Ανάλυση και Σχεδίαση Προγραμμάτων Διαχείρισης Κλειδαρίθμων

(Analysis and Design of Password Managers)

ΓΕΩΡΓΙΟΣ Α. ΜΑΛΑΚΑΤΣΟΠΟΥΛΟΣ
(Georgios A. Malakatsopoulos)

Μεταπτυχιακή Διατριβή
(Master's Thesis)

Φεβρουάριος 2015

Επιβλέπων Καθηγητής:
Ιωάννης Μούντανος

Μέλη Επιτροπής:
Γεώργιος Σταμούλης
Νέστωρ Ευμορφόπουλος

Μεταπτυχιακό Πρόγραμμα Σπουδών
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Πολυτεχνική Σχολή, Πανεπιστήμιο Θεσσαλίας, Βόλος, Ελλάδα

Program of Graduate Studies
Department of Electrical and Computer Engineering
School of Engineering, University of Thessaly, Volos, Greece

Η παρούσα Μεταπτυχιακή Διατριβή του μεταπτυχιακού φοιτητή Γεωργίου Μαλακατσόπουλου (Geom@inf.uth.gr και GeorgeMTech@gmail.com) εγκρίθηκε την 26η Φεβρουαρίου 2015 από την τριμελή εξεταστική επιτροπή του Προγράμματος Μεταπτυχιακών Σπουδών, του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας, Βόλου, Ελλάδας αποτελούμενη από:

The current MSc Dissertation of the postgraduate student Georgios (George) Malakatsopoulos (Geom@inf.uth.gr and GeorgeMTech@gmail.com), was approved on 26th of February 2015 by the three-membered examination committee of the Program of Graduate Studies, Department of Electrical and Computer Engineering, School of Engineering, University of Thessaly, Volos, Greece consisting of:

1) Αναπληρωτής Καθηγητής Δρ. Μούντανος Ιωάννης /
Associate Professor Dr. Moondanos Ioannis
(jmoondan@inf.uth.gr)

……
(υπογραφή / signature)

2) Καθηγητής Δρ. Σταμούλης Γεώργιος /
Professor Dr. Stamoulis Georgios
(georges@inf.uth.gr)

……
(υπογραφή / signature)

3) Επίκουρος Καθηγητής Δρ. Ευμορφόπουλος Νέστωρ /
Assistant Professor Dr. Eumorfopoulos Nestoras
(nestevmo@inf.uth.gr)

……
(υπογραφή / signature)

« Αφιερωμένη για ακόμη μία φορά στους γονείς μου, αλλά
και στους αδελφικούς μου φίλους Άγγελο, Αλέξανδρο, Άρη και Ευτύχη
που με στήριξαν όλα αυτά τα χρόνια των σπουδών μου και μου έδωσαν την
απαραίτητη ενέργεια και θέληση για να προχωρήσω και να φτάσω
στο σημερινό μου (μορφωτικό και μη) επίπεδο »

“ Once again dedicated to my parents, as well as my brotherly friends Angelos, Alexandros, Aris and Eftychis, who have supported me in all the years of my studies and gave me the energy and willingness required to go ahead and reach my current educational state (and not only that) ”

ΠΕΡΙΛΗΨΗ

Ένας Διαχειριστής Κλειδαρίθμων (Password Manager) είναι ένα πρόγραμμα το οποίο χρησιμοποιείται για την συγκέντρωση και την αποθήκευση των διαπιστευτηρίων πρόσβασης και εξουσιοδότησης ενός χρήστη, όπως για παράδειγμα, των ονομάτων χρήστη (usernames) του σε συνδυασμό με τους κωδικούς πρόσβασής του (passwords), των προσωπικών αριθμών αναγνώρισής του (PINs), των αριθμών των πιστωτικών καρτών του (credit card numbers) κ.α.. Τα διαπιστευτήρια αυτά μπορούν να οργανωθούν σε κατηγορίες μέσα στο πρόγραμμα για ευκολότερη αναζήτηση και η έκτασή τους μπορεί να είναι, θεωρητικά, απεριόριστη (οριοθετείται μόνο από το μέσο αποθήκευσης).

Εκτός της εύκολης διαχείρισης, τα προγράμματα αυτά χρησιμοποιούνται και λόγω της εκτεταμένης προστασίας που προσφέρουν. Είναι γεγονός ότι οι χρήστες μπορούν πλέον να χρησιμοποιούν πολύ πιο ισχυρούς και μεγαλύτερου μήκους κωδικούς πρόσβασης απ ότι συνήθιζαν να χρησιμοποιούν στο παρελθόν (αφού δε χρειάζεται πλέον να τους θυμούνται απ έξω), αυξάνοντας έτσι το επίπεδο της ασφάλειάς τους, ενώ παράλληλα μπορούν να αισθάνονται και πιο σίγουροι για το μέρος στο οποίο αποθηκεύουν τα διαπιστευτήριά τους, αφού αυτά βρίσκονται σε κρυπτογραφημένη μορφή μέσα σε μια βάση δεδομένων, η οποία διατηρείται πάντα «κλειδωμένη» και είναι προσβάσιμη μόνο από τον χρήστη ο οποίος είναι εξουσιοδοτημένος να το κάνει.

Είναι όμως τα πράγματα τόσο απλά και ευοίωνα όσο ακούγονται; Υπάρχουν κίνδυνοι που προκύπτουν από τον σχεδιασμό ή από την χρήση των Διαχειριστών Κλειδαρίθμων; Την απάντηση σε αυτά τα ερωτήματα καλείται να δώσει αυτή η διατριβή, η οποία αναλύει εκτενώς τα θέματα ασφαλείας αυτών των προγραμμάτων και προτείνει λύσεις που μπορούν να χρησιμοποιηθούν ανά περίπτωση. Σχεδιάζεται επίσης και υλοποιείται, ειδικά για την διατριβή αυτή ένας Διαχειριστής Κλειδαρίθμων, ο οποίος παρουσιάζει σε πρακτικό επίπεδο τους κινδύνους ασφαλείας που προκύπτουν καθώς και τις αντίστοιχες μεθόδους για την αποτροπή τους.

Λέξεις κλειδιά: διαχειριστής κλειδαρίθμων, κωδικοί πρόσβασης, ασφάλεια προσωπικών δεδομένων, κρυπτογράφηση, GM Password Manager.

ABSTRACT

A Password Manager is a program which is used for the aggregation and storing of the user access and authorization credentials, such as, Usernames with the related Passwords, Personal Identification Numbers (PINs), Credit Card Numbers, etc.. These credentials can be organized into a variety of categories inside this program to facilitate the search process; its range can be (theoretically) unlimited (it only depends on the storage' medium capacity).

Besides the fact of easy managing, these programs are also known for the extended protection they offer. It is a fact, that users are able to use much longer and stronger passwords than they used to in the past (since there is no need to memorize them), by increasing in this way their security level. Moreover, this program is highly reliable since the credentials are stored encrypted inside a database which remains always locked and is accessible for reading only by users with the appropriate authorization.

However, by reading this description, some inquiries may arise: How simple can be such a program and is it as promising as it is sounded? Are there any dangers arise from the design process or the use of Password Managers? The answers for the above questions are included into this M.Sc. dissertation, which analyzes on a great scale the security issues related to these programs and suggests solutions which can be used on each case. Also, specifically for this dissertation needs, it was designed and developed a Password Manager, called "GM Password Manager", which presents in a practical way the safety dangers which can be occurred as well as the appropriate methods to prevent them.

Keywords: GM Password Manager, credentials, private data security, encryption.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 2 - Επιθέσεις, Υποκλοπή Δεδομένων και Τρόποι Προστασίας

ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή στους Διαχειριστές Κλειδαρίθμων

ΕΙΣΑΓΩΓΗ

Το θέμα αυτής της διατριβής είναι η ανάλυση και η σχεδίαση προγραμμάτων Διαχείρισης Κλειδαρίθμων (Password Managers). Ο κύριος σκοπός αυτών των προγραμμάτων είναι η συγκέντρωση, η αποθήκευση και η διαχείριση προσωπικών δεδομένων πρόσβασης των χρηστών με ασφάλεια.

Ένας χρήστης μιας συσκευής (όπως σταθερού ή φορητού υπολογιστή, έξυπνου κινητού τηλεφώνου, κ.α.) χρησιμοποιεί στην καθημερινότητά του δεκάδες προσωπικά δεδομένα πρόσβασης για την πρόσβασή του σε διάφορες ηλεκτρονικές υπηρεσίες. Για παράδειγμα, κάποιος μπορεί να χρησιμοποιήσει το όνομα χρήστη (username) και τον κωδικό πρόσβασής του (password) για την πρόσβασή του στον λογαριασμό ηλεκτρονικού ταχυδρομείου του. Κάποιος άλλος (ή ο ίδιος) μπορεί να χρησιμοποιήσει αντίστοιχα στοιχεία για την πρόσβασή του σε ιστοσελίδες ηλεκτρονικής τραπεζικής, για την πρόσβασή του στο προφίλ του σε ένα κοινωνικό δίκτυο κ.α.. Μπορεί επίσης να χρησιμοποιήσει και αριθμούς από τις πιστωτικές του κάρτες για την αγορά προϊόντων μέσα από ηλεκτρονικά καταστήματα, όπως και για να την κράτηση εισιτηρίων (αεροπορικών, ακτοπλοϊκών, κ.α.). Τέλος, μπορεί να πληκτρολογήσει (με φυσικό τρόπο) αριθμούς κλειδώματος PINs σε διάφορες συσκευές όπως στα ATMs των τραπεζών, σε συστήματα συναγερμών στο σπίτι ή στο γραφείο του, σε χρηματοκιβώτια κ.ο.κ..

Αυτό σημαίνει ότι πρέπει να θυμάται «απ' έξω» όλα τα παραπάνω διαπιστευτήρια για να μπορέσει να τα χρησιμοποιήσει ή να τα έχει σημειωμένα κάπου έτσι ώστε να τα ανακαλεί όποτε χρειάζεται. Το σημείο που κάποιος καταγράφει τα διαπιστευτήριά του παίζει πολύ μεγάλο ρόλο για την ασφάλειά του. Αν υποθέσουμε ότι ο χρήστης τα καταγράφει χειρόγραφα σε ένα προσωπικό σημειωματάριο που έχει πάντα μαζί του, τότε δημιουργείται ένα τεράστιο κενό ασφαλείας: οποιοσδήποτε από το κοντινό του περιβάλλον (συνεργάτης στη δουλειά, μέλος της οικογενείας του, συγγενής, φίλος) μπορεί σε ανύποπτη χρονική στιγμή να υποκλέψει το σημειωματάριο αυτό, να διαβάσει τα διαπιστευτήριά του, να αντιγράψει και έπειτα να το επιστρέψει στην αρχική του θέση άθικτο, χωρίς ο ιδιοκτήτης του να αντιληφθεί το παραμικρό. Ακόμη χειρότερο γίνεται το σενάριο αν το σημειωματάριο αυτό χαθεί κατά λάθος από τον ιδιοκτήτη του και πέσει στα χέρια οποιουδήποτε ανθρώπου. Σε αυτήν την περίπτωση ο οποιοσδήποτε εκτός του ιδιοκτήτη μπορεί να εκμεταλλευτεί αυτά τα δεδομένα προς όφελός του, παριστάνοντας ότι αυτός είναι ο νόμιμος κάτοχός τους και εκτελώντας συναλλαγές βάση αυτών.

Το απλό αυτό προβληματικό σενάριο, όπως και άλλα πιο σύνθετα σενάρια, έρχονται να λύσουν οι Διαχειριστές Κλειδαρίθμων. Χρησιμοποιώντας τους Διαχειριστές Κλειδαρίθμων για την αποθήκευση των διαπιστευτηρίων μας και τη χρήση τους μέσα από αυτούς είμαστε προστατευμένοι από πάρα πολλούς κινδύνους που έχουν να κάνουν με την υποκλοπή δεδομένων και την ασφάλειά μας.

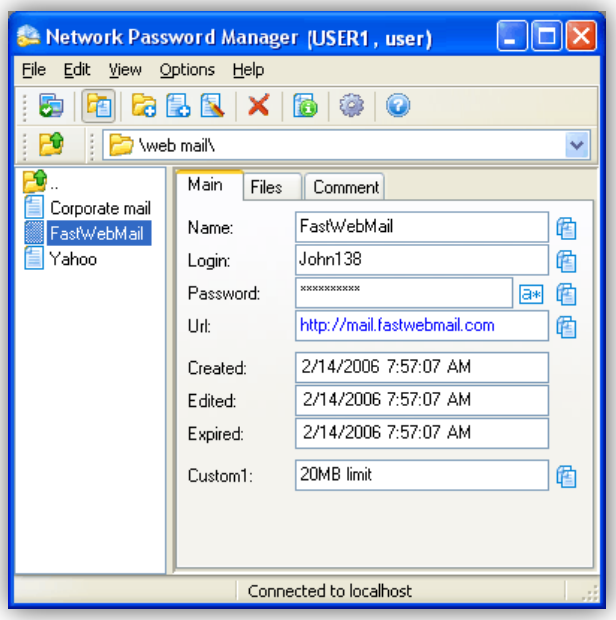
1.1 Τι είναι οι Διαχειριστές Κλειδαρίθμων

Οι Διαχειριστές Κλειδαρίθμων (Password Managers) είναι προγράμματα τα οποία όπως αναφέρει και το όνομά τους, διαχειρίζονται κλειδαρίθμους, δηλαδή, κωδικούς ξεκλειδώματος (ή αλλιώς, κωδικούς πρόσβασης). Στα προγράμματα αυτά οι χρήστες καταχωρούν τα προσωπικά τους δεδομένα πρόσβασης, ή αλλιώς τα διαπιστευτήρια πρόσβασής τους (όπως ονόματα χρήστη, κωδικούς πρόσβασης, αριθμούς κλειδώματος PIN, αριθμούς πιστωτικών καρτών, κ.α.) ούτως ώστε να τα εξασφαλίσουν από τυχών υποκλοπή ή μη εξουσιοδοτημένη πρόσβαση.

Τυπικά, ένας διαχειριστής κλειδαρίθμων αποτελείται από μία γραφική διασύνδεση χρήστη και από μια βάση δεδομένων. Με τον όρο «γραφική διασύνδεση χρήστη» εννοούμε ότι το πρόγραμμα λειτουργεί σε περιβάλλον που χρησιμοποιεί γραφικά (όπως τα λειτουργικά συστήματα Windows, Linux, Android κ.α.) και χρησιμοποιεί «παράθυρα» για την εκτέλεση των λειτουργιών του, όπως την καταχώρηση, την ανάγνωση και την επεξεργασία πληροφοριών σε - και από - αυτό. Η βάση δεδομένων από την άλλη αποτελεί ένα αρχείο δεδομένων στο οποίο ο Διαχειριστής Κλειδαρίθμων αποθηκεύει τα διαπιστευτήρια του χρήστη.

Η διαδικασία καταχώρησης των διαπιστευτηρίων διαφέρει από πρόγραμμα σε πρόγραμμα, αλλά είναι συνήθως απλή. Γενικά, την πρώτη φορά που θα εκτελεστεί ο Διαχειριστής Κλειδαρίθμων ζητάει από τον χρήστη να δημιουργήσει έναν νέο, «κύριο» κωδικό πρόσβασης (master password) και ένα όνομα χρήστη (username), τα οποία θα χρησιμοποιούνται από την επόμενη φορά για την είσοδο στο πρόγραμμα και την πιστοποίηση και εξουσιοδότηση του χρήστη σχετικά με την ανάγνωση και την επεξεργασία των αποθηκευμένων δεδομένων.

Αφού δημιουργηθούν τα διαπιστευτήρια εισόδου, στη συνέχεια το πρόγραμμα εμφανίζει το κύριο παράθυρο των λειτουργιών του μαζί με ένα μενού επιλογών. Μέσα από το μενού, ο χρήστης μπορεί να επιλέξει την προσθήκη μίας νέας καταχώρησης, να πληκτρολογήσει τα διαπιστευτήριά του στα κατάλληλα πλαίσια της φόρμας εισαγωγής δεδομένων και, πατώντας το πλήκτρο ΟΚ, να τα αποθηκεύσει στη βάση δεδομένων. Κλείνοντας το πρόγραμμα διενεργείται (συνήθως) κρυπτογράφηση της βάσης δεδομένων ενώ ανοίγοντάς το ξανά διενεργείται αποκρυπτογράφηση, και η λίστα με τα διαπιστευτήρια εμφανίζεται μόνο αν ο κύριος κωδικός πρόσβασης σε συνδυασμό με το όνομα χρήστη που έχουν εισαχθεί είναι σωστά.



Εικόνα 1.1 - Φόρμα εισαγωγής διαπιστευτηρίων για την προσθήκη μίας νέα καταχώρησης σε έναν Διαχειριστή Κλειδαρίθμων.

Κατά αυτόν τον τρόπο, ένας χρήστης που θέλει να αποκτήσει ασφαλή πρόσβαση στον λογαριασμό του σε μία ιστοσελίδα ή σε μία υπηρεσία, ανοίγει τον Διαχειριστή Κλειδαρίθμων, τοποθετεί τον κύριο κωδικό πρόσβασής του και το όνομα χρήστη του (τα οποία είναι τα μόνα διαπιστευτήρια που πρέπει να θυμάται απ έξω), εντοπίζει την καταχώρηση που θέλει να χρησιμοποιήσει, αντιγράφει τα δεδομένα που θέλει στο πρόχειρο (clipboard) και τα επικολλά στα αντίστοιχα πλαίσια πιστοποίησης της ιστοσελίδας ή της υπηρεσίας που θέλει να πραγματοποιήσει είσοδο.

Πολλοί Διαχειριστές Κλειδαρίθμων επίσης, λειτουργούν και ως **αυτόματοι «συμπληρωτές φορμών» (form fillers)**, δηλαδή συμπληρώνουν αυτόματα σε μία φόρμα μιας ιστοσελίδας τον κωδικό πρόσβασης και το όνομα χρήστη, τη στιγμή που ο χρήστης πλοηγηθεί στην συγκεκριμένη ιστοσελίδα, ούτως ώστε αυτός να μπορεί να πιστοποιηθεί και να συνδεθεί ταχύτατα. Αυτή η δυνατότητα παρέχεται στον χρήστη μέσω μιας επέκτασης που βρίσκεται εγκατεστημένη στο πρόγραμμα περιήγησης (browser extension) και που επικοινωνεί με τον Διαχειριστή Κλειδαρίθμων και την βάση δεδομένων του για την άντληση των διαπιστευτηρίων.

Συνεπώς, χρησιμοποιώντας Διαχειριστές Κλειδαρίθμων για την αποθήκευση των διαπιστευτηρίων, είναι πολύ δύσκολο από κάποιον (έως ανέφικτο) να τελέσει υποκλοπή, αφού ο μόνος τρόπος για να το κάνει είναι μόνο αν γνωρίζει τον κύριο κωδικό πρόσβασης και το όνομα χρήστη που χρησιμοποιήθηκε. Το μέγεθος της ασφάλειας βέβαια έγκειται κατά ένα μεγάλο ποσοστό και στον χρήστη που χρησιμοποιεί τον Διαχειριστή και πιο συγκεκριμένα στο πόσο ισχυρό κωδικό πρόσβασης εισόδου έχει επιλέξει. Αν για παράδειγμα έχει επιλέξει την ημερομηνία γέννησής του ως τον κύριο κωδικό

πρόσβασης του προγράμματος αντί ενός πιο ισχυρού, που θα μπορούσε να περιέχει γράμματα, αριθμούς και σύμβολα, τότε ο ίδιος κάνει πιο εύκολη τη δουλειά των επιτιθέμενων, οι οποίοι προσπαθούν να ανακαλύψουν τον κωδικό αυτόν.

Ως εκ τούτου, με χρήση ενός ισχυρού κύριου κωδικού πρόσβασης, ο χρήστης ακόμα και να χάσει τον φορητό του υπολογιστή ή το κινητό του τηλέφωνο στα οποία βρίσκεται αποθηκευμένη η βάση δεδομένων του Διαχειριστή Κλειδαρίθμων, είναι αδύνατον να αναγνωστούν και να υποκλαπούν τα διαπιστευτήρια από τρίτους, αφού αυτά βρίσκονται σε κρυπτογραφημένη μορφή.

1.2 Λόγοι που οδήγησαν στην κατασκευή Διαχειριστών Κλειδαρίθμων, Κατηγορίες & Πλεονεκτήματα

Ξεκινώντας την ανάλυσή μας από την τεχνική πιστοποίησης των χρηστών μέσω «κωδικού πρόσβασης» και «ονόματος χρήστη», η οποία χρησιμοποιείται ευρέως από προγράμματα, ιστοσελίδες, λειτουργικά συστήματα, μηχανήματα κ.α., μπορούμε να διακρίνουμε πολλά πλεονεκτήματα από τη χρήση της.

Αρχικά, η τεχνική αυτή μπορεί εύκολα να ενσωματωθεί στις περισσότερες εφαρμογές χρησιμοποιώντας τις «Διεπαφές Προγραμματισμού Εφαρμογών» (Application Programming Interfaces - APIs) της εκάστοτε εφαρμογής (όπου υπάρχει η δυνατότητα). Συνεχίζοντας, πλεονέκτημα αποτελεί το γεγονός ότι δεν απαιτεί μεγάλες τροποποιήσεις (έως και καμία) στον υπολογιστή ή στον εξυπηρετητή που πρόκειται να χρησιμοποιηθεί, όπως επίσης και το γεγονός ότι οι χρήστες είναι ήδη εξοικειωμένοι με τη χρήση κωδικών πρόσβασης και γενικά, με τις μεθόδους πιστοποίησης αυτού του είδους.

Βέβαια, αν και η τεχνική αυτή με σωστό προγραμματισμό μπορεί να αποβεί σε μία εξαιρετικά ασφαλή μέθοδο πιστοποίησης, υπάρχει μια μεγάλη αδυναμία στον τρόπο με τον οποίο οι χρήστες επιλέγουν τους κωδικούς πρόσβασής τους και στον τρόπο που τους διαχειρίζονται. Γενικά, έχει παρατηρηθεί ότι οι κωδικοί πρόσβασης που επιλέγονται από τους χρήστες δεν είναι ιδιαίτερα ισχυροί και ανήκουν σε μία τουλάχιστον από τις παρακάτω κατηγορίες:

- **Απλοί κωδικοί:** μικροί σε μήκος (4 - 6 χαρακτήρες) οι οποίοι χρησιμοποιούν λέξεις που μπορούν να βρεθούν σε λεξικά, χωρίς να είναι αναμειγμένοι με κάποιον χαρακτήρα διαφορετικού είδους (όπως νούμερα, σημεία στίξης, μικρά/κεφαλαία γράμματα). Με λίγα λόγια πρόκειται για ανίσχυρους κωδικούς, εύκολο να τους ανακαλύψει κάποιος

(από μόνος του ή μέσω προγράμματος που θα δοκιμάζει διάφορες λέξεις από οποιοδήποτε λεξικό).

- **Κωδικοί γραμμένοι σε σημείο που μπορεί εύκολα να εντοπιστεί:** όπως κωδικοί γραμμένοι χειρόγραφα σε χαρτάκια τα οποία βρίσκονται κολλημένα πάνω στις οθόνες, αιωρούνται πάνω στο γραφείο, βρίσκονται σε σημειώματα καρφιτσωμένα σε πίνακες υπενθυμίσεων ή κολλημένα στις πόρτες των ψυγείων, κωδικοί γραμμένοι χειρόγραφα σε κάποιο προσωπικό σημειωματάριο, κωδικοί γραμμένοι ηλεκτρονικά, μη-κρυπτογραφημένοι σε ένα απλό αρχείο κειμένου (με εύκολα προβλέψιμο όνομα), σημειωμένοι σε μορφή SMS μηνυμάτων ή επαφών σε ένα κινητό τηλέφωνο, κ.τ.λ..

- **Επαναλαμβανόμενοι κωδικοί:** κάποιοι χρήστες επινοούν μία φορά μόνο έναν κωδικό πρόσβασης και χρησιμοποιούν τον ίδιο για πάντα (για λόγους ευκολίας αποστήθισης) ή για αρκετά χρόνια. Τον κωδικό αυτόν τον χρησιμοποιούν για την πρόσβαση σε όλες τις ιστοσελίδες στις οποίες έχουν δημιουργήσει λογαριασμό, σε εφαρμογές, στον υπολογιστή τους, κ.τ.λ..

- **Κοινοποιούμενοι κωδικοί:** κάποιοι χρήστες διαμοιράζονται με άλλους τους προσωπικούς κωδικούς πρόσβασής τους, τους κοινοποιούν άφοβα, χωρίς δεύτερη σκέψη για τις συνέπειες που μπορεί να επιφέρει αυτή η πράξη, στέλνουν μη-κρυπτογραφημένα emails τα οποία περιέχουν κωδικούς πρόσβασης, κ.τ.λ..

- **Κοινόχρηστοι κωδικοί:** οι διαχειριστές του δικτύου μιας εταιρίας επιτρέπουν στους χρήστες της ίδιας ομάδας εργασίας να χρησιμοποιούν τους ίδιους κωδικούς πρόσβασης.

Έχει παρατηρηθεί ότι ένα μεγάλο πλήθος χρηστών επιλέγει μία ή περισσότερες από τις παραπάνω κατηγορίες για τη δημιουργία, την αποθήκευση και των χρήση των κωδικών πρόσβασής τους. Αυτό επωφελεί ιδιαιτέρως τους εισβολείς (hackers), τους ανακτητές κωδικών (crackers), τα κακόβουλα λογισμικά (malware) και τους κυβερνο-κλέφτες (cyber thieves) στο να εισβάλλουν σε ατομικούς λογαριασμούς χρηστών, εταιριών (ανεξαρτήτου μεγέθους), κυβερνητικών οργανισμών, ιδρυμάτων κ.α..

Η προστασία των χρηστών ενάντια στις παραπάνω αδυναμίες οδήγησαν στην εφεύρεση των Διαχειριστών Κλειδαρίθμων και τους έκαναν ιδιαίτερα σημαντικούς στη χρήση τους. Στη σημερινή τους μορφή, οι Διαχειριστές Κλειδαρίθμων διακρίνονται σε έξι, συχνά συνδυαζόμενες, κατηγορίες:

- **Σταθεροί:** διαχειριστές κλειδαρίθμων οι οποίοι εγκαθίστανται σε σταθερούς ή φορητούς υπολογιστές (desktop PCs, laptops, netbooks) και αποθηκεύουν τα διαπιστευτήρια των χρηστών τους στον σκληρό δίσκο του υπολογιστή.

- **Αυτόνομοι:** διαχειριστές κλειδαρίθμων οι οποίοι τρέχουν απευθείας (χωρίς εγκατάσταση) ως αυτόνομα προγράμματα και αποθηκεύουν τα δεδομένα τους σε κινητές συσκευές όπως έξυπνα κινητά τηλέφωνα (smartphones), ταμπλέτες (tablets), PDAs ή που τρέχουν ως αυτόνομες εφαρμογές μέσα από μονάδες μνήμης USB και αποθηκεύουν τα διαπιστευτήριά τους εκεί.

- **Διαδικτυακοί:** διαχειριστές κλειδαρίθμων οι οποίοι είναι ανεπτυγμένοι ως εφαρμογές σε ιστοσελίδες του παγκόσμιου ιστού, μέσω των οποίων οι χρήστες, χρησιμοποιώντας τον κύριο κωδικό πρόσβασής τους και το όνομα χρήστη τους, μπορούν να εμφανίσουν, να αποθηκεύσουν και να επεξεργαστούν τα διαπιστευτήρια τους, ενώ μέσω της διαδικασίας αντιγραφής/επικόλλησης μπορούν να τα χρησιμοποιήσουν εκεί που θέλουν.

- **Επεκτάσεις περιηγητών:** διαχειριστές κλειδαρίθμων οι οποίοι έχουν κατασκευασθεί για να λειτουργούν ως επέκταση (extension, plugin) των προγραμμάτων περιήγησης του παγκόσμιου ιστού (web browsers). Τα προγράμματα αυτά λειτουργούν μόνο όταν ο περιηγητής είναι ανοικτός (τρέχει) και συνήθως έχουν τη μορφή εργαλειοθήκης (toolbar). Η βάση δεδομένων τους μπορεί να αποθηκεύεται τοπικά, στον φάκελο εγκατάστασης του περιηγητή ή απομακρυσμένα, σε κάποιον εξυπηρετητή του προγράμματος. Επίσης τα διαπιστευτήρια στη βάση δεδομένων μπορεί να είναι να αποθηκεύονται σε κρυπτογραφημένη ή μη-κρυπτογραφημένη μορφή.

- **Με πιστοποίηση πολλαπλών κριτηρίων:** τα διαπιστευτήρια των χρηστών, εκτός από τον κύριο κωδικό πρόσβασης του χρήστη και το όνομα χρήστη, προστατεύονται επιπλέον και από εξωτερικές μεθόδους πιστοποίησης, όπως για παράδειγμα οι εξωτερικοί κωδικοί ασφαλείας (security tokens) οι οποίοι παρέχονται συνήθως από τις τράπεζες για την έγκριση ηλεκτρονικών συναλλαγών και παράγονται από ειδικές συσκευές όπως οι extra PIN generators, security token generators, κ.α.. Με αυτόν τον τρόπο παρέχουν πιστοποίηση με κριτήρια παραπάνω του ενός, αφού απαιτούν επιπλέον των βασικών, «κάτι το οποίο ο χρήστης έχει στην κατοχή του» (π.χ. έξυπνη κάρτα, μονάδα ασφαλούς πιστοποίησης USB, κ.α.), «κάτι που μόνο ο νόμιμος χρήστης γνωρίζει» (αριθμός κλειδώματος PIN, extra PIN, κ.α.) ή «κάτι που μόνο ο νόμιμος χρήστης διαθέτει» (χρήση βιομετρικών χαρακτηριστικών όπως: αποτύπωμα δακτύλου, αποτύπωμα χεριού, ανατομία αμφιβληστροειδή, ανατομία προσώπου, κ.α.).

- **Βασιζόμενοι σε κέντρα δεδομένων:** διαχειριστές κλειδαρίθμων των οποίων τα δεδομένα αποθηκεύονται σε κέντρα δεδομένων (datacenters, cloud) και η διαχείρισή τους γίνεται από τοπική εφαρμογή, εγκατεστημένη στον υπολογιστή του χρήστη. Επιβάλλεται η ενεργή σύνδεση στο Διαδίκτυο.

Εικόνα 1.2 - Στη φωτογραφία εικονίζονται οι συσκευές παραγωγής εξωτερικών κωδικών ασφαλείας (safe code generators) (πάνω αριστερά και κάτω στο κέντρο), η μονάδα ασφαλούς πιστοποίησης USB (πάνω, κέντρο) και η έξυπνη κάρτα (πάνω δεξιά).

Οι Διαχειριστές Κλειδαρίθμων **μπορούν επίσης να χρησιμοποιηθούν και σαν μηχανισμοί άμυνας** κατά των φαινομένων εξαπάτησης «ηλεκτρονικού ψαρέματος» (phishing) και της «εσκεμμένης ανακατεύθυνσης» (pharming).

Ως «ηλεκτρονικό ψάρεμα» ορίζεται η διενέργεια εξαπάτησης των χρηστών του διαδικτύου, κατά την οποία ο θύτης (επιτιθέμενος) υποδύεται μία αξιόπιστη οντότητα και εκμεταλλευόμενος την ελλιπή προστασία που παρέχουν τα ηλεκτρονικά εργαλεία και την άγνοια του θύματος (του χρήστη), αποκτά με αθέμιτο τρόπο τα προσωπικά δεδομένα του θύματος. Με απλά λόγια, σε μια επίθεση ηλεκτρονικού ψαρέματος ο επιτιθέμενος φτιάχνει μία πλαστή ιστοσελίδα, όσο το δυνατόν ίδια με κάποια άλλη σημαντική ιστοσελίδα (όπως μιας τράπεζας) και με διάφορους τρόπους (με κάποιο e-mail για παράδειγμα) πείθει τον χρήστη να συνδεθεί σε αυτή αντί για την γνήσια (πιστεύοντας ο χρήστης ότι συνδέεται στη γνήσια) με σκοπό να χρησιμοποιήσει τα διαπιστευτήριά του στην πλαστή για να συνδεθεί. Συνεπώς, εφόσον η σελίδα που τελικά ο χρήστης συνδέθηκε είναι η πλαστή, τα διαπιστευτήρια που μόλις χρησιμοποίησε αποστέλλονται στον εξυπηρετητή του επιτιθέμενου (αντί της τράπεζας) και αποθηκεύονται στη δική του βάση δεδομένων. Έτσι ο επιτιθέμενος έχει ξεγελάσει το θύμα, έχει υποκλέψει τα διαπιστευτήριά του και μπορεί να τα χρησιμοποιήσει προς όφελός του κάνοντας ότι συναλλαγές θελήσει στην πραγματική ιστοσελίδα της τράπεζας.

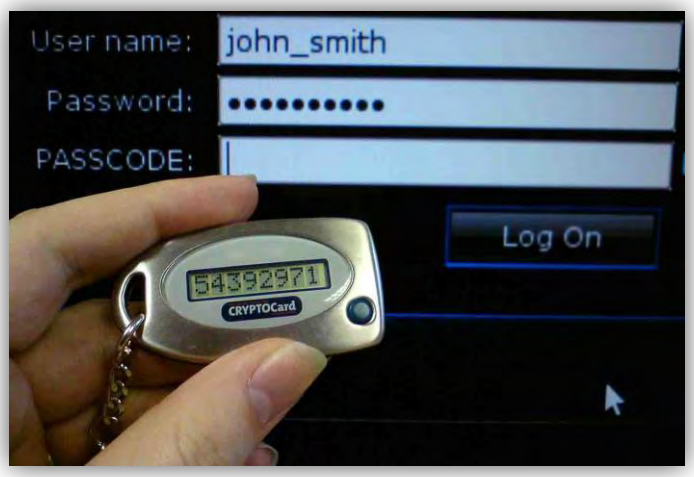
Παρομοίως, **στην επίθεση της «εσκεμμένης ανακατεύθυνσης»**, ο επιτιθέμενος αρχικά έχει αλλοιώσει (με κάποιον τρόπο) το αρχείο αντιστοίχησης ονομάτων υπολογιστών με διευθύνσεις IP (hosts file) που βρίσκεται στον υπολογιστή του θύματος. Έτσι, με βάση αυτήν την αλλαγή, όταν

το θύμα πληκτρολογήσει μια διεύθυνση στο πρόγραμμα περιήγησής του για να συνδεθεί (π.χ. την διεύθυνση της τράπεζάς του), ο υπολογιστής του τον ανακατευθύνει αυτόματα σε άλλη διεύθυνση (αντί αυτής που πληκτρολόγησε αρχικά) και τον συνδέει σε αυτήν που έχει ορίσει ο επιτιθέμενος. Η ανακατεύθυνση από την μία ιστοσελίδα στην άλλη γίνεται στιγμιαία, χωρίς να γίνει εύκολα αντιληπτή από τον χρήστη. Έτσι ο χρήστης συνδέεται στην πλαστή ιστοσελίδα του επιτιθέμενου η οποία (όπως και στο ηλεκτρονικό ψάρεμα) είναι σχεδόν η ίδια (ή πολλές φορές, ακριβώς η ίδια) με τη γνήσια και εκεί γίνεται η υποκλοπή. Μόνο αν ο χρήστης έχει προχωρημένες γνώσεις πληροφορικής και είναι ιδιαίτερα παρατηρητικός μπορεί να αντιληφθεί αυτήν την ανακατεύθυνση και να προστατευθεί.

Ο τρόπος με τον οποίο οι Διαχειριστές Κλειδαρίθμων μπορούν να προστατεύσουν τον χρήστη από αυτού του είδους τις επιθέσεις, είναι με την ταυτόχρονη χρήση **αυτοματοποιημένων σεναρίων σύνδεσης (login scripts)**. Δηλαδή, ο Διαχειριστής Κλειδαρίθμων συνεργάζεται με κάποια κομμάτια κώδικα τα οποία, πρώτα διαβάζουν την τρέχουσα διεύθυνση URL που έχει φορτωθεί στο πρόγραμμα περιήγησης, την συγκρίνουν με αυτές που βρίσκονται αποθηκευμένες στη βάση δεδομένων του Διαχειριστή και μόνο αν βρεθεί κάποια καταχώρηση που περιέχει ακριβώς την ίδια διεύθυνση το login script εξάγει τα κατάλληλα διαπιστευτήρια και τα τοποθετεί στα αντίστοιχα πλαίσια πιστοποίησης της ιστοσελίδας. Συνεπώς, αν δε βρεθεί κάποια καταχώρηση ενώ χρήστης θυμάται ότι είχε καταχωρήσει τα διαπιστευτήρια για αυτήν την ιστοσελίδα στο παρελθόν, τότε αυτό σημαίνει ότι πρόκειται για πιθανή εξαπάτηση. Με αυτόν τον τρόπο, οι Διαχειριστές Κλειδαρίθμων αποκτούν επιπλέον λειτουργικότητα, και εκτός από την αποθήκευση διαπιστευτηρίων ορίζουν και ένα σύστημα διασφάλισης από επιθέσεις ηλεκτρονικής εξαπάτησης.

Οι Διαχειριστές Κλειδαρίθμων επίσης παρέχουν και **προστασία ενάντια στα «προγράμματα καταγραφής πληκτρολογήσεων» (keyloggers)**. Όταν χρησιμοποιείται ένας Διαχειριστής Κλειδαρίθμων βασιζόμενος σε πολλαπλά κριτήρια πρόσβασης (βλέπε κατηγορίες των Διαχειριστών Κλειδαρίθμων), ο οποίος λειτουργεί και ως αυτόματος συμπληρωτής φορμών (κεφ. 1.1), ο χρήστης δεν χρειάζεται να πληκτρολογήσει κάποιο όνομα χρήστη ή κωδικό πρόσβασης μέσω του πληκτρολογίου σε κάποιο πεδίο της ιστοσελίδας. Έτσι δεν υπάρχουν πληκτρολογήσεις ούτως ώστε να καταγραφούν από κάποιο πρόγραμμα καταγραφής.

Αντιθέτως, οι Διαχειριστές Κλειδαρίθμων **δεν μπορούν να προστατεύσουν τον χρήστη από τις επιθέσεις man-in-the-browser**, στις οποίες κάποιο κακόβουλο λογισμικό (malware) βρίσκεται εγκατεστημένο στη συσκευή του και εκτελεί κακόβουλες λειτουργίες, εν αγνοία του. Στις επιθέσεις αυτές ο χρήστης είναι συνδεδεμένος στον λογαριασμό του σε μια ιστοσελίδα (της τράπεζάς του για παράδειγμα) και το κακόβουλο λογισμικό παράλληλα (αλλά όχι φανερά) στέλνει διάφορες εντολές στην ιστοσελίδα αυτή και εκτελεί τραπεζικές δοσοληψίες χωρίς να καταλαβαίνει ο χρήστης το οτιδήποτε.



Εικόνα 1.3 - Παράθυρο πρόσβασης ενός Διαχειριστή Κλειδαρίθμων ο οποίος υποστηρίζει πρόσβαση με «πολλαπλά κριτήρια πιστοποίησης». Στον συγκεκριμένο Διαχειριστή χρησιμοποιείται εξωτερικός κωδικός ασφαλείας (security token), επιπλέον του κύριου κωδικού πρόσβασης και του ονόματος χρήστη.

1.3 Ευπάθειες των Διαχειριστών Κλειδαρίθμων

Οι Διαχειριστές Κλειδαρίθμων στις εκδόσεις τους για σταθερούς υπολογιστές, αλλά και σε αυτές που λειτουργούν ως επεκτάσεις των προγραμμάτων περιήγησης (browser extensions) (λειτουργούν δηλαδή μέσω των περιηγητών ως πρόσθετες υπηρεσίες/δυνατότητες), προσφέρουν ένα μεγάλο ποσοστό άνεσης στους χρήστες αφού αποθηκεύουν και οργανώνουν δεκάδες ή εκατοντάδες διαπιστευτήρια πρόσβασης. Ωστόσο, κάποιοι από αυτούς δεν παρέχουν καμία απολύτως προστασία στη βάση δεδομένων τους, στην οποία αποθηκεύουν τα διαπιστευτήρια πρόσβασης που διαχειρίζονται. Εάν ο υπολογιστής είναι ήδη ενεργοποιημένος δηλαδή, και κάποιος πάει και τον ψάξει, τότε το άτομο αυτό μπορεί να αποκτήσει πρόσβαση χωρίς περιορισμούς στα διαπιστευτήρια και μπορεί να τα υποκλέψει.

Η κατάσταση αυτή έχει βελτιωθεί ελαφρώς με την πάροδο του χρόνου, ζητώντας από τον χρήστη να πληκτρολογήσει έναν **κύριο κωδικό πρόσβασης (master password)** προκειμένου να αποκτήσει πρόσβαση στον χώρο των διαπιστευτηρίων και να τα αναγνώσει. Ωστόσο, παρόλη την προστασία που προσφέρει ο κύριος κωδικός, εάν τα διαπιστευτήρια αποθηκεύονται σε έναν μη κρυπτογραφημένο χώρο, τότε το πρόβλημα της υποκλοπής εξακολουθεί να υφίσταται, απλά έχει μεταφερθεί σε άλλο σημείο.

Κάποιοι πιο «ισχυροί» Διαχειριστές Κλειδαρίθμων, εκτός του ότι λειτουργούν με κύριο κωδικό πρόσβασης (επιλεγμένο από τον χρήστη) ή με μια μεγάλη φράση πρόσβασης (passphrase), χρησιμοποιούν τα χαρακτηριστικά αυτά ως κλειδί για την **κρυπτογράφηση της βάσης δεδομένων** τους. Η ασφάλεια αυτής της προσέγγισης εξαρτάται από την ισχύ του επιλεγμένου

κύριου κωδικού πρόσβασης - τον οποίο κάποιος θα μπορούσε να μαντέψει αν είναι απλός ή να του επιτεθεί μέσω ωμής βίας - και, επίσης, από το αν ο κύριος κωδικός πρόσβασης (ή η φράση πρόσβασης) αποθηκεύεται τοπικά στην μόνιμη μνήμη του μηχανήματος ή όχι, ούτως ώστε ένα κακόβουλο πρόγραμμα ή κάποιο άτομο να μπορέσει να τον διαβάσει.

Εάν ο κύριος κωδικός πρόσβασης (ή η φράση πρόσβασης) δεν είναι κρυπτογραφημένοι, τότε καθιστούν το σύνολο των διαπιστευτηρίων ευάλωτο, για ακόμη μία φορά. Το γεγονός αυτό δείχνει την αντίστροφη σχέση μεταξύ της χρηστικότητας και της ασφάλειας: μπορεί ο κύριος κωδικός πρόσβασης να είναι βολικός σε χρήση για την είσοδο στο πρόγραμμα, αλλά αποτελεί ένα τμήμα του προγράμματος το οποίο, αν παραβιαστεί και διαβαστεί, τότε καθιστά το σύνολο των κρυπτογραφημένων διαπιστευτηρίων σε κίνδυνο.

Προχωρώντας την ανάλυση σε μεγαλύτερο βάθος, μπορούμε να παρατηρήσουμε ότι όπως συμβαίνει και με οποιοδήποτε σύστημα το οποίο απαιτεί την εισαγωγή ενός κωδικού πρόσβασης από τον χρήστη, έτσι και στους Διαχειριστές Κλειδαρίθμων, η εισαγωγή του κύριου κωδικού πρόσβασης μπορεί να γίνει στόχος επίθεσης και ανακάλυψης από **προγράμματα καταγραφής πληκτρολογήσεων (keyloggers)** ή από μεθόδους **ακουστικής κρυπτανάλυσης (acoustic cryptanalysis)**.

Στην προσπάθειά τους για να αποτρέψουν το πρόβλημα αυτό, ορισμένοι Διαχειριστές Κλειδαρίθμων χρησιμοποιούν **εικονικά πληκτρολόγια (virtual keyboards)**, με τα οποία ο χρήστης μπορεί να καταχωρήσει τον κύριο κωδικό πρόσβασής του κάνοντας μόνο κάποια «κλικ» με το ποντίκι του στα αντίστοιχα γράμματα. Η μέθοδος αυτή όμως, είναι και πάλι ευάλωτη σε ορισμένα, πιο προχωρημένα **προγράμματα παρακολούθησης**, τα οποία εκτός από καταγραφή πληκτρολογήσεων **λαμβάνουν και στιγμιότυπα της οθόνης του χρήστη (screenshots)** ανά κάποιο χρονικό διάστημα (ορισμένο από τον επιτιθέμενο). Έτσι, κατά τη διάρκεια που ο χρήστης «κλικάρει» στα αντίστοιχα κουμπιά του εικονικού πληκτρολογίου, το πρόγραμμα παρακολούθησης καταγράφει συνεχόμενα, μέσω εικόνων, τη θέση του ποντικιού με συνέπεια ο επιτιθέμενος να μπορεί να ανακαλύψει τον κύριο κωδικό πρόσβασης συνδυάζοντας τις εικόνες αυτές.

Ο κίνδυνος που εισάγει η παρακολούθηση με στιγμιότυπα μπορεί να μετριαστεί αν οι Διαχειριστές Κλειδαρίθμων βασίζονται σε **πολλαπλά κριτήρια πρόσβασης** (βλέπε κατηγορίες των Διαχειριστών Κλειδαρίθμων). Με χρήση αυτής της κατηγορίας, οι Διαχειριστές Κλειδαρίθμων μπορούν να υποχρεώσουν τον χρήστη να χρησιμοποιήσει **συσκευή παραγωγής εξωτερικών κωδικών ασφαλείας** (κωδικών οι οποίοι μεταβάλλονται με τον χρόνο - ποτέ δε είναι οι ίδιοι) και να τους πληκτρολογήσει αναγκαστικά στο παράθυρο πρόσβασης (login window) του προγράμματος για να μπορέσει να εισέλθει σε αυτό (εικόνα 1.3).

Εικόνα 1.4 - Εικονικό Πληκτρολόγιο (Virtual Keyboard) για την εισαγωγή δεδομένων πιστοποίησης (κωδικών, ονομάτων χρήστη, κ.α.) στους Διαχειριστές Κλειδαρίθμων. Με τη χρήση του εικονικού πληκτρολογίου παρακάμπτεται η δυνατότητα παρακολούθησης των πλήκτρων από τους Καταγραφείς Πληκτρολογήσεων (keyloggers).

Πολλοί Διαχειριστές Κλειδαρίθμων λειτουργούν επίσης **ως επέκταση του προγράμματος περιήγησης** του χρήστη. Σε αυτούς όμως, λόγω της άμεσης σχέσης με ιστοσελίδες και με το Διαδίκτυο, έχει παρατηρηθεί ότι κρύβουν ένα μεγάλο πλήθος από παγίδες που μπορεί να οδηγήσουν σε εκροή δεδομένων. Πρόσφατα (το 2014), το τμήμα «Ηλεκτρολόγων Μηχανικών και Επιστημών Υπολογιστή» του πανεπιστημίου της Καλιφόρνια του Berkeley, κυκλοφόρησε μια λεπτομερή έρευνα η οποία χρησιμοποίησε αρκετούς Διαχειριστές Κλειδαρίθμων τέτοιου είδους και παρατήρησε τα εξής κενά ασφαλείας: [3]

- **Ευπάθειες σελιδοδεικτών τύπου «bookmarklets»:** Πρόκειται για Διαχειριστές Κλειδαρίθμων οι οποίοι βασίζονται σε σελιδοδείκτες τύπου «bookmarklets» για την επίτευξη της σύνδεση (login) των χρηστών τους σε ιστοσελίδες. Αυτού του τύπου οι σελιδοδείκτες διαφέρουν από τους απλούς, στο γεγονός ότι χρησιμοποιούν κώδικα JavaScript για να προσδώσουν επιπλέον δυνατότητες στον χρήστη και στον περιηγητή. Ωστόσο, εάν δεν είναι σωστά υλοποιημένοι τότε μία κακόβουλη ιστοσελίδα μπορεί να τους καταχραστεί για να κλέψει τον κωδικό πρόσβασης ενός χρήστη. Η κύρια αιτία αυτής της ευπάθειας είναι ότι το περιβάλλον της JavaScript μιας ιστοσελίδας δεν μπορεί να θεωρηθεί πάντα έμπιστο, διότι αυτή η ιστοσελίδα μπορεί να τρέξει ότι κώδικα θέλει, κακόβουλο ή μη. [4]

- **Ευπάθειες του παγκόσμιου ιστού:** τυπικές ευπάθειες του παγκόσμιου ιστού (web) μπορούν επίσης να παρουσιαστούν σε αυτό το είδος των Διαχειριστών Κλειδαρίθμων. Ειδικότερα, επιθέσεις τύπου XSS (Cross-site scripting) και CSRF (Cross-site request forgery) οι οποίες μπορούν να εξαπολυθούν από τους επιτιθέμενους (hackers), μπορούν να συμβάλλουν στην υποκλοπή του κωδικού πρόσβασης ενός χρήστη.

- **<u>Ευπάθειες στην εξουσιοδότηση:</u>** Ένα άλλο πιθανό πρόβλημα προέρχεται μπερδεύοντας την επικύρωση (authentication) με την εξουσιοδότηση (authorization). Από την έρευνα διαπιστώθηκε ότι αρκετοί Διαχειριστές Κλειδαρίθμων είχαν τέτοιες ευπάθειες, ειδικά αυτοί που επέτρεπαν στους χρήστες τους να διαμοιράζονται τα διαπιστευτήριά τους με άλλους χρήστες. Έτσι, μπορούσε να γίνει η υποκλοπή μέσω δικτύου (αφού ο διαμοιρασμός γινόταν μη κρυπτογραφημένα) και ο επιτιθέμενος να τους χρησιμοποιήσει για να επικυρωθεί και συνδεθεί κάπου, χωρίς να έχει την εξουσιοδότηση να το κάνει.

- **<u>Ευπάθειες στη διασύνδεση του χρήστη:</u>** Ορισμένοι Διαχειριστές Κλειδαρίθμων ζητούσαν από τον χρήστη να συνδεθεί σε αυτούς μέσω ενός «iframe» (πλαίσιο δηλαδή μέσα σε μια ιστοσελίδα, το οποίο αν και μοιάζει να είναι μέρος της ιστοσελίδας αυτής, τρέχει από άλλη διεύθυνση). Αυτή η διαδικασία είναι δυστυχώς ανασφαλής. Εκπαιδεύει τον χρήστη στο να πληκτρολογεί τον κύριο κωδικό πρόσβασής του τη στιγμή που η διεύθυνση URL που εμφανίζεται στον περιηγητή δεν είναι αυτή του Διαχειριστή Κλειδαρίθμων. Ένας κακόβουλος χρήστης θα μπορούσε να το εκμεταλλευτεί αυτό και μέσω επίθεσης «ηλεκτρονικού ψαρέματος» (βλέπε κεφ. 1.2), να δημιουργήσει ένα ψεύτικο «iframe» με σκοπό να υποκλέψει τα διαπιστευτήρια του χρήστη. Για να ξεφύγουν από αυτήν την ευπάθεια οι Διαχειριστές Κλειδαρίθμων, μια πιο ασφαλής προσέγγιση θα ήταν αντί να χρησιμοποιούν «iframes» να ανοίγουν μια νέα καρτέλα στον περιηγητή ή ένα νέο pop-up παράθυρο, ούτως ώστε οι χρήστες να μπορούν να συνδέονται σε αυτούς βλέποντας ταυτόχρονα και την γνήσια διεύθυνση URL στον περιηγητή.

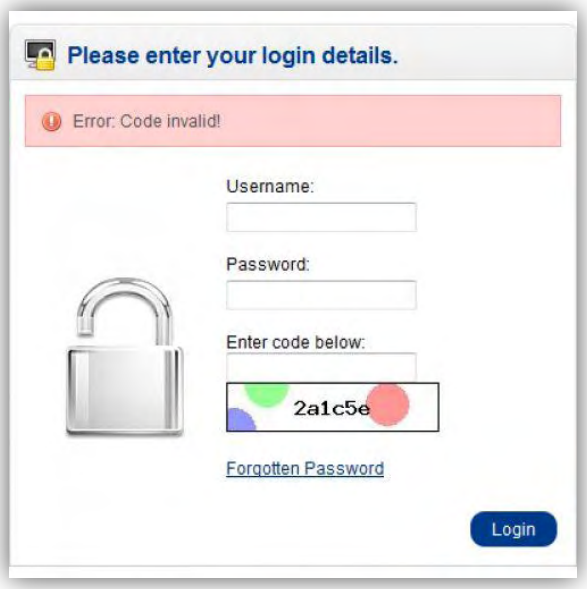
Περνώντας στις δυνατότητες που υποστηρίζουν οι Διαχειριστές Κλειδαρίθμων, ένα άλλο θέμα που απασχολεί την ασφάλεια των διαπιστευτηρίων είναι η ενσωματωμένη **«γεννήτρια κωδικών πρόσβασης» (password generator)** που κάποιοι περιλαμβάνουν. Μέσω αυτής της γεννήτριας, οι Διαχειριστές παρέχουν στον χρήστη τη δυνατότητα παραγωγής ενός ισχυρού κωδικού, αυτόματα, σύμφωνα με τα κριτήρια θα ζητήσει. Για παράδειγμα, ο χρήστης μπορεί να ζητήσει από το πρόγραμμα να του δημιουργήσει έναν ισχυρό κωδικό πρόσβασης, ο οποίος θα έχει μήκος *n* και θα αποτελείται από γράμματα (πεζά και κεφαλαία), σύμβολα και αριθμούς, ούτως ώστε να τον χρησιμοποιήσει σε κάποια ιστοσελίδα κατά τη διαδικασία της εγγραφής του (user register). Οι κωδικοί πρόσβασης που δημιουργούνται μέσω της γεννήτριας όμως, μπορεί να μαντευθούν μετά από κάποιο χρόνο (μέσω κρυπτανάλυσης) εάν ο Διαχειριστής Κλειδαρίθμων χρησιμοποιεί μια τυπική μέθοδο για την παραγωγή ψευδοτυχαίων αριθμών αντί για μία που θα είναι κρυπτογραφικά ασφαλής.

Πρόβλημα ασφαλείας αποτελεί επίσης και η **εναλλαγή της κύριας μνήμης από το λειτουργικό σύστημα (swapping)** στο αρχείο σελιδοποίησης του σκληρού δίσκου (page file). Οι Διαχειριστές Κλειδαρίθμων, όπως και όλα τα προγράμματα που εκτελούνται σε ένα λειτουργικό σύστημα συμμετέχουν στην εναλλαγή, αφού αποτελεί την πλέον σημαντική τεχνική τους για την ταυτόχρονη και απροβλημάτιστη εκτέλεση πολλών εφαρμογών. Οι Διαχειριστές όμως που δεν αποτρέπουν την εναλλαγή του προσωπικού τους χώρου μνήμης στο αρχείο σελιδοποίησης, κινδυνεύουν με υποκλοπή των διαπιστευτηρίων τους απευθείας από τον σκληρό δίσκο (μέσω κακόβουλων προγραμμάτων που διαβάζουν το αρχείο σελιδοποίησης), αφού κατά πάσα πιθανότητα τα διαπιστευτήρια που υπάρχουν εκεί βρίσκονται σε μη κρυπτογραφημένη μορφή. Η απενεργοποίηση της εναλλαγής, ή η εγκατάστασης περισσότερης μνήμης RAM στον υπολογιστή, μπορεί να αποτρέψει τον κίνδυνο αυτό.

Ακόμα όμως και να απενεργοποιηθεί η εναλλαγή, οι Διαχειριστές Κλειδαρίθμων πάλι θα μπορούσαν να κινδυνεύουν από τις μεταβλητές που χρησιμοποιούν στην κύρια μνήμη, αφού υπάρχουν και αντίστοιχα κακόβουλα προγράμματα για αυτόν τον σκοπό, τα οποία ψάχνουν και αναλύουν τη μνήμη του προγράμματος καταγράφοντας τα πάντα ή συγκεκριμένα κομμάτια και μεταβλητές (στις οποίες μπορεί να υπάρχουν τα διαπιστευτήρια του χρήστη σε μη κρυπτογραφημένη μορφή). Τα προγράμματα αυτά ονομάζονται **«Υποκλοπείς Μνήμης RAM» (RAM Scrapers)** και αναλύονται (μαζί με άλλα) στο κεφάλαιο 2.2.

Τέλος, μια επιπλέον μορφή προστασίας, που μπορεί να προσφέρει ένας Διαχειριστής Κλειδαρίθμων είναι να επιτρέπει έναν **συγκεκριμένο αριθμό από αποτυχημένες προσπάθειες εισόδου**. Αυτό σημαίνει ότι μετά από κάποιες φορές ανεπιτυχούς πιστοποίησης του χρήστη, ο Διαχειριστής Κλειδαρίθμων κλειδώνει αυτόματα και απαιτεί από τον κατασκευαστή του ή από την εταιρία που έχει αναλάβει την υποστήριξή του να τον ξεκλειδώσει. Αυτή η διαδικασία αποτελεί τον καλύτερο τρόπο για την προστασία από επιθέσεις ωμής βίας.

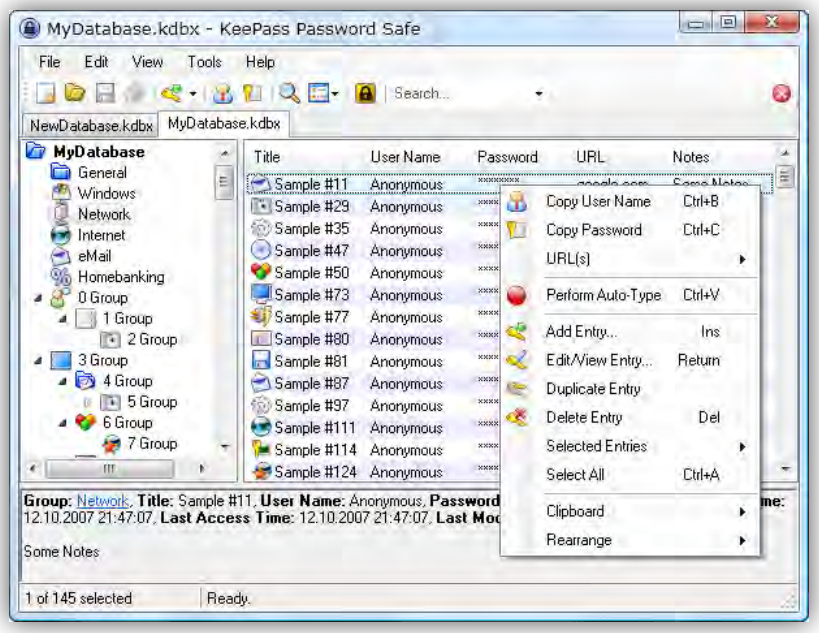
Βελτιστοποίηση αυτής της διαδικασίας, η οποία έτσι όπως δουλεύει μπορεί να αποβεί ενοχλητική από πολλούς χρήστες αν αυτοί δεν είναι ιδιαίτερα εξοικειωμένοι με το πληκτρολόγιο και κάνουν συχνά λάθη κατά την πληκτρολόγηση, αποτελεί η εισαγωγή πιστοποίησης μέσω **επίλυσης τεστ «CAPTCHA»** (Completely Automated Public Turing test to tell Computers and Humans Apart) ως ενδιάμεσο στάδιο, πριν το ολοκληρωτικό κλείδωμα του προγράμματος. Με βάση αυτή τη βελτιστοποίηση, όταν ο χρήστης για παράδειγμα κάνει λάθος τον κύριο κωδικό του 3 φορές, το πρόγραμμα θα μπορούσε να του εμφανίζει μια εικόνα CAPTCHA και ο χρήστης να πληκτρολογεί στο αντίστοιχο πλαίσιο αυτά που βλέπει στην εικόνα. Αν ο χρήστης αποτύχει και στο τεστ CAPTCHA για παραπάνω από 20-30 φορές, τότε το πρόγραμμα μπορεί να κλειδώσει ολοκληρωτικά. Ως επιπλέον μέτρο ασφαλείας, η επίλυση τεστ CAPTCHA θα μπορούσε να είναι ενεργοποιημένη εξ αρχής στο πρόγραμμα, αποτρέποντας τις επιθέσεις ωμής βίας από την πρώτη στιγμή.



Εικόνα 1.5 - Η μέθοδος πιστοποίησης μέσω επίλυσης τεστ «CAPTCHA» (Completely Automated Public Turing test to tell Computers and Humans Apart).

1.4 Υπάρχουσες υλοποιήσεις γνωστών Διαχειριστών Κλειδαρίθμων

1.4.1 Ο Διαχειριστής Κλειδαρίθμων «KeePass Password Safe»

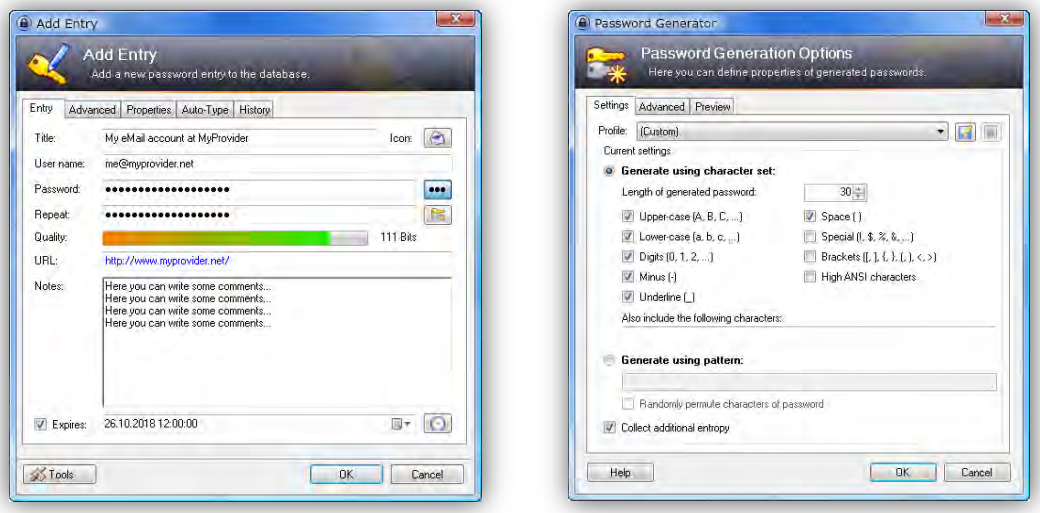


Εικόνα 1.6 - Το κεντρικό παράθυρο (main window) του Διαχειριστή Κλειδαρίθμων «KeePass».

Περιγραφή:

Ο KeePass Password Safe είναι ένας δωρεάν, ανοιχτού κώδικα, πολλαπλής πλατφόρμας (cross-platform) και «ελαφρύς» (από άποψη χρήσης πόρων συστήματος) Διαχειριστής Κλειδαρίθμων σχεδιασμένος για το λειτουργικό σύστημα Microsoft Windows, αλλά και με ανεπίσημες εκδόσεις για Linux, Mac OS X, iOS, Android και Windows Phone (οι οποίες ονομάζονται διαφορετικά ανά περίπτωση). Ο KeePass αποθηκεύει όλα τα ονόματα χρηστών, κωδικούς πρόσβασης, άλλα δεδομένα πιστοποίησης, σημειώσεις ελεύθερης μορφής, σε μια ασφαλή κρυπτογραφημένη βάση δεδομένων, η οποία προστατεύεται από έναν κύριο κωδικό πρόσβασης ή από ένα αρχείο κλειδιού. Ως προεπιλογή, η κρυπτογραφημένη βάση δεδομένων του KeePass δεν αποθηκεύεται σε κάποιο κέντρο δεδομένων (cloud), αλλά σε τοπικό επίπεδο, στο μηχάνημα του χρήστη.

Ο KeePass είναι ευέλικτος και επεκτάσιμος, με πολλές επιλογές διαμόρφωσης. Διαθέτει επιλογές για έλεγχο ταυτοποίησης δύο παραγόντων (two-factor authentication) καθώς και για την υπηρεσία ασφαλούς επιφάνειας εργασίας των Windows (Windows secure desktop) έτσι ώστε να προστατευθεί από προγράμματα καταγραφής πληκτρολογήσεων (keyloggers). Ο KeePass υποστηρίζει επιπλέον την εισαγωγή διαπιστευτηρίων από περισσότερους των 30 συχνά χρησιμοποιούμενων Διαχειριστών Κλειδαρίθμων. Υπάρχει επίσης μια μεγάλη ποικιλία από πρόσθετα (plugins) που μπορούν να εγκατασταθούν σε αυτόν, στα οποία όμως πρέπει να δοθεί ιδιαίτερη προσοχή όταν επιλέγονται και εγκαθίστανται από άλλες πηγές, διαφορετικές από την επίσημη. [8]



Εικόνα 1.7 - Το παράθυρο εισαγωγής νέας εγγραφής στη βάση δεδομένων (αριστερά) και το παράθυρο παραγωγής κωδικού μέσω της γεννήτριας κωδικών (δεξιά) του Διαχειριστή Κλειδαρίθμων «KeePass».

Δυνατότητες:

- Διαχείριση Κλειδαρίθμων.
- Εισαγωγή και εξαγωγή δεδομένων διαπίστευσης προς και από το πρόγραμμα.
- Υποστήριξη πολλαπλών χρηστών. Υπάρχει η δυνατότητα ταυτόχρονης σύνδεσης πολλών χρηστών στη βάση δεδομένων του προγράμματος (συνήθως μέσω δικτυακού διαμοιραζόμενου σκληρού δίσκου), με ταυτόχρονες αλλαγές επίσης στις καταχωρήσεις της.
- Υπηρεσίες αυτόματης συμπλήρωσης, καθολικών πλήκτρων συντόμευσης και μεταφοράς/απόθεσης. Όταν ο KeePass τρέχει στο παρασκήνιο (με ξεκλείδωτη τη βάση δεδομένων του) και ο χρήστης πατήσει έναν

συγκεκριμένο συνδυασμό πλήκτρων συντόμευσης ενώ βρίσκεται με ανοικτό κάποιο παράθυρο που περιμένει την εισαγωγή διαπιστευτηρίων, τότε αυτός αναζητά την κατάλληλη εγγραφή στη βάση δεδομένων και εκτελεί την διαδικασία αυτόματης συμπλήρωσης για το παράθυρο αυτό. Επίσης, όλα τα πεδία, τίτλος, όνομα χρήστη, κωδικός πρόσβασης, διεύθυνση URL και σημειώσεις μπορούν να μεταφερθούν εύκολα μέσω μεταφοράς και απόθεσης (drag-n-drop) σε άλλα παράθυρα αν χρειαστεί.

- Υποστήριξη περιηγητών. Όπως συμβαίνει και με τα παράθυρα των εφαρμογών, έτσι και με τους περιηγητές (Microsoft Internet Explorer και Mozilla Firefox μέχρι στιγμής) ο KeePass λειτουργεί ως αυτόματος συμπληρωτής φορμών, συμπληρώνοντας αυτόματα τα κατάλληλα πεδία σε μία ιστοσελίδα όταν αυτή φορτωθεί.

- Ενσωματωμένη γεννήτρια κωδικών.

- Επέκταση υπάρχουσας λειτουργικότητας μέσω πρόσθετων.

Ασφάλεια:

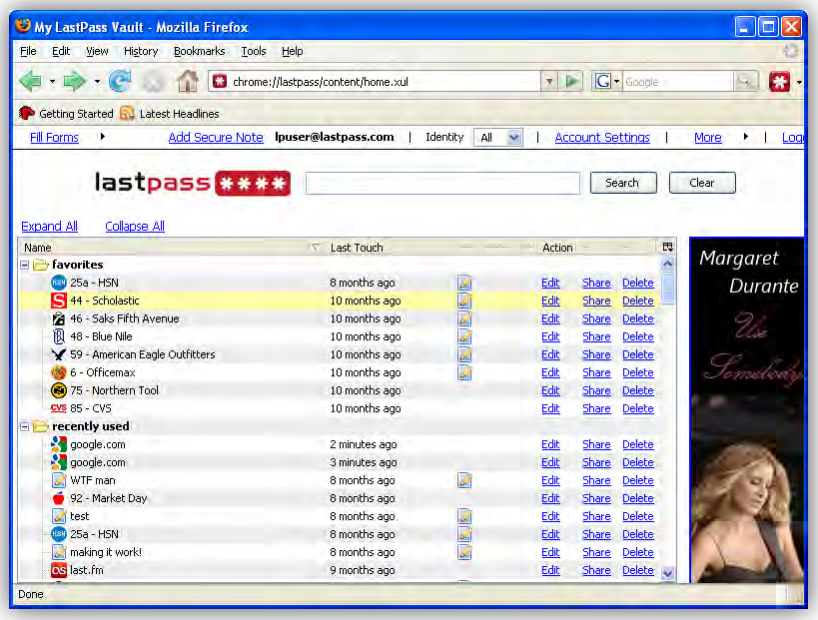
- Οι κωδικοί πρόσβασης αποθηκεύονται πάντα στην κύρια μνήμη σε μια ασφαλή, μη εναλλάξιμη περιοχή (non swappable) όταν ο KeePass τρέχει.

- Η πρόσβαση στη βάση δεδομένων περιορίζεται από έναν κύριο κωδικό πρόσβασης ή από ένα αρχείο κλειδιού. Και οι δύο μέθοδοι μπορούν να συνδυαστούν αποτελώντας ένα συνδυασμένο ισχυρό κύριο κλειδί.

- Η κρυπτογράφηση της βάσης δεδομένων γίνεται μέσω των συμμετρικών αλγορίθμων AES ή Twofish, με τον AES να αποτελεί την προεπιλεγμένη μέθοδο.

Επίσημη ιστοσελίδα: http://keepass.info/

1.4.2 Ο Διαχειριστής Κλειδαρίθμων «LastPass»



Εικόνα 1.8 - Το κεντρικό παράθυρο (main window) με την λίστα των αποθηκευμένων διαπιστευτηρίων του Διαχειριστή Κλειδαρίθμων «LastPass».

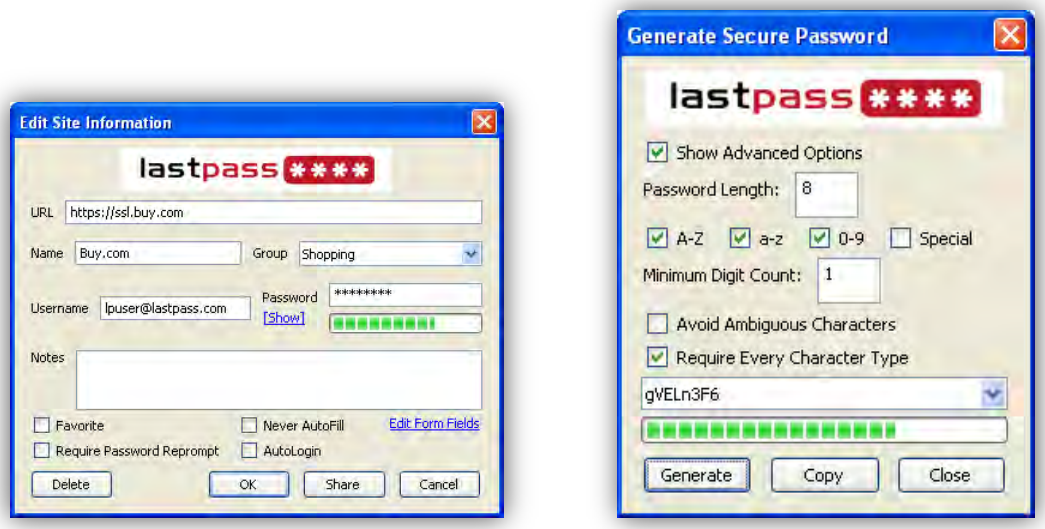
Περιγραφή:

Ο LastPass είναι μία δωρεάν υπηρεσία Διαχείρισης Κλειδαρίθμων η οποία επιδιώκει να λύσει το πρόβλημα της σύγχυσης που δημιουργούν οι πολλοί κωδικοί πρόσβασης με το να συγκεντρώνει και να διαχειρίζεται τα διαπιστευτήρια του χρήστη μέσα σε κέντρα δεδομένων (cloud). Η βασική μορφή του LastPass αποτελεί μία web εφαρμογή που τρέχει από την επίσημη ιστοσελίδα του, αλλά περιλαμβάνει επίσης και πρόσθετα για πολλούς περιηγητές του παγκόσμιου ιστού ούτως ώστε να ενσωματώνεται και να τρέχει απευθείας από εκεί. Στην ενσωματωμένη μορφή του στους περιηγητές, ο LastPass περιλαμβάνει επίσης και υποστήριξη για «bookmarklets». Ο LastPass εξελίσσεται και διανέμεται από την εταιρία Marvasol, Inc..

Τα διαπιστευτήρια των χρηστών στον LastPass προστατεύονται από έναν κύριο κωδικό πρόσβασης (ο οποίος κρυπτογραφείται τοπικά στο μηχάνημα του χρήστη) και συγχρονίζονται με οποιοδήποτε πρόγραμμα περιήγησης. Ο LastPass διαθέτει δυνατότητα αυτόματης συμπλήρωσης φορμών και υποστηρίζει λειτουργίες παραγωγής κωδικών πρόσβασης, κοινής χρήσης ιστοσελίδων όπως και καταγραφής ιστοσελίδων. Επίσης, εκτός από την δωρεάν έκδοση υπάρχει και η επί πληρωμή premium έκδοση η οποία περιλαμβάνει

επιπλέον πρόσβαση μέσω εφαρμογών κινητών τηλεφώνων (Android, iOS, Windows Phone), πιστοποίηση πολλαπλών κριτηρίων και δυνατότητα εκτέλεσης από μονάδα USB μέσω αυτόνομης εφαρμογής, όταν δεν είναι δυνατή η εγκατάσταση του προσθέτου στον περιηγητή ή ο διαχειριστής του υπολογιστή επιτρέπει μόνο συγκεκριμένες διευθύνσεις URL να φορτωθούν από τους περιηγητές.

Αν και πρόκειται για προϊόν κλειστού κώδικα, ο Sameer Kochhar (ένας από τους προγραμματιστές του LastPass), υποστήριξε ότι θεωρητικά, η ακεραιότητα του λογισμικού θα μπορούσε να επαληθευθεί χωρίς αυτό να μετατραπεί σε ανοιχτού κώδικα και ανέφερε ότι οι προγραμματιστές ίσως είναι ανοικτοί στο ενδεχόμενο μελλοντικά η διεπαφή χρήστη του LastPass να γίνει ανοιχτού κώδικα. [9]



Εικόνα 1.9 - Το παράθυρο επεξεργασίας μιας υπάρχουσας εγγραφής από τη βάση δεδομένων (αριστερά) και το παράθυρο παραγωγής κωδικού μέσω της γεννήτριας κωδικών (δεξιά) του Διαχειριστή Κλειδαρίθμων «LastPass».

Δυνατότητες:

- Ένας κύριος κωδικός πρόσβασης για ξεκλείδωμα - κρυπτογραφημένος.
- Συγχρονισμός διαπιστευτηρίων μεταξύ των περιηγητών.
- Παραγωγή ασφαλών κωδικών πρόσβασης.
- Λειτουργία αυτόματου συμπληρωτή φορμών.

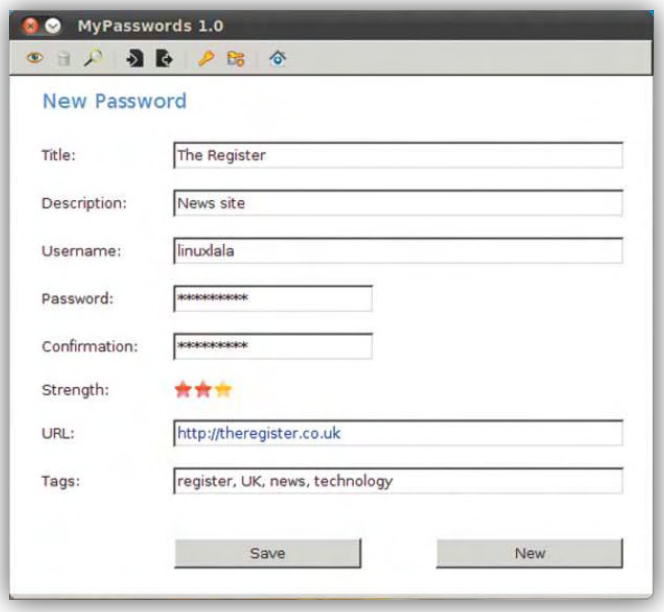
- Εισαγωγή και εξαγωγή κωδικών πρόσβασης προς και από το πρόγραμμα.
- Δυνατότητα αυτόνομης (portable) έκδοσης.
- Πιστοποίηση Πολλαπλών Κριτηρίων.
- Κλείδωμα/Ξεκλείδωμα μέσω δακτυλικών αποτυπωμάτων.
- Διαθέσιμος για πολλαπλές πλατφόρμες (cross-platform).
- Πρόσβαση μέσω κινητών τηλεφώνων.

Ασφάλεια:

- Η βάση δεδομένων με τα διαπιστευτήρια του χρήστη βρίσκεται αποθηκευμένη στους εξυπηρετητές του προγράμματος, πάντα κρυπτογραφημένη και η επικοινωνία γίνεται μόνο μέσω διαδικτύου.
- Σε περίπτωση αδυναμίας σύνδεσης με το Διαδίκτυο το πρόγραμμα κρατάει τοπικό αντίγραφο διαπιστευτηρίων στο τοπικό μηχάνημα του χρήστη, και αυτό σε κρυπτογραφημένη μορφή.
- Χρησιμοποιείται κρυπτογράφηση AES 256-bit για την βάση δεδομένων.
- Τα διαπιστευτήρια της βάσης δεδομένων αποκρυπτογραφούνται μόνο στο τοπικό μηχάνημα του χρήστη μέσω ενός κύριου κωδικού πρόσβασης.

Επίσημη ιστοσελίδα: https://lastpass.com/

1.4.3 Ο Διαχειριστής Κλειδαρίθμων «MyPasswords»

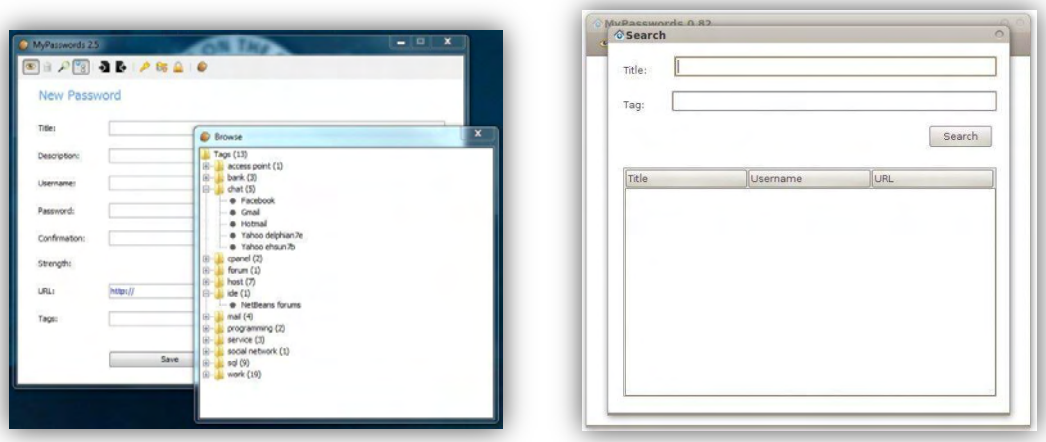


Εικόνα 1.10 - Το παράθυρο προσθήκης μίας νέας εγγραφής στη βάση δεδομένων του Διαχειριστή Κλειδαρίθμων «MyPasswords».

Περιγραφή:

Ο MyPasswords είναι ένας δωρεάν, πολύ-ελαφρύς, ασφαλής και εύκολος στη χρήση Διαχειριστής Κλειδαρίθμων. Αρχικά δημιουργήθηκε για το λειτουργικό σύστημα Linux αλλά στη συνέχεια κυκλοφόρησαν εκδόσεις και για τα Windows, Mac, Unix και Solaris. Ο MyPasswords έχει επιλεγεί ως ο καλύτερος Διαχειριστής Κλειδαρίθμων για το λειτουργικό σύστημα Linux από το περιοδικό LinuxFormat και έχει αποσπάσει βαθμολογία 10/10.

Το καλύτερο πράγμα σχετικά με τον MyPasswords είναι ότι δεν απαιτεί εγκατάσταση (τρέχει κατευθείαν από το εκτελέσιμο αρχείο) και είναι εξαιρετικά γρήγορος. Βασίζεται στη βάση δεδομένων Apache Derby (πολύ ελαφριά, ανοικτού κώδικα σχεσιακή βάση δεδομένων, υλοποιημένη εξολοκλήρου σε Java) και στον αλγόριθμο AES για την κρυπτογράφησή της, έτσι ώστε να δημιουργήσει έναν ασφαλή χώρο αποθήκευσης των διαπιστευτηρίων πρόσβασης. Η διεπαφή του είναι πολύ απλή και κατανοητή. Τα πεδία της φόρμας εισαγωγής διαπιστευτηρίων είναι τα ίδια για όλες τις κατηγορίες, με τη διαφορά ότι για την κάθε καταχώρηση υπάρχει η δυνατότητα προσθήκης «έξυπνων» ετικετών (έτσι ώστε γράφοντας έπειτα κατά την αναζήτηση το όνομα μιας ετικέτας, να εμφανίζονται όλες οι εγγραφές που περιέχουν την ετικέτα αυτή). [15]



Εικόνα 1.11 - Το παράθυρο πλοήγησης στις εγγραφές της βάσης δεδομένων βάσει κατηγοριών (αριστερά). Οι κατηγορίες δημιουργούνται αυτόματα βάσει των ετικετών που περιλαμβάνει η κάθε εγγραφή. Το παράθυρο αναζήτησης εγγραφής βάση τίτλου ή βάση ετικέτας (δεξιά) του Διαχειριστή Κλειδαρίθμων «MyPasswords».

Δυνατότητες:

- Εντελώς αυτόνομος (portable).
- Πολύ ελαφρύς.
- Πολύ απλό περιβάλλον διαχείρισης.
- Βασιζόμενος στην επίσης ελαφριά βάση δεδομένων Derby.
- Περιβάλλον διαχείρισης βασιζόμενο σε Java Swing.
- Υψηλής Ασφάλειας (AES).
- Εισαγωγή και εξαγωγή διαπιστευτηρίων από και σε XML αρχεία.
- Αυτόματη συμπλήρωση μέσω CTRL + SPACE.
- Γεννήτρια κωδικών πρόσβασης.
- Αναλυτής ανθεκτικότητας κωδικών πρόσβασης.
- Κρυπτογράφηση αρχείου και κειμένου.
- Ανίχνευση αδράνειας.

Ασφάλεια:

- Κρυπτογράφηση βάσης δεδομένων μέσω αλγορίθμου AES.

Επίσημη ιστοσελίδα: **http://mypasswords7.com/**
http://sourceforge.net/projects/mypasswords7/

ΚΕΦΑΛΑΙΟ 2

Επιθέσεις, Υποκλοπή Δεδομένων και Τρόποι Προστασίας

2.1 Έρευνα στους χρήστες του Πανεπιστημίου Θεσσαλίας σχετικά με την διαχείριση Διαπιστευτηρίων Πρόσβασης

Στο προηγούμενο κεφάλαιο συζητήθηκαν εκτενώς οι Διαχειριστές Κλειδαρίθμων και αναλύθηκαν τα θέματα ασφαλείας που μπορεί να δημιουργηθούν κατά τη χρήση τους. Παρουσιάστηκαν επίσης διάφορες τεχνικές για την επίλυση αυτών των ευπαθειών σύμφωνα με τις οποίες οι Διαχειριστές Κλειδαρίθμων θα μπορούσαν να γίνουν αδιάρρηκτοι. Τι γνώμη έχουν όμως οι χρήστες για τους Διαχειριστές Κλειδαρίθμων και για την λειτουργικότητα που προσφέρουν; Κατά πόσο είναι διατεθειμένοι να εκτελούν περισσότερες διαδικασίες πιστοποίησης από τις τετριμμένες έτσι ώστε να διατηρήσουν την ασφάλεια των διαπιστευτηρίων τους στο μέγιστο επίπεδο;

Τι απαντήσεις στα παραπάνω ερωτήματα έδωσαν 230 άτομα του Πανεπιστημίου Θεσσαλίας (ανεξαρτήτου τμήματος) απαντώντας στο ηλεκτρονικό ερωτηματολόγιο το οποίο συντάχθηκε και τους διανεμήθηκε μέσω email ειδικά για τους σκοπούς αυτής της διατριβής. Το ερωτηματολόγιο αποτελούταν από 7 ερωτήσεις συν 1 για σχόλια γενικού περιεχομένου οι οποίες ήταν σχετικές με την ασφάλεια των προσωπικών δεδομένων πρόσβασης των χρηστών και με την χρήση των Διαχειριστών Κλειδαρίθμων. Πιο αναλυτικά, οι ερωτήσεις με τις απαντήσεις τους και τα αποτελέσματα που δόθηκαν, εμφανίζονται από κάτω.

2.1.1 Ερωτηματολόγιο

Ερώτηση 1:

Κάνετε εγγραφή (registration) σε κάποιο web site και επιλέγετε υποχρεωτικά έναν κωδικό πρόσβασης. Στην συνέχεια, για να θυμόσαστε τον κωδικό που επιλέξατε στο μέλλον, τι ενέργειες κάνετε συνήθως; (Επιλέξτε 1 ως 3 απαντήσεις)

Απαντήσεις	Ποσοστό
Τον γράφω σε ένα χαρτάκι το οποίο αφήνω μόνιμα πάνω στο γραφείο μου ή το κολλάω πάνω στο πλαίσιο της οθόνης για να τον βλέπω εύκολα.	**7.6%**
Τον γράφω στην τελευταία σελίδα από ένα τετράδιο που χρησιμοποιώ για σημειώσεις για να μην τον δει κανείς.	**14.0%**
Τον γράφω σε ένα txt αρχείο (απλό αρχείο κειμένου notepad) και αποθηκεύω κάπου το αρχείο.	**7.3%**

Τον γράφω σε κάποιο αρχείο Word ή Excel, μαζί με άλλους κωδικούς που έχω, το οποίο έχω κλειδώσει επίσης με κωδικό πρόσβασης.	**7.6%**
Τον γράφω στον τηλεφωνικό κατάλογο του κινητό μου σαν επαφή.	**3.5%**
Τον γράφω στο κινητό μου στις σημειώσεις ή σαν sms στα πρόχειρα.	**9.4%**
Δεν κάνω κάτι γιατί τον αποθηκεύει αυτόματα ο browser μου (πατάω στο πλαίσιο που μου βγάζει και γράφει "Να αποθηκευτεί ο κωδικός μου").	**12.6%**
Χρησιμοποιώ ειδικό πρόγραμμα διαχείρισης κλειδαρίθμων (password manager) που έχω εγκατεστημένο στον υπολογιστή μου ή/και στο κινητό μου και τους αποθηκεύω εκεί.	**4.7%**
Δεν τον σημειώνω κάπου. Χρησιμοποιώ έναν συγκεκριμένο κωδικό σε όλα τα web sites που κάνω εγγραφή και έτσι τον θυμάμαι πάντα απ έξω.	**33.3%**

Ερώτηση 2:

Την στιγμή που πρέπει να επιλέξετε κωδικό (κατά την εγγραφή σας σε ένα web site), ποια μέθοδο χρησιμοποιείτε; (Επιλέξτε 1 απάντηση)

Απαντήσεις	Ποσοστό
Βάζω κάτι εύκολο για να το θυμάμαι πάντα, όπως: ημερομηνία γενεθλίων, το PIN του κινητού μου, το όνομα του ατόμου με το οποίο έχω σχέση, την ημερομηνία γάμου μου ή κάτι σχετικό.	**25.6%**
Χρησιμοποιώ κάτι από τα παραπάνω αλλά προσθέτω ανάμεσά τους και κάποιον αριθμό ή σύμβολο.	**50.0%**
Βάζω τυχαία γράμματα ή/και αριθμούς.	**7.0%**
Βάζω τυχαία γράμματα, αριθμούς και σύμβολα.	**12.2%**
Χρησιμοποιώ ειδικά προγράμματα ή web sites τα οποία λειτουργούν ως γεννήτριες κωδικών (password generators) και μου σχηματίζουν έναν κωδικό ανάλογα με τα κριτήρια μου (με γράμματα, νούμερα, σύμβολα κ.τ.λ.).	**5.2%**

Ερώτηση 3:

Το μήκος του κωδικού μου είναι συνήθως: (Επιλέξτε 1 απάντηση)

Απαντήσεις	Ποσοστό
2 ως 4 ψηφία.	**0.0%**
5 ως 6 ψηφία.	**14.4%**
7 ως 8 ψηφία.	**34.8%**
9 ως 12 ψηφία.	**40.4%**
Πάνω από 12 ψηφία.	**10.4%**

Ερώτηση 4:

Έχετε πέσει ποτέ θύμα υποκλοπής κωδικού; (Επιλέξτε 1 απάντηση)

Απαντήσεις	Ποσοστό
Όχι	**81.0%**
Ναι	**19.0%**

Ερώτηση 5:

Αν απαντήσατε "Ναι" στην προηγούμενη ερώτηση, δώστε κάποιες λεπτομέρειες για το τι πιστεύετε ότι πήγε στραβά και σας υποκλέψανε: (Επιλέξτε όσες απαντήσεις θέλετε)

(Τα ποσοστά προέκυψαν λαμβάνοντας υπ όψιν ότι το 100% είναι οι 43 χρήστες που απάντησαν «Ναι» στην προηγούμενη ερώτηση)

Απαντήσεις	Ποσοστό
Είχα (αποδεδειγμένα) κάποιο malware στον υπολογιστή μου (virus, spyware, trojan horse, κ.α.) το οποίο έκανε την υποκλοπή και το οποίο εντοπίστηκε από κάποιο λογισμικό antivirus αργότερα.	**17.0%**

Έπεσα θύμα απάτης ηλεκτρονικού ψαρέματος (phishing scam). Το web site που μπήκα για να κάνω Log-in φαινόταν να είναι το σωστό εμφανισιακά, αλλά έτρεχε από άλλη διεύθυνση χωρίς να το παρατηρήσω και εξαπατήθηκα.	**10.7%**
Έπεσα θύμα "προσχήματος" (pretexting), δηλαδή: κάποιος μου έκανε προσωπικές ερωτήσεις για να συμμετάσχω σε έναν διαγωνισμό (ή κάτι αντίστοιχο), εγώ του απάντησα και μέσα από τις απαντήσεις μου ανακάλυψε τον κωδικό μου.	**4.2%**
Κάποιος παρακολουθούσε τον υπολογιστή μου έχοντας εγκαταστήσει σε αυτόν πρόγραμμα καταγραφής πληκτρολογήσεων (keylogger).	**8.5%**
Κάποιος διάβασε τα χαρτάκια ή το τετράδιο που σημείωνα τους κωδικούς μου.	**2.1%**
Μάλλον κάποιος μάντεψε τον κωδικό μου γιατί είχα βάλει κάτι γνωστό (ημερομηνία γέννησης κ.α.).	**27.7%**
Μου έκλεψαν το κινητό μου και εκεί μέσα είχα τους κωδικούς μου υπό μορφή επαφών, sms ή σημειώσεων και κάποιος τους διάβασε.	**0.0%**
Είχα κάνει Log-in από τον υπολογιστή της βιβλιοθήκης του πανεπιστημίου μου (ή από κάποιο internet cafe ή από τον υπολογιστή ενός φίλου) και ξέχασα να κάνω Log-out και έτσι μου κλέψανε των κωδικό μου.	**14.9%**
Άλλο.	**14.9%**

Άλλο: (σχόλια χρηστών)
Κάποιος παρακολουθούσε την κίνηση των πακέτων δεδομένων στο δίκτυο και αποκρυπτογράφησε τον κωδικό μου.
Παρακολουθούσαν συστηματικά την πληκτρολόγηση στον υπολογιστή του χώρο εργασίας μου και μετά από δοκιμές κατέληξαν στον σωστό κωδικό.
Μαζική υποκλοπή κωδικών πρόσβασης από τη βάση δεδομένων μιας ιστοσελίδας από hackers.
Έκανα Log-in από το κινητό μου μέσω μη προστατευμένου δικτύου Wi-Fi, τη στιγμή που άλλη συσκευή του δικτύου κατέγραφε τους κωδικούς.
Κάποιος είδε τη διεύθυνση email που εμφανιζόταν στο προφίλ μου στο κοινωνικό δίκτυο που χρησιμοποιούσα και έτσι βρήκε τον κωδικό μου.
Έκανα Log-in από κάποιον υπολογιστή που είχε malware.
Δε ξέρω πως έγινε, δεν έγινε τίποτα από τα παραπάνω.

Ερώτηση 6:

Εάν στην "Ερώτηση 1" έχετε επιλέξει ότι χρησιμοποιείτε κάποιον διαχειριστή κλειδαρίθμων για την αποθήκευση των κωδικών σας, ποιοι παράγοντες σας οδήγησαν να τον χρησιμοποιήσετε? (Επιλέξτε όσες απαντήσεις θέλετε)

(Τα ποσοστά προέκυψαν λαμβάνοντας υπ όψιν ότι το 100% είναι οι 47 απαντήσεις που δόθηκαν συνολικά στην ερώτηση)

Απαντήσεις	Ποσοστό
Οργάνωση. Θέλω να έχω όλα τα δεδομένα πρόσβασής μου (κωδικοί πρόσβασης, ονόματα χρήστη, PINs) αποθηκευμένα σε ένα μόνο σημείο και τακτοποιημένα ανά κατηγορία.	**46.8%**
Κρυπτογράφηση. Θέλω όλα τα δεδομένα μου να είναι αποθηκευμένα σε κάποια βάση δεδομένων υπό μορφή κρυπτογράφησης ούτως ώστε να μην μπορεί να τα διαβάσει κάποιος άλλος χρήστης ή κάποιο κακόβουλο πρόγραμμα αν καταφέρει να τα υποκλέψει.	**31.9%**
Αυτοματοποίηση διαδικασιών. Θέλω μέσα από ένα πρόγραμμα να παράγεται αυτόματα ένας ισχυρός κωδικός πρόσβασης και να τοποθετώ αυτόν κάθε φορά κατά την εγγραφή μου σε κάποιο web site, αντί να σκέφτομαι κάποιον εγώ.	**21.3%**
Άλλοι λόγοι.	**0.0%**

Ερώτηση 7:

Αν χρησιμοποιήσατε κάποια στιγμή έναν διαχειριστή κλειδαρίθμων, και μετά από κάποιο διάστημα σταματήσατε να τον χρησιμοποιείτε ή τον απεγκαταστήσατε, ποιοι παράγοντες σας οδήγησαν να το κάνετε? (Επιλέξτε όσες απαντήσεις θέλετε)

(Τα ποσοστά προέκυψαν λαμβάνοντας υπ όψιν ότι το 100% είναι οι 40 απαντήσεις που δόθηκαν συνολικά στην ερώτηση)

Απαντήσεις	Ποσοστό
Βαρετή και χρονοβόρα διαδικασία. Με κούραζε να ανοίγω συνέχεια κάποιο άλλο πρόγραμμα και να εισάγω τους κωδικούς μου εκεί, όπως και το αντίθετο όταν ήθελα να διαβάσω κάποιον κωδικό για να κάνω Log-in σε κάποιο web site.	**25.0%**

Δύσκολος στην χρήση και δυσνόητος. Το μενού, οι επιλογές και ο τρόπος για να αποθηκεύσω τα δεδομένα πρόσβασής μου ή να τα διαβάσω, ήταν πολύπλοκος και δυσνόητος.	**10.0%**
Βαρύ πρόγραμμα. Κάθε φορά φόρτωνε αυτόματα κατά την εκκίνηση του υπολογιστή ή του κινητού μου και χρησιμοποιούσε πολλούς πόρους από αυτό (μνήμη, επεξεργαστική ισχύ). Έκανε τον υπολογιστή μου ή το κινητό μου να καθυστερεί και να κολλάει (κατά την εκκίνηση ή γενικά) και να καταναλώνει περισσότερη μπαταρία.	**12.5%**
Ενοχλητικό πρόγραμμα. Κάθε φορά που πλοηγούμουνα σε κάποιο web site το οποίο είχε πεδία εισαγωγής κωδικών, μου εμφάνιζε κάποιο "συννεφάκι" ή κάποιο μήνυμα και με ρωτούσε συνέχεια αν θέλω να αποθηκεύσω τον κωδικό μου ή όχι και με ενοχλούσε.	**27.5%**
Έλλειψη φορητότητας: Δεν μπορούσα να αντιγράψω την βάση δεδομένων του προγράμματος σε κάποιο φλασάκι ή στην cloud υπηρεσία στην οποία είμαι συνδρομητής-τρια για να μπορέσω έπειτα να την χρησιμοποιήσω από άλλο μηχάνημα (από άλλον υπολογιστή ή κινητό).	**22.5%**
Άλλοι λόγοι.	**2.5%**

Άλλοι λόγοι: (σχόλια χρηστών)
Είχα εγκαταστήσει μία επέκταση στον περιηγητή μου, αλλά σκέφτηκα ότι αν για οποιονδήποτε λόγο σταματούσε να λειτουργεί (ή έπαυα να έχω πρόσβαση σε αυτή), πρακτικά θα έχανα όλους μου τους κωδικούς.

<u>Ερώτηση 8:</u>

Συμπληρωματικά σχόλια που έχετε να κάνετε σχετικά με την χρήση κωδικών, την ασφάλεια των δεδομένων πρόσβασης ή τα προγράμματα διαχείρισης κλειδαρίθμων.

(σχόλια χρηστών)
Δεν υπάρχει επαρκής πληροφόρηση σχετικά με την ασφάλεια στα εργαλεία Διαχείρισης Κλειδαρίθμων.
Για να αυξήσω το επίπεδο της ασφάλειάς μου, βάζω σε κάθε λογαριασμό email και σε κάθε ιστοσελίδα κοινωνικής δικτύωσης διαφορετικό κωδικό και τους αλλάζω ανά 2 μήνες.

2.1.2 Σχολιασμός αποτελεσμάτων

Τα αποτελέσματα της έρευνας θα μπορούσαμε να πούμε ότι είναι κάπως ανησυχητικά για την εποχή που διανύουμε, όσον αφορά τουλάχιστον τον τρόπο που οι χρήστες επιλέγουν και διαχειρίζονται τους κωδικούς πρόσβασής τους. Ξεκινώντας από τις απαντήσεις της **Ερώτησης 1** και από το γεγονός ότι το μεγαλύτερο ποσοστό των χρηστών χρησιμοποιεί μόνο έναν, ίδιο κωδικό πρόσβασης, για όλες τις ιστοσελίδες στις οποίες έχει λογαριασμό, συμπεραίνουμε ότι οι χρήστες επιθυμούν διαδικασίες οι οποίες θα είναι όσο το δυνατόν πιο απλές και χρηστικές, κάνοντας της ζωή τους εύκολη, αλλά από την άλλη, ίσως να μη γνωρίζουν το πόσο επικίνδυνη για τον εαυτό τους και την ασφάλειά τους είναι αυτή η επιλογή τους.

Το φαινόμενο της επαναχρησιμοποίησης των ίδιων κωδικών αυξάνει κατά πολύ τον κίνδυνο παραβίασης και εκμετάλλευσης των λογαριασμών του χρήστη από κάποιον εισβολέα, αφού αν ανακαλυφθεί ο κωδικός αυτός, τότε αυτόματα ο εισβολέας μπορεί να αποκτήσει πρόσβαση σε όλες τις ιστοσελίδες του χρήστη. Ο εισβολέας μπορεί να στείλει και να διαβάσει e-mails από τον λογαριασμό του χρήστη, μπορεί να εκτελέσει χρηματικές συναλλαγές μέσω της σελίδας της τράπεζάς του προς το συμφέρον του, μπορεί να του υποκλέψει και να κοινοποιήσει στο Διαδίκτυο ευαίσθητες προσωπικές πληροφορίες ή οπτικοακουστικό υλικό (το οποίο έχει πιθανώς αποθηκευμένο σε κάποιο κέντρο δεδομένων του Διαδικτύου ή σε κάποιο κοινωνικό δίκτυο) ή ακόμα και να εκβιάσει τον χρήστη με αντάλλαγμα χρηματική ή άλλου είδους αποζημίωση.

Παρόμοιο ποσοστό κινδύνου μπορεί να επιφέρει και η επιλογή ενός κωδικού πρόσβασης ο οποίος περιλαμβάνει μία ευρέως διαδεδομένη πληροφορία, σχετική με τον χρήστη, όπως η ημερομηνία των γενεθλίων του που αναφέρθηκε στην **Ερώτηση 2**. Το 25.6% απάντησε ότι χρησιμοποιεί τέτοιους εύκολα προβλέψιμους κωδικούς, ενώ το 50% απάντησε ότι χρησιμοποιεί τέτοιους αλλά αναμειγμένους με κάποιου είδους χαρακτήρα (αριθμό ή σύμβολο). Σίγουρα η ανάμειξη με κάποιον αριθμό ή σύμβολο αυξάνει κατακόρυφα την ασφάλεια, αλλά δεν μπορεί και πάλι να την φτάσει στο μέγιστο επίπεδο, αφού ο κωδικός πρόσβασης συνεχίζει να περιέχει μέσα του γνωστές φράσεις, οι οποίες διευκολύνουν μια πιθανή, προσαρμοσμένη επίθεση λεξικού.

Στην **Ερώτηση 3** το μεγαλύτερο (συνδυασμένο) ποσοστό (75.2%) απάντησε ότι χρησιμοποιεί κωδικούς μήκους 7 ως 12 ψηφίων πράγμα πολύ καλό από άποψη ασφάλειας, αρκεί οι κωδικοί αυτοί να είναι όσο το δυνατών τυχαίοι και να έχουν ανάμειξη με σύμβολα και αριθμούς. Στην **Ερώτηση 4** το 81% απάντησε ότι δεν έχει πέσει ποτέ θύμα υποκλοπής κωδικού πρόσβασης. Αυτό σημαίνει πρακτικά ότι υπάρχει μια γενική ενημέρωση, ή καλύτερα κάποια «διαίσθηση» από τους χρήστες για το πώς να προστατεύουν τα διαπιστευτήριά τους από πιθανή υποκλοπή (χρήση αντιϊκών προγραμμάτων στον υπολογιστή τους, αποφυγή πληκτρολόγησης κωδικών σε ενδεχομένως επικίνδυνα συστήματα, κ.τ.λ.). Το υπόλοιπο 19% των χρηστών που δήλωσαν ότι έπεσαν θύμα υποκλοπής υποστηρίζει στην **Ερώτησης 5** ότι αυτό συνέβη κυρίως λόγω της απλότητας των κωδικών που χρησιμοποιούσαν (χρήση κωδικών που

αποτελούνταν από γνωστές λέξεις κατά 27.7%), οι οποίοι μαντεύθηκαν εύκολα από κακόβουλους χρήστες. Σε μικρότερα ποσοστά αλλά αξιοσημείωτα κυμάνθηκαν οι υποκλοπές λόγω ύπαρξης κακόβουλων προγραμμάτων παρακολούθησης σε μηχανήματα ή λόγω απερισκεψιών των χρηστών. Άξια προσοχής είναι επίσης και τα προσωπικά σχόλια των χρηστών που καταγράφηκαν σε αυτήν την ερώτηση σχετικά με το πώς έγινε ή το πώς πιστεύουν ότι έγινε η υποκλοπή.

Τέλος, από τις **Ερωτήσεις 6 και 7** οι οποίες ήταν σχετικές με τη χρήση των Διαχειριστών Κλειδαρίθμων βγαίνει το συμπέρασμα ότι οι χρήστες τους χρησιμοποιούν κυρίως για την οργάνωση των διαπιστευτηρίων που προσφέρουν, αλλά το γεγονός ότι η χρήση τους αποτελεί χρονοβόρα διαδικασία, το ότι δεν παρέχουν φορητότητα (portability) του ίδιου του προγράμματος ή της βάσης δεδομένων τους και το γεγονός ότι πολλές φορές τους ενοχλούν με διάφορα μηνύματα για το αν θέλουν αν αποθηκεύσουν τους κωδικούς τους ή όχι, τους απωθούν από το να τους βάλουν στην καθημερινότητά τους.

2.2 Τα κυριότερα είδη Επιθέσεων με στόχο την υποκλοπή προσωπικών δεδομένων και το Κακόβουλο Λογισμικό

2.2.1 Προγράμματα Καταγραφής Πληκτρολογήσεων και Προγράμματα Κατασκοπίας

Keyloggers - Spyware	
Περιγραφή:	Πρόκειται για κακόβουλο λογισμικό το οποίο έχει σχεδιαστεί για την **παρακολούθηση** και την **καταγραφή** των ενεργειών και των δεδομένων ενός χρήστη ενός λειτουργικού συστήματος. Στην πιο απλή μορφή τους χρησιμοποιούνται για τη καταγραφή ονομάτων χρήστη και κωδικών πρόσβασης ενώ σε πιο σύνθετα σενάρια μπορεί να χρησιμοποιηθούν και για παρακολούθηση των ενεργειών τους. Τα περισσότερα προγράμματα αυτού του είδους τρέχουν κρυφά (στο παρασκήνιο) του λειτουργικού συστήματος, με συνέπεια ο χρήστης να μην ειδοποιείται ποτέ και να μην μπορεί να καταλάβει ότι οι ενέργειές του καταγράφονται.
Τύποι/Παραλλαγές:	Οι παραλλαγές της **καταγραφής πληκτρολογήσεων** (keylogging) διακρίνονται σε υλοποιήσεις υλικού και λογισμικού, με δυνατότητες που μπορεί να φτάσουν μέχρι και την ηλεκτρομαγνητική και ακουστική ανάλυση. Τέτοια προγράμματα όχι μόνο μπορούν να καταγράψουν δεδομένα που υπάρχουν στο λειτουργικό σύστημα ενός μηχανήματος (ή να παρακολουθήσουν το πληκτρολόγιο ενός χρήστη), μπορούν να παρακολουθήσουν και όλες τις άλλες συσκευές που βρίσκονται συνδεδεμένες στο μηχάνημα αυτό, όπως για παράδειγμα μια συσκευή ανάγνωσης καρτών πληρωμής. Τα **προγράμματα κατασκοπίας** (spyware), επιπλέον της καταγραφής πληκτρολογήσεων, μπορούν να κάνουν χρήση και του πολυμεσικού υλικού ενός μηχανήματος (όπως web κάμερα, μικρόφωνο, στιγμιότυπα οθόνης, κ.α.) για να κατασκοπευόσουν τον χρήστη πιο ολοκληρωμένα. Οι πιο συνηθισμένες μορφές τέτοιων προγραμμάτων κάνουν καταγραφή των πληροφοριών τους και τις αποστέλλουν άμεσα (μέσω Διαδικτύου) στον χρήστη που επιτελεί την

	κατασκοπεία, ενώ υπάρχουν και οι άλλες, οι οποίες αποθηκεύουν τοπικά, σε κάποιο σημείο του μηχανήματος τις πληροφορίες που έχουν συλλέξει για μεταγενέστερη ανάκτηση από τον επιτιθέμενο.
Μέθοδοι μόλυνσης:	Η μόλυνση ενός λειτουργικού συστήματος από προγράμματα καταγραφής και κατασκοπίας γίνεται κυρίως μέσα από την περιήγηση του χρήστη στο διαδίκτυο και τη **λήψη μολυσμένων αρχείων** (μπορεί να συμβεί και με εικονικά «νόμιμο» τρόπο, κατεβάζοντας ένα πρόγραμμα το οποίο παραβιάζει την πολιτική χρήσης που έχει διαβάσει και αποδεχθεί νωρίτερα ο χρήστης). Επίσης μπορεί να γίνει και μέσα από κάποιο μολυσμένο συνημμένο αρχείο ενός email, μέσω κάποιου απομακρυσμένου επιτιθέμενου ο οποίος έχει ήδη αποκτήσει πρόσβαση στο σύστημα του χρήστη με κάποιον τρόπο και τοποθετεί την απειλή, ή τέλος, μέσω της παρεμβολής SQL (SQL Injection). Να σημειωθεί ότι οι εκδόσεις των προγραμμάτων αυτών που βασίζονται στην παρακολούθηση μέσω υλικού, απαιτούν φυσική πρόσβαση στο μηχάνημα από τον επιτιθέμενο, πράγμα που κάνει αυτήν την μορφή πιο δύσκολη και λιγότερο χρησιμοποιούμενη.
Στόχοι:	Εγκαθίστανται σε μηχανήματα τελικού χρήστη ή σε εξυπηρετητές (servers). Κύριος στόχος τους είναι η κλοπή δεδομένων πιστοποίησης (διαπιστευτηρίων) για εφαρμογές και υπηρεσίες απομακρυσμένης πρόσβασης. Δευτερεύον στόχος είναι η κλοπή προσωπικών πληροφοριών του χρήστη και άλλου τύπου δεδομένων.

2.2.2 Κερκόπορτες, Τρωικά Άλογα και Προγράμματα Εντολής/Ελέγχου

Backdoors - Trojan Horses - Command/Control Programs	
Περιγραφή:	Πρόκειται για προγράμματα που **παρέχουν απομακρυσμένη πρόσβαση και έλεγχο εξ αποστάσεως** σε μολυσμένα συστήματα. Τα προγράμματα αυτά παρακάμπτουν τους μηχανισμούς πιστοποίησης και τους υπόλοιπους μηχανισμούς ασφαλείας ενός λειτουργικού συστήματος και σχεδιάζονται έτσι ώστε να δρουν κρυφά (χωρίς να γίνονται αντιληπτά από τον χρήστη).
Τύποι/Παραλλαγές:	Πιο κοινό τύπο αυτών των προγραμμάτων αποτελεί το «άγνωστο λογισμικό» (το οποίο το κατεβάζουμε κατά λάθος ή εν γνώσει μας χωρίς να γνωρίζουμε τι κάνει, μπορεί να κατεβεί χωρίς να το καταλάβουμε, μπορεί να βρίσκεται μέσα σε κάποιο email ως συνημμένο μολυσμένο αρχείο ή να έχει μπει στον υπολογιστή μας από τρίτους). Επίσης μπορεί να αποτελεί τμήμα (κρυφής) λειτουργικότητας πειρατικών προγραμμάτων τα οποία είναι κατάλληλα πειραγμένα έτσι ώστε, παράλληλα με την υπάρχουσα λειτουργικότητά τους, να κρατάνε κρυφά ανοικτή και κάποια «πίσω πόρτα» για την πρόσβαση στο σύστημα από τρίτους.
Μέθοδοι μόλυνσης:	Συνήθως εγκαθίσταται από έναν απομακρυσμένο εισβολέα μετά την απόκτηση πρόσβασής του στο σύστημα ή μπορεί να κατεβεί στο ίδιο σύστημα μέσω παρεμβολής SQL (SQL injection). Επίσης η περιήγηση στο Web (ενίοτε θεμιτά, αλλά ως επί το πλείστον μέσω παραβίασης των πολιτικών χρήσης) μπορεί να αποτελέσει παράγοντα μόλυνσης. Πολύ συχνά συναντώνται και σε συνδυασμό με άλλα είδη κακόβουλου λογισμικού, ειδικά με τους υποκλοπείς πακέτων δεδομένων του δικτύου (packet sniffers), μιας και οι επιτιθέμενοι χρειάζονται τα backdoors για την απόκτηση των κλεμμένων δεδομένων που έχουν συλλεχθεί.
Στόχοι της επίθεσης:	Συνήθως εγκαθίστανται σε εξυπηρετητές (servers) με τελικό στόχο να γίνουν αυτά τα προγράμματα τα οποία θα επεξεργάζονται, θα αποθηκεύουν και

	θα μεταδίδουν ευαίσθητα δεδομένα και θα παρέχουν κάποιο πλεονέκτημα στον εισβολέα (δηλαδή, την ευκαιρία να κλιμακώσει ή να παρατείνει την επίθεσή του). Σπανιότερα επίσης, μπορεί να βρεθούν και σε συστήματα τελικού χρήστη.

2.2.3 Υποκλοπή δεδομένων μέσω Παρεμβολής SQL

SQL Injection	
Περιγραφή:	Η **παρεμβολή SQL** (Structured Query Language Injection) είναι μια τεχνική επίθεσης η οποία εκμεταλλεύεται τον τρόπο με τον οποίο οι ιστοσελίδες επικοινωνούν με το λογισμικό διαχείρισης των βάσεων δεδομένων. Κάποιος που θέλει να επιτεθεί με παρεμβολή SQL (για παράδειγμα μέσω μιας ιστοσελίδας) μπορεί απλά να συντάξει ερωτήματα SQL, να τα τοποθετήσει στα πλαίσια εισαγωγής δεδομένων (text boxes) της ιστοσελίδας και πατώντας το κουμπί υποβολής (submit) να τα στείλει στο σύστημα διαχείρισης της βάσης δεδομένων. Αναλόγως την σύνταξη που θα έχουν τα ερωτήματα αυτά, ο εισβολέας μπορεί να πετύχει και τα ανάλογα αποτελέσματα.
Τύποι/Παραλλαγές:	Η παρεμβολή SQL έχει τρεις κύριες χρήσεις: 1) Χρησιμοποιείται για την υποβολή ερωτημάτων προς την βάση δεδομένων και την επιστροφή πληροφοριών από αυτή, 2) Χρησιμοποιείται για τροποποίηση των υπαρχόντων δεδομένων μιας βάσης δεδομένων, και 3) Χρησιμοποιείται για προτροπή του εξυπηρετητή (server) έτσι ώστε να κατεβάσει κακόβουλο λογισμικό από απομακρυσμένες τοποθεσίες. Η ευελιξία και η αποτελεσματικότητα της παρεμβολής SQL λειτουργεί ως «πολυ-εργαλείο»

	για του εγκληματίες του κυβερνοχώρου.
Μέθοδοι μόλυνσης:	Η παρεμβολή SQL αν και αποτελεί μια αυτόνομη μέθοδο επίθεσης, συχνά χρησιμοποιείται και σε συνδυασμό με άλλες τεχνικές για να εισάγει κακόβουλο λογισμικό στο περιβάλλον του θύματος. Μπορεί να συνδυαστεί με υποκλοπείς πακέτων δεδομένων του δικτύου (packet sniffers), με προγράμματα backdoors, με καταγραφείς πληκτρολογήσεων (keyloggers) και με προγράμματα κατασκοπείας (spyware). Αυτού του είδους οι επιθέσεις συμβαίνουν μέσω εφαρμογών ή μέσω φορμών καταχώρησης δεδομένων ιστοσελίδων.
Στόχοι της επίθεσης:	Ο στόχος των επιθέσεων της παρεμβολής SQL είναι οι εξυπηρετητές βάσεων δεδομένων, ιδίως εκείνοι που αποθηκεύουν ευαίσθητα δεδομένα ή που βρίσκονται σε περιβάλλον δικτύου το οποίο περιέχει ευαίσθητα δεδομένα. Η παρεμβολή SQL χρησιμοποιείται για την υποκλοπή όλων των ειδών δεδομένων, αλλά πιο συχνά συνδέεται με την υποκλοπή δεδομένων καρτών πληρωμής και προσωπικών διαπιστευτηρίων χρηστών.

2.2.4 Διείσδυση σε ένα σύστημα μέσω Κακόβουλης Χρήσης των Δικαιωμάτων Πρόσβασης

Abuse of System Access/Privileges	
Περιγραφή:	Σκόπιμη και συνήθως κακόβουλη χρήση των πόρων ενός συστήματος, της πρόσβασης ή των δικαιωμάτων χρήσης που χορηγούνται σε ένα άτομο από έναν οργανισμό.
Τύποι/Παραλλαγές:	Ποικίλλει ανάλογα με το βαθμό πρόσβασης και των δικαιωμάτων χρήσης που παρέχονται σε έναν χρήστη και το είδος των πόρων (συνδεδεμένων με φυσικό, λογικό ή δικτυακό τρόπο).
Μέθοδοι μόλυνσης:	Η φύση αυτής της απειλής είναι τέτοια ώστε να είναι επαρκής για να επιτευχθεί ο στόχος της χωρίς κάποια άλλη βοήθεια. Αν κάποιος δηλαδή έχει ήδη

	στην κατοχή του τα διαπιστευτήρια για προνομιακή πρόσβαση σε ένα σύστημα, τότε δεν χρειάζεται μεθόδους προαγωγής των δικαιωμάτων χρήσης του ή μεθόδους για την παράκαμψη των ελέγχων ασφαλείας για να εγκαταστήσει ένα κακόβουλο λογισμικό ή να τελέσει κάποια υποκλοπή δεδομένων.
Στόχοι της επίθεσης:	Οποιαδήποτε συστήματα και οποιαδήποτε δεδομένα. Μπορεί να χρησιμοποιηθεί για να θέσει σε κίνδυνο όλες τις μορφές των δεδομένων, αλλά πιο συχνά στοχεύει διευθύνσεις IP και άλλες εταιρικές πληροφορίες και όχι για παράδειγμα στοιχεία καρτών πληρωμής.

2.2.5 Μη-Εξουσιοδοτημένη Πρόσβαση σε ένα σύστημα με χρήση Προεπιλεγμένων Διαπιστευτηρίων

Unauthorized Access via Default Credentials	
Περιγραφή:	Αναφέρεται σε περιπτώσεις στις οποίες ένας εισβολέας μπορεί να αποκτήσει πρόσβαση σε ένα σύστημα (ή μία συσκευή) το οποίο **χρησιμοποιεί ένα προκαθορισμένο και ως εκ τούτου, ευρέως διαδεδομένο πρότυπο (ή αλγόριθμο) για την ονοματοδοσία των χρηστών και την παραγωγή κωδικών πρόσβασης.**
Τύποι/Παραλλαγές:	Τα προεπιλεγμένα διαπιστευτήρια των χρηστών ποικίλλουν ανάλογα με το άτομο ή το πρόγραμμα που τα παράγει, αλλά ο τρόπος της επίθεσης για την εκμετάλλευσή τους, αν εκτεθούν οι κανόνες παραγωγής τους, είναι ουσιαστικά ο ίδιος.
Μέθοδοι μόλυνσης:	Τα διαπιστευτήρια των χρηστών μπορούν να ανακαλυφθούν λόγω παραλείψεων και κακών διαμορφώσεων πιστοποίησης. Σαν επίθεση μπορεί να αποτελεί μια αυτόνομη μέθοδο διείσδυσης, αλλά ωστόσο, χρησιμοποιείται συχνά και σε συνδυασμό με άλλα προγράμματα, ειδικά με υποκλοπείς δεδομένων μνήμης RAM (RAM Scrapers), υποκλοπείς πακέτων δεδομένων του δικτύου (packet sniffers), προγραμμάτων backdoors ή

	προγραμμάτων εντολής/ελέγχου (command/ control), έτσι ώστε να εισάγει μέσω αυτών κακόβουλο λογισμικό στο περιβάλλον του θύματος. Μετά την επιτυχή διείσδυση στο σύστημα από τον εισβολέα, ακολουθείται συνήθως και η τεχνική «Μη Εξουσιοδοτημένης Πρόσβαση μέσω Ανίσχυρων ή Κακο-διαμορφωμένων Λιστών Ελέγχου Πρόσβασης» (Unauthorized Access via Weak or Misconfigured Access Control Lists) (αναλύεται παρακάτω) έτσι ώστε να μπορέσουν να αποκαλυφθούν και τα υπόλοιπα συστήματα του δικτύου. Συχνά οι επιθέσεις αυτές εκτελούνται από απόσταση, μέσω προγραμμάτων απομακρυσμένης διαχείρισης.
Στόχοι της επίθεσης:	Κύριοι στόχοι είναι οι εφαρμογές, οι εξυπηρετητές (servers) και οι συσκευές ενός δικτύου. Οι επιθέσεις αυτές έχουν ως σκοπό την αποκάλυψη μεγάλου όγκου δεδομένων καρτών πληρωμής και προσωπικών πληροφοριών χρηστών.

2.2.6 Υποκλοπή δεδομένων λόγω Παραβίασης των Αποδεκτών Πολιτικών Χρήσης

Violation of Acceptable Use and other policies	
Περιγραφή:	Οι **αποδεκτές πολιτικές χρήσης** καθορίζουν τον τρόπο με τον οποίο οι εργαζόμενοι μιας εταιρίας χρησιμοποιούν τις εταιρικές πληροφορίες της. Οι παραβιάσεις συμβαίνουν όταν ένας εργαζόμενος κατά λάθος ή εσκεμμένα αγνοήσει αυτές τις πολιτικές και εκθέσει τις εταιρικές πληροφορίες σε τρίτους ή τις χρησιμοποιήσει με κακόβουλο σκοπό.
Τύποι/Παραλλαγές:	Ποικίλλει σε μεγάλο βαθμό ανάλογα με την πρόθεση, το είδος των εταιρικών πληροφοριών που καταχρούνται, το βαθμό της κατάχρησης και το αποτέλεσμα. Η παραβίαση των αποδεκτών πολιτικών χρήσης μιας εταιρίας συμβαίνει συχνά με τη μορφή της πρόσβασης σε πορνογραφικό υλικό του διαδικτύου μέσω των υπολογιστών της εταιρίας, με χρήση προσωπικών λογαριασμών ηλεκτρονικού ταχυδρομείου για την αποστολή

	εταιρικών πληροφοριών, με αποθήκευση προσωπικών (ενίοτε και παράνομων) δεδομένων σε εταιρικά συστήματα και με τη λήψη και εγκατάσταση μη εξουσιοδοτημένου λογισμικού.
Μέθοδοι μόλυνσης:	Δεν αποτελεί την κύρια μέθοδο για την αποκάλυψη δεδομένων αλλά περισσότερο τον παράγοντα για την παραβίασή τους, οποιουδήποτε τύπου κι αν είναι αυτά. Μπορεί να οδηγήσει στην εισαγωγή και εγκατάσταση κακόβουλου λογισμικού (κυρίως keylogging και spyware) στο περιβάλλον των υπολογιστών της εταιρίας. Παρατηρείται επίσης μια συσχέτιση μεταξύ αυτών των παραβιάσεων και της τάσης στο να συνδυάζονται σε όλο και περισσότερες κακόβουλες μορφές διείσδυσης, όπως στην Κακόβουλη Χρήση των Δικαιωμάτων Πρόσβασης σε ένα σύστημα (Abuse of System Access/Privileges).
Στόχοι της επίθεσης:	Συνήθως εμπλέκονται άμεσα τα συστήματα του τελικού χρήστη, αλλά επηρεάζεται επίσης και ένα ευρύτερο φάσμα των εταιρικών πληροφοριών.

2.2.7 Μη-Εξουσιοδοτημένη Πρόσβαση σε ένα σύστημα λόγω χρήσης Ανίσχυρων ή Κακο-διαμορφωμένων Λιστών Ελέγχου Πρόσβασης

Unauthorized Access via Weak or Misconfigured ACLs	
Περιγραφή:	Οι **λίστες ελέγχου πρόσβασης** (Access Control Lists - ACLs) είναι λίστες σε ηλεκτρονική μορφή, εγκατεστημένες σε διάφορα συστήματα, οι οποίες προσδιορίζουν τους χρήστες στους οποίους θα επιτρέπεται η πρόσβαση σε ένα αντικείμενο, καθώς και τα δικαιώματα που αυτοί θα έχουν για την εκτέλεση διαφόρων ενεργειών. Αν οι λίστες αυτές απουσιάζουν, είναι αδύναμες (με μηδαμινές απαγορεύσεις), έχουν εσφαλμένη έκταση (π.χ. δίνουν δικαιώματα πρόσβασης σε μία ολόκληρη ομάδα εργασίας αντί για κάποιους επιλεγμένους χρήστες) ή είναι κακο-διαμορφωμένες, τότε οι εισβολείς μπορούν να τις εκμεταλλευτούν για να

	αποκτήσουν πρόσβαση στους πόρους ενός περιβάλλοντος και να εκτελέσουν ενέργειες που δεν θα προέρχονται από το θύμα.
Τύποι/Παραλλαγές:	Οι Λίστες Ελέγχου Πρόσβασης μπορούν να εφαρμόζονται για τον έλεγχο πρόσβασης σε συσκευές δικτύου, σε λειτουργικά συστήματα, σε διεργασίες μέσα σε ένα λειτουργικό σύστημα, σε χρήστες, σε ομάδες χρηστών, καθώς και σε κάθε είδους λειτουργίες.
Μέθοδοι μόλυνσης:	Η διείσδυση στα συστήματα μέσω αυτής της μεθόδου μπορεί να συμβεί λόγω παραλείψεων και κακών διαμορφώσεων. Η μέθοδος αυτή αν και αποτελεί μία αυτόνομη μέθοδο διείσδυσης, σε συνδυασμό με άλλα προγράμματα όπως τους υποκλοπείς δεδομένων μνήμης RAM (RAM Scrapers), τους υποκλοπείς πακέτων δεδομένων του δικτύου (packet sniffers) και τα προγράμματα backdoors ή τα προγράμματα εντολής/ελέγχου (command/control), μπορεί να εισάγει κακόβουλο λογισμικό στο περιβάλλον του θύματος.
Στόχοι της επίθεσης:	Στόχος είναι όλοι οι τύποι των εταιρικών πληροφοριών, εκτός των μορφών που βρίσκονται εκτός-σύνδεσης (offline). Δηλαδή, πιο συχνά επηρεάζονται οι συσκευές δικτύου, οι εφαρμογές και οι εξυπηρετητές (servers). Συντελούν στην αποκάλυψη ευαίσθητων δεδομένων (όπως αυτά των καρτών πληρωμής) και προσωπικών πληροφοριών των χρηστών.

2.2.8 Υποκλοπή δεδομένων με χρήση Υποκλοπέων Πακέτων Δεδομένων του Δικτύου

Packet Sniffers	
Περιγραφή:	Η αυξημένη ευαισθητοποίηση γύρω από τα ζητήματα ασφαλείας και οι αυξημένες απαιτήσεις για ποιοτικότερες ρυθμίσεις ασφαλείας, πιέζουν πολλούς οργανισμούς στο να ελαχιστοποιήσουν την διατήρηση ευαίσθητων δεδομένων στους εξυπηρετητές τους ή έστω, αν αυτό δεν είναι

	εφικτό, στο να κρυπτογραφούν όσα είναι απαραίτητο να διατηρηθούν. Οι επιτιθέμενοι χρησιμοποιούν τους **υποκλοπείς πακέτων δεδομένων δικτύου** (packet sniffers) με σκοπό να παρακάμψουν τους ελέγχους πρόσβασης στα προστατευμένα συστήματα και να συλλέξουν τα δεδομένα **κατά τη διαδικασία της μεταφοράς τους μέσω ενός δικτύου**, βασιζόμενοι στο γεγονός ότι τα δεδομένα θα βρίσκονται σε μη-κρυπτογραφημένη μορφή εκείνη τη στιγμή. Έτσι, ένας υποκλοπέας πακέτων δεδομένων, γνωστός και ως υποκλοπέας δικτύου (network sniffer) ή αναλυτής πακέτων (packet analyzer), αποτελεί ένα πρόγραμμα το οποίο παρακολουθεί και αντιγράφει δεδομένα τα οποία διασχίζουν ένα δίκτυο.
Τύποι/Παραλλαγές:	Οι υποκλοπείς πακέτων δεδομένων ενός δικτύου υλοποιούνται συνήθως σε λογισμικό, αλλά υπάρχουν και υλοποιήσεις που βασίζονται σε υλικό. Μπορούν να επεκταθούν στην υποκλοπή πακέτων δεδομένων από συγκεκριμένες, τοπικές υπηρεσίες ενός λειτουργικού συστήματος ή μιας συσκευής, έτσι ώστε να επιτρέψουν σε κακόβουλο λογισμικό την πρόσβαση σε τοπικά εργαλεία διαχείρισης του συστήματος. Οι πιο συνηθισμένες μορφές τέτοιων προγραμμάτων κάνουν καταγραφή των απαραίτητων πληροφοριών και τις αποθηκεύουν σε ένα τοπικό σημείο για μεταγενέστερη ανάκτηση, ενώ υπάρχουν και οι άλλες, οι οποίες τις αποστέλλουν απευθείας (μέσω διαδικτύου) στον επιτιθέμενο.
Μέθοδοι μόλυνσης:	Σχεδόν πάντα εγκαθίστανται από έναν απομακρυσμένο επιτιθέμενο, μετά την απόκτηση πρόσβασης του στο σύστημα ή μπορεί και να κατεβούν στους εξυπηρετητές (servers) μέσω παρεμβολής SQL. Πολύ συχνά τα βλέπουμε σε συνδυασμό με άλλα είδη κακόβουλου λογισμικού, όπως τα προγράμματα backdoors ή τα προγράμματα εντολής/ελέγχου, τα οποία οι επιτιθέμενοι χρησιμοποιούν για να ανακτήσουν τα δεδομένα έχουν καταγραφεί. Οι υποκλοπείς πακέτων δεδομένων χρησιμοποιούνται πολύ συχνά και ως εργαλεία αναγνώρισης, για να χαρτογραφήσουν ένα δίκτυο και στη συνέχεια οι επιτιθέμενοι να εντοπίσουν τα επιθυμητά συστήματα που θα επιτεθούν.
Στόχοι της	Σχεδόν πάντα εγκαθίστανται σε εξυπηρετητές. Ο

επίθεσης:	κύριος στόχος τους είναι τα μηχανήματα που επεξεργάζονται, αποθηκεύουν και μεταδίδουν μεγάλες ποσότητες ευαίσθητων δεδομένων ή περιβάλλοντα δικτύου τα οποία μεταφέρουν επίσης μεγάλες ποσότητες ευαίσθητων δεδομένων. Οι υποκλοπείς πακέτων δεδομένων του δικτύου μπορούν να υποκλέψουν οποιοδήποτε είδος δεδομένων, αλλά κυρίως χρησιμοποιούνται για το κομμάτι των δεδομένων που αφορά αριθμούς καρτών πληρωμής.

2.2.9 Μη-Εξουσιοδοτημένη Πρόσβαση σε ένα σύστημα μέσω Κλεμμένων Διαπιστευτηρίων Πρόσβασης

Unauthorized Access via Stolen Credentials	
Περιγραφή:	Αναφέρεται σε περιπτώσεις στις οποίες **ένας επιτιθέμενος αποκτά πρόσβαση** σε ένα προστατευμένο σύστημα ή μία προστατευμένη συσκευή **χρησιμοποιώντας έγκυρα, αλλά κλεμμένα, διαπιστευτήρια.**
Τύποι/Παραλλαγές:	Συνήθως περιλαμβάνονται ονόματα χρήστη και κωδικοί πρόσβασης, αλλά κατά περιπτώσεις μπορεί να συμπεριληφθούν και άλλα είδη προσωπικών δεδομένων πιστοποίησης, όπως «μυστικές ερωτήσεις» και λοιπά αλφαριθμητικά.
Μέθοδοι μόλυνσης:	Τα δεδομένα πιστοποίησης ενός (ή πολλών) χρηστών μπορούν να αποκτηθούν μέσα από ένα πλήθος μεθόδων, όπως με χρήση καταγραφέων πληκτρολογήσεων (keyloggers) και λογισμικών κατασκοπίας (spyware), μέσω της τεχνικής του προσχήματος (pretexting) και μέσω του ηλεκτρονικού ψαρέματος (phishing) κατά την επίσκεψη ενός χρήστη σε μια ψεύτικη ιστοσελίδα. Μπορούν επίσης να κλαπούν και μέσω διαφόρων μορφών παρακολούθησης ή με γνωστοποίηση των διαπιστευτηρίων του θύματος σε τρίτους. Οι επιθέσεις που αφορούν την γνωστοποίηση σε τρίτους πραγματοποιούνται συνήθως μέσω απομακρυσμένων κακόβουλων προγραμμάτων

	διαχείρισης. Η μη εξουσιοδοτημένη πρόσβαση σε ένα σύστημα συνήθως ακολουθείται και από επίθεση ασθενών ή κακο-διαμορφωμένων Λιστών Ελέγχου Πρόσβασης (ACL) έτσι ώστε να αποκαλυφθούν επιπλέον συστήματα. Μετά από μία επιτυχή πρόσβαση στο σύστημα επίσης, ο επιτιθέμενος εισάγει κακόβουλο λογισμικού σε αυτό, όπως υποκλοπείς πακέτων δεδομένων δικτύου (packet sniffers), προγράμματα backdoors ή προγράμματα εντολής/ελέγχου (command/control).
Στόχοι της επίθεσης:	Σχεδόν πάντα στοχεύονται οι εφαρμογές, οι εξυπηρετητές και οι συσκευές δικτύου.

2.2.10 Υποκλοπή δεδομένων μέσω της τεχνικής επινόησης Προσχημάτων

Pretexting	
Περιγραφή:	Η τεχνική του **προσχήματος (pretexting)** αποτελεί τεχνική Κοινωνικής Μηχανικής (Social Engineering) σύμφωνα με την οποία ο επιτιθέμενος εφευρίσκει ένα σενάριο (ένα πρόσχημα) για να πείσει, να διαχειριστεί ή να ξεγελάσει το θύμα έτσι ώστε αυτό να του αποκαλύψει προσωπικές του πληροφορίες ή να εκτελέσει μια ενέργεια για αυτόν. Αυτές οι επιθέσεις βρίσκουν επιτυχία εκμεταλλευόμενες σφάλματα (bugs) του ανθρώπινου μυαλού, κάτι που δυστυχώς, δεν χρίζει επιδιόρθωσης.
Τύποι/Παραλλαγές:	Τα σενάρια που μπορεί να δημιουργήσει ένας άνθρωπος ποικίλουν σε μεγάλο βαθμό και περιορίζονται μόνο από την φαντασία του ατόμου που τα εφεύρει. Τα προσχήματα συχνά χρησιμοποιούνται σε συνδυασμό με την Κοινωνική Μηχανική, αλλά στην πραγματικότητα αποτελούν μια γενικότερη κατηγορία επίθεσης.
Μέθοδοι μόλυνσης:	Σχετίζεται και συχνά συγχέεται με την επίθεση Ηλεκτρονικού Ψαρέματος (phishing), μια άλλη μέθοδο που δανείζεται τα «εργαλεία» της από την

	εργαλειοθήκη της Κοινωνικής Μηχανικής. Συνήθως τα προσχήματα χρησιμοποιούνται για φυσική κλοπή προσωπικών δεδομένων ή για την κλοπή διαπιστευτηρίων πρόσβασης του συστήματος. Οι μέθοδοι επίθεσης μέσω προσχήματος είναι εκατοντάδες και περιλαμβάνουν: email, τηλέφωνο, επικοινωνία πρόσωπο με πρόσωπο, και άλλα μέσα, τα οποία μπορεί να υποστηρίξει το σενάριο που έχει χρησιμοποιηθεί.
Στόχοι της επίθεσης:	Ο στόχος της τεχνικής του προσχήματος είναι οι άνθρωποι. Ο σκοπός της όμως είναι να παραχωρηθεί πρόσβαση στις προσωπικές πληροφορίες του θύματος. Ως εκ τούτου, ο ουσιαστικός στόχος είναι οι εργαζόμενοι με μεγαλύτερα δικαιώματα πρόσβασης ή με μεγαλύτερες ευθύνες στο πλαίσιο ενός οργανισμού (δηλαδή, το ανθρώπινο δυναμικό του οργανισμού, οι διαχειριστές πληροφορικής, κλπ). Η τεχνική του Προσχήματος χρησιμοποιείται για να αποκαλύψει όλες τις μορφές των δεδομένων, αλλά πιο συχνά στοχεύει στην αποκάλυψη IP διευθύνσεων και άλλων εταιρικών πληροφοριών και όχι απλά σε δεδομένα καρτών πληρωμής.

2.2.11 Διείσδυση σε ένα σύστημα μέσω Παράκαμψης του Μηχανισμού Πιστοποίησης Χρηστών

Authentication Bypass	
Περιγραφή:	Τεχνική επίθεσης η οποία χρησιμοποιείται για την παράκαμψη των μηχανισμών ελέγχου ταυτότητας ενός συστήματος και για την απόκτηση μη εξουσιοδοτημένης πρόσβασης σε αυτό.
Τύποι/Παραλλαγές:	Υπάρχουν πολλές παραλλαγές αυτής της επίθεσης οι οποίες είναι και διαφορετικές μεταξύ τους. Έτσι, το επιθυμητό αποτέλεσμα μπορεί να επιτευχθεί με την εκμετάλλευση τρωτών σημείων στον κώδικα, με κακές διαμορφώσεις των μηχανισμών πιστοποίησης ή με την εκμετάλλευση της υπάρχουσας αρχιτεκτονικής για την εξουσιοδότηση πρόσβασης.

Μέθοδοι μόλυνσης:	Μπορεί να αποτελεί μια αυτόνομη μέθοδο επίθεσης αλλά συχνά χρησιμοποιείται σε συνδυασμό και με άλλες τεχνικές για να εισάγει κακόβουλο λογισμικό στο περιβάλλον του θύματος. Συνδυάζεται με υποκλοπείς πακέτων δεδομένων δικτύου (packet sniffers) και με προγράμματα backdoors ή με προγράμματα Εντολής/Ελέγχου (Command/ Control). Η διαδικασία της παράκαμψης συχνά επιτυγχάνεται μέσω επιθέσεων Υπερχείλισης Ενδιάμεσης Μνήμης (Buffer Overflow attacks). Επίσης, η παράκαμψη μπορεί να επιτευχθεί και μέσω προσαρμοσμένων εφαρμογών του επιτιθέμενου ή μέσω προγραμμάτων απομακρυσμένης πρόσβασης.
Στόχοι της επίθεσης:	Σχεδόν πάντα, στόχο αποτελούν οι εφαρμογές, οι εξυπηρετητές (servers) και οι συσκευές δικτύου.

2.2.12 Διείσδυση σε ένα σύστημα λόγω Φυσικής Κλοπής προσωπικών Περιουσιακών Στοιχείων

Physical Theft of Assets	
Περιγραφή:	Είναι η πράξη της φυσικής κλοπής ενός αγαθού από έναν προσωπικό χώρο εργασίας.
Τύποι/Παραλλαγές:	Οι κλέφτες μπορούν να εισβάλλουν στον χώρο εργασίας μιας εταιρίας ή στον χώρο εργασίας ενός σπιτιού και να δραπετεύσουν με διάφορα αντικείμενα ή με προσωπικές πληροφορίες των χρηστών με ποικίλους τρόπους, ενώ μπορούν να το κάνουν διακριτικά, σε κοινή θέα ή με τη χρήση βίας. Σε μία φυσική κλοπή, αναλόγως το που συνέβη, καθορίζονται και τα σχετικά σημεία ελέγχου που παραβιάστηκαν.
Μέθοδοι μόλυνσης:	Συνήθως αποτελεί μια αυτόνομη μέθοδο επίθεσης, αλλά **μπορεί να υπάρξουν και περιπτώσεις κατά τις οποίες μπορεί να κλαπεί ένα αγαθό, να παραποιηθεί, και στη συνέχεια επιστρέψει στο μέρος όπου κλάπηκε,** ως τμήμα ενός μεγαλύτερου σεναρίου επίθεσης. Όπως συμβαίνει και με τα

	κλεμμένα ανταλλακτικά αυτοκινήτων, τα οποία συνήθως διαλύονται και πωλούνται, έτσι και στα προσωπικά δεδομένα, οι εγκληματίες προσπαθούν να τα υποκλέψουν και να τα πουλήσουν σε άλλους. Υπό αυτές τις συνθήκες (συνήθως σε εξαιρετικά στοχευμένες επιθέσεις), οποιοσδήποτε αριθμός από εργαλεία και τεχνικές μπορεί να χρησιμοποιηθεί.
Στόχοι της επίθεσης:	Στόχο αποτελεί οτιδήποτε σχετικό με αγαθά ή δεδομένα. Περισσότερο κοινά είναι τα αγαθά που μπορούν να μετακινηθούν εύκολα, όπως φορητοί υπολογιστές και φορητές συσκευές αποθήκευσης (τα οποία περιέχουν αποθηκευμένα δεδομένα).

2.2.13 Επίθεση Ωμής Βίας

Brute-Force Attack	
Περιγραφή:	Αποτελεί μια αυτοματοποιημένη διαδικασία επαναλαμβανόμενης δοκιμής ονομάτων χρήστη ή κωδικών πρόσβασης ή και των δύο μαζί ταυτοχρόνως, στο σημείο πιστοποίησης χρήστη ενός συστήματος (log-in section) ή μιας εφαρμογής, μέχρι να μαντευθούν τα σωστά και να γίνει επιτυχής εισαγωγή στο σύστημα ή στην εφαρμογή. Οι κωδικοί και τα ονόματα χρήστη που δοκιμάζονται μέσω της ωμής βίας μπορεί να αποτελούν φράσεις που προέρχονται μέσα από μια βάση δεδομένων (επίθεση λεξικού) ή από φράσεις εντελώς τυχαίες, που παράγονται εκείνη τη στιγμή (on-the-fly), σύμφωνα με τα χαρακτηριστικά που έχει δώσει ο επιτιθέμενος (π.χ. φράσεις που θα χρησιμοποιούν μόνο το αγγλικό αλφάβητο, μόνο πεζά γράμματα και θα είναι αναμιγμένες με αριθμούς). Να σημειωθεί ότι **ο χρόνος για την επιτυχή ανακάλυψη των διαπιστευτηρίων πρόσβασης ή άλλων δεδομένων μπορεί να είναι από λίγες ώρες ως μερικά χρόνια**, με την υπολογιστική ισχύ να μην παίζει μεγάλο ρόλο σε αυτό το σημείο.

Τύποι/Παραλλαγές:	Υπάρχουν διάφοροι τύποι επιθέσεων ωμής βίας, όπως οι επιθέσεις κατά των κλειδιών κρυπτογράφησης σε κρυπτογραφικούς αλγορίθμους, αλλά κυριότερες είναι οι επιθέσεις κατά των διαπιστευτηρίων πρόσβασης σε ένα σύστημα ή σε μία ιστοσελίδα. Ως παραλλαγές των επιθέσεων αυτών μπορεί να θεωρηθούν οι τεχνικές δοκιμής κωδικών οι οποίες βασίζονται σε λεξικό (dictionary-based attack) ή σε πιο πολύπλοκους αλγορίθμους.
Μέθοδοι μόλυνσης:	Αποτελεί μια αυτόνομη μέθοδο επίθεσης, αλλά συχνά χρησιμοποιείται σε συνδυασμό με άλλες τεχνικές για την εισαγωγή κακόβουλου λογισμικού στο περιβάλλον του χρήστη. Μετά από μία επιτυχή διείσδυση στο σύστημα ακολουθείται συνήθως και επίθεση αδύναμων ή κακο-διαμορφωμένων Λιστών Ελέγχου Πρόσβασης (ACL), η οποία μπορεί να αποκαλύψει πληροφορίες από επιπλέον συστήματα στο περιβάλλον του χρήστη.
Στόχοι της επίθεσης:	Ο στόχος της επίθεσης είναι οποιοδήποτε σύστημα ή εφαρμογή απαιτεί έλεγχο πρόσβασης με όνομα χρήστη και κωδικό πρόσβασης (log-in). Γενικά στοχεύονται οι εφαρμογές, οι εξυπηρετητές (servers) και οι συσκευές δικτύου.

2.2.14 Υποκλοπή δεδομένων με χρήση Υποκλοπέων Μνήμης RAM

RAM Scrapers	
Περιγραφή:	Οι **υποκλοπείς μνήμης RAM** είναι μια καινούρια μορφή κακόβουλου λογισμικού η οποία έχει σχεδιαστεί για να αναλύει και να υποκλέπτει δεδομένα από την κύρια μνήμη (RAM) ενός συστήματος.
Τύποι/Παραλλαγές:	Μέχρι στιγμής υπάρχει μόνο ένας τύπος υποκλοπέα μνήμης RAM.
Μέθοδοι μόλυνσης:	Τα προγράμματα αυτά εγκαθίστανται σε ένα σύστημα από κάποιον απομακρυσμένο επιτιθέμενο, μετά από επιτυχή απόκτηση

	πρόσβασής του σε αυτό. Οι υποκλοπείς μνήμης RAM εμφανίζονται συχνά σε συνδυασμό με άλλα είδη κακόβουλου λογισμικού, όπως τα προγράμματα backdoors και τα προγράμματα Εντολής/Ελέγχου (Command/Control) τα οποία οι επιτιθέμενοι χρησιμοποιούν για να αποκτήσουν τα δεδομένα που έχουν υποκλαπεί.
Στόχοι της επίθεσης:	Σχεδόν πάντα εγκαθίστανται σε εξυπηρετητές (servers), κυρίως όμως σε μηχανήματα εξυπηρέτησης πωλήσεων (POS - Point of Sale), τα οποία ασχολούνται με την επεξεργασία, την αποθήκευση και την μετάδοση δεδομένων καρτών πληρωμής.

2.2.15 Υποκλοπή δεδομένων μέσω της τεχνικής Ηλεκτρονικού Ψαρέματος

Phishing	
Περιγραφή:	**Η τεχνική του ηλεκτρονικού ψαρέματος, ή αλλιώς, της «απάτης» ηλεκτρονικού ψαρέματος (phishing scam)**, αποτελεί τεχνική Κοινωνικής Μηχανικής σύμφωνα με την οποία ο επιτιθέμενος χρησιμοποιεί έναν τρόπο ηλεκτρονικής επικοινωνίας για να δελεάσει τον παραλήπτη και να τον εξαπατήσει, οδηγώντας τον σε αποκάλυψη προσωπικών του πληροφοριών. Τις περισσότερες φορές η απάτη συμβαίνει μέσω email. Το email που καταφθάνει στον λογαριασμό του χρήστη έχει Διευθύνσεις Αποστολέα ίδιες με αυτές της τράπεζας στην οποία είναι πελάτης ή ίδιες με κάποιου ηλεκτρονικού καταστήματος (e-shop) μέσα από το οποίο αγοράζει προϊόντα, **κάνοντας το έτσι να φαντάζει γνήσιο.** Στο κυρίως κείμενο του email μπορεί να αναφέρεται ότι πρέπει να αλλάξει τους κωδικούς πρόσβασής του για λόγους ασφαλείας πατώντας ένα επισυναπτόμενο link. Το link αυτό όμως δεν οδηγεί στην ιστοσελίδα της τράπεζας ή του καταστήματος αλλά σε μια άλλη, πλαστή ιστοσελίδα η οποία συνήθως έχει τα ίδια χρώματα, εικόνες και κείμενα

	με την γνήσια. Εκεί ο χρήστης καταχωρεί τα διαπιστευτήριά του και έτσι ο επιτιθέμενος τα υποκλέπτει, χωρίς ο χρήστης να αντιληφθεί τι συνέβη.
Τύποι/Παραλλαγές:	Οι επιθέσεις ηλεκτρονικού ψαρέματος διαφοροποιούνται σε μεγάλο βαθμό ανάλογα με τον τύπο, την τακτική που ακολουθείται και τους στόχους. Όπως αναφέρεται και στην περιγραφή, το email είναι το κύριο μέσο για την τέλεση της απάτης αλλά όχι το μοναδικό. Μπορεί να υπάρξουν επιθέσεις μέσω pop-up παραθύρων, μέσω παραθύρων δηλαδή των περιηγητών τα οποία ανοίγουν αυτόματα κατά την πλοήγηση του χρήστη σε μία ιστοσελίδα, και μπορεί να περιέχουν πλαστές πληροφορίες με σκοπό να εξαπατήσουν τον χρήστη. Κάποιες άλλες επιθέσεις, ζητούν απευθείας από τους χρήστες να στείλουν τις προσωπικές τους πληροφορίες στον επιτιθέμενο, χωρίς να γνωρίζουν ότι ο αποστολέας είναι ο επιτιθέμενος. Τα emails ηλεκτρονικού ψαρέματος διαδίδονται συνήθως σε τυχαίους παραλήπτες όπως τα ανεπιθύμητα (spam) emails, αλλά υπάρχουν και στοχευμένες παραλλαγές, όπως η απάτη Spear Phishing (η οποία απευθύνεται σε ένα συγκεκριμένο οργανισμό) ή η «φαλαινοθηρία» (Whaling) (με στόχο VIPs ή στελέχη εταιριών). Αν και το ηλεκτρονικό ψάρεμα στοχεύει συνήθως καταναλωτές, μπορεί να υπάρξουν και σενάρια όπου στόχος είναι η παραβίαση εταιρικών δεδομένων.
Μέθοδοι μόλυνσης:	Οι επιτυχείς επιθέσεις ηλεκτρονικού ψαρέματος συχνά ακολουθούνται από μη εξουσιοδοτημένη πρόσβαση μέσω κλεμμένων διαπιστευτηρίων. Το email είναι ο κύριος φορέας.
Στόχοι της επίθεσης:	Ο στόχος του ηλεκτρονικού ψαρέματος είναι οι άνθρωποι, αλλά ο απώτερος σκοπός είναι να αποκτηθεί πρόσβαση σε προσωπικές πληροφορίες των χρηστών ή σε συστήματα ηλεκτρονικής τραπεζικής. Συνεπώς, οι επιθέσεις αυτές στοχεύουν κυρίως τα συστήματα τελικών χρηστών, με τους πιο παραβιάσιμους τύπους δεδομένων να αποτελούν τα διαπιστευτήρια πρόσβασης και οι προσωπικές πληροφορίες των χρηστών.

2.2.16 Σχολιασμός επιθέσεων

Οι μορφές επιθέσεων που παρουσιάστηκαν παραπάνω ίσως αφήνουν την υπόνοια ότι είναι θεωρητικές ή ότι μπορεί να συμβούν μόνο σε εταιρίες μεγάλου βεληνεκούς και όχι σε απλούς χρήστες. Αυτό είναι ένα λάθος συμπέρασμα, αν έχει βγει από κάποιον. Σίγουρα οι μεγάλες εταιρίες αποτελούν τον κυριότερο στόχο για τους επιτιθέμενους, αφού αυτές διαθέτουν πολύ περισσότερα και ίσως, πολυτιμότερα δεδομένα από αυτά που διαθέτει ένας χρήστης. Οι απλοί χρήστες όμως, από την άλλη, είναι πολύ πιθανό να μην παίρνουν τα μεγάλα μέτρα ασφαλείας που παίρνουν οι εταιρίες (ίσως δεν παίρνουν και καθόλου, ανά περιπτώσεις) και με δεδομένη την μέχρι ενός σημείου άγνοιά τους σε τεχνολογικά θέματα, συνεπάγεται η ευκολότερη διείσδυση στον υπολογιστή τους για υποκλοπή.

Μια ακόμη διαφορά μεταξύ των εταιριών και των απλών χρηστών είναι το πλήθος τους. Οι εταιρίες είναι τάξεις μεγέθους λιγότερες από ότι οι μεμονωμένοι απλοί χρήστες. Αυτό συνεπάγεται ότι αν και στις εταιρίες θα μπορούσε να υπάρχει ένα φυσικό πρόσωπο ή μια ομάδα φυσικών προσώπων που θα διενεργούσαν την επίθεση (hackers) στα συστήματά τους, στους μεμονωμένους χρήστες αυτό δε μπορεί να συμβεί για ευνόητους λόγους. Δε συμφέρει στους επιτιθέμενους να χάσουν τον χρόνο τους για τη διείσδυση στο μηχάνημα του κάθε μεμονωμένου χρήστη ξεχωριστά με πιθανότητα να μην εντοπίσουν κάτι ωφέλιμο, αλλά ακόμα πιο σημαντικό είναι το γεγονός ότι δεν επαρκεί και το πλήθος των επιτιθέμενων για να καλύψει τα τόσα εκατομμύρια μεμονωμένων χρηστών.

Έτσι, οι επιθέσεις στους μεμονωμένους χρήστες διαφοροποιούνται και διενεργούνται μέσα από ένα μεγάλο πλήθος αυτοματοποιημένων, κακόβουλων προγραμμάτων (μερικά από αυτά αναλύθηκαν και στα παραπάνω κεφάλαια) τα οποία αναλαμβάνουν τα ίδια από μόνα τους την παρακολούθηση των χρηστών, την υποκλοπή των προσωπικών τους πληροφοριών και την αποστολή τους πίσω, σε αυτούς.

Συνεπώς, όλα τα είδη επιθέσεων που παρουσιάστηκαν στα παραπάνω κεφάλαια μπορούν να συμβούν και στον απλό χρήστη, σε μικρότερο βαθμό ίσως και κυρίως μέσω κάποιου κακόβουλου προγράμματος.

2.3 Τρόποι και μέσα προστασίας από επιθέσεις

Οι τρόποι για να προστατευθούν οι μεμονωμένοι χρήστες από τυχών επιθέσεις κακόβουλων χρηστών ή κακόβουλων προγραμμάτων και ιών είναι αρκετοί. Η ασφάλεια, εξαρτάται κυρίως από το μηχάνημα που χρησιμοποιεί κάποιος και από το πόσο αξιόπιστο αυτό θεωρείται. Λέγοντας αξιόπιστο εννοούμε το κατά πόσο αυτό απέχει από εγκατεστημένο κακόβουλο λογισμικό και ιούς και το κατά πόσο αυτό είναι επιρρεπής στις μολύνσεις ή στις επιθέσεις.

Παρακάτω ακολουθεί μία λίστα η οποία περιγράφει μεθόδους για την προστασία των χρηστών από υποκλοπή προσωπικών δεδομένων ή/και διαπιστευτηρίων πρόσβασης από έναν υπολογιστή, ένα έξυπνο κινητό τηλέφωνο και γενικά από οποιαδήποτε συσκευή έχει εγκατεστημένο κάποιο διαδεδομένο Λειτουργικό Σύστημα.

- **Επιλογή ισχυρών κωδικών πρόσβασης:** ένας γενικός κανόνας που πρέπει να εφαρμόζεται σε κάθε συσκευή που απαιτεί χρήση κωδικού πρόσβασης έτσι ώστε να μεγιστοποιείται η ασφάλεια. Ισχυρός κωδικός πρόσβασης θεωρείται ο κωδικός που: α) διαθέτει τυχαίους χαρακτήρες στην δομή του και όχι λέξεις οι οποίες μπορούν να βρεθούν σε κάποιο λεξικό, ούτε γνωστές λέξεις (μάρκες προϊόντων, επιγραφές από πινακίδες) ή γνωστές ημερομηνίες (γενεθλίων, επετείων, κ.τ.λ.) που χρησιμοποιούμε στην καθημερινότητά μας, β) που εκτός από μικρά και κεφαλαία γράμματα είναι αναμειγμένος και με αριθμούς και σύμβολα (ειδικούς χαρακτήρες όπως: *, @, {, “, κ.τ.λ.) και γ) που έχει μέγεθος από 10 χαρακτήρες και πάνω. Για τη δημιουργία τέτοιων κωδικών πρόσβασης συνήθως χρησιμοποιούνται προγράμματα-γεννήτριες κωδικών οι οποίες με τη σειρά τους χρησιμοποιούν κρυπτογραφικά ασφαλείς μεθόδους παραγωγής και μπορούν να προσαρμοστούν στις απαιτήσεις του χρήστη. Ο χρήστης πρέπει να επιλέγει πάντα ισχυρούς κωδικούς πρόσβασης για οποιοδήποτε πρόγραμμα, ιστοσελίδα, υπηρεσία του λειτουργικού συστήματος ή συσκευή που απαιτεί πιστοποίηση και πρέπει να τους αλλάζει τουλάχιστον μια φορά το μήνα. Η αλλαγή αυτή πρέπει να συμβαίνει τόσο συχνά, έτσι ώστε να ελαχιστοποιείται το σενάριο ανακάλυψης των χρησιμοποιούμενων κωδικών λόγω παρακολούθησης (φυσικής παρακολούθησης ή μέσω προγράμματος) και να εκμηδενίζεται κάθε φορά η όποια προσπάθεια μπορεί να υπάρχει σχετικά με την εξεύρεσή τους (π.χ. να καταστρέφεται κάποια επίθεση ωμής βίας που βρίσκεται εν ενεργεία).

- **Ελαχιστοποίηση κοινοποιούμενων πληροφοριών.** Οι χρήστες, άθελά τους και χωρίς να γνωρίζουν την επικινδυνότητα, έχουν την τάση να κοινοποιούν προσωπικές πληροφορίες στο Διαδίκτυο. Δεν είναι λίγοι αυτοί που έχουν λογαριασμούς σε διάφορα κοινωνικά δίκτυα και που καταχωρούν και κοινοποιούν δημόσια όλα τα προσωπικά τους στοιχεία στα προφίλ τους, με κάθε λεπτομέρεια, από τα ονόματα και τα επίθετά

τους ως και τις διευθύνσεις της κατοικίας τους. Κάποιοι επίσης, όπως περιγράφηκε και στο κεφάλαιο 1.2, μπορεί να ανακοινώσουν απευθείας τους προσωπικούς κωδικούς πρόσβασής τους σε φίλους, χωρίς να έχουν αίσθηση του κινδύνου που αυτό μπορεί να επιφέρει, ούτως ώστε αυτοί να τους κάνουν κάποια εξυπηρέτηση ή για άλλους λόγους. Τα παραπάνω θέματα, αν και για κάποιους μπορεί να μοιάζουν ακίνδυνα, αποτελούν την No. 1 αιτία υποκλοπής προσωπικών πληροφοριών από κακόβουλους χρήστες και την ανακάλυψη κωδικών πρόσβασης και ονομάτων χρήστη. Γενικά δε πρέπει να ανακοινώνει κάποιος ποτέ, σε κανέναν και για κανένα λόγο τους κωδικούς πρόσβασής του και πρέπει να περιορίσει, ή ακόμα καλύτερα να διαγράψει τα προσωπικά δεδομένα που μοιράζεται στο Διαδίκτυο ή σε άλλα εμφανή σε τρίτους σημεία, όπως: ημερομηνίες γέννησης, πραγματικό όνομα και επίθετο, τηλέφωνο, φορολογικά στοιχεία, εργοδότες, διευθύνσεις κατοικιών και εργασίας, διευθύνσεις ηλεκτρονικού ταχυδρομείου, ονόματα χρήστη υπηρεσιών chat, κ.α..

- **Γνήσιο Λειτουργικό Σύστημα:** Το Λειτουργικό Σύστημα είναι το βασικότερο κομμάτι που χρησιμοποιεί η συσκευή μας (ο υπολογιστής μας ή το έξυπνο κινητό τηλέφωνό μας). Με βάση αυτό σχεδιάζονται όλα τα προγράμματα που πρόκειται να τρέξουν (συμπεριλαμβανομένων και των κακόβουλων) και με βάση αυτό δημιουργούνται κάποιες από τις επιθέσεις. Ο χρήστης πρέπει να έχει πάντα εγκατεστημένο ένα γνήσιο λειτουργικό σύστημα στον υπολογιστή του (όχι ένα πειρατικό ή κάποια ειδική προσαρμοσμένη έκδοση - πειρατική ή μη - τροποποιημένη από κάποιον άλλον χρήστη), το οποίο θα έχει αγοράσει (εκτός κι αν διανέμεται δωρεάν) και θα έχει κατεβάσει από την επίσημη ιστοσελίδα μόνο (ή από επίσημους mirror εξυπηρετητές) και όχι από τρίτους. Ο λόγος που πρέπει να συμβαίνει αυτό είναι ότι μόνο με αυτόν τον τρόπο οι χρήστες μπορούν να είναι 100% σίγουροι ότι το λειτουργικό τους σύστημα είναι καθαρό από κακόβουλα προγράμματα, από ιούς (φανερούς ή μη, εντοπίσημους ή όχι) και ότι δεν θα έχει κακόβουλες προεπιλεγμένες ρυθμίσεις για πρόσβαση εισβολέων εξ αποστάσεως (ανοικτές πύλες, πειραγμένα host files, εξαιρέσεις προγραμμάτων από ρυθμίσεις ασφαλείας, κ.α.) όπως μπορεί να συμβεί σε ένα πειρατικό ή προσαρμοσμένο λειτουργικό σύστημα.

- **Ενημερωμένο Λειτουργικό Σύστημα:** Εκτός από το να είναι γνήσιο ένα λειτουργικό σύστημα πρέπει να είναι και διαρκώς ενημερωμένο με τις τελευταίες ενημερώσεις ασφαλείας και τις νέες εκδόσεις των υποσυστημάτων του και των εφαρμογών του. Ο λόγος είναι απλός. Σχεδόν κάθε μέρα ανακαλύπτονται διάφορες ευπάθειες του λειτουργικού συστήματος και των υποσυστημάτων του και εκμεταλλεύονται από κακόβουλους χρήστες. Με την πάροδο του χρόνου οι ευπάθειες αυτές γίνονται γνωστές και στους κατασκευαστές του λειτουργικού συστήματος, παράγοντας και αυτοί με τη σειρά τους νέες, ασφαλείς εκδόσεις των υποσυστημάτων αυτών, ούτως ώστε αυτό να μπορεί να διατηρείται προστατευμένο. Αν ο χρήστης όμως δεν το διατηρεί

ενημερωμένο, τότε οι νέες εκδόσεις των υποσυστημάτων του δεν θα εγκατασταθούν ποτέ σε αυτό και αν κάποια στιγμή κατεβεί και τρέξει κάποιο κακόβουλο πρόγραμμα που βασίζεται σε αυτές τις ευπάθειες, τότε αυτό θα τις εκμεταλλευτεί και μπορεί να υποκλέψει προσωπικά δεδομένα λόγω αυτών ή να επιτρέψει την πρόσβαση στο μηχάνημα από τρίτους χωρίς ο χρήστης να αντιληφθεί το παραμικρό. Συνεπώς, έχοντας οι χρήστες πάντα ενημερωμένο το λειτουργικό τους σύστημα αλλά και τις επιπλέον εφαρμογές που έχουν εγκαταστήσει σε αυτό, αυξάνουν το επίπεδο της ασφάλειας προλαμβάνοντας όλο και περισσότερα νέα ήδη επιθέσεων και κακόβουλων προγραμμάτων.

- **Εγκατάσταση ενός γνήσιου αντιϊκού προγράμματος:** Η εγκατάσταση ενός αντιϊκού προγράμματος (antivirus) προλαμβάνει την μόλυνση του λειτουργικού συστήματος από εκατομμύρια κακόβουλα προγράμματα και ιούς που κυκλοφορούν στο Διαδίκτυο αυτή τη στιγμή. Οι κατασκευαστές των αντιϊκών προγραμμάτων ανακαλύπτουν καθημερινά νέα είδη απειλών και ενημερώνουν την βάση δεδομένων των προγραμμάτων τους πολλές φορές μέσα στην ημέρα. Έτσι ο χρήστης, με το να έχει εγκατεστημένο στο μηχάνημά του ένα τέτοιο πρόγραμμα, προστατεύεται αυτόματα από αυτά τα είδη των απειλών. Αυτό σημαίνει ότι αν αυτός επισκεφθεί μία ιστοσελίδα του παγκόσμιου ιστού η οποία είναι μολυσμένη ή είναι γνωστή για την προσπάθεια εισαγωγής κακόβουλου λογισμικού στο μηχάνημα του χρήστη, τότε το αντιϊκό πρόγραμμα θα διακόψει αυτομάτως την επικοινωνία αυτή και θα ενημερώσει τον χρήστη για αυτήν την επικινδυνότητα. Παρομοίως, αν ο χρήστης πάει να κατεβάσει από το Διαδίκτυο ή από το e-mail του κάποιο μολυσμένο αρχείο, το αντιϊκό πρόγραμμα θα τον προστατεύσει διακόπτοντας την συγκεκριμένη σύνδεση και διαγράφοντας το μολυσμένο αρχείο. Φυσικά, όλα αυτά θα συμβούν μόνο αν η βάση δεδομένων του προγράμματος είναι ήδη ενήμερη για την κυκλοφορία αυτών των απειλών. Τα αντιϊκά προγράμματα συχνά συνδυάζονται και με προγράμματα που λειτουργούν ως **Τείχη Προστασίας (firewalls)** μεταξύ του λειτουργικού συστήματος και των συνδέσεων που δημιουργούνται με διάφορους κόμβους του δικτύου (όπως της σύνδεσης με κάποιον άλλον υπολογιστή στο τοπικό δίκτυο ή με κάποιον απομακρυσμένο εξυπηρετητή του Διαδικτύου). Τα τείχη προστασίας φιλτράρουν τα δεδομένα που περνούν μέσα από μία σύνδεση δικτύου και αν εντοπίσουν κομμάτια κακόβουλου λογισμικού ή προσπάθειες επίθεσης, κόβουν την επικοινωνία αυτή, ούτως ώστε το λειτουργικό σύστημα να παραμείνει ασφαλισμένο (και αδιάρρηκτο). Τέλος, η γνησιότητα ενός τέτοιου προγράμματος αντί ενός πειρατικού κατέχει πολύ σημαντικό ρόλο στην ασφάλεια του χρήστη για τους ίδιους λόγους όπως και στην επιλογή ενός γνήσιου λειτουργικού συστήματος.

- **<u>Εγκατάσταση ενός συστήματος ανίχνευσης εισβολής:</u>** Ένα σύστημα ανίχνευσης εισβολής (Intrusion Detection System - IDS) είναι μία συσκευή ή μία εφαρμογή εγκατεστημένη σε ένα λειτουργικό σύστημα η οποία παρακολουθεί και αναλύει τις δραστηριότητες του δικτύου ή του λειτουργικού συστήματος για τυχών κακόβουλες ενέργειες ή παραβιάσεις της πολιτικής ασφαλείας και παράγει κάποιες αναφορές κατάστασης. Υπάρχουν δύο κατηγορίες συστημάτων ανίχνευσης εισβολής, αυτά που βασίζονται στο δίκτυο (Network based IDS - NIDS) και αυτά που βασίζονται στους υπολογιστές (Host based IDS - HIDS). Οι λόγοι εγκατάστασης ενός συστήματος ανίχνευσης εισβολής σε μία συσκευή είναι η πρόληψη προβλημάτων, η ανίχνευση παραβιάσεων, η τεκμηρίωση υπαρκτών απειλών, ο έλεγχος ποιότητας για το σχεδιασμό ασφαλείας, καθώς και η θωράκιση παλαιών συστημάτων σε περίπτωση που κρίνεται αναγκαία η διατήρησή τους. Έτσι, εάν σε ένα σύστημα βρίσκονται ή διακινούνται πολύ σημαντικά δεδομένα, η εγκατάσταση ενός τέτοιου συστήματος κρίνεται αναγκαία για την ασφάλειά τους.

- **<u>Προσοχή στα προγράμματα που πρόκειται να εγκατασταθούν:</u>** Ο χρήστης πρέπει να προσέχει πάντα όλα τα είδη των προγραμμάτων που πρόκειται να εγκαταστήσει στον υπολογιστή ή στο έξυπνο κινητό του τηλέφωνο ούτως ώστε αυτά να μην είναι κακόβουλα, να μην είναι «πειραγμένα» από κακόβουλους χρήστες ή να μην είναι μολυσμένα με ιούς. Κάθε φορά που ο χρήστης βλέπει μια εφαρμογή και σκέφτεται να την εγκαταστήσει ή απλά σκέφτεται να κάνει διπλό κλικ πάνω της και να την τρέξει, πρέπει να το σκεφτεί σοβαρά και να αναρωτηθεί τα εξής: α) ποιος την έχει κατασκευάσει, κάποια αξιόπιστη εταιρία ή ένας άγνωστος χρήστης; β) την προτείνει κάποιος για εγκατάσταση ή είναι παντελώς άγνωστη; Αν ναι, ποιος είναι αυτός, είναι έμπιστος; γ) είναι αξιόπιστη η πηγή προέλευσης από όπου θα κατεβεί η εφαρμογή ή είναι ύποπτη / επικίνδυνη για δολιοφθορές; Μήπως υπάρχει και επίσημο σημείο διανομής για να κατέβει από εκεί; δ) έχουν χρησιμοποιήσει άλλοι χρήστες του Διαδικτύου ή φίλοι την συγκεκριμένη εφαρμογή; Αν ναι, ποια ήταν η εμπειρία τους από τη χρήση της; και τέλος ε) η εφαρμογή είναι ψηφιακά υπογεγραμμένη; Μπορεί να επαληθευτεί η ακεραιότητά της μέσω σύνοψης;. Ακολουθώντας κάποιος τα παραπάνω βήματα, ο χρήστης μπορεί να είναι αρκετά σίγουρος για αυτό που πρόκειται να εγκαταστήσει και να διατηρήσει τον υπολογιστή του ασφαλή. Τέλος, εννοείται πως η εγκατάσταση παράνομων πειρατικών προγραμμάτων πρέπει να αποφεύγεται δια παντός, αφού είναι σχεδόν σίγουρο ότι περιλαμβάνουν κάποιου είδους κακόβουλο λογισμικό το οποίο εγκαθίσταται παράλληλα με αυτά και λειτουργεί στο σύστημα κρυφά ή φανερά.

- **<u>Αποφυγή εγκατάστασης ή εκτέλεσης προγραμμάτων cracking / keygens:</u>** Τα προγράμματα αυτού του είδους φτιάχνονται από εγκληματίες του κυβερνοχώρου και λειτουργούν ως μέθοδοι παράνομου «ξεκλειδώματος» των γνήσιων προγραμμάτων. Συνήθως αποτελούν

μέρος των πειρατικών προγραμμάτων που κυκλοφορούν στο Διαδίκτυο. Ένας παράνομος χρήστης που θέλει να εγκαταστήσει πειρατικό λογισμικό στον υπολογιστή του, μετά την απόκτηση και την εγκατάστασή του, συνήθως του επιβάλλεται να εγκαταστήσει ή να τρέξει ένα πρόγραμμα crack (πρόγραμμα διάσπασης/διάρρηξης) ή ένα keygen (πρόγραμμα παράνομης παραγωγής κωδικών άδειας χρήσης), έτσι ώστε να «ξεκλειδώσει» το λογισμικό του και να το κάνει φαινομενικά γνήσιο, με παράνομη άδεια χρήσης, αορίστου χρόνου. Η μέθοδος αυτή, καθώς και η υποχρέωση παραχώρησης δικαιωμάτων διαχειριστή στα προγράμματα αυτά (ούτως ώστε να μπορέσουν να τρέξουν), τους δίνουν δυνατότητες παραμετροποίησης του λειτουργικού συστήματος χωρίς περιορισμούς αλλά και αλλαγής των ρυθμίσεων ασφαλείας του. Συνεπώς, ενώ από τη μία (ίσως) κάνουν τη δουλειά για την οποία έχουν κατασκευασθεί (την διάρρηξη του προγράμματος), από την άλλη μπορεί να μολύνουν το σύστημα με κακόβουλες ρυθμίσεις και κακόβουλο λογισμικό, ενώ ταυτοχρόνως μπορεί να προσδώσουν και πρόσβαση σε κάποιον εισβολέα εξ αποστάσεως, χωρίς τα παραπάνω να γίνουν αντιληπτά από τον χρήστη.

Αν ένας χρήστης τηρεί τους παραπάνω απλούς κανόνες τότε το μηχάνημά του θα θεωρείται αρκετά ασφαλής από οποιαδήποτε είδη κακόβουλων προγραμμάτων, παρακολουθήσεων και επιθέσεων και μπορεί να το χρησιμοποιεί για την αποθήκευση και την χρήση των ευαίσθητων προσωπικών δεδομένων του και διαπιστευτηρίων με ασφάλεια και με το ελάχιστο δυνατό ρίσκο υποκλοπής.

ΚΕΦΑΛΑΙΟ 3

Κρυπτογραφία, Αλγόριθμοι Κρυπτογράφησης και Διασφάλιση Προσωπικών Δεδομένων

3.1 Κρυπτολογία, Κρυπτογραφία, Κρυπτανάλυση

Κρυπτολογία (Cryptology) είναι ο κλάδος της επιστήμης ο οποίος ασχολείται με τη μελέτη και τη σχεδίαση κρυπτογραφικών τεχνικών, συστημάτων και πρωτοκόλλων (κρυπτογραφία), καθώς και με τη μελέτη διαδικασιών για την παραβίαση αυτών (κρυπτανάλυση). Συνεπώς, η κρυπτολογία από τη μία πλευρά ασχολείται με την απόκρυψη και την κωδικοποίηση ενός μηνύματος και από την άλλη με διαδικασίες αποκάλυψης του περιεχομένου του κωδικοποιημένου μηνύματος. Σήμερα η κρυπτολογία θεωρείται ένα διεπιστημονικό γνωστικό πεδίο των εφαρμοσμένων μαθηματικών, της θεωρητικής πληροφορικής και της επιστήμης του ηλεκτρονικού μηχανικού. Η σημασία της κρυπτολογίας είναι τεράστια στους τομείς της ασφάλειας υπολογιστικών συστημάτων και των τηλεπικοινωνιών.

Κρυπτογραφία (Cryptography) είναι η μελέτη και η υλοποίηση τεχνικών για την ασφαλή επικοινωνία δύο μερών, παράλληλα με την παρεμβολή τρίτων προσώπων (αντιπάλων). Γενικότερα, πρόκειται για την ανάλυση και την κατασκευή πρωτοκόλλων τα οποία αντιστέκονται στις παρεμβολές των αντιπάλων κατά τη διάρκεια μιας επικοινωνίας. Τα πρωτόκολλα αυτά έχουν ως σκοπό να παρέχουν καθ όλη την διάρκεια μιας επικοινωνίας: ιδιωτικότητα δεδομένων, ακεραιότητα δεδομένων, έλεγχο γνησιότητας και μη-δυνατότητα άρνησης της υπογραφής των συμμετεχόντων μερών. Υλοποιήσεις κρυπτογραφίας υπάρχουν αυτή τη στιγμή στις κάρτες για ATMs τραπεζών, στην αποστολή κωδικών πρόσβασης μέσω διαδικτύου (ή στην αποθήκευσή τους σε μια συσκευή), στο ηλεκτρονικό εμπόριο κατά την φάση της πληρωμής με μια πιστωτική κάρτα κ.α..

Η κρυπτογραφία στην αρχαία εποχή ήταν συνώνυμη με την κρυπτογράφηση, την μετατροπή της πληροφορίας δηλαδή από μια ευανάγνωστη κατάσταση σε μία κατάσταση χωρίς νόημα. Ο δημιουργός ενός κρυπτογραφημένου μηνύματος μοιραζόταν την τεχνική αποκρυπτογράφησης που απαιτούνταν για την ανάκτηση της αρχικής πληροφορίας μόνο με συγκεκριμένους παραλήπτες, αποκλείοντας έτσι από τα ανεπιθύμητα πρόσωπα τη δυνατότητα να διαβάσουν το αρχικό κείμενο. Από τον Πρώτο Παγκόσμιο Πόλεμο και έπειτα, κατά την έλευση του ηλεκτρονικού υπολογιστή, η χρήση της κρυπτογραφίας γίνεται ολοένα και πιο επιτακτική με συνέπεια οι μέθοδοι που χρησιμοποιούνται να γίνονται ολοένα και πιο πολύπλοκες, περνώντας έτσι από την κλασσική κρυπτογραφία στην σύγχρονη.

Η Σύγχρονη Κρυπτογραφία (Modern Cryptography) βασίζεται σε μεγάλο βαθμό στην μαθηματική θεωρία σε συνδυασμό με την επιστήμη των υπολογιστών. Οι αλγόριθμοι κρυπτογράφησης σχεδιάζονται πλέον με βάση την αντοχή τους στην επεξεργαστική ισχύ των υπολογιστών, δημιουργώντας έτσι συστήματα τα οποία είναι δύσκολο να σπάσουν, ανεξαρτήτως του αντιπάλου. Τέτοιου είδους συστήματα ονομάζονται **Υπολογιστικά Ασφαλή συστήματα (Computationally Secure)**. Αν και στη θεωρία τέτοια συστήματα είναι δυνατό να σπάσουν, στην πράξη είναι πάρα πολύ δύσκολο, έως ακατόρθωτο. Πρόοδοι της τεχνολογίας βέβαια, όπως για παράδειγμα βελτιώσεις σε αλγορίθμους

παραγοντοποίησης ακεραίων ή αυξήσεις ταχυτήτων υπολογισμού, αναγκάζουν τα συστήματα αυτά να προσαρμόζονται συνεχώς στους νέους ρυθμούς της εποχής.

Επιπλέον, προχωρώντας στην ανώτατη βαθμίδα της ασφάλειας της πληροφορίας, υπάρχουν και τα **Θεωρητικά Ασφαλή συστήματα Πληροφορίας (Information-Theoretically Security systems)**, των οποίων η σχεδίαση και η αρχιτεκτονική ασφαλείας προκύπτει αμιγώς από τη Θεωρία Πληροφορίας (Information Theory). Αυτό σημαίνει ότι τα συστήματα αυτά είναι ασφαλή ακόμα και αν ο αντίπαλος διαθέτει απεριόριστη υπολογιστική ισχύ. Συνεπώς ούτε μία Επίθεση Ωμής Βίας (Brute Force Attack) αλλά ούτε και οποιαδήποτε άλλη επίθεση μπορεί να διασπάσει την ασφάλεια τέτοιων συστημάτων. Μόνο η κλοπή του κλειδιού μπορεί να συντελέσει παράγοντα αποκρυπτογράφησης. Ένα παράδειγμα τέτοιου συστήματος αποτελεί το κρυπτοσύστημα One-Time Pad. Το μόνο πρόβλημα που συναντάμε σε αυτά τα συστήματα και δεν τα χρησιμοποιούμε, βρίσκεται στην πρακτική υλοποίηση. Γι αυτό, προτιμούμε τα υπολογιστικά ασφαλή συστήματα έναντι αυτών, τα οποία διατηρούνε μία ισορροπία ανάμεσα στη ασφάλεια και κόστος υλοποίησης.

Κρυπτανάλυση (Cryptanalysis) είναι ο τομέας της κρυπτολογίας που ασχολείται με τη μαθηματική ανάλυση και τη μελέτη κρυπτοσυστημάτων με σκοπό την ανακάλυψη τρωτών σημείων. Η κρυπτανάλυση χρησιμοποιείται για την παραβίαση κρυπτοσυστημάτων ασφαλείας και την απόκτηση πρόσβασης στα περιεχόμενα των κρυπτογραφημένων μηνυμάτων, ακόμα και αν το κλειδί κρυπτογράφησης είναι άγνωστο.

Επιπρόσθετα από την μαθηματική ανάλυση των κρυπτοσυστημάτων, η κρυπτανάλυση περιλαμβάνει και την μελέτη των επιθέσεων «Πλάγιου Καναλιού» (Side Channel Attacks) οι οποίες δεν στοχεύουν σε αδυναμίες υπολογιστικής αντοχής των κρυπτοσυστημάτων για το σπάσιμό τους, αλλά αντιθέτως, εκμεταλλεύονται τις αδυναμίες που προκύπτουν από την φυσική υλοποίηση αυτών σε κάποιο μηχάνημα - πληροφορίες χρονισμού, καταναλώσεις ενέργειας, ηλεκτρομαγνητικές διαρροές ή ακόμη και ήχοι, μπορούν να θεωρηθούν ως μια επιπλέον πηγή πληροφοριών οι οποίες μπορούν να αξιοποιηθούν για το σπάσιμο του συστήματος.

Συνεπώς, βασικός στόχος της κρυπτανάλυσης είναι (ανάλογα με τις απαιτήσεις του κρυπταναλυτή) η διάσπαση του κρυπτοσυστήματος. Δηλαδή να βρεθεί το κλειδί, το μήνυμα ή ένας ισοδύναμος αλγόριθμος που θα βοηθήσει τον κρυπταναλυτή να αναγνώσει το κρυφό μήνυμα. **Ένα κρυπτοσύστημα λέγεται ότι έχει «σπάσει»**, αν βρεθεί μια μέθοδος (πιθανοτική ή ντετερμινιστική) η οποία μπορεί να ανακαλύψει το μήνυμα ή το κλειδί με υπολογιστική πολυπλοκότητα μικρότερη από αυτήν της επίθεσης ωμής βίας.

3.2 Βασικές έννοιες της Κρυπτογραφίας

Ένα **κρυπτοσύστημα (Cipher, Cryptosystem),** ή αλλιώς ένα κρυπτογραφικός αλγόριθμος, είναι ένα αλγόριθμος (ένα σύνολο προκαθορισμένων βημάτων) ο οποίος έχει ως σκοπό την κρυπτογράφηση και την αποκρυπτογράφηση πληροφοριών ούτως ώστε αυτές να μπορούν να μεταδοθούν με ασφάλεια μέσω κάποιου καναλιού επικοινωνίας (π.χ. Internet, Wi-Fi) ή να αποθηκευτούν με ασφάλεια σε κάποιον υπολογιστή ή άλλη συσκευή.

Κρυπτογράφηση (Encryption) είναι η διαδικασία της μετατροπής της (προσωπικής) πληροφορίας ενός χρήστη σε μια διαφορετική μορφή, η οποία δεν θα παρέχει κανένα νόημα σε οποιονδήποτε προσπαθήσει να την διαβάσει. Η αρχική, προσωπική πληροφορία του χρήστη ονομάζεται **Αρχικό Μήνυμα ή Απλό Κείμενο (Plaintext)** και η νέα, χωρίς νόημα μορφή που προκύπτει, ονομάζεται **Κρυπτοκείμενο ή Κρυπτόγραμμα (Ciphertext or Cryptogram)**.

Αποκρυπτογράφηση (Decryption) είναι η αντίθετη διαδικασία της κρυπτογράφησης, είναι δηλαδή η διαδικασία της μετατροπής του κρυπτοκειμένου στο αρχικό απλό κείμενο.

Γενικά, το να παραγάγει κάποιος ένα κρυπτοκείμενο και να το αποθηκεύσει κάπου ή να το αποστείλει κάπου, δεν σημαίνει αυτομάτως ότι με αυτόν τον τρόπο αποκλείει και οποιονδήποτε άλλον από το να αποκτήσει πρόσβαση σε αυτό και να το διαβάσει. Μπορεί ο οποιοσδήποτε να υποκλέψει και να διαβάσει το κρυπτοκείμενο. Όμως οποιοσδήποτε κι αν το διαβάσει, δε θα βγάλει νόημα και έτσι δεν θα μπορέσει να αξιοποιήσει την αρχική πληροφορία αλλά ούτε και θα μπορέσει να την αλλάξει ή να την αλλοιώσει (αν το κρυπτοσύστημα το προβλέπει).

Κάθε κρυπτοσύστημα βασίζεται σε ένα **Κλειδί Κρυπτογράφησης (Encryption Key, Cipher Key, Key)** για να μπορέσει να λειτουργήσει, βάσει του οποίου παράγεται και το αντίστοιχο κρυπτοκείμενο. Το κλειδί αποτελεί μία φράση διατεταγμένη από διάφορους χαρακτήρες και ανάλογο μήκος, σύμφωνα με τους κανόνες του κρυπτοσυστήματος. Για παράδειγμα τα «γιωργος», «123», «#g&hy4J”6*9f@r4grh7» αποτελούν πιθανά κλειδιά κρυπτογράφησης ενός κρυπτοσυστήματος, αρκεί οι χαρακτήρες τους και τα μήκη τους να είναι επιτρεπτά από το κρυπτοσύστημα. Είναι γεγονός ότι με διαφορετικό κλειδί κρυπτογράφησης παράγεται και διαφορετικό κρυπτοκείμενο, χρησιμοποιώντας το ίδιο κρυπτοσύστημα.

Παράλληλα, υπάρχει και το **Κλειδί Αποκρυπτογράφησης (Decryption Key)** το οποίο χρησιμοποιείται από το κρυπτοσύστημα κατά τη διαδικασία της αποκρυπτογράφησης. Το κλειδί αυτό **μπορεί να είναι το ίδιο** με το κλειδί κρυπτογράφησης, αλλά μπορεί και όχι, εξαρτάται από το κρυπτοσύστημα που χρησιμοποιείται.

Μιλώντας για σύγχρονη κρυπτογραφία πάντα, υπάρχουν δύο πολύ σημαντικές αρχές οι οποίες πρέπει να αναφερθούν και που πρέπει να έχουν υπ όψιν τους οι σχεδιαστές ενός ισχυρού συστήματος κρυπτογράφησης.

Η αρχή του Kerckhoffs (Kerckhoffs' principle): «*Ένα κρυπτοσύστημα πρέπει να συνεχίσει να είναι ασφαλής, ακόμα και αν τα πάντα σχετικά με αυτό είναι δημοσίως γνωστά (εκτός του κλειδιού κρυπτογράφησης)*». Ο Kerckhoffs εννοεί ότι δεν μπορούμε να ξέρουμε ποιος θα είναι ο αντίπαλος που πρόκειται να επιτεθεί στο κρυπτοσύστημά μας και ότι δεν πρέπει να υποτιμάμε τις γνώσεις του και τις δυνατότητές του. Ακόμα και κρυφή να κρατήσουμε την υλοποίηση του κρυπτοσυστήματός μας, κάποιος μετά από ένα χρονικό διάστημα θα εξοικειωθεί με αυτό μέσω πειραματισμών και θα μπορέσει να ανακαλύψει τον αλγόριθμο με τον οποίο δουλεύει. Αυτό στο οποίο πρέπει να δοθεί ιδιαίτερη σημασία είναι το κλειδί κρυπτογράφησης και αποκρυπτογράφησης και συγκεκριμένα: α) στον τρόπο παραγωγής του (δηλαδή τυχαίο ή ντετερμινιστικό, δομημένο με γνωστές λέξεις ή όχι), β) στο μέγεθός του, γ) στο πως θα διανεμηθεί στους παραλήπτες αν χρειαστεί και δ) στο πως θα κρατηθεί κρυφό.

Το αξίωμα του Shannon (Shannon's maxim): Ο Shannon διαμόρφωσε την δική του εκδοχή της Αρχής του Kerckhoffs η οποία λέει απλά ότι «*Ο εχθρός γνωρίζει το σύστημα*». Μας λέει δηλαδή εν ολίγοις ότι πρέπει να ξεκινάμε την διαδικασία σχεδίασης του κρυπτοσυστήματος για την διασφάλιση των πληροφοριών μας με δεδομένο ότι αυτό που φτιάξαμε ο εχθρός το γνωρίζει ήδη. Οπότε πρέπει να εστιάσουμε στην αρχιτεκτονική του κρυπτοσυστήματος αυτού, κάνοντάς το εξ αρχής να αντιστέκεται σε οποιαδήποτε τεχνική διάσπασης μπορεί να χρησιμοποιηθεί από τον εχθρό (αν είναι δυνατό), καθώς και σε άλλα τυχών επιπλέον μέτρα προστασίας που μπορούμε να πάρουμε για την προστασία αυτού και των κρυπτογραφημένων πληροφοριών του.

3.3 Στόχοι της κρυπτογραφίας

Όπως είδαμε και προηγουμένως, η κρυπτογραφία ασχολείται με τη μελέτη και την υλοποίηση αλγορίθμων κρυπτογράφησης, δηλαδή αλγορίθμων οι οποίοι αποκρύπτουν το νόημα από ένα μήνυμα, κάνοντάς το έτσι ιδιωτικό ως προς τον κάτοχό του. Η ιδιωτικότητα δεδομένων είναι ο κυριότερος και ο πιο «παλιός» στόχος της κρυπτογραφίας. Πέραν τούτου όμως, όταν μιλάμε για ιδιωτικότητα ψηφιακών δεδομένων (πληροφοριών δηλαδή σε μορφή bits 0 και 1) τα οποία βρίσκονται αποθηκευμένα κάπου ή μεταδίδονται μέσω ενός καναλιού επικοινωνίας, η κρυπτογραφία καλείται να λύσει και άλλα ζητήματα που αφορούν την ασφάλεια. Τέτοια προβλήματα προκύπτουν ανάλογα με το ποιος επιτίθεται στα δεδομένα, με ποιον τρόπο επιτίθεται καθώς και το τι είδους δεδομένα προσπαθούν να προστατευθούν.

Οι στόχοι της κρυπτογραφίας συνεπώς επεκτείνονται και αφορούν επιπλέον της ιδιωτικότητας δεδομένων: ακεραιότητα, έλεγχο γνησιότητας και μη-δυνατότητα άρνησης μιας πράξης. Οι στόχοι αυτοί αναλύονται εκτενώς στα παρακάτω κεφάλαια.

3.3.1 Ιδιωτικότητα (Privacy)

Ιδιωτικότητα δεδομένων (λέγεται και Μυστικότητα ή Εμπιστευτικότητα) (Secrecy or Confidentiality) είναι η ιδιότητα της απόκρυψης της έννοιας ή της πρόθεσης ενός μηνύματος. Πιο ειδικά, πρόκειται για την απόκρυψη της πληροφορίας σε τυχών ανεπιθύμητα μέρη που μπορεί να παρεμβληθούν σε ένα κανάλι επικοινωνίας όπως είναι μια ασύρματη (Wi-Fi) σύνδεση δικτύου, ένα δίκτυο κινητής τηλεφωνίας, το Διαδίκτυο κ.α..

Η ιδιωτικότητα των δεδομένων επιτυγχάνεται με χρήση κρυπτοσυστημάτων (δηλαδή μέσω κρυπτογράφησης). Τέτοιοι αλγόριθμοι (όπως προαναφέρθηκε) δέχονται στην αρχή ένα μυστικό κλειδί και στη συνέχεια προχωράνε στην κρυπτογράφηση του αρχικού μηνύματος με τελικό στόχο την μετατροπή του σε ένα κομμάτι δεδομένων που ονομάζεται κρυπτοκείμενο. Από την πλευρά της θεωρίας της πληροφορίας, το κρυπτοκείμενο περιέχει την ίδια ποσότητα εντροπίας (δηλαδή αβεβαιότητας), ή θέτοντάς το απλά, περιέχει την ίδια ποσότητα πληροφορίας με το απλό κείμενο. Αυτό σημαίνει ότι ο παραλήπτης του κρυπτοκειμένου δε χρειάζεται κάτι παραπάνω από το ίδιο το μυστικό κλειδί για να ανακατασκευάσει το αρχικό απλό κείμενο.

Για την επίτευξη της ιδιωτικότητας δεδομένων, συνήθως χρησιμοποιούνται «συμμετρικά» κρυπτοσυστήματα, «τμήματος» ή «ροής». Γνωστά και ευρέως χρησιμοποιούμενα κρυπτοσυστήματα ροής αποτελούν τα RC4, Chameleon, FISH, Helix, SOBER, WAKE, ενώ, κρυπτοσυστήματα Τμήματος αποτελούν τα AES (Rijndael), RC5, RC6, DES, Lucifer, IDEA, και Blowfish.

3.3.2 Ακεραιότητα (Integrity)

Ακεραιότητα δεδομένων είναι η ιδιότητα της διασφάλισης της ορθότητας των δεδομένων κατά την απουσία ενεργών αντιπάλων. Ο ορισμός ίσως ακούγεται πιο περίπλοκος από ό, τι πραγματικά είναι. Γενικά, θέλει να πει ότι η ακεραιότητα αποτελεί τη διασφάλιση ότι ένα μήνυμα έχει να παραδοθεί από ένα σημείο Α σε ένα σημείο Β, χωρίς να έχει αλλοιωθεί το περιεχόμενό του ή η έννοιά του (δεν υπήρξαν δηλαδή κατά τη μεταφορά του από το Α στο Β νέες εισαγωγές στο μήνυμα, τροποποιήσεις, αναδιατάξεις ή επαναλήψεις). Για να μπορέσουμε να θεωρήσουμε ορθό το αποτέλεσμα ενός ελέγχου ακεραιότητας, περιοριζόμαστε υποχρεωτικά στις περιπτώσεις όπου δεν υπάρχουν αντίπαλοι (επιτιθέμενοι) που θέλουν να επιτεθούν στα δεδομένα μας κατά την μετάδοσή τους ή υπάρχουν αλλά δεν προσπαθούν να ανατρέψουν την ορθότητά τους (λόγω του ότι δεν υπάρχει κάποια αξία για παράδειγμα για να το κάνουν).

Η ακεραιότητα των δεδομένων επιτυγχάνεται με τη χρήση μονόδρομων **κρυπτογραφικών συναρτήσεων κατακερματισμού (cryptographic hash functions)**. Οι συναρτήσεις αυτές δέχονται ως είσοδο ένα μήνυμα αυθαίρετου μήκους και παράγουν ως έξοδο μια σύνοψη (μία περίληψη) του μηνύματος αυτού. Μία «σύνοψη μηνύματος», ή απλά, «σύνοψη» – ενδεικτικά η «2fd4e1c67a2d28fced849ee1bb76e7391b93eb12» (σε 16δική μορφή) – είναι μια ακολουθία από bits η οποία έχει πάντα σταθερό μήκος, ανεξαρτήτως του μηνύματος εισόδου. Γενικά, μία σύνοψη (αναλόγως του αλγορίθμου κατακερματισμού) κυμαίνεται σε μεγέθη από 160 έως 512 bits και σκοπός της είναι να χρησιμοποιείται ως «εκπρόσωπος» (representative) του μηνύματος εισόδου.

Πως ελέγχεται όμως η ακεραιότητα? Έστω ότι δίδεται ένα μήνυμα Α και ξεχωριστά μια σύνοψή του, η S_A (η οποία αποτελεί τον εκπρόσωπο του μηνύματος αυτού), τότε αν υπολογισθεί εκ νέου η σύνοψη του Α, έστω D_A, χρησιμοποιώντας την ίδια συνάρτηση κατακερματισμού με αυτή που χρησιμοποιήθηκε για τον υπολογισμό της S_A και αυτή ταιριάζει 100% με την αυθεντική, δηλαδή $D_A = S_A$, τότε μπορεί να ειπωθεί με ασφάλεια ότι το μήνυμα Α έχει παραδοθεί ανέπαφο (έχοντας αποκλείσει πάλι το ενδεχόμενο παρουσίας ενεργών αντιπάλων, οι οποίοι αν υπήρχαν, θα μπορούσαν αρχικά να αλλοιώσουν το περιεχόμενο του μηνύματος Α και εν συνεχεία να αλλάξουν και την σύνοψη εκπροσώπησης S_A κατάλληλα, έτσι ώστε να μην αντιληφθεί κανείς τίποτα).

Οι κρυπτογραφικοί αλγόριθμοι κατακερματισμού παρέχουν και άλλες ενδιαφέρουσες ιδιότητες σχετικές με τις συνόψεις που παράγουν, όπως το ότι παράγουν μονόδρομες (one-way) συνόψεις και το ότι οι συνόψεις αυτές προστατεύονται από συγκρούσεις (collision resistance).

Μονής κατεύθυνσης (ή μονόδρομη) σύνοψη σημαίνει ότι **δεν** μπορούμε από μία σύνοψη να υπολογίσουμε το αρχικό μήνυμα μέσα σε κάποιο εφικτό (μικρότερο από εκθετικό) χρονικό διάστημα. Δηλαδή είναι πρακτικά αδύνατον να συμβεί η αντίστροφη διαδικασία. Ο κυριότερος λόγος που οι συνόψεις – ή

αλλιώς, οι «τιμές κατακερματισμού» (hashes) – έχουν σχεδιαστεί κατ αυτόν τον τρόπο, είναι επειδή χρησιμοποιούνται και ως μέθοδοι για την δημιουργία επικυρωτών οι οποίοι βασίζονται σε κωδικούς πρόσβασης (password-based authenticators). Επιπλέον, το να είναι μια σύνοψη μονής κατεύθυνσης είναι και προαπαιτούμενο από διάφορους αλγόριθμους όπως ο HMAC (περιγράφεται στο επόμενο κεφάλαιο), ούτως ώστε αυτοί να μπορούν να χρησιμοποιηθούν ορθά και με ασφάλεια.

Οι τιμές κατακερματισμού απαιτείται επίσης να είναι ανθεκτικές και σε συγκρούσεις. Διακρίνουμε τρεις μορφές ανθεκτικότητας. Η πρώτη, η οποία λέγεται αντίσταση 1ου ορίσματος (pre-image resistance), μας λέει ότι μια τιμή κατακερματισμού είναι προστατευμένη από ανάστροφο υπολογισμό. Δηλαδή, αν μας δοθεί μια τυχαία τιμή κατακερματισμού D_M, τότε είναι δύσκολο να βρεθεί ένα μήνυμα M τέτοιο ώστε η hash(M) = D_M. Η δεύτερη μορφή ανθεκτικότητας, η οποία ονομάζεται αντίσταση 2ου ορίσματος (2nd pre-image resistance) είναι η ανικανότητα του να βρεθούν δύο μηνύματα M1 (γνωστό) και M2 (τυχαία επιλεγμένο), έτσι ώστε η hash(M1) = hash(M2). Μαζί, αυτές οι μορφές, συνθέτουν την τρίτη και τελευταία μορφή ανθεκτικότητας, η οποία ονομάζεται δυσκολία εύρεσης συγκρούσεων (collision resistance) και μας λέει ότι είναι υπολογιστικά δύσκολο να βρεθούν δύο μηνύματα M1 και M2 τέτοια ώστε hash(M1) = hash(M2).

Στο σημείο αυτό, για την αποφυγή ασαφειών και εξαγωγής λάθος συμπερασμάτων, πρέπει να γίνει μία διευκρίνιση σχετικά με τον τρόπο που έχουν γραφτεί κάποια πράγματα στις παραπάνω παραγράφους. Κατά διαστήματα έχει γραφτεί ότι κάτι «δεν είναι εφικτό» να συμβεί και στη συνέχεια της ίδιας πρότασης μπορεί να αναφερθεί ότι αυτό το κάτι, αντί για μη-εφικτό, είναι τελικά εφικτό αλλά «δύσκολο να συμβεί μέσα σε κάποιο χρονικό διάστημα». Ο λόγος που μπορεί να συμβεί αυτό είναι επειδή οι αλγόριθμοι και τα συστήματα κρυπτογράφησης που χρησιμοποιούμε σχεδιάζονται με βάση την αντοχή τους σε μια συγκεκριμένη υπολογιστική ισχύ, με συνέπεια να υπάρχει δυνατότητα διάσπασής τους αν η ισχύ αυτή ξεπεραστεί ή αν περάσει κάποιο χρονικό διάστημα χρησιμοποιώντας μικρότερη. Όταν όμως το διάστημα αυτό είναι τεράστιο και απαιτεί ακόμα και χρόνια συνεχούς υπολογισμού για την διάσπαση, τότε το σύστημα θεωρείται αδιάσπαστο, αφού (συνήθως) δεν υπάρχει νόημα αν μπορέσει να σπάσει κάτι μετά από ένα τεράστιο χρονικό διάστημα.

Συνεχίζοντας στο κεφάλαιο αυτό, πρέπει να αναφερθεί ότι οι αλγόριθμοι των συναρτήσεων κατακερματισμού δεν είναι κλειστού κώδικα, συνεπώς δεν υπάρχουν κρυφές πληροφορίες σχετικά με την υλοποίησή τους για τις οποίες οι επιτιθέμενοι δεν είναι ενήμεροι. Εάν εμείς μπορούμε να υπολογίσουμε τη σύνοψη ενός δημόσιου μηνύματος, το ίδιο μπορούν να κάνουν και αυτοί. Για το λόγο αυτό, κατά την παρουσία ενός επιτιθέμενου, η ακεραιότητα του μηνύματος δεν μπορεί να προσδιοριστεί (αφού ο επιτιθέμενος θα μπορούσε να αλλάξει το περιεχόμενο του μηνύματος και να εκδώσει εκ νέου την σύνοψή του).

Ακόμη όμως και υπό το φως αυτού του προβλήματος, οι αλγόριθμοι αυτοί εξακολουθούν να χρησιμοποιούνται ευρέως στον κόσμο της

πληροφορικής. Για παράδειγμα, οι περισσότερες online διανομές του Linux και του BSD παρέχουν μία σύνοψη ως μέρος των αρχείων δήλωσής τους (manifest files) η οποία προέρχεται από προγράμματα όπως το «md5sum». Κανονικά, κατά τη διαδικασία ενημέρωσης (update) του λειτουργικού συστήματος, ο χρήστης καλείται να κατεβάσει το ενημερωμένο αρχείο (tarball, RPM, *.deb, κλπ) αλλά και την σύνοψη του αρχείου αυτού. Αυτό συμβαίνει υπό την προϋπόθεση ότι το μοντέλο απειλών δεν περιλαμβάνει ενεργούς αντιπάλους (που προσπαθούν να αλλοιώσουν τα αρχεία ενημέρωσης τροποποιώντας τα), αλλά περιλαμβάνει καθαρά και μόνο σφάλματα αποθήκευσης και μεταφοράς που ως συνέπεια μπορεί να οδηγήσουν σε αποκομμένα (truncated) ή κατεστραμμένα (overwritten) αρχεία.

Οι πιο δημοφιλείς και ασφαλείς στη χρήση συναρτήσεις κατακερματισμού που χρησιμοποιούνται σήμερα, είναι οι SHA-1 και SHA-2 (Secure Hash Algorithm - number 1 & 2) του Προτύπου Ασφαλούς Κατακερματισμού (Secure Hash Standard - SHS) του Εθνικού Ιδρύματος Προτύπων και Τεχνολογίας της Αμερικής (National Institute of Standards and Technology - NIST). Η οικογένεια των συναρτήσεων SHA-2 ειδικότερα, είναι πιο ελκυστική από αυτή της SHA-1, καθώς παράγει τιμές κατακερματισμού μεγέθους από 224 έως 512 bits. Πρόκειται για αρκετά αποδοτικές συναρτήσεις, δεδομένου ότι δεν απαιτούν πίνακες ή περίπλοκες οδηγίες και είναι αρκετά εύκολο να αναπαραχθούν από τις προδιαγραφές τους.

Από την άλλη, μια συνάρτηση κατακερματισμού της οποίας η χρήση θα πρέπει να αποφεύγεται, είναι η MD5 (Message Digest algorithm 5), η οποία έχει θεωρηθεί από καιρό αρκετά αδύναμη. Ο Γερμανός κρυπτογράφος Hans Dobbertin (April 17, 1952 - February 2, 2006) διαπίστωσε ελαττώματα σε βασικά σημεία του αλγορίθμου και το 2005 οι ερευνητές εντόπισαν πλήρης συγκρούσεις στην συνάρτηση. Νέες δημοσιεύσεις, που εμφανίστηκαν στις αρχές του 2006, συζητούν για όλο και πιο γρήγορες μεθόδους εξεύρεσης συγκρούσεων σε τέτοιους αλγορίθμους. Επίσης, ακόμα και η SHA-1 πρέπει να αποφεύγεται να χρησιμοποιείται σε κάποιο βαθμό, και στη θέση της να χρησιμοποιείται η νέα SHA-2. Για όσους θέλουν να ακολουθήσουν τα ευρωπαϊκά πρότυπα, υπάρχει επίσης και η συνάρτηση Whirlpool (Vincent Rijmen, Paulo S. L. M. Barreto) που μπορούν να επιλέξουν και να χρησιμοποιήσουν με ασφάλεια. [33]

3.3.3 Επικύρωση (Authentication)

Επικύρωση (λέγεται και Αυθεντικοποίηση ή Ταυτοποίηση ή Έλεγχος Γνησιότητας ή Έλεγχος Αυθεντικότητας), είναι η ιδιότητα της απόδοσης μιας ταυτότητας σε ένα μήνυμα μέσω της οποίας μπορεί να γίνει αργότερα η εξακρίβωση της ακεραιότητάς του.

Ένα κλασικό παράδειγμα επικύρωσης είναι η κέρινη σφραγίδα που τοποθετείται στις επιστολές που στέλνονται με το ταχυδρομείο, όταν κάποιος θέλει να στείλει ένα έγγραφο και να παραμείνει αναλλοίωτο κατά την παραλαβή του. Η κέρινη σφραγίδα είναι δύσκολο να πλαστογραφηθεί τη στιγμή που έχει ήδη χρησιμοποιηθεί. Άρα, η παρουσία μιας άθικτης σφραγίδας στον παραλήπτη συνεπάγεται και την γνησιότητα του εγγράφου της επιστολής.

Μια άλλη μορφή επικύρωσης είναι η διαδικασία της εισαγωγής ενός προσωπικού αριθμού αναγνώρισης (PIN) ή ενός κωδικού πρόσβασης (password) ούτως ώστε να πραγματοποιηθεί μία συναλλαγή, κάτι που επικυρώνει ότι ο χρήστης που πληκτρολόγησε τον σωστό κωδικό είναι ο γνήσιος, ή αλλιώς, αυτός που έχει την εξουσιοδότηση να εκτελέσει την συναλλαγή.

Η διαδικασία της επικύρωσης δε θα πρέπει να συγχέεται με την μη-απάρνηση της υπογραφής (non-repudiation), την μη-δυνατότητα άρνησης μιας συμφωνίας, ή με τα πρωτόκολλα επικύρωσης όσον αφορά τα πρωτόκολλα συμφωνίας κλειδιού (key-agreement protocols) ή τα πρωτόκολλα ανταλλαγής κλειδιού (key exchange, key establishment protocols). Όταν λέμε ότι επικυρώνουμε ένα μήνυμα, εννοούμε ότι εκτελούμε επιπρόσθετα βήματα έτσι ώστε ο παραλήπτης να μπορεί να επαληθεύσει την ακεραιότητα ενός μηνύματος κατά την ενεργή παρουσία ενός αντιπάλου.

Η διαδικασία της διαπραγμάτευσης κλειδιού (key negotiation) και αυτή της επικύρωσης, αποτελούν θέματα πρωτοκόλλων δημοσίου κλειδιού (public key protocols). Τα πρωτόκολλα αυτά χρησιμοποιούν σε μεγάλο βαθμό τις ίδιες αρχές, αλλά έχουν διαφορετικούς περιορισμούς και στόχους. Οι αλγόριθμοι επικύρωσης είναι συνήθως συμμετρικοί, έτσι ώστε όλα τα συμμετέχοντα μέρη μιας επικοινωνίας να μπορούν να παράγουν δεδομένα που μπορούν αργότερα να επαληθευτούν. Η επικύρωση ενός μηνύματος (με δυνατότητα μη-απάρνησης της υπογραφής) γίνεται συνήθως από έναν μόνο συμμετέχοντα (παραγωγός), ενώ το μήνυμα μπορεί να επαληθευτεί από πολλούς ελεγκτές (verifiers).

Στον κόσμο της κρυπτογραφίας, οι αλγόριθμοι επικύρωσης αποκαλούνται συχνά και **Κώδικες Επικύρωσης Μηνυμάτων (Message Authentication Codes - MACs)**, και, όπως και στις συναρτήσεις κατακερματισμού (hash functions), οι αλγόριθμοι αυτοί παράγουν μία έξοδο σταθερού μεγέθους η οποία ονομάζεται Ετικέτα Μηνύματος (Message Tag). Η ετικέτα αποτελεί την πληροφορία που ένας ελεγκτής μπορεί να χρησιμοποιήσει για την επικύρωση ενός εγγράφου. Σε αντίθεση με τις συναρτήσεις κατακερματισμού, το σύνολο των συναρτήσεων MAC απαιτεί ένα μυστικό κλειδί

για να τρέξει η διαδικασία, ούτως ώστε να αποτραπεί από τον οποιονδήποτε η δυνατότητα αλλοίωσης της ετικέτας.

Οι δύο πιο κοινές μορφές των αλγορίθμων MAC είναι ο αλγόριθμος CBC-MAC (Cipher Block Chaining MAC) (ο οποίος τώρα υλοποιείται με βάση τον OMAC1 (One-key MAC number 1) και ονομάζεται CMAC (Cipher-based MAC) στον κόσμο του Εθνικού Ιδρύματος Προτύπων και Τεχνολογίας της Αμερικής (National Institute of Standards and Technology - NIST)), καθώς και ο αλγόριθμος HMAC (Hash-based MAC).

Οι συναρτήσεις του αλγορίθμου CMAC (CBC-MAC) χρησιμοποιούν κρυπτοσυστήματα τμήματος (block ciphers), ενώ οι συναρτήσεις του HMAC χρησιμοποιούν μια συνάρτηση κατακερματισμού. Μια άλλη μέθοδος για την επίτευξη επικύρωσης είναι η χρήση αλγορίθμου δημοσίου κλειδιού όπως ο RSA (Rivest-Shamir-Adleman algorithm), βασιζόμενος στο πρότυπο PKCS #1 (Public Key Cryptography Standards number 1) ή στο πρότυπο Ελλειπτικής Καμπύλης του Αλγορίθμου Ψηφιακής Υπογραφής (Elliptic Curve Digital Signature Algorithm - ECDSA ή ANSI X9.62) του Ομοσπονδιακού Προτύπου Επεξεργασίας Πληροφοριών (Federal Information Processing Standards - FIPS).

Σε αντίθεση με τον CMAC ή τον HMAC, ένας ελεγκτής ταυτότητας (authenticator) βασισμένος στο δημόσιο κλειδί, δεν απαιτεί και από τα δύο μέρη να μοιράζονται προσωπικές πληροφορίες πριν από την επικοινωνία. Συνεπώς, οι αλγόριθμοι δημοσίου κλειδιού δεν περιορίζονται μόνο σε online δοσοληψίες όταν ασχολούνται με τυχαία, άγνωστα μέρη. Δηλαδή, μπορεί κάποιος να υπογράψει ψηφιακά ένα έγγραφο δημιουργώντας μία ψηφιακή υπογραφή και εν συνεχεία, ο καθένας που έχει πρόσβαση στο δημόσιο κλειδί, μπορεί να επαληθεύσει την γνησιότητά του, χωρίς πρωτίστως να έχει επικοινωνήσει με τον εκδότη της ψηφιακής υπογραφής. Οι αλγόριθμοι δημοσίου κλειδιού χρησιμοποιούνται με διαφορετικούς τρόπους από ό, τι στους αλγορίθμους MAC.

Ο τρόπος με τον οποίο χρησιμοποιούνται οι ελεγκτές ταυτότητας (authenticators) εξαρτάται από την κατασκευή τους. Οι ελεγκτές ταυτότητας που βασίζονται στο δημόσιο κλειδί χρησιμοποιούνται συνήθως για να διαπραγματευτούν ένα αρχικό «χαιρετισμό» σε ένα νέο ανοιγμένο κανάλι επικοινωνίας. Για παράδειγμα, όταν κάποιος συνδεθεί για πρώτη φορά σε μία ιστοσελίδα που παρέχει ασφαλή περιήγηση με χρήση πρωτοκόλλου SSL (Secure Sockets Layer), τότε ο ίδιος πιστοποιεί ότι η ψηφιακή υπογραφή του πιστοποιητικού που παρέχει η συγκεκριμένη ιστοσελίδα προκύπτει πραγματικά από μια Κεντρική Αρχή Έκδοσης Ψηφιακών Υπογραφών (Root Signing Authority). Αντίθετα, αλγόριθμοι όπως οι CMAC και HMAC χρησιμοποιούνται αφότου η επικοινωνία μέσω του καναλιού έχει διασφαλιστεί. Χρησιμοποιούνται δηλαδή για να εξασφαλιστεί ότι τα πακέτα δεδομένων της συγκεκριμένης επικοινωνίας θα παραδοθούν στους παραλήπτες χωρίς αλλοίωση. Αφού οι CMAC και HMAC είναι πολύ πιο γρήγοροι στην επεξεργασία των δεδομένων ενός μηνύματος, είναι πολύ πιο χρήσιμοι σε κανάλια υψηλής κυκλοφορίας.

3.3.4 Μη-Απάρνηση (Non-Repudiation)

Μη-Απάρνηση (λέγεται και μη-δυνατότητα άρνησης ή μη-αποποίηση) είναι η ιδιότητα της υποχρέωσης της τήρησης μίας συμφωνίας. Πιο συγκεκριμένα, είναι η αδυναμία της άρνησης μιας πράξης. Για παράδειγμα, αν κάποιος πάρει ένα στυλό και υπογράψει ένα νομικό συμβόλαιο, τότε η υπογραφή του αποτελεί αδιάψευστη μέθοδο μη-απάρνησης για το συμβόλαιο αυτό. Αυτός που υπέγραψε, δεν μπορεί αργότερα να αρνηθεί ότι το έκανε ή να διαφωνήσει με τους όρους που τέθηκαν στο συμβόλαιο αυτό.

Η Μη-Απάρνηση μοιάζει πολύ με την ιδιότητα της Επικύρωσης μιας και οι υλοποιήσεις τους μοιράζονται συχνά πολλά από τα ίδια αρχέτυπα. Για παράδειγμα μία υπογραφή δημοσίου κλειδιού θα μπορούσε να αποτελέσει μέθοδο μη-απάρνησης μόνο αν η παραγωγή υπογραφών συνέβαινε από ένα και μοναδικό μέρος της επικοινωνίας. Για το λόγο αυτό, οι Κώδικες Επικύρωσης Μηνυμάτων (MACs), όπως οι CMAC και HMAC, δεν μπορούν να αποτελέσουν μεθόδους μη-απάρνησης (αφού έχουν γνωστό τρόπο λειτουργίας και ο οποιοδήποτε μπορεί να τους χρησιμοποιήσει για να παραγάγει μία νέα υπογραφή). Τα κινητά τηλέφωνα επίσης χρησιμοποιούν MAC αλγορίθμους ως ελεγκτές ταυτότητας για την χρήση των πόρων τους, και συνεπώς, δεν μπορούν να έχουν ιδιότητες μη-απάρνησης.

Η μη-απάρνηση αποτελεί επίσης πολύ σημαντική ιδιότητα της τιμολόγησης και της πληρωμής προϊόντων, η οποία συχνά αντιμετωπίζεται με λάθος τρόπο. Για παράδειγμα, οι υπογραφές που βάζουμε με το χέρι μας στις αποδείξεις πληρωμής μέσω πιστωτικής κάρτας (αφού πληρώσουμε με αυτήν), επαληθεύονται σπάνια για την γνησιότητά τους από τον υπάλληλο του ταμείου. Αλλά ακόμη και αν ο υπάλληλος κοιτάξει στο πίσω μέρος της κάρτας για να δει τη γνήσια και να τις συγκρίνει, εφόσον αυτός δεν αποτελεί εμπειρογνώμονα γραφολόγο, δεν θα μπορέσει να ξεχωρίσει την αυθεντική από μια πιθανώς πλαστογραφημένη.

3.4 Αλγόριθμοι κρυπτογράφησης Ιδιωτικού Κλειδιού Τμήματος και Ροής

Τα συμμετρικά κρυπτοσυστήματα (ή αλλιώς τα κρυπτοσυστήματα ιδιωτικού κλειδιού) είναι συστήματα κρυπτογράφησης τα οποία χρησιμοποιούν το ίδιο κλειδί για την κρυπτογράφηση και την αποκρυπτογράφηση πληροφοριών. Υλοποιούνται σε δύο μορφές, με την καθεμία να έχει τα δικά της πλεονεκτήματα και μειονεκτήματα.

Η πρώτη μορφή είναι τα **Συμμετρικά Κρυπτοσυστήματα Ροής (Symmetric Stream Ciphers)** με ποιο διαδεδομένα τα RC4, Chameleon, FISH, Helix, SOBER, WAKE. Τα κρυπτοσυστήματα αυτά χρησιμοποιούνται για την κρυπτογράφηση μια συνεχόμενη ροής δεδομένων (data stream) την οποία επιτελούν bit προς bit ή byte προς byte. Για την κρυπτογράφηση επιλέγεται αρχικά μία γεννήτρια κλειδοροής (keystream generator), η οποία δέχεται ως είσοδο το μυστικό κλειδί και παράγει ως έξοδο μία ψευδοτυχαία ακολουθία από bits, η οποία ονομάζεται κλειδοροή (keystream). Στην συνέχεια εφαρμόζεται μία λογική πράξη XOR στα bits ανάμεσα στο αρχικό κείμενο και στην κλειδοροή, με το αποτέλεσμα της συνάρτησης να αποτελεί την τελική κρυπτογραφημένη ροή δεδομένων.

Η δεύτερη μορφή είναι τα **Συμμετρικά Κρυπτοσυστήματα Τμήματος (Symmetric Block Ciphers)** με πιο διαδεδομένα τα RC5, RC6, DES, Lucifer, IDEA, AES (Rijndael) και Blowfish. Οι κρυπτογραφικοί αλγόριθμοι αυτού του είδους τεμαχίζουν σε τμήματα (blocks) το αρχικό κείμενο που πρόκειται να κρυπτογραφηθεί και κρυπτογραφούν κάθε τμήμα ξεχωριστά. Συνηθισμένα μεγέθη ενός τμήματος δεδομένων είναι τα 64 ή 128 bits. Η κρυπτογράφηση κάθε ενός τμήματος γίνεται χρησιμοποιώντας μία μαθηματική συνάρτηση κρυπτογράφησης και το μυστικό κλειδί. Το αποτέλεσμα της διαδικασίας κρυπτογράφησης είναι η παραγωγή ενός κρυπτογραφημένου τμήματος το οποίο στην πλειοψηφία των περιπτώσεων έχει το ίδιο μήκος με το αντίστοιχο τμήμα του αρχικού κειμένου.

Στην κατηγορία αυτή ανήκει και το κρυπτοσύστημα AES (Advanced Encryption Standard). Το κρυπτοσύστημα αυτό εγκαθιδρύθηκε από το Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας της Αμερικής (National Institute of Standards and Technology - NIST) και είναι ιδιαίτερα δημοφιλής καθώς είναι εξαιρετικά αποδοτικό για εκτέλεση σε μεγάλους αλλά και μικρούς επεξεργαστές (processors) και απαιτεί επίσης εξαιρετικά χαμηλό κόστος για την υλοποίησή του απευθείας επάνω σε υλικό (hardware) όταν χρειάζεται. Το κρυπτοσύστημα αυτό αναλύεται εκτενώς στο επόμενο κεφάλαιο και είναι αυτό το οποίο χρησιμοποιήθηκε για την κατασκευή του Διαχειριστή Κλειδαρίθμου αυτής της εργασίας.

3.5 Το Πρότυπο Προχωρημένης Κρυπτογράφησης - AES

3.5.1 Εισαγωγή

Το **«Πρότυπο Προχωρημένης Κρυπτογράφησης» (Advanced Encryption Standard - AES)** δημιουργήθηκε από τις Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας της Αμερικής (National Institute of Standards and Technology - NIST) το 2001 και αποτελεί μια προδιαγραφή για την κρυπτογράφηση ηλεκτρονικών δεδομένων.

Το AES βασίζεται στο κρυπτοσύστημα Rijndael, το οποίο αναπτύχθηκε από δύο βέλγους κρυπτογράφους, τους Joan Daemen και Vincent Rijmen , οι οποίοι υπέβαλαν πρόταση στο NIST κατά τη διάρκεια της διαδικασίας επιλογής αλγορίθμων για το νεοσύστατο τότε AES. Το κρυπτοσύστημα Rijndael αποτελεί μια οικογένεια αλγορίθμων κρυπτογράφησης με μεταβλητό μέγεθος μπλοκ και κλειδιών.

Το NIST επέλεξε για το AES τρεις από τους αλγορίθμους της οικογένειας Rijndael, τον καθένα με σταθερό μέγεθος μπλοκ στα 128 bits αλλά με διαφορετικού μήκους κλειδί: στα 128, 192 και 256 bits.

Το AES έχει υιοθετηθεί από την κυβέρνηση των ΗΠΑ ως το κύριο πρότυπο για την κρυπτογράφηση των κυβερνητικών δεδομένων τους και πλέον χρησιμοποιείται σε όλο τον κόσμο. Αντικαθιστά το **«Πρότυπο Κρυπτογράφησης Δεδομένων» (Data Encryption Standard - DES)** το οποίο δημοσιεύθηκε το 1977. Ο αλγόριθμος που περιγράφεται από το AES είναι αλγόριθμος τμήματος συμμετρικού κλειδιού, κάτι που σημαίνει ότι το ίδιο κλειδί χρησιμοποιείται τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση των δεδομένων.

Στις Ηνωμένες Πολιτείες, το AES ανακοινώθηκε από το NIST με την κωδική ονομασία «US FIPS PUB 197» (ή απλά FIPS 197) στις 26 Νοεμβρίου του 2001. Η ανακοίνωση αυτή ακολούθησε μια διαδικασία προτυποποίησης 5 ετών κατά την οποία παρουσιάστηκαν και αξιολογήθηκαν 15 διαφορετικά ανταγωνιστικά σχέδια και κατόπιν επιλέχθηκε το κρυπτοσύστημα Rijndael ως το πλέον κατάλληλο.

Το AES τέθηκε σε ισχύ ως ομοσπονδιακό κυβερνητικό πρότυπο στις 26 Μαΐου του 2002 μετά από την έγκρισή του από το Υπουργείο Εμπορίου. Το AES περιλαμβάνεται επίσης στο πρότυπο ISO/IEC 18033-3 του Διεθνούς Οργανισμού Προτυποποίησης (International Organization for Standardization - ISO) και της Διεθνής Ηλεκτροτεχνικής Επιτροπής (International Electrotechnical Commission - IEC) για χρήση σε περιπτώσεις όπου απαιτείται εμπιστευτικότητα δεδομένων. Το AES είναι διαθέσιμο σε πολλά και διαφορετικά πακέτα κρυπτογράφησης, και είναι το πρώτο κοινά αποδεκτό και ανοικτού κώδικα σύστημα κρυπτογράφησης το οποίο έχει εγκριθεί και από την Υπηρεσία Εθνικής Ασφάλειας (National

Security Agency - NSA) των ΗΠΑ για την κρυπτογράφηση άκρως απόρρητων πληροφοριών.

3.5.2 Περιγραφή του κρυπτοσυστήματος

Το κρυπτοσύστημα AES βασίζεται στην σχεδιαστική αρχή των «δικτύων αντικατάστασης-μετάθεσης» (substitution-permutation networks), η οποία είναι γρήγορη τόσο σε λογισμικό όσο και σε υλικό. Σε αντίθεση με τον προκάτοχό του «DES», το AES δεν χρησιμοποιεί δίκτυο Feistel. Το AES αποτελεί μια παραλλαγή του κρυπτοσυστήματος Rijndael και έχει σταθερό μέγεθος μπλοκ στα 128 bits με διαφορετικού μήκους κλειδί: στα 128, 192 και 256 bits. Αντίθετα, η προδιαγραφή του Rijndael «per se» ορίζει μεγέθη μπλοκ και κλειδιών τα οποία μπορεί να είναι πολλαπλάσια των 32 bits, με εύρος από 128 έως 256 bits και για τα δύο.

Το κρυπτοσύστημα AES ενεργεί πάνω σε έναν **σταθερού μεγέθους** 4×4 πίνακα από bytes (4 x 4 = 16bytes ή 128bits), με διάταξη κατά στήλες (column-major order), τον οποίο ονομάζει Πίνακα Κατάστασης ή απλά **«Κατάσταση» (State)**. Αντιθέτως, σε ορισμένες εκδόσεις του κρυπτοσυστήματος Rijndael όπου χρησιμοποιείται μεγαλύτερο μέγεθος μπλοκ, υπάρχουν και επιπλέον στήλες στην αντίστοιχη «κατάσταση» τους (πάνω από τέσσερις). Οι περισσότεροι υπολογισμοί του AES εκτελούνται πάνω σε ένα ειδικό πεπερασμένο πεδίο.

Το μέγεθος του κλειδιού που χρησιμοποιείται για την κρυπτογράφηση στο AES καθορίζει και το πλήθος των **γύρων μετασχηματισμού (transformation rounds)** που μετατρέπουν την είσοδο, η οποία ονομάζεται **«απλό κείμενο» (plaintext)**, στο τελικό αποτέλεσμα, το οποίο ονομάζεται **«κρυπτοκείμενο» (ciphertext)**.

Το πλήθος των γύρων μετασχηματισμού, ή αλλιώς, των κύκλων επαναλήψεων (cycles of repetition) για τον πλήρη AES (full AES), αναλόγως του μεγέθους του κλειδιού, καθορίζεται ως εξής:

- 10 γύροι μετασχηματισμού για κλειδιά 128-bit.
- 12 γύροι μετασχηματισμού για κλειδιά 192-bit.
- 14 γύροι μετασχηματισμού για κλειδιά 256-bit.

Κάθε γύρος μετασχηματισμού αποτελείται από διάφορα βήματα επεξεργασίας όπου το καθένα περιλαμβάνει 4 παρόμοια (αλλά όχι ίδια) στάδια, συμπεριλαμβανομένου και του σταδίου που εξαρτάται από το κλειδί κρυπτογράφησης. Για τη μετατροπή του κρυπτοκειμένου πίσω στο αρχικό απλό κείμενο, εφαρμόζεται πάλι ένα σύνολο γύρων επεξεργασίας (ονομάζονται γύροι αντιστροφής) χρησιμοποιώντας το ίδιο κλειδί που χρησιμοποιήθηκε για την

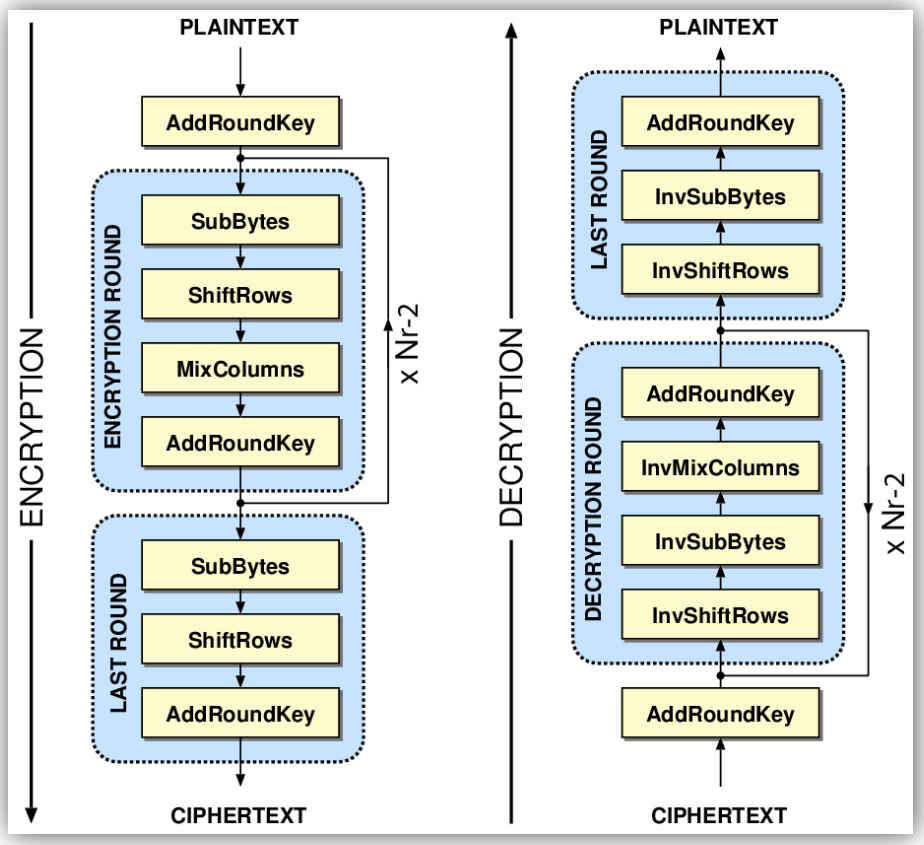
κρυπτογράφηση (και που λειτουργεί αυτή τη φορά ως κλειδί αποκρυπτογράφησης).

3.5.3 Περιγραφή του αλγορίθμου κρυπτογράφησης AES

Ο αλγόριθμος κρυπτογράφησης του AES αποτελείται από 4 στάδια εκτέλεσης. Τα στάδια (1) και (2) τρέχουνε μόνο μία φορά στην αρχή της διαδικασίας, το στάδιο (4) τρέχει κι αυτό μόνο μια φορά στο τέλος της διαδικασίας, ενώ το στάδιο (3) επαναλαμβάνεται τόσες φορές όσες καθορίζει το μέγεθος του κλειδιού κρυπτογράφησης (αναλυτικά στο κεφ. 3.6.2).

Το κάθε στάδιο εν συνεχεία διαιρείται σε κάποια βήματα επεξεργασίας τα οποία εκτελούνται με συγκεκριμένη σειρά και σκοπός τους είναι να μετασχηματίσουν την πληροφορία με διαφορετικό τρόπο το καθένα.

Μετά το πέρας όλων των σταδίων η κρυπτογράφηση έχει ολοκληρωθεί και έτσι το απλό κείμενο έχει μετατραπεί πλέον σε κρυπτοκείμενο.



Εικόνα 3.1 - Δομικό διάγραμμα του αλγορίθμου AES στο οποίο φαίνονται τα στάδια επεξεργασίας των δεδομένων κατά την κρυπτογράφηση (αριστερά) και κατά την αποκρυπτογράφηση (δεξιά).

Στάδιο 1: Επέκταση Κλειδιού (KeyExpansion)

Στο στάδιο αυτό χρησιμοποιώντας το αρχικό κλειδί κρυπτογράφησης και την **«Διαδικασία Επέκτασης Κλειδιού»** του Rijndael **(Rijndael key schedule)**, παράγονται τα κλειδιά για τον κάθε γύρο του αλγορίθμου.

Αν υποθέσουμε ότι έχουμε αρχικό κλειδί κρυπτογράφησης των 128 bit, τότε αυτό συνεπάγεται τη δημιουργία ενός νέου πίνακα *w* (ως έξοδο του αλγορίθμου επέκτασης κλειδιού του Rijndael) του οποίου η κάθε στήλη θα αποτελείται από **«λέξεις» (words)** των 4 bytes η κάθε μία (δηλαδή 4 bytes ανά στήλη). Γνωρίζουμε ότι για κάθε γύρο απαιτείται ένα κλειδί των 128 bits, δηλαδή 4 λέξεων. Γνωρίζουμε επίσης ότι για αυτό το μέγεθος κλειδιού απαιτούνται 10 γύροι, άρα θέλουμε 10 διαφορετικά κλειδιά συν 1 ακόμη για την αρχική πρόσθεση. Κατά συνέπεια ο πίνακας *w* θα αποτελείται από 11 x 4 = 44 λέξεις, οπότε το μέγεθός του θα είναι 44 λέξεις x 4 bytes η κάθε λέξη = 176 bytes = 1408 bits.

Αντιστοίχως, για μέγεθος κλειδιού των 192 bit ο πίνακας *w* θα είναι μεγέθους 52 λέξεων x 4 bytes η κάθε λέξη = 208 bytes = 1664 bits και για κλειδί των 256 bits θα είναι 60 λέξεων x 4 bytes η κάθε λέξη = 240 bytes = 1920 bits.

Άρα το κλειδί για τον 1ο γύρο θα αποτελούν οι λέξεις στις στήλες *w[0]* ως *w[3]*, για τον 2ο γύρο οι λέξεις στις στήλες *w[4]* ως *w[7]* και για τον τελευταίο γύρο οι λέξεις στις στήλες *w[K-4]* ως *w[K-1]* (όπου *K* το μέγεθος του πίνακα *w* σε λέξεις).

Εικόνα 3.2 - Ο πίνακας w που προκύπτει μετά την διαδικασία Επέκτασης Κλειδιού του Rijndael, για κλειδί κρυπτογράφησης αρχικού μεγέθους 128 bits. Τα k_0 έως k_{15} και τα w_0 έως w_{43} αποτελούν λέξεις των 4 bytes η κάθε μία.

Στάδιο 2: Αρχικός Γύρος (InitialRound)

Βήμα (a) - Πρόσθεση του Κλειδιού Γύρου (AddRoundKey):
κάθε byte του πίνακα κατάστασης προστίθεται με κάθε byte του υποπίνακα w[από στήλη i ως j], ο οποίος αποτελεί το κλειδί γύρου, χρησιμοποιώντας δυαδική (bitwise) πράξη XOR.

Στάδιο 3: N-2 Γύροι Κρυπτογράφησης (N-2 Encryption Rounds)

Βήμα (a) - Αντικατάσταση των bytes (SubBytes):
στο βήμα αυτό κάθε byte του πίνακα κατάστασης αντικαθίσταται με κάποιο άλλο byte, χρησιμοποιώντας έναν Πίνακα Αναζήτησης (lookup table), δηλαδή το S-box του Rijndael. Η αντικατάσταση που επιτελείται είναι μη-γραμμική.

Βήμα (b) - Ολίσθηση γραμμών (ShiftRows):
αποτελεί βήμα αντιμετάθεσης σύμφωνα με το οποίο οι τρεις τελευταίες γραμμές του *πίνακα κατάστασης* μετατοπίζονται (ολισθαίνουν) κυκλικά προς τα αριστερά για ένα συγκεκριμένο αριθμό θέσεων η κάθε μία.

Βήμα (c) - Ανάμιξη στηλών (MixColumns):
λειτουργία ανάμιξης η οποία αλλάζει τις τιμές των στηλών του *πίνακα κατάστασης*. Η νέες τιμές των στηλών προκύπτουν μέσα από πράξεις σε σώμα $GF(2^8)$.

Βήμα (d) - Πρόσθεση του Κλειδιού Γύρου (AddRoundKey):
κάθε byte του πίνακα κατάστασης προστίθεται με κάθε byte του υποπίνακα w[από στήλη i ως j], ο οποίος αποτελεί το κλειδί γύρου, χρησιμοποιώντας δυαδική (bitwise) πράξη XOR.

Στάδιο 4: Τελικός Γύρος (Last Round)

Ίδιος με το Στάδιο 3, αλλά χωρίς το βήμα Ανάμιξης Στηλών (MixColumns).

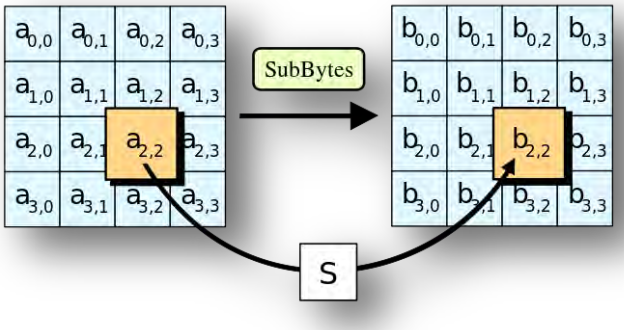
Βήμα (a) - Αντικατάσταση των bytes (SubBytes):
στο βήμα αυτό κάθε byte του πίνακα κατάστασης αντικαθίσταται με κάποιο άλλο byte, χρησιμοποιώντας έναν Πίνακα Αναζήτησης (lookup table), δηλαδή το S-box του Rijndael. Η αντικατάσταση που επιτελείται είναι μη-γραμμική.

Βήμα (b) - Ολίσθηση γραμμών (ShiftRows):
αποτελεί βήμα αντιμετάθεσης σύμφωνα με το οποίο οι τρεις τελευταίες γραμμές του *πίνακα κατάστασης* μετατοπίζονται (ολισθαίνουν) κυκλικά προς τα αριστερά για ένα συγκεκριμένο αριθμό θέσεων η κάθε μία.

Βήμα (c) - Πρόσθεση του Κλειδιού Γύρου (AddRoundKey):
κάθε byte του *πίνακα κατάστασης* προστίθεται με κάθε byte του υποπίνακα *w[από στήλη i ως j]*, ο οποίος αποτελεί το κλειδί γύρου, χρησιμοποιώντας δυαδική (bitwise) πράξη XOR.

3.5.4 Το βήμα «Αντικατάστασης των Bytes» (SubBytes)

Στο βήμα **SubBytes**, κάθε byte $a_{i,j}$ του πίνακα κατάστασης αντικαθίσταται με ένα άλλο byte του ίδιου πίνακα μέσω ενός **«Κουτιού Αντικατάστασης» (Substitution box)** (Rijndael *S*-box), ή αλλιώς **«Μονάδας Αντικατάστασης»**, των 8 bit. Δηλαδή επιτελείται η πράξη $b_{i,j} = S(a_{i,j})$ όπου b, a αποτελούν bytes του πίνακα κατάστασης.



Εικόνα 3.3 - Αντικατάσταση του κάθε byte του Πίνακα Κατάστασης με κάποιο άλλο byte του ίδιου πίνακα, μέσω μιας 8-μπιτης μονάδας αντικατάστασης «S».

Η διαδικασία αυτή συμβαίνει για τη εισαγωγή μη-γραμμικότητας στο κρυπτοσύστημα. Η μονάδα αντικατάστασης *S* στην πράξη αποτελεί έναν **«Πίνακα Αναζήτησης» (Lookup Table)** ο οποίος παράγεται από τον αντίστροφο πολλαπλασιασμό πάνω στο $GF(2^8)$ (Galois field - Πεπερασμένο σώμα), που ως γνωστόν έχει καλές ιδιότητες μη-γραμμικότητας.

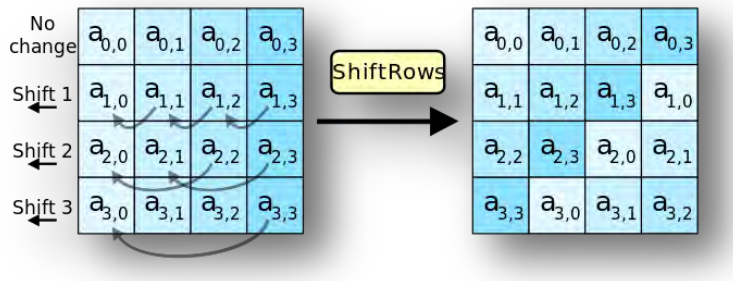
Για να αποφευχθούν επιθέσεις που βασίζονται σε απλές αλγεβρικές ιδιότητες, η μονάδα αντικατάστασης *S* κατασκευάζεται συνδυάζοντας την αντίστροφη συνάρτηση με ένα αντιστρέψιμο μετασχηματισμό πίνακα με συσχέτιση (affine transformation). Η μονάδα αντικατάστασης *S* έχει επίσης επιλεγεί για να αποφευχθούν τυχόν σταθερά σημεία (αποτελεί μία «διαταραχή» (derangement)), δηλαδή $S(a_{i,j}) \neq a_{i,j}$, καθώς και τυχόν αντίθετα σταθερά σημεία (συμπληρωματικά σημεία), δηλαδή $S(a_{i,j}) \oplus a_{i,j} \neq 0xFF$.

Για την εκτέλεση της αποκρυπτογράφησης χρησιμοποιείται το αντίστροφο βήμα SubBytes, το οποίο απαιτεί πρώτα την διενέργεια του μετασχηματισμού πίνακα με συσχέτιση (affine) και στη συνέχεια την εύρεση του αντίστροφου πολλαπλασιασμού (απλή αντιστροφή των βημάτων του SubBytes δηλαδή).

3.5.5 Το βήμα «Ολίσθησης Γραμμών» (ShiftRows)

Το βήμα ShiftRows ασχολείται με τις τιμές των γραμμών του πίνακα κατάστασης. Μετατοπίζει (ή αλλιώς, ολισθαίνει) κυκλικά τα bytes σε κάθε γραμμή, αριστερόστροφα, με τη χρήση ενός συγκεκριμένου offset για την καθεμία.

Αναλυτικότερα, στο κρυπτοσύστημα AES η πρώτη σειρά παραμένει αμετάβλητη. Κάθε byte της δεύτερης σειράς μετατοπίζεται κυκλικά κατά μία θέση προς τα αριστερά. Αντιστοίχως στην τρίτη και στην τέταρτη σειρά τα bytes μετατοπίζονται προς τα αριστερά με ολισθήσεις των δύο και των τριών θέσεων.



Εικόνα 3.4 - Στο βήμα ShiftRows τα bytes της κάθε γραμμής του Πίνακα Κατάστασης μετατοπίζονται κυκλικά προς τα αριστερά. Το πλήθος των μετατοπίσεων διαφέρει ανά γραμμή.

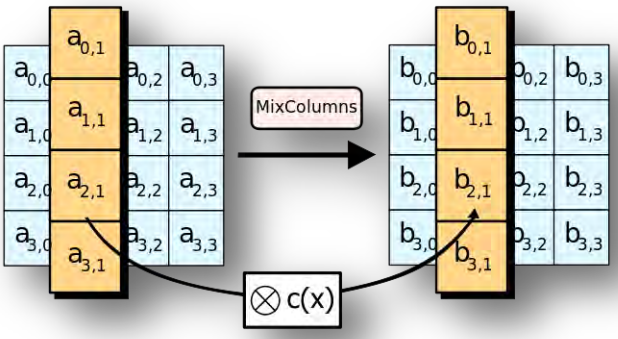
Για τα μπλοκ μεγέθους 128 bits (και 192 bits για τον Rijndael), η διαδικασία μετατόπισης είναι η ίδια. Η γραμμή *n* μετατοπίζεται προς τα αριστερά, κυκλικά, κατά *n-1* bytes. Κατά αυτόν τον τρόπο, κάθε στήλη του πίνακα κατάστασης που θα προκύψει στην έξοδο του βήματος ShiftRows θα αποτελείται από διαφορετικά bytes σε σχέση με την αρχική στήλη του πίνακα κατάστασης που έχει δοθεί ως είσοδος.

(Παραλλαγές του κρυπτοσυστήματος Rijndael με μεγαλύτερα μεγέθη μπλοκ έχουν ελαφρώς διαφορετικές μετατοπίσεις).

Τέλος, για μπλοκ των 256-bit, η πρώτη σειρά παραμένει αμετάβλητη και η μετατόπιση στην δεύτερη, τρίτη και τέταρτη σειρά είναι 1, 3 και 4 θέσεις αντίστοιχα - η αλλαγή αυτή ισχύει μόνο για το κρυπτοσύστημα Rijndael όταν χρησιμοποιείται με μπλοκ των 256-bit , αφού ο AES δεν χρησιμοποιεί μπλοκ 256-bit. Η σημασία αυτού του σταδίου έγκειται στην αποφυγή της γραμμικής ανεξαρτησίας των στηλών, στην οποία περίπτωση ο AES εκφυλίζεται σε τέσσερα ανεξάρτητα κρυπτοσυστήματα τμήματος.

3.5.6 Το βήμα «Ανάμειξης Στηλών» (MixColumns)

Στο βήμα MixColumns, τα 4 bytes της κάθε στήλης του πίνακα κατάστασης συνδυάζονται χρησιμοποιώντας έναν αντιστρέψιμο γραμμικό μετασχηματισμό για να εξάγουν μία νέα τιμή του ενός byte. Η συνάρτηση MixColumns δέχεται ως είσοδο 4 bytes και παράγει άλλα 4 bytes ως έξοδο, όπου κάθε byte εισόδου επηρεάζει όλα τα τέσσερα bytes εξόδου. Το βήμα MixColumns μαζί με το ShiftRows είναι αυτά τα οποία δημιουργούν την διάχυση (diffusion) στο κρυπτοσύστημα.



Εικόνα 3.5 - Στο στάδιο MixColumns, κάθε byte σε μία στήλη του Πίνακα Κατάστασης αντιστοιχίζεται με μία νέα τιμή, μέσω της συνάρτησης c(x) η οποία έχει ως είσοδο και τα 4 bytes της στήλης.

Κατά τη διάρκεια αυτής της διαδικασίας, κάθε στήλη του πίνακα κατάστασης πολλαπλασιάζεται με τον εξής σταθερό πίνακα:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

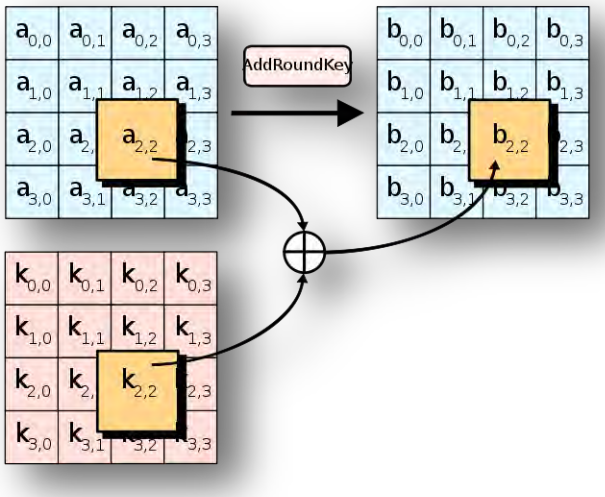
Ο αλγεβρικός πολλαπλασιασμός πινάκων αποτελείται από τον πολλαπλασιασμό και την πρόσθεση των καταχωρήσεών του. Στην περίπτωση αυτή όμως, η πράξη του πολλαπλασιασμού ορίζεται ως εξής: πολλαπλασιασμός με 1 σημαίνει ότι δεν υπάρχει αλλαγή, πολλαπλασιασμός με 2 σημαίνει μετατόπιση προς τα αριστερά και, τέλος, πολλαπλασιασμός με 3 σημαίνει μετατόπιση προς τα αριστερά και στη συνέχεια εκτέλεση πρόσθεσης XOR με την αρχική, μη-μετατοπισμένη τιμή.

Μετά την μετατόπιση, εάν η μετατοπισμένη τιμή είναι μεγαλύτερη από 0xFF, τότε πρέπει να εκτελεστεί μία υπό συνθήκη πρόσθεση XOR με 0x1B. Αυτές είναι ειδικές περιπτώσεις του συνήθους πολλαπλασιασμού σε πεπερασμένα σώματα $GF(2^8)$.

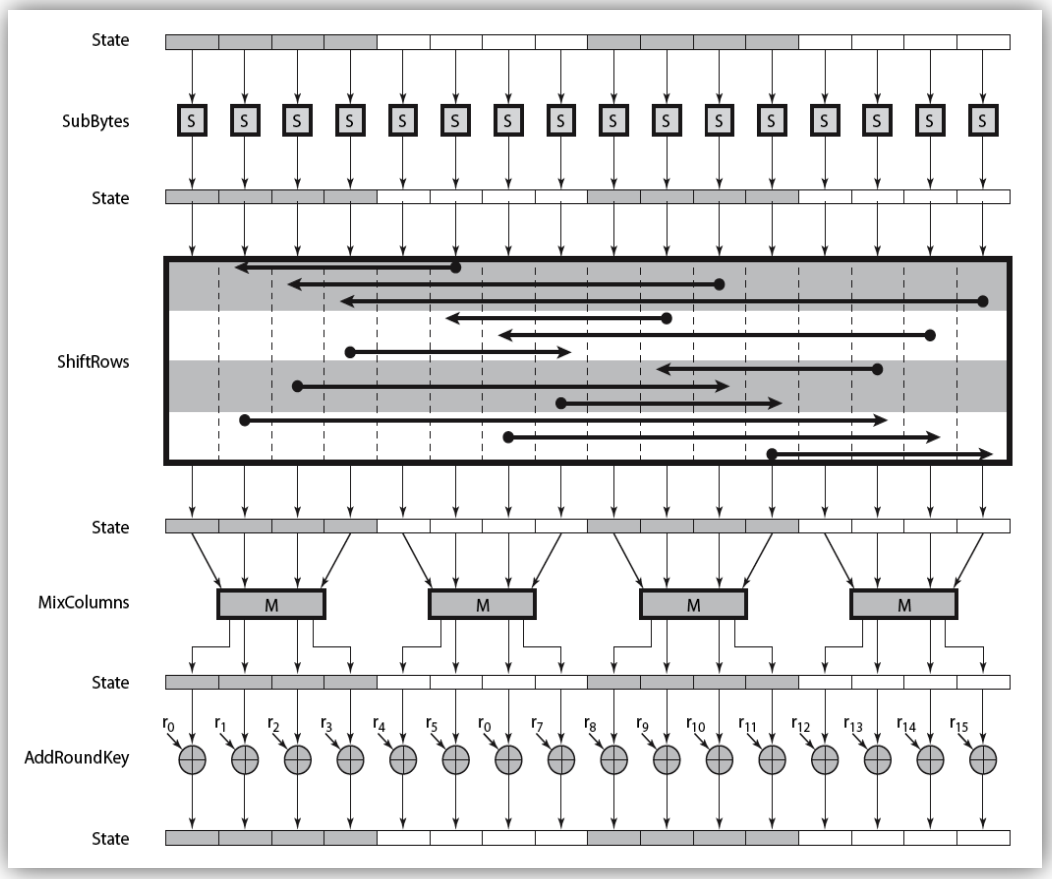
Γενικότερα, κάθε στήλη αντιμετωπίζεται ως ένα πολυώνυμο σε πεπερασμένο σώμα $GF(2^8)$ και στη συνέχεια πολλαπλασιάζεται modulo x^4+1 με ένα σταθερό πολυώνυμο $c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02$ (δηλαδή $[GF(2^8) \cdot c(x)] \ mod \ x^4+1$). Οι συντελεστές εμφανίζονται στο δεκαεξαδικό ισοδύναμο της δυαδικής αναπαράστασης των δυαδικών πολυωνύμων από το $GF(2)[x]$. Το βήμα MixColumns μπορεί επίσης να θεωρηθεί ως ένας πολλαπλασιασμός από τον εμφανιζόμενο ειδικό πίνακα MDS (Maximum Distance Separable) στο πεπερασμένο πεδίο $GF(2^8)$. Η διαδικασία αυτή περιγράφεται περαιτέρω στο τμήμα Mix Columns της τεκμηρίωσης του κρυπτοσυστήματος Rijndael.

3.5.7 Το βήμα «Πρόσθεσης Κλειδιού Γύρου» (AddRoundKey)

Στο βήμα AddRoundKey, το υπο-κλειδί (subkey) του υποπίνακα *w[από στήλη i ως j]* προστίθεται bit-προς-bit με τον πίνακα κατάστασης. Για κάθε γύρο, παράγεται ένα υπο-κλειδί από το αρχικό κλειδί κρυπτογράφησης, χρησιμοποιώντας τη διαδικασία επέκτασης κλειδιού του Rijndael (Rijndael key schedule) όπως αναλύθηκε στο κεφάλαιο 3.6.3. Κάθε υπο-κλειδί έχει το ίδιο μέγεθος με τον πίνακα κατάστασης. Το υπο-κλειδί προστίθεται συνδυάζοντας κάθε byte του πίνακα κατάστασης με το αντίστοιχο byte του υπο-κλειδιού, εκτελώντας δυαδική (bitwise) πράξη XOR.



Εικόνα 3.6 - Στο στάδιο AddRoundKey, κάθε byte του Πίνακα Κατάστασης προστίθεται με το αντίστοιχο byte του Υπο-Κλειδιού Γύρου μέσω δυαδικής πράξης XOR.



Εικόνα 3.7 - Ένα πιο αναλυτικό δομικό διάγραμμα του αλγορίθμου AES για την διαδικασία της κρυπτογράφησης.

3.5.8 Βελτιστοποίηση επιδόσεων στον αλγόριθμο του AES

Σε συστήματα με 32-bit ή μεγαλύτερες λέξεις μνήμης (words), είναι δυνατόν να επιταχυνθεί η εκτέλεση του κρυπτοσυστήματος AES συνδυάζοντας τα στάδια SubBytes και ShiftRows με το στάδιο MixColumns, μετασχηματίζοντάς τα σε μια ακολουθία από πίνακες αναζήτησης. Αυτό απαιτεί 4 πίνακες 256 καταχωρήσεων ο καθένας, με 32-bit (4 bytes) για την κάθε καταχώρηση, δεσμεύοντας συνολικά 4 kilobytes (4096 bytes) χώρου μνήμης - ένα kilobyte για κάθε πίνακα. Έτσι, σε αυτήν την περίπτωση ένας γύρος μπορεί να ολοκληρωθεί με 16 αναζητήσεις στον πίνακα και 12 XOR πράξεις των 32-bit, ακολουθούμενες από 4 XOR πράξεις των 32-bit στο στάδιο του AddRoundKey.

Εάν ο πίνακας του αποτελέσματος μεγέθους 4 kilobytes είναι πολύ μεγάλος για μια συγκεκριμένη πλατφόρμα, τότε η λειτουργία του πίνακα αναζήτησης μπορεί να διεξαχθεί με 1 μόνο πίνακα 256-καταχωρήσεων των 32-bit / καταχώρηση (δηλ. 1 kilobyte) με τη χρήση κυκλικών περιστροφών.

Επίσης, αν χρησιμοποιηθεί προσέγγιση προσανατολισμένη σε bytes (byte-oriented), τότε είναι επιπλέον δυνατό να συνδυαστούν τα στάδια SubBytes, ShiftRows, και MixColumns σε λειτουργία μονού γύρου (single round operation).

3.5.9 Ασφάλεια του κρυπτοσυστήματος AES

Μέχρι το Μάιο του 2009, οι μόνες δημοσιοποιημένες και επιτυχημένες επιθέσεις εναντίον του AES ήταν οι επιθέσεις πλάγιου καναλιού (side-channel attacks) για κάποιες συγκεκριμένες υλοποιήσεις. Η Εθνική Υπηρεσία Ασφάλειας των ΗΠΑ (NSA) επανεξέτασε όλες τις παραλλαγές του AES, συμπεριλαμβανομένης και της Rijndael παραλλαγής, και δήλωσε ότι όλες τους ήταν αρκετά ασφαλείς για χρήση σε αδιαβάθμητα δεδομένα της κυβέρνησής τους. Τον Ιούνιο του 2003, η κυβέρνηση των ΗΠΑ είχε ανακοινώσει ότι το AES θα μπορούσε να χρησιμοποιηθεί για την προστασία διαβαθμισμένων πληροφοριών. Συγκεκριμένα η δήλωση έλεγε:

«Η σχεδίαση και η δύναμη του αλγορίθμου AES για όλα τα μήκη κλειδιού (δηλαδή 128, 192 και 256 bit) είναι επαρκής για την προστασία των διαβαθμισμένων πληροφοριών μέχρι και το επίπεδο του «ΑΠΟΡΡΗΤΟΥ». Για τις «ΑΚΡΩΣ ΑΠΟΡΡΗΤΕΣ» πληροφορίες απαιτείται η χρήση κλειδιών μήκους 192 ή 256 bit. Οι υλοποιήσεις του AES σε προϊόντα τα οποία προορίζονται για την προστασία των εθνικών συστημάτων ασφαλείας ή/και πληροφοριών, πρέπει πρώτα να εξετασθούν και να πιστοποιηθούν από την Εθνική Υπηρεσία Ασφάλειας (NSA) και μετά να χρησιμοποιηθούν.» [12]

Ο αλγόριθμος AES περιλαμβάνει 10 γύρους μετασχηματισμού για κλειδιά των 128-bit, 12 γύρους για κλειδιά των 192-bit, και 14 γύρους για κλειδιά των

256-bit. Μέχρι το 2006, οι πιο γνωστές επιθέσεις συνέβησαν στους 7 γύρους για κλειδιά των 128-bit, στους 8 γύρους για κλειδιά των 192-bit, και στους 9 γύρους για κλειδιά των 256-bit. [13]

3.5.10 Γνωστές επιθέσεις κατά του κρυπτοσυστήματος AES

Για τους κρυπτογράφους, ως «σπάσιμο» ενός κρυπτοσυστήματος μπορεί να θεωρηθεί οτιδήποτε πιο γρήγορο από την χρήση επίθεσης Ωμής Βίας (brute-force), η οποία εκτελεί μία προσπάθεια αποκρυπτογράφησης για κάθε κλειδί. Η διαδικασία αυτή μπορεί να περιλαμβάνει αποτελέσματα τα οποία είναι ανέφικτα με την χρήση της τρέχουσας τεχνολογίας. Η μεγαλύτερη επιτυχημένη και δημοσίως γνωστή επίθεση Ωμής Βίας εναντίον οποιουδήποτε κρυπτο-συστήματος τμήματος (block cipher) ήταν κατά ενός κλειδιού RC5 των 64-bit και συνέβη από τον μη-κερδοσκοπικό οργανισμό distributed.net το 2006. [14]

Ο αλγόριθμος AES έχει μια αρκετά απλή αλγεβρική περιγραφή. **Το 2002**, μια θεωρητική επίθεση που ονομάστηκε **«επίθεση XSL» (eXtended Sparse Linearization attack)**, ανακοινώθηκε από τον Nicolas Courtois και τον Josef Pieprzyk και δημιουργήθηκε για να δείξει μια αδυναμία στον αλγόριθμο AES λόγω της απλής περιγραφής του. [16] Από τότε και έπειτα, διαφορετικές έρευνες (papers) δείξανε ότι η επίθεση έτσι όπως παρουσιάστηκε αρχικά είναι ανεφάρμοστη.

Κατά τη διάρκεια της διαδικασίας επιλογής αλγορίθμων για το κρυπτοσύστημα AES, οι προγραμματιστές των ανταγωνιστικών αλγορίθμων έγραψαν για τον Rijndael: "...ανησυχούμε για την χρήση του ... σε εφαρμογές που η ασφάλεια είναι κρίσιμης σημασίας». [17] Ωστόσο, τον Οκτώβριο του 2000, στο τέλος της διαδικασίας επιλογής του AES, ο Bruce Schneier (κρυπτογράφος και προγραμματιστής του ανταγωνιστικού αλγορίθμου Twofish) έγραψε ότι αν και επινόησε επιτυχημένες θεωρητικές επιθέσεις για τον αλγόριθμο Rijndael οι οποίες θα μπορούσαν να αναπτυχθούν και να εφαρμοστούν κάποια μέρα, δεν "πιστεύει ότι κάποιος θα ανακάλυπτε ποτέ μια επίθεση που θα του επέτρεπε να διαβάσει την κίνηση δεδομένων (traffic) του Rijndael". [18]

Την 1η Ιουλίου του 2009 ο Bruce Schneier έγραψε στο blog του [19] για μια **επίθεση που σχετίζεται με το κλειδί κρυπτογράφησης** τους AES, για τις εκδόσεις των 192-bit και 256-bit, την οποία κατασκεύασαν οι Alex Biryukov και Dmitry Khovratovich, [20], η οποία εκμεταλλεύεται την «κάπως απλή» διαδικασία επέκτασης κλειδιού του AES (AES's somewhat simple key schedule) και έχει πολυπλοκότητα της τάξης του 2^{119}. Το Δεκέμβριο του 2009, η πολυπλοκότητα αυτή βελτιώθηκε στο $2^{99.5}$. Η επίθεση αυτή, στην οποία αναφέρθηκε ο Schneier μέσω του blog του, αποτελεί συνέχεια μιας προηγούμενης επίθεσης που είχε ανακαλυφθεί το 2009 από τον Alex Biryukov, Dmitry Khovratovich και Ivica Nikolić με μια πολυπλοκότητα 2^{96} για ένα από τα 2^{35} κλειδιά. [21]

Μια άλλη επίθεση παρουσιάστηκε επίσης από τον Bruce Schneier [22] στο blog του **στις 30 Ιουλίου 2009** και κυκλοφόρησε ως προσχέδιο επιστημονικής έρευνας (preprint) **στις 3 Αυγούστου 2009**. [23] Η νέα επίθεση, σχεδιασμένη από τους Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, και Adi Shamir ήταν **κατά του AES-256** που χρησιμοποιεί μόνο 2 σχετικά κλειδιά και 2^{39} χρόνο **για να ανακτήσει το πλήρες κλειδί** των 256-bit σε έκδοση 9 γύρων μόνο, ή 2^{45} χρόνο για έκδοση 10 γύρων με ισχυρότερο τύπο επίθεση του σχετικού δευτερεύοντος κλειδιού, ή 2^{70} χρόνο για μια 11 γύρων έκδοση. **Ο πλήρης AES των 256-bit χρησιμοποιεί 14 γύρους, συνεπώς, οι παραπάνω επιθέσεις δεν αποδείχθηκαν αποτελεσματικές όσον αφορά την πλήρη έκδοση του αλγορίθμου**.

Τον Νοέμβριο του 2009, κυκλοφόρησε ως προσχέδιο επιστημονικής έρευνας (preprint) η πρώτη επίθεση **«διαχωρισμού δεδομένων γνωστού κλειδιού» (known-key distinguishing attack)** εναντίον της μειωμένης, 8 γύρων έκδοσης του AES-128. [24] Αυτή η επίθεση αποτελεί βελτίωση της **«στατιστικής επίθεσης σε συναρτήσεις κατακερματισμού» (rebound attack)** ή των **«start-from-the-middle attacks»** για μεταθέσεις τύπου AES, και η διαφορετικότητά της είναι ότι βλέπει δύο διαδοχικούς γύρους μετάθεσης ως την αξιοποίηση (application) του λεγόμενου Super-Sbox. Λειτουργεί στην έκδοση 8 γύρων του AES-128, με χρονική πολυπλοκότητα 2^{48} και με πολυπλοκότητα μνήμης 2^{32}.

Τον Ιούλιο του 2010 ο Vincent Rijmen δημοσίευσε μια ειρωνική έρευνα σχετικά με τις επιθέσεις **«chosen-key-relations-in-the-middle»** στον AES-128. [25]

Οι πρώτες επιθέσεις **«ανάκτησης κλειδιού» (key-recovery attacks)** για τον πλήρη AES συνέβησαν από τους Andrey Bogdanov, Dmitry Khovratovich και Christian Rechberger και **δημοσιεύτηκαν το 2011**. [26] Μία τέτοιου είδους επίθεση αποτελεί επίθεση **«πλήρους διμερούς γράφου» (biclique attack)** και είναι ταχύτερη από ότι η Ωμή Βία (brute-force) κατά ένα συντελεστή περίπου 4 (τέσσερις φορές ταχύτερη). Απαιτεί $2^{126.1}$ πράξεις για να ανακτήσει ένα κλειδί AES-128. Για τους AES-192 και AES-256, οι πράξεις που απαιτούνται κυμαίνονται αντίστοιχα στις $2^{189.7}$ και $2^{254.4}$.

3.5.11 Επιθέσεις πλάγιου καναλιού (Side-Channel Attacks) κατά του AES

Οι **Επιθέσεις πλάγιου καναλιού (Side-Channel Attacks)** δεν είναι θεωρητικές και δεν επιτίθενται στα στάδια της δομής των κρυπτοσυστημάτων (συνεπώς δεν σχετίζονται και με την ασφάλεια σε αυτόν τον τομέα). Αντιθέτως, οι επιθέσεις αυτές επιτίθενται στις υλοποιήσεις των κρυπτοσυστημάτων οι οποίες άθελά τους διαρρέουν δεδομένα. Υπάρχουν αρκετές γνωστές επιθέσεις πλάγιου καναλιού σε ορισμένες υλοποιήσεις του AES.

Τον Απρίλιο του 2005, ο D. J. Bernstein (κρυπτολόγος - μαθηματικός και καθηγητής του Eindhoven University of Technology) ανακοίνωσε την δικιά του **επίθεση «χρονισμού κρυφής μνήμης» (cache-timing attack)**. Την επίθεση αυτή χρησιμοποίησε για να «σπάσει» έναν προσαρμοσμένο εξυπηρετητή (server) ο οποίος χρησιμοποιούσε κρυπτογράφηση OpenSSL AES. [27] Η επίθεση απαιτούσε πάνω από 200 εκατομμύρια επιλεγμένα απλά κείμενα (plaintexts). [28] Ο προσαρμοσμένος εξυπηρετητής (server) σχεδιάστηκε για να εξάγει όσο το δυνατών περισσότερες πληροφορίες χρονισμού γινόταν (ο εξυπηρετητής επέστρεφε δηλαδή τον αριθμό των κύκλων μηχανής που χρειάστηκαν για την λειτουργία της κρυπτογράφησης). Ωστόσο, όπως ο Bernstein τόνισε, «ακόμα και με μείωση της ακρίβειας των χρονοσφραγίδων του εξυπηρετητή ή την εξάλειψή τους εντελώς από τις αποκρίσεις του, δεν συνεπάγεται αυτόματα και τερματισμός της επίθεσης. Το πρόγραμμα-πελάτης σε μια τέτοια περίπτωση μπορεί να χρησιμοποιήσει χρονισμούς μετ' επιστροφής (round-trip timings) βασιζόμενο στο τοπικό του ρολόι και αντισταθμίσει την αύξηση του στατιστικού θορύβου βγάζοντας τον μέσο όρο από έναν μεγαλύτερο αριθμό δειγμάτων». [27]

Τον Οκτώβριο του 2005, οι Dag Arne Osvik, Adi Shamir και Eran Tromer παρουσίασαν μία έρευνα στην οποία αναδείκνυαν διάφορες **επιθέσεις «χρονισμού κρυφής μνήμης» (cache-timing attack)** εναντίον του AES. [29] Μια επίθεση σύμφωνα με την έρευνα ήταν ικανή για να αποκομίσει ολόκληρο το κλειδί κρυπτογράφησης του AES, έπειτα από μόλις 800 πράξεις «πρόκλησης κρυπτογραφήσεων» (800 operations triggering encryptions), σε σύνολο 65 χιλιοστών του δευτερολέπτου. Η επίθεση αυτή απαιτούσε από τον επιτιθέμενο να είναι σε θέση να τρέξει προγράμματα στο ίδιο το σύστημα όπου εκτελούσε AES.

Τον Δεκέμβριο του 2009 μια επίθεση για κάποιες υλοποιήσεις του AES σε υλικό δημοσιοποίησε ότι χρησιμοποίησε την τεχνική της **«διαφορικής ανάλυσης σφάλματος» (differential fault analysis)** και μέσω της τεχνικής αυτής επέτρεπε την ανάκτηση ενός κλειδιού με πολυπλοκότητα 2^{32}. [30]

Τον Νοέμβριο του 2010 οι Endre Bangerter, David Gullasch και Stephan Krenn δημοσίευσαν μία έρευνα στην οποία περιέγραψαν μια πρακτική προσέγγιση σε «σχεδόν πραγματικό χρόνο» για την ανάκτηση μυστικών κλειδιών από τον AES-128, **χωρίς την ανάγκη ύπαρξης είτε κρυπτοκειμένου (ciphertext) είτε απλού κείμενο (plaintext)**. Η προσέγγιση λειτουργεί επίσης

και σε υλοποιήσεις του AES-128 οι οποίες χρησιμοποιούν πίνακες συμπίεσης, όπως το OpenSSL. [31] Όπως και σε προηγούμενες επιθέσεις, έτσι και σε αυτήν την επίθεση, η εκτέλεση κώδικα στο σύστημα το οποίο διενεργεί κρυπτογράφηση AES δεν απαιτεί δικαιώματα διαχειριστή συστήματος (root), με συνέπεια να μπορεί να επιτευχθεί πολύ εύκολα η μόλυνσή του από κακόβουλο λογισμικό και η εκτέλεση της επίθεσης μέσα από αυτό. [32]

ΚΕΦΑΛΑΙΟ 4

Σχεδίαση και Υλοποίηση του Διαχειριστή Κλειδαρίθμων «GM Password Manager»

4.1 Επιλογή της κατάλληλης γλώσσας προγραμματισμού

Κατά τη σχεδίαση ενός Διαχειριστή Κλειδαρίθμων, ο προγραμματιστής πρέπει να δώσει ιδιαίτερη βαρύτητα στα θέματα ασφαλείας που τον αφορούν (αναλύθηκαν εκτενώς στα προηγούμενα κεφάλαια) έτσι ώστε να μπορέσει να τον θωρακίσει όσο το δυνατόν περισσότερο γίνεται. Στα θέματα αυτά περιλαμβάνονται διαφόρων ειδών επιθέσεις και κακόβουλα προγράμματα τα οποία έχουν ως σκοπό την αποκάλυψη των διαπιστευτηρίων πρόσβασης και άλλων προσωπικών δεδομένων του χρήστη και στη συνέχεια την υποκλοπή τους από τον επιτιθέμενο.

Για να γίνει όσο το δυνατόν ποιοτικότερη η θωράκιση τέτοιων προγραμμάτων πρέπει η γλώσσα προγραμματισμού να παρέχει στον προγραμματιστή όσο το δυνατόν περισσότερες επιλογές ασφαλείας γίνεται. Η JAVA είναι μία τέτοια γλώσσα, η οποία λόγω του τεράστιου API (Application Programming Interface) που διαθέτει, δίνει στον προγραμματιστή ένα μεγάλο πλήθος μεθόδων διασφάλισης για τη δημιουργία προγραμμάτων που ως στόχο έχουν την ασφάλεια και την απόκρυψη (από τρίτους) των δεδομένων που επεξεργάζονται. Διαθέτει υλοποιήσεις πολλών κρυπτογραφικών αλγορίθμων οι οποίες έχουν σχεδιαστεί με στόχο το μέγιστο επίπεδο ασφάλειας, ταχύτητας εκτέλεσης και είναι ελεγμένοι και εγγυημένοι για την ορθότητά τους από διάφορους χρήστες του Διαδικτύου.

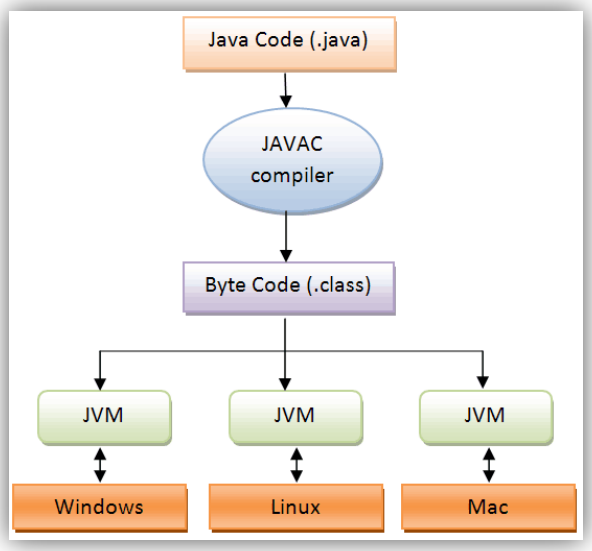
Επίσης, ένα ακόμα σημαντικό θέμα αυτού του είδους των προγραμμάτων είναι οι επιλογές φορητότητας αυτά που υποστηρίζουν. Σχεδιάζοντας έναν Διαχειριστή Κλειδαρίθμων σε γλώσσα JAVA, η οποία είναι μία πολλαπλής πλατφόρμας (cross-platform) γλώσσα προγραμματισμού, συνεπάγεται αυτόματα ότι το ίδιο το πρόγραμμα (το ίδιο αρχείο του προγράμματος) μπορεί να τρέξει σε διάφορα λειτουργικά συστήματα όπως τα Windows, Linux, Android, Macintosh κ.α.. Επιπλέον, ένα πρόγραμμα σε JAVA μπορεί να επεκταθεί ή να συνεργαστεί εύκολα με εφαρμογές του παγκόσμιου ιστού (Web Applications) αλλά και με επεκτάσεις προγραμμάτων περιήγησης (browser plugins - extensions).

Στο επόμενο κεφάλαιο περιγράφεται αναλυτικά η γλώσσα προγραμματισμού JAVA καθώς και το μοντέλο ασφαλείας που αυτή διαθέτει μαζί με τα πλεονεκτήματα και τα μειονεκτήματά του.

4.2 Η γλώσσα προγραμματισμού JAVA

4.2.1 Εισαγωγικά στοιχεία

Η γλώσσα προγραμματισμού Java - σε αντίθεση με την Javascript - είναι μία πλήρης γλώσσα προγραμματισμού η οποία μπορεί να χρησιμοποιηθεί τόσο για τη **συγγραφή εφαρμογών (applications)** (προγραμμάτων δηλαδή που εκτελούνται σε διάφορα Λειτουργικά Συστήματα) όσο και για τη **συγγραφή μικρο-εφαρμογών (applets)** - προγραμμάτων περιορισμένων δυνατοτήτων τα οποία εκτελούνται απευθείας από τα προγράμματα περιήγησης (browsers) και αποτελούν μέρος της λειτουργικότητας κάποιων ιστοσελίδων. Τα προγράμματα Java (οι εφαρμογές) συγγράφονται ως πηγαίος κώδικας και κατόπιν μεταγλωττίζονται για να παραχθεί μία ενδιάμεση αναπαράστασή τους η οποία ονομάζεται «bytecode». Για την εκτέλεση του bytecode απαιτείται ένα «περιβάλλον εκτέλεσης» Java (Java Runtime Environment - JRE). Το JRE περιλαμβάνει την εικονική μηχανή Java (Java Virtual Machine – JVM) συν τις βασικές κλάσεις της γλώσσας. Η όλη διαδικασία ανάπτυξης-εκτέλεσης εφαρμογών Java αναπαρίσταται στο σχήμα που ακολουθεί.



Εικόνα 4.1 - Τα στάδια που περνάει ένα πρόγραμμα Java από την στιγμή συγγραφής του πηγαίου κώδικα (.java αρχεία) μέχρι τη στιγμή της εκτέλεσής του σε ένα Λειτουργικό Σύστημα.

Η Java μπορεί επίσης να χρησιμοποιηθεί και σε διαδικτυακό περιβάλλον για συγγραφή προγραμμάτων τύπου «servlets», τα οποία εκτελούνται από εξυπηρετητές του παγκόσμιου ιστού (web servers) για τη διαμόρφωση των ιστοσελίδων και την αποστολή τους στα προγράμματα περιήγησης.

4.2.2 Πιθανά προβλήματα από τη εκτέλεση προγραμμάτων σε ένα πληροφοριακό σύστημα

Εκτελώντας ένα πρόγραμμα σε ένα πληροφοριακό σύστημα υπάρχει μία σειρά από αρνητικά ενδεχόμενα σχετικά με την ασφάλεια τα οποία μπορεί να κάνουν την εμφάνισή τους. Κάθε ένα από αυτά τα ενδεχόμενα μπορεί να έχει ως συνέπεια τη διαρροή πληροφοριών, την μείωση της διαθεσιμότητας των πόρων του συστήματος (μέσω επιθέσεων «Άρνησης Παροχής Υπηρεσιών» (Denial of Service attacks)), την απώλεια, καταστροφή ή αλλοίωσή της ακεραιότητας των δεδομένων που υπάρχουν στο σύστημα, ή απλά, την παρενόχληση του χρήστη χρησιμοποιώντας κακοβούλως έναν ή περισσότερους από τους πόρους του συστήματος.

Για να αντιμετωπίσει τα αρνητικά αυτά ενδεχόμενα η Java, ενσωματώνει διάφορους μηχανισμούς που μειώνουν τις πιθανότητες να χρησιμοποιηθεί κατά μη πρέποντα τρόπο κάποιος πόρος του συστήματος. Οι μηχανισμοί αυτοί παρέχουν υψηλά επίπεδα προστασίας απέναντι στη διαρροή πληροφοριών και την απώλεια της ακεραιότητάς τους, αν και οι άμυνες απέναντι στην απώλεια της διαθεσιμότητας των πόρων και στην ενόχληση του χρήστη είναι ασθενέστερες. Η ανισοκατανομή των μηχανισμών άμυνας είναι σχεδιαστική επιλογή, καθώς τα ενδεχόμενα της διαρροής και της αλλοίωσης ή της καταστροφής των δεδομένων είναι σαφώς δυσμενέστερα σενάρια από την απώλεια διαθεσιμότητας των πόρων ή την ενόχληση του χρήστη.

4.2.3 Μοντέλο ασφάλειας της Java

Δεδομένου ότι η Java είχε σχεδιαστεί εξ αρχής για διαδικτυακή χρήση, ενσωματώνει ένα τυπικό μοντέλο ασφαλείας, το οποίο ξεκίνησε με μία αρχική μορφή και στη συνέχεια εμπλουτίστηκε με επιπρόσθετα στοιχεία. Το πρωταρχικό ζήτημα σε κάθε περίπτωση είναι το αν εμπιστευόμαστε ή όχι τον κώδικα που πρόκειται να εκτελεστεί. Κάτι που ουσιαστικά ανάγεται στο αν (α) εμπιστευόμαστε τον συγγραφέα του κώδικα και (β) αν εμπιστευόμαστε τη διαδρομή διαμέσου της οποίας ο κώδικας έφθασε στον υπολογιστή μας. Ο κώδικας που εμπιστευόμαστε εκτελείται έχοντας πλήρη πρόσβαση στο σύστημα μέσω της εικονικής μηχανής Java (JVM), ενώ ο κώδικας που δεν εμπιστευόμαστε εκτελείται εντός ενός χώρου αυξημένης ασφάλειας που ονομάζεται **sandbox**. Το sandbox στη γενική περίπτωση απαγορεύει τις κάτωθι λειτουργίες:

- Απαγόρευση Δημιουργίας, Ανάγνωσης, Διαγραφής, Μετονομασίας, Ελέγχου Ύπαρξης και Αναφοράς ιδιοτήτων αρχείων.
- Απαγόρευση Δημιουργίας καταλόγων ή Αναφοράς σε περιεχόμενα καταλόγων.

- Απαγόρευση Σύνδεσης προς διαφορετικό υπολογιστή από τον εξυπηρετητή προέλευσης (originator), όπως και απαγόρευση Δημιουργίας θυρών για την αντιστοίχηση νέων συνδέσεων.

- Απαγόρευση Δημιουργίας παραθύρου πρώτου επιπέδου χωρίς προειδοποίηση ότι πρόκειται για ανασφαλή εφαρμογή.

- Απαγόρευση Συλλογής πληροφοριών χρήστη (όνομα, προσωπικός κατάλογος).

- Απαγόρευση Ορισμού ιδιοτήτων του συστήματος.

- Απαγόρευση Εκτέλεσης προγραμμάτων.

- Απαγόρευση Τερματισμού της εκτελούμενης Εικονικής Μηχανής (JVM).

- Απαγόρευση Φόρτωσης δυναμικών βιβλιοθηκών (DLLs).

- Απαγόρευση Δημιουργίας νέων νημάτων (threads) και πρόσβασης σε νήματα ελέγχου εκτός των δικών της.

- Απαγόρευση Δημιουργίας περιβάλλοντος φόρτωσης κλάσεων ή Διαχείρισης Ασφαλείας

- Απαγόρευση Δημιουργίας διεργασιών ελέγχου δικτύου π.χ. URLStreamHandlerFactory

- Απαγόρευση Ορισμού κλάσεων που ενσωματώνονται στις κλάσεις του υπολογιστή

Αυτή η αρκετά μακροσκελής λίστα απαγορεύσεων επιτρέπει ουσιαστικά τα προγράμματα που δεν θεωρούνται έμπιστα να εκτελούνται χρησιμοποιώντας μόνο την κεντρική μονάδα επεξεργασίας και τη μνήμη. Το μοντέλο αυτό είναι ιδιαίτερα περιοριστικό, καθώς δεν επιτρέπει τη χρήση πολλών διαδεδομένων προγραμματιστικών πρακτικών, όπως π.χ. τη δημιουργία προσωρινών αρχείων. Μια πιο λεπτομερής απόδοση αρμοδιοτήτων είναι εφικτή στη δεύτερη έκδοση της Java, όπου ψηφιακά υπογεγραμμένες εφαρμογές μπορούν να ζητήσουν από τον χρήστη να τους εκχωρήσει προνόμια που αρχικά δεν τους ήταν διαθέσιμα. Σε κάθε περίπτωση ωστόσο είναι εμφανές ότι όσο αυξάνεται το σύνολο προνομίων που έχει στη διάθεσή του ένα πρόγραμμα, τόσο πιο εύκολο του είναι να προβεί σε κάποια ενέργεια που αντίκειται στους κανόνες ασφάλειας.

4.2.4 Περιορισμοί πηγαίου κώδικα με στόχο την ασφάλεια

Όταν ορίζεται μία κλάση στην Java είναι δυνατόν για τον προγραμματιστή να χαρακτηρίσει τις μεταβλητές ή τις μεθόδους της κλάσης ως ιδιωτικές, προστατευμένες ή δημόσιες. Μέσω της χρήσης χαρακτηρισμών είναι δυνατόν στον προγραμματιστή να περιορίσει την πρόσβαση σε μεταβλητές και συναρτήσεις έτσι όπως αυτός επιθυμεί.

Επίσης, όπως είναι γνωστό, ο αντικειμενοστραφής προγραμματισμός δίνει τη δυνατότητα δημιουργίας υποκλάσεων και επανορισμού μεθόδων. Αν και τα χαρακτηριστικά αυτά προσθέτουν ευελιξία στον προγραμματιστή, μπορούν να εκμεταλλευτούν από κακόβουλους χρήστες και να χρησιμοποιηθούν σε μία σειρά από επιθέσεις. Η Java, για να περιορίσει τους κινδύνους που προκύπτουν από τα χαρακτηριστικά αυτά, δίνει τη δυνατότητα στους προγραμματιστές να χαρακτηρίσουν τις κλάσεις ή τις μεθόδους τους ως «τελικές» (final), αφαιρώντας έτσι τη δυνατότητα ορισμού υποκλάσεων ή επανορισμού, αντιστοίχως.

4.2.5 Άλλοι μηχανισμοί ασφάλειας

Πέρα από τους δύο παραπάνω περιορισμούς για την παροχή ασφάλειας, έχουν ενσωματωθεί επιπλέον στη Java και οι επόμενες ασφαλιστικές δικλείδες:

- Τα όρια των πινάκων ελέγχονται σε κάθε πρόσβαση, εξασφαλίζοντας έτσι ότι μέσω ενός πίνακα προσπελαύνονται μόνο τα στοιχεία που ανήκουν σ' αυτόν και όχι αυθαίρετα δεδομένα της μνήμης RAM.

- Η μετατροπή τύπων (typecasting) είναι ιδιαίτερα περιορισμένη. Η ορθή αντίληψη του περιβάλλοντος της Java για τους τύπους δεδομένων είναι ιδιαίτερα σημαντική, και εξηγείται πιο αναλυτικά στη συνέχεια.

- Οι μεταβλητές δεν μπορούν να χρησιμοποιηθούν πριν αρχικοποιηθούν, προκειμένου να μην είναι δυνατή η επόπτευση των δεδομένων που έχουν μείνει στη στοίβα από διαδικασίες που κλήθηκαν σε προγενέστερα χρονικά σημεία.

- Η αυτόματη συλλογή απορριμμάτων ελευθερώνει τη μνήμη που δεν χρειάζεται. Η ύπαρξη μηχανισμού για αυτόματη συλλογή απορριμμάτων αφ' ενός προστατεύει από μία σειρά συνηθισμένων προγραμματιστικών λαθών που σχετίζονται με τη δέσμευση και απελευθέρωση μνήμης, αφ' ετέρου δε καταργεί συνολικά τους δείκτες που είναι ορατοί στον προγραμματιστή, η πρόσβαση διαμέσου των οποίων δεν είναι δυνατόν να ελεγχθεί.

4.3 Σχεδίαση του GM Password Manager

4.3.1 Εισαγωγή

Ο **GM Password Manager** είναι ένα πρόγραμμα **Διαχειριστή Κλειδαρίθμων** το οποίο σχεδιάστηκε για να αποτελέσει το πρακτικό κομμάτι αυτής της διατριβής. Στον GM Password Manager έγινε προσπάθεια υλοποίησης των περισσοτέρων θεμάτων ασφαλείας των Διαχειριστών Κλειδαρίθμων που παρουσιάστηκαν και αναλύθηκαν στα παραπάνω κεφάλαια. Ο Διαχειριστής αυτός είναι υλοποιημένος σε γλώσσα προγραμματισμού Java ούτως ώστε να μπορεί να εκμεταλλευτεί πλήρως το μοντέλο ανάπτυξης και ασφάλειας αυτής, όπως περιγράφηκε στο κεφάλαιο 4.2.

Το πρόγραμμα έχει σχεδιασθεί για εκτέλεση σε περιβάλλον γραφικών και χρησιμοποιεί «παράθυρα» για την διεπαφή του με τον χρήστη. Λόγω περιορισμού χρόνου δεν δόθηκε ιδιαίτερη έμφαση στο καλλιτεχνικό κομμάτι και στην εργονομία του, αλλά ο περισσότερος χρόνος καταναλώθηκε για τον σχεδιασμό και την υλοποίηση ελέγχων και λειτουργιών ασφαλείας.

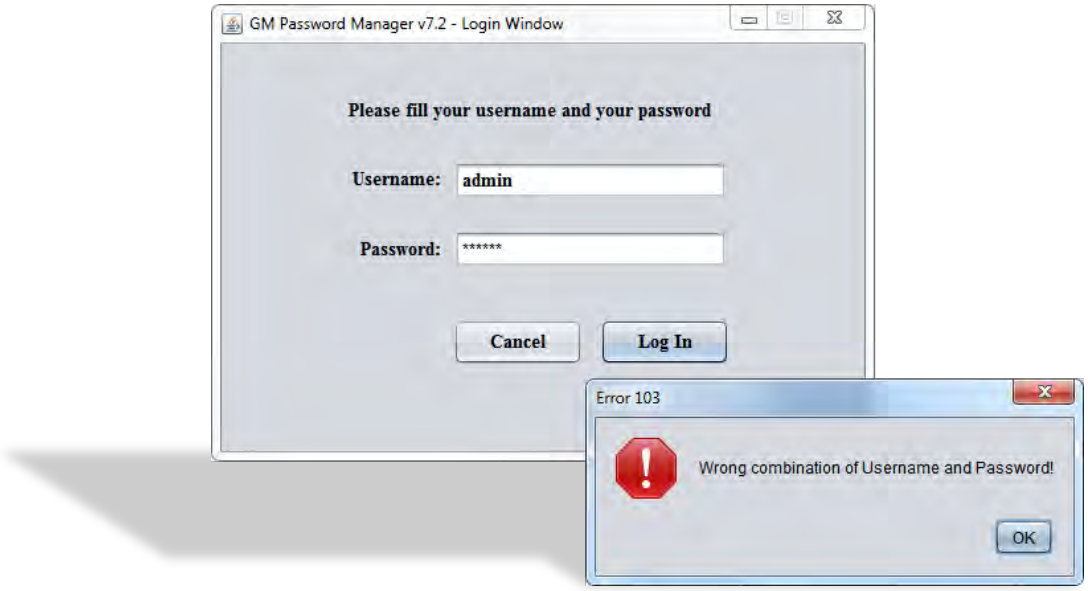
4.3.2 Παρουσίαση και περιγραφή λειτουργίας του προγράμματος

Αρχικά, για την ενεργοποίηση του προγράμματος, ο χρήστης καλείται να εκτελέσει το αρχείο *«User_Login.java»*. Με το που το εκτελέσει, εμφανίζεται το παράθυρο εισόδου του GM Password Manager, στο οποίο ο χρήστης πρέπει να συμπληρώσει το όνομα χρήστη και τον κύριο κωδικό πρόσβασής του. Στην έκδοση αυτή (για λόγους εργονομίας κατά την παρουσίαση και την δοκιμή του), το όνομα χρήστη και ο κύριος κωδικός πρόσβασης υπάρχουν προεγκατεστημένα στο πρόγραμμα (χωρίς τη δυνατότητα αλλαγής τους) και αντιστοιχούν στις λέξεις Username: admin και Password: admin.

Ο χρήστης **είναι υποχρεωμένος** να περάσει από αυτό το στάδιο καθώς αυτό είναι που θα τον πιστοποιήσει και που θα τον εξουσιοδοτήσει στην ανάγνωση των αποθηκευμένων διαπιστευτηρίων του και θα αποκρυπτογραφήσει στη συνέχεια τη βάση δεδομένων του.

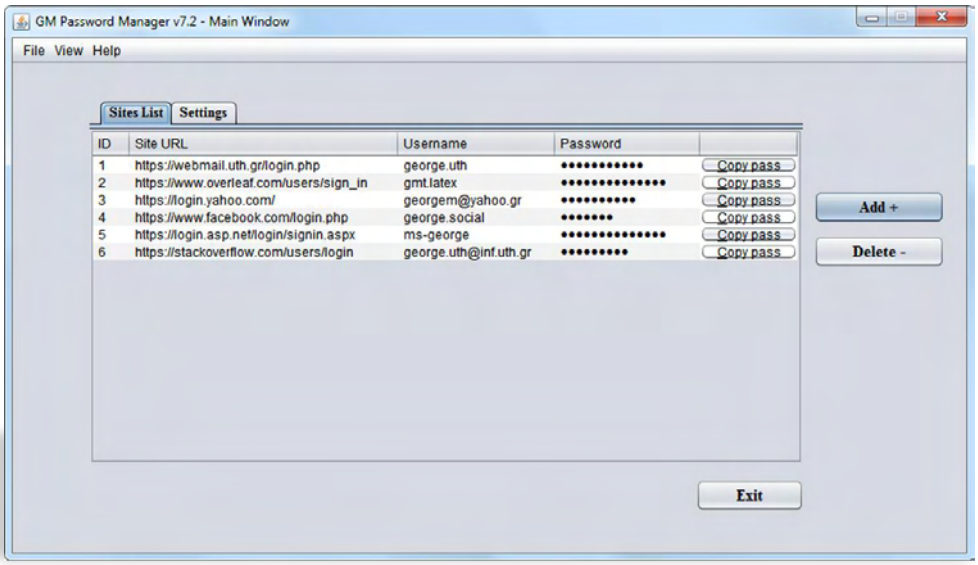
Εικόνα 4.2 - Το παράθυρο πιστοποίησης χρήστη του Διαχειριστή Κλειδαρίθμων «GM Password Manager» για το ξεκλείδωμα του προγράμματος και την εξουσιοδότηση εισόδου σε αυτό. Στην εικόνα αυτή απεικονίζεται η προσπάθεια εισαγωγής στο πρόγραμμα ενός χρήστη με όνομα «admin».

Σε περίπτωση εισαγωγής λανθασμένων δεδομένων πρόσβασης στο παράθυρο εισόδου του Διαχειριστή GM Password Manager εμφανίζεται στον χρήστη το κατάλληλο μήνυμα σφάλματος (Εικόνα 4.3). Το ίδιο συμβαίνει και αν ο χρήστης εισάγει απαγορευμένους (από τους κανονισμούς του προγράμματος) χαρακτήρες στα πλαίσια κειμένου ή αφήσει κάποιο πλαίσιο κενό.



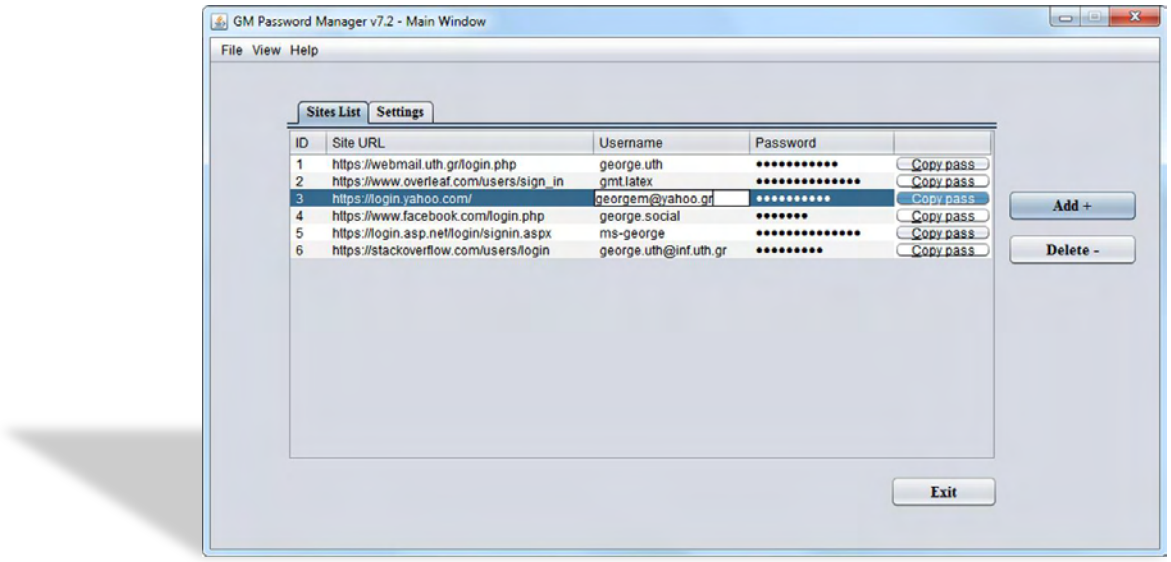
Εικόνα 4.3 - Το παράθυρο πιστοποίησης χρήστη του Διαχειριστή Κλειδαρίθμων «GM Password Manager» για το ξεκλείδωμα του προγράμματος και την εξουσιοδότηση εισόδου σε αυτό. Στην εικόνα αυτή απεικονίζεται η προσπάθεια εισαγωγής στο πρόγραμμα ενός χρήστη με όνομα «admin» με λανθασμένο κύριο κωδικό πρόσβασης ή με μη υπαρκτό όνομα χρήστη...

Μετά από μία επιτυχημένη πιστοποίηση χρήστη από το παράθυρο εισόδου, εμφανίζεται το κύριο παράθυρο του προγράμματος (Εικόνα 4.4). Μέσα από αυτό ο χρήστης μπορεί αρχικά να δει την λίστα με τα αποθηκευμένα δεδομένα διαπίστευσης της βάσης δεδομένων του (στην καρτέλα Sites List). Στη συνέχεια, υπάρχουν και κάποιες λειτουργίες που μπορεί να εκτελέσει, με βασικότερες αυτές της πρόσθεσης μίας νέας εγγραφής διαπιστευτηρίου στην βάση δεδομένων (κουμπί *«Add +»*) και της αντίστοιχης διαγραφής του από τη βάση δεδομένων (κουμπί *«Delete -»*).



Εικόνα 4.4 - Το κύριο παράθυρο του Διαχειριστή Κλειδαρίθμων «GM Password Manager» στο οποίο εμφανίζεται η λίστα με τα αποθηκευμένα διαπιστευτήρια ενός χρήστη από τη βάση δεδομένων του.

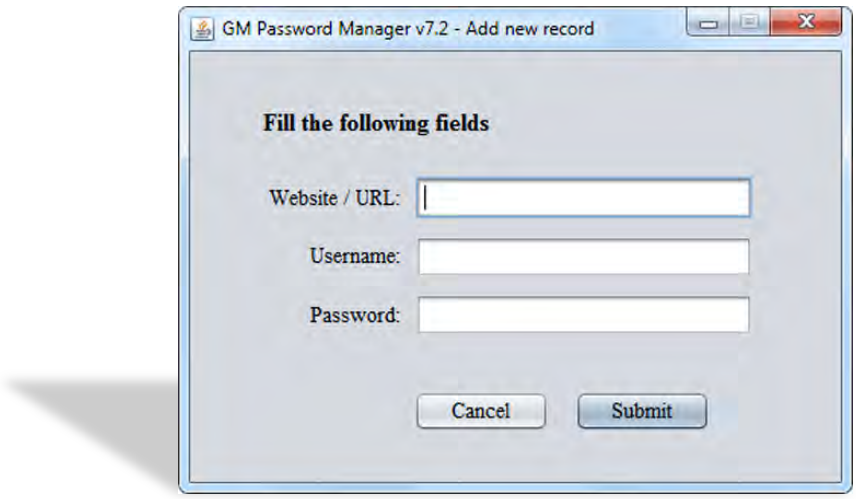
Στην λίστα με τα διαπιστευτήριά του, ο χρήστης, ανά πάσα στιγμή μπορεί να χρησιμοποιήσει τα περιεχόμενα από όποιο τμήμα από αυτά επιθυμεί (εκτός από το τμήμα του Κωδικού Πρόσβασης), κάνοντας απλά διπλό κλικ πάνω του. Με τη διαδικασία αυτή το τμήμα αυτό μετατρέπεται σε αναγνώσιμο πεδίο κειμένου και από εκεί ο χρήστης μπορεί να μαρκάρει και να αντιγράψει την πληροφορία που θέλει πατώντας από το πληκτρολόγιο τον συνδυασμό των πλήκτρων Ctrl + C (Εικόνα 4.5). Να σημειωθεί ότι όλα τα πεδία της λίστας διαπιστευτηρίων είναι «μόνο για ανάγνωση». Αυτό σημαίνει ότι αν συμβεί οποιαδήποτε αλλαγή των περιεχομένων τους από τον χρήστη (εκ προθέσεως ή καταλάθος), δε συνεπάγεται και την αποθήκευσή της στη βάση δεδομένων.



Εικόνα 4.5 - Το κύριο παράθυρο του Διαχειριστή Κλειδαρίθμων «GM Password Manager» στο οποίο έχει επιλεγεί το πεδίο Username της εγγραφής No. 3 για ανάγνωση (και μαρκάρισμα) των περιεχομένων του, κάνοντας διπλό κλικ πάνω σε αυτό.

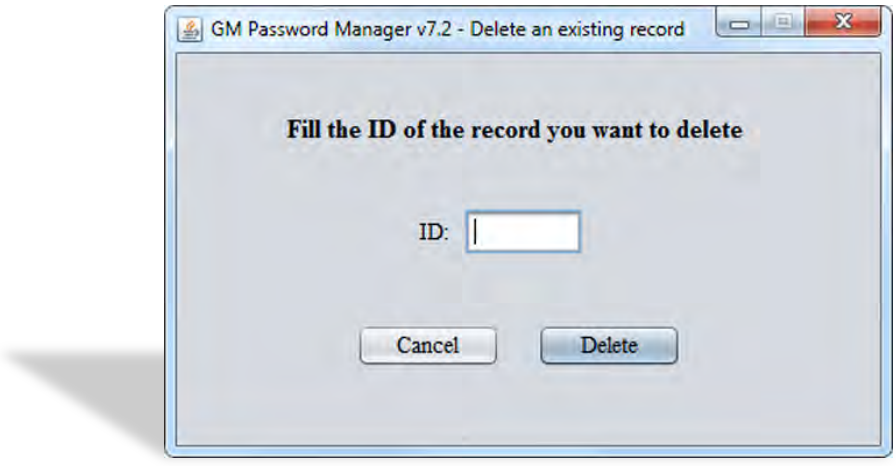
Το τμήμα που περιέχει τον Κωδικό Πρόσβασης του χρήστη δεν μπορεί να αντιγραφεί με την κλασσική μέθοδο Ctrl + C, αφού, όπως φαίνεται και στην Εικόνα 4.5, περιέχει μόνο κουκίδες. Ο λόγος που το τμήμα αυτό έχει αυτήν την ιδιομορφία, είναι για την προστασία των περιεχομένων του από τα αδιάκριτα βλέμματα τρίτων προσώπων (στο περιβάλλον εργασίας του χρήστη για παράδειγμα) ή για την προστασία από την παροχή στιγμιοτύπων (snapshots) από κακόβουλα προγράμματα παρακολούθησης (spyware / keyloggers). Συνεπώς, για να μπορέσει ο χρήστης να αντιγράψει τον Κωδικό Πρόσβασης που επιθυμεί και να τον χρησιμοποιήσει, πρέπει να πατήσει το κουμπί *«Copy pass»* από την γραμμή της εγγραφής του. Κατά το πάτημα αυτού του κουμπιού, το πρόγραμμα διαβάζει την αποκρυπτογραφημένη τιμή του Κωδικού Πρόσβασης της συγκεκριμένης εγγραφής από την μνήμη RAM και την μεταβιβάζει το ίδιο στο «πρόχειρο» (clipboard) του λειτουργικού συστήματος, στο σημείο δηλαδή της μνήμης που χρησιμοποιεί και η διαδικασία αντιγραφής Ctrl + C.

Συνεχίζοντας, όπως αναφέρθηκε και προηγουμένως, μία από τις επιλογές που έχει τη δυνατότητα να κάνει ο χρήστης από το κύριο παράθυρο του προγράμματος είναι αυτή της πρόσθεσης μίας νέας εγγραφής διαπιστευτηρίου στην βάση δεδομένων (κουμπί *«Add +»*). Όταν ο χρήστης πατήσει το κουμπί αυτό, εμφανίζεται η φόρμα εισαγωγής δεδομένων της Εικόνας 4.6. Στη φόρμα αυτή ο χρήστης καλείται να συμπληρώσει τα δεδομένα διαπίστευσης που επιθυμεί και πατώντας το κουμπί *«Submit»* να τα εισάγει στη βάση δεδομένων του ως μία νέα καταχώρηση.



Εικόνα 4.6 - Το παράθυρο εισαγωγής νέας εγγραφής στη βάση δεδομένων του Διαχειριστή Κλειδαρίθμων «GM Password Manager» για την καταχώρηση νέων διαπιστευτηρίων.

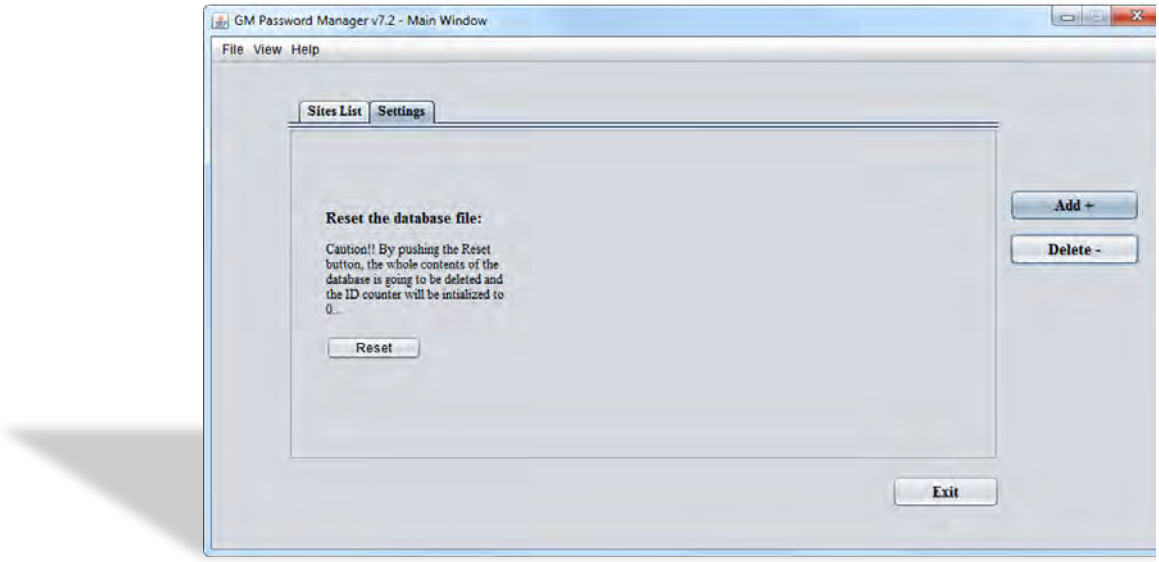
Αντιστοίχως, ο χρήστης μπορεί να διαγράψει μία υπάρχουσα εγγραφή διαπιστευτηρίου από τη βάση δεδομένων, πατώντας το κουμπί *«Delete -»*. Σε αυτήν την περίπτωση εμφανίζεται το παράθυρο της Εικόνας 4.7, το οποίο του ζητάει να συμπληρώσει το αναγνωριστικό (ID) της εγγραφής που θέλει να διαγράψει. Στη συνέχεια, αφού το συμπληρώσει και πατήσει το κουμπί «Delete» του παραθύρου, το πρόγραμμα προχωράει στην εύρεση και στην διαγραφή της.



Εικόνα 4.7 - Το παράθυρο διαγραφής μιας υπάρχουσας εγγραφής από τη βάση δεδομένων του Διαχειριστή Κλειδαρίθμων «GM Password Manager». Ο χρήστης καλείται να συμπληρώσει το αναγνωριστικό (ID) της εγγραφής που θέλει να διαγράψει, κάτι που μπορεί να εντοπίσει μέσα από τη λίστα διαπιστευτηρίων του κύριου παραθύρου του προγράμματος.

Να σημειωθεί επίσης στο σημείο αυτό ότι η λίστα των διαπιστευτηρίων του κύριου παραθύρου ενημερώνεται αυτόματα κάθε φορά που πραγματοποιείται προσθήκη ή διαγραφή προς και από την βάση δεδομένων.

Συνεχίζοντας, στην καρτέλα Settings (ρυθμίσεις) του προγράμματος (Εικόνα 4.8) περιλαμβάνεται η δυνατότητα επαναφοράς της βάσης δεδομένων στην αρχική της κατάσταση. Αυτό σημαίνει ότι πατώντας ο χρήστης το κουμπί *«Reset»* της καρτέλας αυτής, διαγράφονται αυτόματα όλες οι εγγραφές διαπιστευτηρίων του και επαναφέρονται οι αντίστοιχες μεταβλητές του προγράμματος στην αρχική τους θέση. Αυτή η δυνατότητα προστέθηκε κυρίως για λόγους ελέγχου και διόρθωσης σφαλμάτων (debugging) κατά την κατασκευή του προγράμματος, αλλά μπορεί να χρησιμεύσει και σε έναν χρήστη για την ολοκληρωτική διαγραφή της βάσης δεδομένων του με ένα μόνο κλικ, σε περίπτωση που αυτός διαθέτει πολλές εγγραφές καταχωρημένες. Προσοχή! Η ενέργεια αυτή είναι μόνιμη και (για λόγους ασφαλείας) δεν μπορεί να αναιρεθεί αν πραγματοποιηθεί.



Εικόνα 4.8 - Το κύριο παράθυρο του Διαχειριστή Κλειδαρίθμων «GM Password Manager» στο οποίο εμφανίζεται το κουμπί ολοκληρωτικής διαγραφής και επαναφοράς της βάσης δεδομένων του στην αρχική της κατάσταση.

Μία ακόμη πολύ σημαντική δυνατότητα που προσφέρει ο «GM Password Manager», είναι αυτή της φορητότητας της βάσης δεδομένων των διαπιστευτηρίων του, καθώς και της δυνατότητας δημιουργίας αντιγράφων ασφαλείας. Η Βάση Δεδομένων με τα διαπιστευτήρια του χρήστη βρίσκεται αποθηκευμένη σε συγκεκριμένα αρχεία, μέσα στον φάκελο του προγράμματος, όπως αναλύεται και στο επόμενο κεφάλαιο, 4.3.3.

Έτσι, ο χρήστης μπορεί να εντοπίσει τα αρχεία αυτά και ανά κάποιο χρονικό διάστημα που θα επιλέξει ο ίδιος μπορεί να κρατάει αντίγραφα των αρχείων αυτών σε κάποιον δικό του χώρο αποθήκευσης. Ο χώρος αυτός δεν πρέπει απαραίτητα να είναι ιδιωτικός ή να προστατεύεται από κάποιους μηχανισμούς ασφαλείας, αφού η βάση δεδομένων βρίσκεται πάντα σε κρυπτογραφημένη μορφή μέσα στα αρχεία αυτά, με χρήση του ισχυρού προτύπου κρυπτογράφησης AES στα 128 bits. Τα αρχεία αυτά προστατεύονται από το όνομα χρήστη και τον κύριο κωδικό πρόσβασης του χρήστη, μπορούν να αναγνωσθούν μόνο από το ίδιο το πρόγραμμα με χρήση του σωστού συνδυασμού των ανωτέρω διαπιστευτηρίων πρόσβασης, ενώ αποκρυπτογραφούνται μόνο τα περιεχόμενά τους στη μνήμη RAM του υπολογιστή και ποτέ τα ίδια ως ολότητα στον σκληρό δίσκο ή σε κάποια άλλη μονάδα αποθήκευσης.

Επίσης, ο χρήστης, με την ίδια λογική μπορεί να χρησιμοποιήσει τα αρχεία αυτά για λόγους φορητότητας, έχοντας δηλαδή τα διαπιστευτήριά του πάντα μαζί του (σε κάποια μονάδα αποθήκευσης USB για παράδειγμα) και τοποθετώντας τα στο σημείο που βρίσκεται εγκατεστημένο το πρόγραμμα για να τα αναγνώσει (στον υπολογιστή της εργασίας του, του σπιτιού του, σε κοινόχρηστο υπολογιστή σε κάποιο αεροδρόμιο, κ.τ.λ.).

Τέλος, ο χρήστης πατώντας από το μενού «Help» την επιλογή «About GM Password Manager» μπορεί να δει τις πληροφορίες έκδοσης του προγράμματος, της άδειας χρήσης καθώς και της πνευματικής ιδιοκτησίας, όπως φαίνεται και στην επόμενη εικόνα, Εικόνα 4.9.

Εικόνα 4.9 - Το κύριο παράθυρο εμφάνισης πληροφοριών έκδοσης, πνευματικής ιδιοκτησίας και άδειας χρήσης του Διαχειριστή Κλειδαρίθμων «GM Password Manager».

4.3.3 Δομικά στοιχεία του προγράμματος

Το πρόγραμμα αποτελείται συνολικά από 9 αρχεία τα οποία είναι όλα απαραίτητα για την λειτουργία του και πρέπει να βρίσκονται όλα τοποθετημένα στον ίδιο φάκελο. Επιγραμματικά, στα αρχεία αυτά περιλαμβάνονται:

- Το αρχείο ***«User_Login.java»*** το οποίο περιέχει την κλάση *«User_Login»*,
- το αρχείο ***«Main_Window.java»*** το οποίο περιέχει την κλάση *«Main_Window»*,
- το αρχείο ***«Add_Record.java»*** το οποίο περιέχει την κλάση *«Add_Record»*,
- το αρχείο ***«Delete_Record.java»*** το οποίο περιέχει την κλάση *«Delete_Record»,*
- το αρχείο ***«About_Window.java»*** το οποίο περιέχει την κλάση *«About_Window»,*
- το αρχείο ***«Credentials.gdm»***,
- το αρχείο ***«Data.gdm»***,
- το αρχείο ***«Keyc.crp»*** και
- το αρχείο ***«Keyd.crp»***.

Τα 5 πρώτα αρχεία της λίστας (αρχεία **.java*) περιέχουν τον πηγαίο κώδικα του προγράμματος. Επίσης, το καθένα από αυτά τα πέντε κατά την φόρτωσή του δημιουργεί και ένα πλαίσιο *«JFrame»*, δηλαδή ένα παράθυρο γραφικής διεπαφής χρήστη, ούτως ώστε ο χρήστης να μπορεί να χρησιμοποιήσει το πρόγραμμα μέσω γραφικών.

Το αρχείο *«User_Login.java»* είναι το πιο σημαντικό αρχείο του προγράμματος. Η κλάση *«User_Login»* η οποία περιέχεται σε αυτό, είναι υπεύθυνη για την πιστοποίηση του χρήστη, για τους απαραίτητους ελέγχους ασφαλείας και προστασίας του προγράμματος αλλά και για την σωστή αποκρυπτογράφηση των διαπιστευτηρίων της βάσης δεδομένων. Επίσης αποτελεί τη μόνη εκτελέσιμη κλάση του προγράμματος, κάτι που σημαίνει ότι περιλαμβάνει μία μέθοδο *main()* στο κυρίως σώμα του κώδικά της. Συνεπώς, το αρχείο *«User_Login.java»* είναι το μόνο αρχείο το οποίο είναι εκτελέσιμο και ικανό για την εκκίνηση του προγράμματος.

Τα αρχεία με κατάληξη *«*.gdm»* αποτελούν τα κρυπτογραφημένα αρχεία δεδομένων του GM Password Manager.

Αναλυτικά, στο αρχείο *«Credentials.gdm»* περιλαμβάνονται όλοι οι συνδυασμοί ονομάτων χρήστη και κύριων κωδικών πρόσβασης των χρηστών του προγράμματος. Το αρχείο αυτό χρησιμοποιείται από την κλάση *«User_Login»* για την ταυτοποίηση και την εξουσιοδότηση των χρηστών στο πρόγραμμα.

Αντίστοιχα, το αρχείο *«Data.gdm»* περιέχει την βάση δεδομένων διαπίστευσης του χρήστη. Περιέχει δηλαδή το σύνολο των διαπιστευτηρίων που έχει καταχωρήσει ο ίδιος μέσω του προγράμματος. Το αρχείο αυτό χρησιμοποιείται από την κλάση *«Main_Window»* για την ενημέρωση της λίστας διαπιστευτηρίων της καρτέλας *«Sites List»*, του κύριου παραθύρου του προγράμματος.

Συνεχίζοντας, τα αρχεία με κατάληξη *«*.crp»* αποτελούν τα (μη-κρυπτογραφημένα) αρχεία αποθήκευσης των Διανυσμάτων Αρχικοποίησης (Initialization Vectors) του αλγορίθμου κρυπτογράφησης AES του προγράμματος.

Το *«Keyc.crp»* κρατάει αποθηκευμένο το Διάνυσμα Αρχικοποίησης που χρησιμοποιήθηκε για την κρυπτογράφηση και την αποκρυπτογράφηση του αρχείου *«Credentials.gdm»*. Αντίστοιχα, το αρχείο *«Keyd.crp»* κρατάει αποθηκευμένο το Διάνυσμα Αρχικοποίησης που χρησιμοποιήθηκε για την κρυπτογράφηση και την αποκρυπτογράφηση του αρχείου *«Data.gdm»*.

Τα αρχεία αυτά είναι απαραίτητο να υπάρχουν στον φάκελο του προγράμματος, καθώς πάνω σε αυτά βασίζεται η αρχικοποίηση του αλγορίθμου κρυπτογράφησης AES. Φυσικά το περιεχόμενό τους δεν είναι ποτέ το ίδιο και ανανεώνεται κάθε φορά που διενεργείται κρυπτογράφηση στο αντίστοιχο *«*.gdm»* αρχείο από το πρόγραμμα

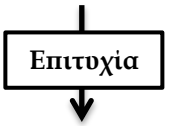
Αν τα αρχεία *«*.crp»* μπλεχτούν για κάποιο λόγο (εσκεμμένα ή κατά λάθος) με τα αντίστοιχα αρχεία άλλης συνεδρίας του προγράμματος (π.χ. με τα αρχεία *crp* από μία πιο παλιά εκτέλεση του προγράμματος), τότε ο αλγόριθμος κρυπτογράφησης θα αρχικοποιηθεί με άλλες τιμές πλην τις τρέχουσες, με συνέπεια να μην μπορέσει να αποδώσει την αποκρυπτογράφηση σωστά και να μη μπορέσει να συμβεί αντίστοιχα η ανάγνωση της βάσης δεδομένων των χρηστών και των διαπιστευτηρίων τους από το πρόγραμμα.

Τέλος, σε περίπτωση που ο χρήστης θελήσει να κρατήσει αντίγραφα ασφαλείας ή φορητότητας των δεδομένων του, πρέπει να σημειωθεί ότι κάθε φορά πρέπει να αντιγράφει (σε ξεχωριστό φάκελο) και τα 4 αρχεία που περιγράφηκαν προηγουμένως, δηλαδή τα: *«Credentials.gdm»*, *«Data.gdm»*, *«Keyc.crp»* και *«Keyd.crp»*.

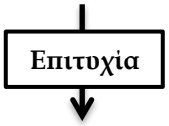
4.3.4 Η κλάση ελέγχου *«User_Login»* του αρχείου *«User_Login.java»*

Όταν τρέξει κάποιος το αρχείο *«User_Login.java»*, η κλάση *«User_Login»* εμφανίζει στην οθόνη το παράθυρο ξεκλειδώματος του προγράμματος (Εικόνα 4.2). Σε αυτό ο χρήστης καλείται να εισάγει το όνομα χρήστη του και τον κύριο κωδικό πρόσβασής του. Μόλις τελειώσει με την εισαγωγή τους και πατήσει το πλήκτρο *«Log In»*, τότε λαμβάνει μέρος ένα μεγάλο πλήθος από διαδικασίες ελέγχου πριν το ολοκληρωτικό ξεκλείδωμα του προγράμματος και την αποκρυπτογράφηση της βάσης δεδομένων του, με τη σειρά που εμφανίζονται παρακάτω. Να σημειωθεί ότι η μετάβαση από το ένα στάδιο ελέγχου στο άλλο συμβαίνει μόνο μετά από επιτυχή έλεγχο του τρέχοντος σταδίου. Σε περίπτωση σφάλματος, οι έλεγχοι των υπόλοιπων σταδίων ακυρώνονται και η διαδικασία ξεκινάει πάλι από την αρχή μόλις ο χρήστης πατήσει ξανά το κουμπί *«Log In»*.

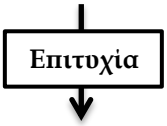
- **1) Ελέγχεται το πεδίο *«Username»*** σχετικά με το αν είναι συμπληρωμένο ή όχι και για το αν αυτό περιέχει απαγορευμένους χαρακτήρες. Σε περίπτωση προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε.



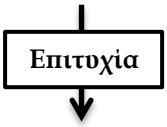
- **2) Ελέγχεται το πεδίο *«Password»*** σχετικά με το αν είναι συμπληρωμένο ή όχι και για το αν αυτό περιέχει απαγορευμένους χαρακτήρες. Σε περίπτωση προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε. Εδώ πρέπει επίσης να τονιστεί το γεγονός ότι το πλαίσιο εισαγωγής του κύριου κωδικού πρόσβασης *«Password»* δεν είναι υλοποιημένο προγραμματιστικά ως απλό πλαίσιο κειμένου τύπου *«JTextField»* αλλά έχει υλοποιηθεί ως πλαίσιο ασφαλούς εισαγωγής κωδικού πρόσβασης τύπου *«JPasswordField»*. Τα πλαίσια αυτού του τύπου έχουν διαφορετικού είδους μετάβαση των δεδομένων τους στη μνήμη αλλά και προγραμματιστικά χρειάζονται διαφορετικές, ασφαλείς μεθόδους για την ανάγνωση των περιεχομένων τους.



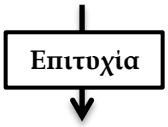
- **3) Επιτελείται έλεγχος σχετικά με το αν το αρχείο χρηστών *«Credentials.gdm»* υπάρχει στην προεπιλεγμένη θέση, αν είναι όντως αρχείο (και όχι κατάλογος) και αν είναι προσβάσιμο για ανάγνωση από το πρόγραμμα.** Σε περίπτωση προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε.



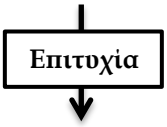
- **4) Αποκρυπτογραφείται στη μνήμη RAM το αρχείο χρηστών *«Credentials.gdm»*** με βάση τον συνδυασμό ονόματος χρήστη και κύριου κωδικού πρόσβασης που δόθηκε ως είσοδος. Για την αποκρυπτογράφηση και την κρυπτογράφηση του αρχείου αυτού, χρησιμοποιείται ο πλήρης αλγόριθμος **AES των 128 bits** (Κεφάλαιο 3.6).



- **5) Ελέγχεται το κατά πόσο ο αλγόριθμος κρυπτογράφησης AES μπόρεσε να ολοκληρώσει όλα τα στάδια της επεξεργασίας του με επιτυχία.** Σε περίπτωση λανθασμένου συνδυασμού ονόματος χρήστη και κύριου κωδικού πρόσβασης, ή, λανθασμένου Διανύσματος Αρχικοποίησης (Κεφάλαιο 4.3.3) ο αλγόριθμος AES δε θα μπορέσει να ολοκληρώσει την συνολική επεξεργασία της αποκρυπτογράφησής του και έτσι ο χρήστης θα ενημερωθεί με ένα παράθυρο σφάλματος το οποίο θα τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε.

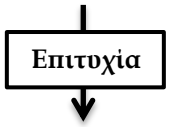


- **6) Ελέγχεται το κατά πόσο τα αποκρυπτογραφημένα περιεχόμενα στη μνήμη RAM του αρχείου χρηστών *«Credentials.gdm»* είναι αναγνώσιμα μετά την αποκρυπτογράφηση ή όχι.** Αν δεν μπορούν να αναγνωσθούν και να αναγνωρισθεί η συγκεκριμένη εγγραφή του χρήστη, συνεπάγεται ότι ο συνδυασμός ονόματος χρήστη και κύριου κωδικού πρόσβασης που χρησιμοποιήθηκε ως κλειδί ήταν λανθασμένος. Σε περίπτωση τέτοιου προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε.



- **7) Καταμετράται το πλήθος των εγγραφών του αποκρυπτο-γραφημένου αρχείου χρηστών *«Credentials.gdm»*** έτσι ώστε να διαπιστωθεί ότι αυτό δεν έχει υποστεί παράνομες αλλαγές (λόγω παρεμβολής τρίτων προσώπων) από την τελευταία φορά που είχε επεξεργασθεί νόμιμα από το πρόγραμμα. Αυτό συμβαίνει για να αποκλεισθεί το ενδεχόμενο αλλοίωσης του αρχείου *«Credentials.gdm»* από κάποιον κακόβουλο χρήστη μέσω κάποιας επίθεσης, στην προσπάθειά του να προσθέσει επιπλέον εγγραφή χρήστη σε αυτό με τα δικά του στοιχεία (όνομα χρήστη και κύριο κωδικό πρόσβασης) με σκοπό το ξεκλείδωμα του προγράμματος. Σε περίπτωση τέτοιου προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει

για το αντίστοιχο πρόβλημα που προέκυψε χρησιμοποιώντας μόνο έναν **κωδικό σφάλματος** ο οποίος είναι γνωστός μόνο στον προγραμματιστή.



- **8) Γίνεται έλεγχος στα αποκρυπτογραφημένα περιεχόμενα του αρχείου χρηστών *«Credentials.gdm»* στη μνήμη RAM σχετικά με την ορθότητα της δομή του** και σχετικά με την ύπαρξη απαγορευμένων χαρακτήρων σε αυτό ή μη εκτυπώσιμων χαρακτήρων ελέγχου. Αυτό συμβαίνει πάλι για να διαπιστωθεί ότι το αρχείο δεν έχει υποστεί κάποια προσπάθεια αλλοίωσης από τρίτους. Σε περίπτωση τέτοιου προβλήματος εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε χρησιμοποιώντας μόνο έναν **κωδικό σφάλματος** ο οποίος είναι γνωστός μόνο στον προγραμματιστή.

Αν το αρχείο χρηστών *«Credentials.gdm»* περάσει με επιτυχία τους παραπάνω ελέγχους γνησιότητας και βρεθεί καταγεγραμμένος σε αυτό κάποιος χρήστης με τα στοιχεία που δόθηκαν στο παράθυρο εισαγωγής, τότε η κλάση *«User_Login»* συνεχίζει εκτελώντας τα εξής τελευταία βήματα:

- **1) Κρυπτογραφεί ξανά το αρχείο χρηστών *«Credentials.gdm»*,** ασχέτως με το γεγονός ότι δεν υπήρξαν αλλαγές στα περιεχόμενά του, μόνο και μόνο για να ενημερωθούν τα κρυπτογραφημένα περιεχόμενά του με νέα, εντελώς διαφορετικά και να δυσκολέψει στο έπακρο την διαδικασία ξεκλειδώματός του από κάποιον κακόβουλο χρήστη ή να ακυρώσει / επανεκκινήσει κάποια πιθανή επίθεση Ωμής Βίας.

- **2) Ενημερώνει ξανά το αρχείο *«Keyc.crp»*,** ούτως ώστε αυτό να περιέχει τα νέα πλέον δεδομένα του Διανύσματος Αρχικοποίησης που μόλις δημιουργήθηκαν από τον αλγόριθμο AES (και χρησιμοποιήθηκαν στην κρυπτογράφηση του αρχείου *«Credentials.gdm»*).

- **3) Διαγράφει τα περιεχόμενα του αρχείου χρηστών *«Credentials.gdm»* που έχουν φορτωθεί στη μνήμη RAM**, καθώς και τις τιμές από όλες τις μεταβλητές που χρησιμοποιήθηκαν από τους ελέγχους, με τη μέθοδο εκχώρησης τυχαίων (αλλά επιτρεπτών) τιμών οι οποίες παράγονται μέσα από μία κρυπτογραφικά ασφαλή μέθοδο παραγωγής ψευδοτυχαίων αριθμών. Κατά αυτόν τον τρόπο δεν είναι δυνατό να μείνουν υπολείμματα διαπιστευτηρίων πρόσβασης στη μνήμη έτσι ώστε να υπάρξει κάποια πιθανή υποκλοπή τους αργότερα από κάποιον κακόβουλο χρήστη κατά την αποδέσμευση αυτού του χώρου μνήμης από το πρόγραμμα.

- **4) Πραγματοποιεί εκτέλεση της κλάσης *«Main_Window»*** για την εμφάνιση του κυρίως παραθύρου του προγράμματος και «περνάει» σε αυτή μέσω της μεθόδου δημιουργού της τον συνδυασμό ονόματος χρήστη και κύριου κωδικού πρόσβασης σε μη-κρυπτογραφημένη μορφή, ούτως ώστε να μπορέσει αυτή με τη σειρά της να αποκρυπτογραφήσει το αρχείο της βάσης δεδομένων διαπιστευτηρίων *«Data.gdm»*.

- **5) Πραγματοποιεί το κλείσιμο του δικού της παραθύρου (παραθύρου εισόδου)**, ελευθερώνει την μνήμη που δέσμευσε κατά τους ελέγχους και απενεργοποιείται.

4.3.5 Η κλάση *«Main_Window»* του κύριου παραθύρου του προγράμματος του αρχείου *«Main_Window.java»*

Μετά την επιτυχή εξουσιοδότηση πρόσβασης στο πρόγραμμα εκκινείται η κλάση *«Main_Window»*, η οποία εμφανίζει στην οθόνη το κύριο παράθυρο του προγράμματος (Εικόνα 4.4). Μέσα από το παράθυρο αυτό ο χρήστης έχει τις εξής δυνατότητες:

- **Να αναγνώσει** τα διαπιστευτήρια που έχει αποθηκεύσει στη βάση δεδομένων,
- **Να χρησιμοποιήσει** όποιο τμήμα των αποθηκευμένων διαπιστευτηρίων επιθυμεί (πλην αυτό του κωδικού πρόσβασης) αντιγράφοντάς το με χρήση του συνδυασμού πλήκτρων Ctrl + C,
- **Να αντιγράψει** (στο πρόχειρο) τον κωδικό πρόσβασης από την εγγραφή διαπιστευτηρίου που επιθυμεί πατώντας το αντίστοιχο κουμπί *«Copy pass»* της εγγραφής του,
- **Να προσθέσει** μία νέα εγγραφή διαπιστευτηρίου στη βάση δεδομένων,
- **Να διαγράψει** μία υπάρχουσα εγγραφή διαπιστευτηρίου από τη βάση δεδομένων,
- **Να διαγράψει ολοκληρωτικά** την βάση δεδομένων και να επαναφέρει τις αντίστοιχες μεταβλητές της στην αρχική τους τιμή (reset).

Γενικά η κλάση *«Main_Window»* περιλαμβάνει λειτουργίες που έχουν σχέση με την διαχείριση των διαπιστευτηρίων του χρήστη. Κατά την εκκίνηση αυτής της κλάσης, διενεργείται η αποκρυπτογράφηση της βάσης δεδομένων διαπιστευτηρίων του αντίστοιχου χρήστη που έκανε είσοδο στο πρόγραμμα, από το αρχείο *«Data.gdm»* στη μνήμη RAM, με βάση τον συνδυασμό ονόματος χρήστη και κύριου κωδικού πρόσβασης που παραλήφθηκε από την κλάση *«User_Login»*, οπότε η πρώτη λειτουργία της είναι να φορτώσει τις εγγραφές διαπιστευτηρίων από την μνήμη RAM στην λίστα της καρτέλας *«Sites List»*.

Οι εγγραφές εμφανίζονται στο κύριο παράθυρο μέσα σε έναν πίνακα *«JTable»* ταξινομημένες κατά το *ID* τους. Ο χρήστης, αν θέλει να χρησιμοποιήσει τα δεδομένα από κάποιο τμήμα των διαπιστευτηρίων του, **εκτός αυτό του Κωδικού Πρόσβασης**, μπορεί πολύ απλά να κάνει διπλό κλικ επάνω στο τμήμα αυτό και να μαρκάρει και να αντιγράψει τα δεδομένα που περιέχονται, εκτελώντας τον συνδυασμό πλήκτρων Ctrl + C. Έπειτα μπορεί να χρησιμοποιήσει τα δεδομένα αυτά επικολλώντας τα στο σημείο της αρεσκείας του (στα πεδία εισαγωγής της φόρμας πιστοποίησης χρήστη μιας ιστοσελίδας, σε κάποιο παράθυρο πιστοποίησης χρήστη ενός άλλου προγράμματος κ.τ.λ.).

Για να μπορέσει ο χρήστης να αντιγράψει τον Κωδικό Πρόσβασης που επιθυμεί και να τον χρησιμοποιήσει, πρέπει να πατήσει το κουμπί *«Copy pass»* της αντίστοιχης γραμμής της εγγραφής του. Κατά το πάτημα αυτού του κουμπιού, το πρόγραμμα διαβάζει την αποκρυπτογραφημένη τιμή του Κωδικού Πρόσβασης της συγκεκριμένης εγγραφής από την μνήμη RAM και την μεταβιβάζει το ίδιο στο «πρόχειρο» (clipboard) του λειτουργικού συστήματος, στο σημείο δηλαδή της μνήμης που χρησιμοποιεί και η διαδικασία αντιγραφής Ctrl + C, έτσι ώστε ο χρήστης να μπορέσει να την επικολλήσει εκεί που επιθυμεί.

Στη συνέχεια, αν ο χρήστης θελήσει να προσθέσει μία νέα εγγραφή διαπιστευτηρίων στη βάση δεδομένων, μπορεί να το κάνει πατώντας το κουμπί *«Add»* του παραθύρου. Τότε, εμφανίζεται ένα παράθυρο «εισαγωγής δεδομένων» (περιγράφεται αναλυτικά στο Κεφάλαιο 4.3.6) στο οποίο ο χρήστης καλείται να συμπληρώσει τα δεδομένα της καταχώρησης. Πατώντας το κουμπί *«Submit»* στο παράθυρο εισαγωγής, η εγγραφή αποθηκεύεται στη βάση δεδομένων και το παράθυρο κλείνει. Η νέα εγγραφή θα εμφανιστεί αυτόματα στον πίνακα *«JTable»* μετά την εισαγωγή της στην βάση δεδομένων, μέσω της μεθόδου *refreshMainTable()* της κλάσης *«Main_Window»*.

Παρομοίως, αν ο χρήστης θελήσει να διαγράψει κάποια εγγραφή διαπιστευτηρίων από τη βάση δεδομένων, πρέπει να πατήσει το κουμπί *«Delete»* του κυρίως παραθύρου. Με την ίδια λογική όπως και προηγουμένως, εμφανίζεται το παράθυρο «διαγραφής δεδομένων» (περιγράφεται αναλυτικά στο Κεφάλαιο 4.3.7) στο οποίο ο χρήστης καλείται να συμπληρώσει τον αριθμό *ID* της εγγραφής που θέλει να διαγράψει. Πατώντας το *«OK»* στο παράθυρο διαγραφής, η εγγραφή διαγράφεται από τη βάση δεδομένων και το παράθυρο κλείνει. Η νέα κατάσταση της βάσης δεδομένων θα εμφανιστεί αυτόματα στον πίνακα *«JTable»* μετά την προηγηθείσα διαγραφή, μέσω της μεθόδου *refreshMainTable()* της κλάσης *«Main_Window»*.

Τέλος, για την απενεργοποίηση του προγράμματος ο χρήστης μπορεί πολύ απλά να πατήσει το κουμπί *«Exit»* του κυρίως παραθύρου, να πατήσει στην επιλογή «Exit» από το μενού «File» ή να πατήσει το εικονίδιο *«X»* στην πάνω δεξιά γωνία του παραθύρου. Εκείνη τη στιγμή, πριν το πρόγραμμα απενεργοποιηθεί ολοκληρωτικά, διενεργείται αυτόματα κρυπτογράφηση της βάσης δεδομένων του που βρίσκεται στη μνήμη RAM (με χρήση του αλγορίθμου **AES 128-bit** και με κλειδί κρυπτογράφησης τον συνδυασμό ονόματος χρήστη και κύριου κωδικού πρόσβασης που χρησιμοποιήθηκε και για την αποκρυπτογράφησή της) και στη συνέχεια τα κρυπτογραφημένα πλέον δεδομένα της βάσης εγγράφονται πίσω στο αρχείο *«Data.gdm»*, διαγράφοντας τα παλιά περιεχόμενα του αρχείου.

Η ίδια ενημέρωση διενεργείται και στο αρχείο *«Keyd.crp»*, ούτως ώστε αυτό να περιέχει πλέον τα δεδομένα του νέου Διανύσματος Αρχικοποίησης που μόλις δημιουργήθηκε από τον αλγόριθμο AES.

4.3.6 Η κλάση πρόσθεσης νέας εγγραφής *«Add_Record»* του αρχείου *«Add_Record.java»*

Η κλάση *«Add_Record»* περιέχει την λειτουργικότητα για την εισαγωγή μιας νέας εγγραφής διαπιστευτηρίων στη βάση δεδομένων του προγράμματος. Ενεργοποιείται μέσω της κλάσης *«Main_Window»* όταν ο χρήστης κάνει κλικ στο κουμπί *«Add +»* του παραθύρου της.

Σχεδιαστικά, περιλαμβάνει 3 πεδία εισαγωγής δεδομένων:

- **Διεύθυνση URL της ιστοσελίδας (Website / URL)** ή διαφορετική ονομασία του συγκεκριμένου διαπιστευτηρίου που πρόκειται να καταχωρηθεί.
- **Όνομα χρήστη (Username)** το οποίο ο χρήστης έχει επιλέξει για την εισαγωγή του στην ιστοσελίδα ή στο πρόγραμμα που περιγράφηκε στο προηγούμενο πεδίο.
- **Κωδικός πρόσβασης (Password)** ο οποίος επίσης έχει επιλεχθεί από τον χρήστη για την εισαγωγή του στην ιστοσελίδα ή στο πρόγραμμα που περιγράφηκε στο προηγούμενο πεδίο.

Ο χρήστης, αφού εισάγει τα δεδομένα που επιθυμεί, πρέπει στη συνέχεια να πατήσει το κουμπί *«Submit»* ούτως ώστε αυτά να καταχωρηθούν στη βάση δεδομένων.

Μόλις πατηθεί το κουμπί *«Submit»*, η κλάση αποκρυπτογραφεί εκ νέου το αρχείο *«Data.gdm»* στη μνήμη RAM του συστήματος, διαβάζει την μεταβλητή *<LAST_ID_USED>* μέσα από αυτά τα αποκρυπτογραφημένα δεδομένα, προετοιμάζει την νέα εγγραφή με αναγνωριστικό εγγραφής το *<LAST_ID_USED> + 1*, προσθέτει την εγγραφή αυτή στη μνήμη, αυξάνει κατά *1* την μεταβλητή *<LAST_ID_USED>* του αρχείου που βρίσκεται αποκρυπτογραφημένα δεδομένα της μνήμης και τέλος, κρυπτογραφεί τα δεδομένα και τα επανεγγράφει στο αρχείο *«Data.gdm»*. Έτσι, με το που απενεργοποιηθεί το παράθυρο της κλάσης *«Add_Record»* και ο έλεγχος επιστρέψει πάλι σε αυτό της *«Main_Window»*, αν έχουν υπάρξει αλλαγές στα δεδομένα (που στην συγκεκριμένη περίπτωση υπήρξαν), η κλάση *«Main_Window»* διαγράφει τα υπάρχοντα δεδομένα από τον χώρο μνήμης της και τα φορτώνει ξανά από το αρχείο *«Data.gdm»* αποκρυπτογραφώντας τα και ενημερώνοντας εκ νέου την λίστα διαπιστευτηρίων του κεντρικού παραθύρου.

Σε διαφορετική περίπτωση, αν ο χρήστης δεν θελήσει τελικά να εισάγει κάποια εγγραφή στη βάση δεδομένων, μπορεί να πατήσει απλά το κουμπί *«Cancel»* το οποίο ακυρώνει τη διαδικασία, κλείνει το τρέχον παράθυρο και επιστρέφει τον έλεγχο στο κύριο παράθυρο του προγράμματος.

Να σημειωθεί επίσης ότι μόλις πατηθεί το κουμπί *«Submit»* από τον χρήστη και πριν την οποιαδήποτε καταχώρηση στη βάση δεδομένων, ελέγχονται από την κλάση τα πεδία *«Website / URL»*, *«Username»* και *«Password»* για το αν είναι συμπληρωμένα ή όχι και για το αν περιέχουν απαγορευμένους χαρακτήρες. Σε περίπτωση ενός εκ των δύο ενδεχομένων εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε καθώς και για την διόρθωσή του.

4.3.7 Η κλάση διαγραφής μιας υπάρχουσας εγγραφής *«Delete_Record»* του αρχείου *«Delete_Record.java»*

Η κλάση *«Delete_Record»* περιέχει την λειτουργικότητα για την διαγραφή μιας υπάρχουσας εγγραφής διαπιστευτηρίων από την βάση δεδομένων του προγράμματος. Ενεργοποιείται μέσω της κλάσης *«Main_Window»* όταν ο χρήστης κάνει κλικ στο κουμπί *«Delete -»* του παραθύρου της.

Σχεδιαστικά, περιλαμβάνει 1 πεδίο εισαγωγής δεδομένων:

- **Αναγνωριστικό εγγραφής (ID)** σύμφωνα με το οποίο η κλάση θα αναζητήσει και θα διαγράψει την εγγραφή από τη βάση δεδομένων που αντιστοιχεί σε αυτό.

Ο χρήστης, αφού εισάγει το αναγνωριστικό της εγγραφής που επιθυμεί να διαγράψει, πρέπει στη συνέχεια να πατήσει το κουμπί *«Submit»* ούτως ώστε να εκκινηθεί η διαδικασία διαγραφής δεδομένων της κλάσης από τη βάση δεδομένων.

Μόλις πατηθεί το κουμπί *«Delete»*, η κλάση αποκρυπτογραφεί εκ νέου το αρχείο *«Data.gdm»* στη μνήμη RAM του συστήματος, διαβάζει τα αποκρυπτογραφημένα περιεχόμενα και ψάχνει να βρει κάποια εγγραφή που να αντιστοιχεί σε αυτήν. Αν δεν υπάρχει εγγραφή με τέτοιο αναγνωριστικό, εμφανίζει ένα μήνυμα λάθους στον χρήστη και τερματίζει. Αν υπάρχει, τότε την αφαιρεί από την βάση δεδομένων, κρυπτογραφεί εκ νέου τα νέα δεδομένα της μνήμης RAM και τα επανεγγράφει στο αρχείο *«Data.gdm»*. Έτσι, με το που απενεργοποιηθεί το παράθυρο της κλάσης *«Delete_Record»* και ο έλεγχος επιστρέψει πάλι σε αυτό της *«Main_Window»*, αν έχουν υπάρξει αλλαγές στα δεδομένα (που στην συγκεκριμένη περίπτωση υπήρξαν), η κλάση *«Main_Window»* διαγράφει τα υπάρχοντα δεδομένα από τον χώρο μνήμης της και τα φορτώνει ξανά από το αρχείο *«Data.gdm»* αποκρυπτογραφώντας τα και ενημερώνοντας εκ νέου την λίστα διαπιστευτηρίων του κεντρικού παραθύρου.

Σε διαφορετική περίπτωση, αν ο χρήστης δεν θελήσει τελικά να διαγράψει κάποια εγγραφή από τη βάση δεδομένων, μπορεί να πατήσει απλά το κουμπί *«Cancel»* το οποίο ακυρώνει τη διαδικασία, κλείνει το τρέχον παράθυρο και επιστρέφει τον έλεγχο στο κύριο παράθυρο του προγράμματος (στην κλάση *«Main_Window»*).

Να σημειωθεί επίσης ότι μόλις πατηθεί το κουμπί *«Delete»* από τον χρήστη και πριν την οποιαδήποτε αναζήτηση στη βάση δεδομένων, ελέγχεται από την κλάση το πεδίο *«ID»* για το αν είναι συμπληρωμένο ή όχι και για το αν περιέχει απαγορευμένους χαρακτήρες. Σε περίπτωση ενός εκ των δύο ενδεχομένων εμφανίζεται ένα παράθυρο σφάλματος στον χρήστη που τον ενημερώνει για το αντίστοιχο πρόβλημα που προέκυψε καθώς και για την διόρθωσή του.

ΕΠΙΛΟΓΟΣ

Με βάση την ανάλυση που έγινε στην διατριβή αυτή, οδηγούμαστε στο συμπέρασμα ότι οι Διαχειριστές Κλειδαρίθμων είναι ιδιαίτερα χρήσιμοι στην καθημερινότητα των χρηστών, αφού τους απαλλάσσει από το γεγονός ότι πρέπει να θυμούνται απ έξω δεκάδες κωδικούς πρόσβασης, ονόματα χρήστη και λοιπά διαπιστευτήρια. Με ένα τέτοιο πρόγραμμα οι χρήστες έχουν όλα τα διαπιστευτήριά τους συγκεντρωμένα σε ένα σημείο, οργανωμένα σε κατηγορίες και όταν χρειαστούν κάτι το βρίσκουν εύκολα και το χρησιμοποιούν ή τους το τοποθετεί αυτόματα ο Διαχειριστής εκεί που πρέπει.

Από την άλλη όμως, εκτός των ευκολιών που αυτοί παρέχουν, υπάρχει και το θέμα της ασφάλειας. Οι Διαχειριστές Κλειδαρίθμων αποθηκεύουν και διαχειρίζονται τα απολύτως προσωπικά και ευαίσθητα δεδομένα των χρηστών. Αυτό σημαίνει ότι αν για οποιοδήποτε λόγο παραβιαστούν αυτά τα δεδομένα ο χρήστης θα βρεθεί φοβερά εκτεθειμένος και η ζημιά που θα του συμβεί μπορεί είναι τεράστια.

Αν και θεωρητικά τα προγράμματα αυτά μπορούν να μετατραπούν σε υπέρ ασφαλή συστήματα αποθήκευσης πληροφοριών - τα οποία μπορεί να είναι πρακτικά αδιάρρηκτα, με την συνεχή εξέλιξη της τεχνολογίας ολοένα και θα αναπτύσσονται νέα κακόβουλα προγράμματα και τεχνικές διείσδυσης οι οποίες θα εκμεταλλεύονται ευπάθειες ασφαλείας (άγνωστες μέχρι εκείνη τη στιγμή) και θα δίνουν τη δυνατότητα της μερικής ή ολοκληρωτικής υποκλοπής των διαπιστευτηρίων των χρηστών.

Συνεπώς, όσο άριστο και να είναι ένα πρόγραμμα αυτή τη στιγμή, το ίδιο άριστο δε συνεπάγεται ότι θα παραμείνει και μετά από κάποιο χρονικό διάστημα. Οι προγραμματιστές των Διαχειριστών Κλειδαρίθμων πρέπει λοιπόν να βρίσκονται σε μία συνεχή ενημέρωση των θεμάτων ασφαλείας και των επιθέσεων της εποχής και να επεκτείνουν συνεχώς το πρόγραμμά τους, προσθέτοντας σε αυτό νέες, διαφορετικές λειτουργίες ασφαλείας ή αναβαθμίζοντας τις ήδη υπάρχουσες.

Ένα θέμα επίσης που αναπτύχθηκε σε αυτήν την διατριβή είναι οι τεχνικές που χρησιμοποιούνται ήδη ή που θα μπορούσαν να χρησιμοποιηθούν για να ασφαλίσουν όσο περισσότερο γίνεται τους Διαχειριστές Κλειδαρίθμων και να τους κάνουν ανθεκτικούς στις επιθέσεις. Στις τεχνικές αυτές συμπεριλήφθηκαν:

- Πρόσβαση στο πρόγραμμα μέσω ενός **κύριου κωδικού πρόσβασης (master password) ή μιας φράσης πρόσβασης (passphrase) ή με συνδυασμό και των δύο**.
- **Κρυπτογράφηση του κύριου κωδικού πρόσβασης ή της φράσης πρόσβασης ή του συνδυασμού τους.**

- **Κρυπτογράφηση της βάσης δεδομένων.**

- **Χρήση εικονικού πληκτρολογίου** (virtual keyboard) για την εισαγωγή του κύριου κωδικού πρόσβασης στο πρόγραμμα (για ξεκλείδωμα), αλλά και την καταχώρηση διαπιστευτηρίων στη βάση δεδομένων.

- **Πιστοποίηση του χρήστη με πολλαπλά κριτήρια πρόσβασης** (multifactor authentication) επιπλέον του βασικού συστήματος πιστοποίησης, όπως με τη χρήση δακτυλικών αποτυπωμάτων, με αναγνώριση αμφιβληστροειδή, με αναγνώριση προσώπου, κ.α..

- **Χρήση συσκευής παραγωγής εξωτερικών κωδικών ασφαλείας** (security token generators) ή/και **έξυπνης κάρτας ασφαλείας** (security smart card), κάτι που αποτελεί υποκατηγορία της πιστοποίησης μέσω πολλαπλών κριτηρίων πρόσβασης.

- **Απενεργοποίηση εναλλαγής** (swapping) του δεσμευμένου χώρου μνήμης του προγράμματος από την μνήμη RAM προς το αρχείο σελιδοποίησης της μνήμης μόνιμης αποθήκευσης.

- **Ολοκληρωτικό κλείδωμα** του προγράμματος μετά από κάποιες αποτυχημένες προσπάθειες εισόδου.

- **Εισαγωγή επίλυσης τεστ CAPTCHA** επιπλέον του βασικού συστήματος πιστοποίησης, κάτι που επίσης αποτελεί υποκατηγορία της πιστοποίησης μέσω πολλαπλών κριτηρίων πρόσβασης.

Διαβάζοντας ξανά όλες τις παραπάνω τεχνικές συμπεραίνουμε ότι με τη χρήση τους μπορούν να μετατρέψουν τον Διαχειριστή Κλειδαρίθμων σε ένα φρούριο πληροφοριών.

Πριν όμως τις υλοποιήσουμε στην πράξη σαν προγραμματιστές πρέπει να σκεφτούμε και την άλλη πλευρά και να φανταστούμε έναν χρήστη ο οποίος κάθε φορά που επιθυμεί να εισέλθει στο πρόγραμμα πρέπει να εκτελεί διαδοχικά σχεδόν όλες τις παραπάνω τεχνικές, αλλά και να έχει πάντα μαζί του τις εξωτερικές συσκευές πιστοποίησης. Εκτός του ότι το παραπάνω σενάριο είναι αρκετά χρονοβόρο και μπορεί να αποβεί πολύ κουραστικό, ας προσθέσουμε επίσης σε αυτό και το ενδεχόμενο ότι μπορεί κάποια στιγμή από μόνος του ο χρήστης να απενεργοποιήσει το πρόγραμμα (σκόπιμα ή κατά λάθος) ή να απενεργοποιηθεί το ίδιο από μόνο του (λόγω του ότι «κόλλησε» ο υπολογιστής ή λόγω κάποιας διακοπής ρεύματος). Σε μια τέτοια περίπτωση αν χρειαστεί να επανενεργοποιηθεί πάλι, πρέπει να εκτελεστούν ξανά από την αρχή όλα τα προηγούμενα βήματα πιστοποίησης.

Αυτό, για πολλούς χρήστες φαντάζει (και αποτελεί) ένα εφιαλτικό και κουραστικό σενάριο. Εδώ φαίνεται ξεκάθαρα η ισορροπία που πρέπει να τηρηθεί μεταξύ της χρηστικότητας και της ασφάλειας. Όπως φάνηκε και από

την έρευνα του Κεφαλαίου 2.1 που πραγματοποιήθηκε στους χρήστες του Πανεπιστημίου Θεσσαλίας, αυτοί θα χρησιμοποιούσαν μόνο ένα απλό στη χρήση πρόγραμμα Διαχείρισης Κλειδαρίθμων, το οποίο θα είναι όσο το δυνατόν ελαφρύ γίνεται (από άποψη κατανάλωσης πόρων της συσκευής τους) και δεν θα τους παρενοχλεί συνεχώς με διάφορα μηνύματα / δεν θα τους κουράζει.

Κατά συνέπεια, για την ύπαρξη ενός τέτοιου προγράμματος, πρέπει αρχικά (από τη μεριά της ασφάλειας) αυτό να περιλαμβάνει μόνο τις βασικές επιλογές πιστοποίησης για την εξουσιοδότηση ενός χρήστη (όπως όνομα χρήστη, κύριος κωδικός πρόσβασης) (τουλάχιστον ως προεπιλογή), αλλά και να είναι όσο το δυνατόν απλό στη χρήση και κατανοητό γίνεται. Πως μπορεί όμως ο χρήστης να νιώθει (και να είναι πράγματι) ασφαλής με χρήση «απλών» τεχνικών πιστοποίησης;

Η απάντηση σε αυτό το ερώτημα δόθηκε στο Κεφάλαιο 2.3, όπου αναφέρεται το γεγονός ότι αν το μηχάνημα που χρησιμοποιεί ένας χρήστης είναι εξασφαλισμένο από επιθέσεις διείσδυσης ή υποκλοπής και από την εγκατάσταση κακόβουλων προγραμμάτων τότε δεν υπάρχει ανησυχία για χρήση «απλών» τεχνικών πιστοποίησης στους Διαχειριστές Κλειδαρίθμων. Για να διατηρηθεί όμως ένα μηχάνημα «καθαρό» και να θεωρείται «έμπιστο» πρέπει να τηρείται συνεχώς μια σειρά από κανόνες. Πρέπει δηλαδή (περιληπτικά):

- Να επιλέγονται και να χρησιμοποιούνται **ισχυροί κωδικοί πρόσβασης** από τους χρήστες, για οποιοδήποτε πρόγραμμα, ιστοσελίδα ή υπηρεσία του λειτουργικού συστήματος απαιτεί πιστοποίηση μέσω αυτών και να αλλάζονται τουλάχιστον μια φορά το μήνα. Οι ισχυροί και πολύ «δύσκολοι» κωδικοί πρόσβασης δεν αποτελούν πρόβλημα για τους χρήστες αφού ποτέ δε θα χρειαστεί να τους αποστηθίσουν στο μυαλό τους, αυτή τη δουλειά την έχει αναλάβει εξ ολοκλήρου ο Διαχειριστής Κλειδαρίθμων.

- **Οι χρήστες να μην ανακοινώνουν ποτέ και σε κανέναν τους κωδικούς πρόσβασής τους** και να περιορίσουν γενικά τα προσωπικά δεδομένα που μοιράζονται δημοσίως στο Διαδίκτυο.

- Να υπάρχει εγκατεστημένο **ένα γνήσιο λειτουργικό σύστημα** στον υπολογιστή (και όχι κάποιο πειρατικό ή κάποια ειδική προσαρμοσμένη έκδοση από άλλον χρήστη), το οποίο θα έχει αγορασθεί (εκτός κι αν διανέμεται δωρεάν) και θα έχει κατεβεί από την επίσημη ιστοσελίδα του κατασκευαστή του και μόνο, όχι από τρίτους.

- Το **λειτουργικό σύστημα να διατηρείται διαρκώς ενημερωμένο**, εγκαθιστώντας πάντα σε αυτό τις τελευταίες εκδόσεις ασφαλείας που έχουν κυκλοφορήσει από τον κατασκευαστή του.

- Να υπάρχει εγκατεστημένο ένα επίσης **γνήσιο αντιϊκό πρόγραμμα (antivirus)** στο μηχάνημα το οποίο θα παρέχει επιπλέον και

δυνατότητα φιλτραρίσματος δεδομένων δικτύου (firewall) και το οποίο θα διατηρείται διαρκώς ενημερωμένο.

- Να υπάρχει εγκατεστημένο (προληπτικά) ένα **σύστημα ανίχνευσης εισβολών (IDS).**

- Να δίνεται πάντα η δέουσα **προσοχή από τον χρήστη σε όλα τα είδη των προγραμμάτων που πρόκειται να εγκαταστήσει** στον υπολογιστή του.

- Να αποφεύγεται παντελώς η εγκατάσταση ή η εκτέλεση παράνομων προγραμμάτων **cracking** και **keygen**.

Αν ένας χρήστης τηρεί τους παραπάνω απλούς κανόνες τότε θεωρείται γενικά ασφαλής από οποιαδήποτε είδη απειλών, δηλαδή παρακολουθήσεων, κακόβουλων προγραμμάτων και επιθέσεων. Οι κανόνες αυτοί δεν αποτελούν βέβαια «νόμο» για την ασφάλειά του (και συνεπώς για την ασφάλεια των διαπιστευτηρίων στους Διαχειριστές Κλειδαρίθμων), αποτελούν όμως πολύ ισχυρά μέτρα προστασίας που θα μπορούσε (και πρέπει) να πάρει.

Συνεπώς, όσο πιο προστατευμένοι είμαστε από το μηχάνημά μας και από τις κινήσεις μας, τόσο πιο απλός και εύκολος στη χρήση του μπορεί να γίνει ο Διαχειριστής Κλειδαρίθμων. Αν αντίθετα δεν μπορούμε να τηρήσουμε τους παραπάνω κανόνες ή δεν έχουμε την δυνατότητα να το κάνουμε (λόγω οικονομικών θεμάτων ή λόγω χρήσης ξένων ή δημόσιων υπολογιστών), τότε επιβάλουμε στους Διαχειριστές Κλειδαρίθμων να χρησιμοποιούν πολύ ισχυρά μέτρα προστασίας, κάνοντας τη ζωή μας ίσως πιο κουραστική.

Γενικά μέσα από την πολυήμερη έρευνα που πραγματοποιήθηκε για την εκπόνηση της διατριβής αυτής παρατηρήθηκε το γεγονός ότι αν και οι Διαχειριστές Κλειδαρίθμων τη στιγμή της ανακάλυψής τους ως προγράμματα φάνταζαν ως ένα τεράστιο τεχνολογικό επίτευγμα στον τομέα της ασφάλειας και της προστασίας των προσωπικών δεδομένων, πλέον, λόγω των κακών τεχνικών σχεδίασης που μπορεί να χρησιμοποιηθούν σε πολλούς από αυτούς, ή λόγω των όλο και νεότερων τεχνικών παραβίασης που έρχονται στο φως, οι ίδιοι οι Διαχειριστές μπορούν να εκθέσουν τους χρήστες σε μεγάλο κίνδυνο με την αποκάλυψη πλήθους, σοβαρών και μη, προσωπικών δεδομένων τους.

Το συμπέρασμα που προκύπτει λοιπόν από αυτή τη διατριβή είναι ότι οι χρήστες δεν θα πρέπει να επαναπαύονται και να αισθάνονται ισχυροί χρησιμοποιώντας τέτοια προγράμματα. Σίγουρα μπορούν να βελτιώσουν το επίπεδο ασφαλείας τους με την χρήση τους, και προτείνεται να το κάνουν, αλλά με τυχών λάθος διαδικασίες χρήσης των προγραμμάτων αυτών, με τυχών μολυσμένα μηχανήματα ή με χρήση Διαχειριστών Κλειδαρίθμων μη ελεγμένων και δοκιμασμένων από αυθεντίες του χώρου, οι χρήστες μπορούν να βρεθούν περισσότερο εκτεθειμένοι από όσο ήταν στην αρχή χωρίς την χρήση τους...

ΠΑΡΑΡΤΗΜΑ

Π1 - Παράδειγμα κρυπτογράφησης και αποκρυπτογράφησης με χρήση αλγορίθμου AES 128-bit σε γλώσσα JAVA

```
import java.io.UnsupportedEncodingException;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

import java.util.Arrays;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

import org.apache.commons.codec.binary.Base64;


public class AES
{


    private static SecretKeySpec secretKey ;
    private static byte[] key ;
    private static String decryptedString;
    private static String encryptedString;



    public static void setKey(String myKey){

        MessageDigest sha = null;

        try {

            key = myKey.getBytes("UTF-8");
            sha = MessageDigest.getInstance("SHA-1");
            key = sha.digest(key);
            key = Arrays.copyOf(key, 16); // use only first 128 bit
            secretKey = new SecretKeySpec(key, "AES");


        } catch (NoSuchAlgorithmException e) {

            e.printStackTrace();

        } catch (UnsupportedEncodingException e) {

            e.printStackTrace();

        }
```

```
}

public static String getDecryptedString() {

    return decryptedString;

}

public static void setDecryptedString(String decryptedString) {

    AES.decryptedString = decryptedString;

}

public static String getEncryptedString() {

    return encryptedString;

}

public static void setEncryptedString(String encryptedString) {

    AES.encryptedString = encryptedString;

}

public static String encrypt(String strToEncrypt) {

    try
    {

        Cipher cipher = Cipher.getInstance("AES/ECB/
                                                PKCS5Padding");

        cipher.init(Cipher.ENCRYPT_MODE, secretKey);


        setEncryptedString(Base64.encodeBase64String(
                cipher.doFinal(strToEncrypt.getBytes("UTF-8"))));

    }
    catch (Exception e) {

        System.out.println("Error while encrypting: "
                                                +e.toString());
    }

    return null;

}

public static String decrypt(String strToDecrypt) {

    try
    {

        Cipher cipher =
                    Cipher.getInstance("AES/ECB/PKCS5PADDING");

        cipher.init(Cipher.DECRYPT_MODE, secretKey);
```

```
            setDecryptedString(new String(
                                cipher.doFinal(
                                 Base64.decodeBase64(strToDecrypt))));

        }
        catch (Exception e) {

            System.out.println("Error while decrypting: "
                                                    +e.toString());
        }

        return null;

    }

    public static void main(String args[]) {

        final String strToEncrypt = "My text to encrypt";
        final String strPssword = "encryptor key";

        AES.setKey(strPssword);
        AES.encrypt(strToEncrypt.trim());

        System.out.println("String to Encrypt: " + strToEncrypt);
        System.out.println("Encrypted: " + AES.getEncryptedString());

        final String strToDecrypt = AES.getEncryptedString();

        AES.decrypt(strToDecrypt.trim());

        System.out.println("String To Decrypt: " + strToDecrypt);
        System.out.println("Decrypted: " + AES.getDecryptedString());

    }

}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Rubenking, Neil J. "Six Great Password Managers". PC Magazine. 11 Μαρτίου 2011. Ανακτήθηκε στις 28 Ιανουαρίου 2014. Ηλεκτρονική διεύθυνση: http://www.pcmag.com/article2/0,2817,2381432,00.asp

[2] Parker, Jason (11 April 2014). "Take control of password chaos with these six password managers". CNET. 11 Απριλίου 2014. Ανακτήθηκε στις 28 Ιανουαρίου 2014. Ηλεκτρονική διεύθυνση: http://www.cnet.com/uk/news/best-password-managers/

[3] Li, Zhiwei; He, Warren; Akhawe, Devdatta; Song, Dawn. "The Emperor's New Password Manager: Security Analysis of Web-based Password Managers" (PDF). 2014. Ανακτήθηκε στις 25 Δεκεμβρίου 2014. Ηλεκτρονική διεύθυνση: https://devd.me/papers/pwdmgr-usenix14.pdf

[4] Adida, Ben; Barth, Adam; Jackson, Collin. "Rootkits for JavaScript Environments Ben" (PDF). 2009. Ανακτήθηκε στις 20 Δεκεμβρίου 2014. Ηλεκτρονική διεύθυνση: http://www.adambarth.com/papers/2009/adida-barth-jackson.pdf

[5] M. Blanchou and P. Youn. "Password managers: Exposing passwords everywhere" (PDF). Nov 2013. Ανακτήθηκε στις 20 Δεκεμβρίου 2014. Ηλεκτρονική διεύθυνση: https://www.isecpartners.com/media/106983/password_managers_nov13.pdf

[6] "Password Manager". Wikipedia, The free encyclopedia. Ημερομηνία πρόσβασης: 5 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://en.wikipedia.org/wiki/Password_manager

[7] "Security Token". Wikipedia, The free encyclopedia. Ημερομηνία πρόσβασης: 6 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://en.wikipedia.org/wiki/Security_token

[8] "KeePass". Wikipedia, The free encyclopedia. Ημερομηνία πρόσβασης: 10 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://en.wikipedia.org/wiki/KeePass

[9] "LastPass". Wikipedia, The free encyclopedia. Ημερομηνία πρόσβασης: 10 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://en.wikipedia.org/wiki/LastPass

[10] "The best Password Managers". PC Magazine (web). Ημερομηνία πρόσβασης: 10 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://www.pmag.com/article2/0,2817,2407168,00.asp

[11] "LastPass 1.50 Review & Rating". PC Magazine (web). Ημερομηνία πρόσβασης: 10 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://www.pcmag.com/article2/0,2817,2343562,00.asp#fbid=rg3fbOOKZ4v

[12] Lynn Hathaway. "National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information" (PDF). Ιούνιος 2003. Ανακτήθηκε στις 15 Φεβρουαρίου 2014. Ηλεκτρονική διεύθυνση: http://csrc.nist.gov/groups/ST/toolkit/documents/aes/CNSS15FS.pdf

[13] John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting. "Improved Cryptanalysis of Rijndael", Seventh Fast Software Encryption Workshop, Springer-Verlag. 2000. pp. 213–230. Ανακτήθηκε στις 17 Φεβρουαρίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/paper-rijndael.html

[14] Ou, George. "Is encryption really crackable?". Ziff-Davis Network. Απρίλιος 2006. Ανακτήθηκε στις 20 Φεβρουαρίου 2014. Ηλεκτρονική διεύθυνση: http://www.zdnet.com/article/is-encryption-really-crackable/

[15] Shashank Sharma. "8 of the best Linux password managers - MyPasswords Password Manager Review". Techradar.com (from Linux Format Issue 139). 5 Ιανουαρίου 2011. Ημερομηνία πρόσβασης: 12 Ιανουαρίου 2015. Ηλεκτρονική διεύθυνση: http://www.techradar.com/news/software/applications/8-of-the-best-linux-password-managers-916152/7

[16] Bruce Schneier. "AES News, Crypto-Gram Newsletter, September 15, 2002". Schneier on Security, A blog covering security and security technology. 2002. Ανακτήθηκε στις 22 Φεβρουαρίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/crypto-gram/archives/2002/0915.html

[17] Niels Ferguson, Richard Schroeppel, Doug Whiting. "Proceedings of Selected Areas in Cryptography, 2001, Lecture Notes in Computer Science" (PDF/PostScript). Springer-Verlag. 2001. pp. 103–111. Ανακτήθηκε στις 6 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: http://web.archive.org/web/20061104080748/http://www.macfergus.com/pub/rdalgeq.html

[18] Bruce Schneier. "AES Announced". Schneier on Security, A blog covering security and security technology. 2000. Ανακτήθηκε στις 6 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/crypto-gram/archives/2000/1015.html

[19] Bruce Schneier. "New Attack on AES". Schneier on Security, A blog covering security and security technology. 1η Ιουλίου 2009. Ανακτήθηκε

στις 11 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/blog/archives/2009/07/new_attack_on_a.html

[20] Biryukov, Alex; Khovratovich, Dmitry. "Related-key Cryptanalysis of the Full AES-192 and AES-256". Submitted to a conference. 4 Δεκεμβρίου 2009. Ανακτήθηκε στις 11 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2009/317

[21] Nikolić, Ivica. "Distinguisher and Related-Key Attack on the Full AES-256". Advances in Cryptology – CRYPTO 2009. Springer Berlin / Heidelberg. 2009. pp. 231–249. Ανακτήθηκε στις 15 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: http://link.springer.com/chapter/10.1007%2F978-3-642-03356-8_14

[22] Bruce Schneier. "Another New AES Attack". Schneier on Security, A blog covering security and security technology. 30 Ιουλίου 2009. Ανακτήθηκε στις 17 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/blog/archives/2009/07/another_new_aes.html

[23] Alex Biryukov; Orr Dunkelman; Nathan Keller; Dmitry Khovratovich; Adi Shamir. "Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds". Submitted to a conference. 19 Αυγούστου 2009. Ανακτήθηκε στις 17 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2009/374

[24] Henri Gilbert; Thomas Peyrin. "Super-Sbox Cryptanalysis: Improved Attacks for AES-like permutations". Submitted to a conference. 9 Νοεμβρίου 2009. Ανακτήθηκε στις 17 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2009/531

[25] Vincent Rijmen. "Practical-Titled Attack on AES-128 Using Chosen-Text Relations" (PDF). 2010. Ανακτήθηκε στις 7 Ιουλίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2010/337.pdf

[26] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. "Biclique Cryptanalysis of the Full AES" (PDF). 2011. Ανακτήθηκε στις 8 Ιουλίου 2014. Ηλεκτρονική διεύθυνση: http://research.microsoft.com/en-us/projects/cryptanalysis/aesbc.pdf

[27] "Index of formal scientific papers". Cr.yp.to. 2004. Ανακτήθηκε στις 7 Ιουλίου 2014. Ηλεκτρονική διεύθυνση: http://cr.yp.to/papers.html#cachetiming

[28] Bruce Schneier. "AES Timing Attack". Schneier on Security, A blog covering security and security technology. 2005. Ανακτήθηκε στις 8 Απριλίου 2014. Ηλεκτρονική διεύθυνση: https://www.schneier.com/blog/archives/2005/05/aes_timing_atta_1.html

[29] Dag Arne Osvik1; Adi Shamir2; Eran Tromer2. “Cache Attacks and Countermeasures: the Case of AES” (PDF). 20 Νοεμβρίου 2005. Ανακτήθηκε στις 20 Απριλίου 2014. Ηλεκτρονική διεύθυνση: http://cs.tau.ac.il/~tromer/papers/cache.pdf

[30] Dhiman Saha, Debdeep Mukhopadhyay, Dipanwita RoyChowdhury. “A Diagonal Fault Attack on the Advanced Encryption Standard” (PDF). 2009. Ανακτήθηκε στις 26 Απριλίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2009/581.pdf

[31] Endre Bangerter, David Gullasch and Stephan Krenn. “Cache Games – Bringing Access-Based Cache Attacks on AES to Practice” (PDF). 2010. Ανακτήθηκε στις 29 Απριλίου 2014. Ηλεκτρονική διεύθυνση: http://eprint.iacr.org/2010/594.pdf

[32] Discussion forum. “Breaking AES-128 in realtime, no ciphertext required | Hacker News”. Hacker News - Y Combinator. 2010. Ανακτήθηκε στις 3 Μαρτίου 2014. Ηλεκτρονική διεύθυνση: https://news.ycombinator.com/item?id=1937902

[33] Tom St Denis, Simon Johnson. “Cryptography for Developers” (hard copy book). Syngress Publishing. 2007. pp. 4-10.

[34] Verizon Business RISK team, “2009 Data Breach Investigations Supplemental Report” (PDF), 2009, Ανακτήθηκε στις 10 Οκτωβρίου 2014, Ηλεκτρονική διεύθυνση: http://www.verizonenterprise.com/resources/security/reports/rp_2009-data-breach-investigations-supplemental-report_en_xg.pdf