ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΧΡΗΣΗ ATTRIBUTE – BASED ENCRYPTION ΣΕ ΠΕΡΙΒΑΛΛΟΝΤΑ IOT

ΚΑΤΑΛΑΓΑΡΙΑΝΟΥ ΝΙΚΗ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΗ

Κοζύρη Μαρία

Επίκουρη Καθηγήτρια

ΣΥΝΕΠΙΒΛΕΠΩΝ

Σπαθούλας Γεώργιος

Μέλος Ε.ΔΙ.Π.

Λαμία 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΧΡΗΣΗ ATTRIBUTE – BASED ENCRYPTION ΣΕ ΠΕΡΙΒΑΛΛΟΝΤΑ IOT

ΚΑΤΑΛΑΓΑΡΙΑΝΟΥ ΝΙΚΗ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΗ

Κοζύρη Μαρία

Επίκουρη Καθηγήτρια

ΣΥΝΕΠΙΒΛΕΠΩΝ

Σπαθούλας Γεώργιος

Μέλος Ε.ΔΙ.Π.

Λαμία 2021

UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

ATTRIBUTE – BASED ENCRYPTION IN IOT ENVIRONMENTS

KATALAGARIANOU NIKI

FINAL THESIS

ADVISOR

Koziri Maria

Assistant Professor

CO ADVISOR

Spathoulas Georgios

Member of Laboratory Teaching Staff

Lamia 2021

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί ***χωρίς να τα περικλείω σε εισαγωγικά*** *και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. Δέχομαι ότι η αυτολεξεί ***παράθεση χωρίς εισαγωγικά****, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 23/02/2021

Η Δηλούσα
Καταλαγαριανού Νίκη

(1) «Οποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον κ. Σπαθούλα Γεώργιο, Μέλος Ε.ΔΙ.Π. του Τμήματος Πληροφορικής με Εφαρμογές στη Βιοϊατρική της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας για όλη τη βοήθειά του κατά τη διάρκεια της εκπόνησης της πτυχιακής μου εργασίας καθώς και για τις νέες γνώσεις που μου έδωσε την ευκαιρία να αποκτήσω με την υλοποίηση της συγκεκριμένης έρευνας.

Επίσης, θα ήθελα να ευχαριστήσω την κ. Κοζύρη, Επίκουρη Καθηγήτρια του Τμήματος Πληροφορικής και Τηλεπικοινωνιών της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας για τη συμμετοχή της στην τριμελή επιτροπή της πτυχιακής εργασίας και την προθυμία της να βοηθήσει, καθώς και τον κ. Κολομβάτσο Κωνσταντίνο Επίκουρο Καθηγητή του Τμήματος Πληροφορικής και Τηλεπικοινωνιών της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας για τη συμμετοχή του στην τριμελή επιτροπή της πτυχιακής.

Ευχαριστώ επίσης, το Τμήμα Πληροφορικής και Τηλεπικοινωνιών της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας για τα γνωστικά εφόδια που μου παρείχε και το άψογο ακαδημαϊκό κλίμα κατά τη διάρκεια των τεσσάρων ετών των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για τη στήριξή τους καθόλη τη διάρκεια των σπουδών μου και κατά τη συγγραφή της πτυχιακής μου.

Περίληψη

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην μελέτη της μεθόδου Attribute – Based Encryption και στην υλοποίησή της στο Internet of Things και πιο εξειδικευμένα στα έξυπνα οχήματα που αποτελούν μέρος του Internet of Vehicles. Σκοπός της χρήσης της συγκεκριμένης μεθόδου είναι η επίτευξη καλύτερης προστασίας των δεδομένων των IoT συσκευών στον σύγχρονο κυβερνοχώρο που κρύβει πλήθος κινδύνων. Για το λόγο αυτό, πραγματοποιείται η δημιουργία μέσω κώδικα ενός συστήματος που εφαρμόζει την τεχνική Attribute – Based Encryption και αποτελείται από μια οντότητα που κρυπτογραφεί μηνύματα, ένα όχημα που τα αποκρυπτογραφεί, μια κεντρική οντότητα που παράγει κλειδιά και μια οντότητα που παράγει τυχαία αλφαριθμητικά. Η εκτέλεση του συστήματος προσομοιωμένο σε μια σύγχρονη πόλη, παρέχει πλήθος αποτελεσμάτων τα οποία δείχνουν ενθαρρυντικά. Αποδεικνύεται πως οι τιμές των επικοινωνιών και των δεδομένων, ενώ είναι υψηλές δεν δημιουργούν προβλήματα στο σύστημα. Συνεπώς, η εφαρμογή του στην πραγματικότητα είναι εφικτή με κάποιες πιθανές προσαρμογές ώστε να ανταπεξέλθει καλύτερα στο μεγάλο πλήθος των συσκευών που υπάρχουν.

Λέξεις – Κλειδιά

Attribute – Based encryption, Internet of Things, ασφάλεια, κρυπτογραφία, κεντρική οντότητα, όχημα, δεδομένα

Abstract

The present dissertation focuses on the study of the Attribute – Based Encryption method and its implementation on the Internet of Things and more specifically on the smart vehicles, part of the Internet of Vehicles. The purpose of the use of this method is to achieve better data protection of IoT devices in the modern cyberspace that hides a number of risks. For this reason, a system that uses the Attribute-Based Encryption technique is created through code and consists of an entity that encrypts messages, a vehicle that decrypts them, a central authority that generates keys and an entity that generates random strings. Running the system simulated in a modern city, provides a number of results that look encouraging. It turns out that the values of communications and data, while high, do not create problems in the system. Therefore, its implementation is actually possible, probably with some adjustments to better cope with the large number of devices that exist.

Πίνακας Περιεχομένων

Περίληψη I

Λέξεις – Κλειδιά I

Abstract II

1. Εισαγωγή 1

2. Ασφάλεια και Ιδιωτικότητα στο Διαδίκτυο των Πραγμάτων 3

2.1. Διαδίκτυο των Πραγμάτων 3

2.1.1. Ιστορική Αναδρομή 3

2.1.2. Εφαρμογές 4

2.2. Προσωπικά Δεδομένα και Πληροφορίες 6

2.3. Απειλές 8

2.4. Ασφάλεια 10

2.5. Κρυπτογραφία 11

2.5.1. Συμμετρική Κρυπτογράφηση 12

2.5.2. Κρυπτογράφηση δημόσιου κλειδιού 13

2.6. Κρυπτογράφηση βασισμένη σε χαρακτηριστικά 15

3. Βιβλιογραφική Ανασκόπηση 16

3.1. Έρευνες στο Attribute – Based Encryption 16

3.2. Attribute – Based Encryption στο IoT 18

3.3. Revocation 19

4. Υλοποίηση 21

4.1. Κρυπτογραφική Μέθοδος 21

4.1.1. Ciphertext – Policy Attribute – Based Encryption 21

4.1.2. Revocation 25

4.2. Τεχνολογίες 25

4.2.1. OpenABE 26

4.2.2. Ubuntu 18.04 29

4.2.3. Python Programming 30

4.2.5. Socket Programming 31

4.3. Υλοποίηση 32

4.3.1. Shared Storage 34

4.3.2. Beacon 34

4.3.3. Messenger 36

4.3.4. Central Authority 38

4.3.5. Client 42

5. Αποτελέσματα 46

5.1. Dataset 46

5.2. Λειτουργία Συστήματος 49

5.3. Αποτελέσματα 52

6. Συμπεράσματα 58

6. Βιβλιογραφία 59

Πίνακας Εικόνων

Εικόνα 1: Λειτουργία Συστήματος 33

Εικόνα 2: Modules του beacon 34

Εικόνα 3: Socket και δημιουργία πολλών clients 35

Εικόνα 4: Δημιουργία τυχαίου string 35

Εικόνα 5: Χρόνος αποστολής string 36

Εικόνα 6: Modules του messenger 36

Εικόνα 7: Path του αρχείου 37

Εικόνα 8: Δημιουργία socket και αίτημα για string 37

Εικόνα 9: Υπολογισμός T και T' 37

Εικόνα 10: Κρυπτογράφηση 38

Εικόνα 11: Αντιγραφή κρυπτογραφημένου αρχείου στο shared storage 38

Εικόνα 12: Modules του central authority και setup της βιβλιοθήκης 39

Εικόνα 13: Ορισμός socket και δημιουργία πολλών clients 39

Εικόνα 14: Δημιουργία τυχαίων attributes με χρόνο 40

Εικόνα 15: Παραγωγή κλειδιών 41

Εικόνα 16: Αντιγραφή του master public key στο shared storage..... 41
Εικόνα 17: Modules του client..... 42
Εικόνα 18: Δημιουργία socket, αίτηση για string και λήψη του..... 43
Εικόνα 19: Thread..... 43
Εικόνα 20: Αποστολή δεδομένων στον server..... 43
Εικόνα 21: Λήψη κλειδιών από τον server..... 44
Εικόνα 22: Έλεγχος ύπαρξης αρχείου για αποκρυπτογράφηση και χρήση του..... 45
Εικόνα 23: Αποκρυπτογράφηση..... 45
Εικόνα 24: Συνολικό αρχείο με ταξί..... 46
Εικόνα 25: Αρχείο ενός ταξί..... 47
Εικόνα 26: Ανάγνωση αρχείου..... 48
Εικόνα 27: Έλεγχος απόστασης..... 49
Εικόνα 28: Έλεγχος απόστασης..... 49
Εικόνα 29: Ανάγνωση argument..... 50
Εικόνα 30: Δεδομένα αρχείου key_management..... 50
Εικόνα 31: Δεδομένα αρχείου msg_management..... 51

Πίνακας Διαγραμμάτων

Διάγραμμα 1: Επικοινωνίες 1ου οχήματος
Διάγραμμα 2: Δεδομένα 1ου οχήματος..... 52
Διάγραμμα 3: Επικοινωνίες 2ου οχήματος
Διάγραμμα 4: Δεδομένα 2ου οχήματος..... 52
Διάγραμμα 5: Επικοινωνίες 3ου οχήματος
Διάγραμμα 6: Δεδομένα 3ου οχήματος..... 52
Διάγραμμα 7: Επικοινωνίες 4ου οχήματος
Διάγραμμα 8: Δεδομένα 4ου οχήματος..... 53
Διάγραμμα 9: Επικοινωνίες 5ου οχήματος
Διάγραμμα 10: Δεδομένα 5ου οχήματος..... 53
Διάγραμμα 11: Επικοινωνίες στο 1ο όχημα
Διάγραμμα 12: Δεδομένα στο 1ο όχημα..... 54
Διάγραμμα 13: Επικοινωνίες στο 2ο όχημα
Διάγραμμα 14: Δεδομένα στο 2ο όχημα..... 55
Διάγραμμα 15: Επικοινωνίες στο 3ο όχημα
Διάγραμμα 16: Δεδομένα στο 3ο όχημα..... 55
Διάγραμμα 17: Επικοινωνίες στο 4ο όχημα
Διάγραμμα 18: Δεδομένα στο 4ο όχημα..... 55
Διάγραμμα 19: Επικοινωνίες στο 5ο όχημα
Διάγραμμα 20: Δεδομένα στο 5ο όχημα..... 56

1. Εισαγωγή

Η ανθρωπότητα βρίσκεται σε μια εποχή που χαρακτηρίζεται από τη ραγδαία εξέλιξη της τεχνολογίας και του διαδικτύου, η χρήση των οποίων έχει ως στόχο τη βελτιστοποίηση των δραστηριοτήτων, είτε ατομικών, είτε σε επίπεδο επιχειρήσεων και οργανισμών. Νέα συστήματα αναπτύσσονται συνεχώς, ενώ καθημερινά ο άνθρωπος χρησιμοποιεί μια πληθώρα συσκευών και εφαρμογών, κινητών και μη, για τηδιεκπεραίωση οποιασδήποτε δραστηριότητας. Παράλληλα, το Διαδίκτυο των Πραγμάτων ή αλλιώς γνωστό ως Internet of Things εξελίσσεται ταχύτατα δίνοντας τη δυνατότητα να συνδέεται πλέον οποιαδήποτε συσκευή με τον παγκόσμιο ιστό και διευκολύνοντας απλές καθημερινές εργασίες. Συγχρόνως, οι επιχειρήσεις κάθε τομέα συμβαδίζουν με τις νέες τεχνολογικές αλλαγές για να εξυπηρετήσουν τους σκοπούς τους. Έτσι λοιπόν, κάθε δραστηριότητα από την πιο απλή μέχρι την πιο περίπλοκη διεκπεραιώνεται πλέον με ηλεκτρονικά μέσα.

Η χρήση όμως όλων των παραπάνω ηλεκτρονικών μέσων συνεπάγεται και δεδομένα, συνήθως προσωπικά, για να αξιοποιηθεί στο έπακρον ότι παρέχουν. Ο άνθρωπος δίνει τα προσωπικά του στοιχεία για να έχει πρόσβαση σε σελίδες, εφαρμογές, για να κάνει τις αγορές του, να επικοινωνήσει με φίλους ή ακόμα και για να επωφεληθεί από υπηρεσίες που του προσφέρουν εταιρείες και επιχειρήσεις. Από τη μεριά τους οι επιχειρήσεις, συλλέγουν ποικίλα δεδομένα, ανάλογα με τις υπηρεσίες που παρέχουν. Αυτή η σωρεία δεδομένων αποθηκεύεται σε κάθε λογής μέσο, σε βάσεις δεδομένων, εφαρμογές, κινητά τηλέφωνα, υπολογιστές, ηλεκτρονικά αρχεία και στο σύννεφο γνωστό ως cloud.

Όπως είναι φυσικό επόμενο όμως, η χρήση του διαδικτύου και όλων των ηλεκτρονικών μέσων και συστημάτων εγκυμονεί απειλές που εξελίσσονται παράλληλα. Οι πληροφορίες και τα δεδομένα μπορούν με πολλούς τρόπους να διαρρεύσουν και πολλές φορές να πέσουν σε λάθος χέρια, συνήθως με καταστροφικές συνέπειες είτε σε ατομικό, είτε σε επιχειρησιακό επίπεδο.

Για να αποφευχθούν οι απειλές αυτές, είναι απαραίτητο να βρεθούν τρόποι προστασίας των δεδομένων και των πληροφοριών. Με αυτό ασχολείται ο τομέας τη Ασφάλειας στον οποίο αναπτύσσεται ένα πλήθος από τεχνικές, συστήματα και μεθοδολογίες ώστε τα δεδομένα να μη διαρρεύσουν, αλλά και στην περίπτωση που αυτό συμβεί, να έχουν όσο το δυνατόν μικρότερο αντίκτυπο στα πιθανά θύματα.

Ένα από τα κομμάτια που αποτελούν αναπόσπαστο τμήμα της ασφάλειας και το οποίο αποτελεί σημαντικό όπλο στην προστασία των δεδομένων, είναι η Κρυπτογραφία. Με το πέρασμα του χρόνου, έχει αναπτυχθεί πληθώρα τεχνικών και μεθόδων κρυπτογράφησης, με κάθε νέα τεχνική να είναι πιο περίπλοκη και δύσκολη στο να σπάσει και να αποκαλυφθεί η πληροφορία που κρύβει. Μια από τις μεθόδους αυτές, είναι και η κρυπτογράφηση που βασίζεται σε χαρακτηριστικά, γνωστή ως attribute – based encryption η οποία στοχεύει σε μια πιο αποτελεσματική κρυπτογράφηση στοχεύοντας στο χαρακτηριστικό της μοναδικότητας ενός κλειδιού. Αποτελεί μια μέθοδο που είναι σχετικά νέα και βρίσκεται ακόμα κατά κύριο λόγο στο στάδιο της μελέτης. Ωστόσο είναι πολλά υποσχόμενη στην ασφάλεια των δεδομένων.

Για το λόγο αυτό, αποτελεί το αντικείμενο μελέτης της παρούσας πτυχιακής εργασίας. Αρχικά, γίνεται υλοποίηση της συγκεκριμένης τεχνικής με τη χρήση βιβλιοθηκών κρυπτογράφησης, ενώ στη συνέχεια, πραγματοποιείται μια προσομοίωση στον πραγματικό κόσμο ώστε να διαπιστωθεί πως θα μπορούσε να εφαρμοστεί στην πράξη. Στο τέλος, διατυπώνονται τα αποτελέσματα και οι παρατηρήσεις που εξάγονται από την έρευνα ώστε να διαπιστωθεί αν είναι τελικά αποτελεσματική και επαρκής.

2. Ασφάλεια και Ιδιωτικότητα στο Διαδίκτυο των Πραγμάτων

2.1. Διαδίκτυο των Πραγμάτων

Διαδίκτυο των Πραγμάτων ή Internet of Things είναι το σύνολο των καθημερινών συσκευών, «πραγμάτων» που χρησιμοποιούν αισθητήρες και συνδέονται μεταξύ τους και με το διαδίκτυο (Παπαζώης, 2019). Ο όρος χρησιμοποιήθηκε για πρώτη φορά το 1999 από τον Kevin Ashton στην προσπάθεια του να εξηγήσει τη σημασία των RFID (Συστήματα Ταυτοποίησης με Ραδιοσυχνότητες), οπότε και περιέγραψε ένα σύστημα στο οποίο πλήθος συσκευών συνδέονται στο διαδίκτυο (Παπασταθοπούλου, 2017).

Οι συσκευές που σχηματίζουν το Internet of Things ονομάζονται έξυπνες συσκευές και έχουν ως χαρακτηριστικό ότι συνδέονται ασύρματα στο διαδίκτυο και διαχειρίζονται πλήθος πληροφοριών και δεδομένων (Tzafestas, 2018). Οι βασικές τεχνολογίες που χρησιμοποιούν είναι οι διευθύνσεις IP, το αναγνωριστικό RFID και το NFC (Παπαζώης, 2019).

Αναπόσπαστο κομμάτι του IoT είναι και το υπολογιστικό νέφος (cloud computing) το οποίο αποτελεί μια οντότητα στο διαδίκτυο που χρησιμοποιείται για την αποθήκευση δεδομένων και είναι προσβάσιμη από όλους (“Cloud Computing”, 2021). Συνεπώς, οι συσκευές του IoT χρησιμοποιούν το cloud για να διατηρήσουν τα δεδομένα που συλλέγουν. Παράλληλα, χρησιμοποιείται και το fog computing το οποίο έχει ως στόχο την πραγματοποίηση των υπολογισμών κατανεμημένα, με αποτέλεσμα στο cloud να αποθηκεύονται μόνο όσα στοιχεία και δεδομένα είναι απολύτως απαραίτητα για την εύρυθμη λειτουργία του IoT (“Fog Computing”, 2020).

2.1.1. Ιστορική Αναδρομή

Παρόλο που ο όρος Internet of Things χρησιμοποιήθηκε για πρώτη φορά το 1999, το ίδιο το Internet of Things είχε ξεκινήσει σαν ιδέα πολύ νωρίτερα. Σε κάποια σημεία μάλιστα η ιστορία του ταυτίζεται με αυτήν του ίδιου του διαδικτύου καθώς οι δυο έννοιες είναι άρρηκτα συνδεδεμένες.

Το 1950, η IBM θέλοντας να δώσει ένα μοναδικό αναγνωριστικό σε κάθε συσκευή της ώστε να ταυτοποιείται άμεσα, δημιούργησε το barcode. Αργότερα, το 1967, ο Hubert Upton δημιούργησε μια συσκευή η οποία έμοιαζε με γυαλιά οράσεως και επέτρεπε στα άτομα με ειδικές ανάγκες να μπορούν να διαβάζουν τα χείλη του συνομιλητή τους όταν τα φοράνε. Το 1982, με την ανακάλυψη του TCP/IP πρωτοκόλλου δημιουργήθηκε πλέον το σημερινό διαδίκτυο αφού πλέον τα δίκτυα μπορούσαν να συνδεθούν μεταξύ τους (Κλαδίσιος, 2019) ενώ το 1989 ο Tim Berners Lee μίλησε για πρώτη φορά για τον Παγκόσμιο Ιστό (www). Μια δεκαετία αργότερα, το 1998 ο Mark Weiser δημιούργησε ένα συντριβάνι στο οποίο η αυξομείωση της ροής του μιμούνταν τον όγκο και τις τιμές στο χρηματιστήριο και το 1999 ο Neil Gershenfelt μίλησε για πράματα που συνδέονται στο δίκτυο και σκέφτονται. Το 1999 σε συνέχιση ενός έργου του Kevin Ashton η Auto-ID Labs άρχισε να χρησιμοποιεί ευρέως την ασύρματη ανάγνωση και εγγραφή δεδομένων (RFID). Το 2000 ανακοινώθηκε το πρώτο έξυπνο ψυγείο, ενώ σιγά σιγά άρχισε να χρησιμοποιείται σε δημοσιεύσεις, άρθρα και συνέδρια ο όρος «Internet of Things». Το 2008 η Ευρωπαϊκή Ένωση στο πρώτο Ευρωπαϊκό Συνέδριο Διαδικτύου των Πραγμάτων, αναγνώρισε επίσημα τον όρο. Από εκεί και μετά, ο τομέας αυτός γνωρίζει συνεχή ανάπτυξη μέρα με τη μέρα με όλο και περισσότερα άτομα στον κόσμο να χρησιμοποιούν έξυπνες συσκευές (Internet of things (IoT) History, 2017, όπως παρατίθεται στο Tzafestas, 2018).

2.1.2. Εφαρμογές

Οι έξυπνες συσκευές βρίσκονται παντού. Μερικές από τις πιο χαρακτηριστικές έξυπνες συσκευές είναι αυτές που φοριούνται και ονομάζονται wearables. Τα wearables όπως και οι υπόλοιπες συσκευές IoT συλλέγουν πληροφορίες για το άτομο που τα φορά. Οι πληροφορίες αυτές, συνήθως μεταφέρονται σε κάποια εφαρμογή ή κάποιες συσκευές τα κρατούν στη μνήμη τους και στη συνέχεια τα μεταφέρουν. Τα wearables χωρίζονται σε τρεις κατηγορίες.

- Wearables που αφορούν την υγεία, τα οποία καταγράφουν τις ζωτικές ενδείξεις ενός ατόμου όπως παλμούς, πίεση, θερμοκρασία σώματος και άλλα και χρησιμοποιούνται συνήθως για την ιατρική παρακολούθηση ασθενών (Cicek, M., 2015, όπως παρατίθεται στο Tzafestas, 2018).

- Ηλεκτρονικά wearables, που μαζί με ζωτικές ενδείξεις καταγράφουν και θερμίδες, αριθμό βημάτων, αριθμό χιλιομέτρων που διανύθηκαν, εικόνα, ήχο και άλλα και χρησιμοποιούνται κυρίως για τη διασκέδαση των χρηστών. Μερικά ηλεκτρονικά wearables είναι τα έξυπνα γυαλιά και ρολόγια και οι ανιχνευτές δραστηριότητας (activity trackers) (Cicek, M., 2015, όπως παρατίθεται στο Tzafestas, 2018).
- Wearable υφάσματα, που έχουν ενσωματωμένη τεχνολογία και χρησιμοποιούνται από τη βιομηχανία της μόδας για λόγους αισθητικής. Ένα παράδειγμα, είναι τα ρούχα που αλλάζουν χρώμα βάσει της συναισθηματικής κατάστασης του ατόμου που τα φορά (Cicek, M., 2015, όπως παρατίθεται στο Tzafestas, 2018).

Φυσικά υπάρχει ένας μεγάλος αριθμός άλλων έξυπνων συσκευών. Σύμφωνα με την Gartner μέχρι το 2020 προβλέπονταν είκοσι δισεκατομμύρια συσκευές συνδεδεμένες με το διαδίκτυο. Με άλλα λόγια η αναλογία είναι τέσσερις συσκευές ανά έναν άνθρωπο, κάτι το οποίο είναι εντυπωσιακό (Hung, 2017). Επίσης, σύμφωνα με την IDC μέχρι το 2025 προβλέπεται ότι θα υπάρχουν 41.6 δισεκατομμύρια συσκευές συνδεδεμένες στο διαδίκτυο (Framingham, 2019).

Το Internet of Things λοιπόν, βρίσκει εφαρμογή σε κάθε τομέα της καθημερινής ζωής του ανθρώπου. Παρακάτω αναφέρονται μερικές σημαντικές εφαρμογές.

- Έξυπνο σπίτι: Οι ηλεκτρικές συσκευές του σπιτιού συνδέονται με το δίκτυο του σπιτιού. Υπάρχουν αισθητήρες υπεύθυνοι για τη θερμοκρασία, το φως, την υγρασία, τις ανάγκες του σπιτιού, το σύστημα καπνού, την ασφάλεια. Συλλέγουν δεδομένα, ελέγχουν τα παραπάνω χαρακτηριστικά και προσφέρουν άνεση και διευκολύνσεις στους ενοίκους που μπορούν να τα χειριστούν όλα μέσω του κινητού, του τάμπλετ ή του λάπτοπ τους (NIST, 2013, όπως παρατίθεται στο Παπαζώης, 2019).

- Έξυπνη μετακίνηση (Internet of Vehicles): Το όχημα περιέχει αισθητήρες που ελέγχουν τις ενδείξεις του αυτοκινήτου, την τοποθεσία, το πόσο κοντά βρίσκεται σε άλλο όχημα, την ύπαρξη ή όχι φωτός στο δρόμο και αισθητήρες που εντοπίζουν πιθανούς κινδύνους στο δρόμο. Έτσι παρέχει στον επιβάτη ασφαλή μετακίνηση κάτω από οποιεσδήποτε συνθήκες (Παπαζώης, 2019).

- Έξυπνη πόλη: Με τη χρήση αισθητήρων και έξυπνων συσκευών πολλές δραστηριότητες και υπηρεσίες αυτοματοποιούνται και βελτιστοποιούνται ενώ παράλληλα οι πολίτες διευκολύνονται στις δουλειές τους, με αποτέλεσμα να υπάρχει μεγαλύτερη ανάπτυξη. Μπορεί για παράδειγμα να γίνεται καλύτερα η διαχείριση της κυκλοφορίας, των έργων, των κτιρίων και του φωτισμού τωνδρόμων (Κλαδίσιος, 2019).

- Έξυπνη υγεία: Με τη χρήση του Internet of Things είναι δυνατό στα νοσοκομεία να υπάρχουν συσκευές για τη διαχείριση των παροχών στα δωμάτια των ασθενών καθώς και για τη διευκόλυνση των ίδιων των ασθενών. Παράλληλα, είναι δυνατή η παρακολούθηση της υγείας τους και εκτός του νοσοκομείου με χρήση συσκευών που φοριούνται, κάτι που είναι πολύ χρήσιμο σε ανθρώπους με δυσκολίες ή σε ηλικιωμένους που χρειάζονται βοήθεια. (Παπαζώης, 2019)

- Έξυπνη βιομηχανία: Σε έναν βιομηχανικό χώρο μπορεί με τη χρήση έξυπνων συσκευών και αισθητήρων να γίνεται η διαχείριση των συνθηκών του εργοστασίου, η καταγραφή της κατάστασης των συσκευών, η καταγραφή της παραγωγής και των εμπορευμάτων και άλλα. Με αυτό τον τρόπο, μειώνονται τα έξοδα και βελτιώνεται η παραγωγή και άρα τα κέρδη της επιχείρησης.

Άλλες επίσης σημαντικές εφαρμογές είναι η έξυπνη γεωργία και κτηνοτροφία, έξυπνες μετρήσεις και ενέργεια, καθώς και τα έξυπνα κτίρια.

2.2. Προσωπικά Δεδομένα και Πληροφορίες

Όπου και να κοιτάξει κάποιος στο διαδίκτυο και σε όλα τα ηλεκτρονικά μέσα υπάρχουν πληροφορίες και προσωπικά δεδομένα. Τα δεδομένα αυτά υπάρχουν σε μεγαλύτερο βαθμό στις συσκευές του Internet of Things. Σύμφωνα με τον Γενικό Κανονισμό Προστασίας Δεδομένων (ΓΚΠΔ ή GDPR) « Τα προσωπικά δεδομένα είναι κάθε πληροφορία που σχετίζεται με ένα ταυτοποιημένο ή αναγνωρίσιμο ζωντανό άτομο (υποκείμενο των δεδομένων). Ένα αναγνωρίσιμο φυσικό πρόσωπο είναι εκείνο που μπορεί να ταυτοποιηθεί, άμεσα ή έμμεσα, ιδίως με αναφορά σε ένα αναγνωριστικό όπως όνομα,

αριθμός αναγνώρισης, δεδομένα τοποθεσίας, ηλεκτρονικό αναγνωριστικό ή με έναν ή περισσότερους παράγοντες και συγκεκριμένα τη φυσική, φυσιολογική, γενετική, διανοητική, οικονομική, πολιτιστική ή κοινωνική ταυτότητα αυτού του φυσικού προσώπου» (Κανονισμός (ΕΕ) 2016/679).

Τα προσωπικά δεδομένα μπορεί να είναι είτε απλά, είτε ειδικής κατηγορίας (ευαίσθητα). Στα απλά ανήκουν δεδομένα όπως το ονοματεπώνυμο, το μέιλ, η διεύθυνση. Από την άλλη, στα ειδικής κατηγορίας δεδομένα ανήκουν δεδομένα που «μπορούν να αποκαλύψουν φυλετική ή εθνοτική καταγωγή, πολιτικές απόψεις, θρησκευτικές ή φιλοσοφικές πεποιθήσεις, ή συνδικαλιστική συμμετοχή και επεξεργασία γενετικών δεδομένων, βιομετρικά δεδομένα με σκοπό τη μοναδική αναγνώριση φυσικού προσώπου, δεδομένα σχετικά με την υγεία ή δεδομένα σχετικά με τη σεξουαλική ζωή ενός φυσικού προσώπου ή τον σεξουαλικό προσανατολισμό.» (Κανονισμός (ΕΕ) 2016/679).

Τα προσωπικά δεδομένα είναι πολύτιμα γιατί μπορούν να οδηγήσουν στην αναγνώριση ενός ατόμου και να προκαλέσουν σοβαρές επιπτώσεις. Επομένως, η προστασία των προσωπικών δεδομένων είναι πολύ σημαντική γεγονός που φαίνεται και από τους νόμους που έχουν θεσπιστεί σε διάφορα μέρη του κόσμου. Ενδεικτικά, ο πιο γνωστός είναι ο Γενικός Κανονισμός Προστασίας Δεδομένων ο οποίος ισχύει για τα κράτη – μέλη της Ευρωπαϊκής Ένωσης καθώς και για επιχειρήσεις και οργανισμούς που βρίσκονται εκτός της Ευρωπαϊκής Ένωσης αλλά επεξεργάζονται δεδομένα ατόμων που ανήκουν σε αυτήν. Επιπλέον, σε συμμόρφωση με τον κανονισμό κάθε κράτος – μέλος έχει ιδρύσει τη δική του αρχή, αντίστοιχη με την Αρχή Προστασίας Προσωπικών Δεδομένων που υπάρχει στην Ελλάδα.

Υπάρχουν τρεις βασικοί άξονες, ένα τρίπτυχο δηλαδή, το οποίο πρέπει πάντα να λαμβάνεται υπόψην και να τηρείται όσον αφορά τα προσωπικά δεδομένα. Ο πρώτος άξονας του τρίπτυχου είναι η Εμπιστευτικότητα, που είναι «η απαίτηση ιδιωτικές ή εμπιστευτικές πληροφορίες να μη διαρρέουν σε μη εξουσιοδοτημένα άτομα» (Nieles et al., 2017). Στη συνέχεια, ακολουθεί η Ακεραιότητα που είναι η απαίτηση τα δεδομένα να μην υποστούν οποιαδήποτε τροποποίηση και η Διαθεσιμότητα, δηλαδή η απαίτηση τα συστήματα και κατ’ επέκταση τα δεδομένα να είναι ανελλιπώς διαθέσιμα (Nieles et al., 2017).

Παράλληλα βέβαια με τα προσωπικά δεδομένα υπάρχουν πληροφορίες και δεδομένα τα οποία μπορεί να μην ταυτοποιούν κάποιο άτομο, αλλά είναι εξίσου σημαντικά

και χρήζουν προστασίας ώστε να μην βρεθούν σε λάθος χέρια. Οι πληροφορίες αυτές μπορεί να αφορούν επικοινωνίες ή ακόμα και σχέδια επιχειρήσεων ή τον τρόπο λειτουργίας των συστημάτων τους τα οποία σε καμία περίπτωση δεν πρέπει να διαρρεύσουν.

2.3. Απειλές

Με την μεταφορά της πλειοψηφίας των ανθρώπινων δραστηριοτήτων στον ηλεκτρονικό χώρο και το πλήθος πληροφοριών και δεδομένων που υπάρχουν, ήταν αναμενόμενη και η εμφάνιση απειλών οι οποίες εξελίσσονται παράλληλα και ελλοχεύουν σε κάθε γωνία. Απειλές οι οποίες αυξάνονται με την διάδοση του Internet of Things, γιατί οι συσκευές που το απαρτίζουν έχουν πρόσβαση σε εξαιρετικά μεγάλο αριθμό πληροφοριών. Έτσι σε συνδυασμό με το γεγονός ότι οι συσκευές είναι συνδεδεμένες στο διαδίκτυο, οι πληροφορίες διατρέχουν μεγάλο κίνδυνο.

Οι απειλές που μπορεί να προκύψουν μπορεί να είναι πολλές και να διαφέρουν στην επικινδυνότητα και τις επιπτώσεις που μπορεί να επιφέρουν. Υπάρχει περίπτωση να πρόκειται απλά για ένα ανθρώπινο λάθος που άφησε τα δεδομένα εκτεθειμένα ή οδήγησε σε διαρροή, απώλεια ή αλλοίωσή τους. Υπάρχουν όμως και άλλου είδους κίνδυνοι. Μία περίπτωση είναι η βιομηχανική κατασκοπία που αποτελεί συλλογή εταιρικών πληροφοριών χωρίς τη γνώση της προσβαλλόμενης επιχείρησης, με στόχο αυτά να αξιοποιηθούν ανταγωνιστικά (“Industrial Espionage”, 2020). Από την άλλη, μπορεί να πρόκειται για ακτιβιστές οι οποίοι αξιοποιούν διάφορες τεχνικές επιθέσεων για να κάνουν πιο γνωστά και να αντιμετωπίσουν διάφορα κοινωνικά, πολιτικά και άλλα ζητήματα (Merriam-Webster, n.d.). Μια ακόμα περίπτωση είναι και οι αγανακτισμένοι υπάλληλοι οι οποίοι είτε δεν είναι ικανοποιημένοι από τη δουλειά τους, είτε τους έχουν πρόσφατα απολύσει και θέλουν να κάνουν κάτι για να εκδικηθούν για αυτό, όπως το να διαρρεύσουν πληροφορίες για την επιχείρηση.

Ως επίθεση λοιπόν ή αλλιώς κυβερνοεπίθεση, νοείται οποιαδήποτε «προσπάθεια για καταστροφή, διακοπή ή απόκτηση μη εξουσιοδοτημένης πρόσβασης» σε ηλεκτρονικό εξοπλισμό σύμφωνα με το Εθνικό Κέντρο Κυβερνοασφάλειας της Αγγλίας (National Cyber Security Centre, 2020). Παράλληλα, επιτιθέμενος είναι οποιοδήποτε άτομο αποκτά μη εξουσιοδοτημένη πρόσβαση στις πληροφορίες και τα δεδομένα που στοχεύει. Τα άτομα αυτά μπορούν να αποκτήσουν πρόσβαση σε δεδομένα με ένα μεγάλο αριθμό τεχνικών και

τα εγκλήματα που διαπράττουν είναι πολλών ειδών. Το σύνολο των εγκλημάτων στον κυβερνοχώρο αποτελεί το κυβερνοέγκλημα.

Κυβερνοέγκλημα, είναι η παράνομη δραστηριότητα εναντίον πληροφοριακών συστημάτων, δικτύων, υπολογιστών και των δεδομένων και πληροφοριών που διαθέτουν και επεξεργάζονται (Παπανικολάου, 2009). Τα είδη κυβερνοεγκλήματος είναι πολλά και παρακάτω θα αναφερθούν μερικά από τα πιο διαδεδομένα.

- Κακόβουλη εισβολή (Hacking): Σύμφωνα με το λεξικό του Cambridge είναι «η δραστηριότητα της χρήσης υπολογιστή για πρόσβαση σε πληροφορίες που είναι αποθηκευμένες σε άλλο σύστημα υπολογιστή χωρίς άδεια ή η διάδοση ιών υπολογιστή (Cambridge Dictionary, n.d.).

- Ηλεκτρονικό ψάρεμα (Phishing): Είναι η «προσπάθεια απόκτησης ευαίσθητων πληροφοριών ή δεδομένων με δόλο.» (“Phishing”, 2021). Συνήθως ο επιτιθέμενος προσποιείται ότι είναι μια αξιόπιστη οντότητα, για παράδειγμα μια τράπεζα και ζητά κωδικούς πρόσβασης και άλλα προσωπικά στοιχεία με σκοπό την απόκτηση πρόσβασης στους λογαριασμούς του θύματος, ενώ συχνά μπορεί να το ξεγελάσει με το να το οδηγήσει σε μια φαινομενικά αλλά όχι πραγματικά ασφαλή και αξιόπιστη σελίδα.

- Κακόβουλο λογισμικό (Malware): Όπως αναφέρει η CISCO, «το κακόβουλο λογισμικό είναι παρεμβατικό λογισμικό που έχει σχεδιαστεί για να κάνει ζημιά και να καταστρέφει υπολογιστές και συστήματα υπολογιστών» (Cisco Systems, Inc.,2020). Συνήθως εισέρχεται στο δίκτυο και στον υπολογιστή και προκαλεί προβλήματα ανάλογα με το είδος του. Τα πιο γνωστά είδη κακόβουλου λογισμικού είναι ιός, worm, ιός Δούρειου Ίππου (Trojan), λογισμικό παρακολούθησης (spyware), λογισμικό διαφημίσεων (adware) και λογισμικό με απαίτηση λύτρων (ransomware).

- Ανεπιθύμητη αλληλογραφία (spam): Η αποστολή μεγάλου αριθμού μηνυμάτων ηλεκτρονικού ταχυδρομείου σε επίσης μεγάλο πλήθος παραληπτών με σκοπό την εκμαίευση πληροφοριών.

- Κλοπή Ταυτότητας (Identity Theft): Η χρήση των προσωπικών πληροφοριών ενός ατόμου και η ιδιοποίηση της ταυτότητάς του με σκοπό την διάπραξη εγκλημάτων (“Identity Theft”, 2020).

- Διαδικτυακή Πειρατεία: Η παράνομη διακίνηση έργων που καλύπτονται από πνευματικά δικαιώματα (Βογιατζής, 2010).

- Κοινωνική Μηχανική (Social Engineering): Η τεχνική χειραγώγησης και εκμετάλλευσης του ανθρώπου και των συναισθημάτων του με στόχο την απόκτηση προσωπικών πληροφοριών.

2.4. Ασφάλεια

Ο άνθρωπος όπως είναι φυσιολογικό επιθυμεί την εξασφάλιση της ιδιωτικότητάς του γι’ αυτό και χρειάζεται να προστατεύσει τα δεδομένα και τις πληροφορίες του. Ο ρόλος της ασφάλειας και ιδιαίτερα της κυβερνοασφάλειας είναι να τον προστατεύσει από όλες τις παραπάνω απειλές και τους κινδύνους με αποτελεσματικό τρόπο. Είναι πλέον αναπόσπαστο κομμάτι της τεχνολογίας και των ηλεκτρονικών μέσων. Συγκεκριμένα, όπως αναφέρει το Ευρωπαϊκό Ελεγκτικό Συνέδριο η κυβερνοασφάλεια «αποτελεί το σύνολο των διασφαλίσεων και μέτρων που υιοθετούνται για την προστασία των συστημάτων πληροφοριών και των χρηστών τους έναντι μη εξουσιοδοτημένης πρόσβασης, επιθέσεων και ζημίας, ώστε να εξασφαλίζονται η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα των δεδομένων» (Ευρωπαϊκό Ελεγκτικό Συνέδριο, 2019).

Για την επίτευξη της ασφάλειας μια επιχείρηση πρέπει να εφαρμόσει ποικίλα μέτρα προστασίας. Είναι σημαντικό να θεσπίσει διαδικασίες οι οποίες να διέπουν κάθε τομέα της και να προβλέπουν τις δράσεις που πρέπει να γίνουν στην περίπτωση μιας επικείμενης απειλής. Επιπλέον, πρέπει να οριστούν πολιτικές ασφάλειας πληροφοριών, οι οποίες να εξηγούν πως πρέπει να γίνεται η διαχείριση των συστημάτων που επεξεργάζονται πληροφορίες καθώς και των ίδιων των πληροφοριών ώστε να επιτευχθεί η μέγιστη προστασία τους. Είναι επίσης απαραίτητο να οριστούν ρόλοι ώστε να είναι ευδιάκριτα τα καθήκοντα και οι αρμοδιότητες κάθε ατόμου στην επιχείρηση, ενώ είναι χρήσιμη και η πραγματοποίηση δοκιμών παρείσδυσης και αποτίμησης κινδύνων και απειλών με σκοπό

να βρεθούν οι ευπάθειες και τα κενά ασφαλείας των συστημάτων. Τέλος, πρέπει να εφαρμοστούν τεχνικά μέτρα προστασίας. Μερικά από αυτά ενδεικτικά, μπορεί να είναι αυθεντικοποίηση χρηστών, διαχείριση προσβάσεων, κρυπτογραφία, μηχανισμοί ασφάλειας δικτύων, συστημάτων, εφαρμογών και πληροφοριών (Deloitte Central Mediterranean & ΣΕΒ, 2020).

2.5. Κρυπτογραφία

Όπως αναφέρθηκε παραπάνω, μια από τις τεχνικές που εφαρμόζονται για την προστασία και την ασφάλεια των συστημάτων και των πληροφοριών είναι η κρυπτογραφία.

Η κρυπτογραφία είναι η «μελέτη και εφαρμογή τεχνικών για ασφαλή επικοινωνία παρουσία τρίτων μερών» (“Cryptography”, 2021). Συγκεκριμένα, στην κρυπτογραφία χρησιμοποιούνται τεχνικές κρυπτογράφησης σε πληροφορίες που μπορεί να έχουν διάφορες μορφές. Στόχος είναι η απόκρυψη των πληροφοριών με τη μετατροπή τους σε μορφή η οποία δεν είναι αναγνώσιμη χωρίς τη γνώση των απαραίτητων στοιχείων για την αποκρυπτογράφησή της (National Cyber Security Centre, 2020). Έτσι, προσφέρει το πλεονέκτημα πως ακόμα και αν η πληροφορία βρεθεί σε λάθος χέρια, είναι ακατανόητη χωρίς τη γνώση ενός κλειδιού.

Η Κρυπτογραφία χρησιμοποιούνταν από τα αρχαία χρόνια στην αποστολή μηνυμάτων που έπρεπε να παραμείνουν μυστικά, κατά κύριο λόγο για στρατιωτικούς και πολιτικούς σκοπούς. Υπάρχουν αναφορές από αρχαίους ιστορικούς όπως τον Ηρόδοτο και τον Σουητώνιο που κάνει λόγο για τον κώδικα του Καίσαρα. Αργότερα, υπάρχουν αναφορές για μια μέθοδο κρυπτογράφησης με τη χρήση καθρέπτη από τον Leonardo Da Vinci, ενώ φτάνοντας στον προηγούμενο αιώνα ένα από τα πιο δημοφιλή κρυπτοσυστήματα είναι το Enigma, το οποίο χρησιμοποιούσαν οι Γερμανοί κατά το Β’ Παγκόσμιο πόλεμο για την κρυπτογράφηση των τηλεπικοινωνιών τους (Ζάχος et al., 2015). Φτάνοντας στο παρόν η κρυπτογραφία αποτελεί πλέον μια επιστήμη που εξελίσσεται συνεχώς για να καλύψει τις ολοένα αυξανόμενες ανάγκες για ασφάλεια συστημάτων και επικοινωνιών, με νέες μεθόδους κρυπτογράφησης να ανακαλύπτονται διαρκώς. Ενώ στην αρχή χρησιμοποιούνταν μέθοδοι με αναδιάταξη και αντικατάσταση πλέον αξιοποιούνται πιο περίπλοκες τεχνικές (Ζάχος et al., 2015). Παράλληλα, βρίσκει

εφαρμογή σε περισσότερους τομείς όπως επικοινωνίες, συστήματα ηλεκτρονικών πληρωμών και συναλλαγών, συσκευές που ανήκουν στο Διαδίκτυο των Πραγμάτων, βάσεις με ιατρικά και άλλα δεδομένα.

Οι τέσσερις κύριοι στόχοι της κρυπτογραφίας σύμφωνα με τους Delfs & Knebl είναι η εμπιστευτικότητα, η ακεραιότητα, η αυθεντικοποίηση και η μη αποκήρυξη και αναλυτικότερα:

- Εμπιστευτικότητα: Η απόκρυψη της πληροφορίας από μη εξουσιοδοτημένους χρήστες.
- Ακεραιότητα δεδομένων: Το να μην υποστούν τροποποίηση τα δεδομένα κατά τη μετάδοσή τους.
- Αυθεντικοποίηση: Ο έλεγχος της προέλευσης των δεδομένων, δηλαδή η επιβεβαίωση πως προέρχονται από το σωστό αποστολέα και δεν είναι αποτέλεσμα παρεμβολής τρίτου.
- Μη αποκήρυξη: Το να μην μπορεί να αρνηθεί ένας χρήστης την αποστολή μηνύματος (Delfs & Knebl, 2007).

Στην κρυπτογραφία υπάρχουν δύο μεγάλες κατηγορίες μεθόδων κρυπτογράφησης οι οποίες χρησιμοποιούνται ευρέως. Η μια μέθοδος είναι η συμμετρική κρυπτογράφηση, ενώ η δεύτερη είναι η κρυπτογράφηση δημόσιου κλειδιού.

2.5.1. Συμμετρική Κρυπτογράφηση

Πρόκειται για την μέθοδο στην οποία και για την κρυπτογράφηση αλλά και για την αποκρυπτογράφηση ενός μηνύματος χρησιμοποιείται το ίδιο κλειδί. Η συμμετρική κρυπτογράφηση αποτελείται από τα παρακάτω μέρη:

- Το αρχικό μήνυμα (plaintext) το οποίο είναι το μήνυμα που χρειάζεται να κρυπτογραφηθεί.
- Το μυστικό κλειδί (secret key) με τη χρήση του οποίου θα κρυπτογραφηθεί το μήνυμα, μέσω ενός αλγορίθμου κρυπτογράφησης.

- Τον αλγόριθμο κρυπτογράφησης (encryption algorithm) που εφαρμόζει μια συγκεκριμένη τεχνική κρυπτογράφησης και παράγει το κρυπτογραφημένο μήνυμα γνωστό ως κρυπτοκείμενο. Συνήθως, χρησιμοποιείται αντικατάσταση και αντιμετάθεση.
- Το κρυπτοκείμενο ή αλλιώς γνωστό ως ciphertext το οποίο είναι το κρυπτογραφημένο μήνυμα που είναι αποτέλεσμα της κρυπτογράφησης.
- Τον αλγόριθμο αποκρυπτογράφησης (decryption algorithm) που εφαρμόζει την αντίθετη τεχνική από αυτή του αλγορίθμου κρυπτογράφησης και με τη χρήση του κλειδιού βρίσκει το αρχικό μήνυμα (Stallings, 2012 σελ.32).

Έτσι, χρησιμοποιώντας έναν αλγόριθμο κρυπτογράφησης και με τη χρήση ενός μυστικού κλειδιού γίνεται μετατροπή του αρχικού μηνύματος σε κρυπτοκείμενο. Το κρυπτοκείμενο αυτό μπορεί να ερμηνευθεί και να αξιοποιηθεί μόνο αν αποκρυπτογραφηθεί με τη χρήση του αλγορίθμου αποκρυπτογράφησης και του ίδιου μυστικού κλειδιού (Stallings, 2012, σελ.32,33).

Υπάρχει ένα πλήθος συμμετρικών αλγορίθμων κρυπτογράφησης που χωρίζονται σε δύο κατηγορίες, τους αλγόριθμους κρυπτογράφησης τμήματος (Block Ciphers) με πιο γνωστούς τους DES, Blowfish και Triple – DES και τους αλγόριθμους κρυπτογράφησης ροής (Stream Ciphers) όπως ο SEAL.

Τα συστήματα συμμετρικής κρυπτογράφησης είναι ευάλωτα σε δύο είδη επιθέσεων, την κρυπτανάλυση και την εξαντλητική αναζήτηση (brute force attack). Η κρυπτανάλυση εκμεταλλεύεται χαρακτηριστικά του κρυπταλγορίθμου που χρησιμοποιείται, του μηνύματος και του κρυπτοκειμένου σε μια προσπάθεια να βρεθεί είτε το κλειδί είτε το αρχικό μήνυμα. Το brute force attack είναι η δοκιμή κάθε πιθανού κλειδιού μέχρι να προκύψει το αρχικό μήνυμα (Stallings, 2012, σελ.36).

2.5.2. Κρυπτογράφηση δημόσιου κλειδιού

Η κρυπτογράφηση δημόσιου κλειδιού είναι μέθοδος ασύμμετρης κρυπτογράφησης. Στη μέθοδο αυτή, τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση ενός μηνύματος χρησιμοποιούνται διαφορετικά κλειδιά. Στην κρυπτογράφηση χρησιμοποιείται

ένα δημόσιο κλειδί ενώ στην αποκρυπτογράφηση ένα ιδιωτικό κλειδί. Αποτελείται από τα εξής μέρη:

- Το αρχικό μήνυμα (plaintext), δηλαδή το μήνυμα που χρειάζεται να κρυπτογραφηθεί.
- Το δημόσιο και το ιδιωτικό κλειδί. Το δημόσιο κλειδί (public key) είναι ένα κλειδί γνωστό σε όσους χρήστες συμμετέχουν στην επικοινωνία. Το ιδιωτικό κλειδί (private key) παράγεται από κάθε χρήστη χωριστά και δε γίνεται γνωστό στους υπόλοιπους χρήστες, είναι επομένως μυστικό. Από αυτά τα δύο κλειδιά το ένα χρησιμοποιείται για την κρυπτογράφηση και το άλλο για την αποκρυπτογράφηση με τους αντίστοιχους απαραίτητους μετασχηματισμούς.
- Τον αλγόριθμο κρυπτογράφησης (encryption algorithm) που εφαρμόζει μετασχηματισμούς και τεχνικές κρυπτογράφησης που συνήθως περιέχουν μαθηματικές συναρτήσεις και παράγει το κρυπτογραφημένο μήνυμα γνωστό ως κρυπτοκείμενο.
- Το κρυπτοκείμενο (ciphertext) το οποίο είναι το κρυπτογραφημένο μήνυμα που προκύπτει από την κρυπτογράφηση.
- Τον αλγόριθμο αποκρυπτογράφησης (decryption algorithm) που χρησιμοποιεί το κλειδί και ανακτά το αρχικό μήνυμα (Stallings, 2012, σελ.284).

Συνοπτικά, ο χρήστης παράγει δύο κλειδιά. Ένα που κρατά μυστικό, το ιδιωτικό κλειδί και ένα που μοιράζεται δημόσια, το δημόσιο κλειδί. Στη συνέχεια με τον αλγόριθμο κρυπτογράφησης και το δημόσιο κλειδί του παραλήπτη του μηνύματος κρυπτογραφεί το μήνυμα και παράγει το κρυπτοκείμενο. Ο παραλήπτης για να διαβάσει το μήνυμα, χρησιμοποιεί το ιδιωτικό του κλειδί και τον αλγόριθμο αποκρυπτογράφησης (Stallings, 2012, σελ. 284,287,288).

Οι πιο γνωστοί αλγόριθμοι κρυπτογράφησης δημόσιου κλειδιού είναι ο RSA και το πρωτόκολλο Diffie – Hellman. Ταυτόχρονα τα τελευταία χρόνια έχει κάνει την εμφάνισή της και η κρυπτογράφηση που βασίζεται σε χαρακτηριστικά ή διαφορετικά η τεχνική attribute – based encryption.

Τα συστήματα κρυπτογράφησης δημόσιου κλειδιού μπορούν επίσης να είναι ευάλωτα στην εξαντλητική αναζήτηση όπως και τα συστήματα συμμετρικής κρυπτογράφησης, αλλά πλέον σχεδιάζονται ώστε το κλειδί που παράγεται να είναι αρκετά μεγάλο ώστε να μην συμφέρει η παραπάνω τεχνική επίθεσης. Μπορεί επίσης να είναι

ευάλωτα σε τεχνικές που χρησιμοποιούν το δημόσιο κλειδί για να βρουν το ιδιωτικό (Stallings, 2012, 294).

2.6. Κρυπτογράφηση βασισμένη σε χαρακτηριστικά

Η Κρυπτογράφηση βασισμένη σε χαρακτηριστικά, εφεξής attribute – based encryption (ABE), είναι ένα είδος κρυπτογράφησης δημόσιου κλειδιού επομένως πρόκειται για ασύμμετρη κρυπτογράφηση. Έκανε την εμφάνισή της σχετικά πρόσφατα, το 2004, γι' αυτό και δεν είναι τόσο διαδεδομένη όσο άλλα είδη κρυπτογράφησης. Αν και βρίσκεται ακόμα σε ερευνητικό επίπεδο είναι μια πολλά υποσχόμενη μέθοδος όπως θα γίνει εμφανές παρακάτω.

Κύριο χαρακτηριστικό του attribute – based encryption είναι το γεγονός πως το ιδιωτικό κλειδί και το κρυπτογραφημένο κείμενο προκύπτουν με τη χρήση χαρακτηριστικών (attributes) του χρήστη. Επιπλέον, για να αποκρυπτογραφηθεί το κρυπτοκείμενο, πρέπει το κρυπτοκείμενο και το κλειδί να περιέχουν το ίδιο σύνολο χαρακτηριστικών ("Attribute-based encryption", 2020). Τα χαρακτηριστικά μπορεί να είναι ηλικία, εθνικότητα, φύλο, επάγγελμα ή οτιδήποτε μπορεί να αποτελεί αναγνωριστικό ενός ατόμου. Ωστόσο, δεν προτείνεται η χρήση δεδομένων ειδικής κατηγορίας ως attributes γιατί πρέπει να είναι μυστικά για να προστατεύονται.

Υπάρχουν δύο είδη attribute – based encryption. Το attribute – based encryption που βασίζεται σε κάποια πολιτική κλειδιών (Key-Policy Attribute – Based Encryption KP-ABE) και το attribute – based encryption που βασίζεται σε κάποια πολιτική κρυπτοκειμένου (Ciphertext-Policy Attribute – Based Encryption CP-ABE). Και οι δύο τεχνικές βρίσκουν εφαρμογή σε διάφορες περιπτώσεις ("Attribute-based encryption", 2020).

Τα τελευταία χρόνια, σε ερευνητικά έργα γίνονται αναφορές και για attribute – based encryption με χρήση πολλών κεντρικών οντοτήτων (multi – authority attribute – based encryption) οι οποίες παράγουν ιδιωτικά κλειδιά ("Attribute-based encryption", 2020).

3. Βιβλιογραφική Ανασκόπηση

Όπως προαναφέρθηκε, η κρυπτογράφηση βασισμένη σε χαρακτηριστικά, αλλιώς attribute – based encryption βρίσκεται σε ερευνητικό στάδιο. Παρόλα αυτά υπάρχουν αρκετά αξιόλογα έργα στα οποία στηρίχτηκε η παρούσα πτυχιακή και τα οποία παρέχουν πλήρη πληροφόρηση πάνω σε αυτή τη νέα μέθοδο κρυπτογράφησης.

3.1. Έρευνες στο Attribute – Based Encryption

Το 2004, οι Amit Sahai και Brent Waters έκαναν για πρώτη φορά λόγο για attribute – based encryption. Συγκεκριμένα, πριν τη δημοσίευσή τους υπήρχε ένα άλλο είδος κρυπτογράφησης που βασίζεται στην ταυτότητα του χρήστη και κρυπτογραφεί ένα μήνυμα χωρίς την ανάγκη δημοσίου κλειδιού, το Identity – Based Encryption. (Shamir, A., 1985, όπως παρατίθεται στο Sahai & Waters., 2005) Πηγαίνοντάς το ένα βήμα παραπέρα, πρότειναν το Fuzzy Identity – Based Encryption όπου οι ταυτότητες των χρηστών είναι πλέον ένα σύνολο από attributes. Έτσι, ένας χρήστης έχει ένα μυστικό κλειδί για την ταυτότητά του και ένα δημόσιο κλειδί αλλά για να γίνει η αποκρυπτογράφηση πρέπει να έχουν ορισμένη απόσταση μεταξύ τους. Σύμφωνα με τους δύο ερευνητές το Fuzzy Identity – Based Encryption μπορούσε να εφαρμοστεί σε μια νέα για τότε τεχνική το attribute – based encryption όπου μια κεντρική οντότητα κρυπτογραφεί ένα μήνυμα βάσει κάποιων συγκεκριμένων attributes των χρηστών (Sahai & Waters., 2005).

Δύο χρόνια μετά, το 2006, οι Goyal et al αναφέρουν πως στην απλή κρυπτογράφηση οι χρήστες δυσκολεύονται να μοιραστούν τα κρυπτογραφημένα δεδομένα τους, καθώς δε γίνεται ούτε να δίνουν το ιδιωτικό τους κλειδί γιατί μαζί δίνουν πρόσβαση σε όλα τους τα δεδομένα, ούτε γίνεται όμως κάθε φορά οι ίδιοι να αποκρυπτογραφούν τα δεδομένα που πρέπει να δώσουν. Έτσι, προτείνουν ως λύση τη χρήση Attribute – Based Encryption την οποία και εξελίσσουν λίγο παραπάνω. Συγκεκριμένα, εφαρμόζουν Key – Policy Attribute – Based Encryption και προτείνουν ένα μηχανισμό για την εφαρμογή του. Πρόκειται για μια μέθοδο στην οποία το κρυπτοκείμενο περιέχει ένα σύνολο από attribute ενώ για το ιδιωτικό κλειδί ορίζεται μια πολιτική που καθορίζει ποιο κρυπτοκείμενο μπορεί να αποκρυπτογραφήσει αυτό το κλειδί (Goyal et al., 2006).

Το 2007 οι John Bethencourt, Amit Sahai and Brent Waters, παρατήρησαν πως υπήρχε μια παράλειψη όσον αφορά το Attribute – Based Encryption. Η κεντρική οντότητα που ελέγχει αν ο χρήστης πληροί ορισμένα χαρακτηριστικά ώστε να έχει πρόσβαση στα δεδομένα, μπορεί εύκολα να καταληφθεί από κακόβουλη οντότητα χωρίς να γίνει γνωστό. Για αυτό πρότειναν την τεχνική Ciphertext – Policy Attribute – Based Encryption, κατά την οποία το ιδιωτικό κλειδί σχηματίζεται βάσει κάποιων attributes και το ciphertext προκύπτει μέσω της κρυπτογράφησης με τη χρήση πολιτικής που βασίζεται επίσης σε κάποια attributes. Συνεπώς, ο χρήστης θα μπορέσει να αποκρυπτογραφήσει το μήνυμα μόνο αν τα attributes του ικανοποιούν την πολιτική του ciphertext (Bethencourt et al., 2007).

Σε όλες τις παραπάνω δημοσιεύσεις για την υλοποίηση του attribute – based encryption χρησιμοποιούνται κατά κύριο λόγο τα ακόλουθα: Μία access structure, μια δομή δηλαδή που περιέχει τα εξουσιοδοτημένα σύνολα από attributes, το bilinear mapping που «συνδυάζει δύο vector spaces για να αποδώσει στοιχεία ενός τρίτου» (“Bilinear map”, 2020) και τέσσερις βασικές συναρτήσεις που αποτελούν την κατασκευή και είναι οι εξής:

- Setup: Ορίζει το πλαίσιο των attributes και δίνει το master key και τα public parameters.
- Encryption: Αλγόριθμος που παράγει το κρυπτογραφημένο μήνυμα με χρήση των attributes, έχοντας σαν είσοδο τα public parameters και το μήνυμα.
- Key Generation: Αλγόριθμος που παράγει το private key με χρήση του master key και ενός συνόλου attributes.
- Decryption: Αλγόριθμος που αποκρυπτογραφεί το ciphertext με είσοδο τα public parameters, ένα private key, και το ciphertext.

Βασιζόμενοι στις παραπάνω δημοσιεύσεις πολλοί ερευνητές έχουν ασχοληθεί με το attribute – based encryption και προτείνουν διαρκώς τρόπους για τη βελτίωσή του.

Έτσι, οι Rafail Ostrovsky, Amit Sahai and Brent Waters, πήγαν ένα βήμα παραπέρα προτείνοντας αντί για τις μονοτονικές δομές πρόσβασης που χρησιμοποιούνταν ως τότε, μη μονοτονικές (Ostrovsky et al., 2007).

Το 2007 από την άλλη, η Melissa Chase προτείνει το Multi – Authority Attribute – Based Encryption. Σε αυτή την περίπτωση γίνεται χρήση πολλαπλών κεντρικών οντοτήτων καθεμιά από τις οποίες ελέγχει ένα συγκεκριμένο σύνολο από attributes και καθορίζει για

αυτά τα μυστικά τους κλειδιά. Το μήνυμα μπορεί να αποκρυπτογραφηθεί μόνο αν ο χρήστης έχει συγκεκριμένο αριθμό χαρακτηριστικών από κάθε authority. Στο παράδειγμα που εφαρμόζει, χρησιμοποιεί τρεις κεντρικές οντότητες και το ciphertext περιέχει ένα attribute από κάθε οντότητα. Καθώς λοιπόν υπάρχουν πολλές και όχι μια οντότητες υπεύθυνες για τα κλειδιά των χρηστών, η κρυπτογράφηση είναι πιο ασφαλής από το να υπάρχει μόνο μια οντότητα η οποία μπορεί να δεχτεί επίθεση (Chase, 2007).

3.2. Attribute – Based Encryption στο IoT

Με την εξέλιξη του Internet of Things και με την ανακάλυψη του Attribute – Based Encryption ήταν αμέσως επόμενο να αξιοποιηθεί αυτή η μέθοδος για τη βελτίωση της ασφάλειας των έξυπνων συσκευών.

Οι Jianbing Ni et al μίλησαν για την ανάγκη ύπαρξης ενός authorization mechanism ώστε να μπορεί να γίνεται με ασφάλεια η διαχείριση των διαφορετικών προσβάσεων των χρηστών, σε υπηρεσίες που προσφέρει το fog computing. Ένα authorization mechanism θα καθορίζει ποιος έχει πρόσβαση στις υπηρεσίες μέσω κάποιων πολιτικών. Έτσι, μίλησαν για έλεγχο πρόσβασης που θα βασίζεται στο attribute – based encryption. Επομένως, οι χρήστες θα έχουν πρόσβαση σε δεδομένα, πόρους και υπηρεσίες μόνο αν έχουν κάποια συγκεκριμένα attributes (Ni et al., 2018).

Οι Xuanxia Yao et al, μίλησαν για την αναγκαιότητα προστασίας των δεδομένων και συνεπώς και της ιδιωτικότητας, επειδή με το Internet of Things γίνεται μετάδοση πλήθους δεδομένων και δε θα πρέπει να επιτρέπεται σε τρίτες μη εξουσιοδοτημένες οντότητες να έχουν πρόσβαση σε αυτά. Προτείνουν λοιπόν τη χρήση Attribute – Based Encryption, καθώς προσφέρει μεγαλύτερη προστασία κατά τη μετάδοση δεδομένων και διατηρεί την εμπιστευτικότητα, ακεραιότητα και διαθεσιμότητά τους. Παρόλα αυτά, η μέθοδος Attribute – Based Encryption χρησιμοποιεί bilinear pairing που είναι μια ακριβή λειτουργία και δεν συμφέρει να χρησιμοποιηθεί στις εφαρμογές του IoT. Προτείνουν έτσι τη χρήση του αλγορίθμου ECC που δε χρησιμοποιεί pairing, αντί του bilinear Diffie - Hellman για να κάνουν την εφαρμογή του Attribute – Based Encryption πιο προσιτή στο IoT (Yao et al., 2015).

Οι Jiguo Li et al αναφέρονται στο γεγονός ότι το IoT πλέον διαχειρίζεται την αποθήκευση, πρόσβαση και μετάδοση των δεδομένων που συλλέγει, με τη χρήση της τεχνολογίας cloud και συγκεκριμένα μιας πλατφόρμας CloudIoT. Είναι λοιπόν φυσικό επόμενο, η πρόσβαση και η χρήση των δεδομένων να γίνονται με ορισμένο τρόπο και να μην είναι δυνατή από χρήστες που δεν έχουν εξουσιοδότηση. Αυτό εξασφαλίζεται με το Ciphertext – Policy Attribute – Based Encryption καθώς με τη χρήση attributes σε πολιτική κατά την κρυπτογράφηση, μπορεί να εξασφαλιστεί η πρόσβαση στο ciphertext μόνο ατόμων των οποίων τα attributes ταιριάζουν με αυτά του ciphertext (Li et al., 2020).

Οι Xinlei Wang et al, αξιολόγησαν και συνέκριναν τα δύο είδη Attribute – Based Encryption, δηλαδή το Key – Policy Attribute – Based Encryption (KP-ABE) και το Ciphertext – Policy Attribute – Based Encryption (CP-ABE) παρατηρώντας τα οφέλη και τα μειονεκτήματά τους, αλλά και την πολυπλοκότητα και την επάρκειά τους σε συνδυασμό με τη χρήση τους στο IoT και συγκεκριμένα σε φορητές συσκευές (Wang et al., 2014).

3.3. Revocation

Αναπόσπαστο μέρος του Attribute – Based Encryption αποτελεί το revocation δηλαδή η ανάκληση κλειδιών μετά τη χρήση τους.

Σύμφωνα με τους Xiaohui Liang et al, στο Attribute – Based Encryption οι χρήστες πιστοποιούνται ότι μπορούν πράγματι να έχουν πρόσβαση στα δεδομένα και να τα αποκρυπτογραφήσουν. Η πρόσβασή τους όμως από τη στιγμή που την αποκτήσουν είναι πλέον μόνιμη. Επομένως, ακόμα και να διακοπεί η πρόσβασή τους για κάποιο λόγο, αυτοί θα μπορούν ακόμα να την έχουν, γεγονός που μπορεί να οδηγήσει πλέον σε κινδύνους για τα δεδομένα που κρατούνται. Γι’ αυτό το λόγο, πρότειναν πως είναι απαραίτητος ένας μηχανισμός για revocation με τον οποίο κάθε χρήστης θα έχει ένα μοναδικό χαρακτηριστικό. Ο system manager αντιστοιχεί μυστικά κλειδιά σε κάθε χρήστη με αναγνωριστικό και ορίζει ένα χρονικό διάστημα. Έτσι, τα δεδομένα κρυπτογραφούνται αλλά για να έχει ο χρήστης πρόσβαση σε αυτά πρέπει το αναγνωριστικό του να βρίσκεται στο συγκεκριμένο χρονικό διάστημα που ορίστηκε (Liang et al, 2011).

Οι Nuttapong Attrapadung και Hideki Imai, αναγνωρίζοντας την ανάγκη του revocation για καλύτερη προστασία των δεδομένων πρότειναν δύο μοντέλα το έμμεσο

(indirect) και το άμεσο (direct) revocation. Στο indirect revocation η κεντρική οντότητα ανανεώνει τα μυστικά κλειδιά περιοδικά. Έτσι, αν κάποιος χρήστης ανακληθεί δε θα έχει πλέον πρόσβαση. Στο direct revocation ο αποστολέας ορίζει μια λίστα με τους χρήστες που έχουν ανακληθεί και ο έλεγχος γίνεται άμεσα τη στιγμή της κρυπτογράφησης (Attrapadung et al., 2009).

Οι Joseph K. Liu et al, προτείνουν μία τεχνική revocation κατά την οποία τα μυστικά κλειδιά ενημερώνονται ανά κάποιο χρονικό διάστημα, επομένως κάποια στιγμή λήγουν, ενώ υπάρχει επίσης μια λίστα με χρήστες με ακόμα λειτουργικά κλειδιά. Η τεχνική αυτή βασίζεται στο direct revocation (Liu et al., 2018).

Τέλος, οι Mei Jiang et al, μίλησαν για το Internet of Vehicles μια επέκταση του IoT που αφορά τα οχήματα και τις μεταφορές, χαρακτηριστικό του οποίου είναι η τοποθεσία και η συχνή και γρήγορη αλλαγή της. Έτσι, πρότειναν μια τεχνική κατά την οποία θα χρησιμοποιείται Attribute – Based Encryption αλλά ταυτόχρονα θα πραγματοποιείται έλεγχος πρόσβασης μέσω τοποθεσίας. Με αυτό εννοούσαν δηλαδή πως ο χρήστης θα μπορεί να έχει πρόσβαση στα δεδομένα που θέλει να αποκρυπτογραφήσει μόνο αν τα attributes που έχει ικανοποιούν την πολιτική που έχει οριστεί και παράλληλα έχει φτάσει σε μια συγκεκριμένη τοποθεσία. Αυτή η τεχνική, αποτελεί παράλληλα λύση για το ζήτημα του revocation που αναφέρθηκε παραπάνω (Jiang et al., 2020).

4. Υλοποίηση

4.1. Κρυπτογραφική Μέθοδος

4.1.1. Ciphertext – Policy Attribute – Based Encryption

Στην παρούσα πτυχιακή εργασία αξιοποιείται η μέθοδος Ciphertext – Policy Attribute – Based Encryption αναφερόμενη από εδώ και στο εξής ως CP-ABE κατά την οποία όπως προαναφέρθηκε το ιδιωτικό κλειδί σχηματίζεται βάσει κάποιων attributes και το ciphertext προκύπτει μέσω της κρυπτογράφησης με τη χρήση πολιτικής που βασίζεται επίσης σε κάποια attributes (Bethencourt et al., 2007).

Ως μέθοδος κρυπτογράφησης παρουσιάζει αρκετά οφέλη στην ασφάλεια και προστασία των δεδομένων. Σε αντίθεση με άλλες τεχνικές η χρήση των attributes προσδίδει μοναδικότητα στη διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης. Ουσιαστικά παρέχει έλεγχο πρόσβασης βασισμένο στους ρόλους των χρηστών. Η μοναδικότητα οφείλεται στο ότι ο κάθε χρήστης διαθέτει τα δικά του χαρακτηριστικά. Αυτό σημαίνει πως το κλειδί του θα διαφέρει από το κλειδί ενός άλλου χρήστη και στη συνέχεια θα μπορεί να αποκρυπτογραφήσει το κρυφό μήνυμα μόνο αν αυτά τα χαρακτηριστικά ταιριάζουν με τα χαρακτηριστικά με τα οποία έχει κρυπτογραφηθεί το μήνυμα. Υπάρχουν λοιπόν περισσότερες συνθήκες που πρέπει να ισχύουν για να γίνει μια επιτυχημένη κρυπτογράφηση και μια επιτυχημένη αποκρυπτογράφηση, σε αντίθεση με πιο απλές τεχνικές κρυπτογράφησης που χρησιμοποιούν ένα κλειδί και έναν αλγόριθμο κρυπτογράφησης.

Συνεπώς, χάρη στα παραπάνω ένας κακόβουλος χρήστης είναι πιο δύσκολο να αποκτήσει μη εξουσιοδοτημένη πρόσβαση στις πληροφορίες. Παράλληλα, η εφαρμογή μαθηματικών συναρτήσεων για την κατασκευή και λειτουργία των αλγορίθμων που χρειάζονται για την κατασκευή του CP – ABE, κάνει τη μέθοδο αυτή πιο περίπλοκη. Ως αποτέλεσμα, ο κώδικας που χρησιμοποιείται σε επόμενο στάδιο για να υλοποιηθούν οι αλγόριθμοι είναι πιο πολύπλοκος. Έτσι, προσδίδεται μεγαλύτερη ασφάλεια καθώς είναι πιο δύσκολο για έναν πιθανό επιτιθέμενο να αποκτήσει πρόσβαση στο τελικό σύστημα κρυπτογράφησης.

Στη μέθοδο αυτή, ο όρος που συναντάται πιο συχνά είναι τα attributes. Τα attributes είναι χαρακτηριστικά που διακρίνουν κάθε άτομο. Μπορεί να έχουν οποιαδήποτε μορφή. Μπορεί να είναι αριθμός, λέξη, γράμμα ή και τα τρία μαζί, μπορεί να είναι χαρακτηριστικά ενός ατόμου. Για παράδειγμα, σε έναν εργασιακό χώρο ένας εργαζόμενος χαρακτηρίζεται από την ειδικότητά του, όπως Information Security Consultant και το γραφείο στο οποίο εργάζεται, όπως Γραφείο 3. Επομένως τα χαρακτηριστικά του είναι Information Security Consultant και Γραφείο 3 και αυτά μπορούν να χρησιμοποιηθούν για να παραχθεί το ιδιωτικό κλειδί του και για να μπορέσουν στη συνέχεια να αποκρυπτογραφήσουν ένα μήνυμα.

Υπάρχει μια κεντρική οντότητα (central authority) η οποία είναι υπεύθυνη για τον έλεγχο των attributes και την παραγωγή των μυστικών κλειδιών (secret keys).

Χρησιμοποιείται ακόμα, μια μονοτονική δομή πρόσβασης (access structure). Η δομή αυτή αποτελείται από υποσύνολα από attributes τα οποία ανήκουν σε ένα ευρύτερο σύνολο και είναι εξουσιοδοτημένα από το central authority. Καλείται μονοτονική γιατί ισχύει ότι:

$\forall$ B, C αν $B \subseteq C$ και $B \in A$ τότε και $C \in A$, όπου A το

υποσύνολο των attributes ενός συνόλου $[P_1, P_2, \dots, P_n]$.

Εδώ, η δομή έχει τη μορφή δέντρου. Συγκεκριμένα, το δέντρο έχει τα φύλλα και τους κόμβους που δεν είναι φύλλα. Τα φύλλα x χαρακτηρίζονται από μια τιμή threshold $k_x = 1$ και attributes. Οι κόμβοι x που δεν είναι φύλλα χαρακτηρίζονται από τον αριθμό των παιδιών που έχουν num_x και επίσης ένα threshold όπου $0 \leq k_x \leq num_x$. Ορίζονται και τρεις συναρτήσεις. Η parent(x) που ορίζει τον γονικό κόμβο του x, η att(x) που ορίζει τα attributes ενός φύλλου του δέντρου και η index(x) που είναι μια αρίθμηση που ορίζει τη θέση του κόμβου στο δέντρο.

Επίσης, χρησιμοποιούνται Bilinear Maps. Ουσιαστικά πρόκειται για μια μέθοδο αντιστοίχισης στοιχείων από δύο σύνολα κρυπτογράφησης σε ένα τρίτο σύνολο (Bethencourt et al., 2007). Έστω ότι υπάρχουν δύο σύνολα G_1 και G_2 και πρέπει να αντιστοιχηθούν τα στοιχεία τους με ένα τρίτο σύνολο G_3. Ισχύει ότι το bilinear map που σχηματίζεται είναι το:

e: $G1 \times G2 \rightarrow G_3$ τέτοιο ώστε για όλα τα $u \in G1$, $v \in G2$, $a, b \in Z$, ισχύει: $e(u^a, v^b) = e(u, v)^{ab}$ (Bethencourt et al., 2007)

Το βασικό κομμάτι αυτής της μεθόδου κρυπτογράφησης είναι οι αλγόριθμοι που υλοποιούν όλη τη διαδικασία της κρυπτογράφησης. Οι αλγόριθμοι αυτοί είναι τέσσερις όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, εδώ ωστόσο θα αναλυθούν περισσότερο βασισμένοι στο Ciphertext – Policy Attribute – Based Encryption.

Setup Algorithm

Ορίζει το πλαίσιο λειτουργίας και παράγει τις δημόσιες παραμέτρους (public parameters PK) και το master key για ένα authority. Για το δημόσιο κλειδί χρησιμοποιεί ένα bilinear σύνολο G_1 πρώτης τάξης p, με ένα generator g και τυχαίους εκθέτες α, β που ανήκουν στο σύνολο Z_p.

Έτσι προκύπτει το public key: $PK = G_1, g, h = g^{\beta}, f = g^{1/\beta}, e(g, g)^{\alpha}$ και το master key: (β, g^{α})

Encryption Algorithm

Ο αλγόριθμος κρυπτογράφησης έχει ως είσοδο τις public parameters PK, το μήνυμα που πρέπει να κρυπτογραφηθεί M, και την access structure που αναφέρθηκε προηγουμένως, έστω A. Εδώ το A είναι μια δομή με μορφή δέντρου, επομένως ονομάζεται T για διευκόλυνση. Το δέντρο αποτελείται από x κόμβους, οι οποίοι ανήκουν σε ένα σύνολο Y. Ο αλγόριθμος ξεκινά ορίζοντας ένα πολυώνυμο q_x για κάθε κόμβο x, ξεκινώντας από τη ρίζα και θέτοντας ως βαθμό $d_x = k_x - 1$, όπου k_x το threshold. Θέτει $q_x(0) = q_{parent}(x)(index(x))$. Η ρίζα θα έχει πολυώνυμο q_R και βαθμό d_R και τίθεται $q_R(0) = s$, όπου $s \in Z_p$. Έτσι, το ciphertext προκύπτει από το

$$CT = (T, \tilde{C} = Me\ (g, g)^{\alpha s}, C = h^s, \forall y \in Y: C_y = g^{q_y(0)}, C'_y = H\ (att(y))^{q_y(0)})$$

και έχει σχηματιστεί βασισμένο σε attributes, μέσω πολιτικών που είναι λογικές εκφράσεις με στοιχεία που διαχωρίζονται με τους τελεστές AND και OR για να διευκρινιστεί ο συσχετισμός τους. Γι' αυτό και γίνεται λόγος για ciphertext – policy.

Έτσι, εισάγω στον αλγόριθμο το M και προκύπτει το ciphertext, το οποίο για να αποκρυπτογραφηθεί, ο χρήστης πρέπει να έχει κάποια attributes που να ικανοποιούν το ciphertext policy ταιριάζοντας με τα attributes που αυτό έχει.

Key Generation Algorithm

Ο αλγόριθμος χρησιμοποιεί το master key MK και ένα σύνολο από attributes S. Κάθε attribute συμβολίζεται με j. Ο αλγόριθμος θέτει $r \in Z_p$ και $r_j \in Z_p$. Έτσι παράγεται το private secret key:

$$SK = (D = g^{(\alpha+r)/\beta}, \forall j \in S: D_j = g^r \cdot H(j)^{r}{}_j, D'_j = g^{r}{}_j).$$

Decryption Algorithm

Ο αλγόριθμος χρησιμοποιεί ως είσοδο το ciphertext CT, έναν κόμβο x από το δέντρο και το private secret key SK. Λειτουργεί αναδρομικά και εφαρμόζονται οι ακόλουθες περιπτώσεις:

Αν ο κόμβος x είναι φύλλο του δέντρου, ορίζεται i = att(x) και:

Αν $i \in S$ τότε:

$$\text{DecryptNode}(CT,SK, x) = e(D_i,Cx) / e(D'_i, C'_x) =$$

$$= e\,(g^r \cdot H(i)^{r}{}_i,\ h^q{}_x(^{(0)} / e\,(gr\ ,{}_i H(i)^q\ (^{(0))}_x$$

$$= e\,(g, g)^{rq}{}_x{}^{(0)}$$

Αν $i \notin S$ τότε: DecryptNode(CT,SK, x) = ⊥

Αν ο κόμβος x δεν είναι φύλλο τότε καλείται το F_z = DecryptNode(CT,SK, z), όπου z τα παιδία του κόμβου x και $z \in S_x$, ώστε $F_z \neq \perp$.

Αν υπάρχει το σύνολο S_x τότε:

$F_x = \prod_{z\ g\ Sx} Fz^{\Delta i,S'x(0)}$, όπου i=index(z) και $S'_x = \{\text{index}(z): z \in Sx\}$

$F_x = \prod_{z\ g\ Sx}(e(g,g)^{r\cdot qz(0)})^{\Delta i,S'x(0)} = \prod_{z\ g\ Sx}(e(g,g)^{r\cdot qparent(z)\cdot(index(z))})^{\Delta i,S'x(0)} = \prod_{z\ g\ Sx}(e(g,g)^{r\cdot qx(i)\cdot\Delta i,S'x(0)}) = \prod_{z\ g\ Sx} e(g,g)^{r\cdot qx(0)}$

Αν δεν υπάρχει το σύνολο S_x τότε: $F_z \neq \perp$ γιατί ο κόμβος δεν ικανοποίησε τα κριτήρια.

Ο αλγόριθμος ξεκινά λοιπόν να εφαρμόζεται πρώτα από τη ρίζα r του δέντρου και αν ικανοποιείται από το σύνολο S, τότε ορίζεται:

$A = \text{DecryptNode}(CT, SK, r) = e\,(g, g)^{rq}{}_R{}^{(0)} = e\,(g, g)^{r}{}_s$

Τα attributes πρέπει να ικανοποιούν το policy του ciphertext για να μπορέσει ο χρήστης να το αποκρυπτογραφήσει και να ανακτήσει το αρχικό μήνυμα.

Τέλος, η αποκρυπτογράφηση πραγματοποιείται τελικά με τη συνάρτηση:
C/ (e (C, D) / A) = C / (e (h^s, $g^{(\alpha+r)/\beta}$) / e(g, g)^r)$_s$= M , όπου M το αρχικό μήνυμα. (Bethencourt et al., 2007)

4.1.2. Revocation

Όπως αναφέρεται σε προηγούμενες δημοσιεύσεις, το central authority δεν είναι σε θέση να γνωρίζει ποιος χρήστης έχει ανακληθεί. Επομένως, δεν μπορεί να ξέρει ότι δεν θα πρέπει ο χρήστης αυτός να λάβει κλειδιά. Αυτό μπορεί να αποτελέσει σοβαρό πρόβλημα λόγω του γεγονότος ότι ο χρήστης θα συνεχίσει να έχει πρόσβαση στα δεδομένα και να αποκρυπτογραφεί μηνύματα τα οποία μπορεί μάλιστα να χρησιμοποιήσει με κακές προθέσεις.

Έπρεπε έτσι να χρησιμοποιηθεί ένας πιο ασφαλής τρόπος ανάκλησης κλειδιών. Σε αυτό συνέβαλε η ανανέωση κλειδιών ανά συγκεκριμένο χρονικό διάστημα, κάτι που σημαίνει ότι τα κλειδιά σταματούν να έχουν ισχύ όταν το διάστημα παρέλθει. Έτσι, αν ένας χρήστης έχει ανακληθεί και δεν επιτρέπεται πλέον να έχει πρόσβαση, το κλειδί του θα έχει λήξει οπότε δε θα μπορεί να αποκρυπτογραφήσει τα δεδομένα.

Συγχρόνως, αξιοποιήθηκε και η τοποθεσία του χρήστη. Έτσι, αν θεωρηθεί πως ο χρήστης κινείται μέσα στο χώρο, ορίζεται πως μπορεί να παραχθεί κλειδί για να προχωρήσει στην κρυπτογράφηση του μηνύματός του, μόνο αν βρίσκεται σε συγκεκριμένη απόσταση από την οντότητα που παράγει κλειδιά. Συνεπώς, δεν μπορεί να κρυπτογραφήσει και αποκρυπτογραφήσει δεδομένα κάποιος χρήστης από όπου και να βρίσκεται στο χώρο.

4.2. Τεχνολογίες

Στη σωστή και επιτυχημένη πραγματοποίηση της έρευνας έπαιξαν ουσιώδη ρόλο ποικίλα εργαλεία. Είναι λοιπόν σημαντικό να αναλυθεί ο τρόπος λειτουργίας καθενός από αυτά χωριστά.

4.2.1. OpenABE

Το OpenABE είναι μια βιβλιοθήκη κρυπτογράφησης. Έχει αναπτυχθεί από την Zeutro LLC, μια εταιρεία που παράγει λογισμικά προηγμένων μεθόδων κρυπτογράφησης.

Περιλαμβάνει ένα εύχρηστο application programming interface (API), μια ποικιλία συναρτήσεων και εργαλείων κρυπτογράφησης και ένα πλήθος αλγορίθμων για την τεχνική Attribute – Based Encryption. Σκοπός της κατασκευής της βιβλιοθήκης είναι η αξιοποίησή της σε τομείς που είναι απαραίτητη η προστασία δεδομένων, κατά κύριο λόγο ευαίσθητων.

Η βιβλιοθήκη είναι γραμμένη σε γλώσσα προγραμματισμού C++ και μπορεί να εγκατασταθεί σε έναν αριθμό από περιβάλλοντα όπως Linux, Windows, Mac και Android. Έχει ενσωματωμένη στον κώδικά της ασφαλή αρχιτεκτονική καθώς και προστασία από γνωστές επιθέσεις ώστε να προσφέρει ακόμα μεγαλύτερη ασφάλεια στους τομείς που μπορεί να χρησιμοποιηθεί. Επίσης, μεταξύ των τρόπων υλοποίησης που προτείνονται χρησιμοποιεί κάθε φορά την πιο ασφαλή εκδοχή αλγορίθμου κρυπτογράφησης.

Προσφέρει εύκολη χρήση των κρυπτογραφικών αλγορίθμων που περιλαμβάνει, είναι κατανοητή και καθώς είναι ανοιχτού κώδικα παρέχει την δυνατότητα επέκτασης με νέες τεχνικές χωρίς δυσκολία.

Για την υλοποίηση των αλγορίθμων κρυπτογράφησης της βιβλιοθήκης έπρεπε να επιλεγεί η πιο σωστή και επαρκής από τις τεχνικές κατασκευής που προτείνονται στις ερευνητικές δημοσιεύσεις. Χρησιμοποιήθηκε το bilinear mapping, αλλά και το access structure με δομή δέντρου που αναφέρθηκε παραπάνω και για την υλοποίησηεφαρμόζονται οι γνωστοί τέσσερις αλγόριθμοι setup, key generation, encrypt και decrypt.

Το OpenABE υποστηρίζει public – key encryption, Key – Policy Attribute Based Encryption, Ciphertext – Policy Attribute Based Encryption αλλά και άλλα είδη attribute - based encryption (ABE) με key encapsulation (KEM) και υβριδική κρυπτογράφηση, ενώ εφαρμόζει σε όλα τα παραπάνω είδη κρυπτογράφησης και chosen – ciphertext security δηλαδή τεχνική ασφαλείας βασισμένη σε ένα επιλεγμένο ciphertext με στόχο την παροχή μεγαλύτερης δυνατής ασφάλειας.

Υποστηρίζει authenticated symmetric – key encryption και ψηφιακές υπογραφές οι οποίες παρέχονται από την βιβλιοθήκη OpenSSL. Συγχρόνως, παρέχει πλήθος συναρτήσεων και εργαλείων κρυπτογράφησης μεταξύ των οποίων ψευδοτυχαίες

συναρτήσεις, ψευδοτυχαίες και τυχαίες γεννήτριες, συναρτήσεις παραγωγής κλειδιών, γραμμικά σχέδια για μυστικό διαμοιρασμό (secret sharing schemes) και γενικάαποθήκευση κλειδιών και των στοιχείων που σχετίζονται με αυτά.

Σημαντικό μέρος του OpenABE είναι επίσης μια Μαθηματική βιβλιοθήκη που συμπεριλαμβάνει πολλές μαθηματικές συναρτήσεις καθώς και ένα pairing module και ένα elliptic curve module που χρησιμοποιούνται από τις τεχνικές κρυπτογράφησης που υποστηρίζει (Zeutro LLC, 2018).

Ο προγραμματιστής αφού κατεβάσει και εγκαταστήσει το OpenABE στο περιβάλλον που επιθυμεί, έχει τη δυνατότητα να χρησιμοποιήσει το OpenABE crypto box API υψηλού επιπέδου και να δημιουργήσει μια υλοποίηση, μια εφαρμογή σε γλώσσα C++, αξιοποιώντας το υλικό που του χρειάζεται από τη βιβλιοθήκη. Μια άλλη επιλογή είναι η χρήση του OpenABE API χαμηλού επιπέδου η οποία ωστόσο δεν προτείνεται για αποφυγή λαθών κατά την υλοποίηση της κρυπτογράφησης από τους χρήστες. Η τρίτη επιλογή είναι η χρήση των βοηθητικών προγραμμάτων για command – line τα οποία βοηθούν στην εφαρμογή των μεθόδων CP – ABE, KP – ABE και public – key encryption, αξιοποιώντας το κυρίως OpenABE API.

Η τρίτη επιλογή είναι και αυτή που χρησιμοποιείται στην έρευνα, καθώς αποτελείται από τέσσερα εργαλεία για command – line. Τα εργαλεία χρησιμοποιούνται μέσα στο πρόγραμμα που δημιουργείται, με μορφή εντολών για να γίνει η επιθυμητή υλοποίηση. Επίσης, για κάθε είδος κρυπτογράφησης που υποστηρίζεται από το OpenABE, τα εργαλεία πραγματοποιούν όλες τις λειτουργίες.

Τα τέσσερα εργαλεία για command – line είναι τα ακόλουθα:

oabe_setup: Υλοποιεί τον αλγόριθμο Setup, επομένως παράγει τα public parameters και το master secret key. Τα ορίσματα που δέχεται είναι:

-s: Το scheme, η τεχνική που χρησιμοποιείται, CP ή KP.

-p: Το prefix, το πρόθεμα για τα public ή secret parameter files του authority που δημιουργείται (επιλεκτικό).

-v: Ενεργοποιεί τη λεκτική λειτουργία.

oabe_keygen: Υλοποιεί τον αλγόριθμο Key Generation, επομένως παράγει ένα private key χρησιμοποιώντας σαν είσοδο ένα σύνολο από attributes (CP – ABE), μια πολιτική (KP - ABE) ή ένα ζεύγος public και private key (public – key encryption ή PK Encryption) με είσοδο μια ταυτότητα ενός χρήστη. Τα ορίσματα που δέχεται είναι:

-s : Το scheme, η τεχνική που χρησιμοποιείται, CP, KP ή PK.

-p: Το prefix, το πρόθεμα για τα public ή secret parameter files του authority που δημιουργείται (επιλεκτικό).

-i: Το input, η είσοδος λίστα attribute για το CP, πολιτική για το KP, key id για το PK.

-o: Το output file , η έξοδος δηλαδή ένα αρχείο για το secret key που παράγεται.

-v: Ενεργοποιεί τη λεκτική λειτουργία.

oabe_enc: Υλοποιεί τον αλγόριθμο Encrypt, επομένως παράγει το ciphertext μέσω μιας πολιτικής (CP – ABE), ενός συνόλου attributes (KP - ABE) ή μιας ταυτότητας ενός χρήστη (PK encryption). Τα ορίσματα που δέχεται είναι:

-s : Το scheme, η τεχνική που χρησιμοποιείται, CP, KP ή PK.

-p: Το prefix, το πρόθεμα για τα public ή secret parameter files του authority που δημιουργείται (επιλεκτικό).

-e: πολιτική για το CP, σύνολο attributes για το KP, ταυτότητας αποστολέα για το PK.

-r: Το key Id του παραλήπτη για το PK.

-i: Το input, συγκεκριμένα το αρχείο εισόδου.

-o: Το output, το αρχείο εξόδου για το ciphertext.

-v: Ενεργοποιεί τη λεκτική λειτουργία.

oabe_dec: Υλοποιεί τον αλγόριθμο Decrypt, επομένως παράγει το αρχικό μήνυμα με είσοδο το ciphertext και το private key. Τα ορίσματα που δέχεται είναι:

-s : Το scheme, η τεχνική που χρησιμοποιείται, CP, KP ή PK.

-p: Το prefix, το πρόθεμα για τα public ή secret parameter files του authority που δημιουργείται (επιλεκτικό).

-k : Το key, το αρχείο του secret key για το CP και το KP, το κλειδί του παραλήπτη για το PK.

-i : Το ciphertext, συγκεκριμένα το αρχείο του ciphertext.

-e: Το key Id του αποστολέα για το PK.

-o: Το output, το αρχείο εξόδου για το plaintext δηλαδή το μήνυμα που εμφανίζεται.

-v: Ενεργοποιεί τη λεκτική λειτουργία.

Στα παραπάνω εργαλεία, τα attributes διαχωρίζονται μεταξύ τους με κάθετη γραμμή (|).

Επίσης τα αρχεία των keys και των ciphertexts έχουν μια επικεφαλίδα που περιλαμβάνει το αναγνωριστικό της μεθόδου κρυπτογράφησης, το αναγνωριστικό της elliptic curve ή pairing curve, την έκδοση της βιβλιοθήκης OpenABE, ένα μοναδικό αναγνωριστικό 128-bit. Στο ciphertext, η επικεφαλίδα και το κρυπτογραφημένο μήνυμα προστατεύονται από επιθέσεις.

Τα παραπάνω εργαλεία όπως φαίνεται έχουν απλή δομή γεγονός που οδήγησε στην ενσωμάτωσή τους με αυτή τη μορφή μέσα στο πρόγραμμα που δημιουργείται για την έρευνα.

4.2.2. Ubuntu 18.04

Στην έρευνα χρησιμοποιείται το Ubuntu και συγκεκριμένα η έκδοση Ubuntu 18.04.

Το Ubuntu είναι ένα λειτουργικό σύστημα το οποίο ανήκει στην ευρύτερη οικογένεια των λειτουργικών συστημάτων Linux, δηλαδή έχει κατασκευαστεί βασισμένο στον πυρήνα του Linux. Το Ubuntu συγκεκριμένα, έχει τις ρίζες του στο Debian. Στο μεγαλύτερο μέρος του έχει δημιουργηθεί από λογισμικό ανοιχτού κώδικα. Εκδόθηκε για πρώτη φορά το 2004 και κάποιες εκδόσεις του χρησιμοποιούνται από συσκευές IoT (“Ubuntu”, 2021).

Ο λόγος που χρησιμοποιήθηκε στην παρούσα έρευνα είναι ότι το τερματικό του είναι πιο εύχρηστο. Επομένως, η εγκατάσταση και η χρήση του OpenABE ήταν πιο απλή και η εκτέλεση των προγραμμάτων πιο εύκολη.

Παράλληλα το Ubuntu έχει ενσωματωμένες τεχνικές ασφαλείας. Δίνει χαμηλά προνόμια στους χρήστες, εκτός αν γίνει χρήση της εντολής sudo οπότε και αποκτούν προνόμια administrator. Ακόμα και τότε όμως, υπάρχουν δικλείδες ασφαλείας που αποτρέπουν σημαντικά προβλήματα ή κενά ασφαλείας που μπορεί να προκληθούν στο λειτουργικό σύστημα είτε εν αγνοία είτε συνειδητά από κάποιον χρήστη. Υπάρχει ενσωματωμένο τείχος προστασίας με δυνατότητες διαχείρισης, full disk encryption και directory encryption, ενώ οι πόρτες δικτύου είναι κλειστές. Αυτά είναι μερικές μόνο από τις τεχνικές ασφαλείας που χρησιμοποιούνται, γεγονός που παρέχει μεγαλύτερη ασφάλεια στην διαδικασία της κρυπτογράφησης και προσφέρει ένα πιο ασφαλές περιβάλλον για όλη την υλοποίηση του συστήματος που δημιουργήθηκε (“Ubuntu”, 2021).

4.2.3. Python Programming

Η γλώσσα προγραμματισμού που χρησιμοποιείται είναι η Python. Πρόκειται για μια γλώσσα προγραμματισμού γενικού σκοπού. Είναι γλώσσα υψηλού επιπέδου. Υποστηρίζει αντικειμενοστρέφεια, δηλαδή τη χρήση αντικειμένων και έχει επιρροές από άλλες γλώσσες προγραμματισμού όπως η C, η C++ και η Java (“Python (programming language)”, 2021).

Η Python είναι γλώσσα ανοιχτού κώδικα και αποτελεί μια εύχρηστη γλώσσα προγραμματισμού λόγω των εντολών της που είναι σχετικά πιο απλές σε σύγκριση με άλλες γλώσσες προγραμματισμού. Επιπλέον, περιέχει πλήθος modules και βιβλιοθηκών καλύπτοντας πλήθος λειτουργιών και διευκολύνοντας τον προγραμματιστή με πιο συντομευμένες και αυτοματοποιημένες λειτουργίες. Παράλληλα, μπορεί να αντικαταστήσει πλέον τα shell scripts και τα batch files. Δε σταματά παρόλα αυτά, να εξελίσσεται και να προσθέτει νέα χαρακτηριστικά.

Μπορεί να χρησιμοποιηθεί για ανάπτυξη ιστοσελίδων, ανάπτυξη λογισμικού και εταιρικές εφαρμογές. Συμβάλει σημαντικά στην υλοποίηση επιστημονικών και αριθμητικών υπολογισμών, περιλαμβάνει GUI και τα τελευταία χρόνια βρίσκει μεγάλη

εφαρμογή στον τομέα της εκπαίδευσης (Python Software Foundation, 2020). Όλο και περισσότεροι προγραμματιστές αναπτύσσουν τα προγράμματά τους σε Python, ενώ χρησιμοποιείται για την ανάπτυξη εφαρμογών ακόμα και σε επαγγελματικό επίπεδο από πολλές εταιρείες ανά τον κόσμο. Από την Google και το YouTube, μέχρι Πανεπιστήμια ή ακόμα και ηλεκτρονικά παιχνίδια, η Python πλέον τα τελευταία χρόνια συναντάται παντού.

Ο λόγος που χρησιμοποιήθηκε για τη συγγραφή του κώδικα της έρευνας, ήταν ακριβώς η ευχρηστότητά της σαν γλώσσα όπως αναφέρθηκε, καθώς τα modules και οι βιβλιοθήκες παρείχαν πολλές λειτουργίες που διευκόλυναν την υλοποίηση της αρχικής ιδέας.

4.2.5. Socket Programming

Για την δημιουργία του συστήματος κρυπτογράφησης χρειάζεται η δημιουργία οντοτήτων οι οποίες επικοινωνούν μεταξύ τους. Αυτό επιτυγχάνεται με το socket programming.

Το socket programming είναι μια διαδικασία στον προγραμματισμό κατά την οποία δημιουργείται μια θύρα, το socket, στο δίκτυο, μέσω της οποίας πραγματοποιείται η επικοινωνία μεταξύ δύο διεργασιών. Μέσω του socket επιτυγχάνεται η ανταλλαγή μηνυμάτων μεταξύ δύο μερών πελάτη ή client και του διακομιστή αλλιώς server, καθένας από τους οποίους έχει το δικό του socket.

Ο server είναι αυτός που δέχεται αιτήσεις (requests) και ο client αυτός που τις στέλνει. Συγκεκριμένα, ο server προετοιμάζεται να δεχτεί συνδέσεις ενώ ο client κάνει αίτηση για σύνδεση. Στη συνέχεια, ο server αφού λάβει την αίτηση, την επεξεργάζεται και δίνει απάντηση στον client. Στη συγκεκριμένη περίπτωση επικοινωνίας, η σύνδεση δε διακόπτεται από τη στιγμή που συνδέονται τα sockets μέχρι να κλείσει η σύνδεση (Μαγκούτης & Νικολάου, 2015).

4.3. Υλοποίηση

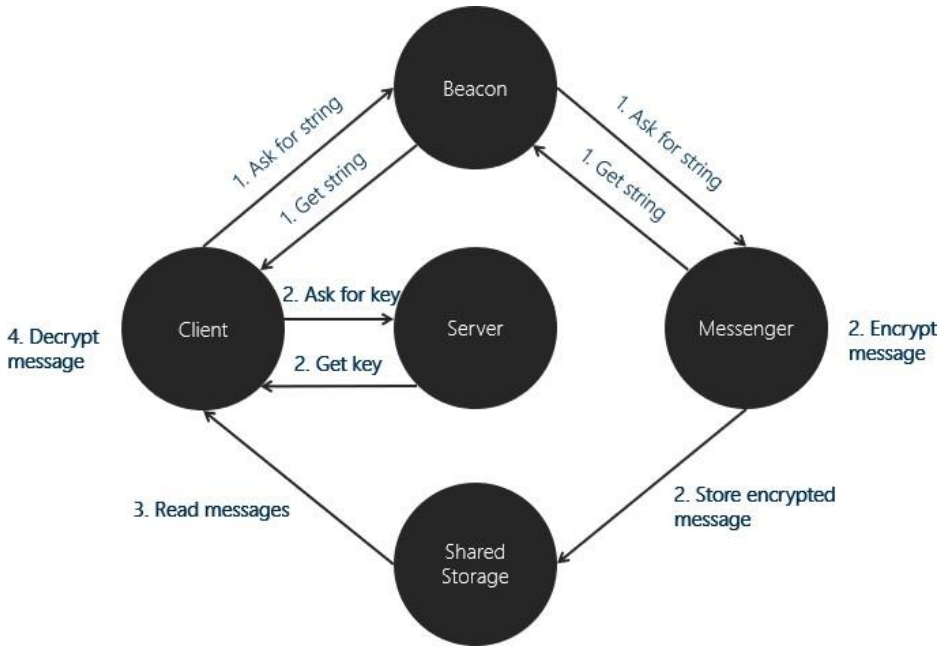
Για να γίνει πιο εύκολα κατανοητή η ερευνητική διαδικασία πρέπει αρχικά να οριστεί το γενικό πλαίσιο και ο σκοπός που δημιουργήθηκε το σύστημα στο οποίο εφαρμόζεται το Attribute – Based Encryption.

Όπως έχει ήδη αναφερθεί υπάρχουν πολυάριθμες έξυπνες συσκευές οι οποίες χρησιμοποιούνται καθημερινά και απαρτίζουν το Internet of Things. Μεταξύ αυτών είναι και τα έξυπνα οχήματα τα οποία στο σύνολό τους αποτελούν μια υποκατηγορία του Internet of Things, αυτή του Internet of Vehicles. Τα οχήματα αυτά ως «έξυπνα», συλλέγουν μέσω αισθητήρων δεδομένα όπως τοποθεσία, στοιχεία οχήματος, πινακίδες, ταχύτητα, ενδείξεις οχήματος, στοιχεία περιβάλλοντος, πιθανούς κινδύνους και άλλα τα οποία στη συνέχεια διαχειρίζονται και ρυθμίζουν για να παρέχεται μια άνετη και ασφαλής εμπειρία οδήγησης. Τα δεδομένα αυτά όμως, πρέπει να διατηρηθούν μυστικά και ασφαλή γιατί σε περίπτωση που βρεθούν σε λάθος χέρια μπορεί να έχουν συνέπειες για το όχημα, τη σωστή λειτουργία του και κυρίως για τον χρήστη του οχήματος, τον οδηγό. Είναι λοιπόν αναγκαίο να κρυπτογραφηθούν και μόνο συγκεκριμένοι χρήστες να έχουν πρόσβαση σε αυτά και γι' αυτό το λόγο χρησιμοποιείται το Attribute – Based Encryption και συγκεκριμένα το Ciphertext – Policy Attribute – Based Encryption. Ιδιαίτερα, η χρήση πολιτικής για να καθοριστεί ποια οχήματα με ποια χαρακτηριστικά μπορούν να έχουν πρόσβαση στο ciphertext χρησιμεύει στην περίπτωση των οχημάτων γιατί μπορεί να καθοριστεί ακριβώς ποια οχήματα θα έχουν πρόσβαση στα δεδομένα.

Στην περίπτωση αυτή, γίνεται μελέτη ενός υποθετικού σεναρίου. Μέσα σε μια μικρή πόλη κινούνται οχήματα σε διάφορες τοποθεσίες. Τα δεδομένα που συλλέγουν τα εν λόγω οχήματα, τα οποία είναι «έξυπνα», κρυπτογραφούνται με χρήση κλειδιών τα οποία περιέχουν τα attributes του κάθε οχήματος αλλά και ένα τυχαίο αλφαριθμητικό (string). Ειδικότερα, το string παράγεται και εκπέμπεται από μια οντότητα, την πηγή, που βρίσκεται σε ένα συγκεκριμένο σημείο της πόλης. Με χρήση των τυχαίων string και των attributes του οχήματος, το central authority παράγει κλειδιά. Τα κλειδιά χρησιμοποιούνται πλέον από μια άλλη ενδιάμεση οντότητα για να κρυπτογραφήσει μηνύματα, τα οποία το κάθε όχημα μόνο αν ικανοποιεί ορισμένες συνθήκες με τα attributes που έχει, μπορεί να τα αποκρυπτογραφήσει και να έχει πρόσβαση σε αυτά. Στη συνέχεια, θα γίνει λεπτομερής περιγραφή των οντοτήτων, του τρόπου λειτουργίας τους και της μεταξύ τους επικοινωνίας.

Όπως φαίνεται και από την περιγραφή που προηγήθηκε στο σύστημα που δημιουργήθηκε υπάρχουν τέσσερις οντότητες. Το central authority – server, το όχημα – client, η πηγή τυχαίων string – beacon και η ενδιάμεση οντότητα – messenger που κάνει κρυπτογράφηση. Συγχρόνως, στην υλοποίηση αξιοποιείται ένα dataset και ένα shared storage για την αποθήκευση κοινών αρχείων.

Ακολουθεί η ανάλυση της δομής και της λειτουργίας καθενός από τα προαναφερόμενα. Πριν από την ανάλυση όμως, αξίζει να αναφερθεί πως όλα τα αρχεία βρίσκονται μέσα στο OpenABE και συγκεκριμένα στον φάκελο cli. Εκεί έχουν δημιουργηθεί με τη σειρά οι φάκελοι b για το beacon, c για τον client, cab για το dataset με τα ταξί, m για τον messenger, s για τον server και ss για το shared storage. Από τους φακέλους, οι b, c, m και s περιλαμβάνουν το αρχείο με τον κώδικα σε python για την υλοποίηση της αντίστοιχης οντότητας με κατάληξη .py αλλά και αρχεία που βρίσκονται πίσω από τα εργαλεία που χρησιμοποιούνται μέσα στον κώδικα για την υλοποίηση της διαδικασίας της κρυπτογράφησης – αποκρυπτογράφησης και αρχεία τα οποία χρησιμοποιούνται σαν είσοδος ή αποτελούν έξοδο της διαδικασίας.



Εικόνα 1: Λειτουργία Συστήματος

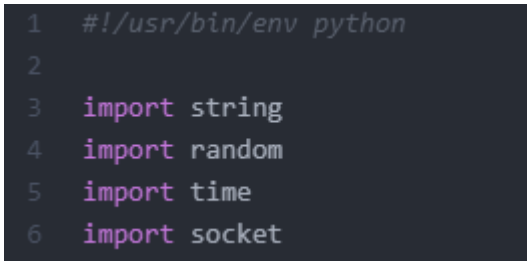
4.3.1. Shared Storage

Το shared storage είναι ένας κοινός φάκελος στον οποίο έχουν πρόσβαση όλες οι οντότητες που δημιουργούνται και στον οποίο αποστέλλονται αρχεία που αξιοποιούνται από περισσότερες από μία οντότητες.

4.3.2. Beacon

Το beacon είναι η πηγή της οποίας ο ρόλος είναι να παράγει τυχαία αλφαριθμητικά, string. Υλοποιείται στο αρχείο beacon.py.Τα string που παράγονται χρησιμοποιούνται από τον client ώστε να δημιουργήσει τυχαία attributes τα οποία κάθε φορά θα είναι μοναδικά. Ακολουθεί αναλυτική περιγραφή της λειτουργίας του beacon.

Αρχικά, γίνεται η εισαγωγή των modules για να λειτουργήσουν σωστά κάποιες από τις εντολές που ακολουθούν. Τα modules είναι τα string, random, time, socket.



Εικόνα 2: Modules του beacon

Δημιουργείται το socket και ορίζεται η διεύθυνση IP και το port που θα χρησιμοποιηθούν για την επικοινωνία. Μέσω του s.bind συσχετίζεται το socket με τα host και port και το beacon περιμένει συνδέσεις καλώντας το s.listen. Θα χρειαστεί όμως να συνδέονται απεριόριστοι clients αφού και τα οχήματα είναι απεριόριστα, γι' αυτό και η αναμονή για συνδέσεις βρίσκεται σε ένα infinite loop. Εκεί πλέον, όταν ο client και ο messenger συνδεθούν και το beacon λάβει τα αιτήματα, το αποδέχεται μέσω του s.accept.

```
s = socket.socket()
host = socket.gethostname()
port = 6565
s.bind((host,port))

production_time = 0
rand_str = ""

clients = list()
def infinity():
    while True:
        yield
for y in infinity():
    s.listen(10)
    print("Beacon is listening ... ")
    con, addr = s.accept()

    clients.append(con)
    print('Client_{} connected to beacon: {}'.format(y, addr))
    print(port, host)
    print(host)
    print("Client is connected to beacon...")
```

Εικόνα 3: Socket και δημιουργία πολλών clients

Τώρα, ο client και ο messenger ζητούν από το beacon να τους δώσει ένα string. Έτσι, το beacon με ένα string generator παράγει τυχαία string που αποτελούνται από αριθμούς και κεφαλαία και πεζά γράμματα.

```
    message = con.recv(1024).decode()
    print("Client sent: " + str(message))


    def string_generator(size=10, characters=string.ascii_letters + string.digits):
        return ''.join(random.choice(characters) for _ in range(size))
```

Εικόνα 4: Δημιουργία τυχαίου string

Πριν όμως στείλει αυτό το string πρέπει να ακολουθήσει μια διαδικασία. Οι δύο οντότητες client και messenger ζητούν συνέχεια να τους στείλει το beacon ένα string. Όμως το beacon θα στείλει καινούριο string μόνο αν έχει περάσει ένα λεπτό μεταξύ της στιγμής που ζήτησαν string και της προηγούμενης φοράς που τους έστειλε string το beacon. Αν έχει περάσει λιγότερος χρόνος τους στέλνει πάλι το προηγούμενο string. Για να γίνει αυτό, υπολογίζεται η στιγμή που ζητείται νέο string, το start_time και η στιγμή που παράγεται το νέο string production_time. Στη συνέχεια, υπολογίζεται το διάστημα (interval) που μεσολάβησε από το production_time μέχρι το start_time. Χρησιμοποιείται ένα if σύμφωνα με το οποίο, αν το interval είναι μικρότερο των εξήντα δευτερολέπτων τότε στέλνεται το

προηγούμενο ίδιο string, αλλιώς παράγεται και στέλνεται ένα νέο string όπως φαίνεται παρακάτω. Τέλος, γίνεται η αποστολή όποιου string αποφασίστηκε να αποσταλεί.

```
38      start_time = time.time()
39      print("Start time is: ", start_time)
40
41      interval = start_time - production_time
42      print("Interval is: ", interval)
43
44      if interval <= 60:
45          print("Take previous string: ", rand_str)
46          production_time = time.time()
47          print("Production time is: ", production_time)
48
49      else:
50          rand_str = string_generator()
51          print("String is: ", rand_str)
52          production_time = time.time()
53          print("Production time is: ", production_time)
54
55      data = rand_str
56      con.send(data.encode())
57
58      con.close()
```

Εικόνα 5: Χρόνος αποστολής string

4.3.3. Messenger

Ο messenger είναι η ενδιάμεση οντότητα ο ρόλος της οποίας είναι να κάνει την κρυπτογράφηση και υλοποιείται με το messenger.py. Η περιγραφή της διαδικασίας γίνεται στη συνέχεια:

Αρχικά, όπως είναι απαραίτητο ορίζονται τα κατάλληλα modules και το path που θα χρειαστεί παρακάτω.

```
1  #!/usr/bin/env python
2
3  from subprocess import Popen, PIPE, STDOUT
4  from time import sleep
5  import time
6  import shutil
7  import os
8  import socket
```

Εικόνα 6: Modules του messenger

```
10  dirpath = "/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/ss/"
11  shutil.rmtree(dirpath)
12  sleep(2)
13  os.mkdir(dirpath)
```

Εικόνα 7: Path του αρχείου

Στη συνέχεια, γίνεται η σύνδεση με το beacon μέσω της συνάρτησης s.connect(), και ο messenger ζητά ένα string από το beacon. Έτσι, λαμβάνει μέσω της recv() το random_string που παράχθηκε και θα χρησιμοποιηθεί σε επόμενο βήμα κατά την κρυπτογράφηση.

```
16      s = socket.socket()
17      host = socket.gethostname()
18      port = 6565
19      s.connect((host,port))
20      print("Connected to beacon... ")
21
22      message = "Give me a string."
23      s.send(message.encode())
24
25      random_string = s.recv(1024).decode()
26      print('String received from beacon: ' + random_string)
27      s.close()
```

Εικόνα 8: Δημιουργία socket και αίτημα για string

Ακολουθεί ο υπολογισμός του χρόνου T καθώς και του χρόνου T' που θα ισχύει έξι δευτερόλεπτα αργότερα, δηλαδή T' = T + 6. Αυτό το T', το οποίο στον κώδικα έχει οριστεί ως timex θα χρησιμοποιηθεί επίσης στην κρυπτογράφηση.

```
29      timenow1 = time.time()
30      timenext = timenow1 + 6
31      timenext = timenext // 1
32      time2 = int (timenext)
33      t2 = str(time2)
34
35      timex = "%s" % t2
```

Εικόνα 9: Υπολογισμός T και T'

Φτάνει λοιπόν το βήμα της κρυπτογράφησης, όπου χρησιμοποιείται το oabe_enc και παίρνει τιμές στα ορίσματά του. Στο -s ορίζεται το σχήμα της κρυπτογράφησης CP.

Στο -e ορίζεται το policy σύμφωνα με το οποίο θα γίνει η κρυπτογράφηση και συγκεκριμένα θα γίνει για όσα οχήματα έχουν το τυχαίο string και χρόνο μικρότερο του timex. Δίνεται επίσης το αρχείο input.txt ως αρχείο εισόδου, το οποίο περιέχει το μήνυμα που θα κρυπτογραφηθεί, έστω εδώ το "hello world!" και το αρχείο εξόδου input.cpabe που περιέχει το ciphertext που προέκυψε.

```
cmd = 'oabe_enc -s CP -p org1 -e "(('+random_string+') and (T < '+timex+' ))" \-i input.txt -o input.cpabe'
print(cmd)
p = Popen(cmd, shell=True, stdin=PIPE, stdout=PIPE, stderr=STDOUT, close_fds=True)
output = p.stdout.read()
print(output)
print("\n Encryption successful. \n")
```

Εικόνα 10: Κρυπτογράφηση

Τέλος, το αρχείο input.cpabe που περιέχει το ciphertext, αφού δημιουργηθεί αντιγράφεται στο shared storage με αύξων αριθμό για να μπορεί να το χρησιμοποιήσει ο client για να το αποκρυπτογραφήσει. Συγκεκριμένα είναι για παράδειγμα 1.cpabe, 2.cpabe.

```
    n = 1
    while os.path.exists('/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/ss/%s.cpabe' % n):
        n += 1

    shutil.copy2('/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/m/input.cpabe' ,
                 '/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/ss/%s.cpabe' % n)
```

Εικόνα 11: Αντιγραφή κρυπτογραφημένου αρχείου στο shared storage

4.3.4. Central Authority

Ο ρόλος του central authority δηλαδή της κεντρικής οντότητας ή αλλιώς του server, είναι να παράγει κλειδιά και να τα στέλνει στο όχημα – client. Η υλοποίηση του authority πραγματοποιείται στο αρχείο server.py. Παρακάτω θα γίνει μια αναλυτική περιγραφή της λειτουργίας του server.

Αρχικά, εισάγονται τα modules που χρειάζονται για να λειτουργήσουν κάποιες ομάδες εντολών. Ιδιαίτερα, γίνεται import shutil, os, time, socket. Εν συνεχεία, καλείται το εργαλείο που θα κάνει setup τη βιβλιοθήκη για το CP – ABE με τις εξής εντολές:

```
#!/usr/bin/env python

import shutil
import os
import time
import socket

from subprocess import Popen, PIPE, STDOUT

cmd = 'oabe_setup -s CP -p org1'
p = Popen(cmd, shell=True, stdin=PIPE, stdout=PIPE, stderr=STDOUT, close_fds=True)
output = p.stdout.read()
print(output)
```

Εικόνα 12: Modules του central authority και setup της βιβλιοθήκης

Στην εντολή oabe_setup με το -s διευκρινίζεται το σχήμα κρυπτογράφησης που θα χρησιμοποιηθεί, δηλαδή το CP (Ciphertext Policy).

Έπειτα, δημιουργείται το socket που λειτουργεί με τον τρόπο που αναφέρθηκε προηγουμένως στο beacon. Ορίζεται η IP και το port, διαφορετικό από αυτό του beacon, τα οποία συσχετίζονται με το socket μέσω του s.bind() και ο server περιμένει συνδέσεις με το s.listen(). Μέσω ενός infinite loop συνδέονται απεριόριστοι clients και όταν ο server λάβει το αίτημα του client, το αποδέχεται με το s.accept().

```
s = socket.socket()
host = socket.gethostname()
port = 9898
s.bind((host,port))

clients = list()
def infinity():
    while True:
        yield
for idx in infinity():
    s.listen(10)
    print("Server is listening ... ")
    con, addr = s.accept()
    clients.append(con)
    print('client_{} connected: {}'.format(idx, addr))

    print(port, host)
    print(host)
    print("Client is connected...")
```

Εικόνα 13: Ορισμός socket και δημιουργία πολλών clients

Ο server λαμβάνει τα attributes που του στέλνει ο client μέσω της συνάρτησης recv() και δημιουργεί ένα αρχείο με το περιεχόμενο που έλαβε, το attributes.txt. Προχωρά κρατώντας το χρόνο για τη δεδομένη στιγμή με τη συνάρτηση time.time() και τον μετατρέπει σε string. Αυτό χρησιμεύει στο επόμενο βήμα όπου λαμβάνεται το περιεχόμενο του αρχείου attributes.txt και τα δύο string ενώνονται σε ένα νέο για να περαστούν αμέσως μετά σε ένα νέο αρχείο το attr_withtime.txt. Με αυτό τον τρόπο, το αρχείο περιλαμβάνει τα attributes αλλά περιέχει και μια χρονική στιγμή T η οποία δείχνει πότε έλαβε τα attributes ο server και πότε θα δημιουργήσει το κλειδί. Ο χρόνος είναι μοναδικός για κάθε κλειδί και το κλειδί δεν μπορεί να χρησιμοποιηθεί μετά την παρέλευσή του.

Τα δεδομένα που περιέχει το attr_withtime.txt, ανατίθενται σε μία νέα μεταβλητή έστω Str ώστε να χρησιμοποιηθούν σαν είσοδος για την εντολή που θα ακολουθήσει.

```
36      file1 = "attributes.txt"
37      file = open(file1, "wb")
38      RecvData = con.recv(8192)
39      file.write(RecvData)
40      file.close()
41      print("\n Attributes have been received successfully. \n")
42
43      T1 = time.time()
44      print(T1)
45      time1 = int (T1)
46      t1 = str(time1)
47      T = "T=%s" % t1
48
49      with open (file1,'r') as file:
50          stry = file.read()
51
52      attr_witht = "%s|%s" % (stry,T)
53      print("Attributes with time: " , attr_witht)
54
55      file = open("attr_withtime.txt",'w')
56      file.write(attr_witht)
57      file.close()
58
59      with open ("attr_withtime.txt",'r') as file:
60          Str = file.read()
61      Str = Str.rstrip("\n")
62      print(Str)
```

Εικόνα 14: Δημιουργία τυχαίων attributes με χρόνο

Έχοντας πλέον την είσοδο χρησιμοποιείται το oabe_keygen για να παραχθεί το κλειδί, στο οποίο περνούν τιμές στα ορίσματα. Στο -s διευκρινίζεται το σχήμα κρυπτογράφησης CP, στο -i η είσοδος, το Str, δηλαδή τα attributes βάσει των οποίων θα γίνει η παραγωγή του κλειδιού και στο -o το αρχείο εξόδου vehicleCP που περιλαμβάνει το κλειδί που παράγεται.

Ο server στέλνει τα κλειδιά του vehicleCP.key στον client μέσω της συνάρτησης send() για να το χρησιμοποιήσει αργότερα για την αποκρυπτογράφηση.

```
64      cmd = 'oabe_keygen -s CP -p org1 -i "'+Str+'" \-o vehicleCP'
65      p = Popen(cmd, shell=True, stdin=PIPE, stdout=PIPE, stderr=STDOUT, close_fds=True)
66      output = p.stdout.read()
67      print(output)
68
69      print("Sending vehicleCP to client")
70
71      file = open("vehicleCP.key", "rb")
72      SendData1 = file.read(8192)
73      con.send(SendData1)
```

Εικόνα 15: Παραγωγή κλειδιών

Ταυτόχρονα, δημιουργείται και ένα ακόμα αρχείο, το org1.mpk.cpabe που περιέχει το master public key. Επειδή το master public key είναι αναγκαίο για την κρυπτογράφηση που πραγματοποιεί η ενδιάμεση οντότητα messenger, ο server αντιγράφει το αρχείο στο φάκελο που βρίσκεται το messenger.py για να μπορέσει να το χρησιμοποιήσει όταν έρθει η ώρα.

Τέλος, στέλνει το κλειδί που περιέχεται στο org1.mpk.cpabe, στον client για να το χρησιμοποιήσει στην αποκρυπτογράφηση.

```
75      con, addr = s.accept()
76
77      shutil.copy2('/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/s/org1.mpk.cpabe' ,
78                   '/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/m/')
79
80      print("Sending master public key to client.")
81
82      file = open("org1.mpk.cpabe", "rb")
83      SendData2 = file.read(8192)
84      con.send(SendData2)
```

Εικόνα 16: Αντιγραφή του master public key στο shared storage

4.3.5. Client

Ο client είναι το όχημα, ο ρόλος του οποίου είναι να στέλνει attributes στον server για να παράγει κλειδιά και να αποκρυπτογραφεί το ciphertext που έχει δημιουργήσει το messenger αν ικανοποιεί τα κριτήρια. Είναι ουσιαστικά ο χρήστης του συστήματος που κατασκευάζεται και υλοποιείται στο αρχείο client.py. Η διαδικασία που ακολουθείται παρατίθεται αναλυτικά παρακάτω.

Αρχικά, όπως και στις προηγούμενες οντότητες πρώτα εισάγονται τα απαραίτητα modules.

```
#!/usr/bin/env python

from subprocess import Popen, PIPE, STDOUT
from time import sleep
import threading
import time
import shutil
from threading import *
import os
import math
import sys
import socket
```

Εικόνα 17: Modules του client

Στη συνέχεια, δημιουργείται ένα thread. Σε αυτό, ο client συνδέεται με το beacon μέσω της συνάρτησης s.connect() και ζητάει το τυχαίο string από αυτό. Αφού λάβει το string μέσω της recv() κάνει συνένωσή του με τα attributes και περνά τα δεδομένα στο αρχείο cabx.txt.

```
56          def func1():
57
58              s = socket.socket()
59              host = socket.gethostname()
60              port = 6565
61              s.connect((host,port))
62              print("Connected to beacon... ")
63
64              message = "Give me a string."
65              s.send(message.encode())
66
67              random_string = s.recv(1024).decode()
68              print('String received from beacon: ' + random_string)
69
70              attribute = "Cab"
71              rand_attributes = "%s|%s" % (attribute,random_string)
72
73              file = open("cabx.txt",'w')
74              file.write(rand_attributes)
75              file.close()
76              s.close()
```

Εικόνα 18: Δημιουργία socket, αίτηση για string και λήψη του

```
186         thread1 = threading.Thread(target = func1)
187         thread1.start()
188         thread1.join()
```

Εικόνα 19: Thread

Συνδέεται μετά με τον server και στέλνει τα δεδομένα του cabx.txt σε αυτόν για να προχωρήσει στην παραγωγή κλειδιών.

```
80              s = socket.socket()
81              host = socket.gethostname()
82              port = 9898
83              s.connect((host,port))
84              print("Connected to server... ")
85              timestamp = time.time()   ###calculate time
86              timestamp = str(timestamp)
87
88              print("Sending attributes to server...", threading.currentThread().getName())
89              file = open("cabx.txt", "rb")
90              SendData = file.read(8192)
91              s.send(SendData)
92              SendData = file.read(8192)
93              print("\n Atrributes have been sent successfully. \n", threading.currentThread().getName())
```

Εικόνα 20: Αποστολή δεδομένων στον server

Λαμβάνει το ιδιωτικό κλειδί που του στέλνει πίσω ο server και ταυτόχρονα λαμβάνει και το master public key καθώς και αυτό δέχεται αλλαγές και είναι απαραίτητο για το στάδιο της αποκρυπτογράφησης.

```
key1 = "vehicleCP.key"
file = open(key1, "wb")
RecvData1 = s.recv(8192)
file.write(RecvData1)
file.close()
print("\n Personal key has been received successfully. \n", threading.currentThread().getName())
size1 = os.stat('vehicleCP.key').st_size
print("SIZE1:", size1)

s.close()
sleep(2)
s = socket.socket()
s.connect((host,port))

masterkey = "org1.mpk.cpabe"
file = open(masterkey, "wb")
RecvData2 = s.recv(8192)
file.write(RecvData2)
file.close()
print("\n Master public key has been received successfully. \n", threading.currentThread().getName())
```

Εικόνα 21: Λήψη κλειδιών από τον server

Μέσα στο func1 καλείται η συνάρτηση decrypt για να κάνει αποκρυπτογράφηση. Πρώτα γίνεται ένας έλεγχος αν υπάρχει αρχείο να αποκρυπτογραφήσει. Η αποκρυπτογράφηση γίνεται αφού πρώτα ο client κάνει έλεγχο, πάρει το ciphertext ενός από τα αρχεία που βρίσκονται στο shared storage και το περάσει στο δικό του αρχείο input.cpabe. Μετά, χρησιμοποιεί το oabe_dec όπου περνά τα παρακάτω ορίσματα: στο -s το σχήμα κρυπτογράφησης CP, στο -k το κλειδί vehicleCP.key, στο -i δίνει ως είσοδο το ciphertext μέσω του αρχείου input.cpabe και στο -o το αρχείο εξόδου plainOK.input.txt οποίο περιέχει το μήνυμα που ανακτάται.

```
145         def decrypt():
146
147             path = '/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/ss'
148             n = 1
149             while True:
150                 try:
151                     with open (path+"/%s.cpabe" % n,"rb") as file:
152                         print("%s.cpabe" % n)
153                         strx = file.read()
154                 except:
155                     break
156
157                 file = open("input.cpabe",'w')
158                 file.write(strx)
159                 file.close()
```

Εικόνα 22: Έλεγχος ύπαρξης αρχείου για αποκρυπτογράφηση και χρήση του

```
176                 cmd = 'oabe_dec -s CP -p org1 -k vehicleCP.key -i input.cpabe \-o plainOK.input.txt'
177                 p = Popen(cmd, shell=True, stdin=PIPE, stdout=PIPE, stderr=STDOUT, close_fds=True)
178                 output = p.stdout.read()
179                 print(output)
180                 print("\n Decryption done. \n")
181
182                 n += 1
```

Εικόνα 23: Αποκρυπτογράφηση

5. Αποτελέσματα

5.1. Dataset

Για τη προσομοίωση του παραπάνω συστήματος και για να γίνει αντιληπτό πώς θα λειτουργούσε στον πραγματικό κόσμο χρησιμοποιήθηκε ένα σύνολο δεδομένων (dataset). Το dataset περιέχει την κίνηση από ταξί που κυκλοφορούσαν στην περιοχή του Κόλπου του Σαν Φρανσίσκο, στην Καλιφόρνια τον Μάιο του 2008. Τα δεδομένα αυτά παρέχονται από το μουσείο επιστήμης, τεχνολογίας και τεχνών του Σαν Φρανσίσκο, το Exploratorium.

Κάθε ταξί διαθέτει μια συσκευή εντοπισμού τοποθεσίας GPS και όλα τα δεδομένα τοποθεσίας από κάθε ταξί αποστέλλονται σε έναν σταθμό. Από εκεί στέλνονται σε υπολογιστές οι οποίοι κρατούν τα δεδομένα αυτά σε πραγματικό χρόνο, μαζί με τον αριθμό του ταξί και το αν είναι κατειλημμένο (Piorkowski et al. & CRAWDAD, 2009).

Το dataset που είναι διαθέσιμο περιέχει ένα αρχείο .txt με όλα τα ταξί για τα οποία έχουν κρατηθεί τα δεδομένα κινητικότητας καθώς και πόσες εγγραφές έχει κάθε ταξί. Τα δεδομένα παρέχονται στην εξής μορφή:

```
<cab id="enyenewl" updates="2414"/>
<cab id="ockoac" updates="8215"/>
<cab id="ikdagcy" updates="18109"/>
<cab id="ayshekki" updates="21343"/>
<cab id="ancorjo" updates="20867"/>
<cab id="idvowwed" updates="27414"/>
<cab id="iatmeuns" updates="14345"/>
<cab id="unwrain" updates="5282"/>
<cab id="atsfiv" updates="4336"/>
<cab id="aupclik" updates="5681"/>
<cab id="afmorc" updates="19783"/>
<cab id="ochotcil" updates="15608"/>
<cab id="enkkand" updates="29560"/>
<cab id="amwibs" updates="22895"/>
<cab id="ibflsruc" updates="25422"/>
```

Εικόνα 24: Συνολικό αρχείο με ταξί

Δίνεται το μοναδικό αναγνωριστικό κάθε ταξί με το id και το πόσες φορές έχει ενημερωθεί το αρχείο για το αντίστοιχο ταξί, με το update.

Στη συνέχεια, για κάθε ένα από τα ταξί του παραπάνω αρχείου υπάρχει ένα διαφορετικό αρχείο .txt ASCII που περιέχει όλες τις εγγραφές για το συγκεκριμένο ταξί με τη μορφή που ακολουθεί. Οι πρώτες δύο τιμές είναι οι γεωγραφικές συντεταγμένες και ειδικότερα η πρώτη τιμή είναι το γεωγραφικό πλάτος και η δεύτερη το γεωγραφικό μήκος. Η τρίτη τιμή δείχνει το αν το ταξί είναι κατειλημμένο, με τιμή 0 αν είναι ελεύθερο και 1 αν δεν είναι. Η τέταρτη είναι ο χρόνος, η χρονική στιγμή για την ακρίβεια που το ταξί βρέθηκε στις συγκεκριμένες συντεταγμένες σε μορφή UNIX epoch.

```
37.75134 -122.39488 0 1213084687
37.75136 -122.39527 0 1213084659
37.75199 -122.3946 0 1213084540
37.7508 -122.39346 0 1213084489
37.75015 -122.39256 0 1213084237
37.75454 -122.39227 0 1213084177
37.75901 -122.3925 0 1213084172
37.77053 -122.39788 0 1213084092
37.77669 -122.39382 0 1213084032
37.78194 -122.38844 0 1213083971
37.78999 -122.38909 0 1213083910
37.79728 -122.39609 0 1213083855
37.79838 -122.40239 0 1213083811
37.79779 -122.40647 0 1213083736
37.79779 -122.40646 1 1213083715
```

Εικόνα 25: Αρχείο ενός ταξί

Το dataset αυτό λοιπόν θα αξιοποιηθεί για να εφαρμοστεί στην πραγματικότητα η υλοποίηση που έχει πραγματοποιηθεί.

Αφού πλέον έχει γίνει η περιγραφή όλων των επιμέρους κομματιών απομένει ένα τελευταίο κομμάτι το οποίο πρέπει να υλοποιηθεί για να μπορεί να σταθεί επαρκώς η συγκεκριμένη υλοποίηση στον πραγματικό κόσμο. Όπως έχει ήδη αναφερθεί υπάρχει το beacon το οποίο παράγει και εκπέμπει κλειδιά. Επομένως, η λογική οδηγεί στη σκέψη ότι δεν γίνεται να μπορεί να εκπέμψει κλειδιά και να μπορεί να τα λάβει ένα όχημα που βρίσκεται στη διπλανή πόλη ή ακόμα και πολύ μακριά από το beacon. Αυτό ισχύει γιατί για να γίνει η μεταξύ τους επικοινωνία θα απαιτείται περισσότερος χρόνος και δε θα υπάρχει αμεσότητα και γρήγορη ανταπόκριση, χαρακτηριστικά που χρειάζονται σε ένα σύστημα επικοινωνίας τέτοιου είδους.

Για το λόγο αυτό, πρέπει σε κάθε όχημα να γίνεται έλεγχος της απόστασης από το beacon έτσι ώστε αν αυτή ξεπερνάει ένα ορισμένο όριο να μην μπορεί να λάβει κλειδιά από το beacon. Αυτό επιτυγχάνεται με τη χρήση του dataset.

Συγκεκριμένα, στον client διαβάζεται το αρχείο κάθε οχήματος, αντιστρέφει τις γραμμές για να βρίσκεται στην αρχή το σημείο από το οποίο ξεκίνησε το όχημα και χρησιμοποιούνται οι συντεταγμένες που έχει. Ορίζονται οι συντεταγμένες του beacon και μετά υπολογίζεται η απόσταση distance των συντεταγμένων του οχήματος από αυτές του beacon. Αυτό γίνεται για κάθε γραμμή του αρχείου και συνεπώς για κάθε σημείο που βρίσκεται το όχημα κατά τη διάρκεια της κίνησής του. Υπολογίζεται βέβαια και ο χρόνος που κάνει το όχημα για να βρεθεί από το ένα σημείο στο άλλο. Έπειτα, ορίζεται μια απόσταση dist = 0.1 μέτρα. Αν το distance είναι μικρότερο από το dist τότε το όχημα είναι μέσα στα όρια της ακτίνας εκπομπής του beacon και μπορεί να ζητήσει το τυχαίο string που χρειάζεται από το beacon και να εκτελέσει τις υπόλοιπες εντολές. Διαφορετικά αν βρίσκεται μακριά από το beacon ελέγχει την επόμενη τοποθεσία του οχήματος.

Παρακάτω φαίνεται πως υλοποιείται η διαδικασία που εξηγήθηκε:

```
18  dir = "/home/niki/Downloads/libopenabe-1.0.0-src/libopenabe-1.0.0/cli/cab"
19
20  fp = open(os.path.join(dir, filename))
21  lines = fp.readlines()
22  lines.reverse()
23  n = len(lines)
24  for i in range(n-1):
25      line = lines[i]
26      nextline = lines[i+1]
27
28      tokens = line.split(" ")
29      x = tokens[0]
30      y = tokens[1]
31      t = int(tokens[3])
```

Εικόνα 26: Ανάγνωση αρχείου

```
33      tokens = nextline.split(" ")
34      nt = int(tokens[3])
35      timediff = nt-t
36
37      x1 = float(x)
38      y1 = float(y)
39
40      x2 = 37.75200
41      y2 = -122.39500
42      dist = 0.1
43
44      distance = math.sqrt(((x2 - x1)**2) + ((y2 - y1)**2))
45      print("Distance = ", distance)
46
47      latitude = str(x1)
48      longitude = str(y1)
49      xx, yy = filename1.split('.')
50      zz,cab_id = xx.split('_')
51
52      if distance <= dist:
53          print("Ask for strings and keys.")
```

Εικόνα 27: Έλεγχος απόστασης

```
187         print("Wait for new location for {} secs".format(timediff))
188         sleep(timediff)
189     else:
190         print("Cab is far from the beacon.")
191         print("Wait for new location for {} secs".format(timediff))
192         sleep(timediff)
```

Εικόνα 28: Έλεγχος απόστασης

Πλέον λοιπόν, έχει ολοκληρωθεί η υλοποίηση του συστήματος κρυπτογράφησης και μπορεί να αξιολογηθεί ως προς διάφορα χαρακτηριστικά του.

5.2. Λειτουργία Συστήματος

Μετά την υλοποίησή του το σύστημα μπορεί να λειτουργήσει κανονικά. Για να παρατηρηθεί ο τρόπος λειτουργίας του, εκτελείται στο τερματικό, όπου πρέπει όλα τα αρχεία να τρέχουν παράλληλα. Πρώτα εκτελείται ο server, μετά το beacon, μετά ο client και μετά ο messenger και τρέχουν ταυτόχρονα. Στον client μαζί με το αρχείο client.py παρέχω και το όνομα του αρχείου του οχήματος που πρέπει να ελεγχθεί γι' αυτό και στον κώδικα υπάρχουν και οι ακόλουθες γραμμές που διαβάζουν το argument.

```
14  argument_list = sys.argv
15  filename = sys.argv[1]
16  print(filename)
17  filename1 = str(filename)
```

Εικόνα 29: Ανάγνωση argument

Καθώς πραγματοποιείται προσομοίωση σε πραγματικό χρόνο όσο αυτό είναι δυνατό, κρίθηκε αναγκαίο το πρόγραμμα να εκτελεστεί για είκοσι clients – οχήματα. Με αυτό τον τρόπο υπάρχει μια ποικιλία αποτελεσμάτων από τα οποία στη συνέχεια επιλέχθηκαν τα πιο ευδιάκριτα από αυτά. Κάθε όχημα εκτελέστηκε για μία ώρα έτσι ώστε να παράγει αποτελέσματα για αρκετές συντεταγμένες στις οποίες βρίσκεται το όχημα και να φανεί μια λεπτομερής μετακίνηση αυτού.

Για κάθε όχημα συγκεντρώθηκαν μερικά στατιστικά. Για το μέγεθος των αρχείων χρησιμοποιήθηκε η συνάρτηση size = os.stat('file').st_size όπου χρειάστηκε να μετρηθεί το μέγεθος των δεδομένων, ενώ για το timestamp χρησιμοποιήθηκε η συνάρτηση timestamp = time.time() όπου χρειάστηκε να υπολογιστεί η χρονική στιγμή.

Έτσι, δημιουργήθηκαν δύο αρχεία. Το ένα είναι το key_management το οποίο έχει cab id (το αναγνωριστικό του ταξί-οχήματος), timestamp (η στιγμή που γίνεται η σύνδεση με τον central authority), μέγεθος αρχείων που στέλνει το όχημα, μέγεθος αρχείων που λαμβάνει το όχημα, γεωγραφικό πλάτος και γεωγραφικό μήκος. Ενδεικτικά:

Cab id	Timestamp	Send Size	Receive size	Latitude(γεωγραφικό πλάτος)	Longitude(γεωγραφικό μήκος)
abboip	1611550687	14	3768	37.75153	-122.39447
abboip	1611550773	14	3768	37.75149	-122.39447
abboip	1611550835.89	14	3768	37.75149	-122.39447
abboip	1611550899.61	14	3768	37.75149	-122.39446
abboip	1611552471.67	14	3768	37.75144	-122.39449
abboip	1611552550.18	14	3768	37.75151	-122.39453
abboip	1611552617.63	14	3768	37.75137	-122.39502
abboip	1611552680.75	14	3768	37.75139	-122.39498
abboip	1611552748.58	14	3768	37.7514	-122.39495

Εικόνα 30: Δεδομένα αρχείου key_management

Το δεύτερο αρχείο που δημιουργήθηκε είναι το msg_management το οποίο έχει cab id (το αναγνωριστικό του ταξί-οχήματος), timestamp (στιγμή που βρίσκει το input.cpabe),

μέγεθος αρχείων που αποκρυπτογραφεί, γεωγραφικό πλάτος και γεωγραφικό μήκος. Για παράδειγμα:

Cab id	Timestamp	decrypt file size	Latitude(γεωγραφικό πλάτος)	Longitude(γεωγραφικό μήκος)
abboip	1611550838.49	6218	37.75149	-122.39447
abboip	1611550902.26	6218	37.75149	-122.39446
abboip	1611550902.39	6414	37.75149	-122.39446
abboip	1611550838.42	6610	37.75149	-122.39447
abboip	1611550902.19	6610	37.75149	-122.39446
abboip	1611550902.32	6610	37.75149	-122.39446
abboip	1611550902.46	6610	37.75149	-122.39446
abboip	1611550838.38	6810	37.75149	-122.39447
abboip	1611550838.45	6810	37.75149	-122.39447

Εικόνα 31: Δεδομένα αρχείου msg_management

Από τα στοιχεία αυτά, θα μελετηθεί το μέγεθος αρχείων ενώ επίσης θα μετρηθεί και θα μελετηθεί ο αριθμός επικοινωνιών του κάθε οχήματος με το central authority – server. Πιο συγκεκριμένα, για να υπολογιστεί ο αριθμός επικοινωνιών του κάθε client με τον server υπολογίζεται πότε έγινε επικοινωνία με τον server σε σύγκριση με τη στιγμή που άρχισε να τρέχει ο client. Αφού βρεθεί αυτή η διαφορά διαιρείται η τιμή με το εξήντα και βρίσκεται το πεντάλεπτο που ξεκίνησε την επικοινωνία του ο client. Τα πεντάλεπτα της μίας ώρας είναι δώδεκα. Το σύνολο των επικοινωνιών σε κάθε πεντάλεπτο είναι ο αριθμός επικοινωνιών του κάθε client με τον server.

Όσον αφορά το μέγεθος των αρχείων, στο key_management υπολογίζεται το μέγεθος των αρχείων που στέλνει το όχημα σε κάθε πεντάλεπτο, ενώ στο msg_management υπολογίζεται το μέγεθος των αρχείων που αποκρυπτογραφεί το όχημα σε κάθε πεντάλεπτο.

Έτσι λοιπόν, στα διαγράμματα του key_management υπάρχουν δύο διαγράμματα για κάθε όχημα τα οποία απεικονίζουν το πλήθος επικοινωνιών ανά πεντάλεπτο και τα απεσταλμένα δεδομένα ανά πεντάλεπτο αντίστοιχα. Ομοίως, υπάρχουν δύο ίδια διαγράμματα για κάθε client και στο msg_management με μόνη διαφορά τα δεδομένα.

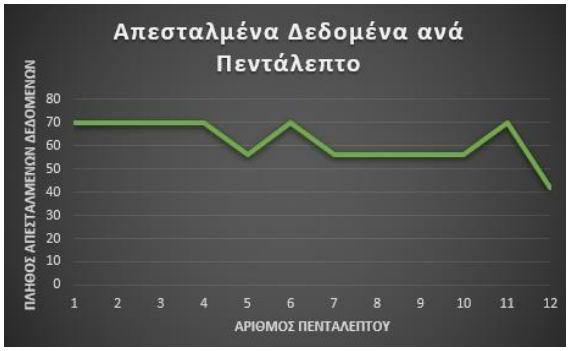
5.3. Αποτελέσματα

Στη συνέχεια, αναλύονται τα αποτελέσματα που προέκυψαν. Παρουσιάζονται αρχικά τα διαγράμματα του key_management.

Διάγραμμα 1: Επικοινωνίες 1ου οχήματος

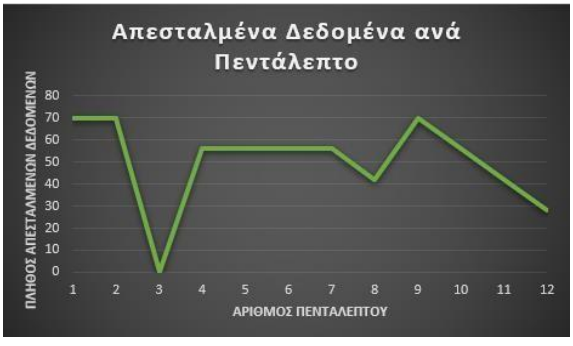
Διάγραμμα 2: Δεδομένα 1ου οχήματος

Διάγραμμα 3: Επικοινωνίες 2ου οχήματος



Διάγραμμα 4: Δεδομένα 2ου οχήματος

Διάγραμμα 5: Επικοινωνίες 3ου οχήματος



Διάγραμμα 6: Δεδομένα 3ου οχήματος

Διάγραμμα 7: Επικοινωνίες 4ου οχήματος

Διάγραμμα 8: Δεδομένα 4ου οχήματος

Διάγραμμα 9: Επικοινωνίες 5ου οχήματος

Διάγραμμα 10: Δεδομένα 5ου οχήματος

Τα διαγράμματα ανά δύο ανήκουν σε διαφορετικό όχημα. Το πρώτο που μπορεί να παρατηρηθεί είναι ότι το διάγραμμα επικοινωνιών ανά πεντάλεπτο είναι ίδιο με το διάγραμμα των απεσταλμένων δεδομένων ανά πεντάλεπτο, κάτι το οποίο είναι αναμενόμενο καθώς τα απεσταλμένα δεδομένα είναι πολλαπλάσιο του πλήθους επικοινωνιών.

Επιπλέον, σε κάθε πεντάλεπτο διαφέρει το πλήθος επικοινωνιών και συνήθως δεν παρατηρείται κάποιο μοτίβο.

Όπως φαίνεται στο πρώτο όχημα (Διαγράμματα 1,2), το όχημα επικοινωνεί με το central authority στο πρώτο πεντάλεπτο και στέλνει δεδομένα, αλλά στα επόμενα τρία δεν υπάρχει καμία επικοινωνία μέχρι το πέμπτο πεντάλεπτο από όπου οι επικοινωνίες αρχίζουν να αυξάνονται σταδιακά. Αυτό συμβαίνει καθώς το όχημα κινείται στο χώρο και επομένως κάποιες φορές απομακρύνεται από το beacon πέρα από την προκαθορισμένη απόσταση. Επομένως, απευθείας δεν πραγματοποιείται επικοινωνία ούτε στέλνονται δεδομένα, αλλά προχωρά ο έλεγχος στο επόμενο ζευγάρι συντεταγμένων.

Από την άλλη, το δεύτερο όχημα (Διαγράμματα 3,4), έχει μια διαρκή σχεδόν σταθερή επικοινωνία με μικρές αυξομειώσεις στο πλήθος των επικοινωνιών και αντίστοιχα στο πλήθος των απεσταλμένων δεδομένων. Παρατηρείται ότι από την αρχή μέχρι το τέταρτο πεντάλεπτο έχει το ίδιο πλήθος επικοινωνιών (πέντε) οι οποίες στη συνέχεια μεταβάλλονται. Αυτό ενισχύει την προηγούμενη άποψη ότι στο πλήθος των επικοινωνιών δεν παρατηρείται κάποιο μοτίβο. Δεν σημαίνει δηλαδή πως αν στα τέσσερα πρώτα πεντάλεπτα υπήρχαν πέντε επικοινωνίες τόσες θα υπάρχουν και στα επόμενα.

Χαρακτηριστική είναι και η περίπτωση των επικοινωνιών στο τρίτο πεντάλεπτο του τρίτου οχήματος (Διαγράμματα 5,6), στο οποίο πέφτουν ραγδαία, για να αυξηθούν ξανά αμέσως μετά στο τέταρτο πεντάλεπτο.

Τέλος, είναι εμφανές και το γεγονός ότι η περίπτωση να μην υπάρχει καμία επικοινωνία είναι απολύτως τυχαία, καθώς εξαρτάται εξολοκλήρου από το που κινείται το όχημα και το πόσο μπορεί να απομακρυνθεί από το beacon και για πόση ώρα. Έτσι, στο πρώτο όχημα η επικοινωνία χάνεται στην αρχή της ώρας (δεύτερο πεντάλεπτο), στο τέταρτο όχημα χάνεται στη μέση της ώρας (πέμπτο μέχρι ένατο πεντάλεπτο), ενώ στο πέμπτο όχημα χάνεται προς το τέλος της ώρας (ένατο μέχρι δέκατο πεντάλεπτο).

Παρακάτω παρουσιάζονται τα διαγράμματα του msg_management.

Διάγραμμα 11: Επικοινωνίες στο 1ο όχημα

Διάγραμμα 12: Δεδομένα στο 1ο όχημα

Διάγραμμα 13: Επικοινωνίες στο 2ο όχημα

Διάγραμμα 14: Δεδομένα στο 2ο όχημα

Διάγραμμα 15: Επικοινωνίες στο 3ο όχημα

Διάγραμμα 16: Δεδομένα στο 3ο όχημα

Διάγραμμα 17: Επικοινωνίες στο 4ο όχημα

Διάγραμμα 18: Δεδομένα στο 4ο όχημα

Διάγραμμα 19: Επικοινωνίες στο 5ο όχημα

Διάγραμμα 20: Δεδομένα στο 5ο όχημα

Στο msg_management τα διαγράμματα ανήκουν επίσης στο ίδιο όχημα ανά δύο. Η διαφορά των παρόντων διαγραμμάτων με τα προηγούμενα έγκειται στο γεγονός ότι τα αρχεία msg_management έχουν πολύ μεγαλύτερο πλήθος δεδομένων αφού κάθε φορά ο client έψαχνε μέσα στο shared storage το αρχείο που μπορούσε να αποκρυπτογραφήσει, προσπαθώντας όμως να αποκρυπτογραφήσει και όλα τα προηγούμενα αρχεία μέχρι να φτάσει σε αυτό. Για το λόγο αυτό στο αρχείο διατηρούνται και στοιχεία από όλα τα αρχεία που προσπαθούσε να αποκρυπτογραφήσει και από όλες τις επικοινωνίες που έκανε για να το πετύχει αυτό. Έτσι, το πλήθος των δεδομένων είναι πολύ μεγαλύτερης κλίμακας.

Επιπροσθέτως, μεταξύ των key_management και του msg_management παρατηρείται συνήθως μια γενική ομοιότητα που μπορεί να βοηθήσει να αναγνωριστεί σε ποιον client ανήκει κάθε διάγραμμα με κύριο και ίδιο χαρακτηριστικό τα σημεία στα οποία δεν υπάρχει καθόλου επικοινωνία.

Όσον αφορά το δεύτερο όχημα (Διαγράμματα 13,14), παρατηρείται μια σταθερή ανοδική πορεία και στο πλήθος επικοινωνιών και στα δεδομένα που αποκρυπτογραφούνται και φαίνεται ότι διατηρείται μια σταθερή επικοινωνία. Όσο περνά η ώρα, τόσο περισσότερα αρχεία ελέγχονται για αποκρυπτογράφηση μέχρι να βρεθεί το επιθυμητό, επομένως αυξάνεται το συνολικό μέγεθος των αρχείων.

Βέβαια, όπως φαίνεται σε άλλα διαγράμματα στην περίπτωση που το όχημα βρίσκεται μακριά από το beacon και δεν ακολουθηθεί η συνήθης διαδικασία για να υπάρχει επικοινωνία, τότε τα δεδομένα είναι μηδενικά, αφού δεν ελέγχονται αρχεία για αποκρυπτογράφηση. Χαρακτηριστικά είναι τα διαγράμματα του πρώτου, του τέταρτου και του πέμπτου οχήματος όπου τα δεδομένα μηδενίζονται σε διάφορες χρονικές στιγμές.

Όπως προαναφέρθηκε, γενικά όσο περνά η ώρα υπάρχουν περισσότερα αρχεία που έχουν δημιουργηθεί για να ελεγχθούν αν είναι δυνατόν να αποκρυπτογραφηθούν, επομένως ελέγχεται μεγαλύτερο μέγεθος αρχείων. Αυτό ωστόσο δεν αποτελεί και τον κανόνα καθώς το μέγεθος των αρχείων ποικίλλει. Για παράδειγμα, αν σε κάποια περίπτωση το συνολικό μέγεθος των αρχείων είναι μικρότερο από ότι στο προηγούμενο πεντάλεπτο θα παρατηρηθεί πτώση της καμπύλης του διαγράμματος.

Συνολικά, από όλα τα διαγράμματα παρατηρείται μια φυσιολογική ροή πληροφοριών και το πρόγραμμα δεν παρουσιάζει δυσκολίες στην εκτέλεσή του ούτε περιέχει δεδομένα που δεν μπορούν να εξηγηθούν.

6. Συμπεράσματα

Μέσα από την έρευνα αυτή έγινε αντιληπτή η σημασία του attribute – based encryption στο Internet of Things και στην προστασία των δεδομένων που κάθε συσκευή IoT περιλαμβάνει, αποθηκεύει και ανταλλάσσει. Η χρήση του attribute – based encryption στα σύγχρονα έξυπνα οχήματα αποτελεί μεγάλο βήμα στην προστασία και των δεδομένων που αξιοποιούνται αλλά και των χρηστών. Η εφαρμογή του με τη χρήση μιας οντότητας που παράγει συνεχώς νέα κλειδιά και το γεγονός ότι ένα όχημα μπορεί να κάνει αποκρυπτογράφηση μόνο αν βρίσκεται κοντά σε αυτή την οντότητα και έχει περάσει συγκεκριμένος χρόνος από όταν έγινε η κρυπτογράφηση, μεγιστοποιούν την ασφάλεια που παρέχει.

Συγχρόνως, τα αποτελέσματα της μελέτης δείχνουν ότι καλύπτονται επαρκώς οι στόχοι που τέθηκαν και είναι ενθαρρυντικά. Από την προσομοίωση φαίνεται πως ημέθοδος μπορεί να εφαρμοστεί στον πραγματικό κόσμο χωρίς ιδιαίτερη δυσκολία, καθώς το σύστημα δεν εμφάνισε προβλήματα στην κλίμακα που εφαρμόστηκε. Τα αποτελέσματα δείχνουν μια φυσιολογική πορεία των επικοινωνιών χωρίς να υπάρχουν υπερβολές και ανεξήγητα φαινόμενα.

Σε μια εφαρμογή σε πραγματική πόλη βέβαια, το πλήθος δεδομένων και επικοινωνιών θα αφορά μια πολύ μεγάλη κλίμακα αλλά με κατάλληλες προσαρμογές και ρυθμίσεις θα μπορούσε να ανταπεξέλθει επαρκώς στις ανάγκες του Internet of Things και να αποτελέσει έναν ευρέως χρησιμοποιούμενο τρόπο κρυπτογράφησης και αποκρυπτογράφησης που θα διατηρεί τα δεδομένα κάθε έξυπνου οχήματος και των χρηστών του ασφαλή, μακριά από τους κινδύνους που υπάρχουν.

6. Βιβλιογραφία

Παπαζώης, Π. (2019). *Ασφάλεια στο Διαδίκτυο των Πραγμάτων* [Διπλωματική Εργασία, Πανεπιστήμιο Αιγαίου]. Ιδρυματικό Αποθετήριο HELLANICUS. http://hdl.handle.net/11610/18539

Παπασταθοπούλου, Α. (2017). *Internet of Things* [Διπλωματική Εργασία, Πανεπιστήμιο Μακεδονίας]. ΨΗΦΙΔΑ, Ψηφιακή Βιβλιοθήκη και Ιδρυματικό Αποθετήριο του Πανεπιστημίου Μακεδονίας. http://dspace.lib.uom.gr/handle/2159/20157

Tzafestas, S. (2018). Ethics and Law in the Internet of Things World. *Smart Cities, 1*(1),98–120. https://doi.org/10.3390/smartcities1010006

Cloud Computing. (2021, January 5). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Cloud_computing&oldid=998471329

Fog Computing. (2020, December 29). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Fog_computing&oldid=997028043

Κλαδίσιος, Λ. (2019). *Διαδίκτυο Των Πραγμάτων* [Διπλωματική Εργασία, Πανεπιστήμιο Πειραιώς]. Ιδρυματικό Αποθετήριο Διώνη του Πανεπιστημίου Πειραιώς http://dione.lib.unipi.gr/xmlui/handle/unipi/12649

Hung, M. (Ed.). (2017). *Leading the IoT: Gartner Insights on How to Lead in a Connected World*. Gartner. https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

Framingham, Mass. (2019, June 18). *The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast*. IDC. https://www.idc.com/getdoc.jsp?containerId=prUS45213219

Κανονισμός (ΕΕ) 2016/679. *Προστασία των φυσικών προσώπων έναντι της επεξεργασίας των δεδομένων προσωπικού χαρακτήρα και για την ελεύθερη κυκλοφορία των δεδομένων αυτών και την κατάργηση της οδηγίας 95/46/ΕΚ (Γενικός Κανονισμός για την Προστασία Δεδομένων).* Ευρωπαϊκό Κοινοβούλιο, Συμβούλιο της Ευρωπαϊκής Ένωσης. https://eur-lex.europa.eu/legal-content/EL/TXT/PDF/?uri=CELEX:32016R0679&qid=1610125918861&from=EN

Nieles, M., Dempsey, K., & Yan Pillitteri, V. (2017). Introduction: Important Terminology. In *An Introduction to Information Security* (σσ. 2, 3). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-12r1.pdf

Industrial Espionage. (2020, December 27). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Industrial_espionage&oldid=996656960

Merriam-Webster. (n.d.). Activist. In Merriam-Webster.com dictionary. Retrieved January 5, 2021, from https://www.merriam-webster.com/dictionary/activist

National Cyber Security Centre. (2020). *Cyber attack*. UK. https://www.ncsc.gov.uk/section/advice-guidance/all-topics?allTopics=true&topics=cyber%20attack&sort=date%2Bdesc

Παπανικολάου, Α. (2009). *Η προστασία της πνευματικής ιδιοκτησίας και των πνευματικών δικαιωμάτων σύμφωνα με τη σύμβαση για το κυβερνοέγκλημα του Συμβουλίου της Ευρώπης* [Διδακτορική Διατριβή, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης]. ΙΚΕΕ Βιβλιοθήκη Αριστοτέλειου Πανεπιστημίου Θεσσαλονίκης http://ikee.lib.auth.gr/record/111020/?ln=en

Cambridge Dictionary. (n.d.). Hacking. In Cambridge Dictionary.org. Retrieved December 28, 2020, from https://dictionary.cambridge.org/dictionary/english/hacking

Phishing. (2021, January 5). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Phishing&oldid=998517717

Cisco Systems, Inc. (2020). What is Malware?. U.S. https://www.cisco.com/c/en/us/products/security/advanced-malware-protection/what-is-malware.html

Identity Theft. (2020, December 29). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Identity_theft&oldid=997054230

Βογιατζής, Γ. (2010). *Πειρατεία πνευματικών έργων στο περιβάλλον του διαδικτύου* [Διπλωματική Εργασία, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης]. ΙΚΕΕ Βιβλιοθήκη Αριστοτέλειου Πανεπιστημίου Θεσσαλονίκης http://ikee.lib.auth.gr/record/115796/?ln=en

Ευρωπαϊκό Ελεγκτικό Συνέδριο. (2019). *Προκλήσεις για μια αποτελεσματική ενωσιακή πολιτική για την κυβερνοασφάλεια* (Ενημερωτικό Έγγραφο). Ε.Ε. Ευρωπαϊκή Ένωση. https://www.eca.europa.eu/Lists/ECADocuments/BRP_CYBERSECURITY/BRP_CYBERSECURITY_EL.pdf

Deloitte Central Mediterranean, & ΣΕΒ. (2020, Ιούλιος). Κυβερνοασφάλεια: Παρατηρητήριο Ψηφιακού Μετασχηματισμού ΣΕΒ. https://www2.deloitte.com/content/dam/Deloitte/gr/Documents/risk/gr_SEV_Deloitte_Cybersecurity_noexp.pdf

Cryptography. (2021, January 5). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Cryptography&oldid=998505251

National Cyber Security Centre. (2020). *Cryptography*. UK. https://www.ncsc.gov.uk/section/advice-guidance/all-topics?allTopics=true&topics=cryptography&sort=date%2Bdesc

Ζάχος, Ε., Παγουρτζής, Α., & Γροντάς, Π. (2015). Εισαγωγή: Ιστορική Αναδρομή. Στο *Υπολογιστική Κρυπτογραφία* (σσ. 15, 16). Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. http://hdl.handle.net/11419/5439

Delfs, H., & Knebl, H. (2007). Introduction: The Objectives of Cryptography. In *Introduction to Cryptography: Principles and Applications* (5th ed., pp. 2-3). Springer.

Stallings, W. (2012). *Κρυπτογραφία και Ασφάλεια Δικτύων: Αρχές και Εφαρμογές*. (1η ελληνική έκδοση). (σσ. 32, 33, 36, 284, 287, 288, 294). Εκδόσεις Ίων.

Attribute-based encryption. (2020, December 23). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Attribute-based_encryption&oldid=995922365

Sahai, A., & Waters, B. (2005). Fuzzy Identity-Based Encryption. In Cramer R. (Eds.), *Lecture Notes in Computer Science: Vol 3494. Advances in Cryptology – EUROCRYPT 2005* (pp. 457-473). Springer. https://doi.org/10.1007/11426639_27

Goyal, V., Pandey, O., Sahai, A., & Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the 13th ACM conference*

on Computer and communications security (CCS 'O6). Association for Computing Machinery, New York, NY, USA, 89-98. https://doi.org/10.1145/1180405.1180418

Bethencourt, J., Sahai, A., & Waters, B. (2007, May 20-23). *Ciphertext-Policy Attribute-Based Encryption* [Conference Presentation]. 2007 IEEE Symposium on Security and Privacy (SP '07), Berkeley, CA. https://doi.org/10.1109/SP.2007.11

Bilinear map. (2020, October 17). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Bilinear_map&oldid=983939299

Ostrovsky, R., Sahai, A., & Waters, B. (2007). Attribute-based encryption with non-monotonic access structures. *Proceedings of the 14th ACM conference on Computer and communications security (CCS '07). Association for Computing Machinery, New York, NY, USA*, 195–203. https://doi.org/10.1145/1315245.1315270

Chase, M. (2007). Multi-authority Attribute Based Encryption. In Vadhan S.P. (Eds.), *Lecture Notes in Computer Science: Vol 4392. Theory of Cryptography* (pp. 515-534). Springer. https://doi.org/10.1007/978-3-540-70936-7_28

Ni, J., Zhang, K., Lin, X., & Shen, X. (2018). Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys & Tutorials, 20*(1), 601-628. https://doi.org/10.1109/COMST.2017.2762345

Yao, X., Chen, Z., & Tian, Y. (2015). A lightweight attribute-based encryption scheme for the Internet of Things. *Future Generation Computer Systems, 49*, 104-112. https://doi.org/10.1016/j.future.2014.10.010

Li, J., Zhang, Y., Ning, J., Huang, X., Poh, G. S., & Wang, D. (2020) Attribute Based Encryption with Privacy Protection and Accountability for CloudIoT. *IEEE Transactions on Cloud Computing, 8.* https://doi.org/10.1109/TCC.2020.2975184

Wang, X., Zhang, J., Schooler, E. M., & Ion M. (2014, June 10-14). *Performanceevaluation of Attribute-Based Encryption: Toward data privacy in the IoT* [Conference Presentation]. 2014 IEEE International Conference on Communications (ICC), Sydney, NSW. https://doi.org/10.1109/ICC.2014.6883405

Liang, X., Lu, R., Lin, X., & Shen X. (2011). *Ciphertext Policy Attribute Based Encryption with Efficient Revocation.* University of Waterloo. http://bbcr.uwaterloo.ca/~x27liang/abe_with_revocation.pdf

Attrapadung, N., & Imai H. (2009) Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes. In Parker M.G. (Eds.), *Lecture Notes in Computer Science: Vol 5921. Cryptography and Coding* (pp. 278-300). Springer. https://doi.org/10.1007/978-3-642-10868-6_17

Liu, J. K., Yuen, T. H., Zhang, P., & Liang, K. (2018). Time-based direct revocable Ciphertext-Policy Attribute-Based Encryption with short revocation list. In B. Preneel B., & Vercauteren F. (Eds.), *Lecture Notes in Computer Science: Vol. 10892. Applied Cryptography and Network Security ACNS 2018* (pp. 516-534). Springer. https://doi.org/10.1007/978-3-319-93387-0_27

Jiang, M., Wang, H., Zhang, W., Qin, H., & Sun, X. (2020). Location-based data access control scheme for Internet of Vehicles. *Computers & Electrical Engineering, 86*. https://doi.org/10.1016/j.compeleceng.2020.106716

Zeutro LLC: Encryption & Data Security. (2018). Openabe Library. U.S. https://github.com/zeutro/openabe

Ubuntu. (2021, January 2). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Ubuntu&oldid=997851390

Python (programming language). (2021, January 5). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=998436019

Python Software Foundation (2020). Applications for Python. U.S. https://www.python.org/about/apps/

Μαγκούτης, Κ., & Νικολάου, Χ., (2015). Δικτυακός Προγραμματισμός. Στο Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python (σσ. 254-256). Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. http://hdl.handle.net/11419/1708

Piorkowski, M., Sarafijanovic-Djukic, N., & Grossglauser M. (2009) *Traceset of mobility data of taxi cabs in San Francisco, USA* (CRAWDAD dataset epfl/mobility; Version 2009-02-24) [Data set]. https://crawdad.org/epfl/mobility/20090224/cab https://doi.org/10.15783/C7J010