to reach a cardinality of $\mathcal{K}$. The dummy trajectories are produced through interpolation applied on the real trajectories (see Section 14.2.3). Once produced, the dummy trajectories are maintained in the database along with the real user trajectories for coherence purposes; followup queries to the database should produce compatible answers with respect to past queries. On the other hand, if the number of retrieved records suffice to provide $\mathcal{K}$–anonymity, the records are returned as–is to the user, without the incorporation of any fakes. In every case, along with the answer to the query returned to the user, the system provides information regarding the percentage of the real records to the returned answer set.

14.2.2 Queries for Trajectory Data

The query engine can handle queries for both trajectory and non–trajectory data. Queries for non–trajectory data include all types of queries that result either in count/aggregate data or in actual views of records that do not include the trajectory component (attribute). In the case of count queries, the system refrains from providing an answer that was computed by using less than $\mathcal{K}$ records. For example, a query for the number of users that have age under 16 will be serviced only if at least $\mathcal{K}$ users match this age limit. The same approach holds when querying for aggregate data (e.g., averages over non–trajectory data)

[2]By elevating the definition of $\mathcal{K}$–anonymity [150] from relational to trajectory data, a user is $\mathcal{K}$–anonymous if he/she can be matched to any among $\mathcal{K}$ different trajectories (commonly referred to as the *anonymity set*). In the best case scenario, given $\mathcal{K}$ trajectories, a user can be matched to the correct one with a probability of $1/\mathcal{K}$.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

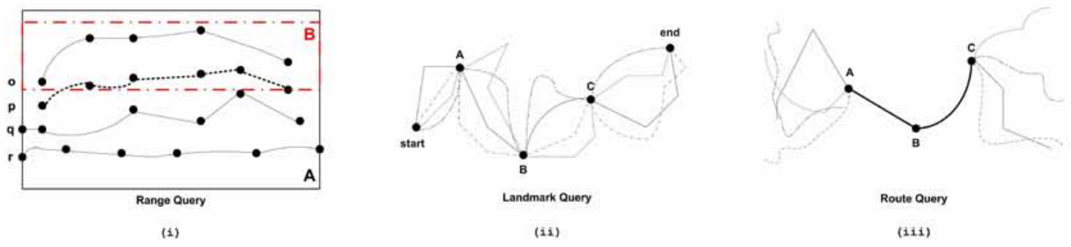


Figure 14.2: The supported types of queries when requesting trajectory data.

or requesting records from the dataset that do not contain the trajectory component (e.g., "list all users having a yearly income above \$15,000"). In all such cases, if $\mathcal{K}$–anonymity cannot be satisfied, the query engine denies answering the query. In what follows, we focus our attention on the supported queries for trajectory data. Figure 14.2 summarizes them.

Range and Nearest Neighbor Queries

In range queries (see Figure 14.2(i)) the objective is to identify user trajectories that either (i) lie within a predefined spatial, temporal or spatiotemporal region, or (ii) are within a given distance d with respect to a point of reference. In what follows, we provide some examples of range queries:

Q1 "Find all user trajectories that crossed the park".

Q2 "Find all user trajectories on Tuesday 1/1/2008, from 6am to 10am".

Q3 "Find all user trajectories that crossed the park this Monday, from 6am to 10am".

Q4 "Find all user trajectories that are at most 1Km away from the Computer Science Department of the University of Minnesota".

In k Nearest Neighbor (k–NN) queries, the objective is to identify the k nearest neighbors given a point of reference. An example of a k–NN query is:

Q5 "Find the five user trajectories that are currently closest to the park".

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

From the perspective of the query engine, all of the **Q1–Q5** queries are handled in exactly the same manner. First, the answer set is computed based on the current information of the database. If it respects the requirements of $\mathcal{K}$–anonymity then this answer set is returned to the user. On the contrary, if it fails to preserve $\mathcal{K}$–anonymity (e.g., when $k < \mathcal{K}$) then a set of fake records is created to elevate the size of the answer set to $\mathcal{K}$.

Landmark Queries

In landmark queries (see Figure 14.2(ii)), the objective of the inquirer is to identify user trajectories that have a standard start/end point and/or pass through one or more given intermediate points. All these points are called *landmarks*. A landmark query can be either spatial or spatiotemporal. Examples follow:

Q6 "Find all user trajectories that start from the museum, move to the library and end at the national garden".

Q7 "Find all user trajectories that visited the cafè last Tuesday at 8am and then the US bank at 9pm".

In the case that at least one of the start/end landmarks is not specified, the user needs to supply a rectangle region that includes all landmarks and serves as the space within which the returned trajectories are depicted.

Route Queries

In route queries (see Figure 14.2(iii)), the objective is to identify user trajectories that followed a common predetermined route. In contrast to landmark queries, in route queries all points in the route need to be part of the retrieved trajectories. Furthermore, the inquirer needs also to supply a rectangle region which contains the route and, as in the case of the landmark queries, lays out the area within which the returned trajectories should be depicted. Here are two examples of route queries:

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Q8 "Find all user trajectories that entered the bridge at point A, continued to the highway at point B and left the highway at exit C".

Q9 "Find all user trajectories that entered the bridge (point A) at 8:30am today, continued to the highway (point B) and left the highway (exit C) at 8:40am".

14.2.3 Interpolation Strategies

Fake records are created by applying interpolation techniques on the real data. In what follows, we consider two types of interpolation: numerical data interpolation and trajectory data interpolation. In both cases we assume that $\mathcal{K}-R$ fake records have to be produced, where R is the number of real records in the answer set.

Numerical Data

Numerical data interpolation involves the selection of appropriate values to fill in the numerical fields of the fake records. Selected values should lie close to the ones of the real records, while preserving the basic statistical properties of the answer set; that is the minimum, maximum and average. Suppose that we want to set the attribute values of a numerical attribute for the fake records. First, we generate $\mathcal{K}-R$ empty records and we randomly partition them into pairs of two. Then, we calculate the mean m of the R records with respect to the considered numerical attribute and we identify the (signed) minimum d_{min} and the maximum d_{max} deviation of the R records from the mean. For each pair of fake records we randomly select two values $\pm d \in [d_{min}, d_{max}]$ and we set the attribute value of one record to $m-d$ and of the other record to $m+d$. Finally, if the number of fake records is odd, the attribute of the non–paired record is assigned to the mean m.

Trajectory Data

The interpolation of trajectory data proceeds in an iterative fashion. The proposed technique randomly selects at each step a pair of trajectories from the answer set. Then, for the selected trajectories, it sweeps the **x**–axis (see Figure 14.3) to collect their location updates in

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

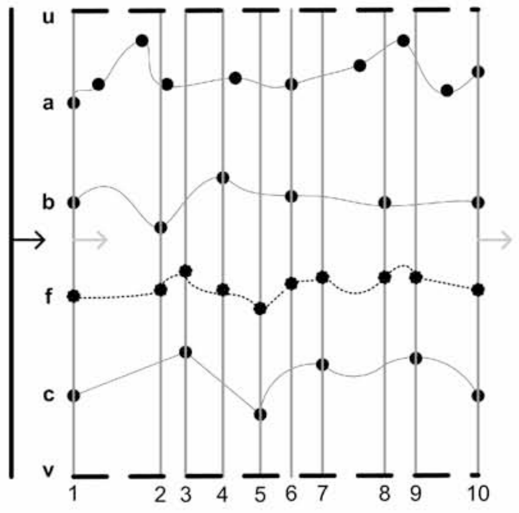


Figure 14.3: Interpolation of trajectory data.

a sorted way. For each collected location update, the algorithm identifies the corresponding (x,y,t) point on the other trajectory that has the same **x**–coordinate. This is accomplished by utilizing knowledge regarding the initially applied trajectory reconstruction approach. For example, HERMES [138, 139] supports the movement types presented in Figure 14.4. Having identified the two points, the algorithm computes a new point (x,y',t), where y' is the average of the two y–coordinates. By following the same process for all available location updates, coming from the two selected trajectories, a set of fake location updates is created for the dummy trajectory. Following that, the algorithm "reconstructs" the movement of the dummy trajectory by using the same reconstruction approach as the one of the real trajectories. For example, by using HERMES, any of the available movement functions (i.e. linear, arc, constant) shown in Figure 14.4 will do. As a last step, the created dummy trajectory is inserted to the current answer set, as well as to the database, and the algorithm proceeds to the next iteration, until the answer set offers $\mathcal{K}$–anonymity.

A slight enhancement is needed to the presented algorithm in the case of range queries, in order to handle the trajectories that are nearest to the border of the considered spatial/temporal or spatiotemporal area. If the algorithm is left as is, then these trajecto-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

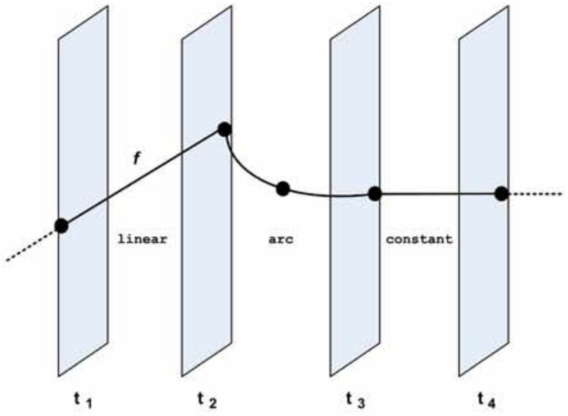


Figure 14.4: Basic movement types supported by HERMES.

ries will always be real. To fix this issue, the algorithm considers the existence of two trajectories that represent constant movement and touch the $y = y_{min}$ and $y = y_{max}$ borders, respectively. These dummy trajectories will not be part of the answer set. However, their existence allows the algorithm to produce other dummies that will lie in–between the real nearest trajectories to the borders and the respective border. For example, in Figure 14.3, dummy trajectories **u** and **v** are used for this reason. The existence of **u** enables the generation of a dummy trajectory in–between **u** and **a**. The same holds for trajectories **c** and **v**. In the case of landmark and route queries the trajectories **u** and **v** are also used but the landmarks are preserved when generating the new trajectories.

14.2.4 An Example of Use

Having presented the elements of the proposed trajectory query engine, we proceed to present an example of its use. Consider the table of Figure 14.1, where the database holder requires $\mathcal{K} = 6$ anonymity. Initially, it consists only of real trajectories. A user poses a range query to the database, requesting all trajectories that lie within a given spatiotemporal region. Assume that from the whole database, only trajectories $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ satisfy the query. As a result, the query engine has to create three fake records as part of the answer set that will be returned to the inquirer. First, three records are created in the database, each having

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

a unique ID and attribute `Dummy` set to `Y`. Second, for each numerical attribute, the query engine applies numerical interpolation to set its value. Attribute `Age` has a mean m of 33 and a deviation of $[min - m, max - m] = [-7, 5]$ around the mean. Since we have three fake records, one will be assigned the mean value (record **h**) and for the other two we need to select two values $\pm d \in [-7, 5]$. Suppose that we select $d = \pm 2$. Then, one record will have `Age = 31` (record **f**) and the other record `Age = 35` (record **g**). The exact same process is followed for variable `Income`. The pairs of fake records are randomly formulated prior to each numerical attribute specification.

After assigning values to all numerical attributes, the query engine has to create dummy trajectories to be associated to the fake records. For clarity purposes, in Figure 14.3, we present the generation process for one dummy, considered to be assigned to record **f**. We assume that at one iteration the trajectory interpolation algorithm chooses trajectories **b** and **c** as the drivers for the generation of dummy trajectory **f**. By sweeping the **x**–axis, the location updates from these two trajectories are collected in the sequence from 1 to 10 (see Figure 14.3). At each stop of the sweeping line (e.g., 1, 2,...), the algorithm computes the mean value in the **y**–axis for the two driver trajectories and generates a "location update" (x, y_{avg}, t) for the dummy trajectory. At the end, the algorithm applies the same reconstruction strategy that was used for the real trajectories of the dataset, to decide how consecutive location updates will be connected to each other. Once created, trajectory **f** is added both to the database (for coherence purposes) and to the current answer set. As a final comment, notice that the generated dummy trajectories can be chosen at following iterations of the trajectory interpolation algorithm as drivers for the construction of new dummies. However, we do not allow the same pair of drivers to be used more than once.

14.2.5 Handling of Attacks

In what follows, we motivate the two types of attack that a malevolent user has at his/her disposal to compromise the dataset, and we provide a methodology that aims at shielding the proposed query engine against such attacks.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

User Identification Attack

In the user identification attack, the snooper creates a set of queries that overlap in the set of attribute–value pairs in an attempt to increase his/her confidence regarding the real identity of a user. A set of approaches for the handling of user identification in statistical databases in presented in [10]. In this work we are interested in the formulation of this attack by means of queries that expose trajectory data. Consider the range query of Figure 14.2(i) that returns the trajectories of region **A**. By assuming 4–anonymity, the query engine produces a dummy trajectory **p** that is the result of interpolation on trajectories **o** and **q**. The real trajectories **o**, **q** and **r** along with the dummy trajectory **p** satisfy 4–anonymity. Now, suppose that a snooper wants to identify whether trajectory **o** is real or dummy. To do so, he/she selects a region **B** that is part of the previous region **A** and contains trajectory **o**. Then, the snooper poses a query to the engine, requesting the available trajectories in region **B**. If the query engine answers this query, it will return trajectory **o** and part of trajectory **p**, along with two dummy trajectories for the offering of 4–anonymity in region **B**. As a result, and since the inquirer knows the trajectories of region **A**, his/her confidence regarding trajectory **o** will dramatically increase from $1/4$ (as of region **A**) to $1/2$ (due to region **B**). It is evident that such an attack can breach the anonymity model that is enforced by the query engine.

To prohibit the user identification attack, we take advantage of the auditing mechanism [10, 92], as applied to relational data, and transform it to operate on spatiotemporal data. In auditing, the system holds up–to–date logs of all the queries initiated by each user and checks for a compromise, whenever a new query is issued by the user. In our case, we need to keep track of the spatiotemporal regions containing the trajectories that formulate the answer set to each query. Then, whenever the user poses a new query, the query engine scans the user history to examine if the spatiotemporal area in the new query overlaps with the area of a previous query. In this case, the system denies answering the query. Otherwise, the system proceeds to answer the query and subsequently stores the corresponding spatiotemporal area in the user history. The area to be stored is the MBR of the trajectories that appear in the answer set (including the dummies). Furthermore, all computed MBRs

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

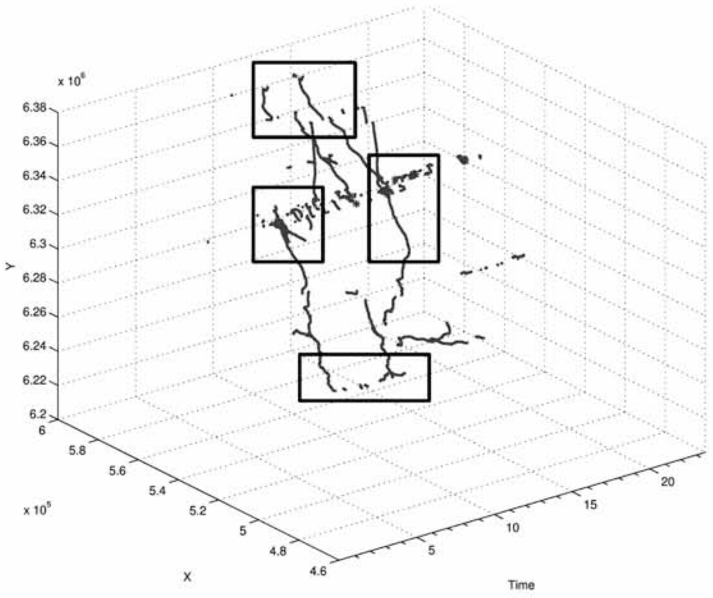


Figure 14.5: Spatiotemporal auditing to handle user identification.

are appropriately indexed to allow for efficient retrieval. For a set of possible indexing schemes (e.g., STR–tree and TB–tree), one can consult [119]. An example of the operation of this approach appears in Figure 14.5, where four MBRs capture the areas of denial for subsequent queries issued by a specific user.

An extension of the auditing mechanism is applied on trajectories that belong to easily identifiable people, such as Bill Gates or Lucky Luck! Specifically, to avoid providing trajectory information for individuals who can become targets of attack, our approach denies answering queries in areas that contain VIP's trajectories.

Sequential Tracking Attack

In the sequential tracking attack, the snooper attempts to "follow" a user trajectory in the database, through a set of focused queries on spatiotemporal regions that are adjacent[3] to each other. If no special action is taken on behalf of the query engine then the attacker will

[3]By adjacency we mean that the corresponding MBR enclosures of the two regions either touch or are within a short distance from each other.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

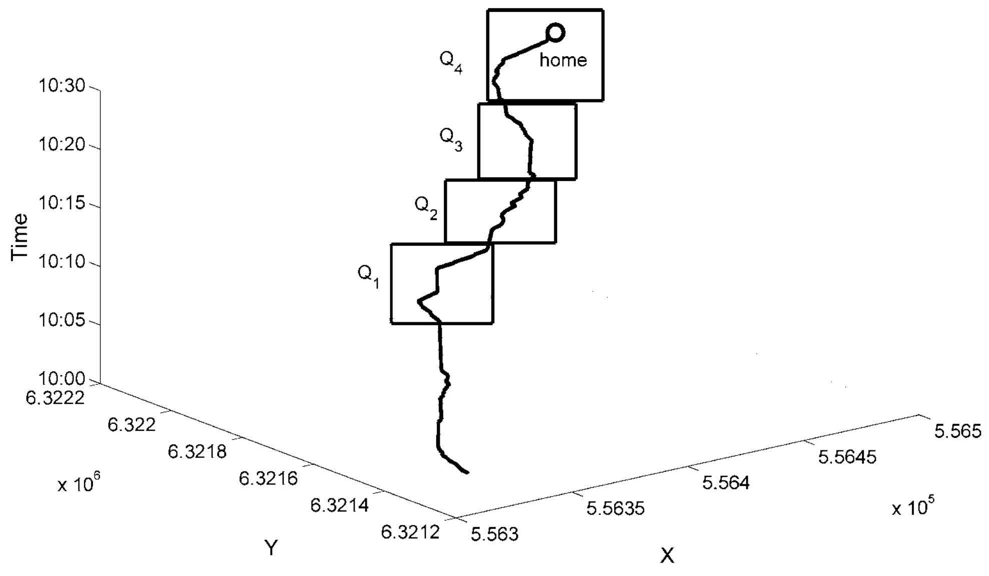


Figure 14.6: Spatiotemporal auditing to handle sequential tracking.

be able to (i) increase his/her confidence regarding the real and the dummy trajectories that were provided as answers of past queries, and (ii) be certain that the trajectory that he/she currently "follows" belongs to a real user and is not dummy.

Consider the range queries Q1–Q4 of Figure 14.6. When a snooper poses query Q1 to the database, the query engine will return the trajectory shown in the MBR, along with $\mathcal{K}$–1 other trajectories in order to satisfy the requirements of $\mathcal{K}$–anonymity. Then, assume that the snooper poses query Q2, requesting the trajectories that exist in a region that is adjacent (here touches) to that of Q1. The real trajectory that was part of the answer set of Q1 will also appear in Q2 and the depicted movement will continue smoothly, as is shown on Figure 14.6. To satisfy $\mathcal{K}$–anonymity in the region of Q2, the query engine will retrieve any other real trajectories in this region and (possibly) generate some dummies. However a naïve dummy generation algorithm, will produce dummies by taking into consideration only the region of Q2 and not that of Q1. As a result, the snooper will easily spot the dummies among the real trajectories and will also be certain that the trajectory that he/she

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

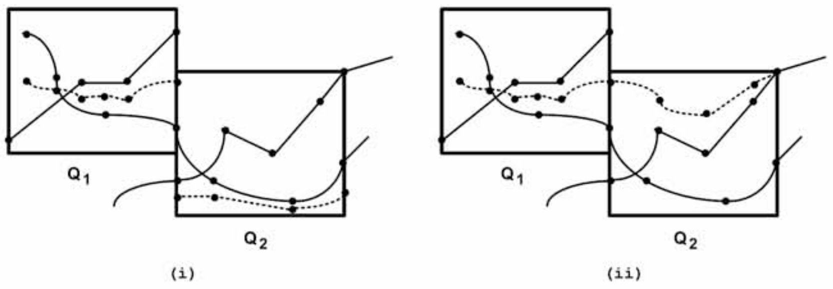


Figure 14.7: Defending the database from sequential tracking.

"follows" is real; no dummy trajectory spans from the region of Q1 to that of Q2. The same situation holds throughout the whole route of the user from the MBR of Q1 to that of Q4 and, as a result, the snooper succeeds in tracking a real user to his/her destination (i.e. in the presented example to his/her home, as indicated in the MBR of Q4).

The situation we just described is captured in Figure 14.7(i), where 3–anonymity is considered. The trajectories appearing with solid lines are real, while those with dotted lines are dummies. As one can observe, the answer of Q2 uncovers the dummy trajectory in Q1, since the "user" movement in Q1 abruptly discontinues in Q2. As a result, the confidence of a snooper regarding the real trajectories in Q1 when he/she receives the answer corresponding to query Q2, increases from $1/3$ to $1/2$. Furthermore, if the user has knowledge of the dummy generation scheme that is used, then he/she can be confident that the sole trajectory that spans from the region of Q1 to that of Q2 is real. Thus, by continuing the sequential tracking attack, the snooper effectively achieves to track down a real user appearing in the dataset to his/her destination.

To prohibit sequential tracking, we use auditing (as before) to identify queries to regions that lie in the spatiotemporal neighborhood of past queries. If such a query is posed, we follow the approach shown in Figure 14.7(ii). Specifically, our proposed algorithm aims at confusing the snooper regarding the real and the dummy trajectories by continuing the movement of dummies in adjacent regions. Although alternative solutions may be devised to tackle this attack, we consider the proposed approach to be both simple and effective.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 14.1 Computation of count/aggregate non-trajectory queries.

```
function COUNT-QUERY($cond$, $K$)
    $conn \leftarrow$ getConnection(string)                                          ▷ Connect to DB
    $stmt \leftarrow conn$.createStatement()
    $rset \leftarrow stmt$.executeQuery(“SELECT count(*) FROM data WHERE $cond$)
    $ans \leftarrow rset$.getValue(1)
    if $ans \geq K$ then return $ans$
    else
        raise_exception(“$K$-anonymity violation”)
```

```
function AGGREGATE-QUERY($func$, $attr$, $cond$, $K$)
    $conn \leftarrow$ getConnection(string)                                          ▷ Connect to DB
    $stmt \leftarrow conn$.createStatement()
    $rset \leftarrow stmt$.executeQuery(“SELECT count(*), $func$($attr$) FROM data WHERE $cond$”)
    $ans \leftarrow rset$.getValue(1)
    if $ans \geq K$ then return $rset$.getValue(2)
    else
        raise_exception(“$K$-anonymity violation”)
```

First, the confidence of the snooper regarding the real trajectories in Q1 remains the same, since the existence of a dummy trajectory cannot be inferred. Second, due to the dummy generation algorithm that designates the behavior of the dummies, the snooper cannot be confident that the trajectory he/she follows belongs to a real user and is not dummy. Thus, given an adequate value of $\mathcal{K}$ in $\mathcal{K}$–anonymity, the snooper will be discouraged to perform sequential tracking since the odds of “following” a real user will be considerably low.

14.3 The Supporting Algorithms

The methodology presented in Section 14.2 is implemented as a set of algorithms, utilized by the query engine to undertake the necessary tasks for privacy preservation. In what follows, we explain the operation of these algorithms and shed light on their key steps.

Algorithm 14.1 collects functions that operate on queries for non–trajectory data. Such queries are handled differently based on whether they are count queries or queries for aggregate statistics. In the first case, the query engine directly services the user query. In the second case, it augments the SELECT statement with a counter of the returned records, prior to servicing the query. Furthermore, in aggregate queries, *func(attr)* signifies the use of an SQL aggregate function, such as `AVG`, `SUM`, `MAX` or `MIN`, applied on attribute *attr* of the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 14.2 Computation of range/distance/k–NN trajectory queries.

```
function RANGE-QUERY(cond, Sgeo, $t_{start}$, $t_{end}$, K)
    if IN_HIST(Sgeo, $t_{start}$, $t_{end}$) then
        raise_exception (“Privacy threat”)
    conn ← getConnection(string)                              ▷ Connect to DB
    cstmt ← conn.prepareCall(RANGE_QR(Sgeo, $t_{start}$, $t_{end}$))
    cstmt.execute()                                           ▷ Execute PL/SQL routine
    if #recs(T) $\geq K$ then
        return T.view(cond)
    FAKE-GEN(K−#recs(T), Sgeo)
    UPDATE_HIST(Sgeo, $t_{start}$, $t_{end}$)
    return T.view(cond)
```

```
function kNN-QUERY(cond,P,$t_{start}$, $t_{end}$, k, K)
    create temporary table tmp like data
    tmp.movement ← user trajectories in [$t_{start}$, $t_{end}$]
    INSERT INTO T (SELECT * FROM tmp WHERE NN(tmp.f_trajectory(), P) = ‘true’ AND rownum $\leq k$)
    if IN_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$) then
        raise_exception (“Privacy threat”)
    if #recs(T) $\geq K$ then
        return T.view(cond)
    FAKE-GEN(K−#recs(T), mbr(aggr(T.f_trajectory())))
    UPDATE_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$)
    return T.view(cond)
```

```
function DISTANCE-QUERY(cond, $X_p$, $Y_p$, d, $t_{start}$, $t_{end}$, K)
    Sgeo ← create_circle ($X_p$, $Y_p$, d)
    RANGE-QUERY(cond, Sgeo, $t_{start}$, $t_{end}$, K)
```

```
PL/SQL procedure RANGE_QR(Sgeo, $t_{start}$, $t_{end}$)
    DECLARE
    CURSOR mobjs IS SELECT movement FROM data
    mo, mlim   Moving_Object
    geom, g    Geometry
    BEGIN
    open (mobjs)
    loop
        fetch mobjs INTO mo
        exit when mobjs%NOTFOUND
        mlim ← mo.at_period($t_{start}$, $t_{end}$)
        geom ← mlim.f_trajectory()
        g ← intersect(geom, Sgeo)
        if g $\neq$ null then
            INSERT INTO T VALUES (. . . , intersect(mo, g))
    close(mobjs)
    END
```

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

data table. In both cases, the WHERE clause in the SELECT statement copies the condition *cond* from the user query. After servicing the query, the query engine computes the number of records from the database that were involved in the answer. It then proceeds to disclose the answer to the user, only if it satisfies the requirements of $\mathcal{K}$–anonymity.

Algorithm 14.2 contains functions that handle range and nearest neighbors queries. From the algorithmic standpoint, we make a slight distinction between distance queries (e.g., **Q4**) and range queries (e.g., **Q2**). Specifically, distance queries are transformed into range equivalents by constructing the range of interest as a circle centered at the provided location **P** and having a radius of *d* (requested distance). To service a range query, the algorithm first ensures that it involves a spatial/temporal or spatiotemporal region that blocks both the user identification and the sequential tracking attacks. If this so happens, then with the aid of a PL/SQL procedure, the algorithm captures the movement of all users in the selected region and stores it in a table **T** (similar to the one that holds the data). Table **T** collects all user records having trajectories in the selected region, along with the user (sub)–trajectories in this region. Then, the algorithm counts the number of records in table **T** to identify if they suffice for the requirements of $\mathcal{K}$–anonymity. If this is the case, then the algorithm returns the user requested view of the results (line 8). Function T.view(*cond*) corresponds to a SELECT query in **T** using condition *cond* in the WHERE clause. On the other hand, if the retrieved records in table **T** are insufficient for $\mathcal{K}$–anonymity, the algorithm requests the generation of the needed fakes and subsequently updates the history of user queries to include the answer region of the serviced query. This update is necessary to block any future attempts of the user to compromise the dataset.

Queries requesting the *k* nearest trajectories to a given point **P**, are handled similarly to the range queries. The *k*-NN–Query function, presented in Algorithm 14.2, uses a temporary table **tmp** to store (apart from all non–trajectory attributes), all user trajectories in the specified time interval (lines 2–3). By using the information in this table, the algorithm identifies the *k* nearest trajectories to point **P** and stores the corresponding records to table **T**. The user history of queries is then examined to identify a potential match in the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 14.3 Computation of landmark(s)/route trajectory queries.

```
1: function LANDMARK-QUERY($cond$, $t_{start}$, $t_{end}$, $K$, $geom = < P_1, P_2, \ldots, P_r >$)
2:     create temporary table tmp like data
3:     tmp.movement $\leftarrow$ user trajectories in [$t_{start}$, $t_{end}$]
4:     INSERT INTO T (SELECT * FROM tmp WHERE
         Contains(tmp.f_trajectory(), $geom.P_1$) = 'true' AND Contains(tmp.f_trajectory(), $geom.P_2$) = 'true' AND ... AND
         Contains(tmp.f_trajectory(), $geom.P_r$) = 'true')
5:     if IN_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$) then
6:         raise_exception ("Privacy threat")
7:     if #recs(T) $\geq K$ then
8:         return T.view($cond$)
9:     FAKE-GEN($K-$#recs(T), mbr(aggr(T.f_trajectory())))
10:    UPDATE_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$)
11:    return T.view($cond$)
```

```
1: function ROUTE-QUERY($cond$, $t_{start}$, $t_{end}$, $K$, $R$)
2:     create temporary table tmp like data
3:     tmp.movement $\leftarrow$ user trajectories in [$t_{start}$, $t_{end}$]
4:     INSERT INTO T (SELECT * FROM tmp WHERE Contains(tmp.f_trajectory(), $R$) = 'true')
5:     if IN_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$) then
6:         raise_exception ("Privacy threat")
7:     if #recs(T) $\geq K$ then
8:         return T.view($cond$)
9:     FAKE-GEN($K-$#recs(T), mbr(aggr(T.f_trajectory())))
10:    UPDATE_HIST(mbr(aggr(T.f_trajectory())), $t_{start}$, $t_{end}$)
11:    return T.view($cond$)
```

spatiotemporal region, where the spatial dimension is the MBR of the (sub)–trajectories of all k nearest neighbors of **P**. If the query is found to be "safe" with respect to the two types of attack, the algorithm checks if it fulfils the requirements of $\mathcal{K}$–anonymity. If it does, then the query is answered and the user history of queries is updated. On the other hand, if less than $\mathcal{K}$ records exist in table **T**, then the algorithm generates all the necessary fakes in the computed spatiotemporal region.

Algorithm 14.3 collects the necessary functions for the handling of landmark and route queries. The followed approach is similar to the k-NN–Query algorithm. First, a temporary table **tmp** stores all the records in the database, involving user (sub)–trajectories in the specified time period. Then, a table **T**, similar to **tmp**, is populated with all the records from **tmp** that contain either trajectories that pass through a number of pre–specified points **P** (landmark query), or trajectories that follow a pre–specified route R (route query). Finally, table **T** is augmented (if necessary) with fake records, the user history of queries is updated and the $\mathcal{K}$–anonymous query answer is returned to the user.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 14.4 Search/Update the user history of queries.

```
1: function IN-HIST($Sgeo$, $t_{start}$, $t_{end}$)
2:     $conn \leftarrow$ getConnection(string)                                    ▷ Connect to DB
3:     $stmt \leftarrow conn$.createStatement()
4:     $rset \leftarrow stmt$.executeQuery(“SELECT count(*) FROM hist WHERE ($t_{end} \geq t_A \wedge t_{start} \leq t_B$) $\wedge$
                          (Overlaps(geom, $Sgeo$) = ‘true’ $\vee$ Neighbors(geom, $Sgeo$) = ‘true’ ”)
5:     if $rset$.getValue(1) = 0 then return true
6:     return false

1: Procedure UPDATE-HIST($Sgeo$, $t_{start}$, $t_{end}$)
2:     $conn \leftarrow$ getConnection(string)                                    ▷ Connect to DB
3:     $stmt \leftarrow conn$.createStatement()
4:     $rset \leftarrow stmt$.executeUpdate(“INSERT INTO hist VALUES ($Sgeo$, $t_{start}$, $t_{end}$) ”)
```

Table `hist`:

geom	t_A	t_B

Algorithm 14.4 presents functions that operate on the user history of queries. The query history of a user is captured in a table **`hist`** that contains a spatial attribute *geom*, storing the spatial region of the user query, and two timestamp variables $\mathbf{t_A}$ and $\mathbf{t_B}$, storing the temporal interval of the query. An update operation to the user history of queries is equivalent to the insertion of a record to table **`hist`**, indicating the spatiotemporal region of the query. On the other hand, the check for a potential attack is performed by identifying the existence of any record in **`hist`** that involves a spatiotemporal region that either overlaps with the spatiotemporal area of the user query, or it neighbors with it. Both cases indicate a potential privacy violation and result in the denial of the query engine to answer the query.

Algorithm 14.5 is responsible for the generation of the fake records. First, the algorithm generates two temporary dummies, **u** and **v** (see Figure 14.3), to make up for the uppermost and the lowermost trajectories in the provided spatial region *Sgeo* (lines 2–4). The dummy trajectories are subsequently included into empty records T_u and T_v and inserted into table **`T`** (lines 5–6). Then, the algorithm generates the required number of fake records and sets the values in all their numerical attributes (lines 7–19). Specifically, for each numerical attribute, the algorithm uses the real records in **`T`** to compute its mean and deviations. Following that, the algorithm performs a random partitioning of the fake records into pairs and for each pair it identifies a value near the attribute mean and uses this value to perform the attribute–value assignment. After assigning values to all the numerical at-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 14.5 Generation of M fake records in spatial region *Sgeo*.

```
1:  procedure FAKE-GEN($M$, $Sgeo$)
2:      compute $y_{min}, y_{max}$ in $Sgeo$
3:      compute $t_{min}, t_{max}$ in T.movement
4:      generate temporary dummies $\mathbf{u}(t) = y_{max}$, $\mathbf{v}(t) = y_{min}$, where $t \in [t_{min}, t_{max}]$ ($\mathbf{u}$,$\mathbf{v}$ have constant movement)
5:      generate empty records $T_u$, $T_v$ and associate with $\mathbf{u}$,$\mathbf{v}$
6:      insert $T_u$, $T_v$ into table T
7:      generate $M$ empty fake records $f_1 \ldots f_M$
8:      if M%2 = 1 then
9:          select $f_1$
10:         for each numerical attribute $\alpha \in A$ do
11:             $f_1.\alpha \leftarrow avg(\alpha, T)$                    ▷ avg over the real records
12:     for each numerical attribute $\alpha \in A$ do
13:         $d_{min} \leftarrow avg(\alpha, T) - min(\alpha, T)$
14:         $d_{max} \leftarrow max(\alpha, T) - avg(\alpha, T)$
15:         randomly group fake records into pairs $p_1 \ldots p_{M/2}$
                                                                      ▷ exclude $f_1$ if $M$ is odd
16:         for each pair $p_i(f_j, f_k) \in [p_1 \ldots p_{M/2}]$ do
17:             $d \leftarrow$ random value in $[0, min(d_{min}, d_{max})]$
18:             $f_j.\alpha \leftarrow avg(\alpha, T) - d$
19:             $f_k.\alpha \leftarrow avg(\alpha, T) + d$
20:     for each fake record $f_j \in [f_1 \ldots f_M]$ do
21:         randomly select two records $T_p, T_q$ from table T
22:         list $L \leftarrow sort_{\mathbf{x}-coord}$ (location-updates($T_p \cup T_q$))
23:         for each location update $lu \in L$ do
24:             if $lu = (x_p, y_p, t_p) \in T_p$ then
25:                 $lu' \leftarrow (x_p, T_q.\text{movement}(x_p)/2, t_p)$
26:             else if $lu = (x_q, y_q, t_q) \in T_q$ then
27:                 $lu' \leftarrow (x_q, T_p.\text{movement}(x_q)/2, t_q)$
28:             insert (list dummy-lu, $lu'$)
29:         for all consecutive $lu_1, lu_2 \in$ dummy-lu do
30:             update ($f_j$.movement, MRECON(rfunc, $lu_1$, $lu_2$))
31:         insert fake record $f_j$ to T
32:     remove $T_u$, $T_v$ from table T
```

tributes in the fake records, the algorithm proceeds to the generation of trajectory dummies (lines 20–31). To accomplish that, for each fake record it randomly selects two records from table **`T`** and orders the corresponding trajectories in an ascending order of their location updates (lines 21–22) with respect to the **x**–coordinate. Then, for each location update, the algorithm identifies the corresponding point in the other trajectory that has the same **x**–coordinate and computes a "dummy" location update as the mean of the two real ones (with respect to the **y**–coordinate). This "dummy" location update is kept in a list **`dummy-lu`** (line 28). After collecting all the "dummy" location updates, the algorithm proceeds to reconstruct the movement of the dummy by selecting consecutive location updates from **`dummy-lu`** and applying a trajectory reconstruction function *rfunc* (e.g., like the ones of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Figure 14.4). As a final step, the reconstructed dummy trajectory is assigned to the corresponding fake record and included in table **T** to be a candidate for the next iteration of the algorithm. The same operation holds for the generation of all the necessary dummies, their assignment to fake records and subsequent inclusion in table **T**. After producing all the fake records, the algorithm removes from table **T** the temporary fakes T_u and T_v.

14.4 Query Extensions

The query formulation mechanism (presented in Section 14.2.2), suffices for laying down queries involving specific landmarks (e.g., **Q1** or **Q6**), or spatial areas of a common granularity (e.g., **Q4**, **Q5**). Furthermore, it effectively handles queries involving temporal data in the form of specific times or time intervals during a given calendar day (e.g., **Q2**). Both these features, when combined together, allow the specification of spatiotemporal queries at a fine granularity (e.g., **Q3**, **Q7** or **Q9**). However, there are many occasions when the inquirer wants to retrieve knowledge at different levels of granularity, possibly coarser than the one of the previous queries. For example, a user might want to pose queries like:

Q10 "Find all user trajectories this Monday".

Q11 "Find all user trajectories in Minneapolis, MN".

Q12 "Find all users that moved on January 2008 in the city center of Minneapolis, MN".

Answers to those queries allow for an advanced statistical analysis of the available movement data and can be proved valuable to draw conclusions regarding the periodicity of certain events. To support the specification of such queries, we introduce two concept hierarchies that depict space and time at different levels of granularity (see Figure 14.8). By using the available levels of granularity, the query engine can effectively handle queries that skip one or more levels of the spatial/temporal hierarchy by first computing the needed aggregations and performing the required materializations in the database. For example, to answer query **Q12** the query engine will first retrieve the region that corresponds to the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

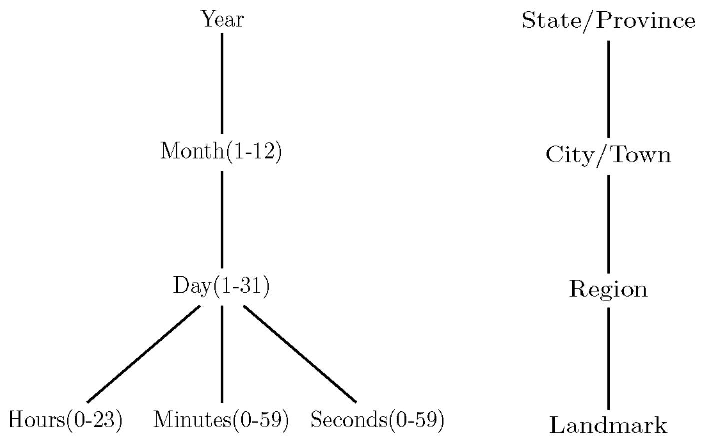


Figure 14.8: Query specification through time/space hierarchies.

city center of Minneapolis by performing an aggregation over all the appropriate streets and landmarks that belong to the city center. Then, it will acquire all the user trajectories that pass through the computed region. Finally, it will remove from the answer set all the trajectories with timestamps that indicate a month other than January and a year other than 2008. Of course, the necessary computations can take place in a different sequence and still lead to the same result. As one can observe, the organization and the distribution of the trajectory data are the major criteria to identify which, among the possible sequences of computations, is bound to perform best.

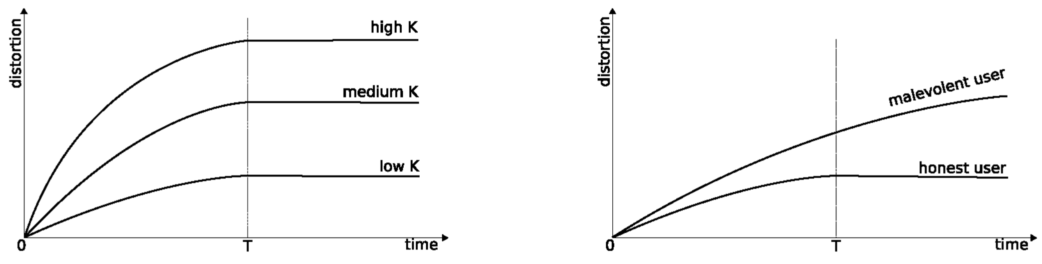
14.5 Qualitative Analysis

In this section, we provide a qualitative analysis of our proposed approach along three principal directions; database distortion, blocking of attacks and computational complexity.

14.5.1 Database Distortion and Blocking of Attacks

Figure 14.9 presents the distortion of the database caused by the incorporation of fakes. The distortion is measured as the percentage of fake records in the database at any given time. Figure 14.9(a) plots the distortion of the database with respect to the posed user queries (in

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Distortion w.r.t. the value of $\mathcal{K}$.

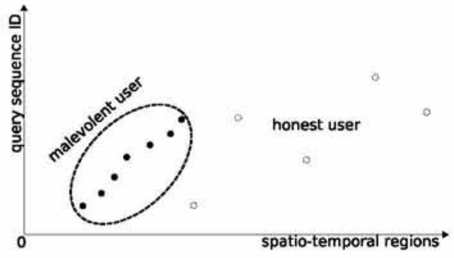
(b) Distortion w.r.t. the user requests.

Figure 14.9: Database distortion due to the incorporation of fake records.

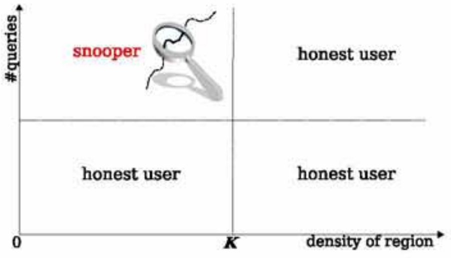
time), for different requirements of anonymity. As expected, lower values of $\mathcal{K}$ result in a smaller distortion of the database, since less fakes are needed to meet the requirements of $\mathcal{K}$–anonymity. Furthermore, for any value of $\mathcal{K}$ there exists some time instance T at which the fakes in the dataset suffice to answer a reasonable amount of user queries, without the need of generating additional fakes. From this point on, the database is typically distorted at a much lower rate than previously. Figure 14.9(b) plots the distortion that is inflicted in the database by two users who are assigned the same $\mathcal{K}$ in anonymity. As one can observe, a significant difference in the distortion of the database that is caused by a user (with respect to the rest of the population) may suggest the user's attempt to compromise the database. This is due to the fact that snoopers try to spot subjects and thus pose a large series of focused queries, leading to more fakes being generated by the query engine.

Figure 14.10 presents two mechanisms to distinguish honest from malevolent users. Figure 14.10(a) is based on the premises that snoopers initiate sequences of trajectory queries involving small spatiotemporal regions that are close to each other. For each user, we generate a plot capturing the spatiotemporal regions in the user history of queries, ordered by proximity to each other (**x**–axis) versus the sequential numbers that indicate the order of queries in the user history (**y**–axis). Then, points that lie close to each other in the plane correspond to sequential user queries involving neighboring regions. Such a behavior is typical for snoopers, while atypical for honest users. Another good indicator of malevolent behavior is the existence of numerous user queries in sparsely populated regions

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Detection of the user tracking attack.



(b) Identification of suspicious behavior.

Figure 14.10: Identifying potential attacks based on the history of user queries.

(Figure 14.10(b)). By using graphs such as the ones presented above, the administrators of the system are capable of blocking most attempts of snoopers to compromise the database.

14.5.2 Computational Complexity

The queries, supported by the query engine, experience different requirements in terms of computational cost. Count and aggregate queries require $\Theta(d)$ runtime, where d is the number of records in the *data* table. Range and distance queries operate similarly. They are both based on the RANGE_QR procedure that relies on an R–tree, experimentally proven to have polylogarithmic behavior. On average, RANGE_QR requires $O(d\ (polylog\ d) + o)$, where *polylog* is applied on the number of trajectories on the *data* table and o is the number of elements returned by the R–tree[4]. In k–NN queries the selection of the user trajectories at the requested time period requires $O(log\ d + o)$ (worst case), given a B–tree index on the timestamp data. The actual identification of the nearest neighbors requires on average $O(s + log\ d)$, where s is the number of trajectories in this time period. The same is the cost for the trajectories selection process in landmarks/route queries. All the aforementioned types of queries, with the exception of the count/aggregate queries, have an additional cost of searching and updating the user history. A search in the user history of queries incurs $O(polylog\ h + o)$ cost, where h is the size of the user history and o is the number of elements

[4]Searching in R–trees/B–trees has an additional multiplicative constant cost that depends on the page size of the underlying index.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

returned by the R–tree index. On the other hand, the cost of updating the user history is in the worst case $O(log\ h)$ due to the insertion of the corresponding region to the R–tree. Finally, the cost of fake records generation equals the cost of creating M empty records, that is $O(M)$, updating their values for the A numerical attributes, that is $O(A\ (T+M))$ (T is the number of trajectories in the *Sgeo* region), and producing the M dummy trajectories, that is $O(M\ L\ log\ L + M\ f\ L + M\ log\ T)$, where L is the number of location updates in $T_p \cup T_q$ and f is the cost of deciding whether a point lies in a given trajectory. The presented analysis is based on empirical estimates of the runtime costs, since we have no access to the implementation details of Oracle.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 15

Summary and Future Work

In this part of the dissertation, we surveyed the existing so far methodologies for the offering of privacy in the publication and the mining of databases containing sensitive mobility data. Then, we presented a privacy aware trajectory tracking query engine that offers $\mathcal{K}$–anonymous answers to user queries. Our proposed engine allows the data to stay in–house and supports a large variety of query types, involving both trajectory and non–trajectory data. To offer $\mathcal{K}$–anonymity in trajectory queries, the engine relies on the generation of fake records involving dummy trajectories. The query engine offers strict guarantees about what can be found by untrusted third parties and is capable of blocking two types of potential privacy breaches, namely *user identification* and *sequential tracking*. Through qualitative analysis, we demonstrated the effectiveness of our proposed approach towards preventing malevolent users from compromising the dataset, while causing small distortion.

Spatiotemporal datasets present a new challenge to the privacy preserving data mining community due to their spatial and temporal characteristics. Few approaches have been proposed so far that achieve to address some of the special requirements of this type of data. A basic drawback of most of the existing methodologies is that they fail to treat space and time equally well. Instead, the vast majority of the approaches that have been proposed so far put their effort on the adequate treatment of either the spatial or the temporal dimension of the data, but not both. As a result, user mobility data is often transformed into sequential

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

data, where the spatial component is reduced to a set of places of interests (events) and the time component (apart from providing the total ordering of these events in the sequence) is disregarded. Thus, we feel that there is currently ample room for improvement in this interesting and very challenging area of research.

As presented earlier, privacy preserving data mining in the context of mobility data has been investigated towards three broad research directions: (i) data perturbation approaches, (ii) distributed (SMC) privacy aware approaches, and (iii) knowledge hiding approaches. Based on the number of published works per direction, it becomes evident that most of the research effort has been placed towards the development of data perturbation methodologies, while few approaches have been devised to support the other two directions. We believe that in the upcoming years, data mining researchers will put more effort in devising novel algorithms for the hiding of user mobility patterns, especially due to the urging need of these methodologies in various application contexts. The hiding of sensitive mobility patterns imposes far greater challenges than traditional knowledge hiding, since specially crafted algorithms are necessary to identify all the important correlations that exist within the datasets. Furthermore, the mining of sensitive knowledge, depicted in the form of associations in mobility datasets, may allow for the use of different measures of pattern interestingness than the commonly employed support and confidence metrics. As an effect, new knowledge hiding techniques will have to be investigated in the upcoming years that will successfully conceal this novel type of sensitive knowledge.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Part III

Trajectory Privacy in Location Based Services

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 16

Introduction

The enormous advances in positioning technologies like GPS, GSM, UMTS and RFID, along with the rapid developments in the wireless communications industry, have made possible the accurate tracking of user location at a low cost [41]. As an example, the differential GPS technology that is currently widely adopted offers a tracking accuracy of less than 3 feet (i.e. less than 1 meter!) [21]. This increased accuracy gave rise to a novel class of applications which are based on user location, spanning from emergency response and search & rescue services (such as E–911), to services that automate everyday tasks, such as online user navigation to avoid traffic jams and/or bad weather conditions, way–finding, store–finding and friend–finding, as well as mobile commerce and surveying. All these services, including localized news and state–of–the–art location–based games that merge physical and virtual spaces, require an extensive use of location data and are collectively known as *Location Based Services* (LBSs).

The benefit of LBSs both to the individual subscribers and to the community, as a whole, is undeniable. With respect to the public welfare, the collection of location data by a governmental or other public agency may enhance the process of decision making regarding tasks such as urban planning, routing, wildlife rescuing and environmental pollution. As is evident, the new computing paradigm is changing the way people live and work. However, it also poses a series of challenges as it touches upon delicate privacy issues [26]. In the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

following sections, we discuss the existing privacy challenges in LBSs and we motivate the necessity of algorithms that guarantee the privacy of the users when requesting LBSs. Following that, we present two privacy models (a centralized and a decentralized one) which collect the existing research methodologies for the offering of privacy in LBSs, we discuss the scenario of operation in each of these two models, as well as the working assumptions that are employed by most of the state–of–the–art privacy approaches. Finally, we present the organization of the rest of the chapters in this part of the dissertation.

16.1 Privacy Challenges in LBSs

The offering of LBSs requires an in–depth knowledge of the subscribers' whereabouts. Thus, with untrustworthy service providers and/or insecure communication channels, the deployment of LBSs may breach the privacy of the mobile users. Consider, for example, a service request originating from the house of a user. The request contains sufficient information to identify the requester, even if it lacks of any other identification data (e.g., the user id, the user name, etc). This is true since the mapping of the exact coordinates that are part of the user request to a publicly available data source of geocoding information (e.g., white pages), can reveal that the request originated from a house and thus increase the confidence of the service provider (or any untrusted entity) that the requester is a member of the household. Moreover, if a series of requests for LBSs are matched to the same individual then it is possible for the service provider to identify places that this user frequently visits, reveal his/her personal habits, political/religious affiliations or alternative lifestyles, as well as build a complete profile of the user based on the history of his/her movement in the system. Furthermore, users may be reluctant to ask innocuous queries such as “find the restaurants in my vicinity” since, once their identity is revealed, they may face unsolicited advertisements. In general, there exist two types of privacy threats in LBSs [39, 179]: (i) privacy threats related to the disclosure of the location from where a request for an LBS was initiated, and (ii) privacy threats related to the disclosure of the actual query that was posed by the user as part of the user request (e.g., “find the nearest clinic to my current location”).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

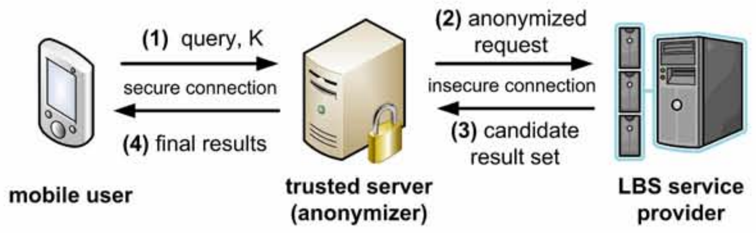


Figure 16.1: The centralized model for privacy in LBSs.

As it becomes evident, without the existence of strict safeguards, the deployment of LBSs and the sharing of location information may easily lead the way to an abuse scenario, similar to Orwell's Big Brother society [16]. To avoid this situation and adequately protect the privacy of the users when requesting LBSs, sophisticated algorithms have to be devised.

16.2 Overview of the Privacy Models

The existing approaches for the offering of privacy in LBSs operate within the regions of a privacy model which defines the different components (and the stakeholders), as well as the role of each of these components in the system. Pertinent to the role of a component is the information that it is entitled to access, the way it can manipulate this information, and the parties to which it can communicate the results of this processing. In a sense, a privacy model provides an abstraction of the reality as it is based on a number of assumptions regarding the capabilities of each component in the system. In what follows, we present the two dominant privacy models that are employed by the majority of the existing methodologies for privacy in LBSs. After presenting the two models, we discuss their assumptions, and we shed light on the way that they operate.

16.2.1 The Centralized (Trusted Server) Privacy Model

Figure 16.1 presents a big picture of the centralized model for the offering of privacy in LBSs. In this model, we consider a population of users who are supported by some telecommunication infrastructure, owned by a telecom operator (e.g., T–mobile, Vodafone, etc).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Every user in the system has a mobile device that periodically transmits a location update to some traffic monitoring system residing in a trusted server of the telecom operator. The communicated location update contains the current location (and time) of the user and is stored by the trusted server. A set of LBSs are available to the subscribed users through service providers that collaborate with the telecom operator. We assume that these service providers are untrusted; if a user submits a request for an LBS directly to the service provider then his/her identity can be revealed and his/her privacy can be compromised. Motivated by this fact, the centralized privacy model requires that every user request for an LBS has to be submitted to a trusted server of the telecom operator via a secure communication channel (i.e. it cannot be intercepted in the air). The role of the trusted server (anonymizer) is to filter the incoming user requests and to produce anonymous counterparts that can be safely forwarded to the (unsafe) service providers in order to be serviced. To produce the anonymous counterpart to an original user request, the trusted server has to incorporate algorithms that (i) remove any obvious identifiers that are part of the user request (e.g., ID, name, etc) and (ii) effectively transform the exact location of request into a spatiotemporal area (a.k.a. the "area of anonymity") that includes a sufficient number of nearby users registered to the system so as to prevent the attacker from locating the actual requester.

An Example of Operation

To better understand the operation of the centralized privacy model, Figure 16.2 presents an example, where Bob asks for the nearest betting office B_i to his current location. This is a typical nearest neighbor query that is commonly met in LBSs. Bob forwards his query (request) Q to the anonymizer. Then the anonymizer, who has knowledge of the current location of each user in the system, identifies 3 users (i.e. Tom, Alice and Maria) who are near Bob and encloses all four users in a region R. Subsequently, instead of sending Bob's location to the LBS provider, the anonymizer sends region R. When the LBS service provider receives region R it computes all the betting offices that can be the nearest neighbor of any point in R. It is important to notice that although the service provider is certain that

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

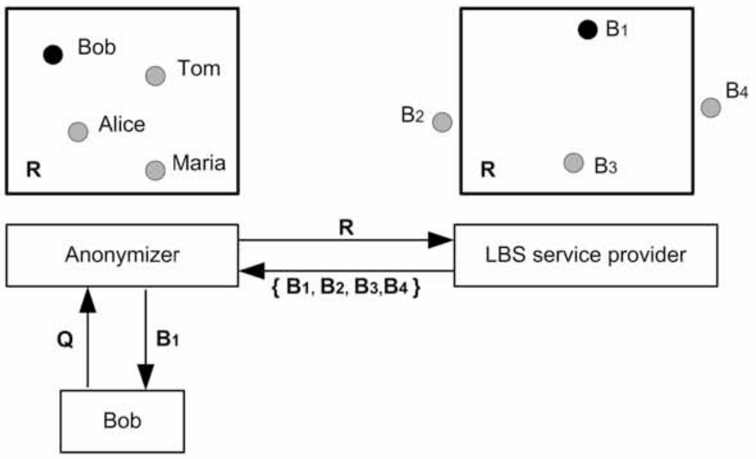


Figure 16.2: A use–case scenario in the regions of the centralized model.

Bob is located within R, it has no other means to identify the exact location of Bob within this region. Furthermore, even if the service provider (or generally an attacker) knew the locations of all the users in R, he/she could only ascertain that the query originated from Bob with a probability of $1/4$ (since there are 4 users inside R). Using its data banks, the service provider generates a candidate set of answers (i.e. $\{B_1, B_2, B_3, B_4\}$) and forwards it to the anonymizer. The anonymizer uses the actual location of Bob inside R to filter out all the false hits and forward the actual nearest neighbor (in this case B_1) to Bob. This step concludes the provision of the requested LBS in a privacy aware manner. The way that region R is formulated, as well as the privacy guarantees that are offered to the requester by the system, are based on the specific privacy methodology that is employed by the trusted server. Chapter 17 discusses a series of methodologies that are based on the centralized model for privacy in LBSs. Furthermore, Chapters 18–23 deliver our contributions in this area, which also rely on the centralized privacy model.

16.2.2 The Distributed (P2P) Privacy Model

Figure 16.3 presents the architecture of the decentralized (P2P) privacy preserving model in LBSs. The general assumptions of the centralized privacy model also hold in the case of the distributed model. In particular, in the distributed privacy model we consider a population of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

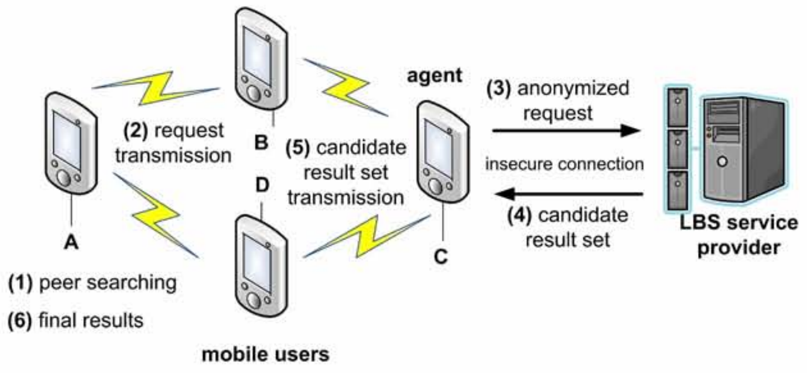


Figure 16.3: The distributed model for privacy in LBSs.

users who are subscribed to a telecom operator through which they can access a set of LBSs. We assume that the providers of the LBSs are not trusted; if a user submits a request for an LBS directly to the service provider then his/her (location) privacy can be compromised. To protect the privacy of the requesters of LBSs, the distributed model requires that the mobile users collaborate with each other to cloak their exact locations into spatial regions prior to submitting the request to the service providers. In general, the operation of the cloaking algorithms, as part of this model, can be divided into three major phases (see [125]):

1. **Peer Searching.** In peer searching the mobile user who wants to request an LBS searches through the telecommunication network for other peers in his/her vicinity.

2. **Location cloaking.** In location cloaking the identified peers form a group with the requester. Then, the requester creates a region R that covers the entire group. This region blurs the exact location of the requester and provides him/her with anonymity.

3. **Query handling.** To prevent an adversary from locating the requester, a mobile user that belongs to the formulated group is randomly selected to act as a representative of the group (a.k.a. an *agent*). The responsibility of the agent is to communicate the request to the LBS provider on behalf of the querying user, to collect the candidate answers from the service provider, and to communicate them back to the requester. Once the requester receives the candidate answers he/she can identify the correct answer simply by using his/her current location.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

An Example of Operation

To better understand the operation of this privacy model, consider Figure 16.3 where user *A* wants to request an LBS in a way that he/she remains indistinguishable among four users. In the first step (peer searching), user *A* has to identify three peers in his/her vicinity. Since he/she has only two neighboring peers, i.e. users *B* and *D*, user *A* has to communicate with mobile user *C* (either through user *B* or *D*) via multi–hop routing. After building the group with users *B*, *C* and *D*, user *A* generates a spatial region *R* that contains him/her along with the other three users. Ideally, within region *R*, user *A* cannot be located with a probability that is higher than $1/4$ and thus remains indistinguishable among three other users. However, if user *A* transmits the cloaked request to the LBS service providers then his/her identity may be revealed through the use of the existing cellular positioning technology. Thus, it is necessary for the querying user to communicate the request to the service providers through another mobile user (the agent).

In our example, user *A* communicates the cloaked request to user *C* through mobile user *B*. Subsequently, user *C* forwards the anonymized request to the appropriate service provider who in turn generates a set of candidate answers (based on region *R*) and communicates them back to user *C*. Following that, the mobile user *C* transmits the candidate answers to the actual requester (i.e. user *A*) who uses his/her current location to remove the false hits and retrieve the actual answer to his/her query. As a final comment, we should point out that there is a probability that the requester may not use an agent to perform the transmission of his/her cloaked request to the service providers. This is due to the fact that if the privacy protocol required that the querying party must always use an agent when requesting an LBS, then an adversary would be certain that the mobile user who communicates with the service providers is not the requester. As a result, in our example, the adversary (or equivalently the LBS service provider) would be certain that user *C* is not the requester and thus user *A* would remain anonymous among two — instead of three — other users (i.e. *B* and *D*). To alleviate from this issue, the agent is selected with uniform probability among all the users that participate in region *R* (i.e. including the requester).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

16.3 Working Assumptions and Threat Model

Several approaches have been proposed so far in the research literature for the offering of privacy in LBSs. In what follows, we present the working assumptions about the capabilities of the attacker that are employed by most of the state–of–the–art privacy approaches (discussed as part of Chapter 17). The knowledge of these assumptions is necessary to compare the different approaches in terms of privacy guarantees that they offer to the requesters of LBSs. Generally, the attacker is assumed to have the following capabilities:

1. The attacker intercepts the region R where anonymity is offered to the requester of an LBS. This implies that either the LBS service provider is untrustworthy, or the communication channel between the LBS service provider and the anonymizer (in the case of the centralized privacy model) or the LBS provider and the mobile users (in the case of the distributed privacy model) is insecure.

2. The attacker has knowledge of the algorithms that are used for the offering of privacy in LBSs. In the case of the centralized privacy model this means that the attacker knows the algorithms that are used by the trusted server to provide anonymity to the requesters of LBSs. On the other hand, in the distributed (P2P) privacy model the attacker is assumed to have knowledge of the privacy protocols that are used for the communication among the various parties, as well as the algorithms for the formulation of region R. This situation is common in the security literature where algorithms are typically publicly available.

3. The attacker can obtain the current location of all the users in the system. This assumption is motivated by the fact that users may often issue queries from locations which can be easily identified e.g., through physical observation, triangulation, telephone catalogs, etc. However, it is rather difficult to model the exact amount of knowledge that an attacker may have at his/her disposal. For this reason, this assumption dictates that the employed privacy methodology must be provably secure under the worst–case scenario.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

4. The attacker tries to breach the location privacy of the users by using only current location data; he/she is unaware of any historical information about the movement of the users in the system, as well as any behavior patterns of particular clients (e.g., a user often asking a particular query at a certain location or time)[1].

16.4 Organization of the Rest of Part III

The rest of Part III is organized as follows: In Chapter 17 we present a short survey of the current state–of–the–art approaches for the offering of privacy in LBSs, while in Chapter 18 we discuss the contributions that we made in this research area. Chapter 19 provides the background and the terminology that is necessary for the proper understanding of our proposed methodologies, given in detail in Chapters 20–23. Specifically, in Chapter 20 we present two privacy methodologies that extend the current state–of–the–art for the offering of historical $\mathcal{K}$–anonymity in LBSs. Moreover, in Chapters 21 and 22 we propose a free–terrain model and a network–based model (respectively) for the offering of trajectory $\mathcal{K}$–anonymity in LBSs. PLOT, the first open–ended toolbox that encompasses a set of centralized methodologies for the offering of privacy in LBSs, is presented in Chapter 23. Finally, Chapter 24 concludes this part of the dissertation and presents some potential future directions in this area of research.

[1]As is presented in Chapter 17 this assumption holds only for location privacy methodologies. In trajectory privacy methodologies (such as [39]) it is possible for an attacker to have knowledge of the recent history of queries placed by the requester as part of the same LBS (i.e. in LBSs that require multiple location updates in order to be offered). Furthermore, the approaches that we proposed in [69, 71, 77, 79] assume that the attacker is also in possession of the frequent routes that are followed by each user of the system.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 17

Related Work

Since the late 90s when the first approaches were proposed for the offering of privacy in LBSs, there has been a substantial body of research in this area of study, leading to the organization of numerous international conferences and workshops. In what follows, we present a taxonomy of the different approaches that have been proposed over the years, highlighting the progress that has been accomplished from the first generation of approaches to modern privacy algorithms. The detail of our presentation is biased in favor of $\mathcal{K}$–anonymity based approaches, as this category is the most prevalent among the different research directions and also it collects our research contributions, presented in detail in the following chapters. The interested readers are encouraged to refer to [37, 74, 171] for some interesting surveys that present alternative taxonomies of the existing privacy methodologies.

In the taxonomy that follows, we have partitioned the existing approaches for the offering of privacy in LBSs into five broad categories, based on the methodology that is used to prohibit the disclosure of the sensitive location data or the sensitive query content to untrusted third parties. Specifically, policy–based approaches (Section 17.1), provide the necessary tools that allow each user to specify a privacy policy that controls access to his/her sensitive location information, as well as to determine the granularity in which his/her location is disclosed to others. The approaches of this category ensure that the location information of the user is disclosed only to the authorized entities and in this sense,

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

these approaches are more related to security than to privacy [171]. On the other hand, cryptography–based approaches (Section 17.2) aim at securing the communication channels by encrypting the sensitive information that is communicated between the users and the service providers prior to its transmission through the (insecure) telecommunication infrastructure. In particular, cryptography–based solutions ensure that the sensitive location information as well as the sensitive query content of the user request, even if intercepted during its transmission, it will be recognizable only by the authorized recipient. The third category of approaches collects centralized methodologies that blur the location of the requester (as transmitted to the trusted server) prior to forwarding the user request to the untrustworthy LBS providers. As part of this category, we survey existing methodologies that shield the identity of the requester (i) by using historical movement data, collected from all the users in the system (Section 17.3.1), (ii) by using the current location of all the users in the system to cover the location of request (Section 17.3.2), and (iii) by using the current location of all the users in the system, as well as their future locations, to cover up the trajectory of the requester from the time of request until the provision of the LBS (Section 17.3.3). The fourth category in our taxonomy collects obfuscation and unlinking approaches (Section 17.4). Obfuscation approaches introduce perturbations to the locations of the users in the system in order to reduce their accuracy. Similarly, unlinking approaches blur the location of the users through the application of simple fuzification strategies. The last category of approaches collects distributed (P2P) methodologies for the offering of privacy in LBSs (Section 17.5). These approaches provide an alternative to the centralized $\mathcal{K}$–anonymity algorithms, since they offer $\mathcal{K}$–anonymity without the need of a trusted server. Instead, they require more powerful mobile devices on the client side as well as an adequate amount of cooperating users.

17.1 Policy–Based Approaches

The first generation of privacy techniques in LBSs was primarily influenced by the properties of specific types of networks, such as the GSM (e.g., [103,145,156]). These approaches,

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

as well as newer approaches that are not dependent on any particular type of hardware, were based on user–specified policies for the offering of privacy in LBSs. In policy–based approaches, each mobile user, supported by the telecommunication infrastructure, is allowed to specify a privacy policy to control *how*, to *whom* and to *what extent* his/her personal location data can be disclosed. Policy–based approaches acknowledge the release of location information to third parties, on a need–to–know basis. As an example, imagine a user that is interested to receive promotional coupons in his/her mobile device coming only from the supermarkets that are located in his/her neighborhood. Consequently, the service provider (who knows the current location of the user), cannot disclose his/her location to an arbitrary advertiser, but only to those who satisfy the conditions expressed by the user. The privacy preferences of the users are dynamically specified with the aid of special policy languages[1], which enable the declaration of a set of rules about who is allowed to access the personal location information of each user and for which purpose. The users are also allowed to define the places from which, as well as the granularity in which, such sensitive location information will be disclosed to others.

The first work that utilized privacy policies to protect the location privacy of the mobile users was proposed by Snekkenes [157]. In this work, each user is linked to a number of observations that include the identity of the user, the location where he/she was observed, the time of the observation, as well as the speed of the user at the time of observation. An observation can be released to other users at different levels of granularity (e.g., exact location, room, building or district where the user resides) depending on parameters such as the identity of the recipient and the intended use. The supported abstractions are modeled with the aid of a lattice structure which enables the user to determine the accuracy in which his/her observations will be disclosed to others. To allow the definition of the privacy policies of the users, the author proposes a formal policy language.

Hengartner and Steenkiste [89, 90] are the first to introduce the use of digital certificates to location privacy. A digital certificate is a signed data structure in which the signer

[1]A survey of the existing privacy policy languages can be found in [109].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

states a decision concerning some other entity. In the proposed approach, when a user initiates a request for the location of an individual, the service provider must first check whether the privacy policy grants access to the requester by building a chain of policy certificates from itself to the requester. When these steps finish successfully, data is securely released to the user who initiated the corresponding request.

Damiani and Bertino [45] propose the combination of an access control mechanism and a location privacy methodology in a system that filters incoming user requests for LBSs and determines whether they should be serviced based on the *role* and the *position* of the requester at the time of request. Each user can assume a variety of roles in the system and each role is assigned some generalization criteria. If accepted by the system, the request is de–identified and the location of the requester is generalized to an area within which he/she is located. The cloaking of the actual user location to the generalized area is based on a privacy policy that is specified by the user, which defines the degree of spatial generalization based on the current *role* that the user assumes in the system.

17.2 Cryptography–Based Approaches

Cryptography–based approaches encrypt the real identity of the individuals who initiate requests for LBSs, authenticate the query source as well as the transmitted requests, and secure the communication channels to ensure that the transmitted messages between the different entities of the system are not intercepted or altered in any way. Furthermore, some cryptographic approaches are capable of assuring the mobile user that the service providers with which he/she communicates for the offering of LBSs, are genuine.

The first cryptography–based methodology for the offering of privacy in LBSs relied on asymmetric key cryptography and is presented in [88]. It uses pseudonyms to reduce the trust that a user places to the service providers and (similarly to policy–based approaches) enables the users to define to whom their location will be disclosed and at which granularity. In the proposed approach, each user acquires a *pseudonym* that he/she uses as an asymmetric key, known only by the trusted server and the user. To disallow

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

any other user from intercepting it, the pseudonym is encrypted with the public key of the trusted server. The supplied key serves as a kind of reference with which potential subjects can address the user in their formulated location–based queries. To prevent an attacker from matching queries of different subjects to the same target user, the appearance of the cipher text of the pseudonym has to be different. For this reason, the pseudonym is also encrypted with a unique piece of information, chosen individually for each user. The benefit of the proposed strategy is that it enables the secure communication of the user with the authenticated service providers without disclosing his/her identity.

Konidala, et al. [107] propose a cost effective privacy protocol which ensures trust (i) between the mobile users and the telecom operators, by generating a master secret key that is shared between the user and the operator, and (ii) between the telecom operators and the service providers, by the existence of a trusted public key infrastructure. The telecom operators obtain a digital certificate and a private key and, similarly, the service providers obtain a digital certificate and a private key. Each telecom operator stores the certificate of each service provider (which contains the public key of the service provider), and the service providers store the certificates of the telecom operators (which contain the public key of the telecom operator). Initially, the user detects his/her current location by using a GPS. Then, he/she securely transmits the location to the telecom operator and requests a list of the LBSs that are offered in his/her neighborhood. The telecom operator replies with a list of services and takes full responsibility on behalf of the user to identify and authenticate the service providers. Then, the user selects a particular LBS among the offered ones and (securely) transmits the LBS parameters to the telecom operator. The operator communicates only the current location details, but not the identity of the user, to the appropriate service provider. Finally, it is important to mention that the service provider cannot maintain the user profile as it does not have knowledge to whom the requested LBS is being offered.

Two architectures that use *Hierarchical Identity–Based Encryption* (HIBE) to preserve the confidentiality of the communicated information, are presented in [91]. The authors propose a proof–based schema and an encryption–based access control schema. In

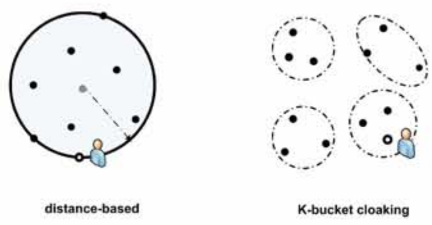
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the proof–based schema, a mobile user needs to assemble some access rights as a proof of access, which demonstrates to the service provider that the user is authorized to access the requested information. In an encryption–based schema, a service provider provides confidential information to any mobile user but only in an encrypted form. Those mobile users who are authorized to access the information, have the corresponding decryption key that will allow them to successfully decrypt it. Both schemas use HIBE [56], which is based on *Identity–Based Encryption* (IBE), where the public key of an individual is an arbitrary string. Ref. [28] analyzes one of the first proposed IBE schemes.

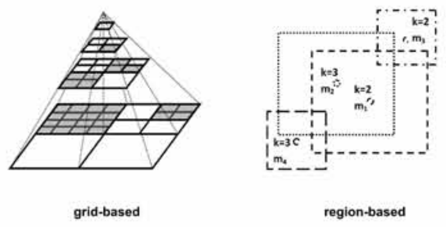
17.3 $\mathcal{K}$–Anonymization Approaches

The main body of research for the offering of privacy in LBSs collects approaches that are based on the notion of $\mathcal{K}$–anonymity. The concept of anonymity (in the sense of computational methods for its offering) was originally investigated in the area of telecommunications, where methodologies were proposed since the late 80s to protect the identity of senders and recipients in message transmissions. According to [141], *anonymity* is defined as “the state of being not identifiable within a set of subjects, the anonymity set”. Chaum [35] introduced the notion of an *anonymity set* as the set of participants who probably could have sent a certain request, as seen by a global observer. The concept of $\mathcal{K}$–anonymity, originally proposed by Samarati and Sweeney [149, 150, 162] in the context of relational data, is among the most prevalent anonymity preserving methodologies. $\mathcal{K}$–anonymity has been extensively applied in relational databases for publishing census, medical and voting registration data (often called *microdata*). The principle of $\mathcal{K}$–anonymity states that “each release of data must be such that every combination of values of private data can be indistinctly matched to at least $\mathcal{K}$ individuals”. In this sense, $\mathcal{K}$–anonymity requires that every record in a released dataset is indistinguishable from at least $\mathcal{K}$–1 other records with respect to a certain set of identifying variables, collectively known as the *quasi–identifier*. Quasi–identifiers are attributes (e.g., family name, date of birth, social security number), which, though not containing an explicit reference to the identity of indi-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Data–dependent cloaking.



(b) Space–dependent cloaking.

Figure 17.1: Cloaking strategies for the offering of $\mathcal{K}$−anonymity in LBSs.

viduals, they can be easily linked to publicly available external data sources (e.g., the yellow pages) to uniquely identify individuals. Several strategies have been proposed for the offering of $\mathcal{K}$–anonymity in relational data, as well as numerous extensions of the concept of $\mathcal{K}$–anonymity have been investigated over the years (e.g., *l*–diversity [115], *p*–sensitive $\mathcal{K}$–anonymity [167], *t*–closeness [112], $(\alpha, \mathcal{K})$–anonymity [177]). The most widely adopted strategy for $\mathcal{K}$–anonymizing a dataset applies a generalization of the values attributed to the quasi–identifiers. The generalization is performed by replacing the original attribute's value with a less specific, but semantically consistent, one (e.g., "divorced" can be replaced with "been_married", which also collects "currently_married" and "widowed").

In the context of LBSs, the "quasi–identifier" is the location of the individuals when requesting LBSs; releasing a request of an individual for an LBS to an untrusted third party should make certain that the actual location of request cannot be associated (at least with a high probability) with the identity of the requester (e.g., through map matching). To satisfy $\mathcal{K}$–anonymity, the most widely adopted anonymization strategy is cloaking. In cloaking, the actual location of request is transformed into a bounded area that is large enough to contain the requester along with (at least) $\mathcal{K}$–1 other (nearby) users in the system. The application of this generalization strategy ensures that the identity of the requester cannot be disclosed with a probability that is significantly larger than $1/\mathcal{K}$, among $\mathcal{K}$–1 other users. Some of the most prevalent cloaking strategies that generalize the actual locations of request to spatially bounded areas, are presented in Figure 17.1. They can be partitioned into two groups: *data–dependent cloaking* and *space–dependent cloaking* strategies.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

In data–dependent cloaking strategies, the anonymity area is formulated based on the actual location of each user in the system and his/her distance from the location of request. Specifically, distance–based cloaking algorithms (e.g., [26, 77, 79, 185]) retrieve the $\mathcal{K}$–1 nearest neighbors of the requester and generate a region that includes all the $\mathcal{K}$ users. In $\mathcal{K}$–bucket cloaking (e.g., [39, 100]) the users are arranged into groups of $\mathcal{K}$ and the anonymity region is computed as the *Minimum Bounding Rectangle* (MBR) that contains the $\mathcal{K}$ users in the group of the requester.

On the other hand, space–dependent cloaking strategies take into consideration the total spatial area that is covered by the system in order to formulate the areas of anonymity. Specifically, grid–based cloaking strategies (e.g., [69, 71, 82, 126]) partition the area in a grid fashion and formulate the anonymity set by retrieving the users in each cell of the grid (starting from the cell of the requester and moving to neighboring cells) until at least $\mathcal{K}$ users are found. Finally, region–based cloaking strategies, such as [54], use the spatial properties of the area to generate rectangles centered at the location of request and to utilize them for the offering of $\mathcal{K}$–anonymity. In what follows, we present a taxonomy of the available $\mathcal{K}$–anonymity approaches as applied for the offering of privacy in LBSs. Alongside the presented methodologies, we include some examples to demonstrate their operation for the provision of $\mathcal{K}$–anonymity.

17.3.1 Historical $\mathcal{K}$–Anonymity

Historical approaches to $\mathcal{K}$–anonymity (e.g., [26, 180, 185]), keep track of the movement history of each user in the system and utilize this information when building the anonymity sets for the requests of the users. Compared to other methodologies for the offering of $\mathcal{K}$–anonymity in LBSs, in historical $\mathcal{K}$–anonymity approaches the participants of the anonymity set are selected based on their history of movement in the system, with the requirement being that at some time in their history of movement these users were close to the point of request. Ref. [180] states this observation as “using users’ footprints instead of their current positions, for cloaking”.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Refs. [26, 185] consider the area that is covered by the trusted server as a set of Places–of–Interest (POIs) defined for each user in the system. Each POI has a spatial extent and can be related to an unanchored temporal interval (e.g., [9am – 10am]). It represents a place that is frequently visited by a user based on his/her history of movement in the system, as well as the approximate time–of–day (represented as a time interval) of these visits. A series of POIs (along with their corresponding time intervals) that are frequently visited in sequel by a user can be considered as hazardous with respect to the privacy of the user when requesting LBSs, as such requests can easily disclose his/her identity. Some strategies have been recently proposed for the automatic derivation of frequent sequences of POIs (e.g., [31, 63, 116]) in datasets representing user mobility.

The trusted server monitors the users to identify when they request LBSs from any of their POIs. When a user transmits a request for an LBS from one of his/her POIs, the trusted server computes an area along with a time interval that contains the requester, as well as $\mathcal{K}$–1 other users who happened at sometime in the near past to be nearby the location of the request. The computed area is said to be (historically) $\mathcal{K}$–anonymous as it protects the requester by guaranteing that his/her location cannot be identified with a probability that is larger than $1/\mathcal{K}$, among the other $\mathcal{K}$–1 users. The whole anonymization process is guided by a set of spatial and temporal constraints; the spatial constraints require that the generated area of anonymity is within some reasonable spatial bounds so as to allow the provision of the requested LBS, while adequately hindering the actual location of request. On the other hand, the temporal constraints impose a barrier on how back (starting from the point of request) should the history of movement of all the users in the system be searched, so as to retrieve the participants of the anonymity set. Following the computation of the anonymity region, the trusted server forwards the request containing the cloaked user location to the service provider for servicing. When a subsequent request is received from the same user, the trusted server tries to match this request to the next POI in his/her sequence of POIs, and if the match is successful it recomputes the area along with the time interval that contains the requester, as well as his/her $\mathcal{K}$–1 original neighbors.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

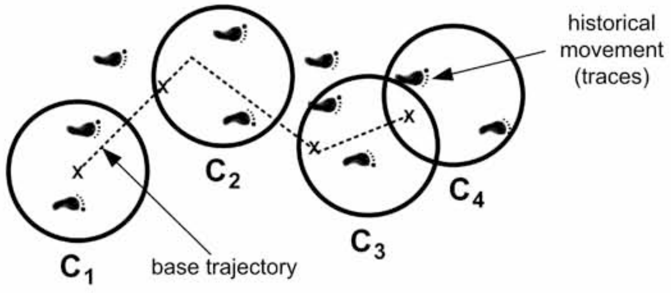


Figure 17.2: $\mathcal{K}$–anonymity based on historical movement.

Ref. [185] extends [26] by enabling each user to have numerous sequences of POIs and by adapting the generalization algorithm of $\mathcal{K}$–anonymity to account for those multiple POIs. The experiments conducted as part of [185] prove the effectiveness of the proposed approach to protect the privacy of the requesters of LBSs.

Xu and Cai [180] use the history of movement of all the users in the system to apply a recursive top–down partitioning of the area that is covered by the trusted server into equi–size quadrants. Each cell of the partitioning is divided into quadrants up to the point that it contains at most N users who have visited this cell sometime in their movement history. By using this structure alongside a hash table that records the user IDs and the trajectories for the users of each cell, the proposed approach requires that each user who wishes to request an LBS has to first communicate a *base* trajectory to the trusted server. A base trajectory $T = \{c_1, c_2, \ldots, c_n\}$ defines the itinerary that the user will follow in the system, where each c_i corresponds to a location update on the trajectory on which the user will move. In response, the trusted server computes a new trajectory $T' = \{C_1, C_2, \ldots, C_n\}$ that provides $\mathcal{K}$–anonymity to the user when using the LBS throughout his/her declared itinerary. Each C_i in T' corresponds to a region of (historical) $\mathcal{K}$–anonymity that contains the requester (base trajectory) along with $\mathcal{K}$–1 users based on their historical trajectories in the system. Whenever the user arrives at c_i he/she informs the trusted server who, in turn, uses the $\mathcal{K}$–anonymity region of C_i to continue to protect his/her privacy. Figure 17.2 demonstrates the operation of the algorithm for the offering of historical 3–anonymity to a requester of an LBS based on his/her computed base trajectory.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

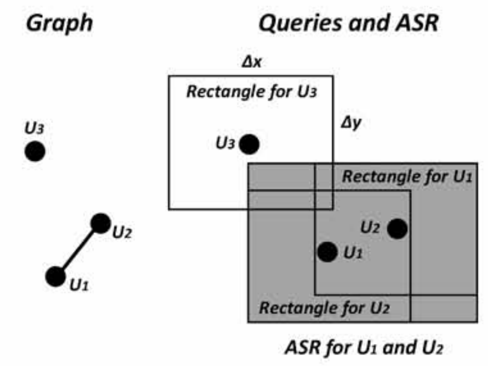


Figure 17.3: An example of *Clique Cloak*.

17.3.2 Location $\mathcal{K}$–Anonymity

Location $\mathcal{K}$–anonymity approaches (e.g., [54, 55, 82, 100, 126]) protect the privacy of the requesters of LBSs by utilizing the current location (instead of the history of collected locations) of each user in the system. They operate on LBSs that require a single location transmission from the requesting party in order to be successfully provided (e.g., store–finder, friend–finder, etc), instead of the communication of multiple location updates (e.g., as in car navigation services). The different cloaking strategies that have been proposed for the offering of location $\mathcal{K}$–anonymity are presented in Figure 17.1. In what follows, we detail over some of the most popular approaches in this category.

Clique Cloak [54] is a graph–based (region–based) approach that has been proposed for the mutual anonymization of multiple incoming requests for LBSs. For each query that is received for servicing, the algorithm generates a rectangle centered at the location of the requester, with its sides being parallel to the considered x and y–axis, respectively having Δx and Δy extents. The new query is then marked as a node in a graph for as long as it awaits for its anonymization. Two vertices (queries) in the graph are connected together if the corresponding users fall in the rectangles of each other. An edge of the graph demonstrates that the requester of each of the two queries can be included in the computed anonymity set of the other, and thus, a $\mathcal{K}$–clique of the graph shows that all the corresponding $\mathcal{K}$ requests can be anonymized together (thus offering $\mathcal{K}$–anonymity to all the $\mathcal{K}$ users that

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

participate in the $\mathcal{K}$–clique). Finally, tight to each request is a temporal interval Δt that defines the maximum amount of time that this request can be retained by the system for its anonymization. If a $\mathcal{K}$–clique cannot be found within Δt then the request is dropped as unserviceable. Figure 17.3 demonstrates the operation of *Click Cloak* in the case scenario where three queries for LBSs (located at U_1, U_2 and U_3) have been synchronously submitted to the trusted server. Assuming that $\mathcal{K} = 2$, the generated rectangles for U_1 and U_2 fall in each other and thus they form a 2–clique in the graph. As a result, the MBR enclosure of the respective rectangles (shown here in gray) represents the *Anonymity Spatial Region* (ASR) where 2–anonymity is offered to these users. On the other hand, the request of U_3 has to wait in the system until a new query (formulating a 2–clique with U_3) arrives. As one can observe, *Clique Cloak* may affect the quality of service that is offered to the users as the servicing of some queries may be substantially delayed, while other queries may be dropped as unserviceable. The approaches that follow do not suffer from these shortcomings.

Center Cloak [100] is a distance–based approach that provides a naïve solution to $\mathcal{K}$–anonymity in LBSs. In *Center Cloak*, the $\mathcal{K}$–1 nearest neighbors of the requester are retrieved and the ASR is computed as the MBR enclosure of all the $\mathcal{K}$ users. By construction, *Center Cloak* suffers from what is known as the "center–of–ASR" attack; the identity of the requester can be accurately guessed with a probability that far exceeds $1/\mathcal{K}$ as he/she is expected to be close to the center of the ASR. The "center–of–ASR" attack is an instance of a more general problem that is worth mentioning. Since cloaking algorithms are expected to be publicly available, attackers can easily exploit any implementation decisions regarding the placement of the requester to the generated ASRs. As an effect, several of the currently available approaches suffer from similar kinds of attacks.

A randomized variant of *Center Cloak*, which offers increased uncertainty regarding the actual location of the requester inside the generated ASR, is Nearest Neighbor Cloak (*NN–Cloak*) [100]. In *NN–Cloak* the ASR is formulated as follows: Given a user query for an LBS, *NN–Cloak* first retrieves the $\mathcal{K}$–1 nearest neighbors of the requester. Second, it randomly selects one among the $\mathcal{K}$ users and identifies his/her $\mathcal{K}$–1 nearest neighbors. Finally,

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

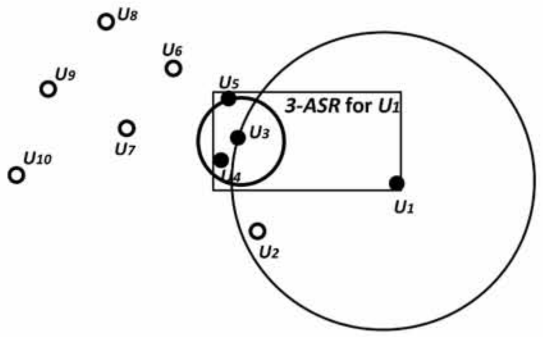


Figure 17.4: An example of *NN–Cloak*.

the $\mathcal{K}$–ASR is constructed as the MBR enclosure of the second set of $\mathcal{K}$ users, augmented (if necessary) to include the requester. Figure 17.4 presents an example where 3–anonymity is offered to user U_1 by using *NN–Cloak*. First, U_1 formulates set $S_1 = \{U_1, U_2, U_3\}$ that contains the requester U_1 along with his/her two nearest neighbors: U_2 and U_3. Second, *NN–Cloak* randomly selects U_3 from S_1 and computes his/her two nearest neighbors in the system. This leads to set $S_2 = \{U_3, U_4, U_5\}$, since U_4 and U_5 are the two nearest neighbors of U_3. Finally, the MBR of S_2 is augmented to include the requester U_1, leading to the 4–ASR for U_1 that is shown in Figure 17.4. Since the probability of selecting the requester from S_1 when formulating S_2 is at most $1/\mathcal{K}$ (due to random choice), *NN–Cloak* is not vulnerable to the "center–of–ASR" attack.

Grid–based approaches to location $\mathcal{K}$–anonymity partition the total area that is covered by the trusted server into a number of *cells*. The cells are then organized with the help of a data structure into clusters, based on their spatial proximity. When a user submits a request to the trusted server, the cell where the requester is located is retrieved and searched for $\mathcal{K}$–1 other (nearby) users. If these users are not found within the same cell, the data structure is used to retrieve as many neighboring cells (to the cell of request) as needed in order to offer $\mathcal{K}$–anonymity to the requester. We should be aware that the returned ASR may contain more than $\mathcal{K}$ users, based on the number of users that exist in each cell. After its cloaking, the request is forwarded to the service providers for servicing and the produced results are filtered by the trusted server based on the exact location of the requester.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

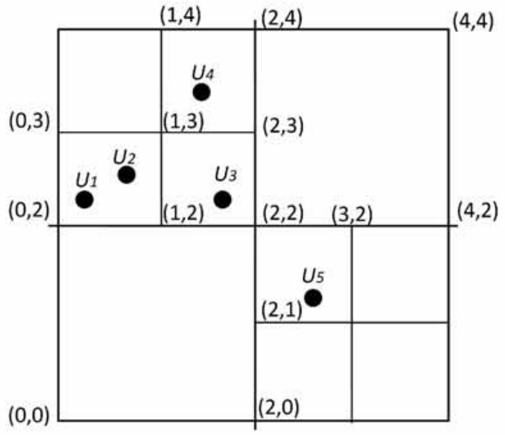


Figure 17.5: An example of *Casper*.

One of the most popular grid–based approaches to location $\mathcal{K}$–anonymity is *Casper* [126]. In *Casper* the entire area that is covered by the trusted server is divided in a grid–fashion and organized in a pyramid data structure of layers that is similar to a Quad–tree [151] (see the pyramid structure in Figure 17.1(b)). The top layer of the pyramid contains the entire area, whereas the lowest level of the pyramid collects the finest–grained granularity of the partitioning. Each cell in the lowest level of the pyramid has a minimum size that corresponds to the anonymity resolution. When a new query for an LBS is received by the trusted server, *Casper* locates the lowest–level cell in the pyramid that contains the requester and examines if this cell also contains $\mathcal{K}$–1 other users. If the cell contains enough users then it becomes the $\mathcal{K}$–ASR. Otherwise, *Casper* searches the horizontal and the vertical neighbors of this cell to identify if the number of users in each of these cells, when combined with the number of users in the cell of the requester, suffice for the provision of location $\mathcal{K}$–anonymity. If this is true, then the corresponding union of cells becomes the $\mathcal{K}$–ASR. Else, *Casper* moves one level up in the pyramid structure to retrieve the parent (cell) of the cell of request and repeats the same process until the $\mathcal{K}$ users that will formulate the ASR are found. Figure 17.5 provides an example of this cloaking operation. Assuming a request coming from cell $\langle(0,2),(1,3)\rangle$ (where (0,2) are the lower–left and (1,3) the upper–right coordinates of the cell) with an anonymity require-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

ment of $\mathcal{K} = 2$, the returned ASR is the same cell. In the event that a query with the same anonymity requirements is issued from cell $\langle(1,2),(2,3)\rangle$ the returned ASR is the union of cells $\langle(1,2),(2,3)\rangle \cup \langle(1,3),(2,4)\rangle$. Finally, if the query is issued by U_5 then the returned ASR is the entire area $\langle(0,0),(4,4)\rangle$.

Interval Cloak [82] is very similar to *Casper* as it also partitions the total area that is covered by the trusted server into equi–sized quadrants and organizes this information in a Quad–tree structure. However, *Interval Cloak* does not consider the neighboring cells at the same level when computing the ASR, but instead it directly ascends to the ancestor level in the pyramid. As an example, in Figure 17.5, a request for an LBS that is issued by U_3 or U_4 will generate the ASR $\langle(0,2),(2,4)\rangle$ (instead of $\langle(1,2),(2,4)\rangle$ for *Casper*). As is evident, *Casper* is more effective in producing compact ASRs when compared to *Interval Cloak*. However, as is proven in [100], both *Interval Cloak* and *Casper* are secure only for uniform data distributions. *NN–Cloak* also faces similar problems.

Hilbert Cloak [100] does not suffer from these shortcomings as it achieves to generate the same $\mathcal{K}$–ASR, no matter which of the participants of the anonymity set has requested the LBS. The proposed approach is based on $\mathcal{K}$–bucket cloaking as it dynamically arranges the users in the system into groups of $\mathcal{K}$ and computes the ASR as the MBR enclosure that contains the $\mathcal{K}$ users in the group of the requester. *Hilbert Cloak* uses the Hilbert space filling curve [127] to create an one–dimensional mapping of the position of each user in the system. In the proposed mapping, locations that are near to each other in the two–dimensional plane, are expected to also lie near each other to its one–dimensional transformation. For each request that is submitted to the trusted server for the provision of an LBS with an anonymity requirement of $\mathcal{K}$, *Hilbert Cloak* partitions each $\mathcal{K}$ users in the system into a bucket according to their Hilbert values. Following this partitioning, each bucket (probably with the exception of the last one) contains exactly $\mathcal{K}$ users. Following that, *Hilbert Cloak* retrieves all the $\mathcal{K}$–1 users that lie in the same bucket as the requester, and formulates the $\mathcal{K}$–ASR as their MBR enclosure. An example of this operation is presented in Figure 17.6, where we consider 10 users whose IDs are sorted in ascending order

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

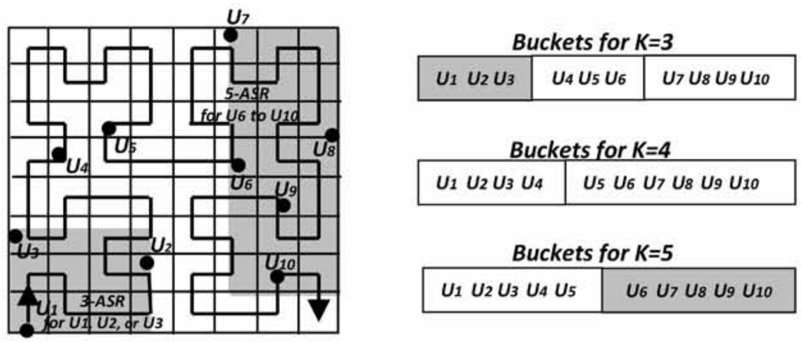


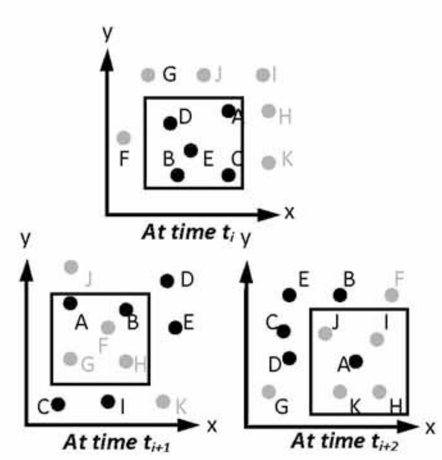
Figure 17.6: An example of *Hilbert Cloak*.

based on their Hilbert values. Given a query for an LBS from U_3 with an anonymity requirement of $\mathcal{K} = 3$, *Hilbert Cloak* uses the rank of the user (here 3) to dynamically identify the bucket in which he/she is partitioned (here is the first bucket). Then, it retrieves all the users who are partitioned in the same bucket as the requester (i.e. U_1, U_2 and U_3) and returns their MBR as the computed ASR for this request (see the 3–ASR shaded region in Figure 17.6). It is important to notice that any query with an anonymity requirement of $\mathcal{K} = 3$ originating from any of U_1, U_2, would generate the exact same 3–ASR as the one that is generated for U_3. Furthermore, it must be noted that *Hilbert Cloak* can generate the bucket that contains the requester on–the–fly based on the rank $r(U)$ of each user U in the system. The bucket where user U is located contains the users whose ranks are in the range of $[s, e]$, where $s = r(U) - (r(U) - 1) \bmod \mathcal{K}$ and $e = s + \mathcal{K} - 1$ (with the exception of the last bucket).

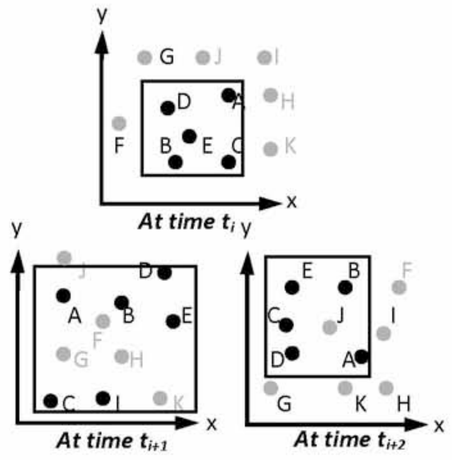
17.3.3 Trajectory $\mathcal{K}$–Anonymity

Trajectory $\mathcal{K}$–anonymity approaches (e.g., [39, 69, 77]) are appropriate for preserving the privacy of the users who request LBSs that cannot be offered in just a single communication of the user with the service provider. As an example, consider a car navigation service, in which the current position of the user has to be communicated to the service provider for as long as the user travels to his/her destination (so that he/she receives updated directions). The approaches of this category are responsible for protecting the whole user trajectory of the requester by offering him/her (trajectory) $\mathcal{K}$–anonymity, from the time of request

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) The attack scenario.



(b) Eliminating the attack.

Figure 17.7: An example of the query tracking type of attack and its elimination.

until the service provision. Such services are called *continuous* and the corresponding requests for these LBSs are termed as *continuous queries*. It is important to mention that contrary to historical $\mathcal{K}$–anonymity approaches (such those of [26, 180]), which can also protect the user trajectory by providing $\mathcal{K}$–anonymity to the requester of LBSs, trajectory $\mathcal{K}$–anonymity approaches generate the $\mathcal{K}$–ASRs by utilizing the *current movement* of the users in the system (instead of their historical traces) to adequately cover up the trajectory of the requester. In what follows, (i) we motivate the necessity for trajectory $\mathcal{K}$–anonymity methodologies by discussing the limitations of location $\mathcal{K}$–anonymity approaches, and (ii) we provide a partitioning of the existing trajectory $\mathcal{K}$–algorithms along two principal directions as well as discuss some of the most prevalent approaches in each direction.

Why Trajectory $\mathcal{K}$–anonymity? — Query Tracking in LBSs

Existing location $\mathcal{K}$–anonymity approaches are suitable for protecting only the location of a user request for a continuous LBS, instead of the complete user trajectory. As [39] indicates, these approaches suffer from correlation attacks, which can reveal the identity of the requester based on the participants of his/her anonymity set. Figure 17.7 demonstrates

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

how this is possible. Imagine a query submitted at time t_i by user A for a navigation service (or generally for a continuous service that requires multiple location updates). The applied location cloaking strategy (e.g., [82, 126]) generates the 5–ASR shown in Figure 17.7(a) that includes the requester along with four of his/her neighbors: B, C, D and E. As the user moves to the selected destination, he/she needs to transmit a new location update to the service provider to request for further directions. Thus, at time t_{i+1} the user sends a new query to the trusted server containing his/her current location. However, the users in the neighborhood of the requester have also moved in the meanwhile and thus the new 5–ASR that is produced by the employed location $\mathcal{K}$–anonymity algorithm (i.e. $\{A, B, F, G, H\}$) has only user B in common with the previously generated ASR. As a result, if an attacker has knowledge of the two ASRs he/she can safely conclude that the requester is either A or B, which significantly reduces the actual degree of anonymity that is offered to A from $1/5$ to $1/2$. In the next location transmission, at t_{i+2}, the identity of the requester is revealed since no other participant in his/her current ASR was also part of all the previous ASRs.

To alleviate from correlation attacks, existing approaches to trajectory $\mathcal{K}$–anonymity ensure that *all* the $\mathcal{K}$ participants of the first $\mathcal{K}$–ASR that is generated for the offering of privacy in a continuous query, will also participate in *all* the following $\mathcal{K}$–ASRs that will be produced by the cloaking algorithm. This way of eliminating the query tracking attack is presented in Figure 17.7(b). We should point out that although in the second ASR (related to time t_{i+1}) it seems that 10–anonymity is offered to the requester, only 5–anonymity is actually offered throughout the provision of the requested LBS to user A.

Generic Approaches to Trajectory $\mathcal{K}$–Anonymity

Generic approaches to trajectory $\mathcal{K}$–anonymity do not take into consideration any particular movement behavior of the requester of an LBS (as well as of the other users in the system), when providing him/her with trajectory $\mathcal{K}$–anonymity. Instead, all requests for LBSs are handled in exactly the same manner by the trusted server, no matter what the location of request is or the trajectory that the user follows in the system during the LBS provision.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

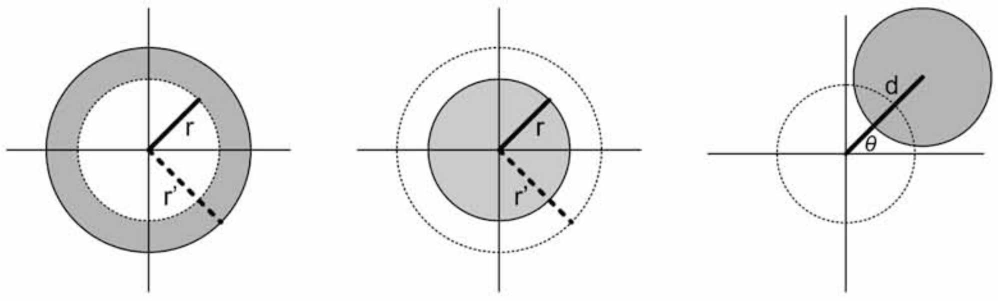
Ref. [39] proposes the first cloaking algorithm for the offering of trajectory $\mathcal{K}$–anonymity in LBSs. The principal idea behind the proposed algorithm is to require that a user belongs in a group of at least $\mathcal{K}$–1 other users prior to sending a continuous query for the provision of an LBS. Users may leave their groups upon completion of the requested service. The generated $\mathcal{K}$–ASR in each location transmission of the user, is simply computed as the MBR enclosure of all the users in the group of the requester, based on their current location in the system. It is important to mention that while a request for an LBS is in progress, no grouped user that participated to the original anonymity set of the requester is allowed to leave the group, as this action would jeopardize the privacy of the requester.

Personalized Approaches to Trajectory $\mathcal{K}$–Anonymity

Personalized approaches to trajectory $\mathcal{K}$–anonymity utilize the history of movement of all the users in the system to cope with correlation attacks in continuous user queries. They differ from generic approaches to privacy in LBSs, primarily due to the following reasons: (i) the movement of each user u in the system is depicted as a continuous function $f(u,x,y,t)$, instead of a set of individual locations and times, (ii) a set of frequent movement patterns are derived based on the history of movement of each user in the system, which are subsequently used to protect the privacy of the user when requesting LBSs, and (iii) trajectory $\mathcal{K}$–anonymity can be offered to the requesters of LBSs through the use of an underlying network topology of user movement, instead of a grid–based, free–terrain solution.

Ref. [69] provides a trajectory $\mathcal{K}$–anonymity solution that uses the history of movement of the users in the system to derive a set of frequent mobility patterns per user. Each of these patterns corresponds to a route (instead of a sequence of POIs and related time periods, as in [26]) that is frequently followed by the corresponding user in the system and is stored as an $f(x,y,t)$ function, having both a spatial and a temporal extent. The proposed algorithm identifies those frequent routes of a user that are rarely followed by many other users in the system. These routes are termed as *unsafe* for this user, as they can disclose his/her identity when requesting LBSs from within any of them. In [69] a grid–based, free–

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) enlarging the radius. (b) reducing the radius. (c) shifting the center.

Figure 17.8: The three obfuscation techniques proposed in [19].

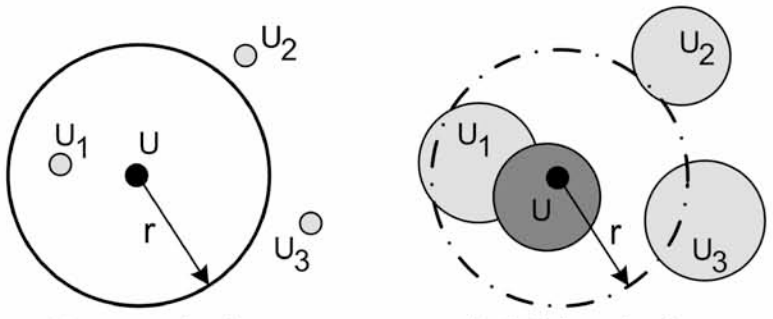
terrain solution is employed that utilizes the computed unsafe routes of the users, in order to provide them with $\mathcal{K}$–anonymity when requesting LBSs.

In [77] a network–based privacy model is proposed that considers an underlying network of user movement in order to derive the unsafe routes of the users and to offer trajectory $\mathcal{K}$–anonymity to the requesters of LBSs. With respect to the offering of $\mathcal{K}$–anonymity, the proposed approach considers two spatial cloaking strategies, depending on the location of the requester at the time of request, as well as his/her subsequent locations until the provision of the service. In particular, *$\mathcal{K}$–present* (the so–called weak) trajectory anonymity identifies $\mathcal{K}$–1 users that are close to the requester at the time of request and thus could have issued the request for the LBS. On the other hand, *$\mathcal{K}$–frequent* (strong) trajectory anonymity, collects the subjects that were near the requester at the time of request and for whom the currently traveled route of the requester is also frequent.

17.4 Obfuscation and Unlinking Approaches

Obfuscation and unlinking approaches provide alternatives to $\mathcal{K}$–anonymity. Obfuscation solutions introduce perturbations to the locations of the users (as collected by the trusted server), in order to decrease their accuracy. On the other hand, unlinking approaches are suitable for services that can track user movement. In what follows, we present some approaches that fall in each of these two categories.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) exact user locations. (b) cloaked user locations.

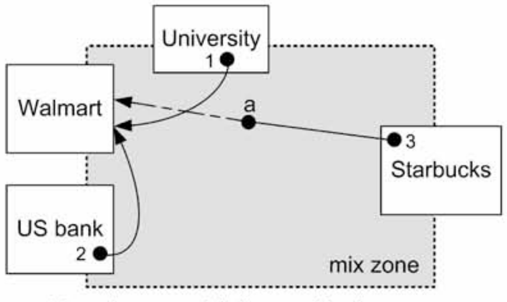
Figure 17.9: An example of *Probabilistic Cloak*.

17.4.1 Obfuscation Approaches

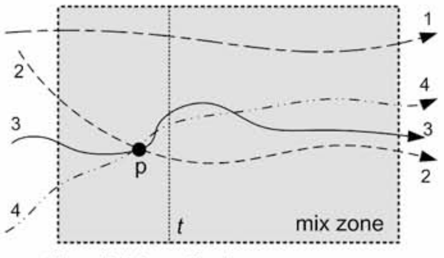
Obfuscation approaches are suitable for LBSs that require a user location at a specific time. In [19] the authors propose three obfuscation techniques (see Figure 17.8) that offer different degrees of location privacy to the requesters of LBSs. The proposed approaches assume that the location of the requester is transmitted to the trusted server as a circular area, inside which the requester can be located at any point. Then, based on the privacy preferences of the user, one among three alternative transformations is employed in order to further degrade the accuracy of locating the requester inside this area. The supported transformations are: obfuscation by enlarging/reducing the radius of the area, and obfuscation by shifting the center of the area. To enhance the privacy that is offered to the requesting parties, a combination of the proposed transformations can also be applied.

Cheng, et al. [38] propose *Probabilistic Cloak*, a cloaking algorithm for the offering of location privacy in LBSs in which the produced ASRs are independent of the number of users that lie inside them. In *Probabilistic Cloak* each user is assumed to have a privacy profile in the system that he/she can use to define the level of uncertainty with which his/her location information will be released to others. When a user requests an LBS, the algorithm produces a cloaked version of his/her precise location based on user supplied privacy requirements. The computed ASR is forwarded to the service provider for servicing and the returned result contains the candidate answers augmented with probability to indicate the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) a mix zone with four application zones.



(b) unlinking of trajectory segments.

Figure 17.10: Two examples of *mix zones*.

confidence of each answer to satisfy the query, based on its location in the ASR. Figure 17.9 presents an example of using *Probabilistic Cloak* in a location–based range query. In this example, we assume that user U wants to identify the subjects that are inside a radius r from his/her location. The system uses the privacy profiles of all the users in the neighborhood of U to cloak their locations into areas. The applied cloaking strategy offers an even chance to each user to be located at any point inside his/her cloaked region. Subsequently, *Probabilistic Cloak* answers the query and returns the candidate results (here $U_{1..3}$) to the requester, along with probabilities that are based on the percentage of overlap of each cloaked location to the region of request.

17.4.2 Unlinking Approaches

Unlinking approaches are suitable for the offering of privacy in continuous queries for LBSs. They operate by devising simple ways to blur the actual location of the requester through the application of simple fuzification strategies.

In [25] the authors partition the area that is covered by the trusted server into a set of application zones and a set of mix zones. When a user is inside an application zone he/she regularly transmits location updates to the service provider, as well as is allowed to use any of the LBSs that he/she has registered in the system. On the contrary, the mix zones are spatial regions in which none, among a set of users, has registered any application callback. As an effect, the users that are inside a mix zone do not transmit any location information

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

to the service provider. When users enter the mix zone, the trusted server tries to "mix" their identities by changing the user pseudonyms. As a result, it becomes difficult for an adversary to link the new identities of the users when exiting the mix zone to those that they had when they entered the zone. In that sense, a mix zone can been considered as an analogy of a black box that protects the user identities, offering privacy to the users. Figure 17.10 presents two examples of mix zones. In the first example (see Figure 17.10(a)), assuming that users 1 and 2 simultaneously depart from the University and the US bank branch (respectively), an adversary is unable to tell who is who when they both arrive at Walmart. The second example (Figure 17.10(b)) shows how the trajectory segments of four users can be unlinked during the movement of the users inside the mix zone.

Mix zones suffer from several shortcomings: (i) they require an adequate number of users to be synchronously located inside the mix zone to provide them with a reasonable level of privacy; as a result they operate well only in high density areas, where user paths frequently meet, (ii) they are statically defined and thus they achieve to protect only those users that are moving towards a mix zone, leaving the rest of the population in the system unprotected, and (iii) they suffer from statistical attacks due to the knowledge of the movement of the users prior to entering the mix zone and their movement behavior when leaving the mix zone. In essence, the entry point of a user in the mix zone, considered together with the previous movement of the user, usually provides a good estimate on the exit point of the user from the mix zone. As an example of (i) consider the mix zone of Figure 17.10(a). In the best case scenario, this mix zone offers $\mathcal{K} = 3$ anonymity to each of the three users. In reality however, based on the movement behavior of the three users prior to entering the mix zone, their speed and direction of movement, the degree of anonymity that is offered to each of the three users may be much lower. Assuming that all users in the mix zone of Figure 17.10(a) depart at the same time from their current application zone and move towards Walmart at the same speed, user 3 is unlikely to reach his/her destination at the same time as the other two users, due to the increased distance of Starbucks to Walmart. As an effect, this mix zone is bound to offer 2–anonymity to each of the users 1 and 2.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Ref. [185] provides a solution to (ii) by allowing the mix zones to be defined on demand. In this work, an unlinking region that includes the requester along with $\mathcal{K}$–1 nearby users in the system is generated. Inside this dynamically defined region, no user is able to transmit any location update to the service provider, as well as use any LBS for a predetermined amount of time. As an effect, it is expected that the identities of the users who move inside this region will be successfully "mixed" in the meanwhile. Ref. [48] proposes a more advanced strategy. To increase the uncertainty regarding the identity of a user after exiting the mix zone, the proposed methodology ensures that the trajectory of this user has been crossed by those of at least N other users in the mix zone, prior to changing the pseudonyms of the users and resuming the regular operation of the system in this region. Each crossing of the trajectory of a user with another user trajectory increases the uncertainty regarding the movement of each of the two users after exiting the mix zone. For example, in Figure 17.10(b) after the crossing of the three user trajectories in point p it is rather difficult to match the three trajectories to their owners. Thus, resuming the regular operation of the system at time t successfully shields the identity of each of the three users $2,3$ and 4, offering them 3–anonymity.

In [105] the authors propose an anonymous communication technique that protects the privacy of the requesters of LBSs by introducing several false position data (a.k.a. *dummies*), along with the true location updates of the users. As an effect, even if the service providers store location data, they cannot be certain about the whereabouts of the real users. On the client side, the mobile device of the user is capable of disregarding answers sent from the service provider that involve dummy locations and present to the mobile user only the answers involving his/her true location updates. The challenge in the proposed methodology is to achieve realistic dummy movements that will confuse an adversary regarding the true locations of the user, as well as to minimize the communication overhead that is introduced by the dummies to the telecom operator. To achieve the first target, the authors propose two dummy generation algorithms that attempt to model real user movement. In both algorithms, the initial locations of the dummies are randomly selected. In the first

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

algorithm, the next position of a dummy is tight to be in the neighborhood of the current position of the dummy. To achieve that, the mobile device of the user memorizes the last location of the dummy and generates dummies around this location. On the other hand, the second dummy generation algorithm is applicable when the mobile device of the user can acquire the location of other users in the system. In this case, the dummies are generated in the neighborhood of these users, while special care is taken to ensure that the density of the users in the region where the dummies are produced remains within acceptable bounds. To minimize the communication overhead that is introduced by the dummies, the authors propose a message compression technique along with four strategies that a user can follow to transmit information unrelated to position data to the service provider. Although this is an interesting approach to privacy in LBSs, it suffers from several shortcomings [74].

17.5 Distributed (P2P) Approaches

The existence of a centralized architecture where a trusted server (anonymizer) is responsible for the offering of privacy in LBSs, incurs several shortcomings [57, 125]: (i) all the users of the system must put trust on the third party anonymizer, (ii) under heavy load situations the anonymizer may become a bottleneck due to receiving many user requests for LBSs, in addition to the continuous monitoring of the location of all the users in the system, (iii) as a trusted component, the anonymizer is a single point of attack: if a malevolent user gains access to the trusted server then the privacy of all the users in the system is compromised, and (iv) the anonymizer requires a large number of trustworthy and cooperate users in order to operate properly. These shortcomings, coupled with the recent advancement in peer–to–peer (P2P) communication technologies, have paved the way for the proposal of distributed approaches to location privacy. In what follows, we review three of the most prevalent P2P solutions, namely *Cloak–P2P*, *PRIVÉ* and *MobiHide*.

Cloak–P2P [40] formulates the ASR by applying a technique that is similar to *Center Cloak*, in a distributed manner. When a user wishes to request an LBS, his/her mobile device has to contact all peers that reside in his/her neighborhood in order to form a $\mathcal{K}$–

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

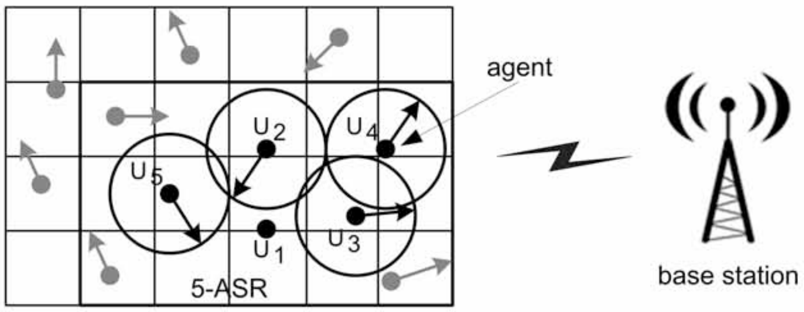


Figure 17.11: An example of *Cloak–P2P*.

ASR. If more than $\mathcal{K}$–1 peers are found near the user, the $\mathcal{K}$–1 ones who are closest to the query source are selected. Otherwise, the process continues in an iterative fashion, with the identified peers issuing requests to all peers in their vicinity in an attempt to enroll more users to the anonymity set of the user. When a total of $\mathcal{K}$–1 peers are enrolled, the user randomly selects one among them and sends him/her the LBS query along with the computed $\mathcal{K}$–ASR. In turn, the selected peer (i.e. the agent) forwards the user query to the LBS service provider through a base station. Based on the computed ASR, the service provider generates a list of candidate answers which are communicated (via the agent) back to the requester. As a final step, the requester uses his/her location to filter out all the false positives and retrieve the correct among the candidate answers.

To formulate the $\mathcal{K}$–ASR, as shown in Figure 17.11, *Cloak–P2P* considers a partitioning of the entire system area in a grid fashion. In the peer searching phase, the mobile user who wishes to request an LBS (here U_1) broadcasts a special type of request. When a peer receives this request it responds by sending its current location and its maximum speed of movement. Assuming $\mathcal{K} = 5$, U_1 will identify the mobile users $U_{2...5}$, which are in his/her neighborhood, and will accordingly form his/her anonymity set. Since the peers keep moving in the system, U_1 uses the information regarding the maximum speed of each of its peers to identify the maximum distance that this peer may have covered from the time of communication to the current time. Accordingly, he/she forms an ASR that contains himself/herselt as well as all his/her $\mathcal{K}$–1 peers, taking into account the maximum distance that they may have covered in the system in the meantime. For example, in Figure 17.11

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

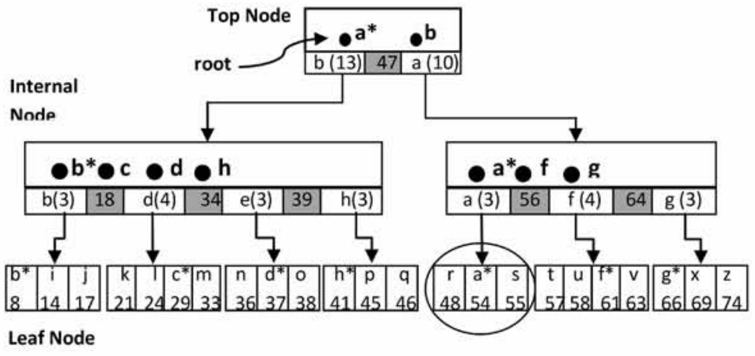


Figure 17.12: The distributed index structure of *PRIVÉ*.

the 5–ASR includes the last row of cells since users U_3 and U_5 may have moved to a cell of this row in the meantime.

PRIVÉ [59] implements the *Hilbert Cloak* strategy [100] in a P2P system for distributed anonymization. The proposed algorithm groups all the users in the system (based on their current locations) into clusters and, within each cluster, it elects a peer to function as the cluster *head* (i.e. the representative of the cluster). Following that, the cluster heads (denoted with a $*$) are grouped recursively to form an annotated B^+–tree, as presented in Figure 17.12. At the top of this hierarchy is a single cluster that we refer to as *top*. The cluster head of the top cluster (here *a*) is called *root*. The top node, as well as each internal node in the B^+–tree, maintains information regarding (i) its cluster head, (ii) the elected cluster heads for its immediate ancestors, along with annotation counters that reflect the number of leaf nodes in the corresponding subtrees (denoted inside parenthesis), and (iii) a set of splitting points (denoted inside gray boxes) which partition the indexed users' locations into buckets based on their Hilbert values (as in regular B^+–trees). In the leaf nodes the index maintains the IP address of each user, as well as his/her Hilbert value that can be used to derive the user coordinates. The size of each cluster of users is bounded between α and 3α, where α is a system–defined parameter (e.g., in Figure 17.12 we have $\alpha = 2$). As users continuously move in the system, the cluster assignments change and the index structure has to be kept updated.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

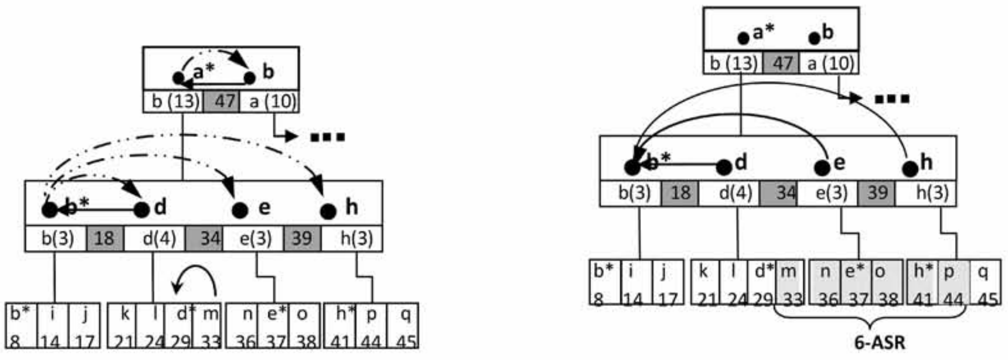
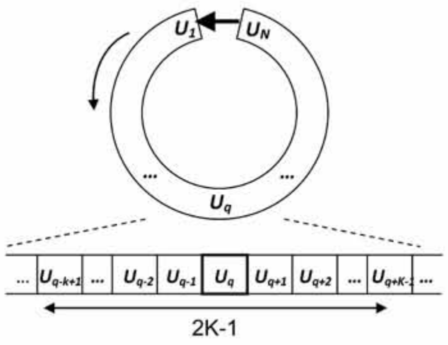

(a) bottom–up traversal. (b) top–down traversal.

Figure 17.13: Computation of a $\mathcal{K}$–ASR in *PRIVÉ*.

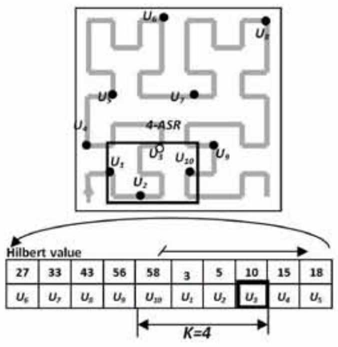
This is the responsibility of the cluster heads who need to be aware of the state of their cluster and make the necessary adjustments based on user movement. Given the index structure, a $\mathcal{K}$–ASR can be computed as shown in Figure 17.13. Assume that user m issues a request for an LBS with an anonymity requirement of $\mathcal{K} = 6$. As a first step, m communicates the request to the head of the cluster in which he/she participates (here d). In turn, the cluster head communicates this request to the head of his/her parent cluster in the clusters hierarchy (here b). The same process continues until the request reaches the top node of the hierarchy (see the solid arrows in Figure 17.13(a)). When the root node (here a) receives the request, it uses the Hilbert value of m's location to compute the regions (i.e. the start point and the end point) of the bucket where m resides (here is $[6\ldots11]$), as in *Hilbert Cloak*. Following that, a *FORM–ASR* request is generated by the root node and transmitted to b who is the head of the cluster where the $\mathcal{K}$ users are sought. When b receives the request it generates three in parallel requests for partial $\mathcal{K}$–ASRs with ranges $[6\ldots6], [7\ldots9]$ and $[10\ldots11]$ and transmits them to d, e and h (respectively). It then awaits to collect the partial ASRs and assemble the $\mathcal{K}$–ASR as their MBR enclosure. Finally, the generated ASR (Figure 17.13(b)) is used for the privacy aware provision of the LBS.

MobiHide [58] provides an alternative implementation of the *Hilbert Cloak* strategy which, unlike *PRIVÉ*, it does not suffer from slow response times. As mentioned earlier, a request in *PRIVÉ* has to ascend all the way from the requesting node to the top cluster in

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136


U_1
U_N
U_q
... U_{q-k+1} ... U_{q-2} U_{q-1} U_q U_{q+1} U_{q+2} ... U_{q+K-1} ...
2K-1

(a) Hilbert sequence ring.


4-ASR
Hilbert value
27 33 43 56 58 3 5 10 15 18
U_6 U_7 U_8 U_9 U_{10} U_1 U_2 U_3 U_4 U_5
K=4

(b) $\mathcal{K}$–ASR construction.

Figure 17.14: An example of *MobiHide*.

the clusters hierarchy, from where the search starts for the identification of the participants of the $\mathcal{K}$–ASR. As an effect, the top nodes in the distributed index hierarchies constitute potential bottlenecks in the operation of *PRIVÉ*. To alleviate from this issue, *MobiHide* uses a more efficient indexing scheme that is based on hierarchical distributed hash tables to arrange users' locations. The new index offers good scalability and increased fault–tolerance, as proven in [159]. To form the ASRs, *MobiHide* considers a ring (instead of an array) of the sequence of all users, ordered by their Hilbert values (see Figure 17.14(a) where U_q is the querying party). Then, it selects $\mathcal{K}$ consecutive users (including the requester) from this structure and computes their MBR enclosure as the region of $\mathcal{K}$–anonymity. To ensure that the constructed ASR will be the same no matter which, among the $\mathcal{K}$ participants of the ASR has requested the LBS (as *Hilbert Cloak* requires), *MobiHide* assumes that a unique offset l is selected for each neighborhood of users. The existence of a common offset per neighborhood, uniquely defines the position of the requester in the computed $\mathcal{K}$–sequence of users and guarantees that all users of an ASR are equiprobable to be the requesters of the LBS. Figure 17.14(b) presents an example of a 4–ASR with an offset of $l = 3$, where U_3 is the requesting party. *MobiHide* generates the 4–ASR (sequence $\langle U_{10}, U_1, U_2, U_3 \rangle$) according to the Hilbert ordering of the users, such that U_3 is the fourth (indexing starts from 0) user in the returned sequence.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 18

Contributions

In Chapter 17 we surveyed the different research directions along with the state–of–the–art methodologies that have been proposed for the offering of privacy in LBSs. As part of this dissertation, we have contributed along two principal lines of research within the $\mathcal{K}$–anonymity direction of privacy in LBSs: *historical $\mathcal{K}$–anonymity* approaches, and *trajectory $\mathcal{K}$–anonymity* approaches. Specifically, along the first line we have proposed two distance–based cloaking strategies (Chapter 20) for the offering of historical $\mathcal{K}$–anonymity in LBSs, which extend the work of Bettini, et al. [26] along several dimensions. Regarding the second line of research, we have proposed two novel approaches for the offering of trajectory $\mathcal{K}$–anonymity both in a networkless (Chapter 21) and in a network aware (Chapter 22) topology of user movement. Finally, we proposed PLOT, an open–ended toolbox (Chapter 23) that allows the implementation and the evaluation of privacy–enhancing algorithms for LBSs. In what follows, we discuss our contributions in each research direction.

18.1 Historical $\mathcal{K}$–Anonymity

To support historical $\mathcal{K}$–anonymity in LBSs, we have proposed two distance–based cloaking approaches which use the historical movement of the users in the system to adequately cover up the requesters of LBSs. The first approach, introduced in [185], extends the exist-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

ing work in the preservation of historical $\mathcal{K}$–anonymity by (i) enabling each user to associate multiple sequences of POIs (and the related time periods) to his/her profile, depicting different itineraries that the user commonly follows in the system and in which his/her privacy has to be protected when requesting LBSs, (ii) adapting the generalization algorithm of [26] to account for the multiple sequences of POIs, (iii) defining a set of novel spatial regions which, when crossed, behave as dynamically constructed mix zones, and (iv) introducing an unlinking algorithm, applied when the generalization algorithm fails, which protects the users by unlinking their future locations from the previous ones. Moreover, as part of this work we constructed a data generator that allowed the composition of spatiotemporal datasets, subsequently used for experimentation. The details of this methodology are presented as part of Chapter 20.

Both [26, 185] assume unconstrained user movement. In this sense, a POI (as defined in [26] and adopted in [185]) corresponds to a place that is frequently visited by a particular user during his/her regular movement (e.g., his/her house, his/her workplace, etc). On this end, a sequence of POIs (along with the related time intervals) simply states that the user is usually in one place within a given time period (remember that times are provided as unanchored time intervals) and then moves to another place within a different time period (e.g., <Home, [8am–9am]>, <Work, [10am–10:30am]>). What it does not state however is *how* the user moved from one POI to another. In a free terrain (unconstrained) environment this makes sense as there are unlimited itineraries that one can follow to go from one location (POI) to another. However, in reality, the different POIs are expected to be landmarks in a map, connected together through a set of roads in a road network. In [66] we acknowledge this limitation in the number of different itineraries that one can follow to visit one place from another, in order to present a more robust modeling of the sequences of POIs. Specifically, in this enhanced modeling, a sequence of POIs can depict not only the different landmarks that the user visits as he/she moves in the road network, but also enables the definition of the exact itinerary that the user follows (as a series of roads or road parts that he/she traverses) in order to move from one location to another. Furthermore, we intro-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

duce a new generalization algorithm and a new unlinking approach that are based on [100] and are proved to be more secure when compared to the ones of [26, 185]. Finally, we push the core functionality of the anonymizer into a spatial DBMS. As the experimental results indicate, by utilizing the spatial capabilities of the employed DBMS, the performance of the anonymizer improves when compared to the existing work in historical $\mathcal{K}$–anonymity. In Chapter 20, we present the details of this approach.

18.2 Trajectory $\mathcal{K}$–Anonymity

Trajectory $\mathcal{K}$–anonymity approaches protect the privacy of the requesters of continuous LBSs by using the current location of all the users in the system as well as their future locations to adequately cover up the trajectory of the requester until the completion of the service provision. This direction of approaches is the most recent one and, to our opinion, it is also the most prevalent due to two reasons: (i) the trajectory privacy approaches protect the privacy of the requester both in services that require a single location transmission as well as in those that require multiple location transmissions in order to be successfully provided, and (ii) the trajectory privacy approaches offer increased privacy guarantees when compared to historical approaches, since they are based on the current location of the users in the system instead of historical movement data. As an effect, trajectory $\mathcal{K}$–anonymity methodologies can operate even when on–site cameras have been placed in the area that is covered by the generated ASRs.

In our work, we improved the state–of–the–art methodologies for the offering of trajectory $\mathcal{K}$–anonymity in LBSs by proposing a new line of research (i.e. *personalized approaches to trajectory $\mathcal{K}$–anonymity* — see Section 17.3.3) along with two novel approaches: a free terrain approach [69] that operates on the centralized HESTIA privacy model of [71], and a network aware approach [79] that operates on the privacy model introduced in [77]. Both approaches consider an attacker who has knowledge of the user movement statistics for each user in the system and can use his/her knowledge of the frequent mobility patterns of each user in order to breach user privacy.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

In [69, 71] we introduce a grid–based methodology that utilizes the collected (by the trusted server) historical movement traces of all the users in the system to build a set of representative mobility patterns for each user. Among the different patterns that can be extracted from the movement history of a user, we are interested in the ones that depict frequent (typical) movement behavior of the user, which happens to be infrequent (atypical) for many other users in the system. Such patterns (as related to the sequences of POIs and their related time intervals) signify locations and times where user privacy is at an elevated risk. By using this knowledge, we employ a grid–based cloaking strategy to protect the users when requesting LBSs from within any location/time that belongs to their mobility patterns. Furthermore, to improve the efficiency of the proposed methodology, we use the functionality that is offered by a modern spatial DBMS to store and manipulate the necessary information. The details of this approach are provided in Chapter 21.

In [77, 79] we propose a privacy model that utilizes an underlying network topology that confines user movement to deliver different levels of user privacy to the requesters of LBSs. Contrary to existing work on privacy in LBSs, our proposed model successfully addresses all of the following real–world challenges: (i) it considers that users typically move in a network constrained environment (e.g., a road network), which allows for certain decisions to be made with respect to the offering of trajectory privacy, (ii) it takes into account that users are in extra need of protection when requesting LBSs from locations/times that are part of their regular mobility patterns; on this end, it offers two anonymization techniques, the $\mathcal{K}$–present (weak) and the $\mathcal{K}$–frequent (strong) trajectory anonymity, based on the location of the user at the time of request, as well as his/her future locations/times until the provision of the service, (iii) it is implemented on top of a spatial database engine which offers a significant portion of the necessary functionality, while provides for its effective deployment. Chapter 22 provides all the details pertaining to this methodology.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

18.3 An Open–Ended Toolbox for Privacy in LBSs

Although many centralized approaches have been proposed so far for the offering of privacy in LBSs, until recently no attempt had been made to integrate these approaches in a unified framework. Such an integration would provide the means for easily implementing and testing new techniques by offering ready–made vanilla system components, as well as allow for both the experimental and analytical evaluation of the implemented methodologies. In [78] we introduce PLOT, the first open–ended toolbox that offers all the essential functionality for the implementation of novel privacy approaches in LBSs, while serving as a testbed for the comparison of existing schemes. PLOT offers a variety of interesting features: (i) it supports both real and synthetic movement data, (ii) it relies on spatial DBMSs to efficiently handle movement data as well as the underlying model of user movement, (iii) it offers tools for mobile data preprocessing, movement reconstruction and segmentation, (iv) it allows the implementation of both network based and free terrain solutions to location/trajectory privacy, (v) it provides the infrastructure for second–chance approaches when the main privacy approach fails, (vi) it implements strategies for the identification of frequent mobility patters in historical user movement data, and finally (vii) it offers an extended set of visualization tools that both provide insight on the workings of the implemented solutions and facilitate the qualitative and quantitative evaluation of their behavior. PLOT is presented in Chapter 23.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 19

Background and Terminology

In this chapter we give all the necessary background along with the terminology that is essential for the understanding of our work (i.e. [66, 69, 78, 79, 185]), as provided in the chapters that follow. To stress the similarity that exists among the underlying principles of our different contributed methodologies, in what follows, we have unified their theories and presented them under a common framework. Furthermore, to enhance the clarity of presentation, we have partitioned the presentation of the terminology into four sections: (i) basics of user mobility, (ii) basics of network modeling, (iii) extraction of user mobility patterns to model privacy risks, (iv) handling of requests for LBSs, and (v) provision of alternative strategies for $\mathcal{K}$–anonymity. Specifically, in (i) we provide all the necessary background for the modeling of user movement and discuss the alternative representations that can be employed to lead to significant conclusions about user mobility. In (ii) we highlight the basic principles of modeling user movement in the regions of an underlying network topology, such as an urban road network. Then, in (iii) we provide the necessary terminology that pertains to the identification of a set of user mobility patterns based on user movement. As we will demonstrate in the following chapters, these useful patterns can enable the privacy algorithm to take important decisions with respect to the type of privacy that has to be offered to a mobile user, when requesting an LBS. In (iv) we define the notion of a user request for an LBS and we present the necessary background for the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

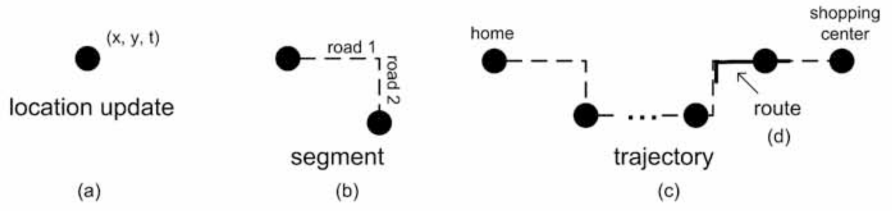


Figure 19.1: The building blocks of the user history of movement.

matching of user requests to the mobility patterns of the requester. Finally, in (v) we define some alternative strategies that are part of our proposed methodologies for the offering of $\mathcal{K}$–anonymity in LBSs.

19.1 Basics of User Mobility

Let u denote a user on the move. A *location update* is a tuple $\langle u, x, y, t\rangle$ stating that user u was located at (x, y) at time t, where t is a detailed description of the time when the recording took place (Figure 19.1(a)). The trusted server (anonymizer) collects for each user that is subscribed to the telecom operator, his/her sequence of transmitted location updates, which depict the *history of movement* $\mathcal{H}$ of this user in the system. The history of movement of a user is formally defined as follows:

Definition 6. ***(History of movement)*** *The history of movement $\mathcal{H}(u)$ of a user u is the sequence of location updates $\langle u,x,y,t\rangle$ that the user transmitted to the trusted server during his/her movement in the system. We consider time t to be depicted in the form: $\langle$Date, Hour, Minutes, Seconds$\rangle$, i.e. corresponding to a detailed description of the time when the recording was done.*

As it can be observed from Definition 6, the history of movement of a user reflects the *whole* movement of this user in the system, as it was recorded by the trusted server. The history of movement of a user cannot be directly used to draw any significant conclusions regarding the movement habits of the user. For this reason, the movement history of each

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

user is decomposed into smaller blocks, the user trajectories, defined as follows:

Definition 7. *(**Trajectory**) A trajectory $\mathcal{T}(u)$ of a user u is part of the movement history of this user (i.e. $\mathcal{T}(u) \subset \mathcal{H}(u)$) that consists of a sequence of location updates, beginning at some instance $(x_{start}, y_{start}, t_{start})$ and terminating at another instance $(x_{end}, y_{end}, t_{end})$, and containing all the intermediate location updates of $\mathcal{H}(u)$.*

Figure 19.1(c) presents an example of a trajectory that was extracted from the history of movement of a user. A user trajectory denotes the evolution of the position of the user (perceived as a point) when moving in space during a given time interval in order to accomplish a certain application–specific goal. In the case of Figure 19.1(c), the goal of the user was to go to the shopping center from his/her residence. Apart from providing detailed information regarding the different itineraries that were followed by a particular user, user trajectories are also the basic construct for the identification of regular (frequent) mobility patterns of the users in the system. In Section 19.3 we shed light on this important aspect.

The decomposition of the history of movement of a user into a set of trajectories, takes place by identifying stops that denote the immobility of the user for a sufficiently large period of time, indicated by the system parameter t_{stop} (e.g., half an hour). Having identified such stops, *the user history of movement that is bounded by two consecutive stops, corresponds to a trajectory*. The time period of a stop is properly adjusted to avoid accounting for delays that are too short (i.e. the stop of a car in a road intersection or in the traffic lights) and may cause unwanted segmentation. The decomposition of the user history of movement into user trajectories is application–specific, meaning that different applications may require a different value of t_{stop} to handle different types of moving objects (e.g., pedestrians, cars, airplanes) and generate meaningful trajectories.

Movement reconstruction

Definitions 6 and 7 capture user movement as a series of transmitted location updates, which provide information regarding the location of the user at a set of discrete time instants. However, in reality, movement is a continuous function in the time domain, stating

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

not only the locations from which the user has passed by, but also *how* he/she moved (i.e. the followed itinerary) from one location to another. Assuming that we have some background knowledge regarding the movement of the users in the system (e.g., we are aware of an underlying road network inside which they all move), it is possible to reconstruct (i.e. to approximate to a reasonable extent) the user movement from the series of transmitted location updates. The reconstructed user movement provides us with a vital source of information for computing the regular (frequent) or irregular (infrequent) movement of the users in the system. In what follows, we will use notation $\hat{\mathcal{H}}$ and $\hat{\mathcal{T}}$ when we consider that the history of user movement or the user trajectories (respectively) are reconstructed. We will refer to specific algorithms for the reconstruction of user movement both in a free terrain and in a network aware environment, as part of Chapters 21 and 22.

A reconstructed trajectory $\hat{\mathcal{T}}(u)$ or a user u consists of a series of segments, where each *segment* connects two consecutive location updates in the trajectory and it defines the itinerary that the user has followed from one location update to the next one (Figure 19.1(b)). Any part of a reconstructed user trajectory, independently of its start point and end point (i.e. it does not have to be a location update), is called a *route* (Figure 19.1(d)).

19.2 Basics of Network Modeling

Network aware solutions to $\mathcal{K}$–anonymity in LBSs (such as [66,79]), assume the existence of an underlying (spatial) network $\mathcal{N}$ that provides a reasonable abstraction of the actual network topology where users move (e.g., a road network in an urban area). In this section, we provide some details that pertain to the modeling of the real network topology of user movement as a graph $\mathcal{N}(V,E)$ in which the users are assumed to be moving.

Definition 8. ***(Spatial network)*** *Given a ground plan of the area to be modeled, we define a spatial network as a graph $\mathcal{N}(V,E)$ that has the following properties: (a) for each roads' intersection there exists a node $n \in V$ in the graph that is a point with a longitude and a latitude as defined by the center of the intersection, (b) each POI is captured as a geometrical*

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

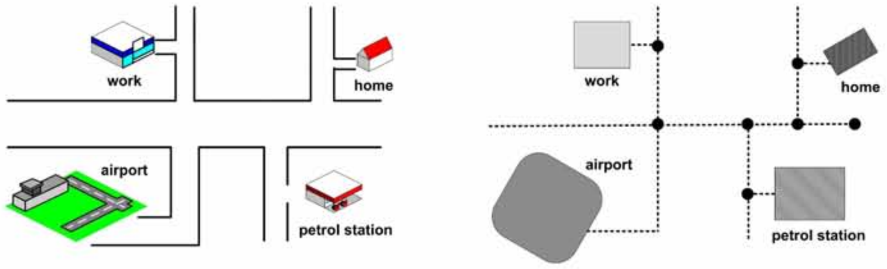


Figure 19.2: Modeling a road network along with a set of landmarks as a graph.

object in $\mathcal{N}$ that simplifies the real construction, and (c) an edge $e \in E$ of the graph models a road (or a road part) that allows user movement from one node to another, as well as it provides accessibility to the POIs.

For the purposes of Definition 8, a POI is considered to be a place, such as a house, a museum, an airport etc., from which the user can request an LBS. Figure 19.2 presents an example of a modeled neighborhood with four POIs. The equivalent spatial network $\mathcal{N}$ for this neighborhood (presented on the right of this figure) consists of a set of nodes, one per roads' intersection and a set of edges that designate the allowable movements.

Definition 9. *(**Network path**) A* path *in the network $\mathcal{N}$ is a sequence $S = \{v_1, e_1, v_2, e_2, \ldots, v_{n-1}, e_{n-1}, v_n\}$ of distinct vertices v and edges e from $\mathcal{N}$ such that for each $v_i, v_{i+1} \in S$ we have that $e_i = (v_i, v_{i+1}) \in S$, i.e. from each vertex $v_1, v_2, \ldots, v_{n-1} \in S$ there is an edge $e_1, e_2, \ldots, e_n \in S$ to the next vertex in the sequence, with v_n being the final vertex (i.e. the destination) in S.*

19.3 Extracting Mobility Patterns to Model Privacy Risks

Mobility data mining, as introduced in Part II of this dissertation as well as surveyed in detail in Ref. [62], is an emerging research direction that allows the extraction of useful patterns that summarize movement data at an appropriate scale and granularity. In the regions of this part of the dissertation, we are interested in patterns that are related to the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

regular movement behavior of each user in the system. Such a movement pattern can be defined as follows:

Definition 10. ***(Mobility pattern)*** *A* mobility pattern $\mathcal{P}(u)$ *of a user u consists of a sequence of spatiotemporal elements* E_j *of the form* $E_j = \langle Region, TimeInterval \rangle$*, where each Region is either a regularly visited POI or a route that user u frequently follows in the system, and the time interval represents the approximate unanchored time of this visit. Mobility patterns represent the regular (frequent) movement habits of a user u that have a pre–specified periodicity of occurrence.*

A POI in a mobility pattern is captured as a geometrical object that abstracts the real construction (as shown on the right part of Figure 19.2), while a route is modeled as an edge that defines the movement of the user in the network, from one node to another. Furthermore, tight to each Region that appears in a user mobility pattern, is a time interval that represents unanchored instants of time. For example, assuming that we are mining events that have a time–of–day periodicity, pattern $\mathcal{P}(u) = \langle$Central Highway, [5pm, 6pm]$\rangle, \langle$Home, [6pm, 7pm]$\rangle$ simply states that user u is frequently located in the Central Highway at sometime between 5pm and 6pm and then at his/her home sometime between 6pm and 7pm, on a daily basis. Please notice that "Central Highway" is a route of user u corresponding to a path of network $\mathcal{N}$, whereas "Home" is a POI. The corresponding time intervals are unanchored in the sense that the time is not tight to a specific time interval on the timeline but rather it involves an infinite set of intervals, one for each day (i.e. considered period).

As stated earlier, the users' trajectories allow us to expose some interesting knowledge for each user in the system by tracking regularities in his/her movement. We argue that the knowledge of these regularities by an adversary (such as the service provider) can be used to breach user privacy when the user requests LBSs, even if the adversary has no other user identification data at his/her disposal. To motivate this argument, think of a scenario where a user goes by certain city areas when he/she commutes to work in more or less the same times in weekdays. This frequent behavior of the user can lead to a possible identification simply by matching either the origin of his/her trajectory to some public domain

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

geocoded information for the house (e.g., through the white pages), or the destination to the location of the business facility where this user works. Furthermore, a possible matching of a series of location updates with a frequently traveled route of a user may also easily lead to his/her identification. Frequent movement behavior is captured as either safe or unsafe with respect to user privacy, based on whether it is also ordinary or atypical for a significantly large number of nearby users. In what follows, we formally define the notion of a frequent route, as well as the notion of a safe and an unsafe route.

Definition 11. ***(Frequent route)*** *A route of a user u is defined as* frequent *if it appears among the trajectories of this user a number of times that is larger than a minimum frequency threshold* freq. *A frequent route is a mobility pattern in which the Region corresponds to a route of the user.*

The number of times that the route appears in the trajectories of user u is called the *frequency* of the route. Furthermore, any route that is not frequent is called *infrequent*. Directly related to the notion of a frequent route are the definitions of a safe and an unsafe route, provided as follows:

Definition 12. ***(Safe route)*** *A route of user u is defined as* safe *if it is frequent for user u and also frequent for at least $\mathcal{K}$–1 other users in the system.*

Similarly, a route of user u is considered as *unsafe* if it is frequent for user u and also frequent for at most $\mathcal{K}$–2 other users in the system.

19.4 Requesting Location Based Services

In what follows, we define the notion of a user request for an LBS and then we provide definitions for the notion of a match between a user request (or a set of user requests) and a mobility pattern of the user.

Definition 13. ***(Request)*** *A user request $\mathcal{R}_i$ for an LBS is a tuple in the form of $\mathcal{R}_i = \langle u, sid, x_i, y_i, t_i, data \rangle$, where u is the requester, sid is the identifier of the requested service,*

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

(x,y) is the location of user u at the time of request t, and data is any necessary service–specific data.

Definition 14. ***(Matching a request to a mobility pattern)*** *If (x_i, y_i, t_i) is the exact location and time of a request $\mathcal{R}_i$ that was received by the trusted server, $\mathcal{R}_i$ matches a spatiotemporal element E_j of a mobility pattern $\mathcal{P}(u)$ of user u if and only if $(x_i, y_i) \in E_j$.Region and $t_i \in E_j$.TimeInterval.*

Definition 15. ***(Matching a set of requests to a mobility pattern)*** *A set of requests $\mathcal{R}$ sent from the same user u match one or more of the user's mobility patterns $\mathcal{P}(u)$ if each request $\mathcal{R}_i \in \mathcal{R}$ matches a spatiotemporal element $E_j \in \mathcal{P}(u)$ (and vice versa).*

19.5 Alternative Strategies for $\mathcal{K}$–Anonymity

In this section we provide the definitions of the alternative strategies that we have considered in our proposed methodologies for the offering of $\mathcal{K}$–anonymity in LBSs. In all our proposed approaches, the type of $\mathcal{K}$–anonymity that is offered to the requester of an LBS depends on the location and the time when the user requested the LBS (in relation to his/her extracted mobility patterns), as well as his/her consecutive locations and times until the provision of the service. To provide the requester with $\mathcal{K}$–anonymity, the original user request $\mathcal{R}_i = \langle u, \text{sid}, x_i, y_i, t_i, \text{data} \rangle$ (as well as each subsequent location update that is required for the offering of the requested LBS) has to be $\mathcal{K}$–anonymized prior to being forwarded to the appropriate service provider for servicing. This is achieved by the transformation of the exact user location and time (x_i, y_i, t_i) that is part of the location update of the user into a spatiotemporal area $(A_{\mathcal{R}_i}, T_{\mathcal{R}_i})$ where $(x_i, y_i) \in A_{\mathcal{R}_i}$ and $t_i \in T_{\mathcal{R}_i}$. This transformation is called *cloaking*, the new request is known as the *cloaked* request, and the area $A_{\mathcal{R}_i}$ is called the anonymity spatial area (or ASR). In the case of the historical approaches to $\mathcal{K}$–anonymity, as proposed in [66, 185], the following definition summarizes when the state of $\mathcal{K}$–anonymity is achieved for a user that sends a set of requests $\mathcal{R}$ to the trusted server:

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Definition 16. ***(Historical $\mathcal{K}$–anonymity)*** *When a user u sends a set of requests $\mathcal{R}$ to the trusted server, we say that historical $\mathcal{K}$–anonymity is achieved, if and only if there exist $\mathcal{K}$–1 users (other than u) for whom the histories of movement $\mathcal{H}(u_1), \mathcal{H}(u_2), \ldots, \mathcal{H}(u_{\mathcal{K}-1})$ are such that each $\mathcal{H}(l)$ $(l \in [1, \mathcal{K}-1])$ is* location and time consistent *with $\mathcal{R}$.*

Definition 17. ***(Location and time consistency)*** *The history of movement $\mathcal{H}(u)$ of a user u is location and time consistent with a set of p requests $\mathcal{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_p\}$ if and only if for each request $\mathcal{R}_j \in \mathcal{R}$ $(1 \leq j \leq p)$ there exists an element $\langle u, x_i, y_i, t_i \rangle$ in $\mathcal{H}(u)$ such that $(x_i, y_i) \in A_{\mathcal{R}_j}$ and $t_i \in T_{\mathcal{R}_j}$.*

As one can notice, the only condition that must be fulfilled by the $\mathcal{K}$–1 users in order to offer historical $\mathcal{K}$–anonymity to the requester of an LBS is that they have passed by the location from where the request was sent sometime in the past. On the other hand, for the trajectory privacy approaches that are proposed in [69, 71, 77, 79], the following definitions apply with respect to the provision of $\mathcal{K}$–anonymity:

Definition 18. ***(Anonymity set)*** *Given a user u sending a request $\mathcal{R}(u)$ for the provision of an LBS, we define the anonymity set of u to be (at least) $\mathcal{K}$–1 other users that are close to u at the given point in time and who could have initiated this request.*

Definition 19. ***($\mathcal{K}$–present anonymity)*** *A user u, when submitting a request $\mathcal{R}(u)$ for an LBS, is $\mathcal{K}$–present anonymous if each participant of his/her anonymity set could be a potential issuer of the request from within an area that is near to the requester and within a time period that is close to the time of the request.*

Definition 20. ***($\mathcal{K}$–frequent anonymity)*** *A user u, when submitting a request $\mathcal{R}(u)$ for an LBS, is $\mathcal{K}$–frequent anonymous if each participant of his/her anonymity set shares the same frequent route with user u and passes by near the location of the requester at a time period that is close to the time of the request.*

As one can notice, in trajectory $\mathcal{K}$–anonymity approaches to privacy in LBSs, the anonymity set of the requester if formulated based on the current (instead of the historical) locations of all the users in the system.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 20

Offering Historical $\mathcal{K}$–Anonymity in LBSs

Historical $\mathcal{K}$–anonymity for the offering of privacy in LBSs, was introduced in [26]. As part of this work, the authors presented a "preliminary investigation on the privacy issues involved in the use of LBSs" and proposed the use of generalization along with unlinking for the preservation of historical $\mathcal{K}$–anonymity. Although the proposed methodology sets a new perspective on the offering of privacy in LBSs, it suffers from several shortcomings as (i) although it discusses the principles of unlinking, it fails to provide a specific algorithm that would team up with the proposed generalization approach to better protect the privacy of the requester, (ii) it lacks the reporting of any experimental results that would shed light on the degree of privacy that is offered by the proposed generalization strategy, (iii) it is based on the rather unrealistic assumption that there exists only one mobility pattern per user capturing the locations/times in which the privacy of this user is under threat when requesting LBSs, and (iv) it assumes that the unlinking algorithm operates on statically defined mix zones, which means that it achieves to protect only those users who request LBSs when moving towards a mix zone, leaving the rest of the population unprotected.

To deal with the shortcomings of [26], we have proposed two approaches. The first approach [185] extends the current work in historical $\mathcal{K}$–anonymity, along two directions:

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

(i) we allow the trusted server to store multiple mobility patterns per user that serve as a fingerprint for the user, in the sense that knowledge regarding a person that matches a given mobility pattern leads to this specific user with high probability of accuracy; we properly adapt the generalization strategy of [26] to account for this fact, and (ii) we provide a novel unlinking algorithm to protect users by unlinking their future requests from the previous ones. For the successful application of the unlinking algorithm, we define a set of spatial regions that behave as dynamically constructed mix zones and allow, when these areas are crossed, for the secure change of the assigned user IDs. Moreover, we evaluate our approach by conducting experiments on synthetic datasets, composed through a spatiotemporal data generator that we created. Section 20.1 presents the details of this approach.

Both [26] and [185] are based on the assumption of a free terrain model of movement, where the movement of the users in the system is unconstrained. Our second approach, presented in Section 20.2 as well as in [66], drops this assumption by considering an underlying network of user movement. A network aware approach to the preservation of historical $\mathcal{K}$–anonymity has an important advantage: it enables users to capture not only the sequences of POIs (along with the corresponding time intervals) where their privacy is under threat, but also to explicitly state which itineraries (i.e. paths in the network) are hazardous to their location privacy. Since there are typically many different paths in a network that one can follow to move from one POI to another, a network aware approach achieves to explicitly define which, among the alternative paths, poses a hazard to the privacy of a particular user. In addition to this, our network aware approach for the preservation of historical $\mathcal{K}$–anonymity pushes the core functionality of the anonymizer into a spatiotemporal DBMS, which offers significant advantages over add–hoc implementations. Section 20.2 provides the details of this approach.

20.1 A Free Terrain Approach to Historical $\mathcal{K}$–Anonymity

In this section, we present our proposed methodology for the offering of historical $\mathcal{K}$–anonymity in the regions of a free terrain model of user movement. Our methodology

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

consists of two basic algorithms: a *generalization* algorithm and an *unlinking* algorithm. Each of these algorithms modifies the location and the time information included in a user request so that the exact user location is not disclosed to the service providers. Both algorithms are assumed to be executed in the environment of a trusted server, just before the request is forwarded to the appropriate service provider. The rest of this section is organized as follows: In Section 20.1.1 we present our solution methodology, along with the extended generalization algorithm and the proposed unlinking algorithm. Section 20.1.2 presents the movement data generator that we constructed in order to create the synthetic datasets that were subsequently used in the experiments, while Section 20.1.3 contains a thorough set of experiments for the evaluation of the proposed methodology.

20.1.1 Solution Methodology

The goal of the generalization and the unlinking algorithms is to ensure that no sensitive data is revealed when a user request reaches the (untrusted) service providers. Similarly to [26], in the regions of this approach, we consider that a user is in need of protection only when requesting an LBS from within one of his/her mobility patterns and for as long as the requested service withstands completion. The mobility patterns of all the users in the system are assumed to be user–defined, i.e. each user supplies these patterns to the trusted server which are then stored in a profile that the user has in the system. Furthermore, the user–defined mobility patterns consist of spatiotemporal elements of the form $\langle$POI, TimeInterval$\rangle$, while elements of the form $\langle$Route, TimeInterval$\rangle$ are not supported. This is a reasonable assumption, since in a free terrain model of movement there exist infinite ways (i.e. itineraries) to go from one POI to another.

When the trusted server receives a user request for an LBS, it employs the generalization algorithm to check if the request matches an element of a mobility pattern of the requester. If such a match occurs, the request is properly modified and then forwarded to the service provider for servicing. On the other hand, if no such element exists, then the request was sent from a location and time which cannot be linked to the requester. There-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

fore there is no need for the request to be modified. Instead, the identity of the requester is just removed from the request and the request is then forwarded to the appropriate service provider, without any further changes.

In case the modification of the spatiotemporal properties of a user request by the generalization algorithm fails, the unlinking algorithm is employed to ensure that future requests coming from the same requester will not be linked to his/her previous ones, thus reveal his/her identity. To achieve this goal, the unlinking algorithm has to change the user ID of the requester and to make all the necessary adjustments such that the requester appears in the system as a new user. In this way, the location privacy of the requester is guaranteed.

The Generalization Algorithm

The purpose of this algorithm is to generalize the information about the time and the place from where the actual request was sent, to a time interval and an area (respectively) where $\mathcal{K}$–1 other users could have been the requesters of the same LBS. As part of this work, the generated ASR is considered to have the shape of a square. However, any other type of region (i.e. rectangular, polygon, etc) can also be supported. The generalization process takes into account two types of constraints when forming the $\mathcal{K}$–ASR. The first type of constraints involves the size of the ASR and ensures that it is within reasonable bounds so that the LBS can be successfully provided to the requester. The second type of constraints limits the search in the histories of user movement when identifying the participants to the anonymity set of the requester. The generalization algorithm consists of two parts:

1. The *first–element matching* part, and

2. The *intermediate–element matching* part.

The first part is executed in two cases: (i) when a user sends a request for the first time, or (ii) when the current request has to match the first element of one or more mobility patterns of the requester. The *first–element matching* part creates a list L which contains all the mobility patterns of the requester which were not matched (partially or at all) with

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

one of his/her previous requests. The current request has to match the first element of one or more mobility patterns which are members of list L. We must highlight that L may contain mobility patterns which have completely matched a series of requests from specific users in the past. These patterns can be members of L, because there is a high probability that the specific requester may follow the same route (which is represented by the specific mobility pattern) in the future. After this list is created, this part checks whether the current request matches one or more of the user's mobility patterns. If no mobility patterns are found to match, this means that the request was sent from a location which can not harm the privacy of the requester. On the other hand, if at least one mobility pattern is identified, this part generalizes the spatial and the temporal data of the request (according to the given constraints) until $\mathcal{K}$–1 neighbors are found. Then, the generalization algorithm ends and the identifiers of the members of the $\mathcal{K}$-anonymity set, as well as the generalized 3D area (where these users were located) are returned as output. If no more generalization can be applied to the request and less than $\mathcal{K}$–1 neighbors were located, then the generalization algorithm has failed and the unlinking algorithm has to be executed.

The *intermediate–element matching* part takes as input the enforced generalization constraints and the identifiers of the $\mathcal{K}$–1 neighbors, located during the previous call of the *first–element matching* part. It then tries to match the current request with the "proper" intermediate element of one or more of the requester's mobility patterns, which are members of a list named P. List P contains all the requester's mobility patterns which are not members of the — previously mentioned — list L. These patterns must have partially matched in the past one or more of the user's requests. By "proper" we mean that if a previous request $\mathcal{R}_i$ of the requester has matched the element E_j of one or more of the his/her mobility patterns, then the current request has to match the element E_{j+1} of these patterns. If the current request does not match the "proper" elements, then the *first–element matching* part is called because the requester may have began a completely new itinerary. If the matching is successful, then this part generalizes the spatiotemporal area from where the request was sent, until $\mathcal{K}$–1 different points in the neighbors' histories of movement, which are space

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 20.1 The generalization algorithm for historical $\mathcal{K}$–anonymity.

Input: The location of request (x_i, y_i, t_i), the desired degree of anonymity $\mathcal{K}$, the temporal constraints *timeCons*, the spatial constraints *spaceCons*, a list *k_neighbors* of the identifiers of the $\mathcal{K}$ nearest neighbors.

Output: The generalized 3D area, a boolean value (*k_anonymity*) representing the outcome of generalization, the list *k_neighbors* (when the *first–element matching part* is executed).

```
1: function GENERALIZE($\mathcal{R}_i$, $\mathcal{K}$, timeCons, spaceCons, k_neighbors)
2:     if list k_neighbors $\neq \varnothing \wedge \mathcal{R}_i$ matches the "proper" element(s) of one or more mobility patterns
       (which are members of list matched_patterns) then          ▷ The intermediate–element matching part.
3:         while constraints spaceCons and timeCons are not violated $\wedge$ less than $\mathcal{K}$–1 users are found do
4:             Generalize $\mathcal{R}_i$ and store the new coordinates in the array 3D_area.
5:             for each user in list k_neighbors who is not marked do
6:                 Search in his/her history of movement for points which belong in the 3D_area.
7:                 Mark this user.
8:         if $\mathcal{K}$–1 users are found then
9:             k_anonymity ← true.
10:            return (k_anonymity, 3D_area)
11:        else
12:            k_anonymity ← false.
13:    else                                                        ▷ The first–element matching part.
14:        new_patterns ← all the mobility patterns of the requester except those in matched_patterns.
15:        if $\mathcal{R}_i$ matches the first element of a mobility pattern in the list new_patterns then
16:            Store the mobility pattern in the list matched_patterns.
17:            while spaceCons and timeCons are not violated $\wedge$ less than $\mathcal{K}$–1 users are found do
18:                Generalize $\mathcal{R}_i$ and store the new coordinates in the array 3D_area.
19:                for each user who is not marked do
20:                    Search in his/her history of movement for points which belong in the 3D_area.
21:                    if such points are found then
22:                        Add his/her identifier in the list k_neighbors.
23:                        Match this user.
24:            if $\mathcal{K}$–1 neighbors are found then
25:                k_anonymity ← true.
26:            else
27:                k_anonymity ← false.
28:    return (k_anonymity, k_neighbors, 3D_area)
```

and time consistent with the current request, are identified. If the spatial and temporal data can not be generalized any more and there were found less than $\mathcal{K}$–1 different points, this means that the generalization algorithm has failed and unlinking has to take over.

Algorithm 20.1 shows the details of the enforced generalization approach. The two basic steps of this algorithm are in lines 2 and 13. Lines 14–28 represent the body of the *first–element matching* part. In line 14 we create a list called *new_patterns* which contains all the mobility patterns of the requester except those which have partially matched with previous requests. In line 15 we check if the current request matches the first element of a mobility pattern in the list *new_patterns*. If at least one mobility pattern is found, we store it in a list called *matched_patterns*. The mobility patterns in this list will be subsequently

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

used in the *intermediate–element matching* part. Otherwise, i.e. if none of the requester's mobility patterns matched the current request, this means that the request was not sent from a location which is dangerous for the privacy of the user. Therefore, there is no need for the request to be modified and the generalization algorithm terminates. In lines 17–23 we generalize the space and the time from where the request was initiated until we reach the time–space boundaries (i.e. through the enforced constraints) or we locate the necessary amount of neighbors. Finally, in line 24 we check if the desired number of members that will participate in the anonymity set of the requester were found. If this holds, then the generalization algorithm has succeeded and we return the identifiers of the neighbors (list *k_neighbors* as well as the 3D area in which they were found. Otherwise, the generalization algorithm fails and the unlinking algorithm has to be executed.

Lines 2–12 represent the body of the *intermediate–element matching* part, when the request has to match the "proper" intermediate element of one or more mobility patterns of the requester. This check takes place in line 2. If at least one mobility pattern in the list *matched_patterns* matches the request, then the algorithm proceeds to line 3. Otherwise, we have to check if the request matches the first element of one of the requester's mobility patterns and thus we move to line 13. In lines 3–7 we generalize the space and the time of the request until we identify $\mathcal{K}$–1 different points in the histories of movement of the other users in the system — one for each neighbor of the requester — or until we reach the provided generalization constraints. In line 8, we check if $\mathcal{K}$–1 different points (users) were found during generalization. If this is true, then the generalization algorithm has succeeded and we return the 3D area (enclosure) in which those points were found. Otherwise, the generalization algorithm has failed and we must execute the unlinking algorithm.

The Unlinking Algorithm

The unlinking algorithm is executed only when the generalization algorithm fails. It tries to change the requester's identifier in a way that none of his/her future requests can be linked with the old ones (at least with a high probability). The above modification takes

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

place whenever the requester passes through a dynamically defined mix zone [25]. We will call such zones as *dzones*. Relative to mix zones, the dzones are defined as dynamically identified physical areas, where services are temporarily disabled to allow for the renewal of user identities. Similarly to the mix zones, their basic characteristic is that (at least ideally) there is no connection between the users who entered the zone and those who have exited it because of the change of their identifiers; the provided identifiers on the exit, are completely new. The primary condition which must occur in order for the dzones to protect the privacy of the users, is that there exists a sufficient number of users within the dzone.

In our proposed methodology, the dzones are defined on demand, whenever the generalization algorithm fails. The dynamic construction of the dzones enables the unlinking algorithm to protect the privacy of the requester no matter what his/her location is at the time of request. Suppose that a request was initiated from a location with spatiotemporal coordinates (x,y,t) and that *spaceCons* and *timeCons* are the given spatial and temporal constraints respectively in the generalization algorithm. We consider a dzone to be a square area whose limits are defined by the following points: $(x+2*spaceCons, y+2*spaceCons)$, $(x+2*spaceCons, y-2*spaceCons)$, $(x-2*spaceCons, y+2*spaceCons)$, and $(x-2*spaceCons, y-2*spaceCons)$. The above assumption about the size of the dzone was made since the limits of the generalization algorithms were defined by the points: $(x+spaceCons, y+spaceCons)$, $(x+spaceCons, y-spaceCons)$, $(x-spaceCons, y+spaceCons)$, and $(x-spaceCons, y-spaceCons)$. Since the generalization algorithm was unable to identify the desired number of neighbors in the above area, we considered the dzone to be a larger area (its size depends on the value of the constraint *spaceCons*), which includes the area where the generalization algorithm searched. In this way the possibility of the unlinking algorithm to fail and subsequently the requester's privacy to be jeopardized, is becoming smaller. Moreover, at least $\mathcal{K}$–1 different users (excluding the requester) must have crossed the zone in the time period $[t-2*timeCons, t+2*timeCons]$ in order for the unlinking to take place. We should mention at this point that while the unlinking algorithm is executed, all services in the selected region (defining the dzone) are postponed.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 20.2 The unlinking algorithm for historical $\mathcal{K}$–anonymity.

Input: The temporal constraints *timeCons*, the spatial constraints *spaceCons*, the number of users $\mathcal{K}$ sought in the dzone.
Output: A boolean value for variable *k_unlinkability* representing the outcome of unlinking.

```
1: function UNLINK(timeCons, spaceCons, $\mathcal{K}$)
2:     while the new generalization constraints of the dzone have not been violated $\wedge$ less than $\mathcal{K}$ users exist in the dzone do
3:         Generalize the spatiotemporal information of the request and store them in an array 3D_area.
4:         Count the number of new users who have entered the dzone after the generalization.
5:     if less than $\mathcal{K}$–1 users (excluding the requester) were found in the dzone $\vee$ the current request matched the last element of one or more of the requester's mobility patterns then
6:         k_unlinkability $\leftarrow$ false.
7:     else
8:         k_unlinkability $\leftarrow$ true.
9:         Change the user ID of the requester and delete his/her history of movement as well as his/her mobility patterns.
10:     return (k_unlinkability)
```

The unlinking algorithm may fail in two cases: either (i) less than $\mathcal{K}$ users crossed the dzone in the desired time period, or (ii) the algorithm was called when the current request matched the last element of one or more of the requester's mobility patterns. In the latter case, the algorithm fails because the user's previous requests matched one or more elements with his/her old identifier. If the requester's identifier changes when his/her request matches the last element of one of his/her mobility patterns, i.e. when the requester has reached the end of this movement pattern, it is very possible that his/her privacy is threatened (even with a new identifier) because in the future the client is very likely to follow the same itinerary. If an attacker monitors all users, then he/she can easily find out that the old and the new identifier refer to the same individual and in this way learn sensitive data. On the other hand, if none of the above conditions are met, the unlinking algorithm succeeds. In such a case, a new identifier is supplied to the requester and his/her mobility patterns as well as his/her history of movement are deleted from the trusted server. Otherwise, the unlinking algorithm fails and the privacy of the requester is at risk.

Algorithm 20.2 presents in detail the enforced unlinking approach. This algorithm is executed only if the generalization algorithm fails to protect the privacy of the requester. Lines 2–4 generalize the user request until $\mathcal{K}$ users (including the requester) are found inside the dzone, or until the new spatiotemporal constraints are not violated. In line 5, the algorithm checks if less than $\mathcal{K}$ users were found or if the current request matched the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

last element of one or more of the requester's mobility patterns. If at least one of these conditions is true, this means that the unlinking algorithm has failed and that the privacy of the requester is at an elevated risk. On the other hand, if none of these conditions is true, then the algorithm has succeeded. In order to unlink the requester's future requests from the previous ones, the unlinking algorithm changes the requester's identifier and deletes his/her mobility patterns as well as his/her history of movement from the trusted server (line 9). The outcome of the unlinking algorithm is returned in line 10.

20.1.2 The Movement Data Generator

To experimentally evaluate the proposed methodology, we implemented a data generator that simulates real users' movement and produces a set of records depicting users' locations at pre–specified time periods. These records constitute the histories of users' movement in the system, used later on to conduct the experiments. Users are assumed to move randomly within a bounded square area, whose size determines the possibility that a sufficient number (related to the value of $\mathcal{K}$) of users lie near the requester at sometime near the time of request. Moreover, the initial positions of the users are randomly selected and the users begin to move simultaneously. To make the movement of the users more realistic, the day is divided into a set of time zones, and probabilities are used to control users' movements within each zone. By taking into consideration these zones (as well as the probability that a user moves during each of them), a set of random positions are generated for each user. Each user can move *forward*, *backward*, *left*, or *right*. The speed of movement is also randomly selected. Once the histories of movement of the various users in the system are formed, the generator produces a set of requests, all equally distributed within a pre–specified time period. The locations and times of the requests are randomly chosen, and to prohibit the production of a large number of requests in a small time interval, the generator avoids producing more than 5 requests within an hour.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 20.1: The considered mobility patterns of the requester.

ID	Elem	Contents	Periodicity
1	1	$< \{(800,345),(805,350)\},[07:00,09:00] >$	*3.Days*2.Weeks*
1	2	$< \{(820,358),(830,363)\},[09:00,10:00] >$	*3.Days*2.Weeks*
1	3	$< \{(835,360),(840,370)\},[10:00,12:30] >$	*3.Days*2.Weeks*
1	4	$< \{(850,355),(855,360)\},[12:30,14:30] >$	*3.Days*2.Weeks*
2	1	$< \{(740,300),(750,310)\},[12:00,14:00] >$	*5.Days*1.Week*
2	2	$< \{(760,300),(780,310)\},[14:00,18:00] >$	*5.Days*1.Week*
2	3	$< \{(700,370),(720,380)\},[10:00,12:30] >$	*5.Days*1.Week*
3	1	$< \{(800,400),(820,420)\},[21:00,23:45] >$	*1.Day*4.Weeks*

20.1.3 Experimental Evaluation

In this section we present the experiments that we conducted to evaluate the proposed methodology for the preservation of historical $\mathcal{K}$–anonymity, when applied in a series of data. These experiments constitute one contribution of this work, since in [26] the effectiveness of the proposed strategy was not evaluated.

In our experiments we considered the existence of 10, 50, 100, 500 and 1000 users, who move randomly in a square area of 1000 square meters. Their speed of movement is randomly assigned and its value lies in $[0,10]m/sec$ (where '0' represents immobility). The direction of movement has four potential values: "north", "south", "west" and "east". Furthermore, we considered that the history of movement for all the users has recordings for a period of a month and that such recordings occur every 15 minutes. Among the users, we selected a single requester who sends 1000 requests during a month's period. The assumption of a single requester does not affect the results of the experiments since we considered that he/she initiated a large number of requests within a very short time interval. Table 20.1 contains the mobility patterns that we considered for the requester, where each POI is defined as a rectangle based on its lower left and upper right coordinates. Furthermore, for each mobility pattern of the requester we provide the periodicity in which it appears in the movement of the user. The selected notation (borrowed from [26]) defines the number of times that this pattern appears in the movement of the user (i.e. *1.Day*4.Weeks* simply states that the user is expected to follow this pattern of movement at least one day every four weeks).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 20.2: The experimental setup.

Parameter	**Values**
Size of Area	$1000m^2$
Number of users	$\{10, 50, 100, 500, 1000\}$
Speed of movement	$[0, 10] m/sec$
Number of requests	1000
Recordings of user movement for	1 month
Frequency of recordings	every 15 min
Level of anonymity (k)	$\{2, 3, 5, 10, 25, 50, 75, 100\}$
Values for temporal constraints	$\{15, 30, 120, 360\}$ min
Values for spatial constraints	$\{100, 200, 500\}$ meters

Table 20.3: Experimental results for 10, 50 and 100 users.

Users	**Temporal cons.**	**Spatial cons.**	$\mathcal{K}$	**Failures**
10	15 min	100 m	2, 3	27, 28
10	30 min	100 m	2, 3	20, 28
10	120 min	200 m	2, 3	18, 26
10	360 min	500 m	2, 3	14, 25
50	15 min	100 m	2, 3, 5	27, 28, 28
50	30 min	100 m	2, 3, 5	19, 27, 28
50	120 min	200 m	2, 3, 5	13, 26, 26
50	360 min	500 m	2, 3, 5	6, 24, 25
100	15 min	100 m	2, 3, 5, 10	22, 27, 28, 28
100	30 min	100 m	2, 3, 5, 10	16, 26, 27, 28
100	120 min	200 m	2, 3, 5, 10	10, 23, 25, 27
100	360 min	500 m	2, 3, 5, 10	6, 22, 23, 25

Regarding the desired level of anonymity, we considered the following values for $\mathcal{K}: \{2, 3, 4, 10, 25, 50, 75, 100\}$. Moreover, to formulate the spatiotemporal constraints which are essential for the proper execution of the algorithms, we regarded time intervals of $\{15, 30, 120, 360\}$ minutes and areas of size $\{100, 200, 500\}$ meters. All the essential experimental setup parameters are summarized in Table 20.2.

Tables 20.3 and 20.4 depict the values that were provided as input to the proposed algorithms, as well as the success ratio for each combination of these values (counted as

Table 20.4: Experimental results for 500 and 1000 users.

Users	**Temp. cons.**	**Spatial cons.**	$\mathcal{K}$	**Failures**
500	15 min	100 m	2, 3, 5, 10, 25, 50	12, 20, 26, 28, 28, 28
500	30 min	100 m	2, 3, 5, 10, 25, 50	6, 18, 26, 26, 28, 28
500	120 min	200 m	2, 3, 5, 10, 25, 50	6, 14, 24, 23, 27, 27
500	360 min	500 m	2, 3, 5, 10, 25, 50	5, 11, 18, 22, 26, 27
1000	15 min	100 m	2, 3, 5, 10, 25, 50, 75, 100	9, 14, 18, 24, 26, 28, 28, 28
1000	30 min	100 m	2, 3, 5, 10, 25, 50, 75, 100	4, 12, 17, 21, 24, 26, 28, 28
1000	120 min	200 m	2, 3, 5, 10, 25, 50, 75, 100	3, 9, 14, 17, 24, 25, 27, 28
1000	360 min	500 m	2, 3, 5, 10, 25, 50, 75, 100	2, 7, 10, 15, 20, 23, 26, 27

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

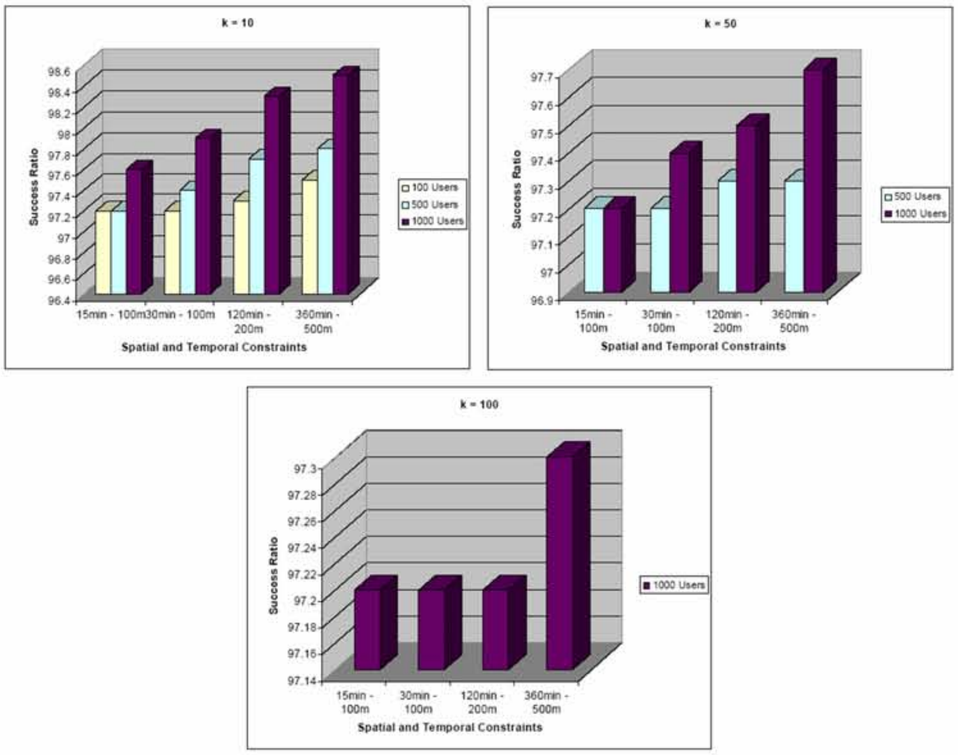


Figure 20.1: Success ratios for 10, 50 and 100–anonymity.

the number of failures in 1000 requests). As one can easily observe, in a total number of 1000 requests, the greatest number of failures was only 28, while the algorithm in its best performance experienced only 2 failures, a fact that indicates the high effectiveness of the proposed approach. Moreover, it is worth mentioning that the largest number of failures appeared when we considered the minimum number of users (i.e. 10 users) as well as the minimum values for the temporal and spatial constraints (i.e. 15 minutes and 100 meters, respectively). Figure 20.1 summarizes the attained experimental results in the form of graphs. For brevity, we present only the figures corresponding to $\mathcal{K} = \{10, 50, 100\}$. As one can easily observe, the largest success ratio appears when we consider the greatest values for the constraints (i.e. 360 minutes and 500 meters) and the lower success ratio appears when we consider the minimum number of users (i.e. 10 users).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

20.2 Network Aware Historical $\mathcal{K}$–Anonymity

In this section, we present the first network aware approach that was ever proposed for the offering of historical $\mathcal{K}$–anonymity in LBSs. The proposed approach is governed by the same underlying principles as the ones of [185], but it offers much more flexibility to the users and is also more secure to attacks. Specifically, the users of the system can explicitly define paths (instead of just sequences of POIs as in [185]) in the underlying network of user movement, where they consider that their privacy is under threat. Furthermore, the employed generalization algorithm is more effective than its counterpart in [185] (which formulated the ASR based on the Euclidian distance of the requester from his/her neighbors), since it is based on the *NN–Cloak* algorithm of [100] which is proved to be more secure as it avoids the placement of the requester near to the center of the computed ASR. The generated ASR is a rectangle (instead of a square) that is defined as the MBR enclosure of the users participating in the anonymity set of the requester. As an effect, the computed regions of anonymity are typically much smaller when compared to the ones that are produced by the generalization algorithm of [185]. If the generalization algorithm fails, an unlinking algorithm is enforced that uses *NN–Cloak*, as well as it effectively models the background knowledge that an attacker may have at his/her disposal with respect to the connection of different requests to the same individual, in order to better protect the privacy of the requester. Both the generalization and the unlinking algorithms are assumed to be executed in the environment of a trusted server, just before the request is forwarded to the appropriate service provider. In what follows, in Section 20.2.1 we present our solution methodology, while Section 20.2.2 contains a thorough set of experiments that we conducted for its evaluation.

20.2.1 Solution Methodology

Our strategy for the offering of historical $\mathcal{K}$–anonymity to the requesters of LBSs consists of two algorithms: a generalization algorithm and an unlinking algorithm. The generalization algorithm is employed when a user request for an LBS matches a mobility pattern

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

of the requester. In our privacy model we consider that the Region in each spatiotemporal element of a mobility pattern can represent either a regularly visited POI or a route in the network topology of user movement that the user frequently follows. As in [26, 185] the mobility patterns of each user are assumed to be known apriori by the trusted server.

The generalization algorithm operates exactly as Algorithm 20.1 with the only difference that the generalization of the ASR that contains the participants of the anonymity set is achieved by using *NN–Cloak*. On the other hand, the unlinking algorithm assumes the knowledge of a function $Link(\mathcal{R}_a, \mathcal{R}_b)$ that provides to the trusted server the probability of an adversary who has knowledge of the two requests $\mathcal{R}_a$ and $\mathcal{R}_b$ to link them to the same individual. Assuming that the trusted server has this kind of knowledge at his/her disposal, he/she can decide when to disclose a new request of a user to the service provider such that it is not linked to a previous request of the same individual. The unlinking algorithm operates as follows: (i) it constructs a dzone that contains the requester as well as $\mathcal{K}$–1 nearby users by using *NN–Cloak*, (ii) from this point on, the trusted server defers from servicing the requests of any of the users inside the dzone, as well as forwarding their location updates to the service providers, (iii) for each received location update coming from the requester (after the one that was transmitted as part of the user request) the trusted server computes the value of function *Link* as if the current location update was part of a new user request, (iv) when the value of the *Link* function drops below a privacy threshold θ the trusted server resumes the regular operation of the system inside the dzone; this means that the time that has passed in the meanwhile, as well as the current location of the requester, is such that his/her privacy is guaranteed.

20.2.2 Experimental Evaluation

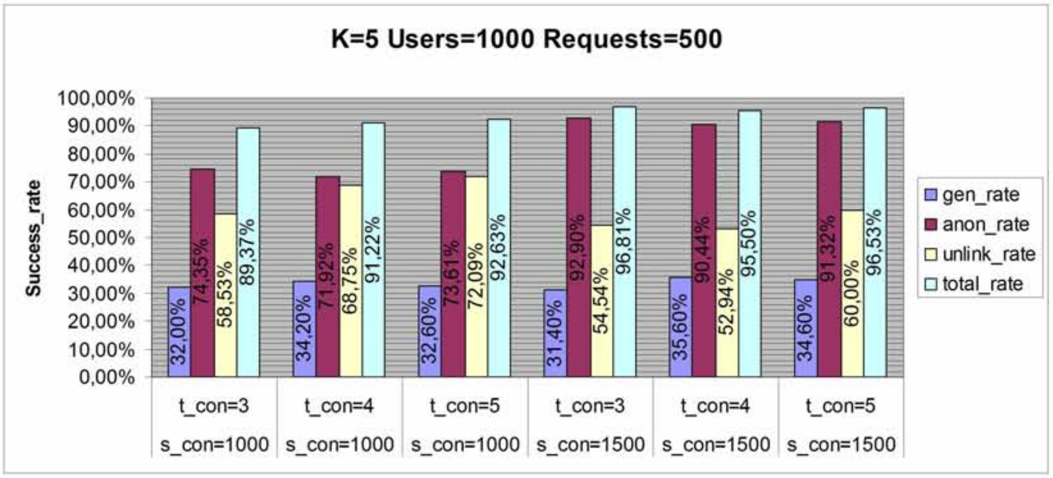
In this section, we present the experiments that we conducted to evaluate the quality of this methodology on a set of synthetic datasets that involved user movement in the regions of an underlying network topology. In all our experiments, we used datasets that were produced by Brinkhoff's network based generator of moving objects [30]. By using

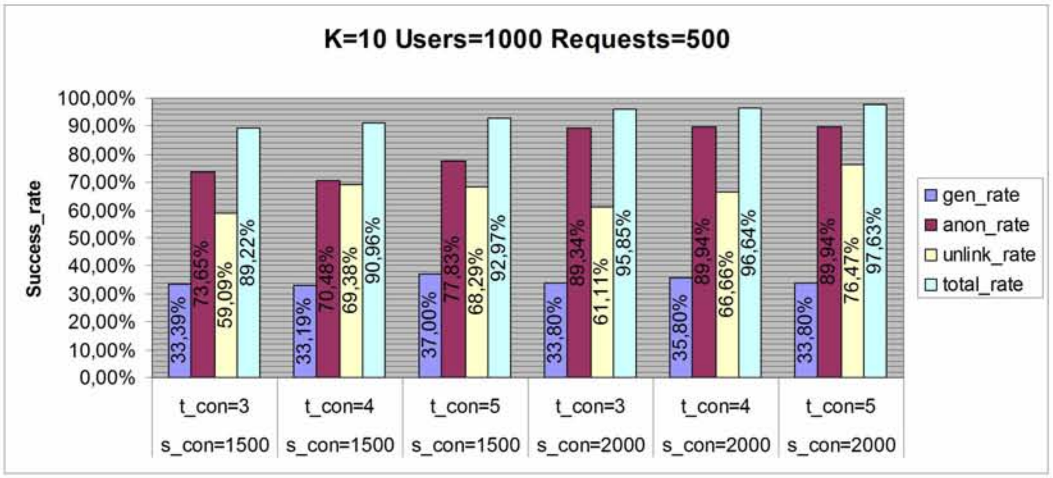
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Brinkhoff's generator we produced three datasets of trajectories based on the road network of the Oldenburg city. In our experiments, we consider the existence of 1000–10000 users, who generate 100–500 requests. The mobility patterns of the users were randomly selected as portions of their histories of movement. To formulate the tolerance constraints, we considered time intervals (t_con) of 1–5 time units and spatial intervals (s_con) of 200–2500 area units. Regarding the desired level of historical $\mathcal{K}$–anonymity, we considered values of $\mathcal{K}$ in the range of 5–200. Finally, to test the unlinking algorithm, we approximated the *Link* function by checking for the existence of $\mathcal{K}$ users in the computed ASR at every location update from the time of request until time t + t_con.

In Figures 20.2,20.3 and 20.4, we provide statistics regarding: (i) the generalization rate (gen_rate) as the percentage of requests that matched a mobility pattern of the requester, (ii) the anonymization rate (anon_rate) as the percentage of the matched requests that were successfully anonymized, (iii) the unlinking rate (unlink_rate) as the percentage of the non–anonymized requests that were successfully unlinked, and (iv) the total success rate (total_rate) as the percentage of requests that were either anonymized or unlinked. We should point out that in all tested cases the average time that was needed by the anonymizer to provide $\mathcal{K}$–anonymity or unlink the user requests was less than 8 seconds.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136





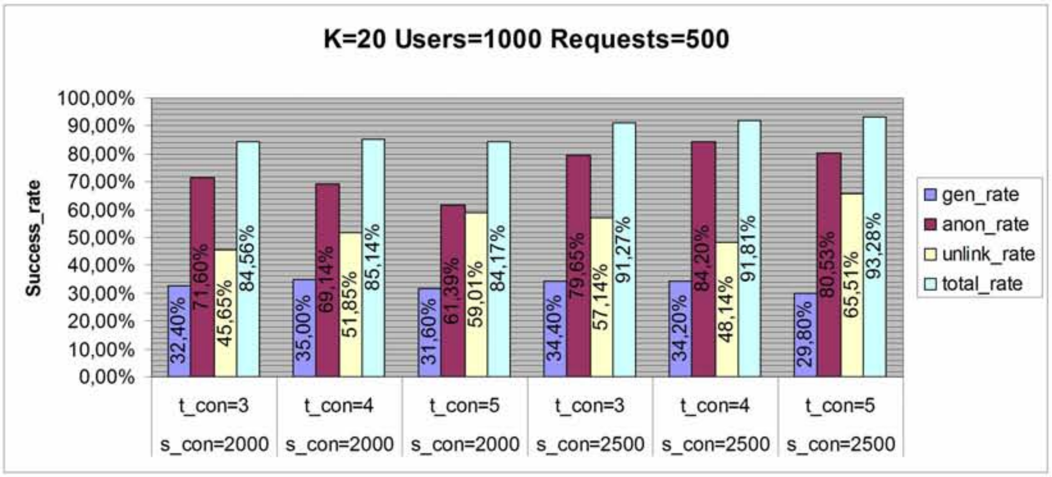
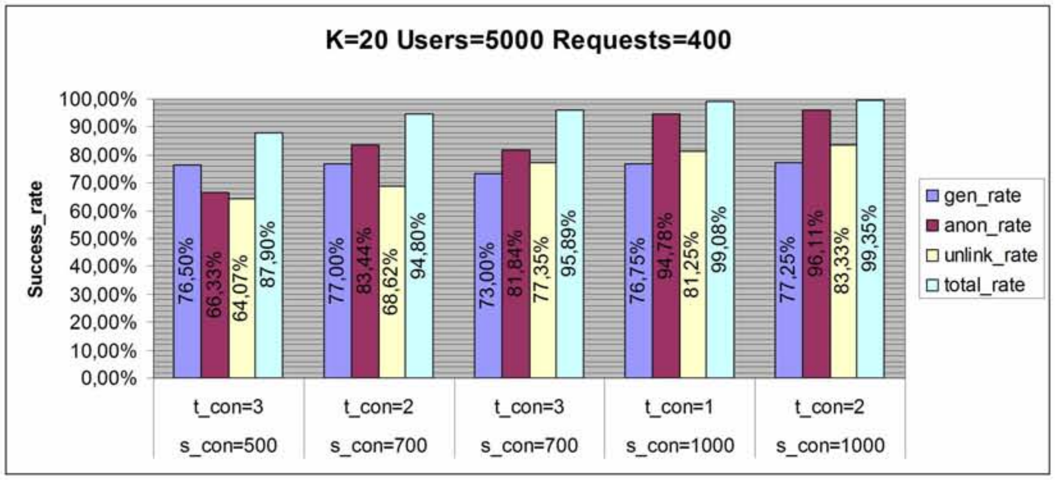
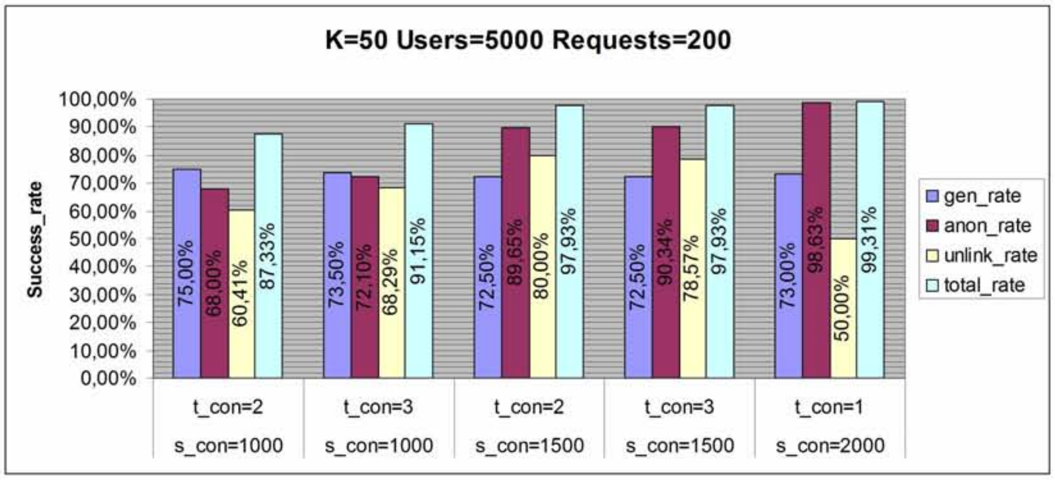


Figure 20.2: Experiments for 5, 10, and 20–anonymity in the first dataset.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136





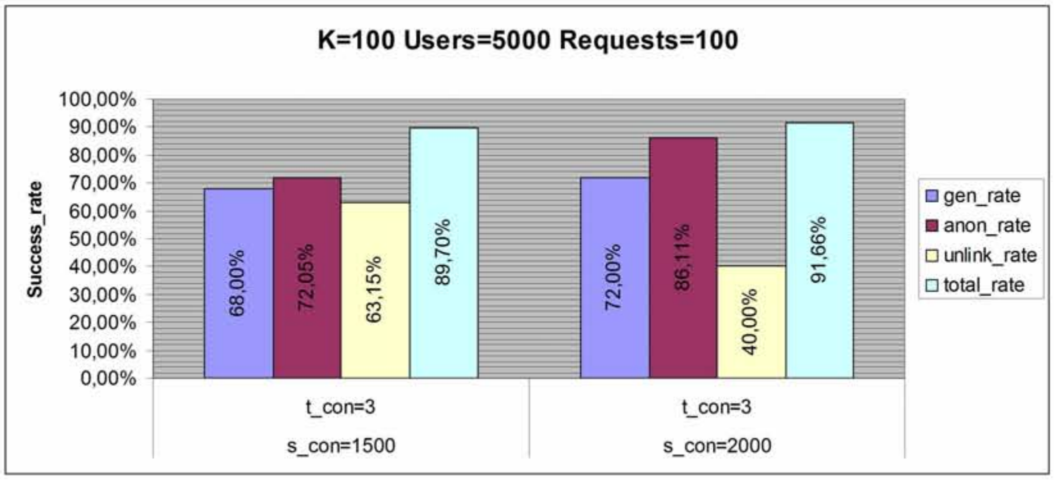
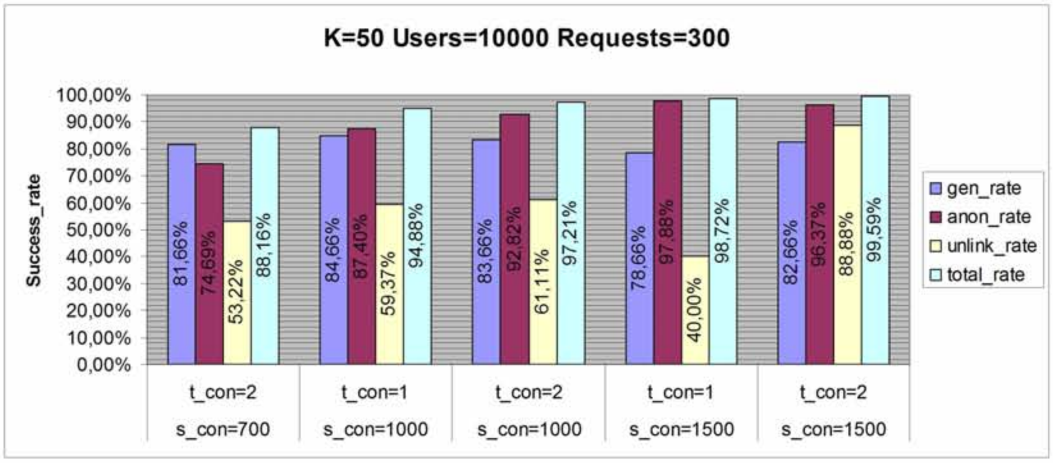
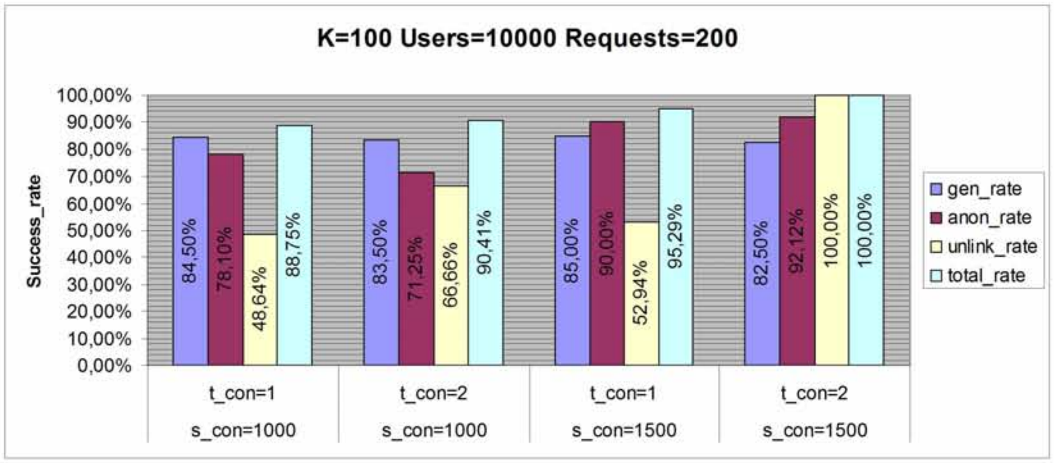


Figure 20.3: Experiments for 20, 50, and 100–anonymity in the second dataset.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136





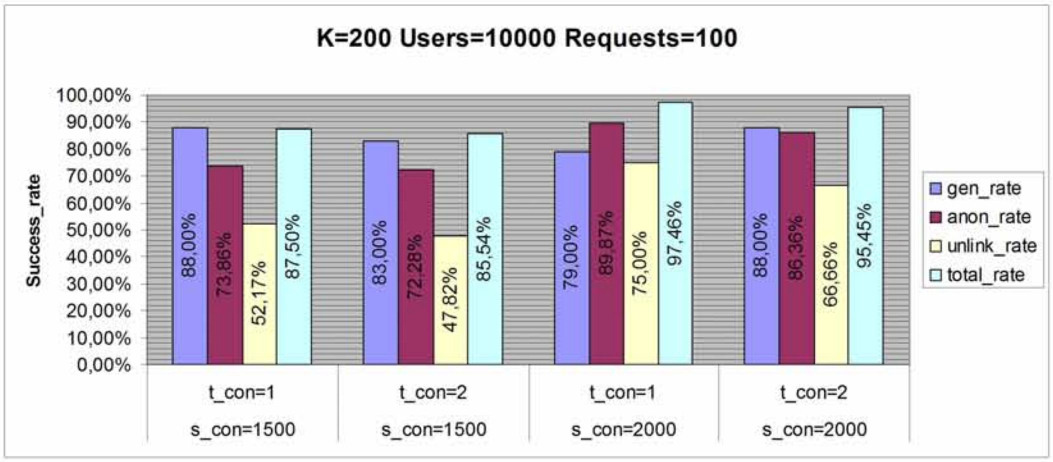


Figure 20.4: Experiments for 50, 100, and 200–anonymity in the third dataset.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 21

A Free Terrain Model for Trajectory $\mathcal{K}$–Anonymity

In this chapter, we introduce a privacy framework for LBSs that utilizes collected movement data to identify parts of user trajectories (i.e. routes), where user privacy is at an elevated risk. To protect the privacy of the user, the proposed methodology transforms the original requests into anonymous counterparts by offering trajectory (instead of historical) $\mathcal{K}$–anonymity to the requesters of LBSs. As a proof of concept, we have built a working prototype that implements our solution approach and is mainly used for experimentation and evaluation purposes. The proposed implementation relies on a spatial DBMS that carries out part of the necessary analysis. Finally, through a set of experiments, we demonstrate the effectiveness of the proposed approach to preserve the $\mathcal{K}$–anonymity of the users for as long as the requested services are in progress.

An important contribution of the proposed methodology is that the trusted server is capable of automatically identifying the mobility patterns that pose a threat to the privacy of each user in the system, instead of having them specified by the users. Once computed, these patterns are stored in the system and utilized by the trusted server to better protect the privacy of the requesters of LBSs.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

The rest of this chapter is organized as follows: In Section 21.1 we introduce and motivate the proposed methodology for trajectory $\mathcal{K}$–anonymity in LBSs. Section 21.2 presents the HESTIA framework [71] that relies on the trusted server paradigm, along with a scenario of system operation. Section 21.3 presents the algorithmic techniques that support the proposed privacy methodology, along with a set of issues that pertain to the implementation of the HESTIA framework. Finally, Section 21.4 experimentally evaluates our proposed methodology.

21.1 Introduction

As highlighted in Chapter 19, the observed regularities in the user trajectories can be used by untrusted entities to breach user privacy, even when no other user identification information is in place (e.g., social security number, family name, etc). To offer privacy to the users when requesting LBSs, in this chapter we adopt a privacy model that is suitable for any scenario where the attacker has knowledge regarding the usual habits of the subjects that he/she wishes to locate. The attacker tries to use this knowledge, along with the anonymized user requests that reach the service providers and (possibly) some special on–site cameras that are placed at locations where the user is expected (based on his/her habits) to pass, to track the users down. This model of attack is based on the reasonable assumption that any attacker is bound to search for the subject at locations that the subject regularly visits. Thus, the attacker tries to steal the identity of the user by observing user requests that originate from places that are frequently visited by this user and by using on–site cameras to locate the user at the area of request (capturing the user trajectory part starting from the point of the user request until the provision of the service). As an effect, in the considered privacy model, only the users who request LBSs from places that they frequently visit are in an immediate need for privacy preservation through trajectory $\mathcal{K}$–anonymity, whereas for the other requesters the removal of the obvious identifiers from the user request is enough to ensure that they are properly covered up, since the attackers do not expect to encounter them at certain locations at specific times (based on their history of movement).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

An immediate advantage of the proposed model is that the knowledge regarding the frequently visited places of a user allows the system to decide when it should provide $\mathcal{K}$–anonymity to the user and thus avoids to pay the computational cost of anonymity unless it is necessary. Our privacy model operates by using the collected location updates of the users to reconstruct their history of movement and to subsequently identify patterns in their movement history where their privacy is at elevated risk. These mobility patterns, referenced both in space and time, are called *unsafe routes* and capture regular movement of a user that happens to be irregular for a sufficiently large number of other users in the system.

We argue that the privacy of a user is at stake only when requesting LBSs from within any of his/her unsafe routes and for as long as the user moves inside his/her unsafe routes and the service withstands completion. To justify this claim, consider the situation where a user requests an LBS from a location and at a time that does not match any of his/her unsafe routes. Then, one of the following statements must hold: either (i) the user is located at a place that is regularly visited (at that time of day) by a very large portion of the population, thus the user is shielded by the crowd and his/her anonymity is protected, or (ii) the user is located at a place that he/she rarely visits (generally or specifically at this time of day), therefore an adversary who has access to the user movement data is unable to associate the location of the user to his/her identity, thus breach the privacy of the user.

The privacy model presented in this chapter has been realized as a system framework which comprises of four phases. The first phase uses the collected location updates at the trusted server to reconstruct the history of user movement $\hat{\mathcal{H}}(u)$ of each user u in the system. This phase involves the construction of a 3D polyline (we consider that the reconstruction takes place in a two–dimensional space and a single–dimensional time) that captures the time series of the user location updates. The benefit of constructing the 3D polyline from the collected location updates rests on the fact that it allows us to deal with situations where the mobile devices of the users have rather low transmission rates of location updates. This introduces important gaps when mapping (at a later point) the location updates to a 3D grid of cells in order to capture the unsafe routes of each user. The ap-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

plied reconstruction process allows us to compensate for these discrepancies and identify meaningful user mobility patterns. Following the reconstruction of the 3D polyline is its decomposition into user trajectories based on application–specific spatial and temporal constraints. The decomposition of the history of user movement into trajectories is necessary to allow for the derivation of the unsafe routes of the user. This is accomplished in the second phase. The unsafe routes capture the time series of locations, where the privacy of the user is at an elevated risk. For their identification, we consider an unconstrained movement scenario in a free terrain, where the user can move freely within a spatially bounded area (e.g., a park, a football ground or — generally — any open field). After having identified the unsafe routes for each user and during the real–time operation of the system, a search is performed for a potential match between the location and time of a user request and any point in the requester's unsafe routes. While a location match has the intuitive meaning of the two locations coinciding in space, a time match refers to the agreement in time in a periodic way, where the period depends on the application. In the third phase, we offer privacy to the users who have requested an LBS from within an unsafe trip through a trajectory $\mathcal{K}$–anonymity approach that is tailored to the handling of user movement data. Finally, the fourth phase of the proposed framework deals with the event of failure in the provision of $\mathcal{K}$–anonymity. In this case, the trusted server postpones the servicing of the user request for a small period of time and attempts at some later point in time to offer $\mathcal{K}$–anonymity to the requester. If the anonymization process fails again, then the privacy of the user is protected by denying the servicing of the request.

The contributions of the proposed methodology are as follows:

- We introduce a framework for the provision of location privacy through the use of trajectory $\mathcal{K}$–anonymity in LBSs, along with a working prototype, the HESTIA system, that is used as a proof of concept, as well as a testbed.
- Our approach offers real–time anonymization to the users by utilizing both their past (captured through the history of user movement) and their present (captured through

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

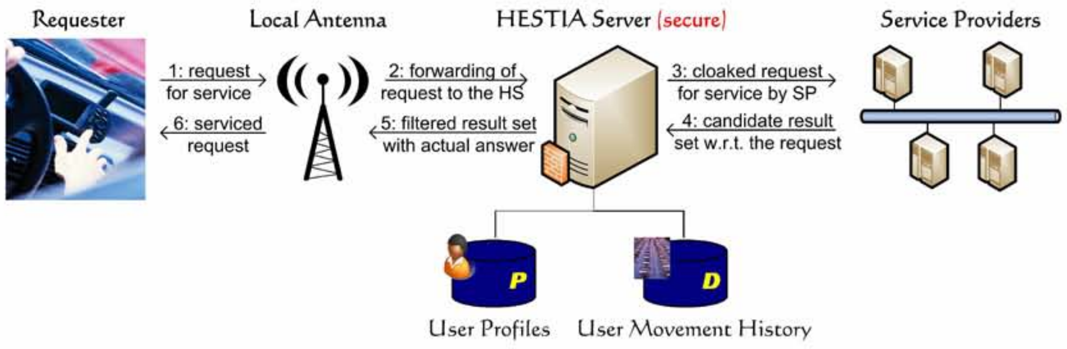

Requester
Local Antenna
HESTIA Server (secure)
Service Providers
1: request for service
6: serviced request
2: forwarding of request to the HS
5: filtered result set with actual answer
3: cloaked request for service by SP
4: candidate result set w.r.t. the request
P
D
User Profiles
User Movement History

Figure 21.1: The big picture of the ***HESTIA*** framework.

the requests that the user sends to receive LBSs). To accomplish that, we offline build a model of user movement by identifying unsafe routes where user privacy is at risk.

- Our approach utilizes a user movement database which keeps the current locations of the users along with their history of locations. By using the functionality of a spatial DBMS, we store user specified privacy parameters along with the users' unsafe routes in the DBMS, and we use this information to offer trajectory privacy to the users when requesting LBSs.

- Finally, we provide a thorough discussion regarding a set of implementation details that allow for a real–time operation of the proposed system as well as its tentative realization by different spatial DBMSs.

21.2 The HESTIA Framework

The HESTIA framework relies on the trusted server paradigm and extends state–of–the–art approaches for location privacy by providing methodologies for the offering of trajectory $\mathcal{K}$–anonymity. As shown in Figure 21.1, the proposed framework considers two types of entities, namely (i) a set of subscribers or users that are eligible to receive LBSs, and (ii) a set of service providers that offer the LBSs, and the following system components: (i)

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

a telecommunication infrastructure that carries out the necessary communication between the requesters and the service providers, and (ii) a secure server, called the HESTIA server, which is assigned the task of preserving the privacy of users from the inferences that can be drawn from users' trajectory data (collected from the users' movement history) when submitting requests to various LBSs. The algorithms that are implemented in the HESTIA server enable its operation in the unrestricted movement of users. Furthermore, the proposed algorithms utilize the spatial infrastructure of a DBMS to provide for the efficient manipulation of trajectory data. In what follows, we present in more detail the role of each component that is involved in the HESTIA framework and provide a scenario that demonstrates the different steps that are followed during regular system operation.

21.2.1 Component–Based Description of HESTIA

The primary component of the HESTIA framework is the HESTIA server, where the task of anonymization and privacy preservation takes place. The HESTIA server manages a user movement history database that keeps track of the location updates received by the users and accordingly updates their stored trajectories. The purpose of this database is two–fold. First, it maintains the history of movement for all users, which is used for the derivation of their unsafe routes. Second, it allows for the computation of the nearest neighbors of the requester at the time of request and at each subsequent location update, until the requested service is successfully provided.

To allow for the offering of privacy in a personalized manner, the HESTIA server stores a unique privacy profile for each subscriber in a profiles database. This profile is managed by the user through his/her mobile device and is subsequently uploaded and stored remotely in the HESTIA server. Furthermore, tied to each user is the set of unsafe routes, derived by the HESTIA server by examining the movement history of the user. The unsafe routes are updated on a regular basis to reflect new movement patterns that constitute a privacy hazard for the user. The offering of the various LBSs to the users of HESTIA is made possible through a set of service providers that compute the answer to the anonymized

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

request and communicate this information, via the HESTIA server, to the user. In this process, the HESTIA server is responsible (i) for the generation and the transmission of the anonymized request to the service providers in a format that is recognizable by the service providers, (ii) the subsequent filtering of the candidate answers, produced by the service providers (since the answer set returned by the service provider is typically a superset of that requested by the user), and (iii) the communication of the correct answer (based on the location of the user) to the requester.

21.2.2 A Scenario of the HESTIA System Operation

During typical system operation, the local communication infrastructure of the telecom operator receives periodic location updates from all registered positioning devices and forwards this information to the HESTIA server at a rate that depends on the capabilities of the client positioning device, the coverage of the network and the traffic congestion on the server–side. When the HESTIA server receives a new location update it stores this information in the user movement database and subsequently processes it to reconstruct the movement that the specific user has followed from the last transmitted location to the current one. Although the movement of a user in subsequent visits through the same route may produce a different sequence of location updates, the reconstruction process of the HESTIA server is able to provide a good approximation of the route that was actually followed by the users. The degree of this approximation is sufficient to allow for the identification of the unsafe routes of the users at a later point.

When collecting the individual location updates from the users, a trajectory identification algorithm is applied to identify discontinuities in the user movement behavior (such as a series of location updates originating from the same place) and break the history of movement into a set of trajectories. After being computed, the identified trajectories are stored back in the user movement database and are subsequently used by the HESTIA server to identify the unsafe routes of the user. Thus, given the trajectories of a user within a certain time period (e.g., a month), the HESTIA server identifies parts of these user trajec-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

tories that are frequently followed by this user and not by many other users in the system. These unsafe routes serve as a pseudo-identifier[1] and for this reason, the system protects the user whenever he/she is requesting services while he/she lies within any of his/her unsafe routes. To ensure adequate protection, a process in the HESTIA server is assumed to frequently search the trajectories of all the users in the system to capture and update the unsafe routes. Once identified, these routes are stored and are utilized by the system whenever the user requests a service.

The HESTIA server acts as an intermediate element that accepts the incoming user requests, removes all the non–spatial quasi–identifiers and proceeds to examine whether the requester is located in one of his/her unsafe routes. If this is the case, the user request has to become $\mathcal{K}$–anonymous in order for the user to be safe. To $\mathcal{K}$–anonymize the request, the trusted server generalizes both its spatial and temporal coordinates to cover a region which has been crossed by at least $\mathcal{K}$ subjects (including the requester) within a period of time close to the time the original request was submitted. Each of the $\mathcal{K}$ subjects can then be considered as a potential issuer of the request. On the other hand, if the user is in a safe route, the location and the time of the anonymized request will match the real information. In any case, the anonymized request will be forwarded to the appropriate service provider for servicing, and the results will be filtered by the HESTIA server to identify the correct answer and forward it back to the requester. This concludes the provision of the service in a privacy aware manner since only the HESTIA server, which is a trusted component of the system, is aware of the real identity of the requester and the service provider has no way of figuring out who the user is.

21.3 The Privacy Model

This section introduces the free terrain model for the offering of privacy in LBSs. The HESTIA framework, as it is applied to the free-terrain model, consists of four phases. Phase I

[1]The term “pseudo” in the identifier is used to emphasize the fact that the knowledge of an unsafe route does not in the general case breach the privacy of the user. However, under certain circumstances, a user may be uniquely identified by the knowledge of one of his/her unsafe routes.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

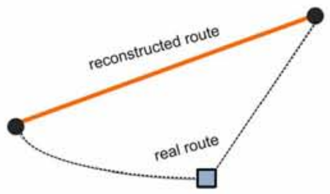


Figure 21.2: Reconstruction of user movement history in the free–terrain model.

is responsible for the reconstruction of the history of user movement and its subsequent decomposition into trajectories. Phase II detects the unsafe routes for each user in the system, while Phase III is responsible for the provision of trajectory $\mathcal{K}$–anonymity. Finally, Phase IV presents a solution that relies on service delay or denial to safeguard the privacy of the users when $\mathcal{K}$–anonymity fails. The four phases are described in Sections 21.3.1, 21.3.2, 21.3.3, and 21.3.4, respectively. Section 21.3.5 discusses the capability of the proposed approach to operate in a real–time environment, while Section 21.3.6 presents a set of issues that pertain to the implementation of the various algorithms in the HESTIA framework.

21.3.1 Phase I: Reconstruction of User Movement History

The first step towards the provision of trajectory $\mathcal{K}$–anonymity focuses on the utilization of the discrete points, collected through the transmitted location updates, to reconstruct the history of user movement. The purpose of reconstruction is to account for situations where the mobile devices of the users have a low transmission rate of location updates. As an effect, large spatiotemporal gaps can be introduced in the history of user movement, a fact that has an immediate consequence on the quality of the derived unsafe routes. The applied reconstruction process allows us to compensate for these discrepancies and identify meaningful user mobility patterns.

Under the free terrain model, users can move freely towards any direction at any time. The lack of constraints in the movement of the users encumbers the reconstruction process. Figure 21.2 demonstrates this claim. Suppose that due to the sampling rate of the mobile device, the system has knowledge of the two "circle" points but lacks knowledge of

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the intermediate “square” point. Thus, under the linear spaghetti model of movement [123], it assumes that the user moves as shown by the thick (red) line. The reconstructed user movement is determined by linear interpolation of the two consecutive points, as demonstrated by the thick line. As it can be noticed in Figure 21.2, if the system had collected the intermediate “square” point, the user movement would better approximate the reality. However, as we demonstrate in the following sections, a strategy that is based on a 3D grid allows us to compensate for certain discrepancies that are introduced as part of the reconstruction process. To perform the reconstruction, we iteratively construct the 3D polyline of the user movement history, by reconstructing the route followed by the user, segment by segment. More specifically, given two consecutive location updates, the proposed algorithm reconstructs the corresponding segment by joining the (x,y) points of the location updates.

After reconstructing the history of movement for each user from the collected spatiotemporal measurements, the next step is to appropriately decompose it into a series of trajectories. This process takes into account the movement behavior of each user, which allows for a different decomposition of his/her history of movement. The identification of the trajectories that exist within the history of movement of a user, is accomplished through the application of a trajectory identification algorithm, which operates by taking into account the spatial and/or temporal displacement between successive location updates in the history of user movement. There are several methodologies that can be applied for this purpose [63]. However, all methodologies are application–specific and thus knowledge is required regarding the parameters that signify the ending of the current user trajectory and the beginning of a new trajectory. Assuming that we have this kind of knowledge, the HESTIA server examines the history of movement for each user, extracts the corresponding trajectories, stores them in the user movement history database and links them to the appropriate user.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

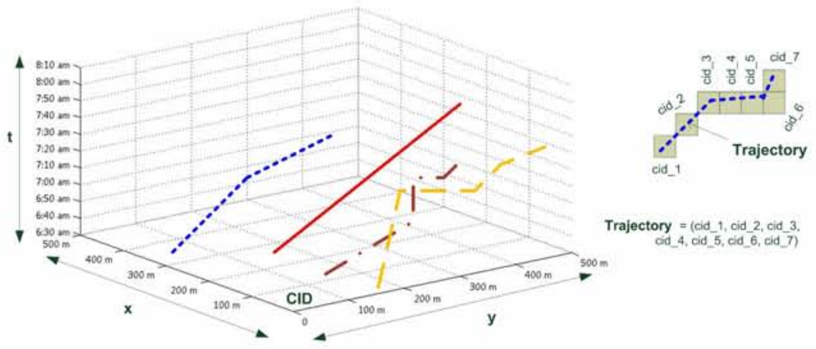


Figure 21.3: Using a three dimensional (3D) grid for the spatiotemporal generalization and the derivation of the unsafe routes.

21.3.2 Phase II: Derivation of the Unsafe Routes

The notion of a frequent route, as provided in Definition 11 (see Section 19.3), requires that the user follows the exact same route for a number of times. However, by taking into account the fact that in the free–terrain model the movement is unconstrained, one can realize that it is rather unrealistic for a person to exhibit this kind of behavior. For this reason, and in order to identify the unsafe routes of each user, in what follows we relax this assumption by using a 3D grid. Specifically, to derive the unsafe routes, we overlay a 3D grid over the total area that is covered by the trusted server and discretize the user movement into a set of spatiotemporal regions or *cells*. Then, instead of using the observed position of the user in this volume, we make use of the ID of the cell that contains the observed position. Thus, the observed positions are substituted by the corresponding cell identifiers.

Consider the illustration of Figure 21.3, where the temporal coordinate rests on the time of day and the trajectories depict movements of the users in the period of a day. The size of the cells in this grid determines the tolerance of the model to noisy data. Following the partitioning of the 3D space, each collected (x, y, t) measurement can be mapped to the appropriate cell through the use of a mapping function $\mathcal{M} : \langle x, y, t \rangle \rightarrow CID$, implemented as part of the spatial DBMS[2]. In the case of Figure 21.3, both x and y coordinates are

[2]This is a simple spatial function that tests the containment of one geometry (the point of request) to a set

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

abstracted to hundreds of meters, while time t is abstracted to tens of minutes. By using this strategy in the partitioned space, each trajectory $\hat{\mathcal{T}}(u)$ of a user u can be represented as a tuple $\hat{\mathcal{T}}(u) = \langle u, s \rangle$, where s is the sequence of cell identifiers (*CID*s) covering the spatiotemporal locations that the user traversed in the corresponding trajectory. This is demonstrated in Figure 21.3 (right), where a trajectory is decomposed into a set of *CID*s $\{cid_1, ..., cid_7\}$ based on the cells that the user traversed. Since each *CID* is referenced in time, without any loss of information, we can consider s to be an unordered *set* instead of a sequence of elements [63]. Thus, the user movement history database will collect the trajectories for all the users as a series of transactions $T = \langle u,\ tid,\ s \rangle$ (tid is the unique identifier of the trajectory). As one can observe, through this abstraction process, a set of spatiotemporal data (3D polylines) is transformed into market–basket data (cells referenced in both space and time), where each trajectory is mapped to a transaction and each point within the trajectory is mapped to an item of the transaction. In order to identify the frequent routes of a user, we apply frequent itemset mining [12, 14, 81, 87, 118, 158, 169, 187] to the transactions produced from all the user's trajectories in the system, as collected from the history of user movement. Since in frequent itemset mining duplicate items are neglected, we remove the *CID*s that appear in a transaction more than once. For the purposes of this process, the time recordings in the user trajectories are transformed into unanchored times in the 3D grid to depict time in a *generic* day (e.g., 3/17/2008, 10:30am $\rightarrow$ 10:30am).

Given the user movement history database $\mathcal{D}$, let $\mathcal{D}_u$ be the portion of $\mathcal{D}$ that collects the trajectories of user u, depicted as transactions in the form $T = \langle u,\ tid,\ s \rangle$. Accordingly, let $\mathcal{D}_{\bar{u}}$ be the view that contains the transactions T in $\mathcal{D} - \mathcal{D}_u$. Algorithm 21.1 can be applied as a patch to any frequent itemset mining approach to enable the discovery of the unsafe routes from the frequent routes of a user. The unsafe routes of each user are captured as the cells of the superimposed 3D grid that are frequently visited by this user and infrequently visited by the rest of the population. The algorithm takes charge at the point where the frequent itemset mining approach decides whether a candidate itemset is frequent or not.

of other geometries (the cells of the 3D grid).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 21.1 Derivation of unsafe routes in the free–terrain model

```
1: procedure UnsafePaths($\mathcal{D}_u, \mathcal{D}_{\bar{u}}, freq, MSO$)
2:     $\mathcal{D}_s \leftarrow \mathcal{D}_{\bar{u}}$ sorted based on the ID of each user $u$
3:     foreach itemset $I \in \mathcal{D}_u, frequency(I) \geq freq$ do
4:         if RSup($I, \mathcal{D}_s, freq$) $\geq MSO$ then
5:             treat $I$ as infrequent in $\mathcal{D}_u$                    ▷ the route is safe
6:         else
7:             proceed as usual                                    ▷ the route is unsafe

8: function RSup($I, \mathcal{D}_s, freq$)
9:     count ← 0
10:    foreach distinct user $u \in \mathcal{D}_s$ do
11:        $T = \{< u, tid, s >$ such that $I \in s\}$
12:        if $|T| \geq freq$ then
13:            count ← count + 1
14:    return count
```

Specifically, if an itemset is found to be frequent for a user u (thus, it appears a number of times that is at least equal to the frequency threshold *freq*), Algorithm 21.1 examines the frequency of this itemset for the rest of the users in the system. Then, if the number of the users in the system for whom this itemset is also frequent exceeds a minimum system–defined threshold *MSO*, this means that too many users support the corresponding route and thus it cannot be considered as unsafe for u. Otherwise, the frequent route is marked as unsafe for u. After identifying the unsafe routes, the HESTIA system stores them and links them to the appropriate user.

Algorithm 21.1 operates as follows. First, it sorts the trajectories (i.e. transactions) of each user in the system (other than u) in ascending order based on the user ID (line 2). Then, it uses the trajectories of user u (as collected in $\mathcal{D}_u$) to identify the frequent itemsets corresponding to the frequent routes of this user (line 3). The next step of the algorithm is to decide whether an identified frequent route of the user is safe or unsafe. This is accomplished in lines 8–14. In the case that a frequent route of user u is also frequent for many other users in the system, this route is safe for u and is therefore discarded from the list of unsafe routes for u (line 5). On the contrary, if this route is not adequately supported by the rest of the population in the system then it is marked as unsafe for user u and is included in the output of the algorithm (line 7). Overall, Algorithm 21.1 returns those itemsets of u that correspond to this user's unsafe routes.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 21.2 Match of user requests in the free terrain model.

```
1: function MatchReqtoUPath($\mathcal{C}, \mathcal{O}, \mathcal{R}_c, frun, \mathcal{R}_p$)
2:     $CID\ L_c \leftarrow \mathcal{M}(\mathcal{R}_c)$
3:     if $frun$ = true then
4:         foreach $l \in \mathcal{C}$ do
5:             if $l$.cell $= L_c$ then
6:                 return true
7:     else
8:         $CID\ L_p \leftarrow \mathcal{M}(\mathcal{R}_p)$
9:         if $\mathcal{O}[L_p][L_c] = 1$ then
10:            return true
11:    return false
```

21.3.3 Phase III: Trajectory $\mathcal{K}$–Anonymity

The derivation of the unsafe routes provides the means for the preservation of trajectory $\mathcal{K}$–anonymity, since these routes define the unsafe (spatiotemporal) regions for each user on the move. The offering of trajectory $\mathcal{K}$–anonymity involves two steps. The first step examines the potential matching of the location of the requester to one of his/her unsafe routes (Algorithm 21.2). Provided that such a matching exists, the second step, presented in Algorithm 21.3, offers privacy to the user by means of trajectory $\mathcal{K}$–anonymity.

Matching of a User Request to an Unsafe Route of this User

When a user requests an LBS from within the area of a cell, the trusted server receives the request and examines the unsafe routes of the requester to decide whether his/her privacy is at stake. Algorithm 21.2 presents an efficient methodology for the determination of (i) a matching of the initial user request $\mathcal{R}_c$ to one of the user's unsafe routes, and (ii) the subsequent matches until the completion of the service.

As a first step, the algorithm uses a function $\mathcal{M}$ (implemented on the spatial DBMS) that maps the user location to the appropriate *CID*. A boolean variable *frun* is provided as input to the algorithm to allow it distinguish between the search for an initial match and a subsequent match. In the case of an initial match, $\mathcal{R}_c$ corresponds to the current request. The algorithm scans the list of locations $\mathcal{C}$, derived as the union of all the *CID*s that appear in all the unsafe routes of the user, to identify if the current *CID* of the user is part of an unsafe route. By including all the *CID*s the algorithm takes into consideration that the user

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

may enter an unsafe route from any safe location. If the *CID* of the user matches an element of $\mathcal{C}$, then trajectory $\mathcal{K}$–anonymity should be offered to the user. Otherwise, the request is anonymized just by removing the obvious identifiers.

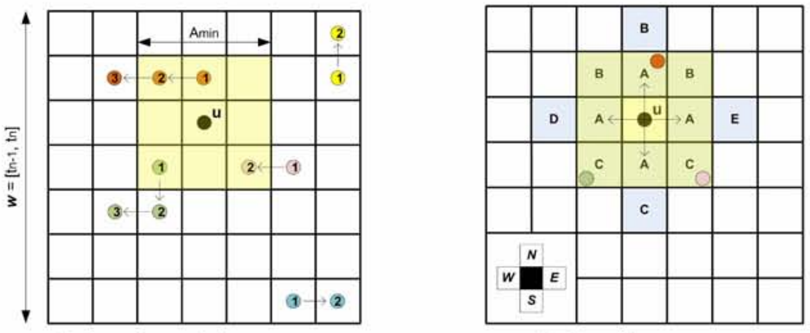
The case of a subsequent match occurs when the requested service needs subsequent location updates in order to be successfully completed. Then, variable $\mathcal{R}_p$ corresponds to the previous location update of the user, while $\mathcal{R}_c$ corresponds to the current location. Supposing that the user has moved to a new cell, a transition table O identifies if the user still moves within an unsafe route. The transition from one cell to the other can be easily identified through the GPS device in the following location update. The transition table O is derived as the upper triangular matrix of the *CID*s, where $O[i][j] = 1$ if the *CID*s i, j are consecutive elements in any of the user's unsafe routes. Otherwise, $O[i][j] = 0$. It should be noted here that since the transition tables of the users are expected to be sparse, the use of a memory–efficient, sparse matrix representation as in [148] can become handy. In the case of a successful match of the user's request to one of his/her unsafe routes, the privacy of the user is at stake and thus the user request has to become $\mathcal{K}$–anonymous.

Offering Trajectory $\mathcal{K}$–Anonymity

To provide $\mathcal{K}$–anonymity, we consider a time interval $\boldsymbol{w}$ that extends from the time of request t_n to some moments back t_{n-1} and identify the routes of all users in the system with respect to $\boldsymbol{w}$. Two parameters regulate the spatial extent of the generalization area. The first parameter, A_{max}, defines the maximum generalization that guarantees the reliable operation of the service. The second parameter, A_{min}, defines the minimum area where the $\mathcal{K}$–1 users of the anonymity set should be found so that the user is adequately protected.

Consider Figure 21.4(a) that shows the trajectories of five users (represented by dots that show the successive cells they moved into during their recent movement history) during a time interval $\boldsymbol{w}$ and the cell of request for a user u. Based on the location accuracy that is necessary for the requested service, the system decides on the appropriate size for $\boldsymbol{w}$ and creates the corresponding (x,y)–projection. The numbers in the colored dots indicate the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Locations of the users. (b) Searching strategy.

Figure 21.4: Offering $\mathcal{K}$–anonymity in the free terrain model.

sequence of cells that were visited by each other user in the system, where number 3 corresponds to the most recent cell. During a location update, the system gathers information regarding the location of all the users and accordingly updates their movement histories. Through the collected information the system is capable of computing the number of users that lie within each cell of the grid. Moreover, by utilizing this information the system can easily construct a view of the trajectories with respect to $\boldsymbol{w}$, similar to that of Figure 21.4(a). The computation of this view provides the 2D grid that constitutes the search space for the anonymity set. A mechanism is then required to conduct this search and identify the $\mathcal{K}$–1 neighbors of user u that are within an area that is at least of size A_{min}. After a successful run, the HESTIA server includes in the $\mathcal{K}$–anonymized version of the request the union of the explored cells, as the region within which the requester u is located[3]. Furthermore, in the anonymized request the actual time of request is replaced by the time interval $\boldsymbol{w}$.

Figure 21.4(b) demonstrates the applied search strategy for the provision of $\mathcal{K}$–anonymity. In the grid of Figure 21.4(b), each cell is considered to have four neighbors; one on its North, one on its South, one on its West, and one on its East. First, the cell where u is located is checked against the requirements of $\mathcal{K}$ and A_{min}. If these requirements are

[3]Another option would be for the HESTIA server to forward to the service providers the MBR enclosure of all the searched cells. This alternative is presented in Figure 21.4 and included as an option in Algorithm 21.3. However, since the diamond–shaped area fulfils the requirements of the anonymity set and is bound to be smaller compared to its MBR, we believe that there is no need to introduce any extra distortion that may unnecessarily reduce the accuracy of the offered LBSs.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 21.3 Offering $\mathcal{K}$–anonymity in the free terrain model.

```
 1: function GENERALIZEREQUEST($\mathcal{R}_c$, $\mathcal{K}$, $A_{min}$, $A_{max}$)
 2:     declare CID $L_c \leftarrow \mathcal{M}(\mathcal{R}_c)$
 3:     declare CID_list $N[4] \leftarrow \varnothing$
 4:     declare Hash $CS \leftarrow \varnothing$
 5:     declare Hash $Users \leftarrow \varnothing$
 6:     declare Queue $Q \leftarrow \varnothing$
 7:     declare Integer $Vcells \leftarrow 0$
 8:     ENQUEUE($Q$, $L_c$)
 9:     Users$\{u\} \leftarrow 1$
10:     while $Q \neq \varnothing \wedge (Vcells \times area\_of\_cell \leq A_{max})$ do
11:         CID $l \leftarrow$ DEQUEUE($Q$)
12:         $CS\{l\} \leftarrow 1$
13:         foreach user $\alpha \in$ cell $l$ do
14:             $Users\{\alpha\} \leftarrow 1$
15:         $Vcells \leftarrow Vcells + 1$
16:         if $Vcells \times area\_of\_cell \geq A_{min} \wedge \%keys(Users) \geq \mathcal{K}$ then
17:             return $CS$                                   ▷ alternatively, return MBR($CS$)
18:         CID_list $N \leftarrow$ GETNEIGHBORCELLS($l$)
19:         for int $i = 0; i < 4; i{+}{+}$ do
20:             if $N[i] \notin keys(CS)$ then
21:                 ENQUEUE($Q$, $N[i]$)
22:     return null
```

not met, the system identifies its neighboring cells (denoted as ‘A’ in Figure 21.4(b)) and checks for each of them if their region, combined to the already searched region, satisfies these requirements. This process iterates in a BFS manner, until either the requirements are met or the explored area exceeds A_{max} prior to the fulfilment of $\mathcal{K}$–anonymity.

Algorithm 21.3 has the details for the provision of $\mathcal{K}$–anonymity in the free terrain model. The proposed approach uses two hashes and one queue to efficiently handle the system resources. Specifically, hash *CS* (line 4) keeps track of the already visited cells so that they are not searched again (due to a potential overlap) as neighbors of other cells. On the other hand, hash *Users* (line 5) stores the distinct users that were found, while searching all the cells in *CS*. After an execution of the algorithm the cells that are stored in *CS* formulate the region where $\mathcal{K}$–anonymity was achieved and the elements of hash *Users* formulate the anonymity set for u (including the requester u). The queue Q (line 6) serves exactly as in a typical BFS implementation as it keeps track of the neighbors of the currently searched node so that they are visited in the next iteration. Finally, the integer variable *Vcells* (line 7) keeps track of the size of hash *CS*, i.e. the number of already visited cells, and is used to check for the satisfaction of the A_{min} requirement.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

At each step the algorithm retrieves a *CID l* from the queue Q (line 11), it inserts it in the hash of checked cells *CS* (line 12) and accordingly updates the hash *Users* to include all the users found in the new cell (lines 13–14). Due to the way that the update is performed, each user is identified only once no matter how many cells he/she appears in. Following that, the number of searched cells is updated to include the current cell (line 15) and a check is performed to identify if the requirements for the provision of $\mathcal{K}$–anonymity hold in the current setting (line 16). If this is the case, then the region where $\mathcal{K}$–anonymity was achieved is returned (line 17). Otherwise, the algorithm proceeds to identify the neighbors of the currently searched cell and include the ones that have not been searched yet, in Q (lines 18–21). The same process iterates until either the requirements of anonymity are met or the whole space is searched prior to the fulfilment of the requirements for $\mathcal{K}$–anonymity.

In a service that requires multiple location updates for its completion, the following strategy is applied. At each location update, the trusted server checks if the previously computed neighbors of the user have changed cell and for those who have, if their new cell lies within the previously computed ASR. If this holds, the algorithm adjusts the temporal information of the request and maintains the same area of anonymity. On the other hand, if some of the $\mathcal{K}$–1 subjects have left the region, the same number of subjects are sought among the other users in the system. If the requested number of missing subjects cannot be found in this region, Algorithm 21.3 is applied to compute a new $\mathcal{K}$–ASR.

21.3.4 Phase IV: Service Delay or Denial

There are situations when the proposed approach for the offering of trajectory $\mathcal{K}$–anonymity to the requesters of LBSs may fail. This situation may occur for several reasons, such as an exceptionally high value of $\mathcal{K}$ requested by the user or tight spatiotemporal generalization constraints (A_{max}, $\boldsymbol{w}$) required for the provision of the service. In all such cases, an approach has to be employed to effectively protect the identity of the requester.

Our proposed approach is influenced from the work of [54, 82] and introduces a small delay in the servicing of the request, in the hope that in the meanwhile (i) more

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

users will approach the location of the requester, thus become part of his/her anonymity set, and/or (ii) the requester will move to a more populated region. Specifically, we defer the servicing of the request for a pre–specified time period and then re–execute Algorithm 21.3 for the provision of trajectory $\mathcal{K}$–anonymity. If the algorithm fails again, then we protect the privacy of the user by denying the servicing of the request.

21.3.5 Achieving Real–Time Operation

Offline operations The most important offline task of the system is the derivation of the unsafe routes of each user. The computation of the unsafe routes is also the most time–consuming operation accomplished by the HESTIA server, with its time bounded by the frequent itemset mining algorithm that is employed. However, since the unsafe routes of a user do not change frequently, a counter may be assigned to each user to reflect the time period that elapsed from the last time that his/her unsafe routes were computed. When this counter expires, a dedicated server can be used to update the unsafe routes of the user and replace the previous ones. Furthermore, the necessary structures for the matching of the requests to the unsafe routes can also be updated as a part of this process and be stored in secondary memory. All the above–mentioned enhancements enable the system to operate in a timely fashion and in a real–world environment.

Online operations During online operation the purpose of the system is to receive user requests, anonymize them if necessary and then forward them to the service providers to be serviced. When a new request is received by the trusted server, the system examines the location of the requester with respect to his/her unsafe routes and proceeds to provide the necessary degree of $\mathcal{K}$–anonymity. The time required for this decision is low. Algorithm 21.2 requires in the worst–case scenario time that is linear to the number of cells that comprise the unsafe routes of the user. Moreover, in the best–case scenario the time that is required is constant. Thus, provided that the necessary structures for Algorithm 21.2 are kept and can be promptly read from disk, the decision for the necessity of $\mathcal{K}$–anonymity can be rapidly taken. Furthermore, subsequent runs of the algorithm for the same request

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

are served much faster since the regions of interest, where the search will be conducted, are further reduced and isolated from the set of all the unsafe routes of the user. Regarding the provision of trajectory anonymity, the generalization strategy depends on the values of $\mathcal{K}$ and A_{min} stored in the privacy profile of the user, as well as on the maximum generalization A_{max} that is allowed for the offering of the requested service. Due to the involvement of these parameters, the time of this step cannot be safely estimated. However, unlike other approaches, our proposed algorithm takes into account a time interval starting from the recent past rather than on the exact time of request, to identify the neighbors of the requester. This fact allows the process to be time-efficient since it requires less time to identify the neighbors, an observation that can be justified through Figure 21.4(a). Finally, the use of the functionality of a spatial DBMS to efficiently identify the neighbors of the requester further enhances the performance of Algorithm 21.3 for the offering of $\mathcal{K}$–anonymity.

21.3.6 System Implementation Details

In this section, we investigate some details that pertain to the implementation of the presented algorithms as part of the HESTIA server. The free terrain model does not necessarily depend on the spatial component of a DBMS in order to operate. However, the use of a spatial DBMS allows for an efficient implementation by undertaking an important portion of the necessary workings. The spatial database technology has matured and is currently supported by most of the major DBMS vendors, including Oracle and IBM. The core part of the HESTIA server is a spatial database engine that stores the user trajectories and provides the necessary spatial functions for the handling of the stored data. Spatial database systems currently provide all the necessary functionality to deal with space, while they do not support time equally well. Since the HESTIA server relies on both space and time, we have had to push the functionality missing so that the time is also appropriately treated.

In our system implementation we used the spatial features of Oracle [3] to store the free terrain as a rectangle area and to partition it to a collection of geometries[4], the

[4]A *geometry* is the inherent data type that is used by DBMSs to store spatial data in a database. In that sense, a geometry can be any spatial object that the DBMS supports, such as a point, a line, or a polygon.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

cells. All the derived cells are kept in a spatial table that allows for an efficient identification of their neighbors. Specifically, given a cell, we use the spatial functionality of Oracle to identify other cells in the table, which share an edge with it. This is accomplished through Oracle's function SDO_RELATE, which carries out the functionality provided by the GETNEIGHBORCELLS function of Algorithm 21.3 through the use of a spatial index (an R*–tree [23, 85, 108]) in the table of the cells. Furthermore, the matching of the user requests to unsafe routes can be accomplished by determining the cell geometry that contains the current location of the user and checking this cell against the stored unsafe routes (cells) of the user. Finally, the MBR enclosure of the cells that formulate the $\mathcal{K}$–ASR, can be efficiently computed by using Oracle's SDO_MBR function on the collection of the cell geometries returned by Algorithm 21.3. Apart from providing an efficient solution to $\mathcal{K}$–anonymity, the spatial DBMS is also used to store the history of movement of each user, the trajectories, and the unsafe routes as sets of geometries in spatial tables.

The derivation of the unsafe routes of the users is the only part of the HESTIA system that does not benefit from the use of a spatial DBMS. Instead, an interface is needed for the execution of the frequent itemset mining algorithm and the communication of the results back to the calling program. This was accomplished through the system functions of the Java programming language. After the execution of Algorithm 21.1 and the reading of the unsafe routes, we used the Oracle Java API [4] to conduct the necessary communication of the results and the update of the corresponding tables in the Oracle database.

21.4 Experimental Evaluation

The proposed algorithms were implemented in Java and Oracle 10g Rel.2 on Windows XP on a 3.2 Ghz Intel Pentium D processor equipped with 4GB of main memory. The implementation made use of the Oracle Java API [4] to allow for the necessary communication with Oracle. Parts of the needed functionality were implemented using Oracle PL/SQL. The methodology was tested on the publicly available INFATI dataset [98]. This dataset contains a set of GPS measurements derived from an intelligent speed adaptation project

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

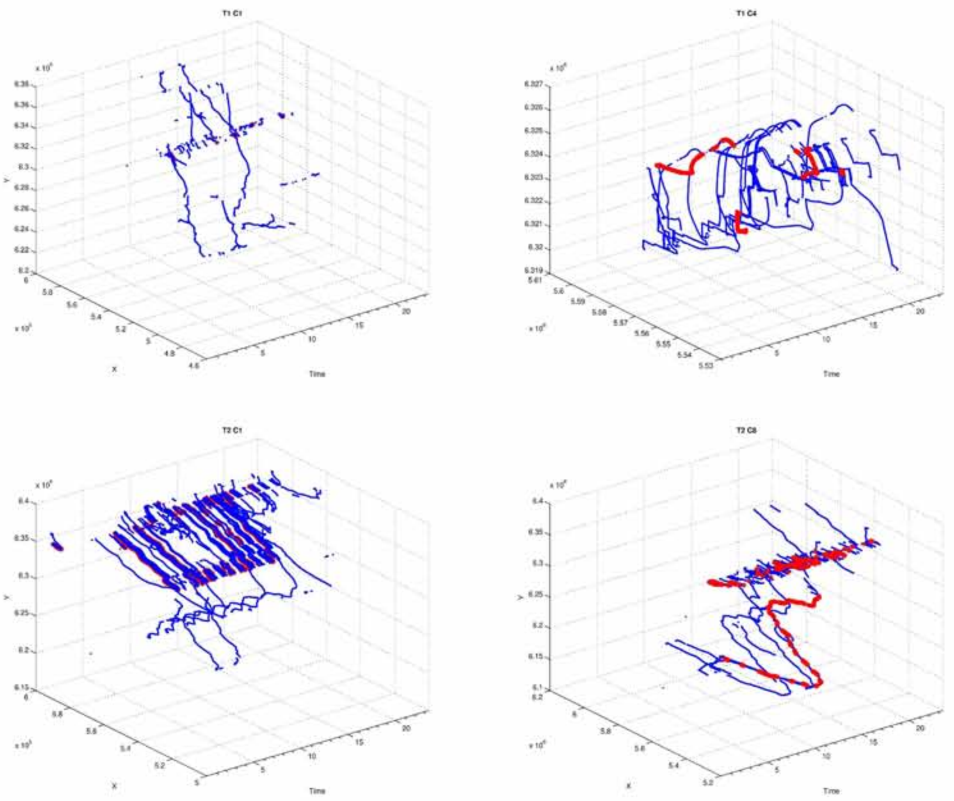


Figure 21.5: The unsafe routes (red) for the trajectories (blue) of the INFATI dataset, using *freq* = 2 and $MSO = 5$.

that was carried out in the Aalborg University in Denmark. The data was collected between December 2000 and March 2001 from a set of 20 cars that were driving in the municipality of Aalborg. Each of these cars was equipped with a GPS receiver and its movement was recorded at a sampling rate of one second, for a period of more than a month. The cars that participated in the experiment were divided into two teams, the first consisting of 11 cars and the second consisting of 9 cars. As a post–processing step, the collected data was examined to remove log entries with GPS positions that were close to the residence of each driver. More information about the dataset can be found in [98].

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

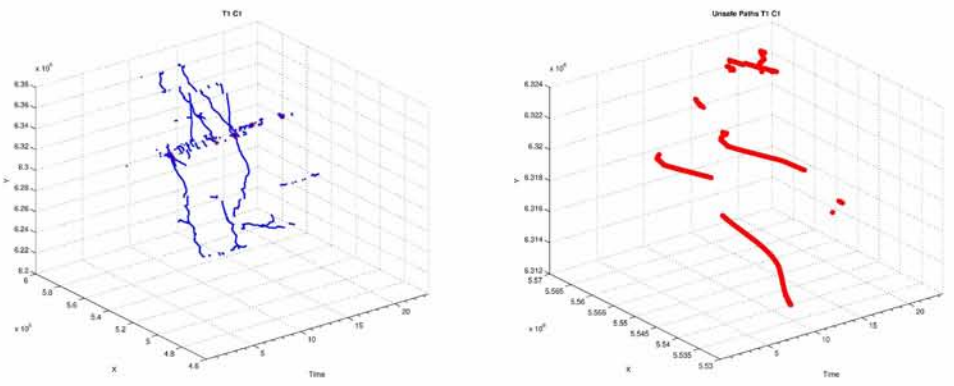


Figure 21.6: The unsafe routes (right) for the driver of car 1 of team 1.

To decompose the history of users' movement into trajectories, we considered a trajectory to be a set of GPS measurements such that no two consecutive readings have a time difference of more than five minutes. To capture the unsafe routes, similarly to [63], we considered 100meters$\times$100meters$\times$5minutes cells in a grid where the actual GPS readings were replaced by the ID of the cell they fell into. Each user trajectory was mapped to a transaction in the universe of the cell identifiers and Algorithm 21.1 was applied on Apriori [14] to derive the unsafe routes for each driver. For the purposes of our experimentation, we used a minimum frequency threshold *freq* of 2, 3, 4 and 5 and an *MSO* of 3, 4, and 5. In Figure 21.5 we present only the unsafe routes in the case where *freq* = 2 and *MSO* = 5 and only for four cars, two from each team. The unsafe routes for the rest of the cars of the INFATI dataset can be found in [71]. The graphs in these figures are titled as T_iC_j to denote the team i to which car j belongs. Furthermore, to enhance visibility, the time in each of these graphs is in military form and the most distant outliers have been removed. Figure 21.6 presents in more detail the identified unsafe routes for car 1 of team 1. By storing these routes in the privacy profile of user 1, the system can effectively protect the user when requesting LBSs from unsafe locations.

After the identification of the unsafe routes, we experimented with the capabilities of the proposed methodology regarding the offering of $\mathcal{K}$–anonymity. To create the free

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

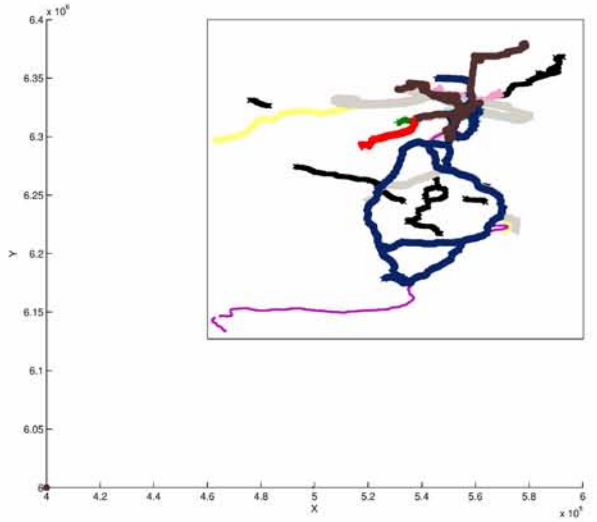


Figure 21.7: The area of the free terrain for team 1 (each color corresponds to a car).

terrain that is covered by the anonymizer, we considered the (x,y)-projection of all the trajectories for all the cars of team 1 excluding any distant outliers. The rectangle region of Figure 21.7 presents the limits of the area covered by the free terrain model. This area was split into a grid of 100meters$\times$100meters $\times$5minutes cells. Following the searching strategy of Figure 21.4 and based on the area of the terrain (see Figure 21.7), this means that the $\mathcal{K}$–anonymity algorithm was applied to a rectangle area of $1,400 \times 2,750$ cells. In our experiments we considered A_{min} to equal the area of a cell and a maximum generalization threshold such that all the $\mathcal{K}$–1 subjects of the anonymity set are to be found at a distance of at most 1 Km far from the requester. Since the drivers in the INFATI data are few for the purposes of our experiments, we randomly assign a set of drivers U in each car, with each driver being located at a different position in the movement history of the car. In what follows, we are only interested in time periods (rather than actual times) in the users' movement history. Thus, we assume that all the users move synchronously based on the (x,y) locations in the in the location updates of their assigned trajectories and disregard the time t of their location updates. Although the actual time t is omitted, the time difference between

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

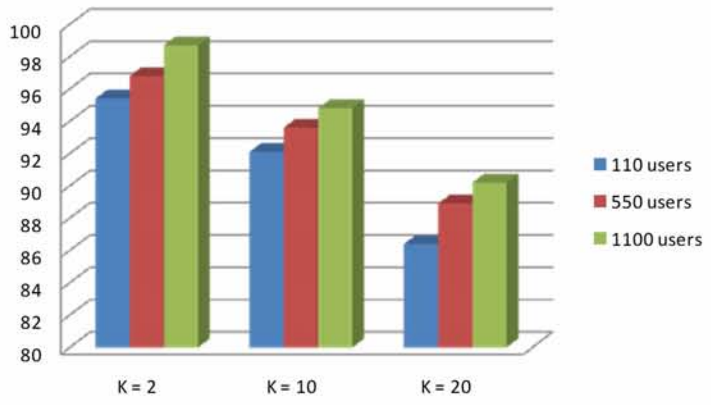
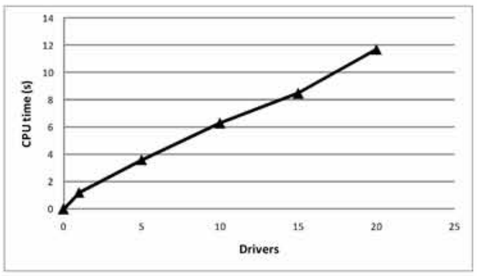


Figure 21.8: Success ratios for 2, 10, 20-anonymity and 110, 550, 1100 users.

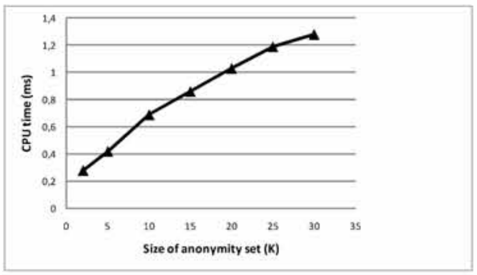
the consecutive location updates in a trajectory is maintained. We use this information to compute the locations of the corresponding user within the last 10 minutes. In the case of the team 1, this process allows us to compute the location of all $11 \cdot U$ drivers. To create a user request for an LBS, we randomly choose U location updates (to position the corresponding mimicked users), each from one of the 10 trajectories and $U{-}1$ location updates from the last trajectory, and assign each mimicked user to one of these points. The last user is positioned in the final trajectory such that he/she is located within one of his/her unsafe routes. This location is also the point of request. The assignment of the users to the cars is performed in a way that users assigned to the same car have no overlapping trajectories for the 10 minute period that is examined.

The trajectory $\mathcal{K}$–anonymity process uses the user movement history to compute the cells where the user was located within the last 10 minutes. This process leads to the construction of a view in the system that is similar to that of Figure 21.4(a), in which $\mathcal{K}$-anonymization takes place. Figure 21.8 presents the success ratio for $\mathcal{K} = \{2, 10, 20\}$ and for a number of users $U = \{10, 50, 100\}$ assigned to each trajectory. The reported ratio is an average over 100 runs coming from different users and unsafe routes and reflects the percentage of the instances in which $\mathcal{K}$–anonymity was successfully provided to the

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Identification of frequent routes.



(b) Provision of $\mathcal{K}$–anonymity.

Figure 21.9: The scalability of the free terrain privacy approach.

requester. As one can observe, the success ratio of the $\mathcal{K}$–anonymity algorithm is higher when the users of the system increase and the value of $\mathcal{K}$ decreases. Moreover, the $\mathcal{K}$–anonymity algorithm consistently achieves a success ratio of over 86%. Figure 21.9 presents the time that was needed by the HESTIA server to identify the frequent routes for the 20 cars of the INFATI dataset and to provide $\mathcal{K}$–anonymity. For the provision of $\mathcal{K}$–anonymity, the reported time is an average over 100 runs.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 22

A Network Aware Model for Trajectory $\mathcal{K}$–Anonymity

Existing work on privacy in LBSs fails to address one or more real–world challenges as (i) it does not consider that users typically move in a network–constrained environment (e.g., a road network), which allows for certain decisions to be made with respect to the offering of privacy, (ii) all user requests are anonymized in the same manner no matter what the location of the user is (with respect to his/her mobility patterns) when requesting the service and his/her future locations for as long as the service withstands completion, (iii) it does not build on existing DBMSs that are proved to provide numerous advantages over ad–hoc/"from scratch" implementations. For example, all the related work (Chapter 17) with the exception of the network aware historical approach to $\mathcal{K}$–anonymity (introduced in Chapter 20), fails to address the first requirement as it considers unconstrained user movement. Furthermore, most of the existing approaches (except from the methodologies of Chapters 20 and 21, as well as few others) fail to address the second requirement, since they anonymize all the incoming user requests in exactly the same manner. Finally, only the HESTIA privacy model (Chapter 21) builds on top of an existing spatial DBMS. However, it suffers from the following shortcomings: (i) it assumes that users are in need of protection only when requesting LBSs from unsafe routes; however, the privacy of the users may also

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

be jeopardized when they request LBSs from “sensitive” locations (e.g., hospitals, red light districts, etc), irrespective of whether these locations belong to their mobility patterns, and (ii) the quality of the extracted user mobility patterns highly depends on the partitioning of the total area that is covered by the anonymizer into 3D cells; the coarser the granularity of the partitioning, the lower the quality of the identified routes. In this chapter, we present a novel approach to trajectory $\mathcal{K}$–anonymity, which improves the current state–of–the–art algorithms for the offering of privacy in LBSs, by providing a solution that addresses all the above mentioned challenges.

The rest of this chapter is organized as follows: Section 22.1 introduces and motivates the proposed methodology, while Section 22.2 presents the algorithms that support it. In Section 22.3, we present a set of issues that pertain to the implementation of the system framework, while Section 22.4 concludes this chapter by providing the experimental evaluation of the proposed methodology.

22.1 Introduction

In this chapter, we present a privacy model that is suitable for the offering of trajectory $\mathcal{K}$–anonymity in LBSs. Our model is based on the same underlying principles as [69, 71], presented in detail in Chapter 21. Specifically, it assumes an attacker who has knowledge of the following: (i) the frequent movement behavior of all the users in the system, computed by the trusted server as part of its functionality, (ii) the location updates of the users, (iii) the anonymized versions of all the requests transmitted by the trusted server to the service providers, and (iv) the algorithms used by the trusted server to support user privacy.

Similarly to [71], the proposed privacy model is implemented as a framework that comprises of four phases. In the first phase, the trusted server uses the collected location updates, along with the underlying network of user movement, to reconstruct the (recent) history of movement for each user in the system. In the second phase, the reconstructed user movement is used to identify the frequent movement behavior of each user. Frequent movement behavior is captured as either safe or unsafe with respect to user privacy, based

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

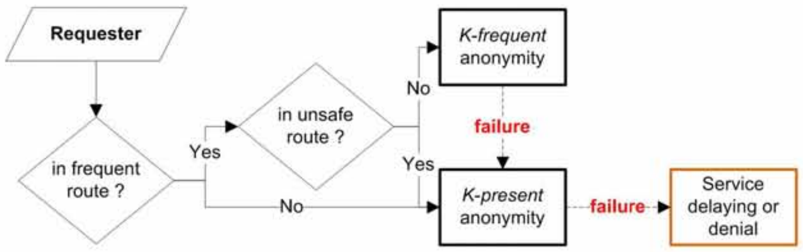


Figure 22.1: The anonymity strategies in the network aware model.

on whether it is also ordinary or atypical for a significantly large number of nearby users. In the third phase, we provide the user with trajectory $\mathcal{K}$–anonymity for handling online user requests on trajectory data. $\mathcal{K}$–anonymity is essential to protect the privacy of the users, starting from the point of request for an LBS and continuing for as long as the requested service withstands completion. As part of our framework, we deliver two types of trajectory $\mathcal{K}$–anonymity. $\mathcal{K}$*–present* (weak) trajectory anonymity identifies $\mathcal{K}$–1 subjects that are close to the requester at the time of request and thus could have issued the request (from the viewpoint of the service provider). On the other hand, $\mathcal{K}$*–frequent* (strong) trajectory anonymity collects the subjects that are near the requester at the time of request and for whom the current route of the requester is also frequent. As shown in Figure 22.1, the choice of the anonymity strategy depends on the location of the requester at the time of request and his/her subsequent locations until the completion of the service provision. In the offering of trajectory $\mathcal{K}$–anonymity, our model takes special care to protect the requester from sequential query tracking (see Section 17.3.3). The final phase of the proposed framework deals with the event of failure in the provision of trajectory $\mathcal{K}$–anonymity. In this case, the trusted server postpones the servicing of the user request for a small period of time. After that, if the anonymization process fails again, the requester is protected simply by blocking the servicing of his/her request.

The proposed privacy framework relies on a user privacy profile that stores the necessary information related to the privacy requirements of the user. This includes (i) the preferred value of $\mathcal{K}$ (in $\mathcal{K}$–anonymity) for each requested LBS, (ii) (a pointer to) the frequent

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

routes of the user, and (iii) the minimum spatial area A_{min}, around the requester, where the participants of the anonymity set should be searched for, so that the user is adequately covered up. Apart from A_{min} the system has knowledge of the coarsest spatial resolution A_{max} required for the provision of each LBS. This is a system–defined parameter that provides the maximum area of the anonymity set (with respect to the requested service) to ensure that the service can be delivered. The contributions of this work are summarized as follows:

- We propose a privacy model that utilizes an underlying network topology that confines user movement to deliver, in an interactive way, different levels of user privacy to online user requests in trajectory data.
- By using the movement history of the users in the system, we automatically extract patterns depicting the frequent movement behavior of each user. By contrasting these patterns with those of the rest of the population in the system, we classify each of them as either safe or unsafe with respect to the privacy of the user.
- We propose a privacy model that utilizes the unsafe patterns of the requester to deliver the most appropriate between two alternatives for trajectory $\mathcal{K}$–anonymity, while protecting the user from sequential query tracking.

22.2 The Privacy Model

This section introduces the network aware privacy model for the offering of trajectory $\mathcal{K}$–anonymity to the requesters of LBSs. The privacy model has been implemented as a framework that consists of four phases. Phase I is reconstructs the history of user movement and decomposes it into trajectories. Phase II detects the frequent routes of each user and separates them into either safe or unsafe with respect to his/her privacy requirements. Phase III is responsible for the provision of trajectory $\mathcal{K}$–anonymity. Finally, Phase IV presents a service delaying or denying mechanism that is used to safeguard the privacy of the users when $\mathcal{K}$–anonymity fails. The four phases are described in Sections 22.2.1, 22.2.2, 22.2.3,

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

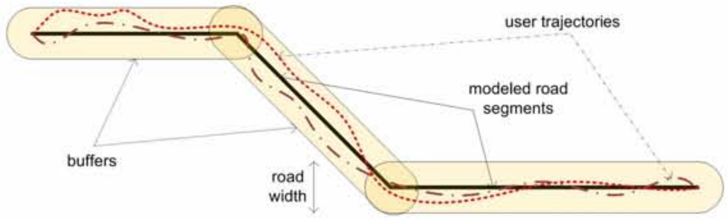


Figure 22.2: Possible motion curves and the role of buffers.

and 22.2.4, respectively. Section 22.2.5 discusses the capability of the proposed methodology to operate in a real–time environment. In what follows, we consider the existence of a spatial network (graph) $\mathcal{N}(V, E)$ that models the real network topology of user movement.

22.2.1 Phase I: Reconstruction of User Movement History

Under the network aware model, movement is restricted to a series of routes that are consistent to the underlying network. The existence of inaccuracies in the location updates that are collected by the trusted server, as well as the existence of irregularities in the movement behavior of the users, make the process of movement history reconstruction complicated. Specifically, given any type of physical network (such as a road network), an object is rather rare to transmit the exact same sequence of location updates when following the same route at two different occasions (e.g., at two distinct days of the month). An example of this situation is highlighted in Figure 22.2, where the user is shown to have crossed the same road links but has sent different location updates in each of the two trajectories (see the red polylines). Thus, a methodology is needed to deal with the uncertainty regarding the recording of an exact location of an object at a certain time.

Research works, such as [120, 123] (among others), study the reconstruction of movement in a network–confined environment. Our approach is a simplified version of [120] that offers good computational speed and requires less data housekeeping. It is shown to perform well, by providing accurate reconstruction at a degree that is sufficient for the workings of the proposed anonymization technique. As part of our model, the links of the underlying network (e.g., corresponding to a road or a road part) are considered to be cylin-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

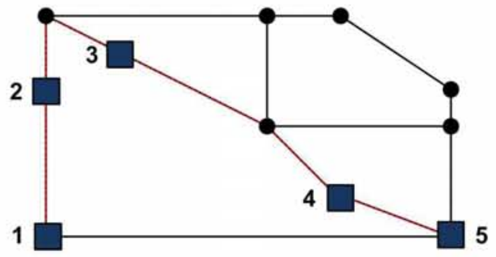


Figure 22.3: Reconstruction of user movement in a network–confined environment.

drical areas in the 3D space, as shown in Figure 22.2. This process is called *buffering* [24] and has been first used in [166]. In our model, the radius of the cylinder defines the spatial extent of the link and depends on its topology in the underlying network. The advantage of this representation is that it abstracts away possible recording and measurement errors and identifies when two trajectories are the same, even if they do not coincide in space and time. Buffers allow us to map the collected points to the appropriate links of the network.

To better understand the distinct nature of network based modeling, consider the movement of a user in the road network of Figure 22.3. The "square" points indicate the captured location updates transmitted by the mobile device of the user. Notice that when reconstructing the movement of the user, points 2 and 3 cannot be joined together by a straight line, since this action would lead to a segment that is inconsistent with respect to the underlying network topology. Furthermore, let us assume that point 4 was unknown to the trusted server. Then, to reconstruct the user movement, the system would have to identify the traveled route from point 3 to point 5. However, since there are several alternatives, the system has to identify the most probable route that the moving object has followed in the network based on the collected location updates. For this reason, it is essential to consider the existence of some network constraints that allow us to choose the most probable among the possibly many alternative routes. Network constraints can be classified as either spatial (e.g., a one–way road that cannot be traveled in the wrong direction or a closed road due to construction works) or temporal (e.g., the existence of a large volume of traffic on a road that makes impossible its traversal based on the low time difference of the collected

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 22.1 Reconstruction of user movement in the network aware model.

Input: Two consecutive location updates A, B.
Output: Reconstructed movement from location A to location B.

```
 1: function RECONSTRUCT($A(x_A, y_A, t_A)$, $B(x_B, y_B, t_B)$)
 2:     INSERTTONET($A$); INSERTTONET($B$)                    ▷ insert $A, B$ to the network
 3:     Cand_routes $P \leftarrow$ GET_$m$_SHORTEST_ROUTES($A$, $B$, $m$)
 4:     foreach $P_i \in P$ do
 5:         $t_{AB,i} \leftarrow$ COMPUTE_TRAVEL_TIME($P_i$)
 6:     REMOVEFROMNET($A$); REMOVEFROMNET($B$)                ▷ remove $A, B$
 7:     return route having $argmin_i(|t_{AB,i} + t_A - t_B|)$

 8: procedure INSERTTONET($p(x_p, y_p, t_p)$)
 9:     if $\nexists$ buffered geometry $G$ with $p \in G$ then
10:         $G \leftarrow$ GET_NEAREST_NEIGHBOR($p$)                ▷ nearest link
11:     $p \leftarrow$ PROJECT($p$, unbuffered $G$)
12:     Let $u, v$ be the end points of $G$
13:     set $G$ as inactive
14:     add edges $up$ and $pv$
15:     update mean_travel_time($up$, $pv$, $G$)          ▷ approx. time to traverse $up$ and $pv$
16:     set edges $up$ and $pv$ as active

17: procedure REMOVEFROMNET($p(x_p, y_p, t_p)$)
18:     identify the edges $Q, R$ where $p \in Q$, $p \in R$
19:     get the other end points $u, v$ of the edges $Q, R$
20:     remove edges $Q, R$ and vertex $p$
21:     reset edge $uv$ to active
```

consecutive location updates). In what follows, we consider the spatial network constraints to be reflected on the graph $\mathcal{N}$ that models the network. On the other hand, we assume the existence of background knowledge regarding the current temporal constraints imposed to the network to identify traffic jams, speed limits and other conditions that affect the time that is needed for the moving object to follow a given itinerary. Overall, the purpose of the reconstruction algorithm is to account for all these diverse constraints and choose the most probable itinerary when reconstructing the movement of the user.

The proposed reconstruction algorithm (Algorithm 22.1) applies an incremental, edge–by–edge strategy that iteratively constructs the 3D polyline of the user history of movement from the corresponding location updates. Specifically, given two consecutive location updates, the algorithm reconstructs the corresponding segment by identifying the most probable route that the users have followed through the (links of the) network. To achieve that, it compares the actual time that was needed by the moving object to cover the distance between two consecutive location updates to the mean travel time as attained

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

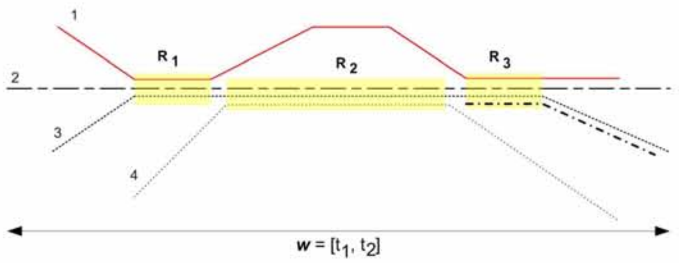


Figure 22.4: An example of frequent routes in the network aware model.

through the network (e.g., through estimates based on the current road traffic condition information). Algorithm 22.1 provides the details of our implementation. We should point out that an inactive node/edge in the network topology is a node/edge that does not take part in the computations. In the rare event that the collected user location update does not lie inside a buffered link of the network, we apply nearest neighbors matching to assign it to a link. Some alternative strategies can be found in [29, 181].

After the reconstruction of the user movement history, the next step is its decomposition into trajectories. This is an application–specific process and thus knowledge is required regarding the appropriate value of t_{stop} with respect to the considered application. Assuming that we have this kind of knowledge, the decomposition of the user movement history into trajectories is a straight–forward process.

22.2.2 Phase II: Identification of Frequent Routes

The knowledge of the trajectories of the users in the system allows us to compute their frequent routes and to classify these routes as either safe or unsafe for each individual user. The identification of the frequent routes of a user, proceeds as follows. Consider Figure 22.4 where a set of five trajectories are presented for a time interval $\boldsymbol{w} = [t_1, t_2]$. Assuming that all four trajectories belong to the same moving object o and the frequency threshold (see Definition 11 of Chapter 19) is 3, areas R_1, R_2 and R_3 correspond to frequently traveled routes for this object.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 22.2 Derivation of frequent routes in the network aware model.

Input: Object o along with all its trajectories tr; time interval $\boldsymbol{w}$ to search for frequent routes and frequency threshold $freq$.
Output: The frequent routes FR of object o in time interval $\boldsymbol{w}$.

```
1:  procedure FREQUENTROUTES($o$, $tr$, $\mathcal{N}$, $\boldsymbol{w}$, $freq$)
2:      $tr_w \leftarrow \{ tr \mid t \in \boldsymbol{w} \}$                    ▷ parts of the trajectories that are in $\boldsymbol{w}$
3:      declare Hash $Count\{\} \leftarrow \varnothing$
4:      foreach trajectory $T \in tr_w$ do
5:          $L \leftarrow$ GET_OVERLAPS(geom($T$), link_geom($\mathcal{N}$))
6:          foreach $l \in L$ do                                 ▷ for each link of $L$
7:              $Count\{l$.linkid$\}$++                        ▷ keep track of the traversal
8:      declare List $F \leftarrow \varnothing$                        ▷ list of frequently visited links
9:      foreach $f \leftarrow key$ (Count) do
10:         if Count$\{f\} \geq freq$ then
11:             INSERT($F, f$)                                   ▷ $f$ is frequent
12:     $FR \leftarrow (F, \boldsymbol{w})$                                 ▷ frequent routes of $o$ in $\boldsymbol{w}$
```

Algorithm 22.2 presents our implementation that derives the frequent routes of an object o when given *all* its trajectories in the system. The algorithm considers a pre–specified time interval $\boldsymbol{w}$ and retrieves all the trajectories (or routes) of o that are time–consistent with respect to $\boldsymbol{w}$. A trajectory (or route) is time–consistent with respect to a given time interval $\boldsymbol{w}$, if all its spatiotemporal points (x,y,t) have $t \in \boldsymbol{w}$. Threshold $freq$ indicates the minimum number of trajectories for the overlapping regions to be considered as frequent routes for o. For each trajectory of o in $\boldsymbol{w}$, the algorithm identifies the links that it traverses and keeps a counter on the traversals. Then, it uses the collected counters to retrieve the links of the network that are frequently traveled by the moving object. The derivation of the frequent routes of object o in $\boldsymbol{w}$ from the corresponding links is performed by examining the end points of the links and by joining together links that share end points. To identify the global frequent routes of a moving object, Algorithm 22.2 has to be executed for the different time intervals $\boldsymbol{w}$. Through the knowledge of the frequent routes of a moving object, the computation of its safe and unsafe routes, given a specific value of $\mathcal{K}$ as declared in the user profile, is straightforward.

In what follows, we investigate some issues that pertain to the identification of the frequent (safe or unsafe) routes of each moving object o in the system.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Selecting the Appropriate Time Intervals w

Since the identification of the frequent routes of a moving object o depends on a set of time intervals w, a first issue that needs further investigation regards the rationale behind the selection process for these intervals. By construction, w allows us to account for small time differences that are expected to be encountered in the regularly followed itineraries of a user. For example, a person may leave his/her house at 8:02am one morning to go to work, while the next day he/she leaves at 8:15am to follow the same itinerary. As is obvious, the system should be capable of identifying that the followed routes are the same (in a spatiotemporal sense) even if they do not perfectly coincide in time. Furthermore, depending on the existing network traffic conditions, a user may need a different amount of time to travel the same distance from one day to another.

Generally speaking, we consider that the selection of the appropriate size of w is an application–specific task. However, the size of w is expected to be smaller when user movement is more probable (e.g., at rush hours) and larger when user immobility is expected (e.g., during the late night hours). By using this convention, a reasonable amount of frequent routes is expected to be derived, leading to the best possible protection of user privacy. Finally, it is also possible to have the different time intervals (i.e. w's) be automatically adjusted by the system based on the movement history of each user (i.e. recorded as part of the user profile and subsequently used to derive the frequent routes for this particular user). This will allow for accounting for different movement behaviors of distinct users in the system when extracting their frequent routes.

Addressing (Computational) Cost–Related Issues

The computation of the frequent routes of a user is generally a costly operation, particularly due to the expected large number of the trajectories that have to be taken into consideration. However, there are two key issues that allow us to proceed towards this direction:

- The computation can proceed offline through the use of a dedicated server (possibly other than the trusted server) and thus do not affect the online operation of the system.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

When computed, the new frequent routes will replace the old ones that are currently associated with the user profile.

- The behavior of a user is not expected to radically change from day to day, and even if it does change, it will still take some time until the old frequent routes become invalid (i.e. until the new routes of the user become frequent). As a result, the frequent routes of a user do not have to be constantly updated but only once in a while, based on the used *freq* threshold.

22.2.3 Phase III: Trajectory $\mathcal{K}$–Anonymity

The third phase in the process of privacy preservation is the provision of trajectory $\mathcal{K}$–anonymity. If $\mathcal{K}$–anonymity is successfully provided to the user, then this is also the last phase of the process. Algorithm 22.2 computed the list of frequent routes *FR* for each moving object *o*. In order for the privacy preservation mechanism to adequately protect the privacy of a requester, the system should be capable of deciding when a user request for an LBS matches a frequent route of the user and, subsequently, whether this route is safe or unsafe. In what follows, we concentrate on the first part. Regarding the second part, the classification of the frequent routes of a user into safe and unsafe is a matter of building an efficient lookup table that maintains this information.

Matching User Requests to Frequent Routes

Consider Figure 22.5 that depicts a user initiating a request *r* for an LBS. Assume that the temporal coordinate of the request matches four of the frequent routes of the user (i.e. ‘a’, ‘b’, ‘c’ and ‘d’), each of which is buffered based on the width of the corresponding road. Moreover, as is shown in Figure 22.5, some of these routes share common regions (e.g., ‘a’ and ‘b’), while others are disjoint (e.g., ‘c’ and ‘d’). The system should be capable of identifying that this request was made from within the regions of a frequent route of the user and proceed to provide $\mathcal{K}$*–frequent* anonymity (if ‘a’ is a safe route) or $\mathcal{K}$*–present* anonymity (if ‘a’ is an unsafe route).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

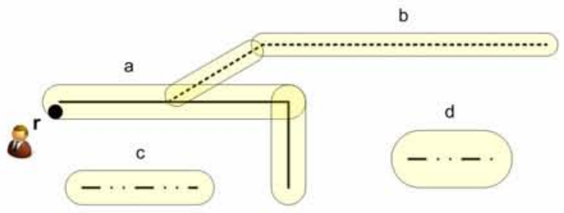


Figure 22.5: Matching user requests to frequent routes.

Algorithm 22.3 Match of requests in the network aware model.

Input: Current user request R_c, user's frequent routes FR, indicator *frun* of whether this is the first or an intermediate user request for this service and (if not the first request) the matched route of the last user request FR'.
Output: Boolean value denoting whether the match was successful or unsuccessful.

```
 1: function MatchRequest($R_c$, $FR$, $frun$, <in-out> $FR'$)
 2:     if $frun$ = true then                                              ▷ this is the first request
 3:         $FR \leftarrow \{(F, \mathbf{w}) \in FR | t \in \mathbf{w}\}$         ▷ frequent routes of the user in $\mathbf{w}$
 4:         $\mathcal{M} \leftarrow$ GET_GEOM_INSIDE($R_c|_{x,y}$, geom($FR$))
 5:         if $\mathcal{M}$ = null then
 6:             return false                                           ▷ request made from infrequent route
 7:         else
 8:             declare List $FR' \leftarrow \varnothing$
 9:             $FR' \leftarrow$ GET_ANYINTERACT($\mathcal{M}$, geom($FR$))    ▷ identify frequent routes in $FR$ that are not disjoint to the
    matched route $\mathcal{M}$
10:             return true                                            ▷ request made from a frequent route
11:     else                                                           ▷ not the first run
12:         $\mathcal{M} \leftarrow$ GET_GEOM_INSIDE($R_c|_{x,y}$, geom($FR'$))
13:         if $\mathcal{M}$ = null then
14:             $frun \leftarrow$ true                                 ▷ initialize for the next run
15:             return false                                           ▷ user left the frequent route
16:         else
17:             return true                                            ▷ user is still in a frequent route
```

The matching part of the trajectory anonymization process proceeds in two steps: The first step examines the location of request and identifies if it lies inside any of the frequent routes of the user. If this is the case, then the system takes the appropriate actions depending on whether the route is safe or unsafe. The second step takes place at each subsequent location update transmitted by the mobile device of the user and for as long as the user remains within a frequent route. If the user leaves his/her frequent routes, then the system offers $\mathcal{K}$*–present* anonymity and continues to check the subsequent locations of the user to detect whether the user entered a frequent route. This process continues until the service is provided to the requester.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 22.3 provides the details of our implementation. In the first run, the algorithm isolates from the set of all the frequent routes of the requester the ones that are time–consistent with respect to the time of request t. Then, we proceed to identify if the location (x,y) of the user at the time of request lies inside any of these geometries. The matching geometries, if any, are maintained in a list $\mathcal{M}$. Notation geom(*FR*) (lines 4, 9 and 12 of Algorithm 22.3) is used to demonstrate that only the spatial component (i.e. the geometry) of the frequent route is currently examined. Thus, the temporal component is ignored. In the event that the request was initiated at a location outside the frequent routes in *FR*, the user is provided with $\mathcal{K}$*–present* anonymity and the algorithm is re–executed in the next location update (unless the request is serviced in the meanwhile). Otherwise, the system identifies all the geometries that are not disjoint to $\mathcal{M}$ and stores them in a list *FR*' that is used in subsequent calls regarding the same service. List *FR*' stores the frequent route where the request was initiated, along with all the related (non–disjoint) frequent routes from *FR*. Considering the scenario of Figure 22.5, *FR*' will store routes 'a' and 'b'. This is due to the fact that since the user initiated the request when inside frequent route 'a', the only possible frequent routes that he/she can follow without traversing an infrequent route, are 'a' and 'b'. Given the fact that a user may have a substantial number of frequent routes in the considered time interval $\boldsymbol{w}$, the use of the routes in *FR*' alleviates the system from a lot of unnecessary work.

Offering Trajectory $\mathcal{K}$–Anonymity

Figure 22.6 presents the general case for the offering of trajectory $\mathcal{K}$–anonymity in the regions of the network aware model. Consider a part of the center of an imaginary city at a non–rush hour. At current time t_n a user makes a request r from the location depicted on the map. The system receives this request, examines its spatiotemporal coordinates, and matches it to a safe or an unsafe route of the requester. Then, the system examines the district where the user is located by depicting all other user (sub)–trajectories in a time period $\boldsymbol{w} = [t_{n-1}, t_n]$ that is close to the time of request. In the example scenario of Figure

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

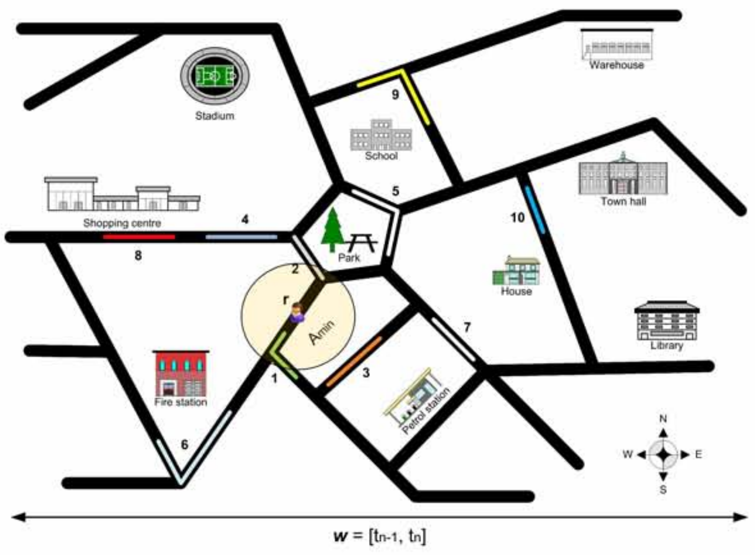


Figure 22.6: A scenario of K–anonymity in the network aware model.

22.6, there are 10 users other than the requester that move in this time period in this part of the city. To protect the user, the system needs to identify the $\mathcal{K}$–1 user trajectories that are nearest to the requester and that satisfy the criteria of the appropriate type of anonymity ($\mathcal{K}$–*present* vs. $\mathcal{K}$–*frequent*). This is achieved through the generalization of the spatiotemporal coordinates of the request. The anonymized request, which is then forwarded by the trusted server to the appropriate service provider, has its spatial coordinates enclose the anonymity set and the temporal coordinate reflect the time period $\boldsymbol{w}$. Furthermore, the generated $\mathcal{K}$–ASR is matched to a circle that is centered at the centroid of the collection of geometries that correspond to the routes of the $\mathcal{K}$–1 nearest neighbors to the requester in the time period $\boldsymbol{w}$, plus the location of request. Depending on the value of A_{min}, the area of the circle is adjusted if necessary to adhere to this constraint. Furthermore, if the computed area exceeds the A_{max} threshold that is required for the provision of this service, $\mathcal{K}$–anonymity fails and the user is notified. Otherwise, the computed ASR replaces the actual coordinates of the original request and provides its $\mathcal{K}$–anonymized equivalent.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Algorithm 22.4 Offering of trajectory $\mathcal{K}$–anonymity in the network aware model.

```
Input: User request $R_c$, value of $K$ in $K$–anonymity, minimum generalization area $A_{min}$.
Output: The computed circular anonymity region $\mathcal{C}$ that satisfies $A_{min}$.

 1: function GENERALIZEREQUEST($R_c$, $K$, $A_{min}$)
 2:     declare List_of_Geom $\mathcal{AS} \leftarrow \varnothing$          ▷ initialize the anonymity set
 3:     declare Geometry $\mathcal{C} \leftarrow \varnothing$
 4:     $\mathcal{AS} \leftarrow$ GET_K_NN($R_c$, $K$–1, $\alpha$)          ▷ find the nearest neighbors of the requester
 5:     $C_K \leftarrow$ GET_CENTROID($\mathcal{AS}$, $\boldsymbol{w}$)          ▷ centroid of the routes in $\boldsymbol{w}$
 6:     $R_K \leftarrow$ GET_DISTANCE($C_K$, $K$–th NN)          ▷ compute the distance from the center to the furthest neighbor
 7:     if $R_K < \sqrt{\frac{A_{min}}{\pi}}$ then
 8:         $R_K \leftarrow \sqrt{\frac{A_{min}}{\pi}}$          ▷ ensure that the distance fulfils the $A_{min}$ requirement
 9:     $\mathcal{C} \leftarrow$ CREATE_CIRCLE($C_K$, $R_K$)          ▷ create the anonymity region $\mathcal{C}$
10:     return $\mathcal{C}$
```

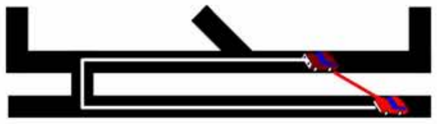


Figure 22.7: Euclidian distance (red) vs network distance (white) for the computation of the participants in the anonymity set.

Algorithm 22.4 presents the way that trajectory $\mathcal{K}$–anonymity is delivered to a user that sent a request R_c given threshold A_{min}. The goal of the algorithm is to identify the $\mathcal{K}$–1 trajectories of the users, for the time interval $\boldsymbol{w} = [t_{n-1}, t_n]$ that satisfy the appropriate anonymity requirements and to return the region that covers all the $\mathcal{K}$ users (including the requester). An important observation is that a trajectory does not need to lie completely inside the considered spatial region, for its user to be encountered in the anonymity set of o. Rather, even if a small part of the trajectory is inside the spatial region at some time in $\boldsymbol{w}$, the corresponding user is counted in the anonymity set. Another important issue relates to the calculation of the distances between the point of request and the various trajectories in the system. In our implementation, the distance between the requester and a trajectory is captured as the minimum distance between the point of request and the nearest point of the trajectory to the requester. This means that distances are not computed based on the underlying network of movement, a choice that is appropriately justified in Figure 22.7.

Algorithm 22.4 identifies the $\mathcal{K}$–1 routes that are nearest to the requester in $\boldsymbol{w}$ and stores them in a list $\mathcal{AS}$. The α parameter in the computation of the $\mathcal{K}$ nearest neigh-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

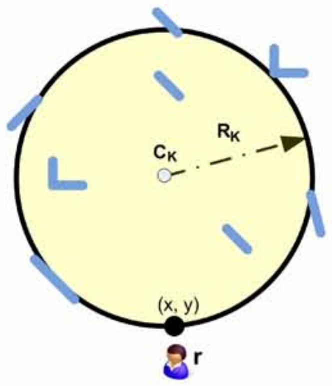


Figure 22.8: An example of computed region of $\mathcal{K}$–anonymity.

bors (including the requester) indicates the type of anonymity that is sought ($\mathcal{K}$–*present* vs. $\mathcal{K}$–*frequent*) to allow the algorithm to search among the corresponding users. Having identified the participants of the anonymity set, the algorithm computes the centroid of the corresponding routes (with respect to $\boldsymbol{w}$) in $\mathcal{AS}$. This point C_K will be the center of the ASR. Following that, the algorithm computes the distance of C_K to the most distant route in $\mathcal{AS}$. The final step of Algorithm 22.4 is to compute the circle $\mathcal{C}$ centered at C_K and having a radius R_K. This circle corresponds to the $\mathcal{K}$–ASR for user r. To ensure that the anonymity region satisfies the A_{min} privacy requirement, the computed radius of the circle is tested against the minimum acceptable radius $\sqrt{A_{min}/\pi}$ and is augmented if necessary. Figure 22.8 presents the way Algorithm 22.4 offers 9–anonymity to user r. The center of the circular anonymity region corresponds to the centroid of the users. Notice that by construction, all the $\mathcal{K}$ users in the anonymity region are equi–probable to be the senders of request r, since the centroid of the routes in $\mathcal{AS}$ is uniquely defined.

In a service that can be completed within one location update, Algorithm 22.4 produces the anonymity region that replaces the space/time coordinates of the original request. On the other hand, if multiple location updates are necessary for the completion of the service, then the following strategy is applied. First, Algorithm 22.4 is executed and the area of request is generalized to satisfy $\mathcal{K}$–anonymity. The $\mathcal{K}$–1 subjects of the computed anonymity set are maintained by the trusted server for as long as the current service with-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

stands completion. At each subsequent location update that needs to be transmitted to the service providers, the trusted server adjusts the time coordinate of the request and applies Algorithm 22.4 to generate a $\mathcal{K}$–ASR that contains (possibly among others) the original $\mathcal{K}$ users. This new ASR, provided that it satisfies the A_{max} threshold, blocks the sequential query tracking attack and offers privacy to the requester. However, if the A_{max} threshold is surpassed, the privacy of the user is under threat and a service delaying or denial mechanism has to take over.

22.2.4 Phase IV: Service Delay or Denial

There are circumstances when the proposed approach for the offering of trajectory $\mathcal{K}$–anonymity to the requesters of LBSs may fail. This situation may occur for several reasons, such as an exceptionally high value of $\mathcal{K}$ defined in the profile of the requester or tight spatiotemporal generalization constraints (A_{max}, $\boldsymbol{w}$) required for the provision of the requested service. In all such cases, an approach has to be employed to effectively protect the identity of the requester. The proposed approach introduces a small delay in the servicing of the request, in the hope that in the meanwhile (i) more users will approach the requester, thus become part of his/her anonymity set, and/or (ii) the requester will move to a more populated region. Specifically, we defer the servicing of the request for a pre–specified time period and then re–execute Algorithm 22.4 for the provision of trajectory $\mathcal{K}$–anonymity. If the algorithm fails again, then we protect the privacy of the user by denying the servicing of his/her request.

22.2.5 Achieving Real–Time Operation

Offline operations The most important offline task of the proposed system is the derivation of the frequent routes of the users. Since these routes are not expected to change very often, their computation does not add an overhead to the regular real–time operation of the system. Furthermore, the use of the spatial functionality of the spatial DBMS allows this process to be conducted in an efficient manner.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Online operations During online operation the purpose of the system is to receive user requests for LBSs and to service them. When a new request r is received by the trusted server, the system examines the location of the requester with respect to his/her frequent routes and proceeds to provide the necessary type of $\mathcal{K}$–anonymity. The time required for this decision is low, since the matching of the user requests to the corresponding frequent routes is performed by making use of the spatial indexes of the DBMS. Furthermore, subsequent runs of the algorithm for the same request are serviced much faster since the regions of interest, where the search will be conducted, are further reduced and isolated from the set of all the frequent routes of the user. Regarding the provision of trajectory $\mathcal{K}$–anonymity, the generalization strategy makes use of the spatial functionality of the employed DBMS to efficiently identify the neighbors of the requester.

22.3 System Implementation Details

In this section, we investigate some details that pertain to the implementation of the presented algorithms as part of the trusted server. Our implementation relies on a spatial database engine that stores all the necessary types of spatial (network of user movement) and spatiotemporal (movement history, trajectories, frequent routes) data and efficiently performs a set of spatial and temporal calculations for the provision of trajectory anonymization. A table stores all the users in the system along with references to their profiles and histories of movement. The profile of a user corresponds to a set of triples $\langle sid, K, A_{min}\rangle$ and is kept in a (spatial) table. Furthermore, tight to each profile is the set of frequent routes of the corresponding user. These, are kept in a (spatial) table consisting of tuples in the form of $\langle routeID, route, TimeInterval\rangle$, where the *route* depicts the traveled route through the network as a collection of the links (i.e. spatial geometries) that were traversed by the user, and the *TimeInterval* corresponds to the (unanchored) time that this route is typically followed by the user. For each frequent route, a table stores the values of $\mathcal{K}$ for which this route is safe or unsafe for the user. The user history of movement is kept in two tables:

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 22.1: Mapping the spatial functions to their equivalents in Oracle and DB2.

Spatial Function	Oracle Spatial DBMS	DB2 Spatial DBMS
GET_m_SHORTEST_ROUTES	ALL_PATHS(*m*, constraints)	*unsupported*
GET_NEAREST_NEIGHBOR	NEAREST_NEIGHBORS(1)	*unsupported*
GET_OVERLAPS	SDO_OVERLAPS	ST_Overlaps
GET_GEOM_INSIDE	SDO_INSIDE	ST_Contains
GET_ANYINTERACT	SDO_ANYINTERACT	ST_Relate
GET_K_NN_DISTANCE	SDO_NN & SDO_WITHIN_DISTANCE	ST_Distance
CREATE_{CIRCLE, SQUARE}	SDO_GEOMETRY	ST_Geometry
GET_CENTROID	SDO_GEOM.SDO_CENTROID	ST_Centroid

(i) the table of the complete history, and (ii) the table of the user trajectories. Both tables store the data as a collection of geometries related to the actual segments, their start–time and end–time. A *sequenceID* is tight to each segment to denote its position in the geometry and to allow the unification of the segments to build larger parts of a trajectory. Besides the user–related data, the spatial DBMS also stores the underlying network topology $\mathcal{N}$ of user movement through a set of special–purpose structures provided by the DBMS. Finally, all the geometries in the spatial tables are indexed and the metadata of the network are updated to include the appropriate costs for the various calculations.

The implementation of the proposed framework is based on the Oracle DBMS and Java, and makes use of the Oracle Java API [4] to allow their communication. The choice of Oracle was made primarily due to its inherent support for spatial networks, a support that is currently missing from other DBMSs. On the other hand, Java allows for the maximum open–endness of our framework. As a first step in our implementation, we use a network aware trajectory data generator that generates a set of trajectories (as series of location updates) which are consistent to a given network. In sequel, both this data and the underlying network are transformed into the appropriate formats and are stored in Oracle. Finally, through the supported spatial functions of the DBMS our privacy model reconstructs the user history, computes the frequent routes and provides trajectory $\mathcal{K}$–anonymity. Although our implementation was based on the spatial engine of Oracle, the proposed framework can be easily migrated (with the necessary adjustments) to any other spatial DBMS. To support this claim, Table 22.1 summarizes the spatial functions that were used in the provided algorithms and presents their equivalents for two major spatial DBMSs, namely Oracle [2]

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 22.2: The characteristics of the three datasets.

Parameter	Dataset 1	Dataset 2	Dataset 3
#objects	4,100	9,200	96,500
#classes	2	6	6
max time	2,000	3,000	8,000
report probability	1	0.9	0.7

and IBM DB2 [5]. Similar functions can be found in other DBMSs with spatial capabilities, as well as functions needed for the computation of buffers, the identification of the length of roads or road parts, and the identification of geometries that are within a distance from an origin. This functionality is delivered in Oracle through the spatial functions SDO_BUFFER, SDO_LENGTH and SDO_WITHIN_DISTANCE. We should point out that although Oracle is a commercial DBMS, the functionality that we used from Oracle is freely available through the Oracle's website (`http://download.oracle.com`).

22.4 Experimental Evaluation

The proposed algorithms were implemented by using Java and Oracle 10g [2] on Windows XP on a 3.2 Ghz Intel Pentium D processor equipped with 4GB of main memory. Several parts of the needed functionality were written in Oracle PL/SQL. To experimentally evaluate the algorithms, we used Brinkhoff's network aware generator of moving objects [30] and generated three datasets of trajectories based on the road network of the Oldenburg city. Brinkhoff's generator is commonly used in most of the papers related to spatiotemporal datasets, as it models real world networks with synthetic data. The generated datasets experience different characteristics in terms of the number of moving objects and their classes, the maximum time of movement, and the probability of an object to report its position at each time. Table 22.2 summarizes their properties. The number of classes denotes the different types of moving objects (e.g., cars, trailers, motorbikes) that are considered in the generated data. Each moving object is assigned to a unique class that defines its maximum allowable speed in the network. Furthermore, the time of generation, the duration

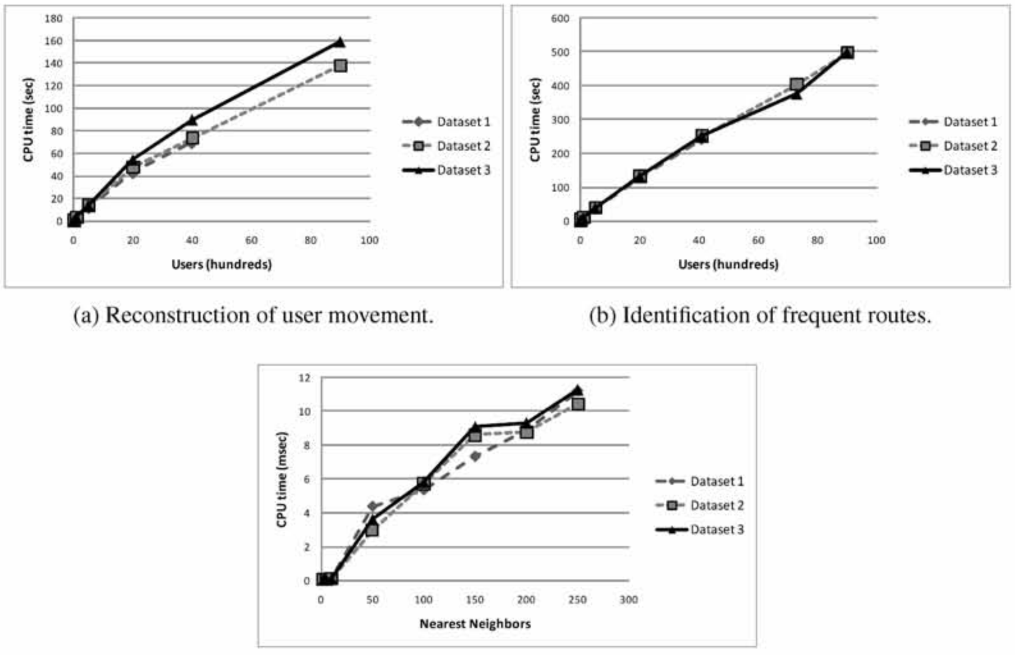
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

of existence and the time of disappearance of a moving object are different for the various objects. Among the three datasets, Dataset 3 provides a better view of how the privacy model would operate on a real system for the offering of LBSs, due to its size and the irregular reporting of location updates. On the other hand, as the experiments demonstrate, the time that is needed for the provision of $\mathcal{K}$–anonymity is approximately the same for all the three datasets and is kept low. In conformance to Brinkhoff's generator, we use an integer type as the unit time instance *un* and set the whole time period from 0 to the corresponding maximum time as reported in Table 22.2. Note that since our framework is unique in the utilization of the underlying network for the offering of online trajectory anonymity, no comparisons have been made with other approaches.

In Brinkoff's generator, each object is routed through the links of the network to move from its origin to its final destination. The use of a routing algorithm for the generation of movement has as a consequence that no subtrajectory of an object is repeated and thereof no frequent routes exist. To solve this issue, for each moving object we randomly selected a set of consecutive location updates (creating a temporal interval $\boldsymbol{w}$) and produced 3 new trajectories of this object that contained only these updates. Thus, we achieve to artificially generate frequent routes for the various users. The resulting datasets were used in the experiments that follow.

We conducted three sets of experiments. The first set, presented in Figure 22.9, tests the scalability of the proposed algorithms. Specifically, Figure 22.9(a) shows the time that was needed to update the last segment of a user's history of movement (for a number of users), given knowledge of the previous location of the user. The approach first checks if the new location update lies in the same link of the network as the last known, and otherwise executes Algorithm 22.1. As one can observe, the runtime for Dataset 3 is higher when compared to the other two datasets, since, due to the irregular location updates, Algorithm 22.1 has to be executed more often. Moreover, since we have no knowledge regarding the time needed for the traversal of each link of the network, Algorithm 22.1 identifies the shortest route with respect to the distance that needs to be traveled by the moving object.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Reconstruction of user movement. (b) Identification of frequent routes.

(c) Provision of $\mathcal{K}$–anonymity.

Figure 22.9: Scalability of the algorithms in the network aware model.

Figure 22.9(b) shows the time needed to identify the frequent routes of each user and to generate the corresponding lookup table. In this experiment, the ***w*** parameter is adjusted based on the values that were selected on the trajectory addition stage, described earlier. The value of *freq* was set to 3. Figure 22.9(c) presents the time needed to compute the nearest neighbors to a requester, which is the primary operation of the proposed $\mathcal{K}$–anonymization strategy. As one can observe, the figure demonstrates the runtime that is needed for the computation of the anonymity set in $\mathcal{K}$*–present* anonymity since it selects all the nearest neighbors of the requester. However, the runtime of $\mathcal{K}$*–frequent* anonymity (for the same value of $\mathcal{K}$) can be approximated through the graph by considering more that neighbors than $\mathcal{K}$ have to be found near the requester. For this experiment, a hundred requests were randomly generated, each by selecting a location update from the history of movement of a user, considering this as the point of request and setting ***w*** to have an extent of 10 location

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

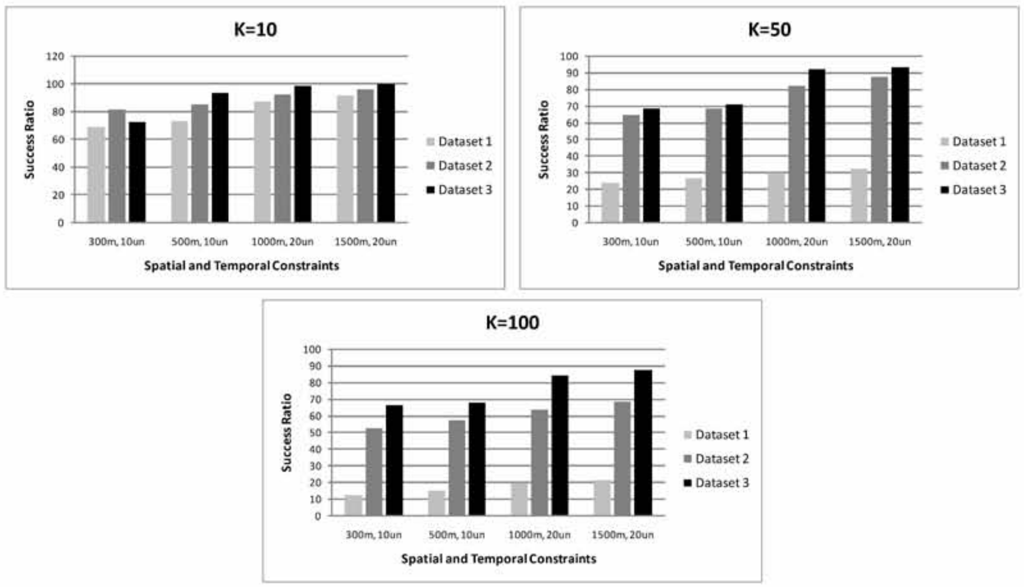


Figure 22.10: Success ratios for 10, 50 and 100−*present* trajectory anonymity on continuous queries.

updates[1]. The reported times are averages over 100 runs.

The second and third sets of experiments (shown in Figure 22.10 and Figure 22.11, respectively) illustrate the success ratio of the proposed $\mathcal{K}$–anonymity approaches, when applied on continuous queries where the servicing of the request requires 3 location updates. Each experiment involves a set of spatial (denoted as the maximum radius of the $\mathcal{K}$–anonymity region) and temporal (denoted as the maximum number of time units *un* prior to the current time) constraints, and a value of $\mathcal{K}$ in 10–100. In all experiments, we set the A_{min} parameter such that the anonymity set is guaranteed to cover a circular area of radius equal to 50 meters, away from the requester. In both cases of trajectory anonymity, failure corresponds to the lack of identifying the necessary neighbors of the requester throughout the period of servicing the request. The reported ratios are an average over 100 runs coming from randomly selected users and spatiotemporal points in their trajectories. We should point out that the success ratio of continuous queries is (as expected) lower when compared

[1]Please notice that this $\boldsymbol{w}$ refers to the extent of the temporal cloaking that is offered by the $\mathcal{K}$–anonymity algorithm and is not related to the time interval that is used for the identification of the frequent routes of a user.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

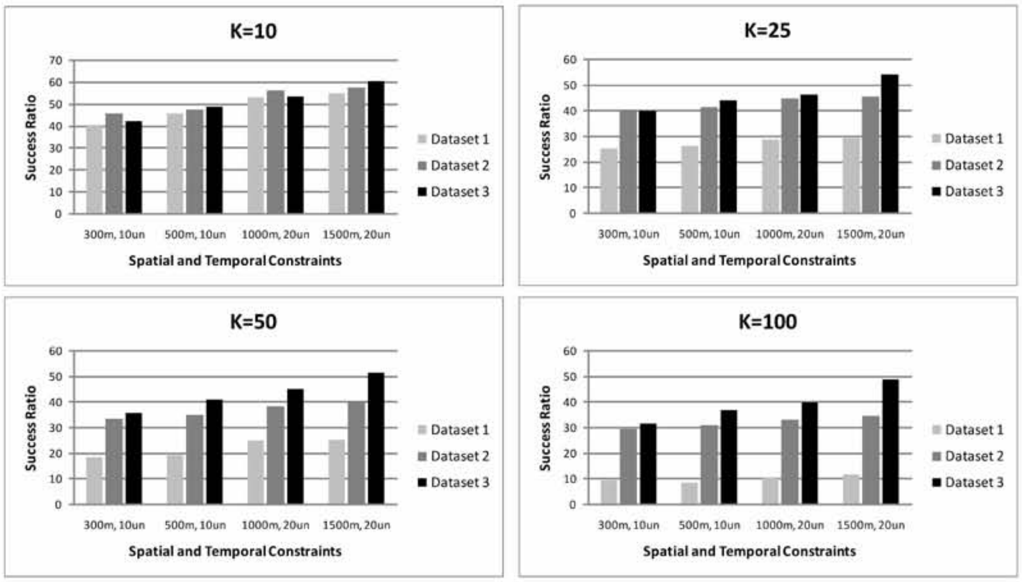


Figure 22.11: Success ratios for 10, 25, 50 and 100−*frequent* trajectory anonymity on continuous queries.

to snapshot queries, since $\mathcal{K}$–anonymity has to be offered to the requester from the time of request until the completion of the requested service. Furthermore, as is shown in Figure 22.11, $\mathcal{K}$–frequent anonymity has a lower success ratio when compared to $\mathcal{K}$–present anonymity for the same values of $\mathcal{K}$, due to the extra requirements that are have to be met by the participants that comprise the $\mathcal{K}$–ASR.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 23

PLOT: Privacy in LBSs: an Open–Ended Toolbox

As highlighted in Chapter 17, a wide variety of approaches have been proposed for the offering of privacy in LBSs. To recapitulate, the research on location privacy has primarily focused on the following directions: (i) the use of access control policies that state explicitly how the subscriber's location information is treated (e.g., [45, 89, 157]), (ii) the use of cryptographic techniques, in which the communicated location information is ciphered to shield the privacy of the user (e.g., [88, 91, 107]), (iii) the use of obfuscation techniques to confuse an adversary regarding the real location of the user (e.g., [19, 25, 49]), and (iv) the use of spatial or spatiotemporal cloaking to lower the resolution of the exact user location prior to submitting the request to the LBS provider (e.g., [39, 82, 100, 126]). The approaches for the offering of privacy in LBSs can be categorized as either centralized or distributed, depending on the entity of the system that is responsible for the offering of location privacy. Between the two categories, centralized approaches have attracted most of the attention, particularly due to the fact that they are based on the trusted server framework that is commonly met in existing systems (e.g., PayPal [137]), and also as they require no special hardware on the clients' mobile devices. Although many centralized location privacy approaches have been proposed, until recently no attempt had been made to provide

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

the means for comparing these approaches in a unified framework.

In this chapter we introduce PLOT (originally proposed in [78]), the first open–ended toolbox that integrates centralized approaches for location privacy in LBSs, in a common framework. The proposed toolbox *offers the essential functionality for the implementation of novel location privacy approaches, while serving as a testbed for the comparison of existing schemes*. The key aspects of PLOT can be summarized as follows:

- PLOT supports privacy algorithms which operate on real and synthetic movement data. Our framework provides tools for mobile data preprocessing, movement reconstruction and movement segmentation to user trajectories.

- PLOT supports the implementation of both network aware and free terrain solutions to location privacy. To this end, it utilizes the spatial capabilities offered by modern DBMSs to model the movement of the users in the system, as well as the underlying network topology of user movement.

- PLOT implements strategies for the identification of the frequent movement patterns of the users in the system. Knowledge of these patterns allows for additional decisions to be made by the privacy algorithms, with respect to the privacy that is offered to the requesters of LBSs.

- PLOT provides tools for testing specific location privacy approaches, as well as for comparing different location privacy algorithms. Furthermore, it offers an extended set of visualization tools that provide insight on the workings of the implemented solutions and facilitates the qualitative and quantitative evaluation of their behavior.

The rest of this chapter is organized as follows: In Section 23.1 we introduce the system architecture of PLOT, while Section 23.2 delivers PLOT's functionality. In Section 23.3, we present the graphical user interfaces of the toolbox, while Section 23.4 sheds light on the key aspects of its implementation. Finally, Section 23.5 delivers a use case scenario of PLOT that demonstrates a portion of its functionality.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

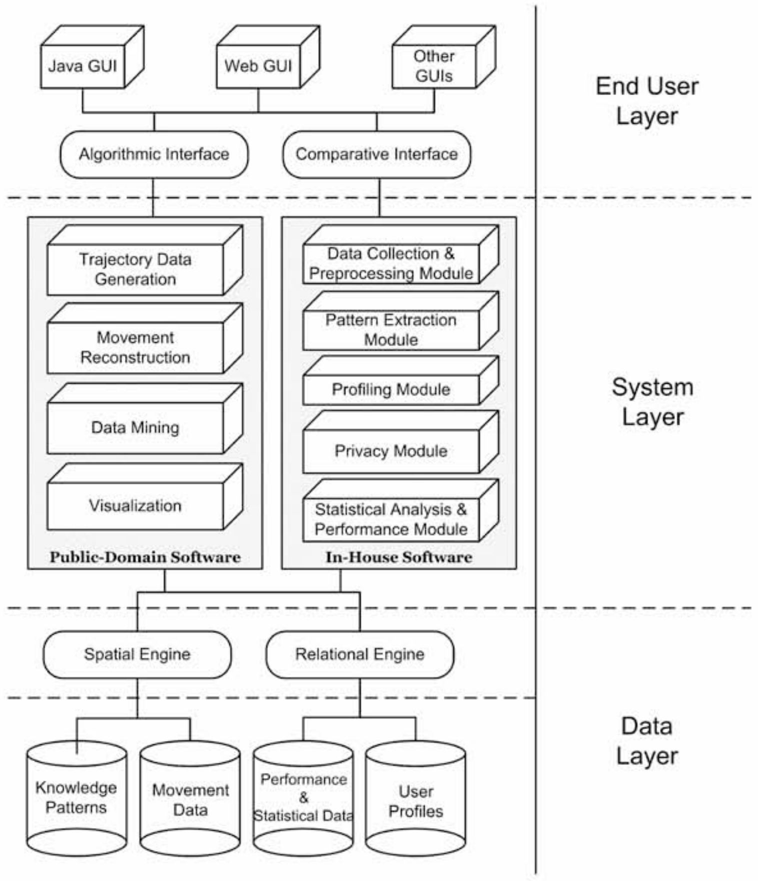


Figure 23.1: The PLOT's three–tier system architecture.

23.1 System Architecture

PLOT is a privacy toolbox that can be used to build complete privacy models for LBSs by tuning a few parameters and plugging in some in–house made and public–domain privacy schemes, as well as to experimentally evaluate different methodologies for the offering of privacy in LBSs. It encompasses a variety of tools and models proposed thus far in the literature for protecting the privacy of the users in LBSs, and includes performance and accuracy metrics to compare these different models. Figure 23.1 presents the three–tier system architecture of PLOT. In what follows, we provide details on the functionality of the different layers that comprise the system architecture.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

23.1.1 End User Layer

Through the End User layer, the end users can gain access to a set of privacy algorithms for LBSs and evaluate them on both real and synthetic movement data, as well as compare different privacy approaches in terms of accuracy, quality of service and computational cost. Furthermore, the End User layer provides the necessary tools to support the implementation of new algorithms for privacy in LBSs, as well as a series of visualization tools.

23.1.2 System Layer

The System layer implements the core functionality that is offered by PLOT. To this end, it supports all the necessary operations to carry out tasks that involve (i) the connection of the end user to the database server, (ii) the generation (or loading) of movement data, as well as topological data regarding existing constraints imposed on the movement of the users in the system, (iii) the preprocessing of the movement data, its reconstruction based on the collected location updates, and its segmentation into a set of user trajectories, (iv) the automatic extraction of mobility patterns from the user trajectories corresponding to frequently traveled routes by each user, (v) the offering of location or trajectory privacy in LBSs, and (vi) the presentation of results and statistics by means of performance tables and graphs that help in the evaluation of the employed privacy methodology.

The System layer is organized as a set of modules, each of which implements a specific task. The modules can be separated into two categories: public–domain and in–house. Public–domain modules involve external implementations of algorithms that can be plugged into PLOT to enhance its capabilities for data generation, movement reconstruction, data mining, privacy preservation and visualization. On the other hand, in–house modules implement a set of methodologies that have been especially designed for PLOT. Currently, PLOT supports in–house and public-domain software methodologies for movement data collection and preprocessing, location and trajectory privacy, statistical analysis and performance evaluation.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 23.1: The PLOT tools list.

Tools Category	**Functionality Supported by PLOT**
Data Input & Preprocessing Methods	– DBMS (Oracle, DB2, SQL Server) – User movement (location updates) – Network topology – User requests – Editing of movement data – Exclusion of portions of movement
Movement Reconstruction & Segmentation Methods	– Linear reconstruction – Arc reconstruction – Link based reconstruction – No segmentation – Temporal displacement – Spatial displacement
Frequent Movement Derivation Methods	– All requests are unsafe – Read from file (user–specified) – Grid/Area based – Link based
Location / Trajectory Privacy Algorithms	– Historical $\mathcal{K}$–anonymity – Spatial $\mathcal{K}$–anonymity – Trajectory $\mathcal{K}$–anonymity – Unlinking algorithms
Results & Statistics Visualization Tools	– Visualization of the anonymity area – Time/Space efficiency – Success ratio – Quality of service – Export report to file

23.1.3 Data Layer

The Data layer facilitates the efficient storage, manipulation and retrieval of the spatial, the spatiotemporal, and the relational data that support the various tasks undertaken by PLOT. The spatiotemporal data that involves the movement of the individuals, as well as the knowledge patterns that are derived from the user trajectories (e.g., the frequent routes of the users) is stored in a set of spatial tables inside the used DBMS. Moreover, a set of spatial tables in the DBMS are used to store the network topology in the case of a network–based privacy methodology. On the other hand, the user profiles, as well as performance and statistical data produced by the employed location privacy methodology, are stored in relational tables inside the DBMS. The user profiles are essential to model the different privacy requirements of each user in the system, which are then used by the privacy algorithm to offer the requested level of privacy to the corresponding user.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

23.2 System Support and Functionality

Table 23.1 presents the full set of tools that are currently available in PLOT to support the offering of privacy in LBSs. PLOT provides to the end user the necessary tools to support a variety of input types, such as different network topologies, multiple DBMSs with spatial capabilities, user movement data as a series of location updates, generation or loading of user requests for LBSs, and user profiles that store the privacy requirements for each user in the system. In what follows, we provide details on the functionality of PLOT.

23.2.1 Data Input and Preprocessing Methods

Data Input

The data input methods enable the end user to (i) connect to the DBMS of choice that will be used to store and manipulate the movement data, (ii) load the movement data for the various users in the system, (iii) load the user requests for LBSs, and (iv) provide the network topology, which will be used by the privacy algorithm to protect the requesters of LBSs in a network based setting. In PLOT, user movement can be either read directly from a file or generated on–the–fly by using Brinkhoff's generator [30]. Furthermore, PLOT supports three major DBMSs with spatial capabilities: Oracle [2], IBM DB2 [5] and Microsoft SQL Server [6]. Among the three DBMSs, only the Oracle DBMS provides the necessary algorithms for the storage of a network topology and thus it is the only one that can be used in conjunction with the network aware model of user movement.

Data Preprocessing

The data preprocessing methods allow the end user to make changes to the movement data prior to its use by the employed privacy methodology. PLOT supports both the direct editing of the users' movement data to allow for immediate changes of some of the existing values, as well as it provides algorithms for the exclusion of portions of users' movement through the application of various filters. The supported exclusion filters are based on spatial and/or

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

temporal constraints that are applied to the movement of one or more users in the system. The outcome of the preprocessing phase can be given to the end user to be informed about the exclusions that were made on the dataset prior to using the selected privacy algorithm.

23.2.2 Movement Reconstruction and Segmentation Methods

Movement Reconstruction

Reconstruction algorithms apply an interpolation approach that utilizes the underlying topology of user movement in order to make the best "guess" regarding the specific path that each user commutes from one location update to another. The reconstruction of the movement of the users is an essential step for the computation of the frequent movement patterns of each user and the subsequent operation of certain privacy methodologies, such as [26,69,79,185]. PLOT supports methods that are suitable for the reconstruction of movement both in a free terrain and in a network based model of movement. In the case of the free terrain, PLOT supports the movement functions of HERMES [138] that involve both linear and non–linear movement interpolation. On the other hand, in the case of the network aware model of movement, PLOT implements the link based approach of [79] that utilizes the topology of the network to decide upon the movement of the users. The results of movement reconstruction can be provided to the end user at the completion of this process.

Movement Segmentation

PLOT offers a set of methodologies that segment the reconstructed histories of users' movements into a set of trajectories per user. To avoid unwanted segmentation, the segmentation algorithm has to be parameterized to take into account the intrinsics of the dataset at hand. PLOT offers the following options with respect to movement segmentation: (i) movement is not segmented into trajectories, (ii) movement is segmented based on the identified temporal displacement of the objects, and (iii) movement is segmented based on the identified spatial displacement of the objects. Methods (ii) and (iii) can be used together to provide a more sophisticated approach to movement segmentation.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

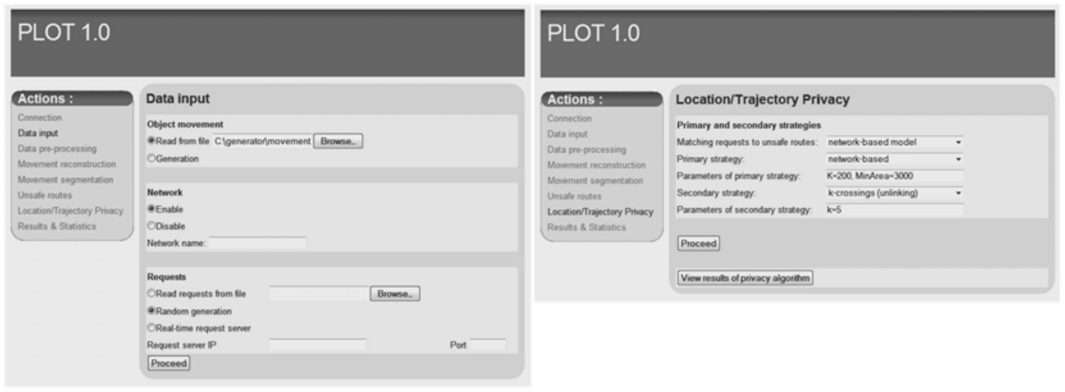
23.2.3 Frequent Movement Derivation Methods

PLOT implements the algorithms that were proposed in [69, 71, 77, 79] for the identification of routes that are frequent for each user. The frequent routes of a user are subsequently categorized as either safe or unsafe for the privacy of the user based on their support by the rest of the population in the system. To derive the frequent and the unsafe routes of a user, PLOT implements the strategy of [69] in the case of the free terrain model and the strategy of [77] in the case of the network based model. Specifically, the end user is provided with the following options: (i) all requests for LBSs are considered as unsafe, (ii) the unsafe routes are read from a file, (iii) the unsafe routes are computed by using the grid/area based strategy of [69], and (iv) the unsafe routes are computed by using the link based strategy of [77]. Furthermore, the end user is free to select the temporal resolution, as well as the time interval in which the frequent routes should be computed.

23.2.4 Location/Trajectory Privacy Algorithms

PLOT can support all types of centralized privacy algorithms that were presented in Chapter 17. The current implementation of PLOT supports the historical $\mathcal{K}$–anonymity approaches of [26, 185], the spatial $\mathcal{K}$–anonymity approaches of [82, 126], as well as the trajectory $\mathcal{K}$–anonymity algorithms of [69] and [77]. We call these approaches as "main privacy algorithms" as they are applied upon the receipt of a user request for an LBS, in order to protect the privacy of the requester. Upon its completion, the main privacy algorithm informs the trusted server whether it succeeded or failed in protecting the privacy of the requester. In the event of failure, a secondary location privacy strategy is used that (typically) has more chances to succeed in protecting the privacy of the requester, however at an additional cost to the quality of the offered service. Currently, PLOT supports the unlinking algorithms proposed in [26, 48, 185], that are used as the secondary privacy algorithms when the main approach fails to guarantee the $\mathcal{K}$–anonymity of the requester.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



Data input

Object movement

[✓ Read from file] C:\generator\movement

Generation

Network

[✓ Enable]

Disable

Network name:

Requests

Read requests from file

[✓ Random generation]

Real-time request server

Request server IP

Port

Location/Trajectory Privacy

Primary and secondary strategies

Matching requests to unsafe routes:	network-based model
Primary strategy:	network-based
Parameters of primary strategy:	K=200, MinArea=3000
Secondary strategy:	k-crossings (unlinking)
Parameters of secondary strategy:	k=5

Figure 23.2: A portion of the web–based client graphical user interface of PLOT.

23.2.5 Results and Statistics Visualization Tools

PLOT incorporates a set of visualization tools that evaluate the performance of the privacy algorithms in terms of time/space efficiency, accuracy (i.e. success ratio) and quality of service (query delaying/dropping and size of the ASR). Through the provided visualization tools, the end user can gain insight on the workings of the implemented privacy solutions, as well as evaluate their behavior under a wide range of different parameter configurations. Furthermore, PLOT provides the end user with the option to visualize the computed ASR, which depicts the area where $\mathcal{K}$–anonymity was delivered to the requester of an LBS.

23.3 Graphical User Interfaces

PLOT provides the end users with two different GUIs (a web–based GUI and a java–based GUI) for interacting with the system. By using either GUI, the end user is able to select one or more centralized privacy algorithms that will operate on real or synthetic movement data, apply a set of data preprocessing, movement reconstruction and movement segmentation techniques, and receive a set of results that provide insight on the workings of the employed methodologies. Furthermore, the end user can adjust a set of parameters to determine the functionality of the various components of the system, as presented in Section 23.2. In Figure 23.2, we present a portion of the web–based GUI frontend that is offered by PLOT.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

In particular, on the left of Figure 23.2 is a screenshot of the data input tab of the web–based GUI. This tab collects the different types of input that have to be provided by the end user prior to executing a location privacy technique. The first type of input involves the collection of the data that represents the movement of the objects in the system. In PLOT, users movement can be either read directly from a file or generated on–the–fly by using Brinkhoff's generator [30]. The second type of input is optional and involves the underlying topology where the users move. Information on the topology of the network of movement can be useful in tasks such as movement reconstruction, identification of unsafe routes and trajectory privacy. The final type of input regards the generation of the requests for LBSs. PLOT supports the following options: (i) reading requests from a file, (ii) generating requests at randomly selected location updates from the trajectories of the users, and (iii) using a real–time request server to generate user requests for LBSs.

23.4 System Implementation

In this section, we provide the details that pertain to the system implementation of PLOT. PLOT relies on the web services paradigm (c.f. [33]), which provides the means for interoperable, machine–to–machine interaction over a network. The application server (hereafter called the "PLOT server") deploys a set of methods over the PLOT web service, which are used to deliver the functionality offered by the toolbox. Each method (residing on the System layer) delivers part of the system functionality and communicates with a database server to store, manipulate and query data that resides within a database. On the client–side (End User layer), a PLOT client can remotely execute any method that is supported by the PLOT server by issuing the corresponding service call.

For the server–side implementation of PLOT, we used the Jboss application server [121], which is an open source implementation of the Java EE suite of services. Jboss provided us with all the necessary components to build our application, while it enabled the easy migration of our solution to any operating system and platform that supports the Java 5.0 Runtime Environment (JRE).

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

To support the efficient handling of movement data, as well as to support the network topology in network based solutions, PLOT uses the spatial capabilities of modern DBMSs, such as Oracle, IBM DB2 and Microsoft SQL Server. The PLOT web service communicates with the appropriate DBMS that resides in a database server to facilitate storage, querying and manipulation of both spatial and relational data. The communication between the Jboss application server and the DBMS is established by using the appropriate JDBC drivers (c.f. [51]) provided by the DBMS vendor. In order to minimize the unavoidable communication cost between the PLOT server and the database server, we decided to implement part of the needed functionality that is undertaken by the DBMS, as PL/SQL procedures and functions that are stored within the DBMS and are remotely executed by the methods that reside in the PLOT server.

On the client–side, PLOT supports two GUIs: a Java client GUI and a web–based GUI. The Java client uses the Java Swing API (see [52]) and was implemented on the Java SE 6.0 Development Kit, which provides advanced features for the handling of the web services that reside on the application server. On the other hand, the web–based client was implemented by using JSP and uses the same middleware components as the PLOT server to consume the web services.

Finally, for the generation of the graphs that visualize the experimental results of the privacy methodologies, as well as for visualizing the computed regions of anonymity in the tested privacy approaches, the PLOT clients use the gnuplot software, which is publicly available at `http://www.gnuplot.info/`.

23.5 A Use Case of PLOT

To demonstrate some of the functionality that is offered by PLOT, in this section we present a use case scenario by applying the network based algorithm of [79] (main privacy algorithm) along with the unlinking strategy of [48] (secondary privacy algorithm). For unlinking, we considered $k = 3$ path crossings in the mix zone (see Section 17.4.2). In what follows, we present our system configuration, the used datasets and the attained results.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

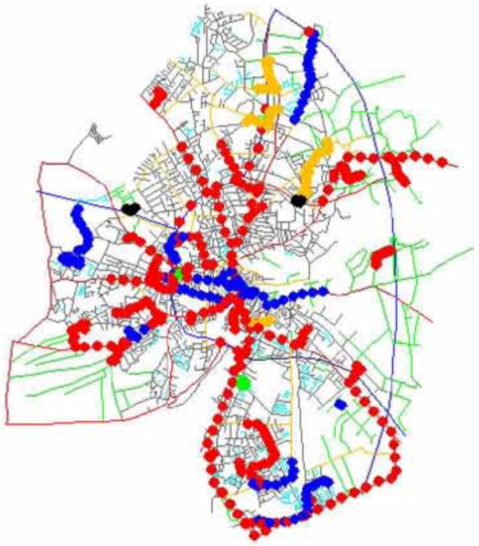


Figure 23.3: Moving objects in the road network of the Oldenburg city.

23.5.1 System Configuration

Our system configuration for PLOT consists of a high performance cluster and a server that hosts respectively the database server and the PLOT server, as well as a PC that runs the PLOT web client. The database server resides on a Dell PowerEdge 6650, equipped with four Intel Xeon 2.2 Ghz processors, each having 2GB of dedicated RAM. Since in our experiments we make use of the network–based model of movement, the DBMS of choice was Oracle Enterprise Edition 10g [2], which supports all the required functionality for the storage and the manipulation of network topologies [24]. The PLOT server resides on a Windows XP 3.2 Ghz Intel Pentium D that is equipped with 4GB of main memory.

23.5.2 Synthetically Generated Datasets

To produce the datasets for our experiments, we used the data input tab of the web client interface with the option of "objects movement generation". This leads to the execution of Brikhoff's generator of moving objects [30] with which we generated two synthetic datasets based on the road network of the Oldenburg city. The datasets we generated for experimen-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Table 23.2: The characteristics of the two synthetic datasets.

Parameter	**Dataset 1**	**Dataset 2**
# objects	1,000	2,000
# classes	2	6
max time	2,000	2,500
report prob.	0.9	0.5

tation experience different characteristics in terms of the number of moving objects and their classes, the maximum time of movement, and the probability of an object to report its position at each time. Table 23.2 summarizes their properties. The number of classes denotes the different types of moving objects (e.g., cars, trailers, motorbikes) that are considered in the generated data. Each object is assigned to a unique class that defines its maximum allowable speed in the network. Furthermore, the time of generation, the duration of existence and the time of disappearance differ for the various objects. Figure 23.3 presents a set of moving objects from the second dataset that we generated. The different colors in the graph correspond to objects that belong to different classes.

After producing the datasets, we used the data preprocessing tools that are offered by PLOT to edit the movement data. This is necessary to enable the identification of unsafe routes in moving objects that are generated through Brinkhoff's data generator. The reason is as follows: in Brinkhoff's generator, each generated object is automatically routed through the links of the network to move from its origin to its final destination. The use of a routing algorithm for the generation of movement has as an effect that no subtrajectory of an object is repeated and thereof no frequent routes exist. To solve this issue, for each moving object, we randomly selected a set of consecutive location updates from its movement and produced a random number of new trajectories of this object that contained only these location updates. This way, we achieved to artificially generate frequent routes for the moving objects, which subsequently lead to the existence of safe and unsafe routes in their movement. The resulting datasets were used for experimentation.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

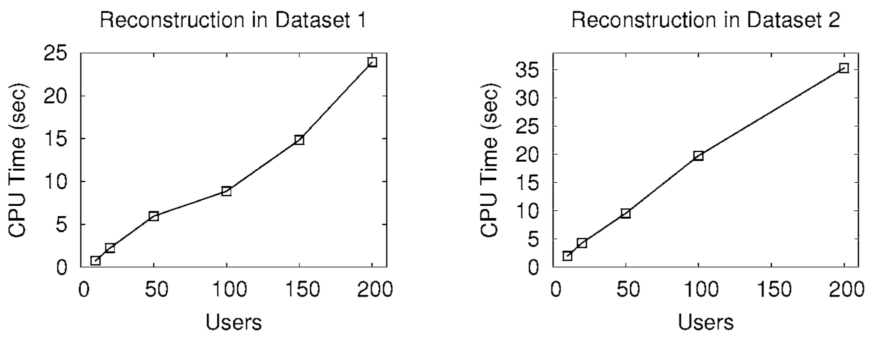


Figure 23.4: Cost of users movement reconstruction.

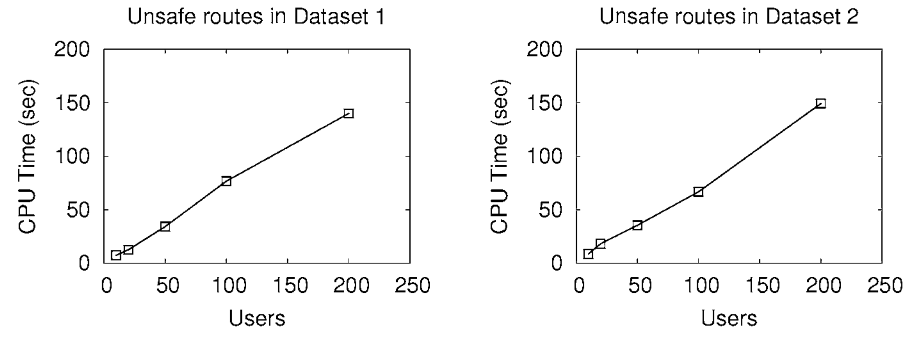


Figure 23.5: Computational cost for the unsafe routes.

23.5.3 Experimental Results

By using PLOT, we conducted three sets of experiments. In all our experiments, the reported results were obtained through the visualization options offered by the toolbox. In what follows, we present the conducted experiments.

In the first set of experiments, we computed the runtime that was required by PLOT to reconstruct the movement of the users in both datasets. Figure 23.4 presents the results we obtained, where the movement was reconstructed for different sets of randomly selected users from each dataset. In this figure, the reported times correspond to the complete reconstruction of the movement for the selected users.

The second set of experiments is presented in Figure 23.5. It involves the identification of the unsafe routes in the two datasets. Prior to computing the unsafe routes, we

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

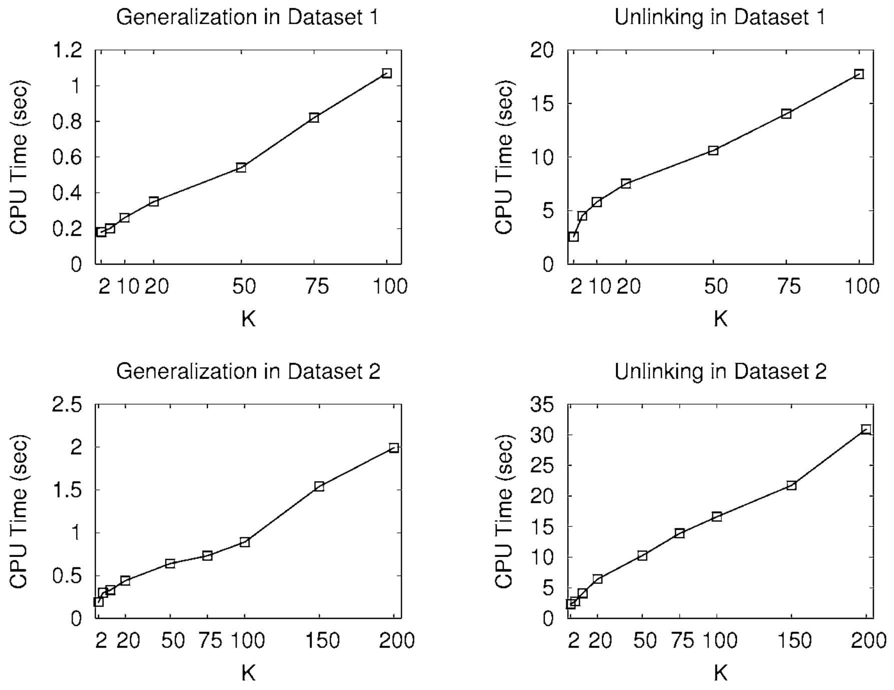
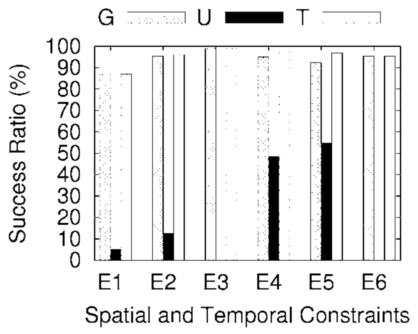


Figure 23.6: Generalization and unlinking cost for various $\mathcal{K}$.

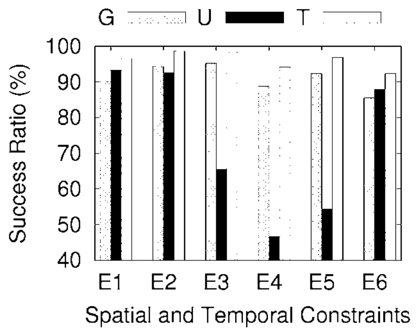
reconstructed and segmented the movement for all the users in the system. Then, we identified the unsafe routes for a number of users in each dataset, ranging from 2 to 200. For the purposes of our experiment, we considered a route to be frequent if it appears at least 3 times in the trajectories of the user.

The third set of experiments involves the provision of location privacy by the employed privacy algorithms. We experimented with different values of $\mathcal{K}$ in anonymity for the two datasets. Figure 23.6 shows the cost of generalization and unlinking for the various values of $\mathcal{K}$ that we considered for the provision of location privacy in the two datasets. In Figure 23.7 we present the success ratio of the generalization (denoted as "G") and the unlinking (denoted as "U") strategies in both datasets. As a secondary privacy algorithm, the unlinking approach is employed only if the generalization algorithm fails to protect the privacy of the requester. As one can observe in Figure 23.7 the total success ratio (denoted as "T") remains above 85% in all tested cases, involving different spatial and temporal

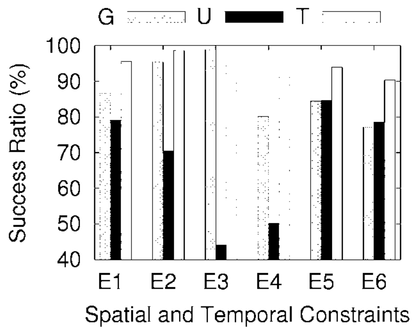
Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



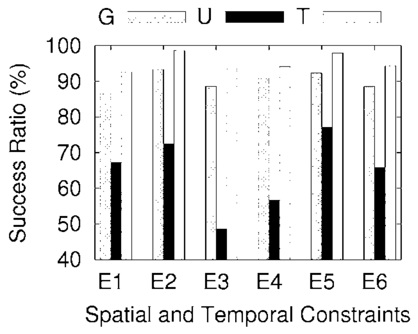
(a) Dataset 1, $\mathcal{K} = 50$



(b) Dataset 1, $\mathcal{K} = 100$



(c) Dataset 2, $\mathcal{K} = 50$



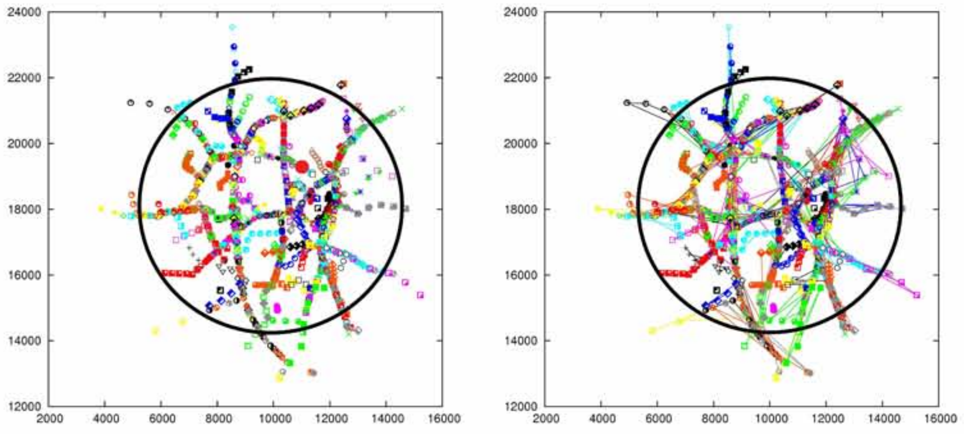
(d) Dataset 2, $\mathcal{K} = 100$

Figure 23.7: The success ratio in the two datasets.

Table 23.3: The considered spatiotemporal constraints.

Dataset, $\mathcal{K}$	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**
Dataset 1, $\mathcal{K} = 50$	1 tu, 5500au	3 tu, 5500au	5 tu, 5500au	1 tu, 6000au	3 tu, 6000au	5 tu, 6000au
Dataset 1, $\mathcal{K} = 100$	1 tu, 8500au	3 tu, 8500au	5 tu, 8500au	1 tu, 9000au	3 tu, 9000au	5 tu, 9000au
Dataset 2, $\mathcal{K} = 50$	1 tu, 4500au	3 tu, 4500au	5 tu, 4500au	1 tu, 5000au	3 tu, 5000au	5 tu, 5000au
Dataset 2, $\mathcal{K} = 100$	1 tu, 7500au	3 tu, 7500au	5 tu, 7500au	1 tu, 8000au	3 tu, 8000au	5 tu, 8000au

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136



(a) Prior reconstruction.   (b) After reconstruction.

Figure 23.8: An example of a 200−anonymity region.

constraints imposed on the privacy algorithm (see Table 23.3). Finally, in Figure 23.8 we present one 200−anonymity region that was generated by applying the generalization algorithm on Dataset 2. The red mark in this region (shown in the figure on the left) denotes the location of the actual requester at the time of request. As one can notice, graphs like Figure 23.8 provide the end user with insight regarding the capabilities of specific privacy approaches to effectively protect the privacy of the requesters of LBSs. In Figure 23.8(a) we present the region of anonymity where the users are depicted as a set of collected location updates, whereas Figure 23.8(b) presents the same region where the users are depicted based on their corresponding user trajectories.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 24

Summary and Future Work

This chapter summarizes the third (last) part of this dissertation. In this part, we focused our attention on methodologies that have been proposed for the offering of privacy in location based services. In Chapter 16, we introduced the problem and shed light on the existing privacy challenges in this area of research. Moreover, we presented the two dominant privacy models, which collect the state–of–the–art methodologies for the offering of privacy in LBSs. Then, in Chapter 17, we surveyed the different lines in which research has progressed in this area, paying special attention to the methodologies that have been proposed so far for the offering of $\mathcal{K}$–anonymity in LBSs. Following that, in Chapter 18, we briefly highlighted our contributions in this research area, which involve (i) two approaches for historical $\mathcal{K}$–anonymity, (ii) two approaches for trajectory $\mathcal{K}$–anonymity, and (iii) an open–ended toolbox for the integration of centralized privacy approaches in LBSs under a unified framework. The details of our contributions were presented in Chapters 19–23.

Recently, the scientific interest regarding cryptographic approaches for Private Information Retrieval (PIR) has revived. In PIR, a set of cryptographic protocols enable a user to privately retrieve information from a database, without leaving traces that will allow the database server to identify which particular information was retrieved. The PIR framework can be adopted to offer privacy in LBSs. However, it is proven that computational PIR is prohibitively costly to be applied in practice. Khoshgozaran and Shahabi [104] present

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

an initial PIR approach which employs one–dimensional transformation and encryption to protect requests for LBSs. In [57] the authors propose a methodology that significantly reduces the cost of PIR to an extent that it becomes reasonable. Moreover, as part of this work, the authors demonstrate that the computations that are required for the offering of privacy in LBSs through PIR, can be easily parallelized. Given the computational power of modern multiprocessor systems, we believe that this direction of research has to offer a significant push to this area of research in the coming years.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Chapter 25

Contributions of this Dissertation

In this dissertation, we investigated a set of methodologies for the preservation of privacy in different contexts, data domains, and application scenarios. In the first part of the dissertation, we focused our attention on transactional data that are mined for association rules. We considered that some of the extracted association rules from the original database are sensitive and proposed a set of methodologies that generate a sanitized version of the database where this knowledge is properly protected. Our proposed hiding algorithms guarantee optimality in the hiding solution by introducing the least amount of side–effects (if any) and by minimally distorting the original data to facilitate sensitive knowledge hiding. The detailed contributions of this part were provided in Chapter 4.

In the second part of this dissertation, we extended the applicability area of association rule hiding by applying similar techniques for the hiding of various types of temporally and spatially annotated data. In this part of the thesis, we proposed a novel privacy aware trajectory query engine that enables a set of untrusted end users to query trajectory data which are stored in a database. The proposed query engine guarantees that the answers that are returned to the end users meet specific privacy standards, as well as it effectively blocks certain types of attacks regarding user identification and/or tracking.

In the third part of the dissertation, we went even further by investigating trajectory hiding in a dynamic, real–time environment. In this part, we contributed along two princi-

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

pal lines of research: historical $\mathcal{K}$–anonymity and trajectory $\mathcal{K}$–anonymity, by proposing two privacy methodologies in each direction. All our contributed methodologies protect not only the location of request but the whole trajectory of the requester from the time of request until the provision of the requested service. We proposed privacy methodologies both for the handling of unconstrained and network–constrained user movement. Furthermore, we designed algorithms for the identification of the routes of each user where his/her privacy is in danger, based on his/her historical movement in the system. Finally, we contributed PLOT, the first open–ended toolbox for the offering of privacy in location based services. Chapter 18 presents in detail our contributions in this last part of the dissertation.

In what follows, we summarize the key contributions of our research:

- We proposed a new line of research in association rule hiding that consists of non–heuristic methodologies for the offering of exact knowledge hiding in transactional data. The methodologies that had been proposed in the past were based on heuristics to perform the sanitization of the data, and as a result there were many cases in which they failed to identify optimal hiding solutions. Our proposed algorithms formulate the hiding process as a Constraints Satisfaction Problem, where the holding of the constraints guarantees the lack of side–effects in the hiding solution. Among the potentially many different solutions of the formulated CSP, we select the one that minimizes the distortion of the original database, thus guarantees optimal hiding.
- We proposed a novel privacy aware trajectory tracking query engine that provides on–site, restricted access to in–house data to facilitate privacy aware data publishing. The methodologies that had been proposed in the past were based solely on count and/or sum queries in statistical databases, since no other information was made available to the inquirer. Our proposed engine supports a large variety of queries, involving both trajectory and non–trajectory data. Furthermore, it provides all the necessary mechanisms to secure the database against disclosure of confidential information, as well as it offers strict guarantees about what can be observed by untrusted third parties.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Specifically, in the case of trajectory–based queries, the computed answer as part of a user query is augmented (if necessary) with fake trajectory data such that the end user is unable to identify a specific user in the database with a probability that is above a minimum safety threshold. Moreover, the proposed engine can effectively block attacks that aim at the tracking of users based on their trajectories in the database.

- We proposed a set of methodologies for the offering of trajectory privacy in location based services both in a networkless and in a network aware topology of user movement. This was the first time that trajectory privacy in location based services was examined in a network confined environment, where movement is constrained to a series of allowable routes in the network. Furthermore, we proposed a set of methodologies that use the historical movement data of the users, as collected by the system, in order to automatically derive movement patterns (routes) that signify the locations and times where the privacy of the user is under threat. The knowledge of these patterns enables the system to take a series of beneficial decisions with respect to the privacy that is offered to the user, such that the identity of the user is adequately protected when he/she requests location based services.

- We proposed PLOT, the first open–ended toolbox that integrates centralized approaches for location privacy in location based services, in a common framework. PLOT offers the essential functionality for the implementation of novel location and trajectory privacy approaches, while serving as a testbed for the comparison of different location privacy algorithms. PLOT supports both networkless and network aware solutions to location privacy, as well as it implements strategies for the identification of the frequent movement patterns of the users in the system. Finally, it offers an extended set of visualization tools that provide insight on the workings of the implemented solutions and facilitates the qualitative and quantitative evaluation of their behavior.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Bibliography

[1] National institutes of health: Final nih statement on sharing research data, 2003. NOT–OD–03–032.

[2] Oracle 10g Release 2, 2003. Available at `http://www.oracle.com/technology/software/products/database/oracle10g/`.

[3] Oracle spatial 10g rel. 2 user's guide and reference, 2004. Available at `http://download.oracle.com/docs/pdf/B14255_01.pdf`.

[4] Oracle Database JDBC Developer's guide and reference 10g release 2, 2004. Available at `http://download-uk.oracle.com/docs/cd/B19306_01/java.102/b14355/`.

[5] IBM DB2 release 9, 2006. Available at `http://www-306.ibm.com/software/data/db2/9/`.

[6] Microsoft SQL Server 2008, 2008. Available at `http://www.microsoft.com/sqlserver/2008/en/us/default.aspx`.

[7] O. Abul, , F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 376–385, 2008.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[8] O. Abul, M. Atzori, F. Bonchi, and F. Giannotti. Hiding sensitive trajectory patterns. In *Proceedings of the 7th IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 693–698, 2007.

[9] O. Abul, M. Atzori, F. Bonchi, and F. Giannotti. Hiding sequences. In *Proceedings of the 23rd International Conference on Data Engineering Workshops (ICDEW)*, pages 147–156, 2007.

[10] N. R. Adam and J. C. Worthmann. Security–control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[11] C. C. Aggarwal and P. S. Yu. *Privacy Preserving Data Mining: Models and Algorithms*. Springer–Verlag, 2008.

[12] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[13] R. Agrawal and J. C. Shafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(1):962–969, 1996.

[14] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, pages 487–499, 1994.

[15] R. Agrawal and R. Srikant. Privacy preserving data mining. *SIGMOD Record*, 29(2):439–450, 2000.

[16] P. E. Agre. Transport informatics and the new landscape of privacy issues. *Computer Professionals for Social Responsibility (CPSR) Newsletter*, 13(3), 1995.

[17] A. Amiri. Dare to share: Protecting sensitive knowledge with data sanitization. *Decision Support Systems*, 43(1):181–191, 2007.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[18] N. Andrienko, G. Andrienko, N. Pelekis, and S. Spaccapietra. *Basic Concepts of Movement Data*, chapter 1, pages 15–38. Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer Berlin Heidelberg, 2008.

[19] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. Location privacy protection through obfuscation–based techniques. In *Proceedings of the 21st IFIP Conference on Data and Applications Security (DBSEC)*, pages 47–60, 2007.

[20] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)*, pages 45–52, 1999.

[21] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal. GPS: Location–tracking technology. *IEEE Computer*, 35(4):92–94, 2002.

[22] R. Bayardo. Efficiently mining long patterns from databases. *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, 1998.

[23] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*–tree: An efficient and robust access method for points and rectangles. *ACM SIGMOD Record*, 19(2):322–331, 1990.

[24] E. Beinat, A. Godfrind, and R. V. Kothuri. *Pro Oracle Spatial*. Apress, Inc., 2004.

[25] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[26] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location–based personal identification. In *Proceedings of the Second VLDB Workshop on Secure Data Management (SDM)*, pages 185–199, 2005.

[27] F. Bonchi, Y. Saygin, V. S. Verykios, M. Atzori, A. Gkoulalas-Divanis, S. V. Kaya, and E. Savas. *Privacy in Spatiotemporal Data Mining*, chapter 11, pages 297–333.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer Berlin Heidelberg, 2008.

[28] D. Boneh and M. K. Franklin. Identity–based encryption from the weil pairing. In *Proceedings of the 21st Annual International Cryptology Conference (CRYPTO)*, pages 213–229, 2001.

[29] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map–matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, pages 853–864, 2005.

[30] T. Brinkhoff. A framework for generating network–based moving objects. *Geoinformatica*, 6(2):153–180, 2002.

[31] H. Cao, N. Mamoulis, and D. W. Cheung. Mining frequent spatio–temporal sequential patterns. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 82–89, 2005.

[32] H. Cao, N. Mamoulis, and D. W. Cheung. Discovery of collocation episodes in spatiotemporal data. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*, pages 823–827, 2006.

[33] E. Cerami. *Web Services Essentials.* O'Reilly & Associates, Inc., 2002.

[34] L. Chang and I. S. Moskowitz. Parsimonious downgrading and decision trees applied to the inference problem. In *Proceedings of the 1998 Workshop on New Security Paradigms (NSPW)*, pages 82–89, 1998.

[35] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[36] X. Chen, M. Orlowska, and X. Li. A new framework of privacy preserving data sharing. In *Proceedings of the 4th IEEE International Workshop on Privacy and Security Aspects of Data Mining*, pages 47–56, 2004.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[37] R. Cheng. A survey on location privacy. Technical report, Department of Computer Science, Purdue University, May 2003.

[38] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET)*, pages 393–412, 2006.

[39] C. Y. Chow and M. F. Mokbel. Enabling private continuous queries for revealed user locations. In *Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 258–275, 2007.

[40] C. Y. Chow, M. F. Mokbel, and X. Liu. A peer–to–peer spatial cloaking algorithm for anonymous location based services. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (GIS)*, pages 171–178, 2006.

[41] R. Clarke. Person location and person tracking — technologies, risks and policy implications. *Information Technology and People*, 14(2):206–231, 2001.

[42] C. Clifton, M. Kantarcioglou, X. Lin, and M. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explorations*, 4(2):28–34, 2002.

[43] C. W. Clifton and D. Marks. Security and privacy implications of data mining. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 15–19, Feb. 1996.

[44] J. C. da Silva and M. Klusch. Inference on distributed data clustering. In *Proceedings of the 4th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 610–619, 2005.

[45] M. L. Damiani and E. Bertino. Access control and privacy in location–aware services for mobile organizations. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM)*, pages 11–21, 2006.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[46] E. Dasseni, V. S. Verykios, A. K. Elmagarmid, and E. Bertino. Hiding association rules by using confidence and support. In *Proceedings of the 4th International Workshop on Information Hiding*, pages 369–383, 2001.

[47] J. E. Dobson and P. F. Fisher. Geoslavery. *IEEE Technology and Society Magazine*, 22(1):47–52, 2003.

[48] D. Douras, A. Gkoulalas-Divanis, and V. S. Verykios. A $\mathcal{K}$–anonymity model for location based services. Technical report, Computer & Communication Enginnering Department, University of Thessaly, 2008.

[49] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proceedings of the 3rd International Conference on Pervasive Computing (PERV)*, pages 152–170, 2005.

[50] C. Farkas and S. Jajodia. The inference problem: A survey. *SIGKDD Explorations*, 4(2):6–11, 2002.

[51] M. Fisher, J. Ellis, and J. C. Bruce. *JDBC API Tutorial and Reference*. Pearson Education, 2003.

[52] A. Fowler. A swing architecture overview. Available at `http://java.sun.com/products/jfc/tsc/articles/architecture/`.

[53] F. S. G. Gottlob, N. Leone. A comparison of structural csp decomposition methods. *Artificial Intelligence*, 124(2):243–282, 2000.

[54] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 620–629, 2005.

[55] B. Gedik and L. Liu. Protecting location privacy with personalized $\mathcal{K}$–anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[56] C. Gentry and A. Silverberg. Hierarchical ID–based cryptography. In *Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, pages 548–566, 2002.

[57] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: Anonymizers are not necessary. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 121–132, 2008.

[58] G. Ghinita, P. Kalnis, and S. Skiadopoulos. MobiHide: A mobilea peer–to–peer system for anonymous location–based queries. In *Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 221–238, 2007.

[59] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVÉ: Anonymous location–based queries in distributed mobile systems. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pages 371–380, 2007.

[60] F. Giannotti, M. Nanni, and D. Pedreschi. Efficient mining of temporally annotated sequences. In *Proceedings of the 6th SIAM International Conference on Data Mining (SDM)*, pages 346–357, 2006.

[61] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 330–339, 2007.

[62] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy: Geographic Knowledge Discovery*. Springer Publishing Company, Incorporated, 2008.

[63] G. Gidófalvi and T. B. Pedersen. Mining long, sharable patterns in trajectories of moving objects. In *Proceedings of the 3rd Workshop on Spatio–Temporal Database Management (STDBM)*, pages 49–58, 2006.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[64] A. Gkoulalas-Divanis and V. S. Verykios. An integer programming approach for frequent itemset hiding. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 748–757, 2006.

[65] A. Gkoulalas-Divanis and V. S. Verykios. A hybrid approach to frequent itemset hiding. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 297–304, 2007.

[66] A. Gkoulalas-Divanis and V. S. Verykios. Concealing the position of individuals in location based services. In *Proceedings of the 20th National Conference on Corporation Research and Touristic Progress (EEEE)*, pages 973–984, 2008.

[67] A. Gkoulalas-Divanis and V. S. Verykios. Exact knowledge hiding in transactional databases. *International Journal on Artificial Intelligence Tools*, 2008. Accepted — to be published.

[68] A. Gkoulalas-Divanis and V. S. Verykios. Exact knowledge hiding through database extension. *IEEE Transactions on Knowledge and Data Engineering*, 2008. Accepted — appears online in preprint form.

[69] A. Gkoulalas-Divanis and V. S. Verykios. A free terrain model for trajectory $\mathcal{K}$–anonymity. In *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA)*, pages 49–56, 2008.

[70] A. Gkoulalas-Divanis and V. S. Verykios. Hiding sensitive knowledge without side effects. *Knowledge and Information Systems*, 2008. Accepted — to be published.

[71] A. Gkoulalas-Divanis and V. S. Verykios. *HESTIA: Historically–Enabled Spatio–Temporal Information Anonymity*, chapter 11, pages 1–28. Data Mining. Nova Science Publishers, Inc., 2008.

[72] A. Gkoulalas-Divanis and V. S. Verykios. A parallelization framework for exact knowledge hiding in transactional databases. In *Proceedings of the 23rd International Information Security Conference (SEC)*, pages 349–363, 2008.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[73] A. Gkoulalas-Divanis and V. S. Verykios. A privacy–aware trajectory tracking query engine. *SIGKDD Explorations*, 10(1):40–49, 2008.

[74] A. Gkoulalas-Divanis and V. S. Verykios. *Privacy in Trajectory Data*, chapter 11, pages 199–212. Social Implications of Data Mining and Information Privacy: Interdisciplinary Frameworks and Solutions. Information Science Reference, 2008.

[75] A. Gkoulalas-Divanis and V. S. Verykios. *Privacy Issues in Spatiotemporal Data Mining*. Privacy Aware Knowledge Discovery: Novel Applications and New Techniques (Data Mining and Knowledge Discovery Book Series). Chapman & Hall / CRC Press, 2009. Accepted — to be published.

[76] A. Gkoulalas-Divanis and V. S. Verykios. *Privacy Preserving Data Mining: How far can we go?*, pages 1–21. Handbook of Research on Data Mining in Public and Private Sectors: Organizational and Governmental Applications. IGI Global, 2009. Accepted — Currently in press.

[77] A. Gkoulalas-Divanis, V. S. Verykios, and P. Bozanis. A network aware privacy model for online requests in trajectory data. *Data and Knowledge Engineering*, pages 1–24, 2008. Accepted — appears online in preprint form.

[78] A. Gkoulalas-Divanis, V. S. Verykios, and D. Eleftheriou. PLOT: Privacy in location–based services: an open–ended toolbox. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware (MDM)*, 2009.

[79] A. Gkoulalas-Divanis, V. S. Verykios, and M. F. Mokbel. Identifying unsafe routes for network–based trajectory privacy. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2009.

[80] B. Goethals. The fimi repository, 2003. Available at `http://fimi.cs.helsinki.fi/`.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[81] G. Grahne and J. Zhu. Fast algorithms for frequent itemset mining using FP–trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1347–1362, 2005.

[82] M. Gruteser and D. Grunwald. Anonymous usage of location–based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, pages 31–42, 2003.

[83] C. Gueret, C. Prins, and M. Sevaux. *Applications of Optimization with Xpress–MP*. Dash Optimization Ltd., 2002.

[84] D. Gusfield. Efficient methods for multiple sequence alignment with guaranteed error bounds. *Bulletin of Mathematical Biology*, 55(1):141–154, 1993.

[85] A. Guttman. R–trees: A dynamic index structure for spatial searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

[86] E. H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 277–288, 2007.

[87] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[88] C. Hauser and M. Kabatnik. Towards privacy support in a global location service. In *Proceedings of the IFIP Workshop on IP and ATM Traffic Management (WATM/EUNICE)*, pages 81–89, 2001.

[89] U. Hengartner and P. Steenkiste. Protecting access to people location information. In *Proceedings of the 1st International Conference on Security in Pervasive Computing (SPC)*, pages 25–38, 2003.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[90] U. Hengartner and P. Steenkiste. Implementing access control to people location information. In *Proceedings of the 9th ACM Symposium on Access Control Models and Technologies (SACMAT)*, pages 11–20, 2004.

[91] U. Hengartner and P. Steenkiste. Exploiting hierarchical identity–based encryption for access control to pervasive computing information. In *Proceedings of the 1st IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SECURECOMM)*, pages 384–393, 2005.

[92] L. J. Hoffman. *Modern Methods for Computer Security and Privacy*. Prentice–Hall, Englewood Cliffs, New Jersey, USA, 1977.

[93] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, pages 194–205, 2005.

[94] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty–aware path cloaking. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, pages 161–171, 2007.

[95] I. ILOG. *CPLEX 9.0 User's Manual.* Mountain View, CA, Oct. 2005. Available at `http://www.ilog.com/`.

[96] A. Inan and Y. Saygin. Privacy preserving spatiotemporal clustering on horizontally partitioned data. In *Proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery Engineering (DAWAK)*, pages 459–468, 2006.

[97] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A new privacy–preserving distributed *k*–clustering algorithm. In *Proceedings of the 2006 SIAM International Conference on Data Mining (SDM)*, 2006.

[98] C. S. Jensen, H. Lahrmann, S. Pakalnis, and J. Runge. The infati data. Technical report, Time Center TR–79, Aalborg University, Denmark, 2004. Available at `http://www.cs.aau.dk/TimeCenter/`.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[99] S. Jha, L. Kruger, and P. McDaniel. Privacy preserving clustering. In *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS)*, pages 397–417, 2005.

[100] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location–based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[101] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20(1):359–392, 1998.

[102] A. Katsarou, A. Gkoulalas-Divanis, and V. S. Verykios. Reconstruction–based classification rule hiding through controlled data modification. In *Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI)*, 2009.

[103] D. Kesdogan, P. Reichl, and K. Junghärtchen. Distributed temporary pseudonyms: A new approach for protecting location information in mobile communication networks. In *Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS)*, pages 295–312, 1998.

[104] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 239–257, 2007.

[105] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location based services. In *Proceedings of the International Conference on Pervasive Services (ICPS)*, pages 88–97, 2005.

[106] R. Kohavi, C. Brodley, B. Frasca, L. Mason, and Z. Zheng. KDD–Cup 2000 organizers' report: Peeling the onion. *SIGKDD Explorations*, 2(2):86–98, 2000. http://www.ecn.purdue.edu/KDDCUP.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[107] D. Konidala, C. Y. Yeun, and K. Kim. A secure and privacy enhanced protocol for location based services in ubiquitous society. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 931–936, 2004.

[108] R. K. V. Kothuri, S. Ravada, and D. Abugov. Quadtree and R–tree indexes in oracle spatial: a comparison using GIS data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 546–557, 2002.

[109] P. Kumaraguru, L. Cranor, J. Lobo, and S. Calo. A survey of privacy policy languages. In *Proceedings of the 3rd Symposium on Usable Privacy and Security (SOUPS)*, 2007.

[110] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, 2006.

[111] G. Lee, C. Y. Chang, and A. L. P. Chen. Hiding sensitive patterns in association rules mining. In *Proceedings of the 28th International Computer Software and Applications Conference (COMPSAC)*, pages 424–429, 2004.

[112] N. Li, T. Li, and S. Venkatasubramanian. t–closeness: Privacy beyond $\mathcal{K}$–anonymity and l–diversity. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[113] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):36–54, 2000.

[114] D. Luenberger. *Introduction to Linear and Non–Linear Programming*. Addison–Wesley, 1973.

[115] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l–diversity: Privacy beyond $\mathcal{K}$–anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):1–52, 2007.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[116] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung. Mining, indexing, and querying historical spatio–temporal data. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Dining*, pages 236–245, 2004.

[117] H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledgediscovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[118] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 181–192, 1994.

[119] Y. Manolopoulos, Y. Theodoridis, and V. J. Tsotras. *Advanced database indexing*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[120] F. Marchal, J. Hackney, and K. W. Axhausen. Efficient map matching of large global positioning system data sets: Tests on speed–monitoring experiment in zurich. *Transportation Research Record*, 1935:93–100, 2006.

[121] T. Marrs and S. Davis. *JBoss at Work: A Practical Guide*. O'Reilly Media, Inc., 2005.

[122] S. Menon, S. Sarkar, and S. Mukherjee. Maximizing accuracy of shared databases when concealing sensitive patterns. *Information Systems Research*, 16(3):256–270, 2005.

[123] N. Meratnia and R. A. de By. Trajectory representation in location–based services: Problems and solution. In *Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops*, 2003.

[124] T. Mielikainen. On inverse frequent set mining. In *Proceedings of the 2nd Workshop on Privacy Preserving Data Mining*, pages 18–23, 2003.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[125] M. F. Mokbel and C. Y. Chow. Challenges in preserving location privacy in peer–to–peer environments. In *Proceedings of the 7th International Conference on Web–Age Information Management Workshops (WAIMW)*, pages 1–8, 2006.

[126] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 763–774, 2006.

[127] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space–filling curve. *IEEE Transanctions on Knowledge and Data Engineering*, 13(1):124–141, 2001.

[128] G. V. Moustakides and V. S. Verykios. A max–min approach for hiding frequent itemsets. In *Workshops Proceedings of the 6th IEEE International Conference on Data Mining (ICDM)*, pages 502–506, 2006.

[129] J. Natwichai, X. Li, and M. Orlowska. Hiding classification rules for data sharing with privacy preservation. In *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DAWAK)*, pages 468–477, 2005.

[130] J. Natwichai, X. Li, and M. Orlowska. A reconstruction–based algorithm for classiciation rules hiding. In *Proceedings of the 17th Australasian Database Conference (ADC)*, pages 49–58, 2006.

[131] M. E. Nergiz, M. Atzori, and Y. Saygin. Towards trajectory anonymization: A generalization–based approach. In *Proceedings of the ACM GIS Workshop on Security and Privacy in GIS and LBS*, pages 1–10, 2008.

[132] D. E. O'Leary. Knowledge discovery as a threat to database security. In *Proceedings of the 1st International Conference on Knowledge Discovery in Databases (KDD)*, pages 507–516, 1991.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[133] S. R. M. Oliveira and O. R. Zaïane. Privacy preserving frequent itemset mining. In *Proceedings of the 2002 IEEE International Conference on Privacy, Security and Data Mining (CRPITS)*, pages 43–54, 2002.

[134] S. R. M. Oliveira and O. R. Zaïane. Protecting sensitive knowledge by data sanitization. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 211–218, 2003.

[135] S. R. M. Oliveira and O. R. Zaïane. Achieving privacy preservation when sharing data for clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM)*, 2004.

[136] S. R. M. Oliveira and O. R. Zaïane. Privacy–preserving clustering by object similarity–based representation and dimensionality reduction transformation. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM)*, pages 21–30, 2004.

[137] Paypal. Secure online payments. Available at `https://www.paypal.com`.

[138] N. Pelekis and Y. Theodoridis. Boosting location–based services with a moving object database engine. In *Proceedings of the 5th ACM International Workshop on Data engineering for Wireless and Mobile Access*, pages 3–10, 2006.

[139] N. Pelekis, Y. Theodoridis, S. Vosinakis, and T. Panayiotopoulos. Hermes — a framework for location–based data management. In *Proceedings of the 11th International Conference on Extending Database Technology (EDBT)*, pages 1130–1134, 2006.

[140] R. G. Pensa, A. Monreale, F. Pinelli, and D. Pedreschi. Pattern–preserving $\mathcal{K}$–anonymization of sequences and its application to mobility data mining. In *Proceedings of the International Workshop on Privacy in Location Based Applications*, pages 1–17, 2008.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[141] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity — a proposal for terminology. In *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9, 2000.

[142] B. Pinkas. Cryptographic techniques for privacy preserving data mining. *SIGKDD Explorations*, 4(2):12–19, 2002.

[143] E. Pontikakis, Y. Theodoridis, A. Tsitsonis, L. Chang, and V. S. Verykios. A quantitative and qualitative analysis of blocking in association rule hiding. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society (WPES)*, pages 29–30, 2004.

[144] E. D. Pontikakis, A. A. Tsitsonis, and V. S. Verykios. An experimental study of distortion–based techniques for association rule hiding. In *Proceedings of the 18th Conference on Database Security (DBSEC)*, pages 325–339, 2004.

[145] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location–support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, 2000.

[146] S. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB)*, 2002.

[147] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice–Hall, 2nd edition, 2003.

[148] Y. Saad. SPARSKIT: A basic toolkit for sparse matrix computations (version 2). Technical report, University of Minnesota, Minneapolis, MN, 1994. http://www-users.cs.umn.edu/ saad/software/SPARSKIT/sparskit.html.

[149] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[150] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: $\mathcal{K}$–anonymity and its enforcement through generalization and suppresion. In *Proceedings of the IEEE Symposium on Research in Security and Privacy (SRSP)*, pages 384–393, 1998.

[151] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison–Wesley Longman Publishing Co., Inc., 1990.

[152] Y. Saygin, V. S. Verykios, and C. W. Clifton. Using unknowns to prevent discovery of association rules. *ACM SIGMOD Record*, 30(4):45–54, 2001.

[153] Y. Saygin, V. S. Verykios, and A. K. Elmagarmid. Privacy preserving association rule mining. In *Proceedings of the 2002 International Workshop on Research Issues in Data Engineering: Engineering E–Commerce/E–Business Systems (RIDE)*, pages 151–163, 2002.

[154] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai. *Trends in Spatial Data Mining*, chapter 3, pages 357–380. Data Mining: Next Generation Challenges and Future Directions. AAAI/MIT Press, 2004.

[155] D. A. Simovici and C. Djeraba. *Mathematical Tools for Data Mining: Set Theory, Partial Orders, Combinatorics*. Springer Publishing Company, Incorporated, 2008.

[156] A. Smailagic and D. Kogan. Location sensing and privacy in a context–aware computing environment. *IEEE Wireless Communications*, 9(5):10–17, Oct 2002.

[157] E. Snekkenes. Concepts for personal location privacy policies. In *Proceedings of the 3rd ACM Conference on Electronic Commerce (EC)*, pages 48–57, 2001.

[158] M. Song and S. Rajasekaran. Finding frequent itemsets by transaction mapping. In *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC)*, pages 488–492, 2005.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[159] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer–to–peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networks*, 11(1):17–32, 2003.

[160] X. Sun and P. S. Yu. A border–based approach for hiding sensitive frequent itemsets. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 426–433, 2005.

[161] X. Sun and P. S. Yu. Hiding sensitive frequent itemsets by a border–based approach. *Computing science and engineering*, 1(1):74–94, 2007.

[162] L. Sweeney. $\mathcal{K}$–anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge Based Systems*, 10(5):557–570, 2002.

[163] P. N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison–Wesley, 2005.

[164] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proceedings of the The 9th International Conference on Mobile Data Management (MDM)*, pages 65–72, 2008.

[165] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems*, 29(3):463–507, 2004.

[166] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Transactions on Database Systems*, 29(3):463–507, 2004.

[167] T. M. Truta and B. Vinay. Privacy protection: p–sensitive $\mathcal{K}$–anonymity property. In *Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW)*, pages 94–103, 2006.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[168] J. Vaidya, C. W. Clifton, and Y. M. Zhu. *Privacy Preserving Data Mining*. Springer–Verlag, 2006.

[169] F. Verhein and S. Chawla. Geometrically inspired itemset mining. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*, pages 655–666, 2006.

[170] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State–of–the–art in privacy preserving data mining. *ACM SIGMOD Record*, 33(1):50–57, 2004.

[171] V. S. Verykios, M. L. Damiani, and A. Gkoulalas-Divanis. *Privacy and Security in Spatiotemporal Data and Trajectories*, chapter 8, pages 213–240. Mobility, Data Mining and Privacy: Geographic Knowledge Discovery. Springer Berlin Heidelberg, 2008.

[172] V. S. Verykios, A. K. Emagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association rule hiding. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):434–447, 2004.

[173] V. S. Verykios and A. Gkoulalas-Divanis. *A Survey of Association Rule Hiding Methods for Privacy*, chapter 11, pages 267–289. Privacy Preserving Data Mining: Models and Algorithms. Springer Berlin Heidelberg, 2008.

[174] V. S. Verykios, A. Gkoulalas-Divanis, Y. Theodoridis, and N. Pelekis. *Privacy Preservation in Trajectories of Moving Objects*, chapter 15, pages 1–21. Privacy Preservation on Computer and Communication Technologies: Technical and Legal Issues. Papasotiriou, 2009. Accepted — to be published.

[175] K. Wang, B. C. M. Fung, and P. S. Yu. Template–based privacy preservation in classification problems. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM)*, pages 466–473, 2005.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[176] S. L. Wang and A. Jafari. Using unknowns for hiding sensitive predictive association rules. In *Proceedings of the 2005 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 223–228, 2005.

[177] R. C. W. Wong, J. Li, A. W. C. Fu, and K. Wang. $(\alpha, \mathcal{K})$–anonymity: An enhanced $\mathcal{K}$–anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD Conference on KDD*, pages 754–759, 2006.

[178] Y. H. Wu, C. M. Chiang, and A. L. P. Chen. Hiding sensitive association rules with limited side effects. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):29–42, 2007.

[179] Z. Xiao, X. Meng, and J. Xu. Quality–aware privacy protection for location based services. In *Proceedings of the 14th International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 434–446, 2007.

[180] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location–based services. In *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM)*, pages 547–555, 2008.

[181] H. Yin and O. Wolfson. A weight–based map matching method in moving objects databases. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 437–446, 2004.

[182] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):673–685, 1998.

[183] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: A review. *Autonomous Agents and Multi–Agent Systems*, 3(2):185–207, 2000.

[184] B. Yu and S. H. Kim. Interpolating and using most likely trajectories in moving–objects databases. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA)*, pages 718–727, 2006.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

[185] P. Zacharouli, A. Gkoulalas-Divanis, and V. S. Verykios. A $\mathcal{K}$–anonymity model for spatiotemporal data. In *Proceedings of the IEEE Workshop on Spatio–Temporal Data Mining (STDM)*, pages 555–564, 2007.

[186] O. R. Zaïane, M. El-Hajj, and P. Lu. Fast parallel association rule mining without candidacy generation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 665–668, 2001.

[187] M. J. Zaki. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3):372–390, 2000.

[188] Q. Zhao and S. S. Bhowmick. Sequential pattern mining: A survey. Technical report, Center for Advanced Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136

Vita

Aris Gkoulalas-Divanis received the Diploma degree in Computer Science from the University of Ioannina, Greece, in 2003 and the MS degree from the University of Minnesota in 2005. Since 2005, he is a PhD student in the Department of Computer and Communication Engineering at the University of Thessaly, in Volos, Greece. Aris Gkoulalas-Divanis has served as a research assistant in both the University of Minnesota, in the Department of Computer Science and Engineering (2003–2005) and in the University of Manchester, in the School of Informatics (2006). Since 2001 he has participated in several IST/EU projects including Childcare, Citation, and GeoPKDD. His research interests are in the fields of privacy in location based services and privacy preserving data mining, with focus on methodologies for association rule hiding and trajectory hiding. He is a student member of IEEE, ACM and SIAM, as well as a regular reviewer for ACM Computing Reviews and Elsevier's Data & Knowledge Engineering. He currently serves in the Editorial board of ACM Crossroads as an associate editor.

Permanent Address: 23 Kautatzoglou Str., GR–11144, Athens, Greece.

This dissertation was typeset with LaTeX 2_ε[1] by Aris Gkoulalas-Divanis.

[1]LaTeX 2_ε extends LaTeX, a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin.

Institutional Repository - Library & Information Centre - University of Thessaly
01/06/2024 18:14:17 EEST - 18.221.200.136