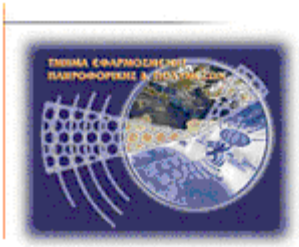
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή εργασία

Ασφαλής μετάδοση μηνυμάτων πάνω από ένα ασύρματο ομότιμο δίκτυο

Σαρρής Παρασκευάς (Αριθμός Μητρώου: 358)
E-mail: parsarr@gmail.com

Ηράκλειο – Ιούνιος 2012

Επιβλέπων Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

Υπεύθυνη Δήλωση: *Βεβαιώνω ότι είμαι συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην πτυχιακή εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η πτυχιακή εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του προγράμματος σπουδών του Τμήματος Εφαρμοσμένης Πληροφορικής και Πολυμέσων του Τ.Ε.Ι. Κρήτης.*

Ευχαριστίες

Η ολοκλήρωση της παρούσης πτυχιακής εργασίας, και κατ' επέκταση η απόκτηση του πτυχίου μου από το τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων, είναι χωρίς καμία αμφιβολία ένα πολύ σημαντικό γεγονός στην έως τώρα πορεία μου.

Κατά τη διάρκεια αυτής προσπάθειας υπήρξαν αρκετά άτομα που με βοήθησαν σημαντικά, έτσι κρίνω ότι θα έπρεπε να εκφράσω μέσα από αυτές τις γραμμές την ευγνωμοσύνη μου προς τα πρόσωπα τους.

Θεωρώ ότι τη μεγαλύτερη συμβολή για την ολοκλήρωση του παρόντος έργου την έχει ο αξιότιμος Δόκτωρ Χαράλαμπος Μανιφάβας καθότι, πριν ακόμα ξεκινήσω με τη συγγραφή της πτυχιακής εργασίας, είχα την τύχη να παρακολουθήσω τη σειρά μαθημάτων της Ασφάλειας Πληροφοριακών Συστημάτων που παραδίδονται από το Δόκτωρ Μανιφάβα.

Μέσα από τη συνεργασία αυτή ήρθα σε επαφή με μια σειρά από νέες έννοιες και ιδέες που σχετίζονται με τον τομέα της ασφάλειας πληροφοριών, γεγονός που μου κίνησε το ενδιαφέρον και με ώθησε στο να ασχοληθώ σε μεγαλύτερο βάθος με τον εν λόγω τομέα.

Έτσι, χάρηκα ιδιαίτερα με την ευκαιρία που μου παρουσιάστηκε και πραγματικά είμαι ευγνώμων προς το Δόκτωρ Μανιφάβα που μου ανέθεσε το θέμα της εργασίας, ενώ στη συνέχεια δίνοντας μου τις κατάλληλες κατευθυντήριες γραμμές κατέστησε εφικτή την τελική υλοποίηση του παρόντος έργου.

Στη συνέχεια, θα ήθελα να ευχαριστήσω ιδιαιτέρως τη μητέρα μου και τον αδερφό μου, καθώς και όλα τα μέλη του στενού μου οικογενειακού περιβάλλοντος, τόσο για το ενδιαφέρον που επέδειξαν όσο και για την ψυχολογική και υλική υποστήριξη που έλαβα σε όλη τη διάρκεια των σπουδών μου.

Ακόμη, θα πρέπει να ευχαριστήσω τους στενούς μου φίλους και παράλληλα να τους ζητήσω συγγνώμη που για ένα μεγάλο διάστημα αποκόπηκα από αυτούς και πολλές φορές, χωρίς να το θέλω ή χωρίς να το καταλαβαίνω, δεν είχα φερθεί σωστά προς αυτούς.

Κλείνοντας με το κομμάτι των ευχαριστιών, δεν θα μπορούσα να μην αναφερθώ στον αγαπημένο μου πατέρα, όπου αν και νωρίς εγκατέλειψε τον κόσμο που ζούμε, είχε καταλυτικό ρόλο στην ανατροφή μου και τη διαμόρφωση του χαρακτήρα μου. Έτσι, πιστεύω ότι το ελάχιστο που θα μπορούσα να κάνω από την πλευρά μου θα ήταν να αφιερώσω στη μνήμη του πατέρα μου το αποτέλεσμα που προκύπτει από την ολοκλήρωση της παρούσης πτυχιακής εργασίας.

Ιστορικό εκδόσεων

Ημερομηνία	Έκδοση	Λεπτομέρειες
1/11/2011	1.0	Ολοκλήρωση κεφαλαίων 1, 2 και 3
11/12/2011	2.0	Διορθώσεις πάνω στην προηγούμενη έκδοση, προσθήκη 2.2.4
23/12/2011	3.0	Προσθήκη κεφαλαίου 4
11/2/2012	4.0	Προσθήκη κεφαλαίου 5
7/3/2012	5.0	Προσθήκη κεφαλαίου 6
17/4/2012	6.0	Δημιουργία παραρτημάτων
29/4/2012	7.0	Προσθήκη υποενοτήτων που σχετίζονται με την αποτίμηση της ασφάλειας που παρέχεται από την εφαρμογή ασφαλούς ανταλλαγής σύντομων μηνυμάτων
21/5/2012	8.0	Προσθήκη υποενοτήτων που σχετίζονται με την ασφάλεια που παρέχεται από το πρωτόκολλο πιστοποίησης της ταυτότητας του χρήστη μιας διαδικτυακής υπηρεσίας
26/5/2012	9.0	Διορθώσεις σε ότι έχει προηγηθεί, προσθήκη της παρουσίασης εφαρμογής ασφαλούς ανταλλαγής σύντομων μηνυμάτων
28/5/2012	10.0	Προσθήκη δημοσίευσης της εφαρμογής ασφαλούς ανταλλαγής σύντομων μηνυμάτων

Περίληψη

Η ακόλουθη πτυχιακή εργασία απαρτίζεται από δύο μέρη και παρέχει λύσεις σε μια σειρά από θέματα που σχετίζονται με τη διαφύλαξη προσωπικών δεδομένων.

Το πρώτο σκέλος της εργασίας εστιάζει στην υλοποίηση μιας εφαρμογής για φορητές συσκευές, όπως είναι τα κινητά τηλέφωνα ή Smart Phones που είναι εφοδιασμένα με το περιβάλλον Java Micro Edition.

Η εν λόγω εφαρμογή συνδυάζει τα πλεονεκτήματα που παρέχονται από την ασύμμετρη και τη συμμετρική κρυπτογραφία και σχηματίζει με αυτό τον τρόπο ένα μονοπάτι ασφαλούς επικοινωνίας, μέσω του οποίου διασφαλίζεται το περιεχόμενο των σύντομων μηνυμάτων που ανταλλάσσονται από τους χρήστες της εφαρμογής.

Στο έτερο σκέλος της εργασίας έχουμε το σχεδιασμό και την υλοποίηση ενός Two Factor Authentication Protocol, δηλαδή ενός πρωτοκόλλου πιστοποίησης ταυτότητας που βασίζεται σε δύο παράγοντες.

Το συγκεκριμένο πρωτόκολλο βασίζεται σε μια υπόθεση, σύμφωνα με την οποία η εξακρίβωση της ταυτότητας ενός χρήστη μιας διαδικτυακής υπηρεσίας προκύπτει μέσα από τη συνεργασία μιας διαδικτυακής εφαρμογής και μιας εφαρμογής για φορητές συσκευές.

Στο τρέχον σενάριο, ο χρήστης που κατορθώνει να αποδείξει ότι όντως είναι αυτός που ισχυρίζεται ότι είναι, αποκτά πρόσβαση σε μία προστατευμένη ιστοσελίδα.

Για τις ανάγκες του πρωτοκόλλου αναπτύχθηκαν οι κατάλληλες εφαρμογές, δηλαδή μια διαδικτυακή εφαρμογή, σύμφωνα με τις προδιαγραφές της πλατφόρμας Java Enterprise Edition, και μια εφαρμογή που προορίζεται για τις φορητές συσκευές που διαθέτουν το περιβάλλον Java Micro Edition.

Abstract

The following thesis is comprised by two parts and provides clear solutions to a series of matters related to the assurance of personal information.

The first stage focuses on implementing an application that is suitable for a mobile device, like a mobile phone or a smart phone that is equipped with the Java Micro Edition environment.

The implemented application takes advantage of the combined use of symmetric and public key cryptography and during its run-time establishes a secure communications route, through which the users of the application are able to exchange encrypted short messages.

The second part of the thesis contains the design and the realisation of a Two Factor Authentication Protocol.

This protocol is based on a scenario, according to which a user of web service is fully identified through the cooperation of a web application and a mobile device application.

In the current scenario, the user that manages to prove that he is exactly who he claims to be, is granted with access to a protected web page.

In order to fulfil the needs of this protocol the appropriate applications were created. The perfect combination was a web application, conformed to the specifications of Java Enterprise Edition platform, and an application aiming for a mobile device equipped with the Java Micro Edition environment.

Πίνακας Περιεχομένων

Ευχαριστίες ... **iii**
Ιστορικό εκδόσεων ... **iv**
Περίληψη ... **v**
Abstract ... **vi**
Πίνακας Περιεχομένων ... **vii**
Πίνακας Εικόνων ... **x**
Πίνακας Πινάκων ... **xvii**
Εισαγωγή ... **1**
Γενική Ιδέα ... 1
Σκοπός της εργασίας ... 1
Συνοπτική περιγραφή ... 3
Διάρθρωση της εργασίας ... 4
Μέρος Πρώτο ... **6**
Κεφάλαιο 1 - Δίκτυα κινητής τηλεφωνίας ... **7**
1.1 Η εξέλιξη των δικτύων κινητής τηλεφωνίας ... 7
1.1.1 Πριν από την πρώτη γενιά κινητής τηλεφωνίας ... 8
1.1.2 Η πρώτη γενιά κινητής τηλεφωνίας ... 9
1.1.3 Η δεύτερη γενιά κινητής τηλεφωνίας ... 10
1.1.4 Ανάμεσα σε δεύτερη και τρίτη γενιά ... 13
1.1.5 Η τρίτη γενιά κινητής τηλεφωνίας ... 15
1.1.6 Μια ματιά στο μέλλον ... 18
1.2 Το δίκτυο GSM ... 21
1.2.1 Η αρχή της κυψέλης ... 21
1.2.2 Η αρχιτεκτονική του GSM ... 23
1.3 Το GPRS ... 29
1.3.1 Η αρχιτεκτονική του δικτύου GSM μετά την ενσωμάτωση του GPRS ... 29
1.4 Το δίκτυο UMTS ... 31
1.4.2 Η αρχιτεκτονική του UMTS σύμφωνα με τις προδιαγραφές Release 4 ... 36
1.4.3 Η αρχιτεκτονική του UMTS σύμφωνα με τις προδιαγραφές Release 5 ... 37
Κεφάλαιο 2 - Υπηρεσία σύντομων μηνυμάτων ... **39**
2.1 Περιγραφή της υπηρεσίας σύντομων μηνυμάτων ... 39
2.1.1 Εφαρμογές που βασίζονται στη χρήση SMS ... 39
2.1.2 Μετάδοση σύντομων μηνυμάτων ... 41
2.1.3 Τεχνικά χαρακτηριστικά ενός Σύντομου Μηνύματος ... 47
2.2 Οι εντολές AT και ο τρόπος αποστολής SMS μέσω PC ... 54
2.2.1 Οι πιθανοί τρόποι αποστολής SMS μηνυμάτων μέσω ενός PC ... 54
2.2.2 Οι εντολές AT ... 55
2.2.3 Η διαδικασία αποστολής ενός σύντομου μηνύματος μέσω PC ... 58
2.2.4 Η χρησιμοποίηση του PDUspy ... 70
Κεφάλαιο 3 - Η ασφάλεια των δικτύων κινητής τηλεφωνίας ... **77**
3.1 Δίκτυα πρώτης γενιάς ... 77
3.2 Δίκτυα δεύτερης γενιάς ... 77
3.2.1 Το μοντέλο ασφάλειας του δικτύου GSM ... 78
3.2.2 Αυθεντικοποίηση στο GSM ... 78
3.2.3 Η ανωνυμία στο GSM ... 84
3.2.4 Η εμπιστευτικότητα των επικοινωνιών στο GSM ... 87
3.2.5 Τρωτά σημεία στην ασφάλεια του GSM ... 89

3.3 Δίκτυα τρίτης γενιάς 93
3.3.1 Η ασφάλεια του δικτύου UMTS 94
3.3.2 Η αυθεντικοποίηση ανάμεσα στο χρήστη και το δίκτυο UMTS 95
3.3.3 Η δημιουργία των διανυσμάτων αυθεντικοποίησης 98
3.3.4 Η χρησιμοποίηση προσωρινών ταυτοτήτων 102
3.3.5 Κρυπτογράφηση του UTRAN 104
3.3.6 Η διαφύλαξη της ακεραιότητας των σημάτων στο UTRAN 107
3.4 Η ασφάλεια που παρέχεται σε πραγματικές συνθήκες 109
3.4.1 Η περίπτωση των υποκλοπών στην Αθήνα 110
Κεφάλαιο 4 - Κρυπτογραφία 112
4.1 Κρυπτογραφικά συστήματα 112
4.2 Συμμετρική κρυπτογραφία 113
4.2.1 Βασικές αρχές της συμμετρικής κρυπτογραφίας 113
4.2.2 Η επιλογή του αλγορίθμου για το AES 115
4.2.3 Ο τρόπος λειτουργίας του AES 116
4.2.4 Κρυπτανάλυση του AES 128
4.3 Κρυπτογραφία δημοσίου κλειδιού 128
4.3.1 Βασικές αρχές της κρυπτογραφίας δημοσίου κλειδιού 129
4.3.2 Εφαρμογές της κρυπτογραφίας δημοσίου κλειδιού 130
4.3.3 Ο αλγόριθμος RSA 132
4.3.4 Κρυπτανάλυση του RSA 139
4.4 Συναρτήσεις Κατακερματισμού 140
4.4.1 Οι συναρτήσεις κατακερματισμού SHA 142
4.4.2 Η συνάρτηση κατακερματισμού SHA-256 143
4.4.3 Η χρήση συναρτήσεων κατακερματισμού για την παραγωγή συμμετρικών κλειδιών 151
4.4.4 Κρυπτανάλυση των συναρτήσεων κατακερματισμού 155
Κεφάλαιο 5 - Ασφαλής μετάδοση σύντομων μηνυμάτων 157
5.1 Java Micro Edition 157
5.2 Ανάλυση της αναπτυχθείσας εφαρμογής 162
5.2.1 Οι απαιτήσεις της εφαρμογής σε σχέση με το περιβάλλον εκτέλεσης 164
5.2.2 Η δημιουργία ενός διαύλου ασφαλούς επικοινωνίας 165
5.2.3 Οι επιλογές που παρέχονται στο χρήστη κατά την εκτέλεση της εφαρμογής 167
5.3 Αποτίμηση της παρεχόμενης ασφάλειας 205
5.3.1 Ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA 205
5.3.2 Ανταλλαγή κλειδιού για μια συνεδρία κρυπτογραφημένων μηνυμάτων 207
5.3.3 Ανταλλαγή κρυπτογραφημένων μηνυμάτων 208
5.3.4 Τελικό συμπέρασμα 209
5.4 Αποτελέσματα Εργασίας - Μελλοντική Έρευνα 210
Μέρος Δεύτερο 211
Κεφάλαιο 6 - Απομακρυσμένη πιστοποίηση ταυτότητας 212
6.1 Αυθεντικοποίηση δύο παραγόντων 212
6.2 Σκυτάλες ασφάλειας 213
6.3 Πρωτόκολλο για την απομακρυσμένη πιστοποίηση ταυτότητας 216
6.3.1 Η αναπτυχθείσα Διαδικτυακή Εφαρμογή 218
6.3.2 Ο ρόλος του GSM Modem 224
6.3.3 Η σκυτάλη ασφάλειας 226
6.3.4 Ο υπολογιστής του πελάτη 226
6.3.5 Τι απαιτείται πριν από τη συμμετοχή στο πρωτόκολλο; 227

6.3.6 Παρουσίαση του πρωτοκόλλου σε λειτουργία 238
6.4 Αποτίμηση της ασφάλειας που παρέχεται από το πρωτόκολλο 254
6.4.1 Η επικοινωνία στο δίκτυο κινητής τηλεφωνίας 254
6.4.2 Η επικοινωνία στο Διαδίκτυο 256
6.4.3 Τελικό συμπέρασμα 258
6.5 Αποτελέσματα Εργασίας – Μελλοντική Έρευνα 259
Παράρτημα 1 - Οδηγός Ρυθμίσεων 261
Εγκατάσταση του Java Standard Edition Development Kit 261
Εγκατάσταση του Integrated Development Environment NetBeans 263
Δημιουργία και εκτέλεση ενός MIDlet μέσω του NetBeans 266
Εγκατάσταση του Sony Ericsson SDK 272
Ενσωμάτωση του Sony Ericsson SDK στο IDE NetBeans 275
Εκτέλεση MIDlets μέσω του Sony Ericsson SDK 280
Εγκατάσταση του MySQL Server 283
Ρύθμιση του MySQL Server 286
Εγκατάσταση του Apache Tomcat Server 292
Ρύθμιση των συνδέσεων SSL-TLS 295
Εγκατάσταση του Java Communications API και του MySQL Connector/J 304
Προετοιμασία για την εκτέλεση της Διαδικτυακής εφαρμογής 305
Παράρτημα 2 - Πίνακας Συντομογραφιών 314
Παράρτημα 3 - Βιβλιογραφία 319
Παράρτημα 4 - Διαδικτυακές Πηγές 321
Παράρτημα 5 - Παρουσιάσεις PowerPoint 322
Παρουσίαση εφαρμογής για την ασφαλή ανταλλαγή σύντομων μηνυμάτων 322
Παρουσίαση πρωτοκόλλου για την απομακρυσμένη πιστοποίηση ταυτότητας ενός χρήστη 333
Παράρτημα 6 - Δημοσιεύσεις 344
Δημοσίευση της εφαρμογής για την ασφαλή ανταλλαγή μηνυμάτων 344
Δημοσίευση του πρωτοκόλλου για την απομακρυσμένη πιστοποίηση ταυτότητας ενός χρήστη 350

Πίνακας Εικόνων

Εικόνα 1: Η εξέλιξη των δικτύων κινητής τηλεφωνίας....8
Εικόνα 2: Μια τηλεφωνική συσκευή της δεκαετίας του 1960 σε σύγκριση με μια συσκευή του 2000....9
Εικόνα 3: Η οργάνωση των 3GPP και 3GPP2....17
Εικόνα 4: Η οικογένεια προτύπων του IMT-2000....18
Εικόνα 5: Η αληθινή όψη ενός δικτύου κυψέλης σε σύγκριση με τη φανταστική όψη....21
Εικόνα 6: Το μέγεθος μιας κυψέλης προσαρμόζεται ανάλογα με τις ανάγκες....22
Εικόνα 7: Επαναχρησιμοποίηση συχνοτήτων σε ένα δίκτυο κυψέλης....22
Εικόνα 8: Η αρχιτεκτονική του δικτύου GSM....23
Εικόνα 9: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Σταθμού Βάσης....24
Εικόνα 10: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Δικτύου και Μεταγωγής....25
Εικόνα 11: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Υποστήριξης Λειτουργίας....27
Εικόνα 12: Η αρχιτεκτονική του δικτύου GSM μετά από την ενσωμάτωση του GPRS....30
Εικόνα 13: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 99....33
Εικόνα 14: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 4....36
Εικόνα 15: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 5....38
Εικόνα 16: Το πρωτόκολλο μεταφοράς σύντομων μηνυμάτων....42
Εικόνα 17: Η πορεία από σημείο προς σημείο που ακολουθείται ένα μήνυμα MO-SM....44
Εικόνα 18: Η πορεία σημείο προς σημείο που ακολουθείται κατά τη μετάδοση ενός MT-SM μηνύματος....46
Εικόνα 19: Η δομή του πλαισίου ενός MO-SM μηνύματος....49
Εικόνα 20: Η δομή του πλαισίου ενός MT-SM μηνύματος....52
Εικόνα 21: Εύρεση του port που χρησιμοποιείται για την επικοινωνία ανάμεσα στο PC και το GSM modem....59
Εικόνα 22: Πλαίσιο διαλόγου για τη δημιουργία νέας σύνδεσης στο Microsoft HyperTerminal....60
Εικόνα 23: Πλαίσιο για την επιλογή της θύρας επικοινωνιών....60
Εικόνα 24: Εισαγωγή της εντολής "AT"....61
Εικόνα 25: Εισαγωγή της εντολής "AT+CPIN"....62
Εικόνα 26: Εισαγωγή της εντολής "AT+CMGF"....62
Εικόνα 27: Εισαγωγή της εντολής "AT+CSCA"....63
Εικόνα 28: Η δομή που έχει το πλαίσιο ενός SMS-SUBMIT μηνύματος....66
Εικόνα 29: Το αποτέλεσμα που προκύπτει από την εκτέλεση της εντολής "AT+CMGS"....70
Εικόνα 30: Η εμφάνιση της θύρας COM που χρησιμοποιείται από το GSM modem 72
Εικόνα 31: Η πρώτη οθόνη που συναντάμε κατά την εκτέλεση του προγράμματος PDUspy....73
Εικόνα 32: Το tab με τις ρυθμίσεις του προγράμματος PDUspy....74
Εικόνα 33: Το tab "Create" και η αλλαγή του τύπου του αριθμού του παραλήπτη....75
Εικόνα 34: Το αποτέλεσμα που εμφανίζεται στο tab "Decode"....76
Εικόνα 35: Η διαδικασία αυθεντικοποίησης ενός συνδρομητή του δικτύου GSM....80
Εικόνα 36: Η εκτέλεση του αλγορίθμου A3....81

Εικόνα 37: Η εκτέλεση του αλγορίθμου A8 82
Εικόνα 38: Η δημιουργία των τριάδων ασφάλειας του GSM στο περιβάλλον του οικείου δικτύου ενός συνδρομητή και η προώθηση τους στο δίκτυο εξυπηρέτησης .. 82
Εικόνα 39: Επιλογή μιας από τις τριάδες ασφάλειας που βρίσκονται στο MSC/VLR και προώθηση της τιμής RAND προς τον κινητό σταθμό του συνδρομητή. 83
Εικόνα 40: Η εκτέλεση των αλγορίθμων A3 και A8 στο περιβάλλον της κάρτας SIM 84
Εικόνα 41: Η ανανέωση της ταυτότητας TMSI όταν δεν υπάρχει μετακίνηση του συνδρομητή 86
Εικόνα 42: Η ανανέωση της ταυτότητας TMSI όταν ο συνδρομητής μετακινείται σε νέα περιοχή 86
Εικόνα 43: Η κρυπτογράφηση των επικοινωνιών στο GSM 87
Εικόνα 44: Η διαδικασία κρυπτογράφησης στο GSM 88
Εικόνα 45: Η λήψη του IMSI ενός συνδρομητή στέλνοντας ψευδείς πληροφορίες από ένα ψεύτικο σταθμό βάσης 92
Εικόνα 46: Ο προσδιορισμός του Ki μέσα από τη χρήση ενός ψεύτικου σταθμού βάσης και την πραγματοποίηση διαδοχικών αυθεντικοποιήσεων 93
Εικόνα 47: Η αυθεντικοποίηση ενός συνδρομητή στο δίκτυο UMTS 96
Εικόνα 48: Η εκτέλεση των πέντε συναρτήσεων ασφαλείας στο περιβάλλον του Auc/HSS 99
Εικόνα 49: Η δομή ενός διανύσματος αυθεντικοποίησης και του πεδίου AUTN 100
Εικόνα 50: Η εκτέλεση των πέντε συναρτήσεων ασφαλείας στο περιβάλλον της κάρτας USIM 101
Εικόνα 51: Η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης στο UTRAN 106
Εικόνα 52: Η διαδικασία με την οποία δημιουργείται ο κωδικός αυθεντικοποίησης MAC-I 108
Εικόνα 53: Σύστημα στο οποίο εφαρμόζεται συμμετρική κρυπτογράφηση 113
Εικόνα 54: Η μορφή των πινάκων εισόδου, εσωτερικής κατάστασης και εξόδου 117
Εικόνα 55: Εφαρμογή S-Box 117
Εικόνα 56: Ο μετασχηματισμός μέσω του οποίου προκύπτει η τιμή του κάθε bit του S-Box 118
Εικόνα 57: Εφαρμογή του μετασχηματισμού ShiftRows 120
Εικόνα 58: Εφαρμογή του μετασχηματισμού InvShiftRows 121
Εικόνα 59: Εφαρμογή του μετασχηματισμού MixColumns 121
Εικόνα 60: Το σταθερό πολυώνυμο που χρησιμοποιείται κατά το μετασχηματισμό MixColumns 122
Εικόνα 61: Το αποτέλεσμα του πολλαπλασιασμού των πινάκων κατά το μετασχηματισμό MixColumns 122
Εικόνα 62: Το σταθερό πολυώνυμο που χρησιμοποιείται στο μετασχηματισμό InvMixColumns 122
Εικόνα 63: Το αποτέλεσμα του πολλαπλασιασμού πινάκων κατά το μετασχηματισμό InvMixColumns 123
Εικόνα 64: Η εφαρμογή του μετασχηματισμού AddRoundKey 123
Εικόνα 65: Ολοκληρωμένος κύκλος εκτέλεσης του αλγορίθμου AES 127
Εικόνα 66: Η κρυπτογράφηση ενός μηνύματος με έναν αλγόριθμο δημοσίου κλειδιού 131
Εικόνα 67: Η επαλήθευση μιας ηλεκτρονικής υπογραφής 132
Εικόνα 68: Η δημιουργία και η επαλήθευση μιας ψηφιακής υπογραφής 142
Εικόνα 69: Η προσθήκη message padding 144
Εικόνα 70: Μια επανάληψη της διαδικασίας συμπίεσης της SHA-256 147

Εικόνα 71: Η οικογένεια της Java 157
Εικόνα 72: Η αρχιτεκτονική J2ME 159
Εικόνα 73: Η σχέση κληρονομικότητας ανάμεσα σε J2SE και J2ME 160
Εικόνα 74: Η οργάνωση μιας σουίτας MIDlet 161
Εικόνα 75: Ο κύκλος εκτέλεσης ενός MIDlet 162
Εικόνα 76: Η στοίβα του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων 163
Εικόνα 77: Το πρώτο βήμα, όπου εγκαθίσταται η επικοινωνία μέσω του RSA 166
Εικόνα 78: Το δεύτερο βήμα, όπου ανταλλάσσεται το session key που θα χρησιμοποιηθεί 166
Εικόνα 79: Το τρίτο βήμα, όπου πραγματοποιείται η ασφαλής ανταλλαγή σύντομων μηνυμάτων 167
Εικόνα 80: Το μενού όπου επιλέγεται η εφαρμογή που θα αρχίσει την εκτέλεση της 169
Εικόνα 81: Η μορφή που έχει το εσωτερικό ενός Record Store 170
Εικόνα 82: Η σχέση που υπάρχει ανάμεσα σε MIDlet Suite και RecordStore 170
Εικόνα 83: Παράδειγμα εγγραφής στο Record Store contacts 170
Εικόνα 84: Προειδοποίηση σχετικά με το ζεύγος των κλειδιών για τον αλγόριθμο RSA και σύσταση για τη δημιουργία ενός νέου ζεύγους 172
Εικόνα 85: Η εγγραφή που πραγματοποιείται στο Record Store myKeyPair 173
Εικόνα 86: Η ένδειξη της προόδου που σημειώνεται κατά τη δημιουργία των κλειδιών RSA 174
Εικόνα 87: Το κεντρικό μενού της εφαρμογής, όπου επιλέγεται η διαχείριση των επαφών 176
Εικόνα 88: Το κεντρικό μενού που παρέχεται από τη διαχείριση επαφών 177
Εικόνα 89: Εισαγωγή ονόματος και τηλεφωνικού αριθμού κατά τη διάρκεια της προσθήκης μιας νέας επαφής 178
Εικόνα 90: Το κεντρικό μενού της εφαρμογής και η επιλογή για την αποστολή του δημοσίου κλειδιού 180
Εικόνα 91: Η λίστα με τις καταχωρημένες επαφές στις οποίες μπορεί να σταλεί το δημόσιο κλειδί 181
Εικόνα 92: Λήψη του δημοσίου κλειδιού από μια μη καταχωρημένη επαφή 183
Εικόνα 93: Η επιλογή για τη διαχείριση επαφών 184
Εικόνα 94: Η επιλογή για την τροποποίηση μιας επαφής 185
Εικόνα 95: Η λίστα όπου παρουσιάζεται μία μη καταχωρημένη επαφή 186
Εικόνα 96: Η τροποποίηση μιας επαφής 187
Εικόνα 97: Το κεντρικό μενού της εφαρμογής και η επιλογή για την αποστολή των στοιχείων που δημιουργούν το session key 188
Εικόνα 98: Η συμπλήρωση των πεδίων με το password και την τιμή salt 189
Εικόνα 99: Αποστολή των password και salt 191
Εικόνα 100: Ανάκτηση των password και salt 192
Εικόνα 101: Επιλογή για σύνταξη ενός κρυπτογραφημένου μηνύματος 194
Εικόνα 102: Εισαγωγή των password και salt από τα οποία δημιουργείται το session key που θα χρησιμοποιηθεί 195
Εικόνα 103: Πληκτρολόγηση του αρχικού μηνύματος 196
Εικόνα 104: Επιλογή του χρήστη για τον οποίο προορίζεται το κρυπτογραφημένο μήνυμα 197
Εικόνα 105: Ειδοποίηση για τη λήψη ενός νέου κρυπτογραφημένου μηνύματος 200
Εικόνα 106: Εισαγωγή των παραμέτρων που θα δημιουργήσουν το session key 201
Εικόνα 107: Ανάκτηση του αρχικού μηνύματος 202

Εικόνα 108: Το κεντρικό μενού της εφαρμογής, όπου διακρίνεται η επιλογή της εξόδου 204
Εικόνα 109: Η ροή δεδομένων κατά την ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA 206
Εικόνα 110: Η ροή δεδομένων κατά την ανταλλαγή κλειδιού μιας συνεδρίας μηνυμάτων 207
Εικόνα 111: Η ανταλλαγή κρυπτογραφημένων μηνυμάτων ανάμεσα στις δύο επικοινωνούσες πλευρές 209
Εικόνα 112: Διάφορες υλοποιήσεις μιας σκυτάλης ασφάλειας 214
Εικόνα 113: Συνοπτική παρουσίαση του πρωτοκόλλου 217
Εικόνα 114: Η αρχιτεκτονική ενός three-tiered application σύμφωνα με την πλατφόρμα Java EE 218
Εικόνα 115: Η προσθήκη του πρωτοκόλλου SSL/TLS στη στοίβα των πρωτοκόλλων TCP/IP 221
Εικόνα 116: Ενέργειες που πραγματοποιούνται από την πλευρά του πελάτη 228
Εικόνα 117: Η αρχική σελίδα της διαδικτυακής εφαρμογής 229
Εικόνα 118: Σελίδα εγγραφής στη διαδικτυακή υπηρεσία 230
Εικόνα 119: Η εισαγωγή των στοιχείων ενός νέου πελάτη στη βάση δεδομένων 231
Εικόνα 120: Εμφάνιση της εγγραφής ενός πελάτη στη βάση δεδομένων 231
Εικόνα 121: Λήψη του MIDlet αμέσως μετά την εγγραφή του πελάτη 232
Εικόνα 122: Login πριν από τη λήψη του MIDlet 233
Εικόνα 123: Έλεγχος των στοιχείων που εισάγει ο πελάτης πριν από τη λήψη του MIDlet 233
Εικόνα 124: Λήψη MIDlet μετά από login 234
Εικόνα 125: Οθόνη από την πρώτη εκτέλεση του MIDlet 235
Εικόνα 126: Οθόνη εισαγωγής των στοιχείων με τα οποία θα γίνεται login στο MIDlet 235
Εικόνα 127: Τιμή hash που παράγεται από τα στοιχεία του login 235
Εικόνα 128: Οθόνη αποστολής του παραγόμενου hash προς την πλευρά του εξυπηρετητή 236
Εικόνα 129: Η κρυπτογράφηση του hash που παράχθηκε από το MIDlet 236
Εικόνα 130: Εκτέλεση εντολής AT+CMGL για την εμφάνιση νέων μηνυμάτων 237
Εικόνα 131: Η αποθήκευση της τιμής hash που δημιουργήθηκε στο κινητό τηλέφωνο του πελάτη 238
Εικόνα 132: Εμφάνιση της ανανεωμένης εγγραφής ενός πελάτη στη βάση δεδομένων 238
Εικόνα 133: Οι ενέργειες που περιλαμβάνονται στο πρωτόκολλο πιστοποίησης ταυτότητας 239
Εικόνα 134: Η σελίδα όπου πραγματοποιείται το login στη διαδικτυακή υπηρεσία 240
Εικόνα 135: Οθόνη για Login στο MIDlet που βρίσκεται στο κινητό τηλέφωνο του πελάτη 241
Εικόνα 136: Περιεχόμενα της κονσόλας του NetBeans κατά το login στο MIDlet 241
Εικόνα 137: Επαλήθευση του username και του password που χρησιμοποιήθηκαν σε ένα login 242
Εικόνα 138: Σελίδα που εμφανίζεται έπειτα από επιτυχημένο login 242
Εικόνα 139: Εκτύπωση των στοιχείων του πελάτη στην κονσόλα του Apache Tomcat 243
Εικόνα 140: Επιβεβαίωση της εισαγωγής λανθασμένων στοιχείων από την πλευρά του πελάτη 243
Εικόνα 141: Η σελίδα που εμφανίζεται στην περίπτωση ενός αποτυχημένου login. 244

Εικόνα 142: Η κονσόλα του Apache Tomcat κατά τη δημιουργία και την αποστολή ενός SMS που περιέχει ένα Challenge....245
Εικόνα 143: Οθόνη ειδοποίησης για τη λήψη ενός νέου μηνύματος....246
Εικόνα 144: Η δημιουργία του Response όπως εμφανίζεται στην κονσόλα του IDE NetBeans....247
Εικόνα 145: Εμφάνιση του συμπιεσμένου Response με μήκος 8 χαρακτήρων στην οθόνη του κινητού τηλεφώνου του πελάτη....248
Εικόνα 146: Η ιστοσελίδα που εμφανίζεται στην πλευρά ενός πελάτη που έχει περάσει με επιτυχία το στάδιο του login....249
Εικόνα 147: Αύξηση του μετρητή που διατηρείται από την πλευρά του εξυπηρετητή κατά μια μονάδα....249
Εικόνα 148: Η κονσόλα του Apache Tomcat κατά τη δημιουργία του Response από την πλευρά του εξυπηρετητή και τη σύγκριση με το Response που έχει υποβάλλει η πλευρά του πελάτη....250
Εικόνα 149: Η ιστοσελίδα με εμπιστευτικές πληροφορίες που εμφανίζεται σε έναν πελάτη που μόλις πέρασε με επιτυχία και το δεύτερο παράγοντα αυθεντικοποίησης....251
Εικόνα 150: Η σελίδα που εμφανίζεται στην πλευρά του χρήστη όταν έχουμε μιαν αποτυχημένη προσπάθεια πιστοποίησης ταυτότητας.....252
Εικόνα 151: Κεντρική οθόνη του MIDlet, όπου παρέχεται η δυνατότητα αποστολής ενός SMS συγχρονισμού....253
Εικόνα 152: Η κονσόλα του Apache Tomcat όταν πραγματοποιείται η ανανέωση του μετρητή του πελάτη....253
Εικόνα 153: Άδεια χρήσης του Java Standard Edition Development Kit....262
Εικόνα 154: Επιλογή στοιχείων και τοποθεσίας JDK....262
Εικόνα 155: Επιλογή τοποθεσίας εγκατάστασης του Java Runtime Environment....263
Εικόνα 156: Ολοκλήρωση της εγκατάστασης του JDK....263
Εικόνα 157: Εμφάνιση πακέτων που θα εγκατασταθούν με το NetBeans....264
Εικόνα 158: Επιλογή των πακέτων που θα εγκατασταθούν με το NetBeans....265
Εικόνα 159: Άδεια χρήσης του περιβάλλοντος NetBeans....265
Εικόνα 160: Επιλογή φακέλων εγκατάστασης....266
Εικόνα 161: Ολοκλήρωση εγκατάστασης NetBeans....266
Εικόνα 162: Άνοιγμα του project από το menu του NetBeans IDE....267
Εικόνα 163: Επιλογή του project μέσω του παρεχόμενου file chooser....267
Εικόνα 164: Εμφάνιση του πηγαίου κώδικα του MIDlet....268
Εικόνα 165: Επιλογή για Clean and Build....268
Εικόνα 166: Η επιλογή για την εκτέλεση του project....268
Εικόνα 167: Εκκίνηση του emulator....269
Εικόνα 168: Επιλογή των properties του project....269
Εικόνα 169: Προσθήκη νέου Configuration....270
Εικόνα 170: Επιλογή του κατάλληλου template....271
Εικόνα 171: Επιλογή CLDC, MIDP και προαιρετικών πακέτων του Configuration 272
Εικόνα 172: Ενημέρωση του χρήστη σχετικά με το Java Development Kit που πρόκειται να χρησιμοποιηθεί....273
Εικόνα 173: Οθόνη υποδοχής του χρήστη στο Sony Ericsson SDK....273
Εικόνα 174: Επιλογή των στοιχείων που πρόκειται να εγκατασταθούν από το Sony Ericsson SDK....274
Εικόνα 175: Επιλογή του φακέλου στον οποίο προορίζονται να εγκατασταθούν τα στοιχεία του Sony Ericsson SDK....274

Εικόνα 176: Οθόνη με την οποία ολοκληρώνεται η εγκατάσταση του Sony Ericsson SDK....275
Εικόνα 177: Το πρώτο βήμα για την ενσωμάτωση μιας πλατφόρμας Java στο IDE NetBeans....275
Εικόνα 178: Η διαχείριση των Java Platforms που βρίσκονται ενσωματωμένες στο NetBeans....276
Εικόνα 179: Επιλογή του τύπου της πλατφόρμας που πρόκειται να ενσωματωθεί στο NetBeans....277
Εικόνα 180: Η επιλογή των φακέλων που πρόκειται να αναζητηθούν νέες πλατφόρμες....278
Εικόνα 181: Εμφάνιση των πλατφόρμων που ανιχνεύθηκαν από το NetBeans....279
Εικόνα 182: Η διαχείριση των πλατφόρμων Java, όπου τώρα εμφανίζονται οι νέες πλατφόρμες που εγκαταστήσαμε....280
Εικόνα 183: Εντοπισμός του προγράμματος KToolbar....281
Εικόνα 184: Η κονσόλα που εμφανίζεται από το πρόγραμμα KToolbar....281
Εικόνα 185: Η επιλογή για τη δημιουργία ενός project από JAD και JAR που ήδη υπάρχουν....282
Εικόνα 186: Η επιλογή του JAD από το οποίο θα δημιουργηθεί το project....282
Εικόνα 187: Η επιλογή της συσκευής που πρόκειται να χρησιμοποιηθεί κατά την προσομοίωση του MIDlet....283
Εικόνα 188: Καλωσόρισμα στην εγκατάσταση του MySQL Server....283
Εικόνα 189: Επιλογή του τύπου εγκατάστασης του MySQL Server....284
Εικόνα 190: Εμφάνιση φακέλων με τους οποίους σχετίζεται ο MySQL Server....284
Εικόνα 191: Το πρώτο διαφημιστικό παράθυρο της MySQL....285
Εικόνα 192: Το δεύτερο διαφημιστικό παράθυρο της MySQL....285
Εικόνα 193: Ολοκλήρωση της εγκατάστασης του MySQL Server....286
Εικόνα 194: Καλωσόρισμα στον οδηγό ρύθμισης του MySQL Server....286
Εικόνα 195: Επιλογή του τύπου ρύθμισης του MySQL Server....287
Εικόνα 196: Επιλογή του τύπου του MySQL Server....287
Εικόνα 197: Επιλογή της χρήσης που θα έχουν οι βάσεις δεδομένων....288
Εικόνα 198: Επιλογή της τοποθεσίας αποθήκευσης των δεδομένων του InnoDB....288
Εικόνα 199: Επιλογή του αριθμού των ταυτόχρονων συνδέσεων....289
Εικόνα 200: Ρύθμιση των επιλογών που αφορούν τη σύνδεση με το MySQL Server....289
Εικόνα 201: Επιλογή του default character set....290
Εικόνα 202: Εγκατάσταση του MySQL Server σαν Windows Service και προσθήκη φακέλου bin στη μεταβλητή PATH....290
Εικόνα 203: Δημιουργία χρήστη root ή ανώνυμου χρήστη....291
Εικόνα 204: Παράθυρο για τη δημιουργία νέου configuration file....291
Εικόνα 205: Επιτυχημένη δημιουργία νέου configuration file....292
Εικόνα 206: Το tab Advanced όπου βρίσκεται η διαχείριση των μεταβλητών του λειτουργικού συστήματος....293
Εικόνα 207: Μετά από τον ορισμό των τριών μεταβλητών....294
Εικόνα 208: Η σελίδα που μας καλωσορίζει στον Apache Tomcat....295
Εικόνα 209: Η δημιουργία ενός ψηφιακού πιστοποιητικού μέσω του keytool.exe....297
Εικόνα 210: Η οθόνη που εμφανίζεται από το Mozilla Firefox σε ένα μη έμπιστο ψηφιακό πιστοποιητικό....299
Εικόνα 211: Η προσθήκη εξαίρεσης για ένα μη έμπιστο ψηφιακό πιστοποιητικό....300
Εικόνα 212: Η εμφάνιση του tab που περιέχει γενικά στοιχεία σχετικά με το ψηφιακό πιστοποιητικό....301

Εικόνα 213: Εμφάνιση του tab με τα ειδικά στοιχεία του χρησιμοποιούμενου ψηφιακού πιστοποιητικού....302
Εικόνα 214: Πατώντας στο χρωματιστό τμήμα της διεύθυνσης αποκτούμε πρόσβαση σε περισσότερο τεχνικές πληροφορίες....303
Εικόνα 215: Παράθυρο που εμφανίζεται με τεχνικές πληροφορίες....303
Εικόνα 216: Η διαχείριση των ψηφιακών πιστοποιητικών μέσα από τον browser...304
Εικόνα 217: Εμφάνιση του φακέλου webapps του Apache Tomcat....306
Εικόνα 218: Προβολή των εγκατεστημένων modems και των θυρών όπου συνδέονται....307
Εικόνα 219: Ενδεικτική επικοινωνία μέσω terminal emulator με το GSM Modem.308
Εικόνα 220: Εντοπισμός της γραμμής κώδικα όπου εισάγεται το τηλέφωνο του GSM Modem που λαμβάνει μηνύματα SMS....309
Εικόνα 221: Το αρχείο κειμένου που περιέχει το δημόσιο κλειδί που έχει δημιουργηθεί από τη διαδικτυακή εφαρμογή....310
Εικόνα 222: Αντιγραφή του δημοσίου κλειδιού μέσα από έναν επεξεργαστή αρχείων κειμένου....310
Εικόνα 223: Εντοπισμός της γραμμής κώδικα όπου εισάγεται το δημόσιο κλειδί...310
Εικόνα 224: Τα παραγόμενα JAR και JAD μέσα από τα projects του IDE NetBeans....310
Εικόνα 225: Εκκίνηση του Apache Tomcat από το command-line....311
Εικόνα 226: Εκκίνηση του Apache Tomcat μέσω του Windows Explorer....312
Εικόνα 227: Η κονσόλα που εμφανίζεται κατά την εκκίνηση του Apache Tomcat..313

Πίνακας Πινάκων

Πίνακας 1: Η οργάνωση χαρακτήρων μεγέθους 7 δυαδικών ψηφίων σε οκτάδες bit.66
Πίνακας 2: Οι δυαδικές τιμές που εισάγονται στα πεδία της πρώτης οκτάδας του πλαισίου TPDU.....67
Πίνακας 3: Παρουσίαση των δεκαεξαδικών τιμών που εισάγονται στα πεδία του πλαισίου TPDU.....69
Πίνακας 4: Η σχέση ανάμεσα στο μέγεθος του κλειδιού και τις παραμέτρους του αλγορίθμου.....116
Πίνακας 5: Οι τιμές του πίνακα S-Box.....118
Πίνακας 6: Οι τιμές του αντιστρόφου του πίνακα S-Box.....119
Πίνακας 7: Ο πολλαπλασιασμός των πινάκων που πραγματοποιείται κατά τον μετασχηματισμό MixColumns.....122
Πίνακας 8: Ο πολλαπλασιασμός των πινάκων που πραγματοποιείται κατά τον μετασχηματισμό InvMixColumns.....123
Πίνακας 9: Οι δυνάμεις του byte 02 στο πεδίο GF(2^8).....125
Πίνακας 10: Ένα απόσπασμα από τη διαδικασία Key Expansion Schedule.....126
Πίνακας 11: Αλγόριθμοι δημοσίου κλειδιού και οι εφαρμογές που παρέχουν.....130
Πίνακας 12: Σύγκριση ανάμεσα στο μήκος του δημοσίου κλειδιού και τον απαιτούμενο χρόνο για να δημιουργηθεί το ζεύγος κλειδιών RSA.....175
Πίνακας 13: Η δημιουργία του συμπιεσμένου Response.....248

Εισαγωγή

Γενική Ιδέα

Τα τελευταία χρόνια πραγματοποιείται η σύγκλιση δύο τομέων που έχουν σημειώσει σημαντική πρόοδο και εξέλιξη, πρόκειται για τους τομείς της πληροφορικής και των τηλεπικοινωνιών.

Το αποτέλεσμα της σύγκλισης είναι η δημιουργία πολύ-λειτουργικών συσκευών που συνδυάζουν τις δυνατότητες ενός κινητού τηλεφώνου και ενός φορητού υπολογιστή, συνδέονται σε δίκτυα δεδομένων και λαμβάνουν πληθώρα υπηρεσιών.

Χάρις στις συγκεκριμένες συσκευές έχει αλλάξει ριζικά και έχει απλουστευθεί ο τρόπος με τον οποίο διεξάγονται κάποιες καθημερινές μας δραστηριότητες, όπως είναι η επικοινωνία με συνανθρώπους μας, οι εμπορικές συναλλαγές και η ψυχαγωγία.

Σε κάποιες από αυτές τις δραστηριότητες, όπως λόγου χάρη στις εμπορικές συναλλαγές, χρησιμοποιούνται εμπιστευτικά δεδομένα. Το γεγονός αυτό καθιστά αρκετά σημαντική την προστασία των δεδομένων και κατ' επέκταση την ασφάλεια όσων εμπλέκονται με αυτές τις δραστηριότητες.

Σκοπός της εργασίας

Η εργασία που πραγματοποιήθηκε έχει σαν κύριο στόχο την ανάδειξη προβλημάτων που σχετίζονται με την ασφάλεια των κινητών τηλεπικοινωνιών και την παροχή μιας σειράς λύσεων, μέσω των οποίων ενισχύεται η ασφάλεια των χρηστών.

Για τις ανάγκες της παρούσης πτυχιακής εργασίας υλοποιούνται τα ακόλουθα δύο σενάρια:

- Μια P2P εφαρμογή για την ασφαλή ανταλλαγή μηνυμάτων πάνω από ένα ασύρματο δίκτυο. Η εν λόγω εφαρμογή εκτελείται σε συσκευές κινητών τηλεφώνων παρέχοντας έτσι μια λύση end-to-end encryption.

- Η πιστοποίηση της ταυτότητας του χρήστη ενός Web Service μέσω ενός One-Time Password που θα αποστέλλεται με SMS από την πλευρά του Server που φιλοξενεί το Web Service, με αυτό τον τρόπο το κινητό τηλέφωνο του χρήστη μετατρέπεται σε Authentication Token.

Προκειμένου να επιτευχθούν τα δύο προαναφερθέντα σενάρια έπρεπε να προηγηθεί μια σειρά από ενέργειες. Έτσι, στην περίπτωση της εφαρμογής ασφαλούς ανταλλαγής μηνυμάτων έπρεπε να γίνουν τα ακόλουθα:

- Μελέτη του τρόπου λειτουργίας της υπηρεσίας μετάδοσης σύντομων μηνυμάτων που συναντάμε σε δίκτυα κινητής τηλεφωνίας.

- Μελέτη της αρχιτεκτονικής της πλατφόρμας Java Micro Edition και εξοικείωση με τη δημιουργία των εφαρμογών τύπου MIDlet που εκτελούνται σε αυτήν.

- Μελέτη της λειτουργίας των threads ώστε το αναπτυχθέν MIDlet να μπορεί να στέλνει και να δέχεται μηνύματα ταυτόχρονα.

- Μελέτη των κρυπτογραφικών αλγορίθμων που χρησιμοποιήθηκαν και υλοποίηση τους σε κλάσεις Java. Πιο συγκεκριμένα, χρησιμοποιήθηκαν:
    - Ο συμμετρικός αλγόριθμος AES, με τον οποίο κρυπτογραφείται το περιεχόμενο των μηνυμάτων.
    - Η οικογένεια των συναρτήσεων κατακερματισμού SHA-2, οι οποίες χρησιμοποιούνται σύμφωνα με το πρότυπο PBKDF2 ως γεννήτριες συμμετρικών κλειδιών για τον αλγόριθμο AES.
    - Ο αλγόριθμος δημοσίου κλειδιού RSA, με τον οποίο κρυπτογραφούνται τα κλειδιά που χρησιμοποιούνται σε μια συνεδρία ανταλλαγής μηνυμάτων.

- Επίδειξη της εφαρμογής τόσο σε emulator κινητού τηλεφώνου όσο και σε πραγματική συσκευή που υποστηρίζει J2ME (κινητό τηλέφωνο, PDA, κλπ)

Για την υλοποίηση του πρωτόκολλου για την απομακρυσμένη πιστοποίηση της ταυτότητας ενός χρήστη απαιτήθηκαν οι ακόλουθες ενέργειες:

- Εξέταση των μεθόδων με τις οποίες μπορεί να αποσταλεί ένα SMS από ένα PC και η λειτουργία ενός κινητού τηλεφώνου ως GSM Modem.

- Μελέτη του τρόπου με τον οποίο πραγματοποιείται η αυθεντικοποίηση δύο παραγόντων και του τρόπου με τον οποίον λειτουργούν οι σκυτάλες ασφαλείας.

- Μελέτη της αρχιτεκτονικής της πλατφόρμας Java Enterprise Edition και εξοικείωση με τη δημιουργία κλάσεων Servlet και σελίδων JSP που υποστηρίζονται από τη συγκεκριμένη πλατφόρμα.

- Μελέτη των ασφαλών συνδέσεων TLS.

- Ανάπτυξη three-tier διαδικτυακής εφαρμογής που προσομοιώνει τη λειτουργία ενός Web Service που παρέχει εμπιστευτικές πληροφορίες.

- Ανάπτυξη ενός MIDlet με το οποίο το κινητό τηλέφωνο του χρήστη μετατρέπεται σε σκυτάλη ασφαλείας.

Συνοπτική περιγραφή

Στο πρώτο κεφάλαιο που συναντάμε, το οποίο όμως δεν συμμετέχει στη γενικότερη αρίθμηση, φιλοξενείται η εισαγωγή στην πτυχιακή εργασία, κατά την οποία αναλύεται ο σκοπός που εξυπηρετείται από αυτήν, ενώ παράλληλα παρουσιάζεται ο τρόπος με τον οποίο δομείται η αναφορά που έχει συνταχθεί.

Μετά από την εισαγωγή περνάμε στο πρώτο μέρος της αναφοράς, το οποίο αποτελείται από πέντε κεφάλαια.

Στο κεφάλαιο 1 πραγματοποιείται ιστορική αναδρομή των δικτύων κινητής τηλεφωνίας, διεξάγεται ανάλυση της αρχιτεκτονικής των πιο διαδεδομένων δικτύων (GSM, GSM+GPRS, UMTS) και αναφέρονται τα κυριότερα χαρακτηριστικά των δικτύων της $4^{ης}$ γενιάς κινητής τηλεφωνίας.

Το κεφάλαιο 2 πραγματεύεται την παροχή της υπηρεσίας σύντομων μηνυμάτων, έτσι σε αυτό, έχουμε την παρουσίαση των τεχνικών χαρακτηριστικών των μηνυμάτων SMS, την παρουσίαση των εντολών AT και ενός παραδείγματος χρήσης τους, κατά το οποίο αποστέλλουμε ένα SMS μέσω ενός PC που συνδέεται με ένα GSM modem.

Στο κεφάλαιο 3 μελετώνται οι μηχανισμοί ασφαλείας των δικτύων GSM και UMTS, παρουσιάζονται όσα τρωτά σημεία έχουν εντοπιστεί και αναφέρονται περιπτώσεις όπου είχαμε παραβιάσεις στην ασφάλεια των συγκεκριμένων δικτύων.

Στο κεφάλαιο 4 παρουσιάζονται όσα κρυπτογραφικά στοιχεία εμπλέκονται στη δημιουργία του πρακτικού μέρους της πτυχιακής εργασίας. Μεταξύ άλλων, αναλύονται:

- η συμμετρική κρυπτογραφία και ο αλγόριθμος AES
- η κρυπτογραφία δημοσίου κλειδιού και ο αλγόριθμος RSA
- οι συναρτήσεις κατακερματισμού SHA και η λειτουργία της SHA-256
- η διαδικασία παραγωγής κωδικών μέσω της $2^{ης}$ έκδοσης της μεθόδου Password Based Key Derivation Function

Στο κεφάλαιο 5 παρουσιάζεται το περιβάλλον Java Micro Edition, στο οποίο εκτελείται η εφαρμογή που αναπτύχθηκε. Παράλληλα, μέσα από εκτενή χρήση screenshots που ελήφθησαν από έναν Java Micro Edition Emulator, παρουσιάζεται αναλυτικά ο τρόπος με τον οποίο λειτουργεί η εφαρμογή που επιτρέπει την ασφαλή μετάδοση σύντομων μηνυμάτων.

Στη συνέχεια περνάμε στο δεύτερο μέρος της αναφοράς, όπου περιλαμβάνεται ένα κεφάλαιο, πιο συγκεκριμένα το κεφάλαιο 6, όπου παρουσιάζεται το περιβάλλον Java Enterprise Edition και τα στοιχεία που χρησιμοποιήθηκαν για την υλοποίηση του εναλλακτικού σεναρίου, όπου η εφαρμογή ασφαλούς ανταλλαγής μηνυμάτων τροποποιείται κατάλληλα, συνεργάζεται με μια three-tier web εφαρμογή, και χρησιμοποιείται για την πιστοποίηση της ταυτότητας ενός απομακρυσμένου χρήστη.

Στο τέλος της αναφοράς συναντούμε μια σειρά από παραρτήματα που σχετίζονται με τα διάφορα θέματα και φέρουν τους ακόλουθους τίτλους:

- Οδηγός ρυθμίσεων, όπου αναφέρεται τι πρέπει να γίνει για να εκτελεσθούν όσες εφαρμογές αναπτύχθηκαν κατά το πρακτικό μέρος της πτυχιακής εργασίας.
- Πίνακας Συντομογραφιών, όπου αναλύονται τα δεκάδες αρκτικόλεξα που χρησιμοποιούνται κατά κόρον σε όλη την πτυχιακή.
- Βιβλιογραφία, όπου καταγράφονται τα βιβλία και οι επιστημονικές δημοσιεύσεις που συνετέλεσαν στη διαμόρφωση του θεωρητικού υπόβαθρου της πτυχιακής.
- Διαδικτυακές πηγές, απ' όπου αντλήθηκαν επιπρόσθετες πληροφορίες σχετικά με τα στοιχεία από τα οποία δομείται η εργασία.
- Παρουσιάσεις PowerPoint, όπου έχουν επισυναφθεί αντίγραφα των παρουσιάσεων που χρησιμοποιήθηκαν κατά την παρουσίαση και αξιολόγηση της πτυχιακής εργασίας.
- Πρότυπα Δημοσιεύσεων, στα οποία περιέχονται ολιγοσέλιδες περιλήψεις των σεναρίων που μελετήθηκαν και υλοποιήθηκαν για τις ανάγκες της παρούσης πτυχιακής εργασίας.

Διάρθρωση της εργασίας

Η εργασία έχει την ακόλουθη δομή:

Αριθμός κεφαλαίου	Τίτλος
	Εισαγωγή
Μέρος Πρώτο	
1	Δίκτυα κινητής τηλεφωνίας
2	Υπηρεσία σύντομων μηνυμάτων
3	Η ασφάλεια των δικτύων κινητής τηλεφωνίας
4	Κρυπτογραφία
5	Ασφαλής μετάδοση σύντομων μηνυμάτων

Μέρος Δεύτερο	
6	Απομακρυσμένη πιστοποίηση ταυτότητας
Παράρτημα 1	Οδηγός για την εκτέλεση των αναπτυχθέντων εφαρμογών
Παράρτημα 2	Πίνακας συντομογραφιών
Παράρτημα 3	Βιβλιογραφία
Παράρτημα 4	Διαδικτυακές Πηγές
Παράρτημα 5	Παρουσιάσεις PowerPoint
Παράρτημα 6	Δημοσιεύσεις

Μέρος Πρώτο

Το πρώτο μέρος της τρέχουσας αναφοράς αποτελείται από πέντε κεφάλαια και σχετίζεται με την ασφαλή ανταλλαγή σύντομων μηνυμάτων πάνω από ένα ασύρματο ομότιμο δίκτυο.

Στα τέσσερα πρώτα, εκ των πέντε κεφαλαίων, παρουσιάζεται το θεωρητικό υπόβαθρο στο οποίο βασίστηκε η υλοποιηθείσα εφαρμογή. Έτσι λοιπόν, συνοπτικά αναφέρουμε τι περιλαμβάνεται σε αυτά:

- Στο κεφάλαιο 1 αναλύεται η αρχιτεκτονική των πιο διαδεδομένων δικτύων κινητής τηλεφωνίας.
- Στο κεφάλαιο 2 παρουσιάζονται τα τεχνικά χαρακτηριστικά των μηνυμάτων SMS μαζί με τον τρόπο που χρησιμοποιούνται οι εντολές AT όταν επιθυμούμε να αποστείλουμε ένα σύντομο μήνυμα μέσω ενός PC που συνδέεται με ένα GSM modem.
- Στο κεφάλαιο 3 μελετώνται οι μηχανισμοί ασφαλείας των δικτύων GSM και UMTS, παρουσιάζονται όσα τρωτά σημεία έχουν εντοπιστεί και αναφέρονται περιπτώσεις όπου είχαμε παραβιάσεις στην ασφάλεια των συγκεκριμένων δικτύων.
- Στο κεφάλαιο 4 παρουσιάζονται όσα κρυπτογραφικά στοιχεία εμπλέκονται στη δημιουργία του πρακτικού μέρους της πτυχιακής εργασίας. Μεταξύ άλλων, αναλύονται:
    - η συμμετρική κρυπτογραφία και ο αλγόριθμος AES
    - η κρυπτογραφία δημοσίου κλειδιού και ο αλγόριθμος RSA
    - οι συναρτήσεις κατακερματισμού SHA και η λειτουργία της SHA-256
    - η διαδικασία παραγωγής κωδικών μέσω της $2^{ης}$ έκδοσης της μεθόδου Password Based Key Derivation Function

Στο εναπομένον κεφάλαιο 5 παρουσιάζεται το περιβάλλον Java Micro Edition, στο οποίο εκτελείται η εφαρμογή που αναπτύχθηκε. Παράλληλα, μέσα από εκτενή χρήση screenshots που ελήφθησαν από έναν Java Micro Edition Emulator, παρουσιάζεται αναλυτικά ο τρόπος με τον οποίο λειτουργεί η εφαρμογή που επιτρέπει την ασφαλή μετάδοση σύντομων μηνυμάτων.

Κεφάλαιο 1 - Δίκτυα κινητής τηλεφωνίας

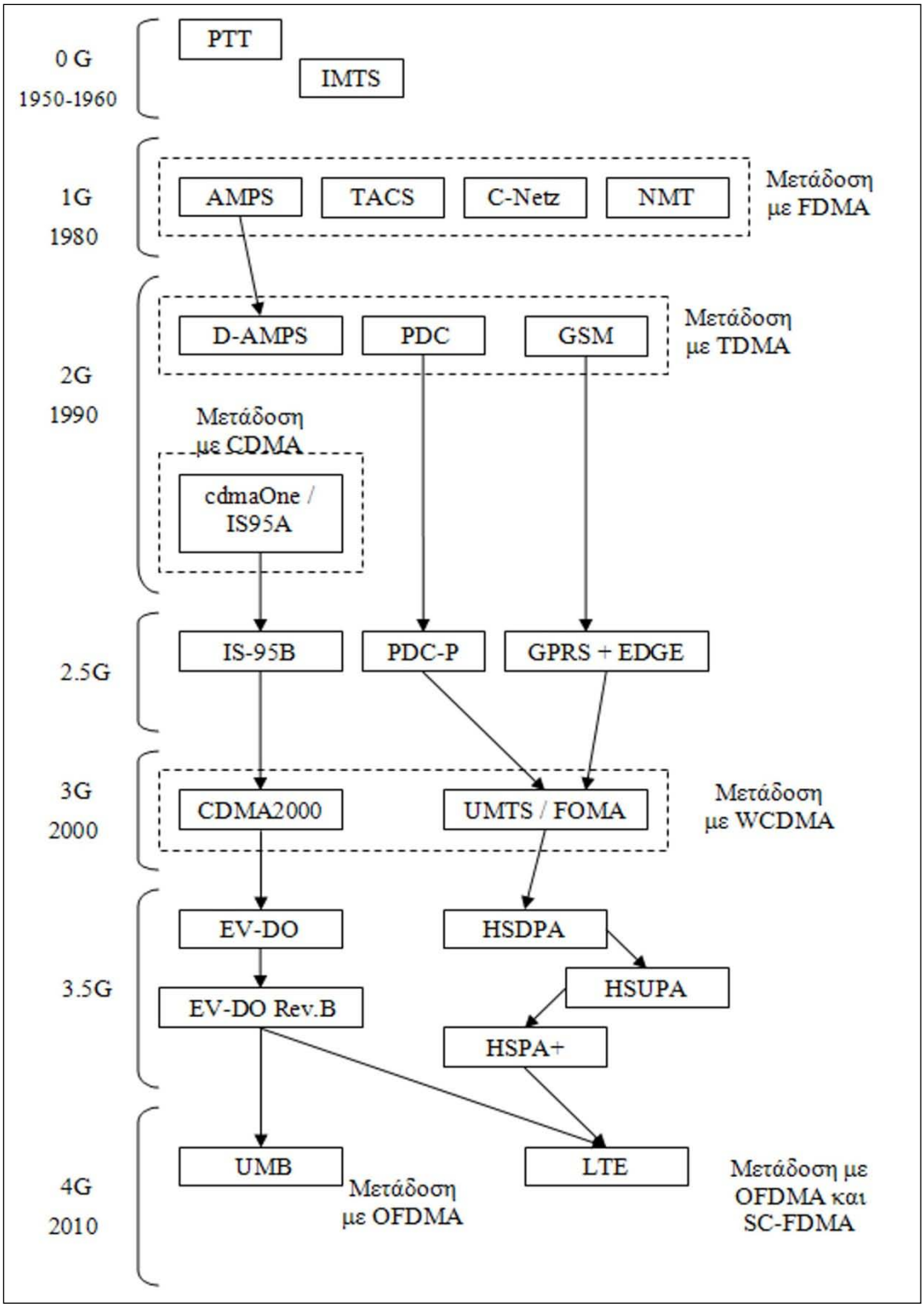
Στο τρέχον κεφάλαιο πραγματοποιείται, αρχικά, μια ιστορική αναδρομή που σχετίζεται με την πληθώρα των εξελίξεων που παρουσιάστηκαν στα δίκτυα κινητής τηλεφωνίας, ενώ στη συνέχεια παρουσιάζεται το πιο δημοφιλές και ευρύτερα χρησιμοποιούμενο εξ αυτών, το δίκτυο κινητής τηλεφωνίας που ονομάζεται GSM.

Ακόμη, παρουσιάζονται οι αναβαθμίσεις που εφαρμόστηκαν στο δίκτυο GSM και οδήγησαν στην παροχή νέων υπηρεσιών πάνω από αυτό, αλλά και το δίκτυο UMTS, που βασίζεται κατά μεγάλο βαθμό στις προδιαγραφές του GSM και αποτελεί τη διάδοχη κατάσταση για εκείνο.

1.1 Η εξέλιξη των δικτύων κινητής τηλεφωνίας

Μέχρι σήμερα έχουν συντελεστεί αρκετές εξελίξεις στον τομέα των κινητών επικοινωνιών, καθώς έχουν παρουσιαστεί τρεις γενιές δικτύων κινητής τηλεφωνίας, ενώ και η τέταρτη γενιά βρίσκεται προ των πυλών.

Όμως, πριν περάσουμε σε περισσότερες λεπτομέρειες για κάθε γενιά, θα ήταν καλύτερο να δούμε το σχήμα που ακολουθεί, όπου παρουσιάζεται το μονοπάτι εξέλιξης των δικτύων κινητής τηλεφωνίας μαζί με τις τεχνικές μετάδοσης που χρησιμοποιούνται σε κάθε περίπτωση.



Εικόνα 1: Η εξέλιξη των δικτύων κινητής τηλεφωνίας

1.1.1 Πριν από την πρώτη γενιά κινητής τηλεφωνίας

Οι πρώτες προσπάθειες για ασύρματη δικτύωση και κινητή τηλεφωνία έγιναν τις δεκαετίες του 1950 και του 1960 στις Ηνωμένες Πολιτείες Αμερικής με την εμφάνιση των συστημάτων Push-To-Talk(PTT) και Improved Mobile Telephone System(IMTS). Όμως τα συστήματα αυτά αντιμετώπισαν αρκετές δυσκολίες και δεν διαδόθηκαν αρκετά.

Ένας από τους σημαντικότερους παράγοντες που περιόρισε την διάδοση αυτών των συστημάτων ήταν η αρχιτεκτονική δικτύου στην οποία βασίζονταν. Ο τρόπος με τον οποίο δομούνταν τα συγκεκριμένα δίκτυα δεν επέτρεπε την κάλυψη μεγάλων περιοχών, ενώ και οι περιοχές που παρείχαν κάλυψη μπορούσαν να εξυπηρετούν έναν περιορισμένο αριθμό χρηστών.

Ακόμη, για την επικοινωνία των χρηστών απαιτούνταν πομποί και δέκτες μεγάλης ισχύος, έτσι σύμφωνα με τις δυνατότητες που υπήρχαν εκείνη την εποχή, οι συσκευές που ικανοποιούσαν τις απαιτήσεις ήταν αρκετά ογκώδεις, χρειάζονταν αρκετή ενέργεια και παρείχαν μηδαμινές δυνατότητες φορητότητας, γι' αυτό και οι πρώτες συσκευές που κυκλοφόρησαν απαιτούσαν εγκατάσταση σε κάποιο αυτοκίνητο.

Παρ' όλα αυτά, το πρώτο βήμα προς τη διάδοση των δικτύων κινητής τηλεφωνίας είχε γίνει.

Εικόνα 2: Μια τηλεφωνική συσκευή της δεκαετίας του 1960 σε σύγκριση με μια συσκευή του 2000

1.1.2 Η πρώτη γενιά κινητής τηλεφωνίας

Στις αρχές της δεκαετίας του 1980 παρουσιάστηκαν τα συστήματα που αποτέλεσαν την πρώτη γενιά κινητής τηλεφωνίας (First Generation ή συντομότερα 1G). Τα συστήματα 1G αποτέλεσαν τα πρώτα Κυψελοειδή Δίκτυα (Cellular Networks), αφού βασίστηκαν στην ιδέα του διαχωρισμού μιας περιοχής σε τομείς ή κυψέλες, έχοντας σαν στόχο να παρέχουν καλύτερη κάλυψη στους συνδρομητές.

Χάρις στην καλύτερη κάλυψη που προσέφερε η χρησιμοποίηση κυψελοειδών δικτύων δόθηκε η δυνατότητα για χρήση πομπών και δεκτών μικρότερης ισχύος και μικρότερου μεγέθους. Παράλληλα, η εξέλιξη της τεχνολογίας των ολοκληρωμένων κυκλωμάτων οδήγησε στην υλοποίηση τηλεφωνικών συσκευών με μικρότερο μέγεθος και μικρότερο κόστος από αυτές του παρελθόντος, ενώ ακόμη, οι νέες

συσκευές απέκτησαν μακροβιότερες μπαταρίες που προσέφεραν περισσότερη αυτονομία στον χρήστη.

Τα παραπάνω στοιχεία οδήγησαν στην ύπαρξη συσκευών κινητής τηλεφωνίας που προσέφεραν αρκετές δυνατότητες φορητότητας, είχαν ένα λογικό κόστος απόκτησης και είχαν αρχίσει να γίνονται αποδεκτές από αρκετό κόσμο. Όλοι αυτοί οι παράγοντες διαδραμάτισαν σημαντικό ρόλο στην εξάπλωση των δικτύων πρώτης γενιάς.

Τα περισσότερα από τα δίκτυα πρώτης γενιάς αναπτύχθηκαν ξεχωριστά σε κάθε χώρα και έτσι οι ασυμβατότητες που υπήρχαν μεταξύ τους ήταν αναπόφευκτες.

Ο πρωταρχικός στόχος των συστημάτων πρώτης γενιάς ήταν η μεταφορά φωνής. Η μετάδοση γινόταν με πλήρως αναλογικό τρόπο χρησιμοποιώντας την τεχνική της διαμόρφωσης συχνότητας (Frequency Modulation – FM), ενώ για καλύτερη αξιοποίηση των διαθέσιμων καναλιών επικοινωνίας εφαρμοζόταν η τεχνική πολυπλεξίας με διαίρεση συχνότητας (Frequency Division Multiple Access – FDMA).

Κάποια από τα πρώτα συστήματα κυψέλης που αναπτύχθηκαν ήταν τα:

- Advanced Mobile Phone System (AMPS) στις Η.Π.Α.
- Total Access Communication System(TACS) στη Μεγάλη Βρετανία
- C-Netz στη Δυτική Γερμανία
- Nordic Mobile Telephone system(NMT) στις Σκανδιναβικές χώρες

Το δίκτυο NMT παρουσίαζε ένα αρκετά σημαντικό στοιχείο το οποίο θα αποτελούσε και μία από τις απαιτήσεις των συστημάτων των επόμενων γενεών.

Το γεγονός ότι το NMT είχε υιοθετηθεί από κοινού από τις Σκανδιναβικές χώρες έδωσε στους τοπικούς φορείς κινητής τηλεφωνίας τη δυνατότητα να εκμεταλλευτούν την απουσία ασυμβατοτήτων και να συνάψουν συμφωνίες που έδιναν τη δυνατότητα της περιαγωγής χρηστών (Roaming).

Χάρις σε αυτό το χαρακτηριστικό, όταν ένας συνδρομητής επισκέπτεται μια ξένη χώρα μπορεί να διατηρεί την ίδια τηλεφωνική συσκευή και τον ίδιο τηλεφωνικό αριθμό, αρκεί το δίκτυο να είναι συμβατό και οι φορείς κινητής τηλεφωνίας να έχουν συμφωνήσει για παροχή των υπηρεσιών σε συνδρομητές-επισκέπτες.

1.1.3 Η δεύτερη γενιά κινητής τηλεφωνίας

Ένα από τα κύρια χαρακτηριστικά των συστημάτων πρώτης γενιάς ήταν η μετάδοση της φωνής με αναλογικό τρόπο. Το επόμενο βήμα στην εξέλιξη της κινητής τηλεφωνίας ήταν η μετάβαση από την αναλογική στην ψηφιακή μετάδοση.

Έτσι, τα συστήματα που ανήκουν στην δεύτερη γενιά κινητής τηλεφωνίας (Second Generation ή 2G) μεταδίδουν τη φωνή με πλήρως ψηφιακό τρόπο, ενώ χρησιμοποιούν και τεχνικές πολυπλεξίας με τις οποίες αυξάνεται ο αριθμός των χρηστών που εξυπηρετούνται από τα συστήματα.

Η πλειονότητα των συστημάτων χρησιμοποιεί την τεχνική διαίρεσης χρόνου Time Division Multiple Access (TDMA).

Στα κυριότερα συστήματα δεύτερης γενιάς που αναπτύχθηκαν βρίσκονται τα:

- Interim Standard 95(IS-95) ή cdmaOne
- Interim Standard 54(IS-54) ή Digital AMPS(D-AMPS)
- Personal Digital Cellular(PDC)
- Global System for Mobile communications(GSM)

Τα συστήματα cdmaOne και D-AMPS αναπτύχθηκαν περίπου την ίδια χρονική περίοδο στις Ηνωμένες Πολιτείες Αμερικής. Το cdmaOne ήταν το μοναδικό σύστημα που χρησιμοποίησε την τεχνική πολυπλεξίας Code Division Multiple Access (CDMA). Αντιθέτως, το D-AMPS χρησιμοποιούσε την τεχνική TDMA και αποτελούσε τον ψηφιακό διάδοχο του συστήματος πρώτης γενιάς AMPS, που ήταν ήδη εδραιωμένο στην αμερικανική αγορά.

Εκτός από τις ΗΠΑ, τα δύο συστήματα επεκτάθηκαν και στις υπόλοιπες χώρες της Αμερικανικής ηπείρου αλλά και σε μερικές χώρες της Ασίας και της Μέσης Ανατολής. Γενικότερα υπήρξε ισχυρός ανταγωνισμός ανάμεσα στα δύο συστήματα, με τελικό νικητή το σύστημα cdmaOne που απέκτησε και το μεγαλύτερο αριθμό συνδρομητών.

Το PDC ήταν σύστημα κινητής τηλεφωνίας δεύτερης γενιάς που αναπτύχθηκε και χρησιμοποιήθηκε αποκλειστικά στην Ιαπωνία. Χρησιμοποιούσε και αυτό την τεχνική πολυπλεξίας TDMA και εκτός από τη μετάδοση φωνής παρείχε και κάποιες υπηρεσίες δεδομένων.

Το GSM όμως είναι το σύστημα που παρουσιάζει το περισσότερο ενδιαφέρον, καθώς σταδιακά εξελίχθηκε σε αυτό που υποδηλώνουν και τα αρχικά της ονομασίας του, έγινε δηλαδή το παγκόσμιο πρότυπο για τις κινητές επικοινωνίες.

Η ανάπτυξη του GSM ξεκίνησε το 1982, όταν το αρμόδιο όργανο για τον τομέα των τηλεπικοινωνιών της Ευρωπαϊκής Ένωσης, το CEPT (Conférence Européenne des Postes et des Télécommunications), πήρε την απόφαση να δημιουργήσει την επιτροπή GSM (Groupe Spéciale Mobile).

Στόχος της επιτροπής, εκτός από την ονομασία του συστήματος με τη διατήρηση των αρχικών GSM, ήταν να καθορίσει τις προδιαγραφές σύμφωνα με τις οποίες θα αναπτυχθεί το νέο σύστημα κινητής τηλεφωνίας, το οποίο θα πρέπει να ικανοποιεί τέσσερις κεντρικές απαιτήσεις.

Έτσι αποφασίστηκε ότι το υπό ανάπτυξη σύστημα:

- θα είναι πλήρως ψηφιακό
- θα παρέχει δυνατότητες για μετάδοση φωνής και δεδομένων
- θα επιτρέπει την περιαγωγή χρηστών
- θα υιοθετηθεί από όλες τις Ευρωπαϊκές χώρες

Για αυτό το σκοπό το CEPT δέσμευσε τις απαραίτητες ομάδες συχνοτήτων στην περιοχή των 900MHz. Τα επόμενα χρόνια ακολούθησαν οι δοκιμές και η αξιολόγηση των υποψηφίων λύσεων, οι οποίες το 1991 θα οδηγούσαν στη θεμελίωση του

προτύπου GSM-900, δηλαδή το GSM που λειτουργεί στη ζώνη συχνοτήτων των 900 MHz. Το GSM χρησιμοποιούσε και αυτό την τεχνική πολυπλεξίας TDMA.

Ανάμεσα στις υπηρεσίες δεδομένων που προσέφερε το GSM ήταν και η υπηρεσία σύντομων μηνυμάτων Short Messaging Service (SMS).

Η αρχική σκέψη για τα SMS μηνύματα ήταν να ενημερώνουν τον συνδρομητή για νέα ηχητικά μηνύματα που έχει στον τηλεφωνητή του και να ειδοποιούν για τηλεφωνικές κλήσεις που δέχτηκε ενώ το τηλέφωνο του ήταν εκτός λειτουργίας ή εκτός περιοχής κάλυψης.

Αργότερα άλλαξε η υλοποίηση της υπηρεσίας και δόθηκε η δυνατότητα στους συνδρομητές για την αποστολή και λήψη σύντομων μηνυμάτων που οι ίδιοι συνέτασσαν.

Η αναβαθμισμένη υπηρεσία ήταν αρκετά επιτυχημένη και αποτελεί πλέον μία σημαντική πηγή εσόδων για τους φορείς κινητής τηλεφωνίας, αφού κάθε χρόνο ανταλλάσσονται δεκάδες δισεκατομμύρια μηνύματα ανάμεσα στους συνδρομητές. Αυτός είναι και ένας από τους λόγους που οδήγησαν στη διατήρηση της υπηρεσίας SMS και στα δίκτυα κινητής τηλεφωνίας της επόμενης γενιάς.

Αξίζει να αναφέρουμε ότι τα μηνύματα SMS που στάλθηκαν μόνο για το έτος 2005 ξεπέρασαν το 1 τρισεκατομμύριο[*].

Πολύ σύντομα το GSM-900 ξεπέρασε τον αρχικό του στόχο, που δεν ήταν άλλος από την χρησιμοποίηση του στις Ευρωπαϊκές χώρες. Έτσι το σύστημα διαδόθηκε και σε αρκετές ακόμα περιοχές, όπως είναι οι χώρες στο βόρειο τμήμα της Αφρικής, οι χώρες της Μέσης Ανατολής, αλλά και ένα μεγάλο μέρος των Ασιατικών χωρών και των χωρών της Ωκεανίας.

Αργότερα προέκυψαν και δύο παράγωγα συστήματα του GSM-900:

- Το GSM-1800, που λειτουργούσε στη ζώνη συχνοτήτων των 1800 MHz και επεκτάθηκε στις χώρες της Ασίας και της Νοτίου Αμερικής.
- Το GSM-1900, που χρησιμοποιούσε την περιοχή των 1900 MHz και χρησιμοποιήθηκε για την κάλυψη των περιοχών της Βορείου Αμερικής, μπαίνοντας έτσι στον ανταγωνισμό με τα συστήματα cdmaOne και D-AMPS.

Αρχικά υπήρχαν κάποιες ασυμβατότητες με τις συσκευές κινητής τηλεφωνίας και τις διαφορετικές ζώνες συχνοτήτων. Για παράδειγμα μια συσκευή που λειτουργούσε στην Ευρώπη δεν ήταν σε θέση να λειτουργεί και στη Νότια Αμερική, όμως αργότερα κυκλοφόρησαν συσκευές με δυνατότητα χρήσης σε δύο ζώνες συχνοτήτων (Dual Band mode), ενώ τελικά εμφανίστηκαν συσκευές με δυνατότητα λειτουργίας και στις τρεις διαθέσιμες ζώνες συχνοτήτων (Triple Band mode) στα 900, 1800 και 1900 MHz.

Το γεγονός αυτό, σε συνδυασμό με το ότι οι περισσότεροι πάροχοι κινητής τηλεφωνίας ανά τον κόσμο σύναψαν μεταξύ τους συμφωνίες για Roaming, έδωσε τη δυνατότητα σε ένα συνδρομητή κινητής τηλεφωνίας να ταξιδεύει σε διάφορα μέρη

[*] Όπως αναφέρεται στην επίσημη ιστοσελίδα του GSM στο URL www.gsmworld.com

του κόσμου και να συνεχίζει να απολαμβάνει τις υπηρεσίες κινητής τηλεφωνίας έχοντας μία συσκευή, τον ίδιο αριθμό τηλεφώνου και μία χρέωση από τον οικείο φορέα στον οποίο είναι συνδεδεμένος.

Με αυτό τον τρόπο το GSM εξελίχθηκε σε παγκόσμιο πρότυπο για τις κινητές επικοινωνίες. Αυτό επιβεβαιώνεται και από τα περίπου 3 δισεκατομμύρια συνδρομητών που εξυπηρετούνται από δίκτυα GSM, τα οποία έχουν αναπτυχθεί σε ποσοστό 95% επί του συνόλου των χωρών της υφηλίου[*].

Ο αριθμός συνδρομητών του GSM ισοδυναμεί με ένα ποσοστό κοντά στο 80% των συνδέσεων παγκοσμίως, με δεύτερο να ακολουθεί το cdmaOne έχοντας ένα ποσοστό κοντά στο 17% των συνδέσεων ανά την υφήλιο[*].

1.1.4 Ανάμεσα σε δεύτερη και τρίτη γενιά

Παράλληλα με την άνθηση των δικτύων κινητής τηλεφωνίας ξεκίνησε και η εκρηκτική ανάπτυξη του Internet. Με την πάροδο του χρόνου δημιουργήθηκε από τους χρήστες κινητών τηλεφώνων η ανάγκη για περισσότερες και πιο προηγμένες υπηρεσίες δεδομένων, όπως είναι η δυνατότητα ασύρματης πρόσβασης στο Internet.

Η διαδικασία προτυποποίησης των συστημάτων τρίτης γενιάς είχε ήδη αρχίσει, όμως ο πρωταρχικός τους στόχος δεν ήταν η παροχή εξελιγμένων υπηρεσιών δεδομένων.

Τότε αποφασίστηκε η αναβάθμιση των συστημάτων δεύτερης γενιάς και η εκ νέου προτυποποίηση των συστημάτων τρίτης γενιάς για να είναι σε θέση να ικανοποιήσουν τις νέες ανάγκες.

Τα αναβαθμισμένα συστήματα δεύτερης γενιάς αποτέλεσαν μιαν ενδιάμεση γενιά, η οποία πολύ συχνά λέγεται γενιά 2.5 ή συντομότερα 2.5G. Τα συστήματα 2.5G θα κάλυπταν ένα μέρος από τις ανάγκες των συνδρομητών μέχρι να θεμελιωθούν και να τεθούν σε πλήρη λειτουργία τα συστήματα τρίτης γενιάς.

Η αναβάθμιση των συστημάτων δεύτερης γενιάς ήταν αναγκαία, καθώς οι νέες υπηρεσίες έχουν τουλάχιστον ένα από τα ακόλουθα στοιχεία που χαρακτηρίζουν την κίνηση στο δίκτυο:

- Μετάδοση με καταιγισμούς δεδομένων
- Αρκετά συχνή μετάδοση μικρού όγκου δεδομένων
- Λιγότερο συχνή μετάδοση μεγάλου όγκου δεδομένων

Τα συστήματα δεύτερης γενιάς βασίζονται στη χρήση συνδέσεων με μεταγωγή κυκλώματος (Circuit-Switched Connections).

Καθότι η αρχική σκέψη ήταν να διατηρηθεί η χρήση μεταγωγής κυκλώματος, έτσι, μία από τις αναβαθμίσεις που παρουσιάστηκε ήταν η τεχνολογία High Speed Circuit

[*] Όλα τα στοιχεία προέρχονται από την επίσημη ιστοσελίδα του GSM στο URL www.gsmworld.com
[*] Όμοια με την παραπάνω παραπομπή

Switched Data(HSCSD), η οποία δίνει την δυνατότητα ταχύτερης μεταφοράς δεδομένων μέσα από συνδέσεις μεταγωγής κυκλώματος.

Η μέγιστη ταχύτητα που παρείχε η αναβάθμιση HSCSD ήταν 56kbps, η οποία ισοδυναμεί με την ταχύτητα μιας dial-up σύνδεσης.

Όμως αποδείχθηκε ότι τα παραπάνω χαρακτηριστικά κίνησης δεν ευνοούν την χρήση μεταγωγής κυκλώματος, αφού κατά την διάρκεια μετάδοσης με καταιγισμούς δεδομένων θα είχαμε κάποια διαστήματα με κίνηση δεδομένων που θα τα διαδέχονταν άλλα διαστήματα χωρίς καμία δραστηριότητα.

Αυτό το γεγονός θα οδηγούσε σε αναποτελεσματική χρήση των διαθέσιμων πόρων, γιατί η μεταγωγή κυκλώματος απαιτεί σταθερή σύνδεση για όλη την διάρκεια της μετάδοσης. Η λύση δόθηκε με την χρήση συνδέσεων μεταγωγής πακέτων (Packet-Switched Connections).

Με την μεταγωγή πακέτων δεν απαιτείται σταθερή σύνδεση κι έτσι δεσμεύονται πόροι μόνο στην περίπτωση που υπάρχουν δεδομένα προς μετάδοση. Με αυτόν τον τρόπο έχουμε καλύτερη αξιοποίηση του δικτύου, αφού με την ολοκλήρωση της μετάδοσης οι δεσμευμένοι πόροι απελευθερώνονται και δίνεται η δυνατότητα πολλαπλής πρόσβασης σε περισσότερους συνδρομητές.

Οι ακόλουθες είναι μερικές από τις αναβαθμίσεις που εμφανίστηκαν:

- General Packet Radio Service(GPRS)
- Enhanced Data for GSM/Global Evolution(EDGE)
- PDC-P
- Interim Standard 95B(IS-95B)

Το General Packet Radio Service(GPRS) παρουσιάστηκε σαν μια προσθήκη του δικτύου GSM που επιτρέπει στους συνδρομητές την αποστολή και λήψη δεδομένων μέσα από συνδέσεις μεταγωγής πακέτων.

Σύμφωνα με τις τεχνικές προδιαγραφές, το GPRS προσφέρει ταχύτητα μέχρι 115kbps έχοντας ενεργοποιημένο τον έλεγχο λαθών μετάδοσης. Εναλλακτικά, έχοντας απενεργοποιημένο τον έλεγχο λαθών, παρέχεται ταχύτητα μέχρι και 170kbps.

Για έναν φορέα κινητής τηλεφωνίας το GPRS αποτελεί συνήθως το πρώτο βήμα στη διαδρομή της εξέλιξης προς τα συστήματα τρίτης γενιάς, καθώς, όπως θα δούμε και παρακάτω, η συγκεκριμένη αναβάθμιση διαδραματίζει ένα σημαντικό ρόλο σε ένα σύστημα τρίτης γενιάς.

Η τεχνολογία Enhanced Data for GSM/Global Evolution(EDGE) βασίζεται σε ένα νέο σχήμα διαμόρφωσης συχνοτήτων και καταφέρνει να τριπλασιάσει τον ρυθμό μεταφοράς στο ασύρματο τμήμα του δικτύου GSM.

Η προσθήκη της τεχνολογίας EDGE σε ένα δίκτυο είναι συνήθως μια ελκυστική πρόταση, αφού για να υλοποιηθεί χρειάζεται μόνο την αναβάθμιση του λογισμικού που χρησιμοποιείται από τους σταθμούς βάσης του δικτύου. Ένα ακόμη σημαντικό στοιχείο είναι η δυνατότητα που έχει το EDGE να λειτουργεί σε συνεργασία με άλλες αναβαθμίσεις.

Έτσι, στην περίπτωση όπου συνδυάζονται τα EDGE και GPRS, με τον συνδυασμό να ονομάζεται Enhanced GPRS ή EGPRS, έχουμε ταχύτητα μέχρι και 384kbps.

Το PDC-P αποτέλεσε το δίκτυο μεταγωγής πακέτου που ενσωματώθηκε στο PDC, το δίκτυο δεύτερης γενιάς που είχε αναπτυχθεί στην Ιαπωνία.

Μέσω της συγκεκριμένης αναβάθμισης είχαμε και την πρώτη εμφάνιση της υπηρεσίας i–mode, η οποία παρείχε ασύρματη πρόσβαση στο Internet μαζί με άλλες συμπληρωματικές υπηρεσίες δεδομένων και γνώρισε τεράστια επιτυχία στην ιαπωνική αγορά, καθώς είχε περίπου σαράντα εκατομμύρια συνδρομητές.

Τέλος, η αναβάθμιση IS-95B ήταν αυτή που με τη σειρά της προσέθεσε στο δίκτυο δεύτερης γενιάς cdmaOne την δυνατότητα μεταγωγής πακέτων με ταχύτητες που πλησίαζαν τα 64kbps.

1.1.5 Η τρίτη γενιά κινητής τηλεφωνίας

Στις αρχές της δεκαετίας του 1990 άρχισαν να καθιερώνονται τα συστήματα κινητής τηλεφωνίας δεύτερης γενιάς, ενώ παράλληλα την ίδια χρονική περίοδο, πιο συγκεκριμένα το 1992, ξεκίνησε από τη Διεθνή Ένωση Τηλεπικοινωνιών (International Telecommunications Union – ITU) η υλοποίηση ενός αρκετά φιλόδοξου οράματος.

Ο στόχος ήταν η δημιουργία ενός προτύπου επικοινωνιών που θα υιοθετηθεί από κάθε χώρα και θα παρέχει κάλυψη στους συνδρομητές σε «οποιοδήποτε σημείο του πλανήτη, οποιαδήποτε στιγμή». Το νέο σύστημα ονομάστηκε Future Public Land Mobile Telecommunications System(FPLMTS).

Το FPLMTS, για να εκπληρώσει την προϋπόθεση της απανταχού και παντοτινής κάλυψης, προέβλεπε την ενοποίηση τηλεπικοινωνιακών δικτύων και στόχευε στην απρόσκοπτη περιαγωγή των συνδρομητών ανά την υφήλιο μέσω της χρήσης πολλαπλών διασυνδεδεμένων επίγειων και δορυφορικών δικτύων. Όμως η ραγδαία εξάπλωση του Internet δεν θα άφηνε ανεπηρέαστη την πορεία εξέλιξης του FPLMTS, γεγονός που οδήγησε το 1997 στον επαναπροσδιορισμό των στόχων του συστήματος.

Η νέα εκδοχή του FPLMTS, που μετονομάστηκε σε International Mobile Telecommunications standard – 2000 ή πιο σύντομα IMT-2000, αφορούσε ένα σύστημα κινητής τηλεφωνίας τρίτης γενιάς (Third Generation – 3G) που θα έχει σαν στόχο να υιοθετηθεί παγκοσμίως και να δώσει περισσότερη έμφαση στην παροχή κινητών ευρυζωνικών υπηρεσιών δεδομένων.

Ο αριθμός 2000 συμβόλιζε τρεις από τις απαιτήσεις του συστήματος IMT-2000:

- τη χρονιά που θα έμπαινε σε πλήρη λειτουργία
- τα 2000kbps ή περίπου 2Mbps που θα προσφέρονταν σαν μέγιστη ταχύτητα μεταφοράς δεδομένων
- τη λειτουργία του στην περιοχή συχνοτήτων των 2000 MHz

Παράλληλα όμως με την διαδικασία εξέλιξης του IMT-2000 διεξάγονταν μεμονωμένες έρευνες πάνω στα δίκτυα κινητής τηλεφωνίας. Οι περισσότερες εξ αυτών προέρχονταν από τρεις πλευρές, από την Ευρωπαϊκή Ένωση, από τις Ηνωμένες Πολιτείες Αμερικής και από την Ιαπωνία.

Κάθε μια από τις προαναφερθείσες πλευρές είχε συγκεκριμένους λόγους για να επηρεάσει την εξέλιξη του IMT-2000 και επιθυμούσε την προώθηση του δικού της συστήματος.

Στην Ευρωπαϊκή Ένωση επιθυμούσαν την ύπαρξη συμβατότητας με το GSM και θα ήθελαν να είχαν την δυνατότητα να επαναχρησιμοποιήσουν το δίκτυο κορμού που προϋπήρχε. Οπότε προέκυψε το σύστημα επικοινωνιών Universal Mobile Telecommunications System(UMTS), οι προδιαγραφές του οποίου σχηματίστηκαν μέσα από ερευνητικά προγράμματα όπως τα:

- Research in Advanced Communications in Europe(RACE)
- Advanced Communications Technologies and Services(ACTS)
- Future Radio Wideband Multiple Access Systems(FRAMES)

Οι Ηνωμένες Πολιτείες Αμερικής προωθούσαν την τεχνική CDMA και θα ήθελαν το δικό τους δίκτυο τρίτης γενιάς να βασίζεται στο σύστημα δεύτερης γενιάς που προϋπήρχε, το cdmaOne.

Στην Ιαπωνία το σύστημα δεύτερης γενιάς ήταν υπερφορτωμένο. Για αυτό το λόγο διεξάγονταν έρευνες πάνω στην τεχνική Wideband CDMA(WCDMA) με σκοπό την αύξηση της χωρητικότητας χρηστών και τη μέγιστη δυνατή αξιοποίηση του διαθέσιμου φάσματος.

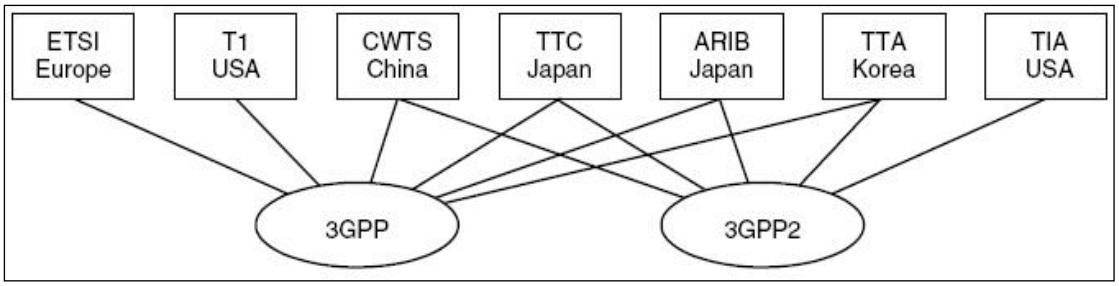
Όλα τα παραπάνω γεγονότα καθιστούσαν εξαιρετικά δύσκολη τη δημιουργία και τη διάδοση ενός κοινού προτύπου τηλεπικοινωνιών, καθώς η κάθε πλευρά κατέθεσε στην ITU τη δική της πρόταση δημιουργίας συστημάτων τρίτης γενιάς. Ενώ η ITU με τη σειρά της έθεσε ένα σύνολο από τεχνικές απαιτήσεις και ενέκρινε όσες προτάσεις ικανοποιούσαν τις απαιτήσεις για τα νέα συστήματα.

Με τη δεδομένη κατάσταση το IMT-2000 μετατράπηκε από ένα κοινό πρότυπο σε ένα πλαίσιο από τεχνικές προδιαγραφές και απαιτήσεις συστήματος, χωρίστηκε σε πέντε πρότυπα και δημιούργησε μιαν οικογένεια τηλεπικοινωνιακών συστημάτων τρίτης γενιάς.

Για την ανάπτυξη των προτύπων συστάθηκαν από τις εμπλεκόμενες χώρες και από τους αντίστοιχους φορείς προτυποποίησης τηλεπικοινωνιών δύο ομάδες συνεργασίας για τα συστήματα τρίτης γενιάς. Πρόκειται για τις ομάδες:

- Third Generation Partnership Project (3GPP), που την αποτελούν φορείς από την Ευρώπη, την Αμερική και την Ασία.
- Third Generation Partnership Project 2(3GPP2), που αποτελείται από οργανισμούς προτυποποίησης από την Ασία και την Αμερική.

Στο σχήμα που ακολουθεί εμφανίζονται οι οργανισμοί που απαρτίζουν τα 3GPP και 3GPP2.



Εικόνα 3: Η οργάνωση των 3GPP και 3GPP2

Η παγκόσμια καθιέρωση ενός κοινού προτύπου ήταν πλέον ανέφικτη. Όμως ένας από τους στόχους των ομάδων συνεργασίας ήταν να παρέχεται όσο το δυνατόν μεγαλύτερη συμβατότητα ανάμεσα στα συστήματα που αναπτύσσονται. Έτσι αποφασίστηκε η χρήση της τεχνικής πολυπλεξίας WCDMA σε κάθε ένα από τα καινούργια δίκτυα.

Το 3GPP ανέπτυξε δύο από τα πέντε πρότυπα του IMT-2000. Το ένα πρότυπο ήταν το ITU-DS και το άλλο ήταν το ITU-SC.

Με το πρότυπο ITU-DS τυποποιήθηκε η ευρωπαϊκή πρόταση, το σύστημα Universal Mobile Telecommunications System(UMTS), η ανάπτυξη του οποίου ξεκίνησε από την Ευρώπη, οπότε το δίκτυο κορμού που χρησιμοποιείται σε αυτό έχει την βάση του στο GSM και το GPRS.

Μια παραλλαγή του UMTS αναπτύχθηκε από κοινού από το ευρωπαϊκό ίδρυμα προτυποποίησης τηλεπικοινωνιών, το European Telecommunications Standards Institute(ETSI), και την ιαπωνική ένωση ARIB.

Το σύστημα που προέκυψε εμφανίστηκε στην Ιαπωνική αγορά το 2001 με την ονομασία Freedom Of mobile Multimedia Access(FOMA) και έγινε το πρώτο δίκτυο 3G που τέθηκε σε εμπορική χρήση.

Το δεύτερο πρότυπο, με την ονομασία ITU-SC, αποτελεί την προτυποποίηση του EDGE, της τεχνικής που εφαρμοζόταν σαν αναβάθμιση του δικτύου δεύτερης γενιάς GSM. Η προτυποποίηση που έγινε έχει σαν στόχο την χρήση του EDGE και στα δίκτυα τρίτης γενιάς.

Γι’ αυτό και άλλαξε η ονομασία που όριζαν τα αρχικά του EDGE από Enhanced Data for GSM Evolution σε Enhanced Data for Global Evolution.

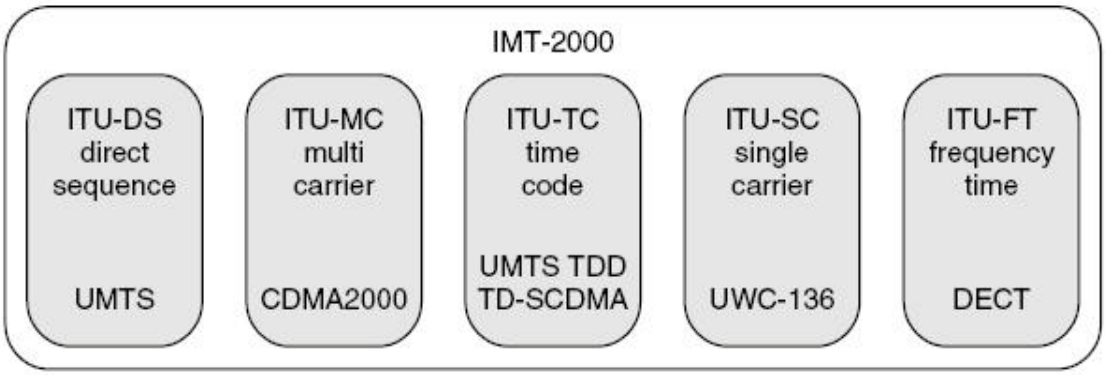
Η ομάδα συνεργασίας 3GPP2 ανέλαβε την ανάπτυξη του προτύπου ITU-MC. Με το συγκεκριμένο πρότυπο τέθηκαν τα θεμέλια για τη δημιουργία του συστήματος που αναπτύχθηκε στις Ηνωμένες Πολιτείες Αμερικής και έγινε γνωστό στην τοπική αγορά τηλεπικοινωνιών με την ονομασία CDMA2000.

Το πρότυπο ITU-FT αναπτύχθηκε από το ETSI και σχετίζεται με την ασύρματη τηλεφωνία, από εκεί προκύπτει το σύστημα Digital Enhanced Cordless Telephone(DECT), το οποίο είναι το ευρωπαϊκό πρότυπο για ασύρματα τηλέφωνα.

Τέλος, το πρότυπο ITU-TC αποτελεί μια παραλλαγή του συστήματος UMTS που προτάθηκε από την κινεζική αρχή CWTS. Το συγκεκριμένο σύστημα 3G

αναπτύσσεται αποκλειστικά στην Κίνα. Μετά από ένα εκτεταμένο στάδιο δοκιμών, που ξεκίνησε το 2006, το σύστημα τέθηκε σε εμπορική χρήση από τον Ιανουάριο του 2009 και ήδη έχει αποκτήσει τα πρώτα εκατομμύρια συνδρομητών από την κινεζική αγορά.

Στο σχήμα που ακολουθεί παρουσιάζονται τα πρότυπα που περιλαμβάνει το IMT-2000 και τα συστήματα που προκύπτουν με βάση τις συγκεκριμένες προδιαγραφές.



Εικόνα 4: Η οικογένεια προτύπων του IMT-2000

Σύμφωνα με τελευταία στοιχεία, το UMTS αποτελεί το πλέον διαδεδομένο σύστημα τρίτης γενιάς. Αφού μέχρι τα μέσα του 2010 έχουν αναπτυχθεί περίπου 320 δίκτυα UMTS που εξυπηρετούν 600 εκατομμύρια συνδρομητών. Το δεύτερο πιο διαδεδομένο σύστημα είναι το CDMA2000, καθώς μέχρι το πρώτο τέταρτο του 2010 έχουν δημιουργηθεί για αυτό περίπου 500 εκατομμύρια συνδέσεων [*].

1.1.6 Μια ματιά στο μέλλον

Ο διαχωρισμός ανάμεσα στα δίκτυα πρώτης και δεύτερης γενιάς βασίστηκε στη μετάβαση από την αναλογική στη ψηφιακή τεχνολογία μετάδοσης, τα μεν δίκτυα πρώτης γενιάς ήταν αναλογικά, τα δε δίκτυα δεύτερης γενιάς ήταν ψηφιακά.

Με ανάλογο τρόπο, έχοντας αυτή τη φορά σαν κριτήριο με ποιο τρόπο γίνεται μεταγωγή της κίνησης, θα γίνει και ο διαχωρισμός ανάμεσα σε δίκτυα τρίτης και τέταρτης γενιάς.

Τα δίκτυα τρίτης γενιάς συνδυάζουν τη μεταγωγή κυκλώματος με τη μεταγωγή πακέτων. Η μεταγωγή κυκλώματος χρησιμοποιείται σε εφαρμογές πραγματικού χρόνου(real-time), όπως είναι η μετάδοση φωνής, ενώ η μεταγωγή πακέτου χρησιμοποιείται για την παροχή υπηρεσιών δεδομένων.

Στα δίκτυα τέταρτης γενιάς το δίκτυο κορμού θα μετατραπεί εξ' ολοκλήρου σε IP δίκτυο(All-IP-Network), όλες οι υπηρεσίες θα παρέχονται μόνο με τη χρησιμοποίηση μεταγωγής πακέτου και είναι πολύ πιθανό να χρησιμοποιείται η έκτη έκδοση του πρωτοκόλλου IP(IPv6).

[*] Τα στοιχεία προέρχονται από την ιστοσελίδα www.umts-forum.org

Η τέταρτη γενιά δικτύων κινητής τηλεφωνίας (Fourth Generation – 4G) βρίσκεται ακόμα σε φάση ανάπτυξης. Για την ώρα έχουν εμφανιστεί ορισμένες αναβαθμίσεις για τα δίκτυα τρίτης γενιάς και οι οποίες κατά ένα μέρος επιβεβαιώνουν την κατεύθυνση προς την οποία θα κινηθούν τα συστήματα τέταρτης γενιάς.

Οι συγκεκριμένες αναβαθμίσεις επιτρέπουν την πρόσβαση σε ταχύτερη μεταγωγή πακέτων και στοχεύουν στη βελτίωση των παρεχόμενων υπηρεσιών δεδομένων.

Για το δίκτυο CDMA2000 υπάρχει η αναβάθμιση Evolution-Data Optimized ή Evolution-Data Only(EV-DO). Η πιο πρόσφατη αναθεωρημένη έκδοση, που κυκλοφορεί με την ονομασία EV-DO Rev.B, δίνει ταχύτητες που πλησιάζουν τα 15Mbps για λήψη και τα 1.8Mbps για την αποστολή δεδομένων.

Για το δίκτυο UMTS υπάρχει η συλλογή πρωτοκόλλων High Speed Packet Access(HSPA). Στην οικογένεια του HSPA ανήκουν τρία πρωτόκολλα:

- το High Speed Downlink Packet Access(HSDPA)
- το Enhanced Uplink(EUL) ή High Speed Uplink Packet Access(HSUPA)
- το Evolved HSPA ή HSPA Evolution(HSPA+)

Το HSDPA πρωτοεμφανίστηκε το 2005, και αφού προστεθεί σε ένα δίκτυο UMTS παρέχει μέγιστες ταχύτητες Λήψης/Αποστολής δεδομένων(Downlink/Uplink) που αγγίζουν τα 14.4Mbps/384kbps.

Όμως ελάχιστοι φορείς κινητής τηλεφωνίας παρέχουν αυτές τις ταχύτητες, αφού το μεγαλύτερο ποσοστό από τα 300 περίπου δίκτυα που έχουν ενσωματωμένο το HSDPA λειτουργούν σε ταχύτητες που φτάνουν στα 3.6Mbps[*].

Το HSUPA κυκλοφόρησε στην αγορά το 2007 και αποτελεί μια βελτιωμένη εκδοχή του HSDPA. Το HSUPA έχει την ίδια ταχύτητα λήψης με το HSDPA αλλά παρέχει γρηγορότερη αποστολή δεδομένων, με τη μέγιστη ταχύτητα να φτάνει τα 5.76Mbps. Μέχρι στιγμής έχουν αναπτυχθεί πάνω από 110 δίκτυα που χρησιμοποιούν το πρωτόκολλο HSUPA, με τα περισσότερα από αυτά να παρέχουν ταχύτητα αποστολής δεδομένων στα 1.92Mbps[*].

Στις αρχές του 2009 ολοκληρώθηκαν οι εργασίες αναβάθμισης και τεθήκαν σε λειτουργία τα πρώτα δίκτυα που θα ενσωματώνουν το HSPA+. Τα αναβαθμισμένα δίκτυα είναι σε θέση να προσφέρουν ακόμα πιο υψηλές ταχύτητες μεταφοράς δεδομένων, οι οποίες πλησιάζουν τα 42Mbps κατά τη λήψη και τα 22Mbps κατά την αποστολή.

Μέχρι τα μέσα του 2010 είχαν αναπτυχθεί πάνω από 60 δίκτυα HSPA+ σε 35 διαφορετικές χώρες, ενώ μέχρι το τέλος της χρονιάς, και άλλα 110 δίκτυα αναμένεται να τεθούν σε λειτουργία μετά από τις δεσμεύσεις των αρμόδιων πάροχων κινητής τηλεφωνίας[*].

[*] Τα στοιχεία προέρχονται από την ιστοσελίδα http://www.gsacom.com/

Το HSPA+ θεωρείται σαν το τελευταίο βήμα πριν από την χρησιμοποίηση των δικτύων της νέας γενιάς, καθώς χρησιμοποιεί μερικές από τις τεχνικές που συναντιόνται στην τέταρτη γενιά.

Ο κυριότερος εκφραστής της νέας γενιάς είναι το Long Term Evolution(LTE), που αναπτύσσεται από την ομάδα συνεργασίας 3GPP, ενώ για μια περίοδο υπήρχε και η εναλλακτική του Ultra Mobile Broadband(UMB), πάνω στο οποίο εργάζεται η ομάδα συνεργασίας 3GPP2.

Η αναβάθμιση του LTE πραγματοποιείται σε δύο τμήματα:

- Το Evolved Universal Terrestrial Access Network(E-UTRAN). Το E-UTRAN είναι η εξελιγμένη μορφή που έχει η ασύρματη διασύνδεση των συνδρομητών με το δίκτυο κορμού.
- Το System Architecture Evolution(SAE). Το SAE αποτελεί την εξέλιξη του δικτύου κορμού, το οποίο βασίζεται στο IP πρωτόκολλο και η κίνηση γίνεται με μεταγωγή πακέτου.

Η νέα διασύνδεση υπόσχεται την αποτελεσματικότερη χρησιμοποίηση του φάσματος, τη βελτίωση της κάλυψης, την αύξηση της χωρητικότητας του συστήματος και τη χρήση δυναμικού και κλιμακωτού εύρους ζώνης με τιμές από 1.25MHz έως 20MHz.

Όλα αυτά είναι εφικτά χάρη στη συνδυασμένη χρήση προηγμένων κεραιών Multiple Input Multiple Output(MIMO) και των τεχνικών πολυπλεξίας Orthogonal Frequency Division Multiple Access(OFDMA) για το downlink, και Single Carrier Frequency Division Multiple Access(SC-FDMA) για το uplink.

Σύμφωνα με τελευταίες δοκιμές λειτουργίας του LTE, στις οποίες χρησιμοποιήθηκαν κεραίες MIMO 4x4 μαζί με εύρος ζώνης 20MHz, μετρήθηκε ταχύτητα λήψης κοντά στα 326Mbps και ταχύτητα αποστολής στα 86Mbps.

Το UMB αποτελούσε το μονοπάτι εξέλιξης των δικτύων τρίτης γενιάς CDMA2000, ενώ το LTE αρχικά προοριζόταν για αναβάθμιση του δικτύου τρίτης γενιάς UMTS.

Όμως, έπειτα από συνεργασία ανάμεσα στις ομάδες 3GPP και 3GPP2, δόθηκε και στα δίκτυα CDMA2000 η δυνατότητα για αναβάθμιση μέσω του LTE.

Το γεγονός αυτό, σε συνδυασμό με το ελάχιστο ενδιαφέρον που παρουσίασαν οι πάροχοι κινητής τηλεφωνίας για το UMB οδήγησε την ομάδα εργασίας του εν λόγω δικτύου να σταματήσει την ανάπτυξη του, έτσι, από το τέλος του 2008 το LTE αποτελεί τη μοναδική λύση για την αναβάθμιση των δικτύων UMTS και των δικτύων CDMA2000[*].

Το Δεκέμβριο του 2009 τέθηκαν σε εμπορική χρήση τα πρώτα 2 δίκτυα LTE, άλλα 22 δίκτυα αναμένεται να τεθούν σε λειτουργία μέχρι το τέλος του 2010, ενώ, μέχρι τα μέσα της ίδιας χρονιάς, γύρω στους 80 πάροχους από 31 χώρες έχουν δεσμευθεί ότι θα αναπτύξουν και αυτοί τα δικά τους δίκτυα LTE.

[*] http://www.reuters.com/article/idUSN1335969420081113?rpc=401&

1.2 Το δίκτυο GSM

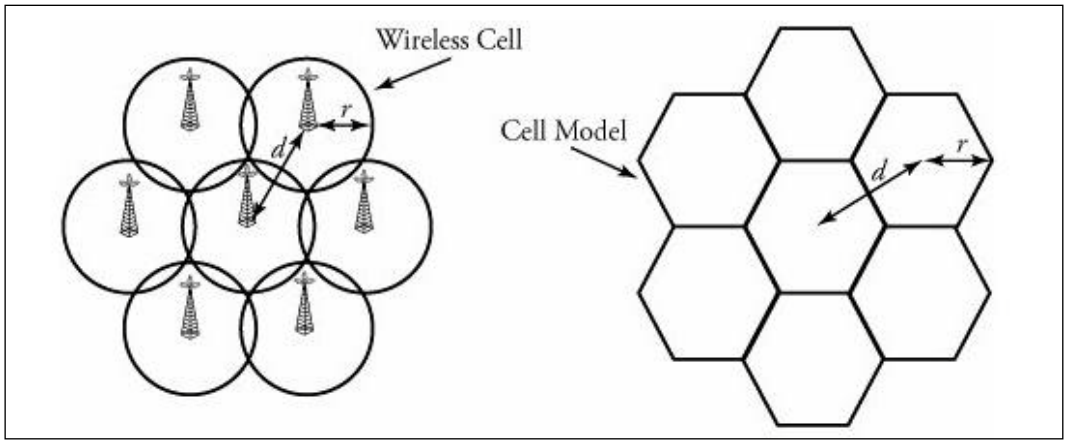
Το Global System for Mobile communications (GSM) είναι ένα κυψελοειδές δίκτυο κινητής τηλεφωνίας δεύτερης γενιάς που υποστηρίζει ψηφιακή μετάδοση φωνής και υπηρεσίες δεδομένων με χαμηλό ρυθμό μετάδοσης(low-rate data services).

1.2.1 Η αρχή της κυψέλης

Οι ζώνες συχνοτήτων που έχει στη διάθεση του το δίκτυο GSM αντιπροσωπεύουν ένα μικρό μέρος του συνολικού φάσματος και αυτό τις καθιστά εξαιρετικά πολύτιμες. Για την καλύτερη αξιοποίηση των διαθέσιμων πόρων χρησιμοποιούνται συγκεκριμένες τεχνικές.

Αρχικά, γίνεται διαχωρισμός της περιοχής κάλυψης του δικτύου σε κυψέλες και ακολουθεί η τεχνική της επαναχρησιμοποίησης των διαθέσιμων συχνοτήτων. Κάθε κυψέλη περιέχει ένα σταθερό πομποδέκτη που ονομάζεται Πομποδέκτης Σταθμού Βάσης ή Σταθμός Βάσης (Base Transceiver Station συντομότερα BTS ή Base Station και πιο σύντομα BS).

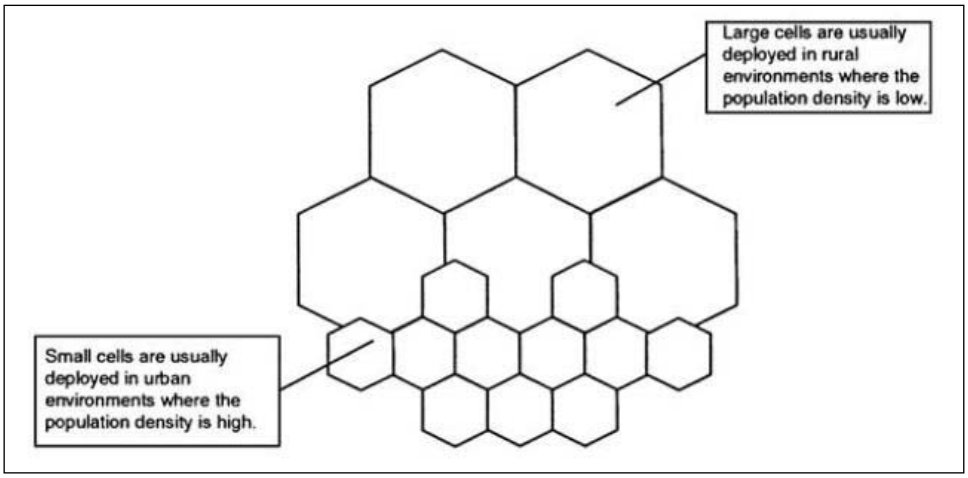
Ο BTS επικοινωνεί με τις φορητές συσκευές που βρίσκονται εντός της περιοχής κάλυψης και είναι το σημείο εισόδου προς το υπόλοιπο δίκτυο. Στην πραγματικότητα, ο BTS καλύπτει μια κυκλική περιοχή, αλλά για περισσότερη ευκολία και καλύτερη οργάνωση φανταζόμαστε ότι η καλυπτόμενη περιοχή έχει το σχήμα ενός εξαγώνου, παρόμοιο με αυτό της κυψέλης.



Εικόνα 5: Η αληθινή όψη ενός δικτύου κυψέλης σε σύγκριση με τη φανταστική όψη

Η ακτίνα κάλυψης ενός BTS είναι ανάλογη με την ισχύ εκπομπής του σταθμού. Στην περίπτωση που χρησιμοποιούνται πολλές κυψέλες μικρής ακτίνας τότε έχουμε περισσότερες φορές επαναχρησιμοποίηση συχνοτήτων, άρα έχουμε και καλύτερη αξιοποίηση των διαθέσιμων πόρων.

Συνήθως σε αστικές περιοχές που έχουν μεγαλύτερη πυκνότητα πληθυσμού χρησιμοποιούνται κυψέλες μικρής ακτίνας. Ενώ σε αγροτικές περιοχές που έχουν λιγότερο πληθυσμό προτιμώνται οι κυψέλες μεγάλης ακτίνας.

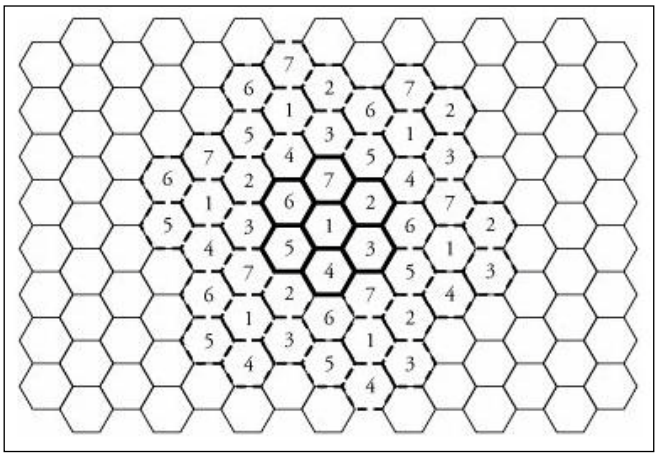


Εικόνα 6: Το μέγεθος μιας κυψέλης προσαρμόζεται ανάλογα με τις ανάγκες

Η επαναχρησιμοποίηση συχνοτήτων γίνεται με το μοίρασμα των συχνοτήτων σε ομάδες. Μόνο μία ομάδα συχνοτήτων θα χρησιμοποιείται από μία κυψέλη. Όλες οι συχνότητες μοιράζονται σε γειτονικές κυψέλες έχοντας σαν στόχο τον σχηματισμό συστάδων(clusters).

Με σωστή τοποθέτηση των clusters επιτυγχάνεται η καλύτερη δυνατή επαναχρησιμοποίηση συχνοτήτων και η ταυτόχρονη εξάλειψη παρεμβολών από τις γειτονικές κυψέλες. Συνήθως δημιουργούνται clusters από 4,7 ή 12 κυψέλες.

Στην απεικόνιση που ακολουθεί έχουμε την περίπτωση όπου το διαθέσιμο εύρος συχνοτήτων έχει χωριστεί σε 7 μέρη. Κάθε ομάδα συχνοτήτων αντιστοιχεί σε μία κυψέλη, οπότε δημιουργούνται clusters των εφτά κυψελών που είναι τοποθετημένα με τρόπο που να δημιουργεί μια απόσταση δύο κυψελών ανάμεσα σε κυψέλες που χρησιμοποιούν κοινές συχνότητες. Χάρις σε αυτή την απόσταση επιτυγχάνεται η απόλυτη εξάλειψη των παρεμβολών.



Εικόνα 7: Επαναχρησιμοποίηση συχνοτήτων σε ένα δίκτυο κυψέλης

1.2.2 Η αρχιτεκτονική του GSM

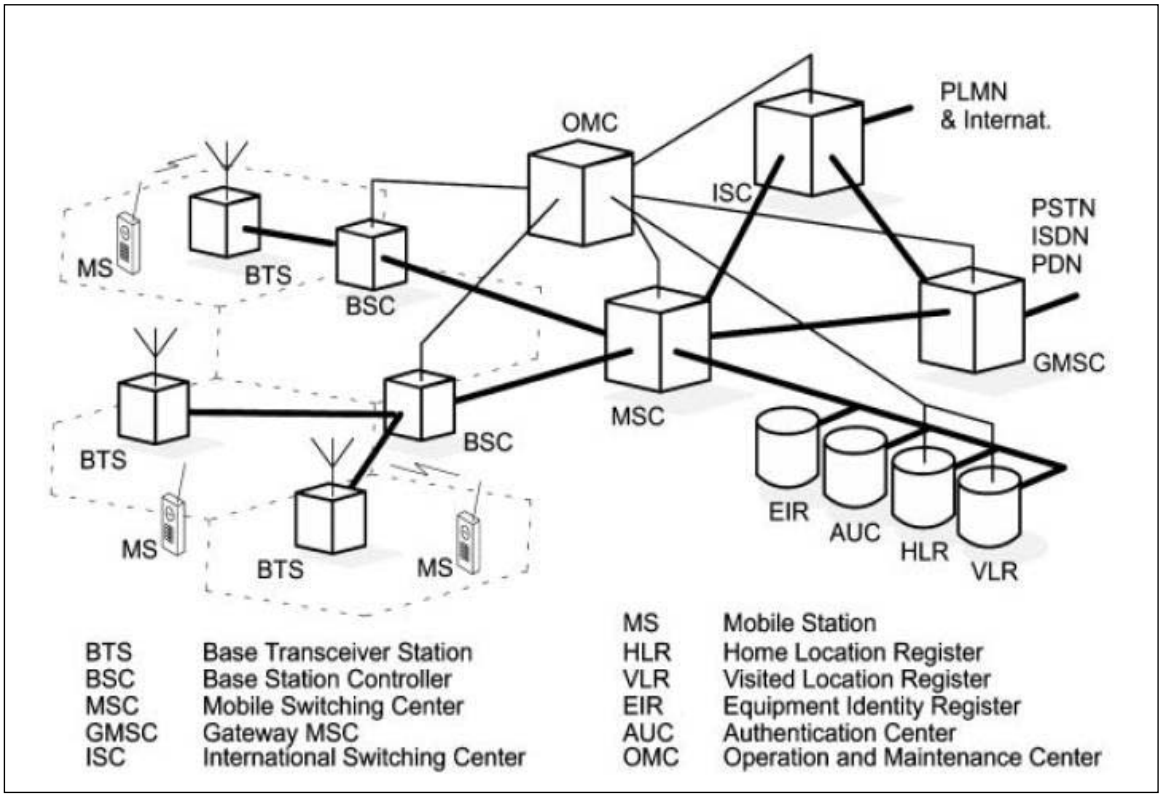
Το GSM χωρίζεται σε δύο τμήματα, με το ένα τμήμα να είναι το σταθερό δίκτυο που αποτελεί την υποδομή και το άλλο τμήμα να είναι οι Κινούμενοι Συνδρομητές που χρησιμοποιούν τις υπηρεσίες που παρέχονται από το δίκτυο.

Το σταθερό τμήμα χωρίζεται με την σειρά του σε τρία υποσυστήματα:

- το Υποσύστημα Σταθμού Βάσης
- το Υποσύστημα Δικτύου και Μεταγωγής
- το Υποσύστημα Υποστήριξης Λειτουργίας

Ένας συνδρομητής αποκτά πρόσβαση στο δίκτυο μέσω του Κινητού Σταθμού (Mobile Station - MS) που χρησιμοποιεί.

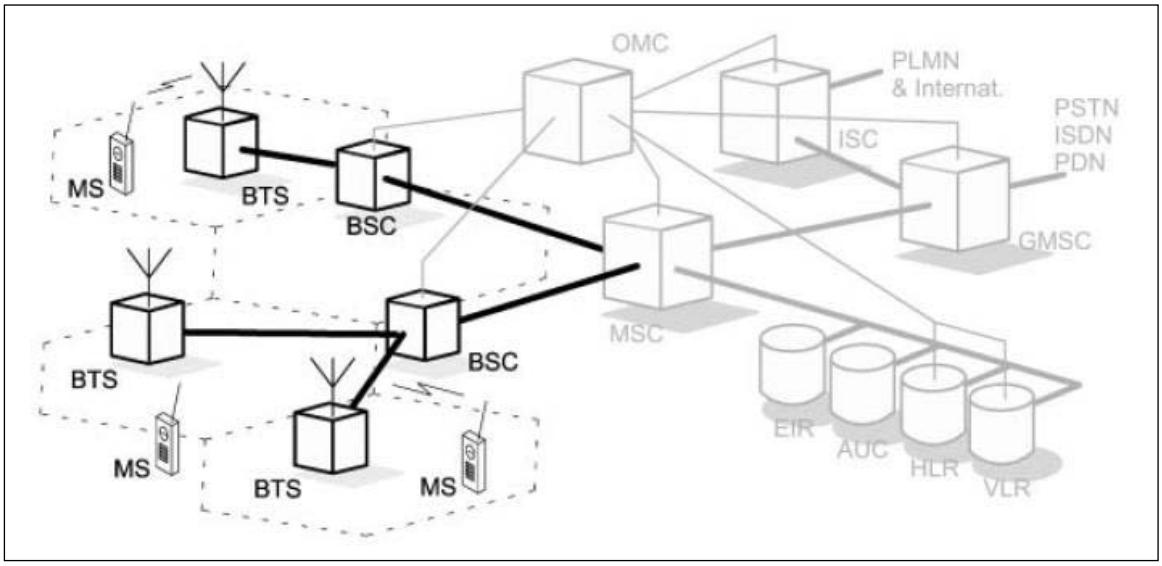
Στο σχήμα που ακολουθεί έχουμε μιαν ολοκληρωμένη απεικόνιση της αρχιτεκτονικής του δικτύου GSM.



Εικόνα 8: Η αρχιτεκτονική του δικτύου GSM

Το Υποσύστημα Σταθμού Βάσης

Το Υποσύστημα Σταθμού Βάσης (Base Station Subsystem – BSS) περιλαμβάνει τις οντότητες του Σταθμού Πομποδέκτη Βάσης (Base Transceiver Station – BTS) και του Ελεγκτή Σταθμού Βάσης (Base Station Controller – BSC).



Εικόνα 9: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Σταθμού Βάσης

Το BSS αναλαμβάνει την εκπομπή και λήψη ράδιο-σημάτων και συνδέει τον Κινητό Σταθμό με το Υποσύστημα Δικτύου και Μεταγωγής.

Όπως αναφέραμε νωρίτερα κάθε κυψέλη έχει ένα BTS που αποτελείται από κεραίες που διεξάγουν εκπομπή και λήψη ράδιο-σημάτων. Ακόμη, ένα BTS συνδέεται σε ένα BSC και έχει την ευθύνη για την παροχή των κατάλληλων καναλιών για πρόσβαση των χρηστών στο δίκτυο και για την επικοινωνία με άλλους συνδρομητές.

Ένα BSC έχει υπό τον έλεγχο του δεκάδες BTSs και περιλαμβάνει συσκευές επεξεργασίας σήματος και ελέγχου του δικτύου, ταυτόχρονα όμως, αποτελεί το συνδετικό κρίκο μεταξύ του Υποσυστήματος Σταθμού Βάσης και του Υποσυστήματος Δικτύου και Μεταγωγής.

Στις αρμοδιότητες του BSC περιλαμβάνονται:

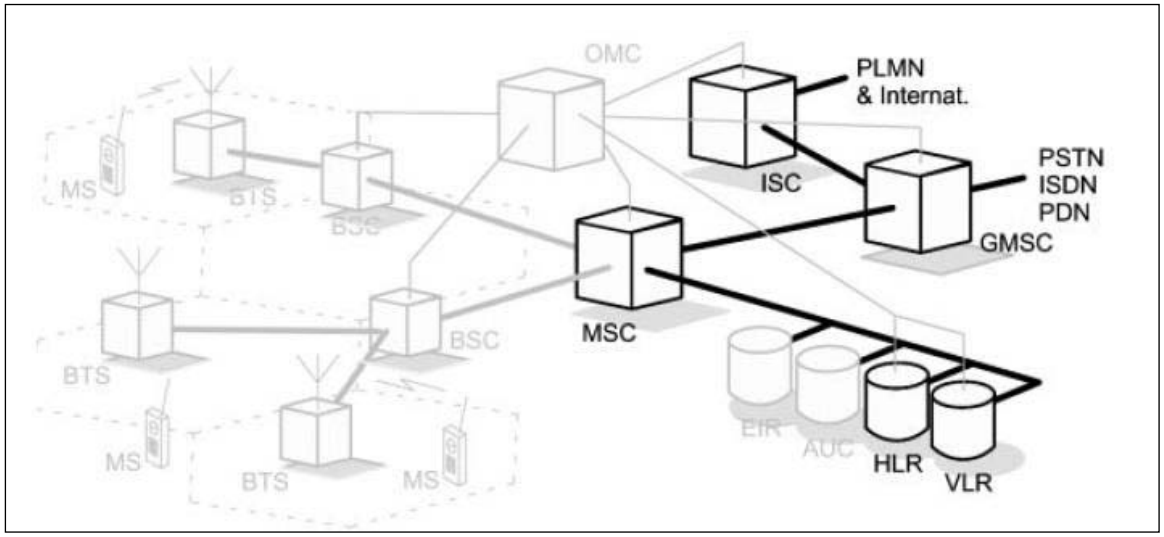
- η ανάθεση συχνοτήτων στα BTSs που έχει υπό τον έλεγχο του
- η μεταφορά κίνησης με κατεύθυνση από τα BTSs προς το Υποσύστημα Δικτύου και Μεταγωγής και αντίστροφα
- η εποπτεία της διαδικασίας της μεταπομπής συνδρομητή(Handover)

Η μεταπομπή ενός συνδρομητή πραγματοποιείται όταν εκείνος βρίσκεται σε κίνηση και αλλάζει κυψέλη, οπότε τότε αλλάζει και το BTS που είναι υπεύθυνο γι' αυτόν και του παρέχει κανάλι επικοινωνίας, έτσι το υπεύθυνο BSC κάνει όλες τις απαραίτητες ενέργειες για να έχει ο συνδρομητής κάποιο διαθέσιμο κανάλι επικοινωνίας στην κυψέλη που εισέρχεται και παράλληλα απελευθερώνει το κανάλι που χρησιμοποιείτο στην προηγούμενη κυψέλη.

Όλοι οι BSCs συνδέονται σε ένα Κέντρο Μεταγωγής Κινητών Υπηρεσιών (Mobile services Switching Centre – MSC), το οποίο αποτελεί το πιο σημαντικό στοιχείο το Υποσυστήματος Δικτύου και Μεταγωγής.

Το Υποσύστημα Δικτύου και Μεταγωγής

Στο Υποσύστημα Δικτύου και Μεταγωγής (Network & Switching Subsystem – NSS) πραγματοποιείται η διαχείριση των κλήσεων. Επίσης, το NSS παρέχει μια σειρά από βάσεις δεδομένων που αποσκοπούν στον έλεγχο των συνδρομητών και στην παροχή υπηρεσιών Roaming.



Εικόνα 10: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Δικτύου και Μεταγωγής

Το NSS αποτελείται από:

- το Κέντρο Μεταγωγής Κινητών Υπηρεσιών (Mobile services Switching Centre – MSC)
- την Πύλη Μεταγωγής Κινητών Υπηρεσιών (Gateway of Mobile services Switching Centre – GMSC)
- το Διεθνές Κέντρο Μεταγωγής (International Switching Centre – ISC)
- το Μητρώο Εγχώριων Συνδρομητών (Home Location Register – HLR)
- το Μητρώο Επισκεπτόμενων Συνδρομητών (Visitor Location Register – VLR)

Το MSC είναι η καρδιά του NSS, εκεί διεκπεραιώνονται όλες οι απαραίτητες λειτουργίες μεταγωγής του δικτύου και γίνεται η διασύνδεση με άλλα MSCs και με άλλους τύπους δικτύων, όπως είναι τα δίκτυα PSTN και ISDN.

Η διασύνδεση με άλλους τύπους δικτύων γίνεται μέσα από το GMSC, ενώ η διασύνδεση με άλλα δίκτυα κινητής τηλεφωνίας ή δίκτυα ξένων χωρών γίνεται με τη βοήθεια του ISC.

Συνήθως το GMSC παρέχεται με μορφή λογισμικού και βρίσκεται ενσωματωμένο σε ένα MSC. Ακόμη, το MSC μπορεί να έχει ενσωματωμένο και το Κέντρο για SMS (SMS Centre – SMSC). Θα αναφερθούμε στη λειτουργία του SMSC αργότερα όταν και θα εξετάσουμε τον τρόπο λειτουργίας των μηνυμάτων SMS.

Το HLR είναι μια βάση δεδομένων, όπου έχει εγγραφές για κάθε συνδρομητή που ανήκει στο δίκτυο. Στο HLR αποθηκεύονται στατικές πληροφορίες, όπως είναι:

- ο αριθμός τηλεφώνου του συνδρομητή
- ο τύπος της συνδρομής
- οι υπηρεσίες στις οποίες παρέχεται πρόσβαση

Ενώ αποθηκεύονται και πληροφορίες που ανανεώνονται δυναμικά, όπως είναι η θέση που βρίσκεται ο συνδρομητής. Συνήθως υπάρχει ένα κεντρικό HLR σε κάθε δίκτυο GSM.

Το VLR είναι μια βάση δεδομένων που χρησιμοποιείται για την διαχείριση των Roaming συνδρομητών. Συνήθως ένα VLR αναπτύσσεται μαζί με ένα MSC, έτσι περιέχει πληροφορίες για τους επισκέπτες-συνδρομητές που βρίσκονται εντός της περιοχής που έχει υπό τον έλεγχο του το MSC με το οποίο γίνεται η σύνδεση.

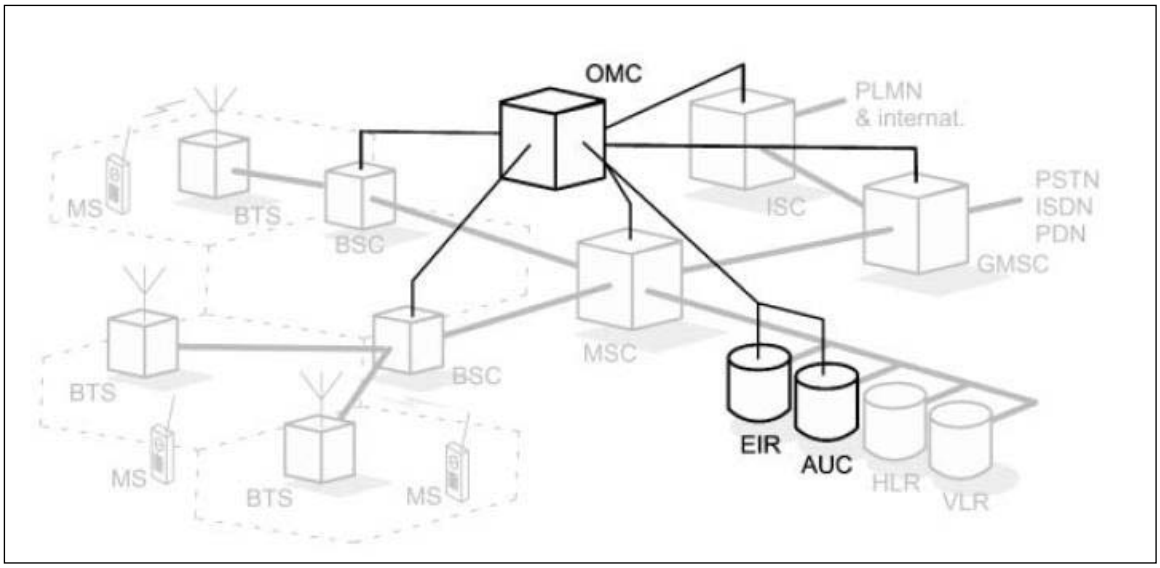
Η ενημέρωση του VLR γίνεται με τον ακόλουθο τρόπο. Όταν ένας συνδρομητής εμφανιστεί στην περιοχή κάλυψης ενός ξένου MSC, τότε από το ξένο δίκτυο θα αποσταλεί προς το υπεύθυνο δίκτυο, και πιο συγκεκριμένα στο υπεύθυνο HLR, ένα αίτημα για μετάδοση των πληροφοριών που αφορούν τον επισκέπτη-συνδρομητή.

Αυτές οι πληροφορίες αποθηκεύονται στο VLR του ξένου δικτύου. Έτσι για όσο διάστημα ο επισκέπτης βρίσκεται στο χώρο ευθύνης του ξένου MSC θα απολαμβάνει τις προβλεπόμενες υπηρεσίες χωρίς να απαιτείται κάθε φορά η άντληση σχετικών πληροφοριών από το HLR και το δίκτυο που ανήκει ο συνδρομητής.

Το Υποσύστημα Υποστήριξης Λειτουργίας

Το Υποσύστημα Υποστήριξης Λειτουργίας (Operation Support Subsystem – OSS) αποτελείται από:

- το Κέντρο Λειτουργίας και Συντήρησης (Operation and Maintenance Centre – OMC)
- το Κέντρο Πιστοποίησης Ταυτότητας (Authentication Centre – AuC)
- το Μητρώο Ταυτότητας Εξοπλισμού (Equipment Identity Register – EIR)



Εικόνα 11: Με έντονο χρώμα απεικονίζεται το Υποσύστημα Υποστήριξης Λειτουργίας

Το OMC είναι η κεντρική μονάδα ελέγχου και εποπτείας του δικτύου, καθώς συνδέεται με στοιχεία τόσο από το Υποσύστημα Σταθμού Βάσης όσο και το Υποσύστημα Δικτύου και Μεταγωγής, παρατηρεί την κατάσταση στην οποία βρίσκονται και εγγυάται για την καλύτερη δυνατή λειτουργία και απόδοση του δικτύου.

Στις αρμοδιότητες του OMC περιλαμβάνονται:

- η διαχείριση των συνδρομητών και του εξοπλισμού
- η διαχείριση των χρεώσεων
- η εξαγωγή στατιστικών στοιχείων που σχετίζονται με την κατάσταση και το φόρτο κίνησης που έχουν όλα τα συστατικά μέρη του δικτύου

Η ασφάλεια του GSM βασίζεται σε ένα μεγάλο βαθμό στην εξακρίβωση της ταυτότητας ενός συνδρομητή και στον έλεγχο του εξοπλισμού που χρησιμοποιείται. Γι' αυτόν ακριβώς το λόγο το AuC και το EIR διαδραματίζουν σημαντικό ρόλο για την επίτευξη της ασφάλειας.

Στο AuC εκτελούνται αλγόριθμοι ασφαλείας, ενώ δημιουργούνται και φυλάσσονται δεδομένα που σχετίζονται με την ασφάλεια, όπως είναι το μυστικό Κλειδί Πιστοποίησης Ταυτότητας (Authentication Key - K_i) των συνδρομητών.

Σε ένα από τα ακόλουθα κεφάλαια θα εξετάσουμε πιο αναλυτικά τους αλγόριθμους που χρησιμοποιούνται για να ενισχύσουν την ασφάλεια του GSM.

Ο έλεγχος εγκυρότητας όλων των συσκευών που συνδέονται στο δίκτυο πραγματοποιείται μέσω του EIR, το οποίο είναι μια κεντρική βάση δεδομένων που περιέχει έναν κατάλογο με όλους τους αριθμούς συνδρομητών και τα στοιχεία ταυτότητας των συσκευών που τους αντιστοιχούν.

Κάθε συσκευή που είναι συμβατή με το δίκτυο GSM έχει έναν μοναδικό αριθμό ταυτότητας που παρέχεται από την κατασκευάστρια εταιρεία και λέγεται Διεθνής Ταυτότητα Κινητού Εξοπλισμού (International Mobile Equipment Identity – IMEI).

Το EIR περιλαμβάνει τρεις λίστες στις οποίες καταχωρούνται οι αριθμοί IMEI που αντιστοιχούν σε όσες συσκευές συμμετέχουν στον δίκτυο:

- τη Λευκή λίστα
- τη Μαύρη λίστα
- τη Γκρι λίστα

Όσοι αριθμοί IMEI βρίσκονται στην Λευκή λίστα αντιπροσωπεύουν συσκευές που είναι απολύτως νόμιμες και έχουν δικαίωμα πρόσβασης στο δίκτυο.

Στη Γκρι λίστα βρίσκονται οι αριθμοί IMEI που ανήκουν σε συσκευές που είναι πιθανό να έχουν κάποιο τεχνικό πρόβλημα, οπότε λαμβάνουν τις υπηρεσίες του δικτύου αλλά βρίσκονται υπό επιτήρηση.

Τέλος, όσες συσκευές βρίσκονται στη Μαύρη λίστα δεν έχουν δικαίωμα πρόσβασης στο δίκτυο, αφού στη λίστα αυτή συνήθως βρίσκονται οι αριθμοί IMEI που αντιστοιχούν σε συσκευές που έχουν κλαπεί ή έχουν χαθεί.

Ο Κινητός Σταθμός

Ο Κινητός Σταθμός (Mobile Station – MS) είναι το μέσο με το οποίο οι συνδρομητές έχουν πρόσβαση στις υπηρεσίες του δικτύου GSM.

Ένας Κινητός Σταθμός αποτελείται από δύο στοιχεία, το πρώτο από αυτά είναι ο Κινητός Εξοπλισμός (Mobile Equipment – ME), που δεν είναι τίποτε άλλο από τη συσκευή κινητού τηλεφώνου που είναι συμβατή με το δίκτυο.

Το δεύτερο και σημαντικότερο στοιχείο είναι η Μονάδα Ταυτότητας Συνδρομητή (Subscriber Identity Module – SIM).

Η Μονάδα Ταυτότητας Συνδρομητή εγκαθίσταται σε ειδική υποδοχή της συσκευής Κινητού Εξοπλισμού και λειτουργεί σαν κλειδί που επιτρέπει την πρόσβαση στις υπηρεσίες του GSM.

Η συγκεκριμένη ταυτότητα παρέχεται στο χρήστη από τον φορέα κινητής τηλεφωνίας και έχει τη μορφή Έξυπνης Κάρτας(Smart Card), γι’ αυτό και πολύ συχνά η Μονάδα Ταυτότητας Συνδρομητή λέγεται και Κάρτα SIM(SIM Card).

Η Κάρτα SIM έχει αποθηκευμένα τα στοιχεία που αφορούν τον συνδρομητή, όπως είναι:

- η ταυτότητα του, που λέγεται και Διεθνής Ταυτότητα Κινητού Συνδρομητή (International Mobile Subscriber Identity – IMSI)
- ο αριθμός τηλεφώνου που του αντιστοιχεί, που είναι γνωστός και με την ονομασία ISDN Αριθμός Κινητού Συνδρομητή (Mobile Subscriber ISDN Number – MSISDN)
- το μυστικό Κλειδί Πιστοποίησης Ταυτότητας (Authentication Key - K_i)

Στην Κάρτα SIM εκτελούνται αλγόριθμοι που σχετίζονται με την ασφάλεια και την επαλήθευση της ταυτότητας του συνδρομητή από στοιχεία του δικτύου όπως είναι το AuC.

Ακόμη, η Κάρτα SIM έχει αποθηκευμένες ρυθμίσεις από τον φορέα κινητής τηλεφωνίας, ενώ δίνει και στον συνδρομητή την ευχέρεια να αποθηκεύσει προσωπικά του δεδομένα όπως είναι ο τηλεφωνικός του κατάλογος ή τα γραπτά του μηνύματα.

Το γεγονός αυτό είναι αρκετά σημαντικό, αφού δίνει στον συνδρομητή την δυνατότητα φορητότητας του αριθμού και των δεδομένων του, δηλαδή ο συνδρομητής μπορεί να εγκαταστήσει την Κάρτα SIM σε όποια έγκυρη Μονάδα Εξοπλισμού επιθυμεί και παράλληλα διατηρεί τον τηλεφωνικό του αριθμό και τα δεδομένα του χωρίς να δεσμεύεται από την συσκευή που χρησιμοποιεί.

Για λόγους ασφαλείας η Κάρτα SIM, ή και ολόκληρος ο Κινητός Σταθμός, επιτρέπουν την χρήση τους μόνο μετά από την επιτυχημένη εισαγωγή ενός τετραψήφιου κωδικού που ονομάζεται Προσωπικός Αριθμός Ταυτότητας (Personal Identification Number – PIN).

1.3 Το GPRS

Το δίκτυο Global System for Mobile communications(GSM) παρείχε τις υπηρεσίες του χρησιμοποιώντας συνδέσεις με μεταγωγή κυκλώματος (Circuit-Switched Connections).

Το General Packet Radio Service (GPRS) παρουσιάστηκε σαν μια προσθήκη του δικτύου GSM που επιτρέπει στους συνδρομητές την αποστολή και λήψη δεδομένων μέσα από συνδέσεις μεταγωγής πακέτων (Packet-Switched Connections). Η χρήση του GPRS είναι κατάλληλη για υπηρεσίες όπως είναι η ασύρματη πρόσβαση στο Internet.

1.3.1 Η αρχιτεκτονική του δικτύου GSM μετά την ενσωμάτωση του GPRS

Η ενσωμάτωση του GPRS στο δίκτυο GSM έχει μικρό αντίκτυπο στην ήδη υπάρχουσα αρχιτεκτονική.

Οι αλλαγές που πραγματοποιούνται είναι στο Υποσύστημα Σταθμού Βάσης και στο Υποσύστημα Δικτύου και Μεταγωγής.

Τα στοιχεία του Υποσυστήματος Σταθμού Βάσης, δηλαδή ο Πομποδέκτης Σταθμού Βάσης και ο Ελεγκτής Σταθμού Βάσης, εκσυγχρονίζονται για να είναι σε θέση να διαχειρίζονται την ασύρματη μεταγωγή πακέτων.

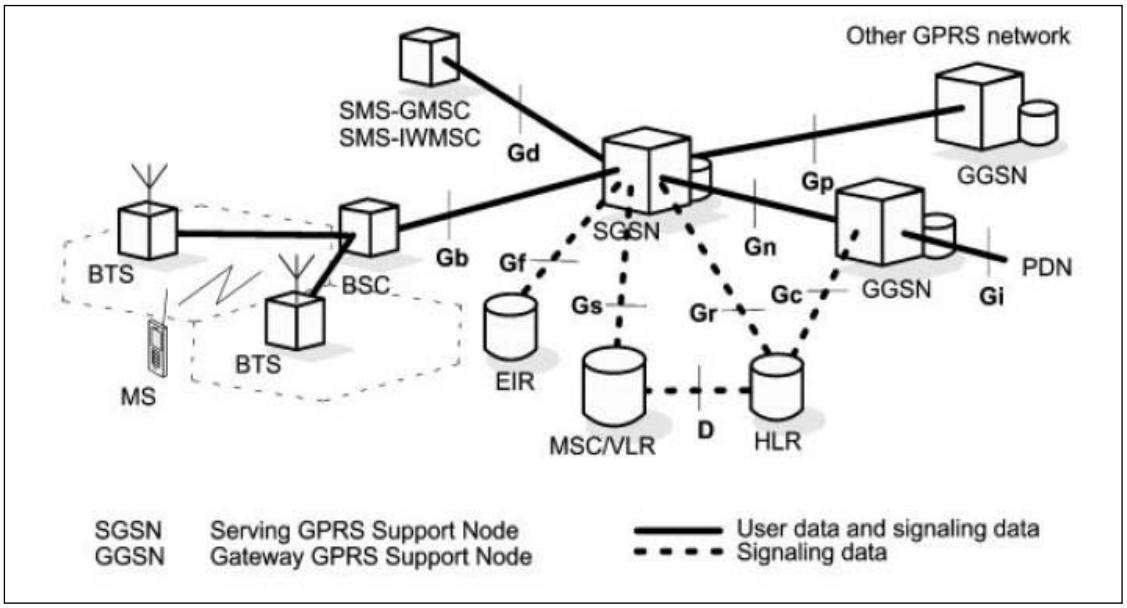
Στο Υποσύστημα Δικτύου και Μεταγωγής αναβαθμίζονται οι βάσεις των HLR και VLR για να δέχονται πρόσθετες πληροφορίες που αφορούν την επικοινωνία μέσω

GPRS, ενώ προστίθενται και δύο καινούργιες οντότητες που ονομάζονται Κόμβοι Υποστήριξης GPRS (GPRS Support Nodes – GSNs).

Οι δύο νέες οντότητες έχουν την ευθύνη για την δρομολόγηση και την παράδοση πακέτων δεδομένων ανάμεσα σε κινητούς σταθμούς και σε εξωτερικά δίκτυα με πακέτα δεδομένων (Packet Data Networks – PDNs).

Στην ακόλουθη εικόνα παρουσιάζεται η αρχιτεκτονική του δικτύου GSM όπως προκύπτει μετά την ενσωμάτωση των δύο νέων οντοτήτων που υποστηρίζουν το GPRS. Οι νέες οντότητες είναι:

- ο κόμβος Serving GPRS Support Node (SGSN)
- η πύλη Gateway GPRS Support Node (GGSN)



Εικόνα 12: Η αρχιτεκτονική του δικτύου GSM μετά από την ενσωμάτωση του GPRS

Ο κόμβος SGSN συνδέεται σε ένα ή περισσότερα Υποσυστήματα Σταθμού Βάσης και λειτουργεί σαν δρομολογητής(router) πακέτων από και προς όλους τους κινητούς σταθμούς που βρίσκονται στην περιοχή που έχει υπό την ευθύνη του.

Επιπλέον, ο κόμβος διατηρεί δεδομένα για την θέση και το προφίλ των συνδρομητών, ελέγχει την σύνδεση και την αποσύνδεση των συνδρομητών από το GPRS, ενώ πραγματοποιεί και πιστοποίηση της ταυτότητας των συνδρομητών.

Η πύλη GGSN είναι το σημείο διασύνδεσης του δικτύου με άλλα PDNs και αναλαμβάνει την μετατροπή και την προώθηση των πακέτων που φτάνουν σε αυτήν.

Στην περίπτωση που ένα πακέτο έρχεται από το SGSN, τότε η πύλη αναλαμβάνει την μετατροπή στο format που αντιστοιχεί στο πρωτόκολλο που χρησιμοποιεί το εξωτερικό δίκτυο, αν για παράδειγμα χρησιμοποιείται IP ή X.25 πρωτόκολλο. Στη συνέχεια η πύλη προωθεί το τροποποιημένο πακέτο προς το κατάλληλο εξωτερικό δίκτυο.

Όταν η πύλη δέχεται ένα πακέτο από ένα εξωτερικό δίκτυο δεδομένων, τότε το μετατρέπει σε πακέτο GPRS, το οποίο μετά το προωθεί στο SGSN που έχει υπό την

ευθύνη του τον κινητό σταθμό του παραλήπτη. Γι' αυτό το σκοπό η πύλη GGSN διατηρεί μια βάση δεδομένων με τους συνδρομητές και το SGSN στο οποίο συνδέονται.

1.4 Το δίκτυο UMTS

Η έρευνα και η ανάπτυξη του δικτύου Universal Mobile Telecommunications System(UMTS) ξεκίνησε στην Ευρωπαϊκή Ένωση. Αρχικά, το UMTS προοριζόταν για τη διαδοχή του δικτύου δεύτερης γενιάς GSM. Όμως τελικά αποτέλεσε την πρόταση της Ευρωπαϊκής Ένωσης για τα δίκτυα κινητών τηλεπικοινωνιών τρίτης γενιάς που περιλαμβάνονται στην οικογένεια προτύπων International Mobile Telecommunications – 2000 (IMT - 2000).

Η θεμελίωση των προδιαγραφών του UMTS ανατέθηκε στο Third Generation Partnership Project(3GPP), το οποίο είναι μια ομάδα συνεργασίας Ευρωπαϊκών, Αμερικανικών και Ασιατικών φορέων που σχετίζονται με την προτυποποίηση των τηλεπικοινωνιών.

Το 3GPP δημιουργήθηκε το 1998 και από τότε εκδίδει σε τακτά χρονικά διαστήματα νέες προδιαγραφές για το UMTS με τις οποίες προσθέτει νέα στοιχεία ή αναθεωρεί κάποια από τα παλαιότερα.

Οι πρώτες προδιαγραφές που έγιναν διαθέσιμες από το 3GPP είχαν την ονομασία Release 1999, ή συντομότερα Rel.99, αποτέλεσαν τη βάση ανάπτυξης και καθόρισαν την αρχική αρχιτεκτονική του δικτύου UMTS καθώς και τις υπηρεσίες που θα παρέχονται μέσα από αυτό.

Σύμφωνα με τις προδιαγραφές Rel.99, οι ρυθμοί μεταφοράς δεδομένων που προσφέρει το UMTS εξαρτώνται άμεσα από το βαθμό κινητικότητας του συνδρομητή και από την περιοχή στην οποία βρίσκεται. Έτσι έχουμε τους ακόλουθους ρυθμούς μετάδοσης δεδομένων:

- 144 kbps, για συνδρομητές που κινούνται σε εξωτερικό χώρο με μεγάλη ταχύτητα ή βρίσκονται σε αγροτική περιοχή
- 384 kbps, για συνδρομητές που κινούνται σε εξωτερικό χώρο με μικρή ταχύτητα και βρίσκονται σε αστική περιοχή
- 2 Mbps, για συνδρομητές που βρίσκονται σε αστική περιοχή, σε εσωτερικό χώρο και μετακινούνται ελάχιστα

Οι υπηρεσίες που παρέχονται από το UMTS χωρίζονται σε τέσσερις κατηγορίες, οι οποίες δημιουργούνται έχοντας σαν κριτήριο διαχωρισμού τα χαρακτηριστικά της κίνησης και την ποιότητα υπηρεσίας (Quality of Service – QoS) που απαιτείται από το δίκτυο.

Ακολουθούν οι κλάσεις υπηρεσιών και μερικά παραδείγματα από τις υπηρεσίες που αντιστοιχούν σε κάθε κατηγορία:

- Conversational class – Μετάδοση φωνής, Βίντεο κλήση(Video Telephony)
- Streaming class – Μετάδοση ροών πολυμέσων, λήψη Video on Demand
- Interactive class – Πλοήγηση στο Internet, δικτυακά παιχνίδια

- Background class – Μηνύματα SMS και MMS, E-mail, λήψη αρχείων(file downloading)

Οι προδιαγραφές που κυκλοφόρησαν τα επόμενα χρόνια και διαδέχθηκαν το Rel.99 εισήγαγαν αρκετές καινοτομίες, επέφεραν σημαντικές αλλαγές στον τομέα της αρχιτεκτονικής του δικτύου και έδωσαν τη δυνατότητα για την προσθήκη νέων υπηρεσιών.

Το μεγαλύτερο ενδιαφέρον από την πλευρά της αρχιτεκτονικής του δικτύου παρουσιάζεται στις προδιαγραφές Release 4 και Release 5.

Στο Release 4 προετοιμάζεται το έδαφος για τη χρήση του πρωτοκόλλου IP σε όλους τους τομείς του δικτύου.

Με το Release 5 ολοκληρώνεται η μετατροπή του UMTS σε All-IP δίκτυο, ενώ παράλληλα έχουμε την εμφάνιση του υποσυστήματος IP Multimedia Subsystem(IMS) και της τεχνικής High Speed Downlink Packet Access(HSDPA).

Ακολουθεί το Release 6, όπου εμπλουτίζεται το IMS με νέες υπηρεσίες, τα ασύρματα τοπικά δίκτυα(Wireless Local Area Networks - WLANs) θεωρούνται τμήμα του δικτύου και ενσωματώνονται στο UMTS, ενώ προστίθεται και η τεχνική Enhanced Uplink(EUL) ή High Speed Uplink Packet Access(HSUPA).

Το Release 7 αποτελεί την πιο πρόσφατη έκδοση των προδιαγραφών της ομάδας 3GPP, όπου περιλαμβάνονται τεχνικές όπως το High Speed Packet Access Evolution(HSPA+), οι οποίες φέρνουν το UMTS πιο κοντά προς τα δίκτυα τέταρτης γενιάς.

Αυτή τη στιγμή βρίσκεται σε εξέλιξη η διαδικασία συγγραφής των προδιαγραφών Release 8, η έκδοση τους αναμένεται μέσα στο 2009. Με το Release 8 μπαίνουμε στην τέταρτη γενιά των δικτύων κινητής τηλεφωνίας, αφού στις νέες προδιαγραφές καθορίζονται οι αλλαγές που απαιτούνται στο UMTS για να μετατραπεί στο Long Term Evolution(LTE), ένα από τα πρώτα δίκτυα τέταρτης γενιάς που αναμένεται να τεθεί σε εμπορική χρήση εντός του 2010.

1.4.1 Η αρχιτεκτονική του UMTS σύμφωνα με τις προδιαγραφές Release 1999

Σύμφωνα με το Release 1999, ή Rel.99, το δίκτυο UMTS χωρίζεται σε τρεις τομείς:

- στο δίκτυο κορμού
- στο δίκτυο ασύρματης διασύνδεσης
- στον εξοπλισμό του χρήστη

<u>Το δίκτυο κορμού</u>

Το δίκτυο κορμού (Core Network – CN) που συναντάται στην πρώτη μορφή του UMTS συνδυάζει τη μεταγωγή κυκλώματος με τη μεταγωγή πακέτου, και

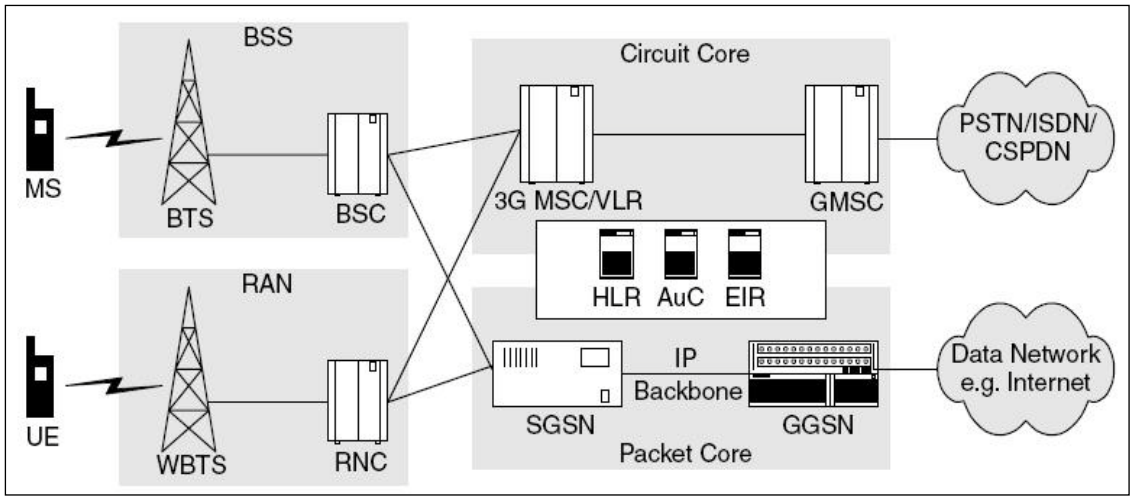
αποτελείται από τις ίδιες οντότητες που βρίσκονται και στο δίκτυο δεύτερης γενιάς GSM και στην αναβάθμιση GPRS.

Αυτό οφείλεται στο γεγονός ότι όλοι οι πάροχοι κινητής τηλεφωνίας στις χώρες της Ευρωπαϊκής Ένωσης είχαν δίκτυα GSM, έτσι ήταν ευκολότερη η εγκατάσταση δικτύων UMTS αφού ένα τμήμα της υποδομής ήταν ήδη έτοιμο.

Επιπλέον, με αυτό τον τρόπο προστατεύονταν οι επενδύσεις που είχαν γίνει σε εξοπλισμό για το GSM, ενώ παράλληλα μειωνόταν το κόστος εγκατάστασης του UMTS, αφού δινόταν η δυνατότητα επαναχρησιμοποίησης δικτυακού εξοπλισμού που υπήρχε εκ των προτέρων.

Η μόνη διαφορά ανάμεσα στο δίκτυο κορμού του UMTS και του GSM/GPRS είναι στο αναβαθμισμένο Mobile services Switching Centre(MSC) ή 3G MSC. Η αναβάθμιση του MSC σε 3G MSC γίνεται με την προσθήκη νέου λογισμικού. Χάρις στο νέο λογισμικό το MSC είναι σε θέση να διαχειρίζεται την κίνηση του UMTS που διαφέρει στην κωδικοποίηση από την κίνηση του GSM/GPRS.

Στο σχήμα που ακολουθεί παρουσιάζεται η αρχιτεκτονική του δικτύου UMTS σύμφωνα με όσα ορίζουν οι προδιαγραφές Rel.99.



Εικόνα 13: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 99

Το δίκτυο ασύρματης διασύνδεσης

Στο παραπάνω σχήμα παρατηρούμε ότι υπάρχει ένα κοινό δίκτυο κορμού, που κληρονομείται από το GSM/GPRS, και συνυπάρχουν δύο διαφορετικά δίκτυα ασύρματης διασύνδεσης. Τα δύο δίκτυα ασύρματης διασύνδεσης είναι:

- το Base Station Subsystem(BSS)
- το Radio Access Network(RAN) ή UMTS Terrestrial Radio Access Network(UTRAN)

Το BSS αποτελεί κληρονομιά του δικτύου GSM. Στο BSS χρησιμοποιείται η τεχνική πολυπλεξίας Time Division Multiple Access(TDMA) και παρέχεται πρόσβαση στους συνδρομητές του δικτύου GSM/GPRS.

Το RAN ή UTRAN χρησιμοποιείται για την πρόσβαση των συνδρομητών στις υπηρεσίες του δικτύου UMTS. Για την επίτευξη των υψηλών ρυθμών μεταφοράς δεδομένων που υπόσχεται το UMTS χρησιμοποιείται μια παραλλαγή της τεχνικής πολυπλεξίας Code Division Multiple Access(CDMA), η τεχνική αυτή λέγεται Wideband Code Division Multiple Access(WCDMA).

Το UTRAN αποτελείται από δύο στοιχεία δικτύου, τον ελεγκτή του δικτύου ράδιο-σημάτων (Radio Network Controller – RNC) και τον πομποδέκτη σταθμό βάσης WCDMA (WCDMA Base Transceiver Station – WBTS). Αν και στις προδιαγραφές 3GPP συνηθίζεται ο σταθμός βάσης WCDMA να ονομάζεται και σαν κόμβος B(Node B) ή να λέγεται πολύ απλούστερα σταθμός βάσης (Base Station – BS).

Ο Node B συνήθως καλύπτει μια ή περισσότερες κυψέλες και είναι το στοιχείο του δικτύου που παρέχει ασύρματη διασύνδεση ανάμεσα στους συνδρομητές και το υπόλοιπο σταθερό δίκτυο.

Στις κυριότερες αρμοδιότητες του Node B περιλαμβάνονται η επικοινωνία με τους συνδρομητές που βρίσκονται στην περιοχή ευθύνης του κόμβου και η μετατροπή της κίνησης που διέρχεται από εκεί.

Η επικοινωνία με τους συνδρομητές γίνεται με τη λήψη και την αποστολή σημάτων κωδικοποιημένων με CDMA.

Η μετατροπή της κίνησης περιλαμβάνει δύο σενάρια. Στη μία περίπτωση ο Node B δέχεται σήματα από τους συνδρομητές και τα μετατρέπει σε πακέτα που αντιστοιχούν στο πρωτόκολλο που χρησιμοποιείται στο δίκτυο κορμού, ενώ στην άλλη περίπτωση μετατρέπει τα πακέτα σε σήματα που εκπέμπει προς τους συνδρομητές.

Στο Rel.99 προτείνεται η μεταφορά δεδομένων με τη χρήση του ασύγχρονου τρόπου μετάδοσης Asynchronous Transfer Mode(ATM).

Το RNC είναι υπεύθυνο για τη διαχείριση των πόρων του UTRAN και τη δρομολόγηση κίνησης από και προς το δίκτυο κορμού.

Σε κάθε ένα RNC συνδέονται ένας ή περισσότεροι Nodes B.

Το RNC αναλαμβάνει μεταξύ άλλων:

- την εκχώρηση καναλιών επικοινωνίας στους κόμβους
- την παρακολούθηση του φορτίου κίνησης στις κυψέλες με τις οποίες συνδέεται
- τον εντοπισμό πιθανών συνθηκών συμφόρησης
- τη διαδικασία της μεταπομπής συνδρομητή(Handover)

Ο εξοπλισμός του χρήστη

Ο εξοπλισμός του χρήστη(User Equipment – UE) επιτρέπει τη σύνδεση με το σταθερό δίκτυο κορμού και παρέχει πρόσβαση στις υπηρεσίες που προσφέρει το UMTS.

Στο δίκτυο GSM/GPRS υπήρχε η ιδέα του φυσικού διαχωρισμού ανάμεσα στον τερματικό εξοπλισμό και την ταυτότητα του συνδρομητή. Η ιδέα αυτή χρησιμοποιείται και στο UMTS, έτσι και εδώ έχουμε τον εξοπλισμό ενός χρήστη του δικτύου UMTS να προκύπτει από τον συνδυασμό του τερματικού εξοπλισμού και της μονάδας ταυτότητας του συνδρομητή.

Ο τερματικός εξοπλισμός (Terminal Equipment - TE) που είναι συμβατός με το δίκτυο UMTS έχει προσαρμοσμένο τον κατάλληλο πομποδέκτη, ο οποίος μέσω της τεχνικής WCDMA επιτρέπει την πρόσβαση στο UTRAN και κατ’ επέκταση στο υπόλοιπο δίκτυο.

Ο τερματικός εξοπλισμός έχει τρεις τρόπους λειτουργίας:

- Συνδέεται μόνο στον τομέα του δικτύου που χρησιμοποιεί μεταγωγή κυκλώματος(Circuit Switching mode) και λαμβάνει μόνο τις αντίστοιχες προβλεπόμενες υπηρεσίες
- Συνδέεται μόνο στον τομέα που χρησιμοποιεί μεταγωγή πακέτου(Packet Switching mode)
- Συνδέεται ταυτόχρονα και στους δύο τομείς του δικτύου

Οι πιο συνηθισμένες μορφές τερματικού εξοπλισμού είναι αυτές του κινητού τηλεφώνου και του φορητού υπολογιστή.

Συνήθως τα κινητά τηλέφωνα χρησιμοποιούν τον τρίτο τρόπο λειτουργίας και συνδέονται ταυτόχρονα στους δύο τομείς του δικτύου, ενώ οι φορητοί υπολογιστές συνδέονται στον τομέα μεταγωγής πακέτου και αποκτούν τη δυνατότητα ασύρματης πρόσβασης σε δίκτυα μεταγωγής πακέτου όπως είναι το Internet.

Η μονάδα ταυτότητας του συνδρομητή λέγεται UMTS Subscriber Identity Module(USIM), παρέχεται με τη μορφή έξυπνης κάρτας(smart card) και συνδέεται με τον τερματικό εξοπλισμό του συνδρομητή. Η κάρτα USIM παρέχει λειτουργίες αντίστοιχες με αυτές της κάρτας SIM στα δίκτυα GSM/GPRS.

Μια από τις χρήσεις της κάρτας είναι για την αποθήκευση δεδομένων, όπως είναι:

- η ταυτότητα του συνδρομητή International Mobile Subscriber Identity(IMSI)
- οι τιμές που έχουν τα κλειδιά κρυπτογράφησης και εξακρίβωσης ταυτότητας
- προσωπικά δεδομένα του συνδρομητή

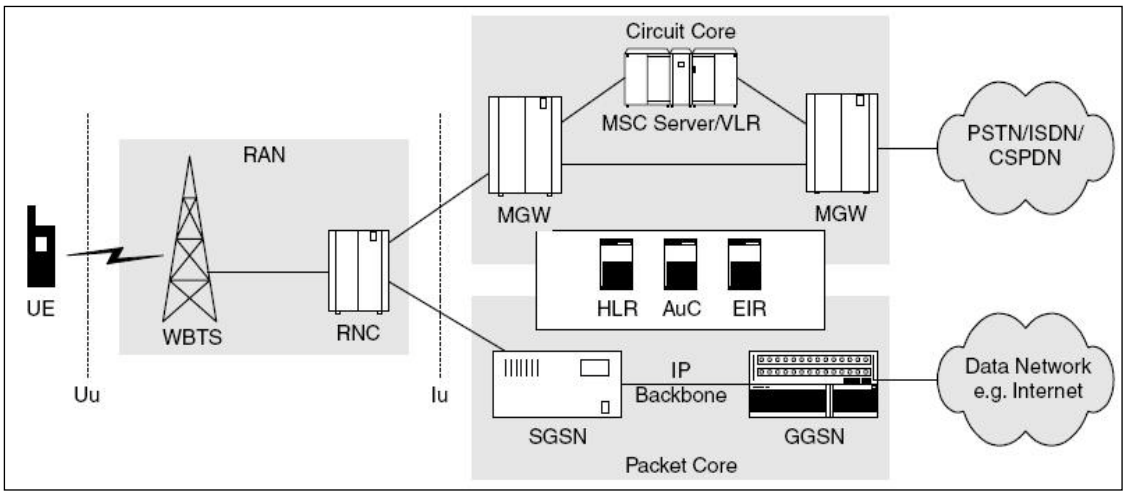
Ακόμη, η κάρτα χρησιμοποιείται και για την εκτέλεση αλγορίθμων που σχετίζονται με την ασφάλεια των συνδρομητών και παρέχει τα στοιχεία που απαιτούνται κατά την εκτέλεση των μηχανισμών ασφαλείας.

1.4.2 Η αρχιτεκτονική του UMTS σύμφωνα με τις προδιαγραφές Release 4

Η υλοποίηση του UMTS σύμφωνα με τις προδιαγραφές Release 99 ήταν στην ουσία μια εξέλιξη του GSM, η οποία είχε δύο δίκτυα ασύρματης πρόσβασης και μετέφερε την κίνηση χρησιμοποιώντας μεταγωγή κυκλώματος και μεταγωγή πακέτου.

Στις προδιαγραφές Release 4, ή R4, εισάγονται κάποιες αλλαγές στην αρχιτεκτονική του δικτύου που στοχεύουν στην σταδιακή ενοποίηση των τομέων μεταγωγής κυκλώματος και μεταγωγής πακέτου, με τελικό στόχο την εξέλιξη του UMTS σε ένα δίκτυο μεταγωγής πακέτου όπου παντού θα χρησιμοποιείται το πρωτόκολλο IP.

Στο σχήμα που ακολουθεί παρουσιάζεται η αρχιτεκτονική του δικτύου UMTS όταν αυτό εναρμονίζεται με τις προδιαγραφές Release 4.



Εικόνα 14: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 4

Οι σημαντικότερες αλλαγές από την πλευρά της αρχιτεκτονικής είναι η αναθεωρημένη στοίβα πρωτοκόλλων του δικτύου που βασίζεται στο IP πρωτόκολλο και ο τροποποιημένος τομέας μεταγωγής κυκλώματος στον οποίο το παραδοσιακό MSC αντικαθίσταται από δύο νέες οντότητες.

Στην προηγούμενη εκδοχή του τομέα μεταγωγής κυκλώματος που υπήρχε στις προδιαγραφές Rel.99 είχαμε το MSC να είναι υπεύθυνο για την διαχείριση των συνδέσεων και τη μεταγωγή της κίνησης.

Στο νέο τομέα μεταγωγής κυκλώματος γίνεται διαχωρισμός των αρμοδιοτήτων του MSC. Με αυτό τον τρόπο δίνονται περισσότερες δυνατότητες κλιμάκωσης του συστήματος, ενώ παράλληλα προετοιμάζεται το έδαφος για τη σύγκλιση του τομέα μεταγωγής κυκλώματος με τον τομέα μεταγωγής πακέτου.

Οι αρμοδιότητες του MSC μοιράζονται σε δύο νέους κόμβους:

- την πύλη Circuit Switched Media Gateway(CS-MGW) ή συντομότερα Media Gateway(MGW)
- τον εξυπηρετητή MSC Server(MSS)

Ο εξυπηρετητής MSS και η πύλη MGW έχουν μια σχέση ένα προς πολλά, δηλαδή ένας εξυπηρετητής MSS συνδέεται με κάποιες δεκάδες από πύλες MGW.

Η πύλη MGW αναλαμβάνει ότι έχει σχέση με τη μεταγωγή κίνησης. Έτσι δέχεται κίνηση τόσο από το δίκτυο ασύρματης πρόσβασης του UMTS όσο και από εξωτερικά δίκτυα μεταγωγής κυκλώματος, όπως είναι το PSTN. Σκοπός της πύλης είναι η μετατροπή της υποδεχόμενης κίνησης στην κατάλληλη μορφή και η δρομολόγηση προς τη σωστή κατεύθυνση.

Ο εξυπηρετητής MSS αναλαμβάνει τη διαχείριση των συνδέσεων, η οποία γίνεται με την αποστολή των κατάλληλων εντολών ελέγχου προς τις πύλες MGW στις οποίες συνδέεται ο MSC Server. Οι εντολές αφορούν την εγκαθίδρυση νέων ή την απελευθέρωση υπαρχόντων συνδέσεων. Μια επιπλέον αρμοδιότητα του εξυπηρετητή σχετίζεται με λειτουργίες που αφορούν την κινητικότητα των συνδρομητών, γι' αυτό και ο MSS έχει ενσωματωμένο ένα VLR και εκτελεί τις κατάλληλες λειτουργίες.

1.4.3 Η αρχιτεκτονική του UMTS σύμφωνα με τις προδιαγραφές Release 5

Με τις προδιαγραφές Release 4 έγινε το πρώτο βήμα για την εξέλιξη του UMTS σε δίκτυο All-IP. Το πρωτόκολλο IP χρησιμοποιείται σε όλο το δίκτυο, εκτός από τον τομέα μεταγωγής κυκλώματος στον οποίο χρησιμοποιείται το πρωτόκολλο ATM.

Με το σύνολο των προδιαγραφών Release 5, ή R5, ολοκληρώνεται η μετατροπή του UMTS σε All-IP δίκτυο, ενώ παράλληλα, εισάγεται και η τεχνική High Speed Downlink Packet Access(HSDPA), η οποία οδηγεί σε μια σημαντική αύξηση του ρυθμού λήψης δεδομένων.

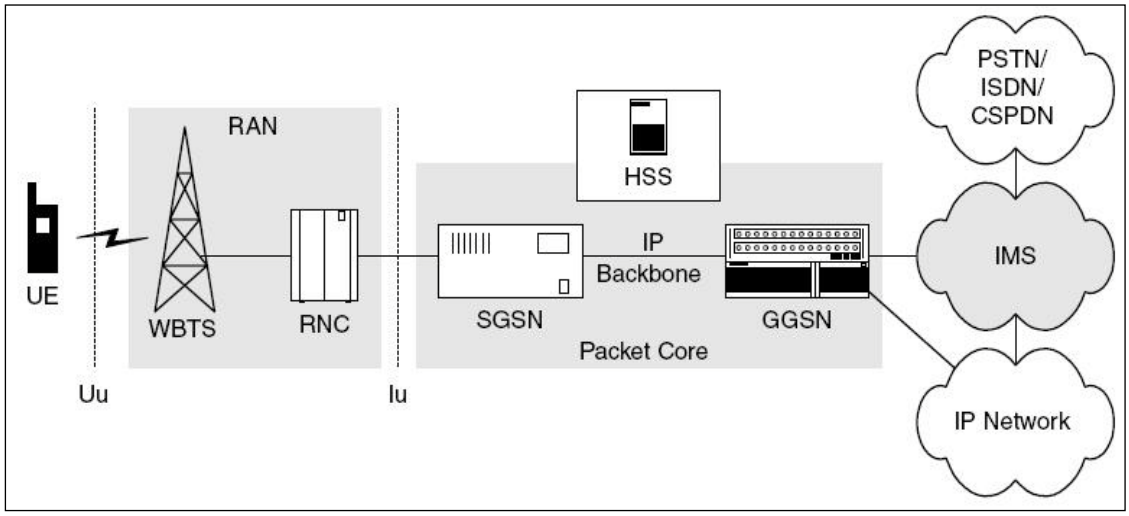
Στόχος του R5 είναι να παρέχει τη δυνατότητα χρησιμοποίησης του πρωτοκόλλου IP σε όλο το μήκος του δικτύου, ενώ παράλληλα γίνεται προσπάθεια για απλούστευση της αρχιτεκτονικής του δικτύου.

Ο συγκεκριμένος στόχος επιτυγχάνεται με την εισαγωγή ενός νέου υποσυστήματος, του IP Multimedia Subsystem(IMS), το οποίο παρέχει όλο το εύρος των υπηρεσιών πολυμέσων πάνω από το πρωτόκολλο IP.

Με το IMS είναι δυνατή η αντικατάσταση των υπηρεσιών που παρείχε ο τομέας μεταγωγής κυκλώματος με ισοδύναμες υπηρεσίες που παρέχονται με τη μεταγωγή πακέτων IP.

Ένα παράδειγμα είναι η μετάδοση φωνής, η οποία παραδοσιακά ήταν διαθέσιμη με μεταγωγή κυκλώματος, όπου στο All-IP δίκτυο η φωνή μεταδίδεται μέσω του πρωτοκόλλου IP και η υπηρεσία λέγεται Voice over IP(VoIP).

Χάρις στην παρουσία του IMS δεν είναι απαραίτητος ο τομέας μεταγωγής κυκλώματος, ενώ αντίθετα είναι απαραίτητος ο τομέας μεταγωγής πακέτου, χωρίς να έχει σημαντικές αλλαγές σε σχέση με το παρελθόν. Όπως φαίνεται και στο σχήμα που ακολουθεί, το δίκτυο αποκτά απλούστερη δομή και ταυτόχρονα χρησιμοποιεί παντού το ίδιο πρωτόκολλο μεταφοράς.



Εικόνα 15: Η αρχιτεκτονική του δικτύου UMTS σύμφωνα με το Release 5

Στον τομέα μεταγωγής κυκλώματος πραγματοποιείται μια αλλαγή σε σχέση με το παρελθόν. Πρόκειται για την ενοποίηση οντοτήτων που είχαν κοινή χρήση σε τομείς και υποσυστήματα που υπήρχαν στις παλαιότερες υλοποιήσεις.

Οι καταχωρητές Home Location Register(HLR), Equipment Identity Register(EIR) και το κέντρο Authentication Centre(AuC) ενώνονται και δημιουργούν την οντότητα Home Subscriber Server(HSS).

Το HSS διατηρεί τις λειτουργίες που είχαν παλαιότερα οι οντότητες που το αποτελούν, με τη μόνη διαφορά ότι ορισμένες είναι πλέον προσαρμοσμένες στις ανάγκες του IMS.

Το UTRAN διατηρεί τις ίδιες οντότητες με αυτές που υπήρχαν σε προηγούμενες υλοποιήσεις. Όμως η πλειονότητα των υπηρεσιών που παρέχονται από το δίκτυο έχει ασύμμετρη μορφή, καθώς υπάρχει περισσότερη κίνηση κατά τη λήψη δεδομένων απ' ότι υπάρχει κατά την αποστολή.

Για την καλύτερη διαχείριση αυτής της κατάστασης πραγματοποιούνται ορισμένες αλλαγές στα κανάλια ράδιο-σημάτων του UTRAN, μέσα από αυτές τις αλλαγές γίνεται εισαγωγή του HSDPA.

Με τη συγκεκριμένη τεχνική αυξάνεται ο μέγιστος ρυθμός λήψης δεδομένων, από τα 2Mbps που όριζε το Release 99 φτάνουμε στο μέγιστο των 14.4Mbps του Release 5.

Κεφάλαιο 2 - Υπηρεσία σύντομων μηνυμάτων

2.1 Περιγραφή της υπηρεσίας σύντομων μηνυμάτων

Η Υπηρεσία Σύντομων Μηνυμάτων(Short Message Service - SMS) αναπτύχθηκε για το δίκτυο GSM, ενώ αργότερα χρησιμοποιήθηκε και στα δίκτυα CDMA και UMTS.

Αρχικά προοριζόταν για μονόδρομη επικοινωνία, μέσω σύντομων μηνυμάτων κειμένου που θα αποστέλλονται από το φορέα κινητής τηλεφωνίας προς τους συνδρομητές.

Με αυτό τον τρόπο ο φορέας θα ενημέρωνε τους συνδρομητές για κλήσεις που είχαν πραγματοποιηθεί και οι συνδρομητές για κάποιο λόγο δεν είχαν τη δυνατότητα να απαντήσουν.

Τελικώς η υπηρεσία τροποποιήθηκε για να δίνει στους συνδρομητές ενός δικτύου κινητής τηλεφωνίας τη δυνατότητα αμφίδρομης επικοινωνίας μέσα από την ανταλλαγή μηνυμάτων κειμένου.

Το πρώτο μήνυμα SMS στάλθηκε το 1992 κατά τη διάρκεια δοκιμών σε κάποιο Ευρωπαϊκό δίκτυο GSM. Τα επόμενα χρόνια η υπηρεσία γνώρισε τεράστια επιτυχία, αφού αναπτύχθηκαν πάρα πολλές υπηρεσίες που βασίζονταν στην ανταλλαγή μηνυμάτων SMS.

2.1.1 Εφαρμογές που βασίζονται στη χρήση SMS

Τα μηνύματα SMS, εκτός από το μέσο για την ανταλλαγή πληροφοριών ανάμεσα στους συνδρομητές κινητής τηλεφωνίας, αποτέλεσαν το θεμέλιο λίθο για την ανάπτυξη αρκετών εφαρμογών, όπως είναι οι εμπορικές εφαρμογές που προορίζονται για τους συνδρομητές, εφαρμογές που αναπτύχθηκαν για εταιρική χρήση και εφαρμογές που αναπτύχθηκαν και χρησιμοποιούνται από τους φορείς τηλεπικοινωνιών.

<u>Εμπορικές Εφαρμογές</u>

Σε αυτή την κατηγορία ανήκουν εφαρμογές όπως:

- η ανταλλαγή μηνυμάτων
- οι υπηρεσίες πληροφόρησης
- οι υπηρεσίες που επιτρέπουν στον συνδρομητή τη λήψη δεδομένων

Η ανταλλαγή μηνυμάτων είναι η συνηθέστερη περίπτωση χρήσης των SMS. Ο συνδρομητής που επιθυμεί να αποστείλει ένα SMS πρέπει πρώτα να συνθέσει το κείμενο του μηνύματος εισάγοντας χαρακτήρες μέσω του πληκτρολογίου της

φορητής συσκευής, το επόμενο βήμα είναι η εισαγωγή του αριθμού τηλεφώνου του συνδρομητή-παραλήπτη και η αποστολή προς αυτόν.

Το ολοκληρωμένο SMS, με κείμενο και στοιχεία αποστολέα και παραλήπτη, θα περάσει από ένα ή περισσότερα δίκτυα κινητής τηλεφωνίας και τελικά θα φτάσει στον παραλήπτη. Ο παραλήπτης θα ειδοποιηθεί από τη συσκευή του για το νέο μήνυμα που έλαβε και μετά θα είναι σε θέση να αναγνώσει το μήνυμα από την οθόνη της συσκευής του.

Η περίπτωση των υπηρεσιών πληροφόρησης είναι άλλο ένα συνηθισμένο σενάριο χρήσης των SMS. Εταιρείες παροχής τέτοιων υπηρεσιών ετοιμάζουν μηνύματα με περιεχόμενο όπως το δελτίο καιρού, η κίνηση σε διεθνείς χρηματαγορές, τα αποτελέσματα αθλητικών συναντήσεων και άλλες ειδήσεις. Οι πάροχοι προωθούν αυτά τα μηνύματα αυτόματα σε συνδρομητές με τους οποίους έχουν συνάψει συμφωνία ή αποστέλλουν τα μηνύματα μετά από αίτημα των συνδρομητών.

Συνήθως το αίτημα είναι ένα μήνυμα SMS που περιέχει κάποιες λέξεις-κλειδιά και αποστέλλεται σε έναν προκαθορισμένο αριθμό που ανήκει στον πάροχο ενημερώσεων.

Μια παρόμοια περίπτωση είναι οι ειδοποιήσεις που στέλνουν οι φορείς κινητής τηλεφωνίας για να ενημερώσουν για ένα νέο ηχητικό μήνυμα ή ένα νέο Email που έχει λάβει ο συνδρομητής.

Η λήψη μηνυμάτων που περιέχουν δυαδικά δεδομένα είναι άλλη μια πετυχημένη εφαρμογή των SMS, αφού βρίσκει τεράστια απήχηση σε συνδρομητές που επιθυμούν την παραμετροποίηση του κινητού τηλεφώνου τους.

Η παραμετροποίηση γίνεται με τη λήψη ενός ή παραπάνω SMS που περιέχουν νέες μελωδίες και προστίθενται στους ήχους κλήσης που υπήρχαν στη συσκευή. Ακόμη οι συνδρομητές μπορούν να λάβουν κινούμενες εικόνες ή να αλλάξουν το γραφικό περιβάλλον που προσφέρεται από τον κατασκευαστή του τηλεφώνου.

Εταιρικές Εφαρμογές

Στην κατηγορία αυτή ανήκουν εφαρμογές που έχουν υλοποιηθεί με στόχο την ικανοποίηση των αναγκών μιας εταιρείας. Ένα χαρακτηριστικό παράδειγμα μιας εφαρμογής αυτού του τύπου είναι η εποπτεία λειτουργίας ενός απομακρυσμένου εξυπηρετητή, σε περίπτωση που εμφανιστεί κάποιο πρόβλημα τότε οι διαχειριστές του συστήματος ειδοποιούνται με ένα SMS.

Εφαρμογές που σχετίζονται με τους φορείς κινητής τηλεφωνίας

Οι φορείς κινητής τηλεφωνίας χρησιμοποίησαν το SMS σαν κύριο συστατικό για τη δημιουργία εφαρμογών που σχετίζονται με τη λειτουργία του δικτύου και των συσκευών που συνδέονται σε αυτό.

Ένα παράδειγμα είναι η απομακρυσμένη ενημέρωση καρτών SIM, με την αποστολή ενός ή παραπάνω SMS γίνεται τροποποίηση παραμέτρων, όπως είναι ο αριθμός για το κέντρο μηνυμάτων, που είναι αποθηκευμένες στην κάρτα SIM.

2.1.2 Μετάδοση σύντομων μηνυμάτων

Η μετάδοση σύντομων μηνυμάτων διεξάγεται με τη χρήση του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων(Short Message Transfer Protocol – SM-TP), το οποίο βασίζεται στην ιδέα Αποθηκεύω και Προωθώ(Store and Forward).

Τα σύντομα μηνύματα δημιουργούνται από τις Οντότητες Σύντομων Μηνυμάτων, προωθούνται σε άλλα στοιχεία του δικτύου και καταλήγουν για αποθήκευση στο Κέντρο SMS.

Η Οντότητα Σύντομων Μηνυμάτων(Short Message Entity - SME) είναι ένα στοιχείο του δικτύου που έχει τη δυνατότητα αποστολής και λήψης SMS. Σαν SME λογίζεται μια εφαρμογή λογισμικού που εκτελείται σε μια συσκευή κινητού τηλεφώνου ή σε έναν απομακρυσμένο εξυπηρετητή.

Ένα μήνυμα που δημιουργείται σε μία οντότητα σύντομων μηνυμάτων και καταλήγει για αποθήκευση στο δίκτυο λέγεται Mobile Originated - Short Message ή συντομότερα MO-SM.

Το μήνυμα που ακολουθεί την αντίθετη πορεία, δηλαδή ξεκινά από το δίκτυο και τερματίζει σε μια οντότητα σύντομων μηνυμάτων, λέγεται Mobile Terminated – Short Message ή πιο σύντομα MT-SM.

Το Κέντρο SMS(SMS Centre - SMSC) ή Κέντρο Υπηρεσίας(Service Centre - SC) υλοποιεί τη βασική ιδέα της υπηρεσίας, καθώς αποθηκεύει όλα τα MO-SM μηνύματα που λαμβάνει και αργότερα τα προωθεί, με τη μορφή MT-SM, προς την οντότητα του αντίστοιχου παραλήπτη.

Το SMSC μπορεί να είναι μια αυτόνομη μονάδα ή να βρίσκεται ενσωματωμένο σε ένα MSC. Η επικοινωνία ανάμεσα στο SMSC και το MSC πραγματοποιείται χάρις σε δύο λειτουργίες που έχουν μορφή λογισμικού και βρίσκονται ενσωματωμένες στο SMSC.

Οι εν λόγω λειτουργίες είναι:

- Η Πύλη-Κέντρο Μεταγωγής Κινητών Υπηρεσιών για την Υπηρεσία Σύντομων Μηνυμάτων(Short Message Service - Gateway Mobile services Switching Centre - SMS-GMSC)

- Το Κέντρο Διασύνδεσης-Μεταγωγής Κινητών Υπηρεσιών για την Υπηρεσία Σύντομων Μηνυμάτων(Short Message Service – InterWorking Mobile services Switching Centre – SMS - IWMSC)

Στόχος του SMS-GMSC είναι η λήψη MT-SM μηνυμάτων από το SMSC, η αναζήτηση της θέσης του παραλήπτη και η προώθηση του μηνύματος στο κατάλληλο MSC που εξυπηρετεί τον παραλήπτη.

Το SMS-IWMSC δέχεται ένα MO-SM μήνυμα από το MSC που εξυπηρετεί την πηγή του μηνύματος και το προωθεί για αποθήκευση στο SMSC.

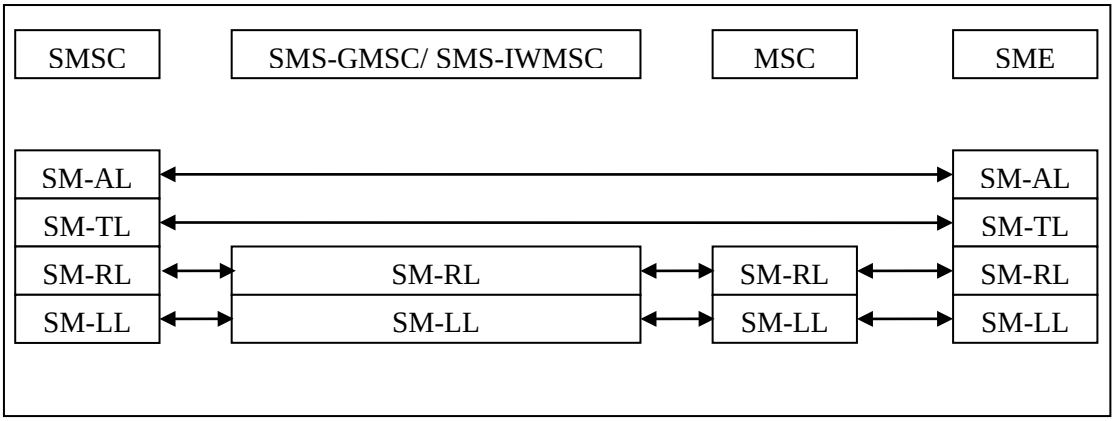
Το πρωτόκολλο μεταφοράς σύντομων μηνυμάτων αποτελείται από μια στοίβα τεσσάρων επιπέδων ή στρωμάτων.

Στο υψηλότερο επίπεδο συναντάμε το στρώμα εφαρμογής για σύντομα μηνύματα(Short Message-Application Layer – SM-AL).

Ακολουθούν το στρώμα μεταφοράς σύντομων μηνυμάτων(Short Message-Transfer Layer – SM-TL) και το στρώμα αναμετάδοσης σύντομων μηνυμάτων(Short Message-Relay Layer – SM-RL).

Στο τελευταίο επίπεδο έχουμε το στρώμα ζεύξης σύντομων μηνυμάτων(Short Message-Link Layer – SM-LL).

Στο σχήμα που ακολουθεί παρουσιάζονται η στοίβα του πρωτοκόλλου μεταφοράς και οι σημαντικότερες δικτυακές οντότητες που εμπλέκονται κατά τη μετάδοση μηνυμάτων.



Εικόνα 16: Το πρωτόκολλο μεταφοράς σύντομων μηνυμάτων

Το στρώμα εφαρμογής παρέχεται στις οντότητες σύντομων μηνυμάτων με τη μορφή λογισμικού. Σε αυτό το στρώμα ένα σύντομο μήνυμα παρουσιάζεται με μορφή κειμένου(text-mode) που είναι εύκολα αντιληπτή από τον άνθρωπο. Το στρώμα εφαρμογής παρέχει την κατάλληλη διεπαφή με την οποία γίνονται ενέργειες που αφορούν τη διαχείριση μηνυμάτων.

Στο στρώμα μεταφοράς το μήνυμα θεωρείται σαν μια αλληλουχία από οκτάδες δυαδικών ψηφίων(octets ή bytes), το σύντομο μήνυμα που έχει αυτή τη μορφή

ονομάζεται μονάδα δεδομένων του πρωτοκόλλου μεταφοράς(Transfer Protocol Data Unit - TPDU).

Ένα TPDU περιέχει, εκτός από το κείμενο του μηνύματος, μια σειρά από επιπρόσθετες πληροφορίες, όπως είναι:

- το μήκος του μηνύματος
- ο αριθμός του αποστολέα
- ο αριθμός του παραλήπτη
- επιπρόσθετα στοιχεία που βοηθούν στη δρομολόγηση του μηνύματος

Το στρώμα αναμετάδοσης συνεργάζεται με το στρώμα μεταφοράς και αναλαμβάνει την αναμετάδοση των TPDU ανάμεσα στα στοιχεία του δικτύου που εμπλέκονται κατά την προώθηση σύντομων μηνυμάτων.

Το στρώμα ζεύξης επιτρέπει τη μετάδοση του μηνύματος σε φυσικό επίπεδο, ενώ παράλληλα είναι υπεύθυνο για την ποιότητα των συνδέσεων και έχει σαν στόχο την προφύλαξη του μηνύματος από πιθανά λάθη που γίνονται κατά τη μετάδοση.

Ας δούμε όμως αναλυτικότερα πως διεξάγεται η μετάδοση των MO-SM και MT-SM μηνυμάτων.

Η μετάδοση ενός MO-SM

Όπως αναφέραμε και νωρίτερα, ένα MO-SM μήνυμα πηγάζει από μια SME και καταλήγει στο SMSC. Στην κατηγορία των MO-SM μηνυμάτων περιλαμβάνονται:

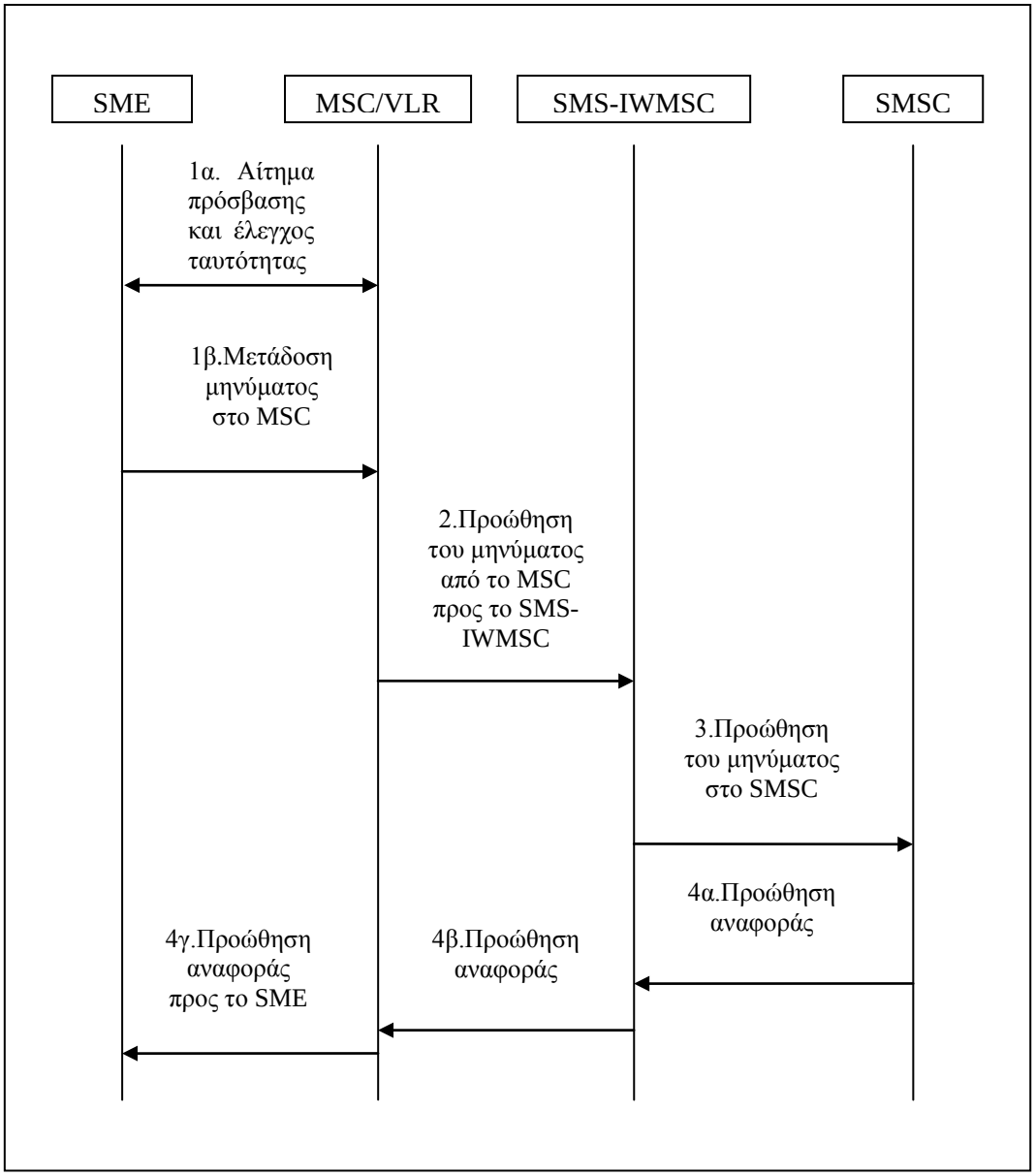
- τα μηνύματα SMS-SUBMIT
- τα μηνύματα SMS-DELIVER-REPORT
- τα μηνύματα SMS-COMMAND

Το SMS-SUBMIT είναι το τυπικό μήνυμα κειμένου που αποστέλλεται από μια οντότητα σύντομων μηνυμάτων και καταλήγει στο SMSC.

Το μήνυμα SMS-DELIVER-REPORT αποστέλλεται από την οντότητα σύντομων μηνυμάτων προς το SMSC για να επιβεβαιώσει την επιτυχή λήψη ενός μηνύματος που ήρθε από το SMSC.

Το μήνυμα SMS-COMMAND αποστέλλεται από την οντότητα σύντομων μηνυμάτων και περιέχει ένα αίτημα για την εκτέλεση κάποιας ειδικής εντολής από το SMSC.

Στο σχήμα που ακολουθεί παρουσιάζεται, βήμα προς βήμα, το γενικό σενάριο που ακολουθείται κατά τη μετάδοση ενός μηνύματος MO-SM.



Εικόνα 17: Η πορεία από σημείο προς σημείο που ακολουθείται ένα μήνυμα MO-SM

1. Η οντότητα σύντομων μηνυμάτων αποστέλλει, μέσω του υποσυστήματος σταθμού βάσης, το μήνυμα στο MSC. Παράλληλα επικοινωνεί με το VLR και αποστέλλει αίτημα για τη μετάδοση του σύντομου μηνύματος. Το VLR ελέγχει την ταυτότητα του συνδρομητή και εξετάζει αν η μετάδοση του μηνύματος υπόκειται σε κάποιους περιορισμούς του δικτύου ή είναι αντίθετη με τις υπηρεσίες που λαμβάνει ο συνδρομητής.

2. Το VLR ενημερώνει το MSC για το αποτέλεσμα του ελέγχου, αν δεν υπάρχει κάποιο πρόβλημα τότε το μήνυμα προωθείται από το MSC προς το SMS-IWMSC.

3. Το SMS-IWMSC λαμβάνει το μήνυμα από το MSC και το προωθεί στο SMSC με το οποίο συνδέεται. Από εκεί και πέρα το SMSC αποκτά τον πλήρη έλεγχο του μηνύματος.

4. Το SMSC δημιουργεί μιαν αναφορά με την οποία γνωστοποιεί στις εμπλεκόμενες οντότητες το αποτέλεσμα της διαδικασίας. Τελικός αποδέκτης της αναφοράς είναι η οντότητα που έστειλε το σύντομο μήνυμα.

Το SMSC αποθηκεύει το μήνυμα σε μια ουρά μηνυμάτων μέχρι να έρθει η στιγμή που θα το επεξεργαστεί. Την κατάλληλη στιγμή το SMSC θα εξετάσει την εγκυρότητα του μηνύματος, αν το μήνυμα δεν είναι έγκυρο τότε αυτό διαγράφεται από την ουρά μηνυμάτων του SMSC.

Σε περίπτωση που το μήνυμα είναι έγκυρο, τότε το SMSC ή θα μετατρέψει το μήνυμα σε MT-SM και θα ξεκινήσει τη διαδικασία της παράδοσης του ή θα το προωθήσει σε άλλο SMSC που εξυπηρετεί τον παραλήπτη.

Η μετάδοση ενός MT-SM

Ένα μήνυμα MT-SM ξεκινά από το κέντρο SMSC και έχει σαν τελικό αποδέκτη μιαν SME. Στην κατηγορία των MT-SM μηνυμάτων περιλαμβάνονται:

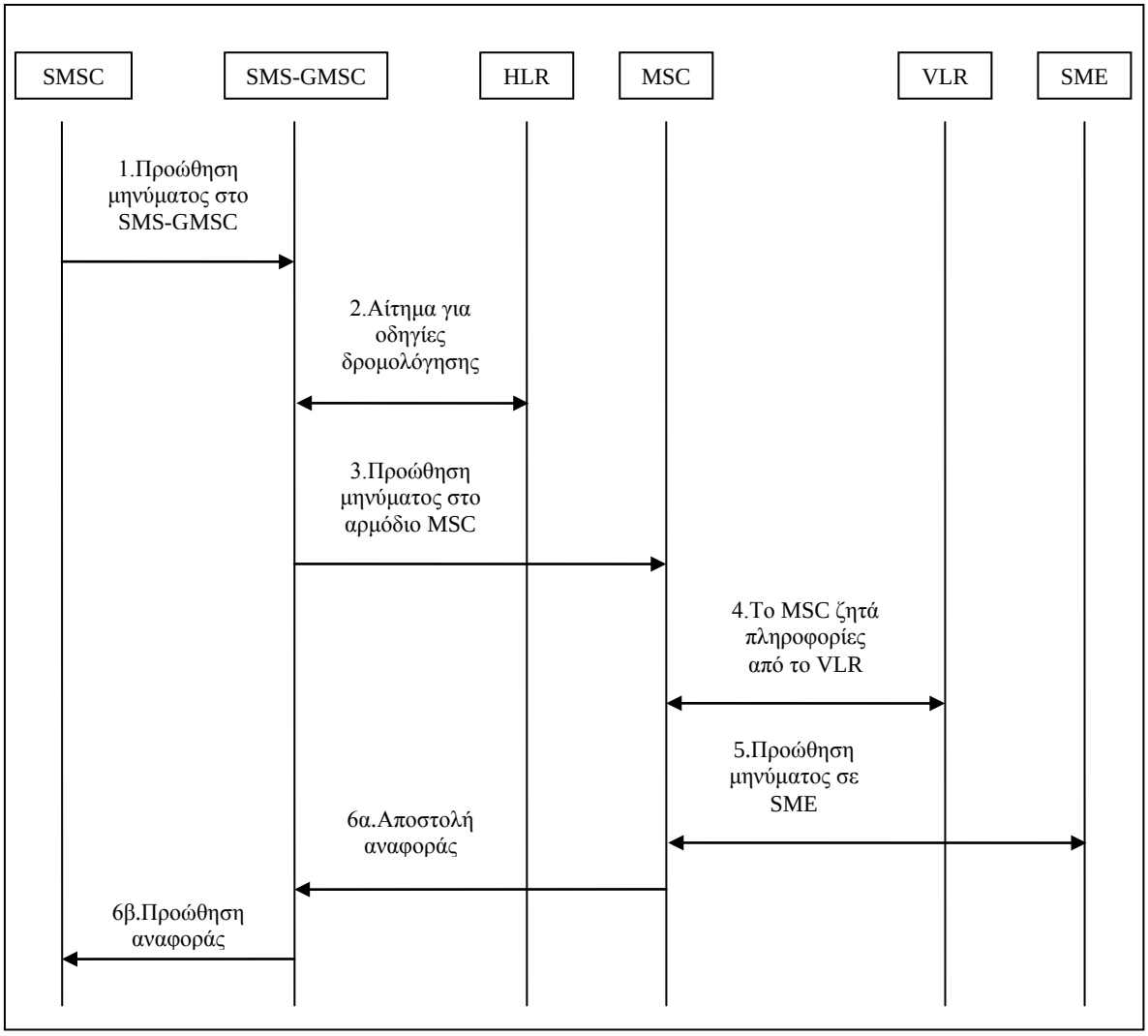
- τα μηνύματα SMS-DELIVER
- τα μηνύματα SMS-SUBMIT-REPORT
- τα μηνύματα SMS-STATUS-REPORT

Το SMS-DELIVER είναι το τυπικό μήνυμα κειμένου που φεύγει από το κέντρο υπηρεσίας σύντομων μηνυμάτων και καταλήγει σε μια οντότητα σύντομων μηνυμάτων.

Το μήνυμα SMS-SUBMIT-REPORT πάει από το SMSC σε μια οντότητα σύντομων μηνυμάτων για να επιβεβαιώσει την επιτυχή λήψη ενός μηνύματος SMS-SUBMIT.

Το SMS-STATUS-REPORT αποστέλλεται από το κέντρο σύντομων μηνυμάτων για να ενημερώσει τον αποστολέα ενός μηνύματος για την κατάληξη που είχε το μήνυμα το οποίο έστειλε.

Στο σχήμα που ακολουθεί εμφανίζεται η πορεία που ακολουθεί ένα μήνυμα MT-SM.



Εικόνα 18: Η πορεία σημείο προς σημείο που ακολουθείται κατά τη μετάδοση ενός MT-SM μηνύματος

1. Το SMSC έχει μια ουρά με αποθηκευμένα MO-SM μηνύματα που έχει λάβει από εξωτερικές οντότητες σύντομων μηνυμάτων. Ακολουθώντας τη σειρά των μηνυμάτων, το SMSC εξετάζει το πρώτο από αυτά που είναι έτοιμο για μετάδοση και ελέγχει αν είναι έγκυρο.

Αν το μήνυμα δεν είναι έγκυρο, τότε το SMSC θα το διαγράψει από την ουρά και θα προχωρήσει στον έλεγχο του επόμενου μηνύματος.

Σε αντίθετη περίπτωση, αν το μήνυμα είναι έγκυρο, τότε το SMSC προωθεί το μήνυμα στο SMS-GMSC.

2. Το SMS-GMSC ζητά οδηγίες δρομολόγησης από το HLR, εκείνο με τη σειρά του επιστρέφει τη διεύθυνση του MSC στο οποίο ανήκει ο συνδρομητής.

3. Το SMS-GMSC προωθεί το μήνυμα στο MSC που είναι υπεύθυνο για την οντότητα σύντομων μηνυμάτων του παραλήπτη.

4. Το MSC επικοινωνεί με το VLR και ζητά πληροφορίες για τη δρομολόγηση του μηνύματος. Από την πλευρά του, το VLR επιστρέφει στο MSC πληροφορίες που σχετίζονται με τη θέση και την κατάσταση της οντότητας του παραλήπτη.

5. Αν οι πληροφορίες του VLR επιβεβαιώνουν ότι ο παραλήπτης έχει τη δυνατότητα να λάβει το μήνυμα, τότε το μήνυμα προωθείται από το MSC, με τη βοήθεια του υποσυστήματος σταθμού βάσης, προς την οντότητα σύντομων μηνυμάτων του παραλήπτη.

Στην περίπτωση της επιτυχημένης λήψης του μηνύματος από την οντότητα μηνυμάτων ακολουθεί η αποστολή επιβεβαίωσης προς το MSC.

6. Το MSC δημιουργεί μια αναφορά με το αποτέλεσμα της διαδικασίας μετάδοσης, η οποία προωθείται αρχικά στο SMS-GMSC και από εκεί φτάνει στο SMSC.

Το περιεχόμενο της αναφοράς επιβεβαιώνει την επιτυχία ή την αποτυχία της μετάδοσης ενός σύντομου μηνύματος.

Αν για κάποιο λόγο έχουμε αποτυχημένη μετάδοση, είτε γιατί ο παραλήπτης έχει τη συσκευή του απενεργοποιημένη είτε γιατί βρίσκεται σε περιοχή εκτός κάλυψης, τότε το SMSC θα διατηρήσει αποθηκευμένο το μήνυμα και περιοδικά θα ξαναδοκιμάσει να το στείλει.

Αν η μετάδοση είναι επιτυχημένη, τότε το SMSC διαγράφει το αποθηκευμένο μήνυμα, ενώ παράλληλα ξεκινά τη διαδικασία δρομολόγησης ενός μηνύματος που προορίζεται για την οντότητα του αποστολέα και επιβεβαιώνει ότι το μήνυμα έφτασε στον παραλήπτη του.

2.1.3 Τεχνικά χαρακτηριστικά ενός Σύντομου Μηνύματος

Η αποστολή και η λήψη Σύντομων Μηνυμάτων βασίζεται στην ανταλλαγή πλαισίων TPDU που περιέχουν ένα σύνολο από επικεφαλίδες και ένα τμήμα που περιέχει τα δεδομένα του χρήστη.

Οι επικεφαλίδες σχετίζονται με οδηγίες δρομολόγησης και τον τύπο του μηνύματος.

Το κομμάτι με τα δεδομένα του χρήστη μπορεί να δεχθεί μέχρι 1120 bits ή 140 bytes δεδομένων, τα οποία μπορεί να είναι αλφαριθμητικοί χαρακτήρες που σχηματίζουν κείμενο ή να είναι άλλου είδους πληροφορίες, όπως είναι μια εικόνα.

Στην περίπτωση που τα μεταφερόμενα δεδομένα είναι κείμενο τότε χρησιμοποιούνται συγκεκριμένα σχήματα κωδικοποίησης(Encoding Schemes), όπως είναι το GSM-7 και το UCS-2.

Το σχήμα GSM-7 χρησιμοποιεί 7 bit για κάθε χαρακτήρα και περιλαμβάνει χαρακτήρες και σύμβολα που υπάρχουν στο αλφάβητο των περισσοτέρων ευρωπαϊκών χωρών.

Η κωδικοποίηση UCS-2 χρησιμοποιεί 16 bit για κάθε χαρακτήρα και υποστηρίζει το Κυριλλικό αλφάβητο, το Αραβικό αλφάβητο και τους χαρακτήρες που χρησιμοποιούνται στα αλφάβητα των Ασιατικών γλωσσών.

Ανάλογα με την κωδικοποίηση που χρησιμοποιείται αλλάζει και το πλήθος των χαρακτήρων που μπορούν να χωρέσουν σε ένα μήνυμα, έτσι για τα δύο προαναφερθέντα encoding schemes ισχύει:

- για το GSM-7: μέγιστη χωρητικότητα 1120bit / 7bit = 160 χαρακτήρες των 7bit ανά μήνυμα
- για το UCS-2: μέγιστη χωρητικότητα 1120bit / 16bit = 70 χαρακτήρες των 16bit ανά μήνυμα

Ας δούμε όμως αναλυτικότερα την δομή που έχει το πλαίσιο TPDU ενός μηνύματος MO-SM και το πλαίσιο TPDU ενός μηνύματος MT-SM.

Το πλαίσιο TPDU ενός MO-SM

Ένα μήνυμα τύπου MO-SM(Mobile Originated – Short Message) πηγάζει από μιαν Οντότητα Σύντομων Μηνυμάτων και καταλήγει στο Κέντρο Υπηρεσίας των Σύντομων Μηνυμάτων.

Στο πλαίσιο του μηνύματος περιέχονται τα δεδομένα του χρήστη και επιπρόσθετες οδηγίες για τη δρομολόγηση και την επεξεργασία του μηνύματος από το Κέντρο Υπηρεσίας.

Στο σχήμα που ακολουθεί παρουσιάζεται η δομή που έχει το πλαίσιο ενός μηνύματος τύπου MO-SM, το παράδειγμα μας είναι το SMS-SUBMIT.

	Bit Index							
No. of octets	7	6	5	4	3	2	1	0
1	TP-RP	TP-UDHI	TP-SRR	TP-VPF		TP-RD	TP-MTI	
1	TP-Message Reference							
2…12	TP-Destination Address							
1	TP-Protocol Identifier							
1	TP-Data Coding Scheme							
0,1 or 7	TP-Validity Period							
1	TP-User Data Length							
0…140	TP-User Data							

Εικόνα 19: Η δομή του πλαισίου ενός MO-SM μηνύματος

Ας δούμε πιο αναλυτικά τι περιλαμβάνεται στο πλαίσιο ενός μηνύματος MO-SM, ξεκινώντας από την πρώτη οκτάδα δυαδικών ψηφίων όπου έχουμε τα ακόλουθα στοιχεία:

- TP-MTI – Transfer Protocol - Message Type Indicator
- TP-RD – Transfer Protocol - Reject Duplicates
- TP-VPF – Transfer Protocol - Validity Period Format
- TP-SRR – Transfer Protocol - Status Report Request
- TP-UDHI – Transfer Protocol - User Data Header Indicator
- TP-RP – Transfer Protocol - Reply Path

Τα bits 0 και 1 περιέχουν την τιμή του TP-MTI(Transfer Protocol - Message Type Indicator) το οποίο αποτελεί ενδεικτικό του τύπου του μηνύματος. Στην περίπτωση μας, με το μήνυμα να είναι τύπου MO-SM το bit στη θέση 0 έχει τιμή 1 και το γειτονικό bit στη θέση 1 έχει τιμή 0.

Το bit 2 που αντιπροσωπεύει την τιμή TP-RD(Transfer Protocol - Reject Duplicates) υποδεικνύει στο Κέντρο Υπηρεσίας για SMS αν θα δέχεται ή θα απορρίπτει τα αντίγραφα που ενδέχεται να υπάρχουν κατά τη μετάδοση ενός μηνύματος.

Η τιμή του συγκεκριμένου δυαδικού ψηφίου καθορίζεται από τον πάροχο της υπηρεσίας σύμφωνα με τις ρυθμίσεις που ισχύουν στο δίκτυο του.

Στο bit 3 και στο bit 4 αποθηκεύεται η τιμή του TP-VPF(Transfer Protocol - Value Period Format) η οποία ξεκαθαρίζει αν υπάρχει κάποια περίοδος εγκυρότητας για το μήνυμα και σε περίπτωση που υπάρχει με ποιο τρόπο αυτή αναπαρίσταται.

Υπάρχουν τέσσερις συνολικά τρόποι, όμως αυτοί που χρησιμοποιούνται πιο συχνά είναι οι ακόλουθοι δύο:

- με απόλυτη τιμή, όπου καθορίζεται η ημερομηνία που θα τερματιστεί η περίοδος εγκυρότητας
- με την τιμή να σχετίζεται με τη χρονική στιγμή που το Κέντρο Υπηρεσίας για SMS έλαβε το μήνυμα

Στο bit 5 έχουμε την τιμή του TP-SRR(Transfer Protocol - Status Report Request) που καθορίζει αν ο αποστολέας του μηνύματος επιθυμεί να λάβει επιβεβαίωση για την επιτυχημένη παράδοση του μηνύματος που έστειλε.

Το bit στη θέση 6 φιλοξενεί την τιμή του TP-UDHI(Transfer Protocol - User Data Header Indicator) και χρησιμοποιείται σαν ενδεικτικό για την ύπαρξη ή την απουσία κάποιας επικεφαλίδας μέσα στο τμήμα με τα δεδομένα του χρήστη.

Τέλος, στο bit 7 υπάρχει η τιμή του TP-RP(Transfer Protocol - Reply Path) και υποδεικνύει αν ο αποστολέας του μηνύματος επιθυμεί να χρεωθεί και το μήνυμα απάντησης.

Η αμέσως επόμενη οκτάδα δυαδικών ψηφίων περιλαμβάνει την τιμή ενός ακεραίου αριθμού αναφοράς μηνύματος(Transfer Protocol - Message Reference), ο οποίος χρησιμοποιείται σαν ταυτότητα στα μηνύματα που έχουν σταλεί από την Οντότητα Σύντομων Μηνυμάτων στο Κέντρο Υπηρεσίας για SMS.

Με τον αριθμό αναφοράς γίνεται ευκολότερος ο έλεγχος για αντίγραφα μηνυμάτων που έχουν φτάσει στο Κέντρο Υπηρεσίας για SMS, το οποίο ανάλογα με τις ρυθμίσεις επιτρέπει ή απορρίπτει τα πολλαπλά αντίγραφα μηνυμάτων.

Ακόμη, το πεδίο με τον αριθμό αναφοράς υπάρχει και στα μηνύματα επιβεβαίωσης, έτσι όταν απαιτείται ένα μήνυμα επιβεβαίωσης για SMS από το Κέντρο Υπηρεσίας, τότε το κέντρο εισάγει στο συγκεκριμένο πεδίο τον ίδιο αριθμό αναφοράς που είχε το μήνυμα που έλαβε από τον αποστολέα, με αυτό τον τρόπο ξεκαθαρίζει στον αποστολέα ότι το μήνυμα με τον συγκεκριμένο αριθμό αναφοράς έφτασε επιτυχώς στον παραλήπτη.

Στο πεδίο που ακολουθεί αποθηκεύεται η διεύθυνση για την οποία προορίζεται το μήνυμα(Transfer Protocol - Destination Address). Η διεύθυνση γράφεται σύμφωνα με τη διεθνή μορφή που ορίζεται από την ITU, δηλαδή έχει την ακόλουθη μορφή:

“+<Τηλεφωνικός_Κωδικός_Χωρας><Αριθμός_Τηλεφώνου>”.

Για κάθε ψηφίο της διεύθυνσης χρησιμοποιούνται 4 bits και το μέγιστο μήκος που μπορεί να καταλάβει η διεύθυνση είναι μέχρι και 12 οκτάδες δυαδικών ψηφίων.

Μετά το πεδίο της διευθύνσεως προορισμού βρίσκεται το πεδίο που περιέχει το αναγνωριστικό του πρωτοκόλλου(Transfer Protocol - Protocol Identifier) που χρησιμοποιείται για τη μετάδοση των μηνυμάτων. Το πεδίο αυτό έχει μήκος 8 δυαδικών ψηφίων.

Το επόμενο πεδίο έχει μήκος μιας οκτάδας δυαδικών ψηφίων και περιέχει πληροφορίες για το σχήμα κωδικοποίησης που πιθανότατα να εφαρμόζεται στο τμήμα με τα δεδομένα του χρήστη(Transfer Protocol - Data Coding Scheme).

Όπως αναφέραμε και παραπάνω τα πιο συνηθισμένα σχήματα κωδικοποίησης είναι το GSM-7 και το UCS-2, ενώ υπάρχει και η περίπτωση που δεν χρησιμοποιείται κάποιο σχήμα κωδικοποίησης και το τμήμα με τα δεδομένα του χρήστη μεταφέρει δεδομένα σε δυαδική μορφή.

Το πεδίο που ακολουθεί δέχεται την τιμή της περιόδου εγκυρότητας του μηνύματος(Transfer Protocol - Validity Period). Ο τρόπος με τον οποίο θα απεικονίζεται η περίοδος εγκυρότητας εξαρτάται από την τιμή που έχει το VPF στην πρώτη οκτάδα δυαδικών ψηφίων.

Ανάλογα με τον τρόπο απεικόνισης έχουμε και διαφορά στο πλήθος των οκτάδων που καταλαμβάνονται από την περίοδο εγκυρότητας, έτσι στην περίπτωση που χρησιμοποιείται ο σχετικός τρόπος απεικόνισης καταλαμβάνεται μια οκτάδα από δυαδικά ψηφία, ενώ στην περίπτωση που χρησιμοποιείται η απόλυτη τιμή καταλαμβάνονται εφτά οκτάδες δυαδικών ψηφίων.

Το προτελευταίο πεδίο έχει μήκος μιας οκτάδας δυαδικών ψηφίων και περιέχει έναν ακέραιο αριθμό που σε συνάρτηση με το σχήμα κωδικοποίησης που χρησιμοποιείται συμβολίζει το μήκος που θα έχει το τμήμα με τα δεδομένα του χρήστη(Transfer Protocol - User Data Length).

Στην περίπτωση που χρησιμοποιείται το σχήμα GSM-7 τότε κάθε χαρακτήρας του μηνύματος καταλαμβάνει 7 bit, έτσι το μήκος δεδομένων εκφράζει το σύνολο των χαρακτήρων ή των επτάδων δυαδικών ψηφίων που ακολουθούν στο τμήμα με τα δεδομένα του χρήστη.

Στην περίπτωση που χρησιμοποιείται το σχήμα UCS-2 με 16 bit ανά χαρακτήρα τότε το μήκος αντιπροσωπεύει τον αριθμό των οκτάδων από δυαδικά ψηφία που βρίσκονται στα δεδομένα του χρήστη.

Τέλος, στην περίπτωση που το μήνυμα έχει δυαδικό φορτίο και δεν χρησιμοποιείται κάποιο σχήμα κωδικοποίησης τότε ο αριθμός του μήκους αντιπροσωπεύει τις οκτάδες από bit που βρίσκονται στο τμήμα με τα δεδομένα του χρήστη.

Στο τελευταίο πεδίο του πλαισίου έχουμε το τμήμα με τα δεδομένα του χρήστη(Transfer Protocol - User Data). Το πεδίο αυτό έχει μέγιστη χωρητικότητα που φτάνει τις 140 οκτάδες δυαδικών ψηφίων.

Όμως, όπως αναφέραμε και νωρίτερα, ανάλογα με την κωδικοποίηση που πιθανότατα να εφαρμόζεται ο αριθμός των 140 οκτάδων bit μεταφράζεται σε 160 χαρακτήρες ανά μήνυμα αν το σχήμα κωδικοποίησης είναι το GSM-7, ενώ στην περίπτωση που εφαρμόζεται το σχήμα κωδικοποίησης UCS-2 ο μέγιστος αριθμός χαρακτήρων ανά μήνυμα είναι 70.

Το πλαίσιο TPDU ενός MT-SM

Ένα μήνυμα τύπου MT-SM(Mobile Terminated Short Message) ξεκινά από το Κέντρο Υπηρεσίας για Σύντομα Μηνύματα και καταλήγει σε μιαν Οντότητα Σύντομων Μηνυμάτων.

Στο σχήμα που ακολουθεί παρουσιάζεται η δομή που έχει το πλαίσιο ενός μηνύματος MT-SM. Όπως φαίνεται και από το σχήμα το πλαίσιο ενός μηνύματος MT-SM δεν έχει αρκετές διαφορές σε σχέση με το πλαίσιο ενός MO-SM μηνύματος.

	Bit Index							
No. of octets	7	6	5	4	3	2	1	0
1	TP-RP	TP-UDHI	TP-SRI	Unused		TP-MMS	TP-MTI	
2…12	TP-Originator Address							
1	TP-Protocol Identifier							
1	TP-Data Coding Scheme							
1…7	TP-Service Centre Time Stamp							
1	TP-User Data Length							
0…140	TP-User Data							

Εικόνα 20: Η δομή του πλαισίου ενός MT-SM μηνύματος

Στο πλαίσιο του μηνύματος περιέχονται τα δεδομένα του χρήστη μαζί με οδηγίες για την παράδοση του μηνύματος από το Κέντρο Υπηρεσίας για SMS προς την Οντότητα Σύντομων Μηνυμάτων του παραλήπτη.

Ξεκινώντας από την πρώτη οκτάδα δυαδικών ψηφίων η οποία χωρίζεται στα ακόλουθα πεδία:

- TP-MTI – Transfer Protocol - Message Type Indicator
- TP-MMS – Transfer Protocol - More Messages to Send
- TP-SRI – Transfer Protocol - Status Report Indicator
- TP-UDHI – Transfer Protocol - User Data Header Indicator
- TP-RP – Transfer Protocol - Reply Path

Τα bit 0 και 1 της πρώτης οκτάδας περιέχουν την τιμή του TP-MTI(Transfer Protocol - Message Type Indicator), το οποίο λειτουργεί με τον ίδιο τρόπο που λειτουργεί το TP-MTI στο πλαίσιο ενός MO-SM μηνύματος, δηλαδή χρησιμοποιείται σαν αναγνωριστικό του τύπου του μηνύματος.

Στην περίπτωση του μηνύματος MT-SM και τα δύο δυαδικά ψηφία έχουν τιμή 0.

Το bit που ακολουθεί στη θέση 2 έχει την τιμή για το TP-MMS(Transfer Protocol - More Messages to Send) και ανάλογα με την τιμή αυτή υποδηλώνει την ύπαρξη άλλων μηνυμάτων που βρίσκονται στο Κέντρο Υπηρεσίας για Σύντομα Μηνύματα και περιμένουν για να προωθηθούν στον παραλήπτη.

Το στοιχείο αυτό είναι αρκετά χρήσιμο, αφού σε περίπτωση που υπάρχουν πολλά μηνύματα προς τον ίδιο παραλήπτη τότε αυτά μεταδίδονται αμέσως διατηρώντας το ίδιο κανάλι μετάδοσης, έτσι μειώνεται ο παραπανίσιος φόρτος στο δίκτυο και γίνεται καλύτερη χρήση των διαθέσιμων πόρων.

Τα bit στις θέσεις 3 και 4 δεν χρησιμοποιούνται και η τιμή που υπάρχει και στις δύο θέσεις εκ των προτέρων είναι 0.

Στο bit της θέσης 5 βρίσκεται το TP-SRI(Transfer Protocol - Status Report Indicator). Σε περίπτωση που η Οντότητα Σύντομων Μηνυμάτων του παραλήπτη λάβει επιτυχώς ένα μήνυμα τότε ανάλογα με την τιμή του TP-SRI θα γνωρίζει αν θα πρέπει να αποστείλει μήνυμα αναφοράς προς τον αποστολέα που θα επιβεβαιώνει την επιτυχημένη λήψη του μηνύματος.

Το γειτονικό δυαδικό ψηφίο που βρίσκεται στη θέση 6 περιέχει την τιμή του TP-UDHI(Transfer Protocol - User Data Header Indicator).

Το TP-UDHI όπως και στην περίπτωση του MO-SM πλαισίου ενημερώνει την Οντότητα Σύντομων Μηνυμάτων για την ύπαρξη κάποιας επιπρόσθετης επικεφαλίδας η οποία βρίσκεται εντός των δεδομένων του χρήστη.

Στο έβδομο και τελευταίο δυαδικό ψηφίο της πρώτης οκτάδας έχουμε το TP-RP(Transfer Protocol - Reply Path), για το οποίο ισχύει ότι και στην αντίστοιχη περίπτωση του MO-SM πλαισίου, το TP-RP ενημερώνει την Οντότητα που λαμβάνει ότι ο αποστολέας επιθυμεί να χρεωθεί και το μήνυμα απάντησης.

Το επόμενο πεδίο έχει μέγιστο μήκος 12 οκτάδες δυαδικών ψηφίων και έχει αποθηκευμένη την διεύθυνση του αποστολέα(Transfer Protocol - Originator Address).

Τα δύο πεδία που ακολουθούν, με το αναγνωριστικό πρωτοκόλλου(Transfer Protocol - Protocol Identifier) και το σχήμα κωδικοποίησης(Transfer Protocol - Data Coding Scheme), έχουν το ίδιο μήκος και την ίδια λειτουργία με αυτή που είχαν και στο πλαίσιο που αναλύθηκε νωρίτερα και αφορούσε ένα MO-SM μήνυμα.

Το πεδίο που ακολουθεί έχει αποθηκευμένη τη χρονική στιγμή που το Κέντρο Υπηρεσίας έλαβε το μήνυμα από τον αποστολέα(Transfer Protocol - Service Centre Time Stamp). Το πεδίο με το Time Stamp έχει μήκος 7 οκτάδες δυαδικών ψηφίων στις οποίες ο χρόνος αναπαρίσταται με τον εξής τρόπο:

Έτος:Μήνας:Ημέρα:Ώρα:Λεπτό:Δευτερόλεπτο:Χρονική Ζώνη

Κάθε στοιχείο αποτελείται από δύο ψηφία. Ως Χρονική Ζώνη δίνεται η θετική ή αρνητική διαφορά που είναι πιθανόν να υπάρχει από τη ζώνη GMT.

Για τα δύο τελευταία πεδία, με το μήκος των δεδομένων του χρήστη(TP-User Data Length) και τα δεδομένα του χρήστη(TP-User Data), ισχύει ότι και στην περίπτωση του πλαισίου του MO-SM μηνύματος.

2.2 Οι εντολές AT και ο τρόπος αποστολής SMS μέσω PC

Ο συνηθέστερος τρόπος με τον οποίο γίνεται ανταλλαγή σύντομων μηνυμάτων είναι μέσα από τη χρήση κινητών τηλεφώνων. Εκτός από αυτό τον τρόπο υπάρχουν και άλλες μέθοδοι αποστολής και λήψης μηνυμάτων SMS στις οποίες περιλαμβάνεται η χρησιμοποίηση προσωπικού υπολογιστή(Personal Computer - PC).

2.2.1 Οι πιθανοί τρόποι αποστολής SMS μηνυμάτων μέσω ενός PC

Η αποστολή σύντομων μηνυμάτων μέσα από ένα PC είναι εφικτή με δύο τρόπους:

- την άμεση σύνδεση του PC με το κέντρο υπηρεσίας για SMS
- τη σύνδεση του PC σε ένα GSM modem, μέσω του οποίου αποκτά πρόσβαση σε ένα δίκτυο GSM

Απ’ ευθείας σύνδεση προσωπικού υπολογιστή και κέντρου υπηρεσίας SMS

Ο συγκεκριμένος τρόπος αποστολής μηνυμάτων προτιμάται για περιπτώσεις στις οποίες υπάρχει υψηλή κίνηση μηνυμάτων από και προς τον χρήστη.

Ο χρήστης ενός PC μπορεί να συνδεθεί με ένα SMSC αφού γίνει πρώτα συμφωνία με την εταιρεία παροχής κινητής τηλεφωνίας στην οποία ανήκει το SMSC.

Συνήθως ο χρήστης αγοράζει το δικαίωμα αποστολής μιας παρτίδας μηνυμάτων ή γίνεται συμφωνία για σταθερή μηνιαία αποστολή μηνυμάτων.

Η ανταλλαγή μηνυμάτων γίνεται με τη χρήση κάποιας διεπαφής ή ενός πρωτοκόλλου που υποστηρίζεται από το SMSC.

Το πρωτόκολλο που υποστηρίζει το SMSC εξαρτάται από τις προδιαγραφές που έχουν τεθεί από την κατασκευάστρια εταιρεία του.

Η διεπαφή που παρέχεται για την αποστολή μηνυμάτων μπορεί να είναι κάποια ιστοσελίδα ή κάποια εφαρμογή λογισμικού.

Σύνδεση προσωπικού υπολογιστή με GSM modem

Αυτή η μέθοδος χρησιμοποιείται σε περιπτώσεις όπου έχουμε αποστολή και λήψη ενός περιορισμένου αριθμού μηνυμάτων.

Για την υλοποίηση της συγκεκριμένης μεθόδου απαιτείται ένα GSM modem. Το GSM modem λειτουργεί όπως και ένα dial-up modem, με τη διαφορά ότι δίνει στο PC με το οποίο συνδέεται τη δυνατότητα ασύρματης πρόσβασης σε ένα δίκτυο GSM.

Για την πρόσβαση στο δίκτυο απαιτείται η εγκατάσταση μιας έγκυρης κάρτας SIM στο GSM modem που θα χρησιμοποιηθεί.

Το GSM modem συνδέεται με ένα PC είτε εξωτερικά ή εσωτερικά.

Το modem συνδέεται με κάποιο desktop PC εξωτερικά χρησιμοποιώντας έναν από τους πολλούς διαθέσιμους τρόπους σύνδεσης, που μπορεί να είναι:

- με καλώδιο RS-232
- με καλώδιο USB
- με σύνδεση Bluetooth
- με σύνδεση υπερύθρων

Ενώ, το modem μπορεί να συνδεθεί εσωτερικά σε ένα φορητό υπολογιστή χρησιμοποιώντας μια κάρτα PCMCIA.

Βέβαια, υπάρχει και μια επιπλέον λύση που περιλαμβάνει τη χρήση ενός κινητού τηλεφώνου. Το κινητό τηλέφωνο συνδέεται με έναν από τρόπους σύνδεσης μιας εξωτερικής συσκευής και προσομοιώνει τον τρόπο λειτουργίας ενός GSM modem.

Το modem και ο υπολογιστής επικοινωνούν μέσα από το σύνολο των εντολών AT. Με τις κατάλληλες εντολές ο υπολογιστής καθοδηγεί το modem και ελέγχει την διαδικασία ανταλλαγής μηνυμάτων.

2.2.2 Οι εντολές AT

Η ονομασία των εντολών AT(AT Commands) προκύπτει από τη συντόμευση της λέξης Attention, αλλά και από το γεγονός ότι κάθε εντολή περιλαμβάνει το πρόθεμα "`AT`" ή "`at`", καθώς οι εντολές δεν διαφέρουν αν γράφονται με κεφαλαίους ή με πεζούς χαρακτήρες.

Οι εντολές χρησιμοποιούνται κυρίως για τον έλεγχο ενός ενσύρματου dial-up modem, όμως υπάρχει μια επέκταση του συνόλου εντολών που σχετίζεται με λειτουργίες του δικτύου GSM, και κατ' επέκταση με την αποστολή και λήψη μηνυμάτων SMS.

Αυτό το σύνολο υποστηρίζεται πλήρως από τα GSM modems, όμως οι συσκευές κινητών τηλεφώνων που χρησιμοποιούνται σαν GSM modems μπορεί να μην παρέχουν υποστήριξη για όλες τις εντολές.

Οι ακόλουθες είναι μερικές από τις λειτουργίες που υποστηρίζονται από τις εντολές AT:

- Λήψη βασικών πληροφοριών για τη συσκευή GSM modem ή για τη συσκευή κινητού τηλεφώνου, όπως:
  - το όνομα του κατασκευαστή (AT+CGMI)
  - ο αριθμός του μοντέλου (AT+CGMM)
  - ο αριθμός IMEI (AT+CGSN)
  - η έκδοση λογισμικού (AT+CGMR).
- Λήψη βασικών πληροφοριών σχετικά με το συνδρομητή, όπως
  - ο αριθμός MSISDN (AT+CNUM)
  - ο αριθμός IMSI (AT+CIMI).
- Διαχείριση των επαφών που βρίσκονται αποθηκευμένες στη συσκευή:
  - Ανάγνωση (AT+CPBR)
  - Εγγραφή (AT+CPBW)
  - Αναζήτηση (AT+CPBF)
- Εγκαθίδρυση σύνδεσης δεδομένων ή σύνδεσης φωνής με ένα απομακρυσμένο modem(μέσω των εντολών ATD και ATA).
- Αποστολή και λήψη fax (AT+F).
- Διαχείριση των μηνυμάτων SMS:
  - Αποστολή ενός αποθηκευμένου μηνύματος (AT+CMSS)
  - Άμεση αποστολή ενός μηνύματος (AT+CMGS)
  - Ανάγνωση ενός αποθηκευμένου μηνύματος (AT+CMGR)
  - Εμφάνιση των φακέλων μηνυμάτων (AT+CMGL)
  - Αποθήκευση ενός μηνύματος (AT+CMGW)
  - Διαγραφή ενός μηνύματος (AT+CMGD)
  - Ειδοποίηση για νέα μηνύματα που έφτασαν στη συσκευή (AT+CNMI)

Στις σελίδες που ακολουθούν θα ασχοληθούμε περισσότερο με τις εντολές που χρησιμοποιούνται για την αποστολή και τη λήψη σύντομων μηνυμάτων.

Η σύνταξη των συγκεκριμένων εντολών εξαρτάται από τον τρόπο λειτουργίας που υποστηρίζεται από το GSM modem ή το κινητό τηλέφωνο που μπορεί να χρησιμοποιείται.

Υπάρχουν δύο τρόποι λειτουργίας για τα σύντομα μηνύματα. Πιο συχνά συναντάται ο τρόπος λειτουργίας που ονομάζεται SMS PDU mode, ενώ λιγότερο συχνά χρησιμοποιείται ο τρόπος λειτουργίας SMS Text mode.

Η περίπτωση του SMS Text mode είναι η απλούστερη, καθώς το περιεχόμενο του μηνύματος έχει τη μορφή κειμένου και δεν απαιτείται από το χρήστη να γνωρίζει τη δομή που έχει το TPDU πλαίσιο ενός μηνύματος.

Αντίθετα με το SMS Text mode, στο SMS PDU mode απαιτείται η γνώση της δομής ενός TPDU πλαισίου. Ο χρήστης δημιουργεί το μήνυμα εισάγοντας τις κατάλληλες δεκαεξαδικές τιμές στα πεδία ενός TPDU πλαισίου.

Σε αυτό τον τρόπο λειτουργίας τα δεδομένα του χρήστη εισάγονται κωδικοποιημένα σύμφωνα με το σχήμα κωδικοποίησης που έχει επιλεγεί από αυτόν. Η χρήση του PDU mode μπορεί να φαίνεται πολύπλοκη, όμως παρέχει περισσότερες δυνατότητες στο χρήστη και έχει μεγαλύτερη ευελιξία απ' ότι η χρήση του Text mode.

Το συντακτικό των εντολών AT

Το σύνολο των εντολών AT που σχετίζονται με τις λειτουργίες ενός GSM modem ή ενός κινητού τηλεφώνου έχει ορισμένους κανόνες συντακτικού που οφείλουμε να τηρούμε κατά τη χρήση τους.

Κάθε εντολή ξεκινά με "AT" και τελειώνει με τον χαρακτήρα της επιστροφής φορέα(Carriage Return), τον οποίο θα συμβολίζουμε ως <CR>.

Ένα παράδειγμα του κανόνα έχουμε όταν επιθυμούμε να εμφανιστεί το όνομα του κατασκευαστή της συσκευής. Τότε ξεκινάμε πληκτρολογώντας "AT", ακολουθεί η εντολή "+CGMI" και κλείνουμε με <CR>, ο χαρακτήρας επιστροφής ισοδυναμεί με ένα πάτημα του πλήκτρου Enter. Η ολοκληρωμένη εντολή έχει αυτή τη μορφή: AT+CGMI<CR>

Σε μια γραμμή εντολών είναι δυνατό να περιέχονται παραπάνω από μια εντολές AT. Σε αυτή την περίπτωση μόνο η πρώτη εντολή λαμβάνει το πρόθεμα "AT", ενώ οι υπόλοιπες εντολές χωρίζονται με τον χαρακτήρα ";". Ένα παράδειγμα δύο εντολών που βρίσκονται στην ίδια γραμμή: AT+CGMI;+CMGL<CR>

Οι απαντήσεις και τα αποτελέσματα που έρχονται από τις εντολές ξεκινούν με χαρακτήρα επιστροφής φορέα και με χαρακτήρα τροφοδοσίας γραμμής(Line Feed) και τελειώνουν με τον ίδιο τρόπο.

Μόλις ολοκληρωθεί η διαδικασία της απάντησης εμφανίζεται ένα "OK", το οποίο υποδεικνύει ότι δεν υπάρχουν άλλα δεδομένα για αποστολή από τη συσκευή του modem ή κινητού τηλεφώνου προς το PC. Για παράδειγμα το αποτέλεσμα της εντολής AT+CGMI<CR> θα είναι:

<CR><LF>Όνομα κατασκευαστή<CR><LF>
<CR><LF>OK<CR><LF>

Σε αυτό το σημείο μπορούμε με τη βοήθεια ενός παραδείγματος να συγκρίνουμε τη διαφορά που έχει η σύνταξη της εντολής AT+CGMS όταν χρησιμοποιείται Text mode και όταν χρησιμοποιείται PDU mode.

Ας υποθέσουμε ότι θέλουμε να στείλουμε το κείμενο "It is easy to send text messages." στον αριθμό +85291234567. Σε Text mode έχουμε:

AT+CMGS="+85291234567"<CR>It is easy to send text messages.<Ctrl+z>

Ενώ σε PDU mode, όπου εισάγεται το ίδιο κείμενο χρησιμοποιώντας το σχήμα κωδικοποίησης GSM-7, η εντολή έχει την ακόλουθη σύνταξη:

```
AT+CMGS=42<CR>07915892000000F001000B915892214365F70000214
93A283D0795C3F33C88FE06CDCB6E32885EC6D341EDF27C1E3E97E72E
<Ctrl+z>
```

2.2.3 Η διαδικασία αποστολής ενός σύντομου μηνύματος μέσω PC

Σε αυτό το σημείο θα δούμε αναλυτικότερα την περίπτωση στην οποία συνδέονται ένα κινητό τηλέφωνο και ένας προσωπικός υπολογιστής.

Το κινητό τηλέφωνο λειτουργεί σαν GSM modem και επικοινωνεί με το PC μέσω των εντολών AT.

Χρησιμοποιώντας τις κατάλληλες εντολές συνθέτουμε ένα σύντομο μήνυμα, το οποίο αργότερα προωθούμε μέσω του κινητού τηλεφώνου/GSM modem.

Για την πραγματοποίηση αυτής της διαδικασίας απαιτείται μια έγκυρη κάρτα SIM, η οποία εισάγεται στην συσκευή κινητού τηλεφώνου που πρόκειται να συνδεθεί με τον προσωπικό υπολογιστή.

Η σύνδεση του τηλεφώνου και του PC εξαρτάται από τις δυνατότητες διασύνδεσης που παρέχονται από τη συσκευή του κινητού τηλεφώνου.

Έτσι είναι δυνατόν να έχουμε είτε ενσύρματη διασύνδεση, με κάποιο καλώδιο που συνδέεται σε θύρα RS-232 ή USB, είτε ασύρματη διασύνδεση, χρησιμοποιώντας κάποια ζεύξη υπερύθρων ή Bluetooth.

Είναι πολύ πιθανό μετά την εγκαθίδρυση της σύνδεσης να χρειαστεί η εγκατάσταση οδηγών(drivers) που θα επιτρέπουν τη λειτουργία του τηλεφώνου σαν ασύρματο modem.

Συνήθως οι drivers παρέχονται από τον κατασκευαστή του τηλεφώνου σε κάποιο συνοδευτικό οπτικό δίσκο ή βρίσκονται διαθέσιμοι στην επίσημη ιστοσελίδα της κατασκευάστριας εταιρείας.

Έχοντας πια τη συσκευή κινητού τηλεφώνου συνδεδεμένη και σε θέση αναμονής, απομένει να βρεθεί ένας τρόπος με τον οποίο θα υποβάλλονται οι εντολές AT στη συσκευή.

Η λύση δίνεται με ένα πρόγραμμα τερματικού, όπως είναι το Microsoft HyperTerminal που βρίσκεται ενσωματωμένο στις περισσότερες εκδόσεις του λειτουργικού συστήματος Microsoft Windows.

Σαν εναλλακτική λύση μπορούμε να χρησιμοποιήσουμε το terminal emulator Putty που διατίθεται ελεύθερα από το διαδικτυακό τόπο: http://www.putty.org/.

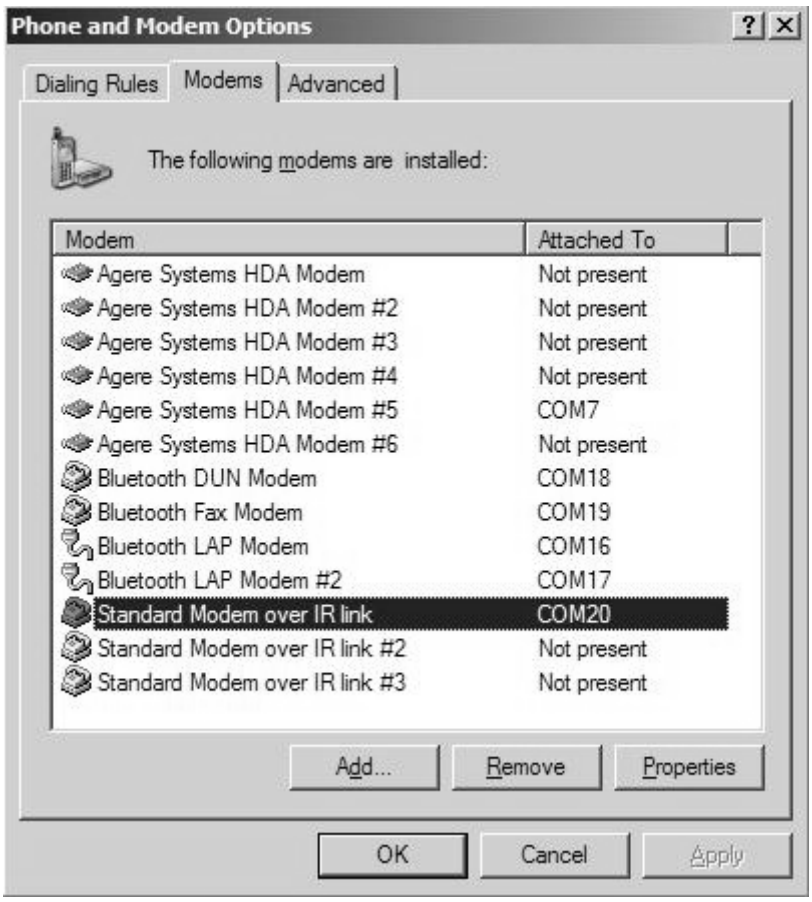
Στις εικόνες που ακολουθούν κατά την παρουσίαση της διαδικασίας αποστολής ενός σύντομου μηνύματος χρησιμοποιείται το HyperTerminal.

Όμως, πριν να περάσουμε στο HyperTerminal, πρέπει να γνωρίζουμε σε ποια θύρα επικοινωνιών COM συνδέεται η συσκευή κινητού τηλεφώνου/GSM modem.

Αυτό το μαθαίνουμε από τον πίνακα ελέγχου του λειτουργικού συστήματος, επιλέγοντας την κατηγορία "Phone and Modem Options".

Από εκεί επιλέγουμε το tab που αναγράφει τη λέξη "Modems" και βρίσκουμε το modem που χρησιμοποιείται και το port που του αντιστοιχεί.

Στην εικόνα που ακολουθεί βλέπουμε το port που αντιστοιχεί για την σύνδεση του modem μέσω υπερύθρων.



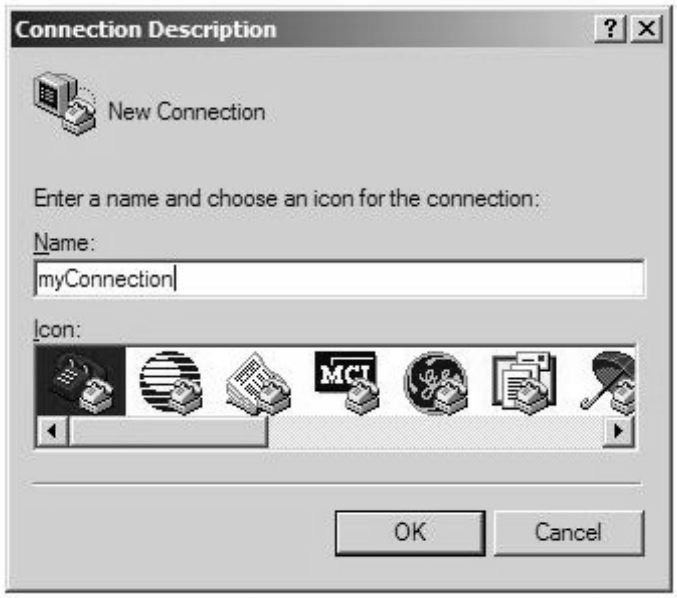
Εικόνα 21: Εύρεση του port που χρησιμοποιείται για την επικοινωνία ανάμεσα στο PC και το GSM modem

Γνωρίζοντας πια το port που χρησιμοποιείται για τη σύνδεση του κινητού τηλεφώνου/GSM modem περνάμε στη δημιουργία σύνδεσης με το HyperTerminal.

Για να ξεκινήσει η λειτουργία του HyperTerminal ακολουθούμε από το μενού του λειτουργικού συστήματος Microsoft Windows τη διαδρομή:
"`Start -> Programs -> Accessories -> Communications -> HyperTerminal`".

Με την εκκίνηση του HyperTerminal εμφανίζεται το πρώτο πλαίσιο διαλόγου για τη δημιουργία μιας νέας σύνδεσης.

Στην εικόνα που ακολουθεί έχουμε το πρώτο πλαίσιο που εμφανίζεται στο HyperTerminal και αφορά στην περιγραφή της υπό δημιουργία σύνδεσης.



Εικόνα 22: Πλαίσιο διαλόγου για τη δημιουργία νέας σύνδεσης στο Microsoft HyperTerminal

Μετά την εισαγωγή του ονόματος που θα έχει η σύνδεση και την επιλογή ενός εικονιδίου που θα την αντιπροσωπεύει, περνάμε στο επόμενο πλαίσιο, όπου επιλέγουμε τη θύρα COM με την οποία θέλουμε να συνδεθεί το HyperTerminal.

Στην περίπτωση μας επιλέγουμε τη θύρα στην οποία συνδέεται το κινητό τηλέφωνο/GSM modem με τον υπολογιστή.

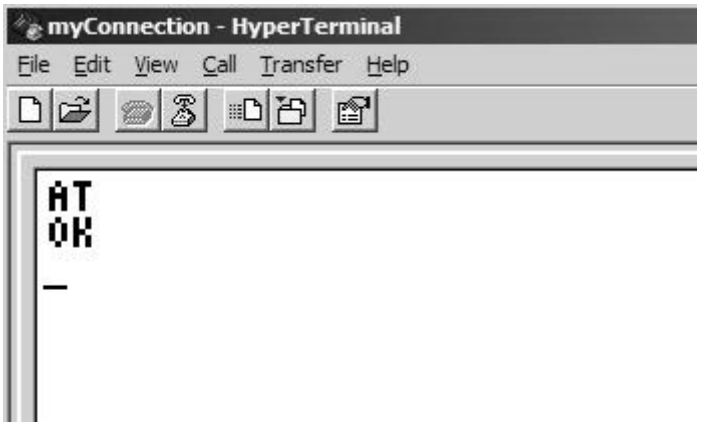
Εικόνα 23: Πλαίσιο για την επιλογή της θύρας επικοινωνιών

Από εκεί περνάμε στο κεντρικό παράθυρο του HyperTerminal, όπου θα πληκτρολογήσουμε ορισμένες εντολές με τις οποίες θα δοκιμάσουμε την επικοινωνία ανάμεσα στο κινητό τηλέφωνο/GSM modem και τον προσωπικό υπολογιστή.

Σε αυτό το σημείο πρέπει να ξεκαθαρίσουμε ότι όλες οι εντολές που αναφέρονται στις επόμενες σελίδες συνοδεύονται πάντα από εισαγωγικά. Όμως η εισαγωγή τους στο HyperTerminal δεν απαιτεί την ύπαρξη εισαγωγικών.

Το ίδιο συμβαίνει και με τις απαντήσεις που προκύπτουν από την εκτέλεση των εντολών, όσες εμφανίζονται στο HyperTerminal δεν έχουν εισαγωγικά, όμως όποια απάντηση αναφέρεται στις σελίδες που ακολουθούν έχει πάντα εισαγωγικά.

Για να εξετάσουμε αν το modem είναι σε θέση να επικοινωνεί με τη χρήση των εντολών AT πληκτρολογούμε "AT" στο κεντρικό παράθυρο του HyperTerminal. Αν η απάντηση που λάβουμε είναι "OK", τότε επιβεβαιώνεται ότι οι δύο πλευρές επικοινωνούν με εντολές AT.

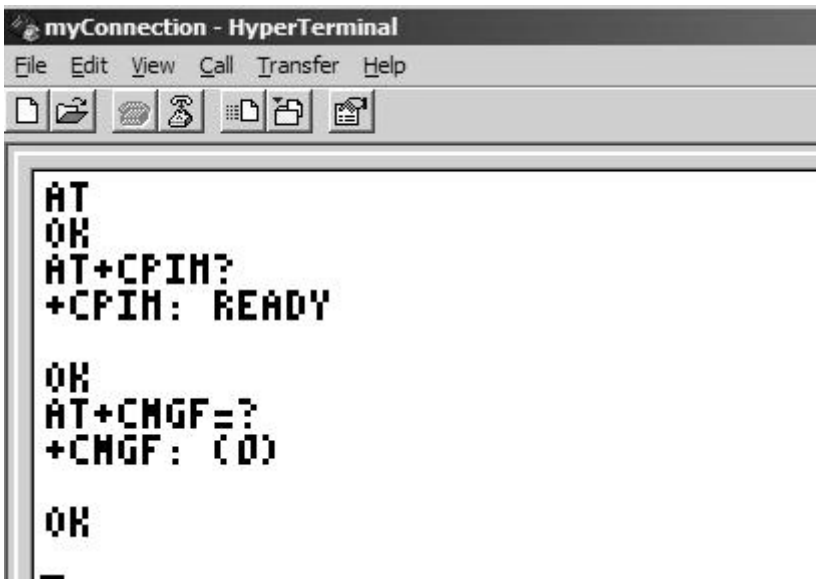


Εικόνα 24: Εισαγωγή της εντολής "AT"

Στη συνέχεια πληκτρολογούμε "AT+CPIN?". Με αυτή την εντολή ερωτάται το modem αν περιμένει την εισαγωγή κωδικού PIN. Με την απάντηση "+CPIN: READY", το modem ενημερώνει ότι είναι έτοιμο προς χρήση και δεν απαιτεί την εισαγωγή κάποιου PIN. Σε διαφορετική περίπτωση, θα στέλναμε τον αριθμό PIN με την εντολή "AT+CPIN=<τιμή PIN>".

Εικόνα 25: Εισαγωγή της εντολής "AT+CPIN"

Αμέσως μετά εξετάζουμε αν υποστηρίζονται σύντομα μηνύματα από τη συσκευή, και αν η απάντηση είναι θετική, τότε γίνεται γνωστό ποιος τρόπος λειτουργίας υποστηρίζεται από το GSM modem. Αυτό γίνεται με την εντολή "AT+CMGF=?".



Εικόνα 26: Εισαγωγή της εντολής "AT+CMGF"

Η απάντηση που λαμβάνουμε, "+CMGF: (0)", μας γνωστοποιεί ότι το modem υποστηρίζει μόνο το SMS PDU mode.

Αν το modem υποστήριζε αποκλειστικά το SMS Text mode θα είχαμε λάβει την απάντηση "+CMGF: (1)".

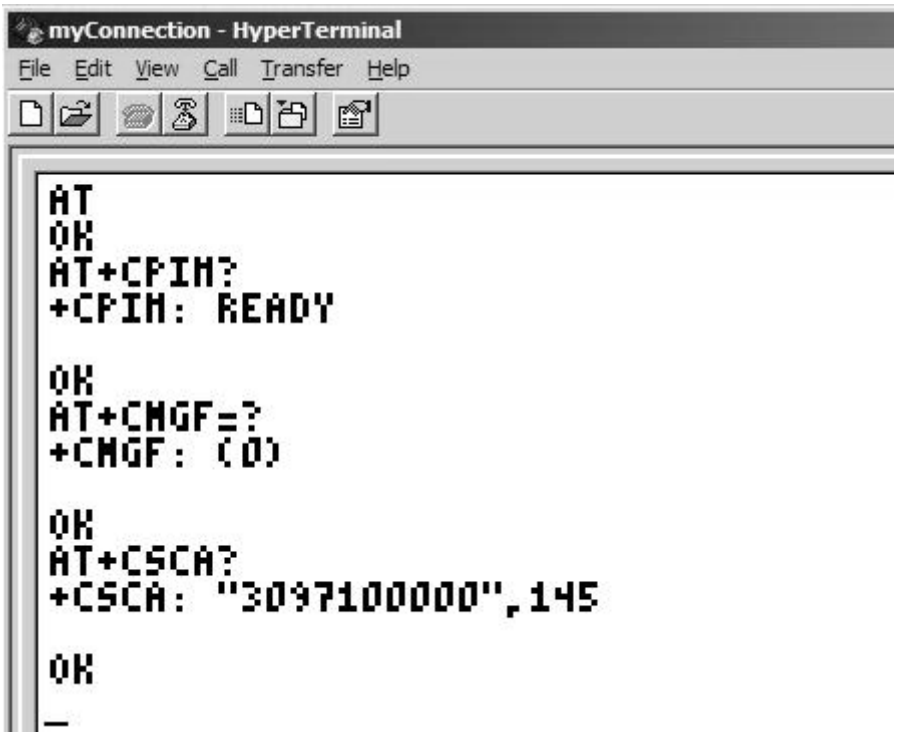
Ενώ στην περίπτωση που το modem υποστήριζε και τους δύο διαθέσιμους τρόπους λειτουργίας θα επέστρεφε την απάντηση "+CMGF: (0, 1)". Τότε θα έπρεπε να βρούμε ποιος τρόπος βρισκόταν ήδη σε χρήση, πληκτρολογώντας την εντολή "AT+CMGF?". Στη συνέχεια θα έπρεπε να επιλέξουμε το mode που μας εξυπηρετεί, αυτό γίνεται με την εντολή "AT+CMGF=0" αν θέλουμε SMS PDU mode, ενώ αν θέλουμε SMS Text mode πληκτρολογούμε "AT+CMGF=1".

Στο επόμενο μας βήμα ελέγχουμε αν υπάρχει αποθηκευμένος στην κάρτα SIM ο αριθμός τηλεφώνου που αντιστοιχεί στο κέντρο SMSC. Συνηθίζεται από τις εταιρείες παροχής κινητής τηλεφωνίας να εισάγουν εκ των προτέρων στην κάρτα SIM τον αριθμό του SMSC μαζί με άλλες ρυθμίσεις.

Πάντως, αν δεν υπάρχει κάποια ρύθμιση για το κέντρο μηνυμάτων, τότε μπορούμε να θέσουμε εμείς έναν έγκυρο αριθμό για το SMSC που αντιστοιχεί στο δίκτυο που ανήκει η κάρτα SIM.

Για να ελέγξουμε την τιμή του SMSC πληκτρολογούμε "AT+CSCA?". Η απάντηση που αναμένουμε αποτελείται από δύο πεδία, με το πρώτο να αντιπροσωπεύει τον τηλεφωνικό αριθμό του SMSC και το δεύτερο πεδίο να υποδεικνύει αν ο αριθμός περιλαμβάνει τον χαρακτήρα '+' μαζί με το διεθνή τηλεφωνικό κωδικό της χώρας στην οποία βρίσκεται το δίκτυο.

Στην εικόνα που ακολουθεί έχουμε την απάντηση στην εντολή "AT+CSCA?", και βλέπουμε ότι ο αριθμός "+3097100000" βρίσκεται στην κάρτα SIM του GSM modem, ενώ η τιμή 145 υποδηλώνει ότι ο αριθμός είναι αποθηκευμένος με τη διεθνή μορφή.



Εικόνα 27: Εισαγωγή της εντολής "AT+CSCA"

Είμαστε πλέον έτοιμοι να περάσουμε στην εντολή "AT+CMGS", με την οποία αποστέλλεται ένα μήνυμα SMS. Η εντολή συντάσσεται με τον ακόλουθο τρόπο:
"AT+CMGS=[μήκος_TPDU]<CR>[αριθμός_SMSC][πλαίσιο_TPDU]
<Ctrl+z>"

Η πρώτη παράμετρος της εντολής, το μήκος του TPDU, είναι ένας αριθμός που αντιπροσωπεύει το πλήθος των οκτάδων δυαδικών ψηφίων(octets) που καταλαμβάνει

το πλαίσιο του TPDU. Για αυτό και η συγκεκριμένη παράμετρος υπολογίζεται μόνο μετά την ολοκλήρωση της δημιουργίας της δεύτερης παραμέτρου.

Η δεύτερη παράμετρος περιλαμβάνει δύο τμήματα. Το πρώτο τμήμα είναι προαιρετικό, και περιέχει τον αριθμό του SMSC όπου πρόκειται να δρομολογηθεί το μήνυμα. Το δεύτερο τμήμα περιέχει τα πεδία του πλαισίου TPDU που πρόκειται να μεταδοθεί. Οι τιμές των πεδίων είναι σε δεκαεξαδική μορφή, έτσι έχουμε δύο οκτάδες δυαδικών ψηφίων να αντιστοιχούν σε μία τιμή.

Ας δούμε ως παράδειγμα τη μορφή που θα έχει το πλαίσιο TPDU ενός μηνύματος SMS-SUBMIT, το οποίο έχει αποδέκτη τον συνδρομητή με αριθμό "+306982ΑΒΓΔΕΖ" και περιέχει το κείμενο "It is easy to send text messages."

Ταυτόχρονα, θα θεωρήσουμε ότι το πλαίσιο υποβάλλεται στο SMSC με αριθμό "+3097100000", οπότε θα σχηματίσουμε και το αντίστοιχο πεδίο που προηγείται του πλαισίου TPDU.

Το τμήμα με τον αριθμό του SMSC είναι αυτό που σχηματίζεται πρώτο, αποτελείται από τρία μέρη και έχει την ακόλουθη δομή:
[μήκος_των_δύο_πεδίων_που_ακολουθούν][τύπος_αριθμού_SMSC] [αριθμός_SMSC]

Το πρώτο μέρος προσδιορίζει το πλήθος από οκτάδες δυαδικών ψηφίων που αποτελούν τα δύο τμήματα που ακολουθούν.

Στο δεύτερο μέρος, που καταλαμβάνει μιαν οκτάδα δυαδικών ψηφίων, διευκρινίζεται αν ο αριθμός του SMSC περιλαμβάνει τον διεθνή τηλεφωνικό κωδικό.

Έτσι το συγκεκριμένο τμήμα, στην περίπτωση που είναι ξεκάθαρο ότι ο αριθμός είναι διεθνής, λαμβάνει τη δεκαδική τιμή 145 ή τη δεκαεξαδική τιμή 91.

Διαφορετικά, στην περίπτωση που ο αριθμός του SMSC δεν είναι σίγουρο ότι περιέχει διεθνή κωδικό χρησιμοποιείται η δεκαδική τιμή 129 ή η δεκαεξαδική τιμή 81.

Ο αριθμός SMSC που χρησιμοποιείται στο παράδειγμα μας περιλαμβάνει διεθνή κωδικό, οπότε χρησιμοποιούμε τη δεκαεξαδική τιμή 91.

Στο τρίτο τμήμα αποθηκεύεται με δεκαεξαδική μορφή ο αριθμός του SMSC. Η τιμή που χρησιμοποιείται λαμβάνεται με τον ακόλουθο τρόπο που αποτελείται από τρία βήματα:

1. Ο αριθμός χωρίζεται σε ζεύγη ψηφίων, οπότε ο αριθμός "3097100000" του παραδείγματος γίνεται 30 97 10 00 00. Αν ο αριθμός είχε περιττό πλήθος ψηφίων τότε μένει το τελευταίο ψηφίο μόνο του.

2. Εξετάζεται το πλήθος των ψηφίων του αριθμού. Αν ο αριθμός αποτελείται από περιττό πλήθος ψηφίων, έχει δηλαδή ένα ψηφίο μόνο του, τότε προστίθεται ένα "F" στο συγκεκριμένο ψηφίο και σχηματίζεται ζεύγος. Με

αυτό τον τρόπο θα δημιουργηθούν σωστά οι οκτάδες δυαδικών ψηφίων που αντιπροσωπεύουν τον αριθμό του SMSC. Στο παράδειγμα μας ο αριθμός έχει άρτιο πλήθος ψηφίων, οπότε αυτή η ενέργεια δεν είναι απαραίτητη.

3. Πραγματοποιείται ανταλλαγή θέσης ανάμεσα στα ψηφία που σχηματίζουν κάθε ζεύγος. Το αποτέλεσμα αποθηκεύεται σαν τη δεκαεξαδική τιμή που αντιπροσωπεύει τον αριθμό του SMSC. Στο παράδειγμα μας με τον αριθμό "3097100000" τα ζεύγη που λαμβάνονται από τα προηγούμενα βήματα, για την περίπτωση αυτή είχαμε πέντε ζεύγη με τιμές 30 97 10 00 00, μετατρέπονται σε πέντε αντίστοιχες οκτάδες δυαδικών ψηφίων με τιμές 03 79 01 00 00.

Σε αυτό το σημείο μπορούμε να σχηματίσουμε ολοκληρωμένη εικόνα για το πεδίο του αριθμού SMSC. Αφού είμαστε σε θέση να υπολογίσουμε το πλήθος από octets που απαρτίζει το δεύτερο και το τρίτο τμήμα του πεδίου με τον αριθμό SMSC, και μετά να θέσουμε την τιμή στο πρώτο τμήμα.

Για το παράδειγμα μας έχουμε συνολικά έξι octets, αφού απαιτείται μια οκτάδα δυαδικών ψηφίων για τον τύπο του αριθμού και πέντε οκτάδες για τον αριθμό SMSC. Οπότε το πεδίο λαμβάνει τις τιμές: [06] [91] [0379010000]

Όπως αναφέρθηκε και νωρίτερα το συγκεκριμένο πεδίο είναι προαιρετικό. Αρκεί η κάρτα SIM που χρησιμοποιείται στο κινητό τηλέφωνο/GSM modem να έχει αποθηκευμένο τον επιθυμητό αριθμό SMSC.

Με την παράλειψη του πεδίου ή την εισαγωγή της τιμής 00, δίνεται η εντολή στη συσκευή να χρησιμοποιήσει τον αριθμό του SMSC που έχει διαθέσιμο από την κάρτα SIM.

Μετά από αυτό περνάμε στη δημιουργία του πλαισίου TPDU που πρόκειται να μεταδοθεί. Υπενθυμίζουμε ότι ως παράδειγμα θα χρησιμοποιήσουμε ένα μήνυμα τύπου SMS-SUBMIT, το οποίο έχει αποδέκτη τον συνδρομητή με αριθμό "+306982ΑΒΓΔΕΖ" και περιέχει το κείμενο "It is easy to send text messages.".

Το πλαίσιο ενός SMS-SUBMIT μηνύματος έχει τη δομή του σχήματος που ακολουθεί.

	Bit Index							
No. of octets	7	6	5	4	3	2	1	0
1	TP-RP	TP-UDHI	TP-SRR	TP-VPF		TP-RD	TP-MTI	
1	TP-Message Reference							
2…12	TP-Destination Address							
1	TP-Protocol Identifier							
1	TP-Data Coding Scheme							
0,1 or 7	TP-Validity Period							
1	TP-User Data Length							
0…140	TP-User Data							

Εικόνα 28: Η δομή που έχει το πλαίσιο ενός SMS-SUBMIT μηνύματος

Πριν τη συμπλήρωση των τιμών του πλαισίου πρέπει να παρουσιάσουμε τον τρόπο με τον οποίο κωδικοποιείται το κείμενο του μηνύματος σύμφωνα με το σχήμα κωδικοποίησης χρησιμοποιείται.

Το GSM-7 αποτελεί το προκαθορισμένο σχήμα κωδικοποίησης. Σε κάθε χαρακτήρα του μηνύματος αντιστοιχούν 7 δυαδικά ψηφία. Όμως το πεδίο του πλαισίου που δέχεται τους κωδικοποιημένους χαρακτήρες είναι οργανωμένο σε οκτάδες δυαδικών ψηφίων.

Αυτό σημαίνει ότι αν μεταδίδονταν με αυτό τον τρόπο οι χαρακτήρες θα ξοδευόταν ένα bit για κάθε ένα χαρακτήρα. Για την αποφυγή αυτού του γεγονότος, και για την βελτιστοποίηση της μετάδοσης, υπάρχει η ακόλουθη διαδικασία με την οποία οι χαρακτήρες των 7-bit οργανώνονται σε οκτάδες δυαδικών ψηφίων.

Στον παρακάτω πίνακα εμφανίζονται οκτώ χαρακτήρες μεγέθους 7 δυαδικών ψηφίων, οι οποίοι οργανώνονται σε οκτάδες δυαδικών ψηφίων.

Χαρακτήρες							
A	B	C	D	E	F	G	H
A_6-A_0	B_6-B_0	C_6-C_0	D_6-D_0	E_6-E_0	F_6-F_0	G_6-G_0	H_6-H_0
B_0A_6-A_0	$C_1C_0B_6$-B_1	D_2-D_0C_6-C_2	E_3-E_0D_6-D_3	F_4-F_0E_6-E_4	G_5-G_0F_6-F_5	H_6-H_0G_6	

Πίνακας 1: Η οργάνωση χαρακτήρων μεγέθους 7 δυαδικών ψηφίων σε οκτάδες bit

Ξεκινάμε από το χαρακτήρα A, με τα 7 bit που του αντιστοιχούν να απεικονίζονται με A_0 ως A_6. Η θέση του περισσότερο σημαντικού bit είναι κενή, άρα απαιτείται ένα επιπλέον bit για να δημιουργηθεί μια οκτάδα δυαδικών ψηφίων. Το bit αυτό θα έρθει από το χαρακτήρα B. Πιο συγκεκριμένα το B_0 ή διαφορετικά το λιγότερο σημαντικό bit του B.

Ο χαρακτήρας B μένει με 6 bit, τα οποία ολισθαίνουν 1 θέση προς τα δεξιά και καλύπτουν τη θέση που έμεινε κενή από το B_0. Οι θέσεις των 2 σημαντικότερων bit είναι κενές, έτσι έρχονται τα δύο λιγότερα σημαντικά bit του χαρακτήρα C.

Ο C έχει μείνει με 5 bit, πραγματοποιείται ολίσθηση 2 θέσεων προς τα δεξιά, και καλύπτεται το κενό από τα C_0 και C_1 που αναχώρησαν για τις θέσεις των δύο σημαντικότερων bit του B. Για τη συμπλήρωση οκτάδας έρχονται 3 bit από το γειτονικό χαρακτήρα D και αποθηκεύονται στις θέσεις των σημαντικότερων bit.

Η διαδικασία επαναλαμβάνεται με παρόμοια λογική μέχρι και τον τελευταίο χαρακτήρα του μηνύματος. Με αυτό τον τρόπο αξιοποιούνται όλα τα bits που βρίσκονται στο πεδίο με τα δεδομένα του χρήστη, φτάνοντας έτσι στη μέγιστη χωρητικότητα του πεδίου που είναι οι 160 χαρακτήρες.

Εφαρμόζοντας την παραπάνω μέθοδο στο κείμενο “It is easy to send text messages.” που χρησιμοποιούμε στο παράδειγμα μας θα λάβουμε τα octets που θα εισαχθούν στο πεδίο με τα δεδομένα του χρήστη. Το μήνυμα αποτελείται από 33 χαρακτήρες και μεταφράζεται σε 29 οκτάδες δυαδικών ψηφίων, οι οποίες έχουν τις εξής δεκαεξαδικές τιμές: `49 3A 28 3D 07 95 C3 F3 3C 88 FE 06 CD CB 6E 32 88 5E C6 D3 41 ED F2 7C 1E 3E 97 E7 2E`

Περνάμε στη συμπλήρωση των πεδίων του πλαισίου, ξεκινώντας από την πρώτη οκτάδα δυαδικών ψηφίων που περιλαμβάνει τα πεδία για το Reply Path, για την ύπαρξη επικεφαλίδας στα δεδομένα του χρήστη, για το αίτημα αναφοράς, για την περίοδο ισχύος του μηνύματος, για την απόρριψη διπλών μηνυμάτων από το SMSC και το πεδίο με την ένδειξη του τύπου του μηνύματος.

Θα δώσουμε στα πρώτα οκτώ bit την τιμή `00000001`, ή πιο σύντομα τη δεκαεξαδική τιμή `01`. Έτσι στα πεδία της πρώτης οκτάδας αντιστοιχούν οι τιμές που εμφανίζονται στον πίνακα που ακολουθεί

TP-RP	TP-UDHI	TP-SRR	TP-VPF	TP-RD	TP-MTI
0	0	0	00	0	01

Πίνακας 2: Οι δυαδικές τιμές που εισάγονται στα πεδία της πρώτης οκτάδας του πλαισίου TPDU

Δίνοντας την τιμή `01` στο πεδίο TP-MTI υποδεικνύεται ότι το μήνυμα που μεταδίδεται είναι SMS-SUBMIT.

Με την τιμή `0` στο πεδίο TP-RD το SMSC θα δέχεται τα πολλαπλά αντίγραφα κάποιου μηνύματος.

Η τιμή `00` που εισάγεται στο πεδίο TP-VPF υποδεικνύει ότι στο συγκεκριμένο μήνυμα δεν έχει οριστεί κάποια περίοδος ισχύος, κατ' επέκταση το πεδίο TP-Validity Period απουσιάζει από το συγκεκριμένο πλαίσιο.

Στο πεδίο TP-SRR εισάγεται η τιμή `0`, έτσι δεν απαιτείται αναφορά για την κατάσταση του μηνύματος.

Θέτοντας την τιμή `0` στο πεδίο TP-UDHI ξεκαθαρίζεται ότι το πεδίο με τα δεδομένα του χρήστη δεν περιέχει κάποια επιπλέον επικεφαλίδα.

Το πεδίο TP-RP λαμβάνει την τιμή `0`, έτσι δεν χρησιμοποιείται Reply Path.

Συνεχίζουμε με το πεδίο TP-Message Reference, όπου παίρνει την τιμή `0` για κάθε ένα από τα οκτώ δυαδικά ψηφία από τα οποία αποτελείται. Με αυτό τον τρόπο η συσκευή κινητού τηλεφώνου/GSM modem αναλαμβάνει την αυτόματη ανάθεση ενός αριθμού αναφοράς στο μήνυμα.

Το πεδίο που ακολουθεί περιέχει τη διεύθυνση προορισμού και χωρίζεται σε τρία τμήματα.

Το πρώτο τμήμα καταλαμβάνει μιαν οκτάδα δυαδικών ψηφίων και αντιπροσωπεύει το πλήθος των ψηφίων του αριθμού τηλεφώνου για τον οποίο προορίζεται το μήνυμα.

Στο τρίτο τμήμα αποθηκεύεται ο αριθμός ακολουθώντας τον τρόπο που είδαμε νωρίτερα, ενώ το δεύτερο τμήμα ξεκαθαρίζει αν ο αριθμός είναι διεθνής.

Για το παράδειγμα μας χρησιμοποιούμε τον αριθμό "`+306982ΑΒΓΔΕΖ`", ο οποίος αποτελείται από 12 ψηφία και περιλαμβάνει διεθνή τηλεφωνικό κωδικό. Οπότε το πεδίο TP-Destination Address παίρνει τη δεκαεξαδική τιμή `0C 91 03 96 28 ΒΑ ΔΓ ΖΕ`.

Το επόμενο πεδίο φιλοξενεί την τιμή TP-Protocol Identifier. Τα οκτώ δυαδικά ψηφία του πεδίου παίρνουν την τιμή `0`, καθώς χρησιμοποιείται η πιο κοινή περίπτωση με την επικοινωνία να διεξάγεται ανάμεσα σε δύο οντότητες σύντομων μηνυμάτων.

Το σχήμα κωδικοποίησης που χρησιμοποιείται είναι το GSM-7. Για αυτό το λόγο, το πεδίο TP-Data Coding Scheme παίρνει την τιμή `0` για όλα τα δυαδικά του ψηφία.

Το πεδίο που ακολουθεί περιέχει την περίοδο ισχύος του μηνύματος. Όμως στο πεδίο TP-VPF της πρώτης οκτάδας επιλέχθηκε να μην υπάρχει κάποια περίοδος ισχύος για το συγκεκριμένο μήνυμα. Έτσι το πεδίο TP-Validity Period παραλείπεται και κατ' επέκταση δεν λαμβάνει κάποια τιμή.

Τα δύο τελευταία πεδία συνδέονται στενά με το σχήμα κωδικοποίησης που χρησιμοποιείται.

Στο παράδειγμα μας χρησιμοποιείται το σχήμα κωδικοποίησης GSM-7, οπότε στο πεδίο TP-User Data Length αποθηκεύεται το πλήθος των χαρακτήρων που βρίσκονται στο πεδίο TP-User Data. Στην περίπτωση μας έχουμε τη μετάδοση `33`

χαρακτήρων, έτσι το TP-User Data Length παίρνει τη δεκαεξαδική τιμή 21, που αντιστοιχεί στη δεκαδική τιμή του 33.

Είδαμε νωρίτερα ότι οι 33 χαρακτήρες των 7 bit κωδικοποιούνται σε 29 οκτάδες δυαδικών ψηφίων. Με αυτή τη μορφή αποθηκεύονται στο πεδίο TP-User Data και έτσι βελτιστοποιείται η μετάδοση τους.

Σε αυτό το σημείο μπορούμε να παρουσιάσουμε την ολοκληρωμένη εικόνα που θα έχει το TPDU πλαίσιο του μηνύματος SMS-SUBMIT.

Πεδίο	**Δεκαεξαδική τιμή**	**Πλήθος Οκτάδων**
TP-RP, TP-UDHI, TP-SRR,TP-VPF, TP-RD, TP-MTI	01	1
TP-Message Reference	00	1
TP-Destination Address	0C91039628ΒΑΔΓΖΕ	8
TP-Protocol Identifier	00	1
TP-Coding Scheme	00	1
TP-Validity Period	Δεν χρησιμοποιείται κάποια τιμή	0
TP-User Data Length	21	1
TP-User Data	493A283D0795C3F33C88FE06CDCB 6E32885EC6D341EDF27C1E3E97E72E	29

Πίνακας 3: Παρουσίαση των δεκαεξαδικών τιμών που εισάγονται στα πεδία του πλαισίου TPDU

Το μήκος του TPDU πλαισίου, και κατ' επέκταση το συνολικό πλήθος των οκτάδων δυαδικών ψηφίων που το απαρτίζουν, είναι 42. Έτσι έχουμε στη διάθεση μας όλες τις απαραίτητες παραμέτρους που θα επιτρέψουν τη σύνταξη της εντολής "AT+CMGS", με την οποία αποστέλλεται ένα μήνυμα SMS.

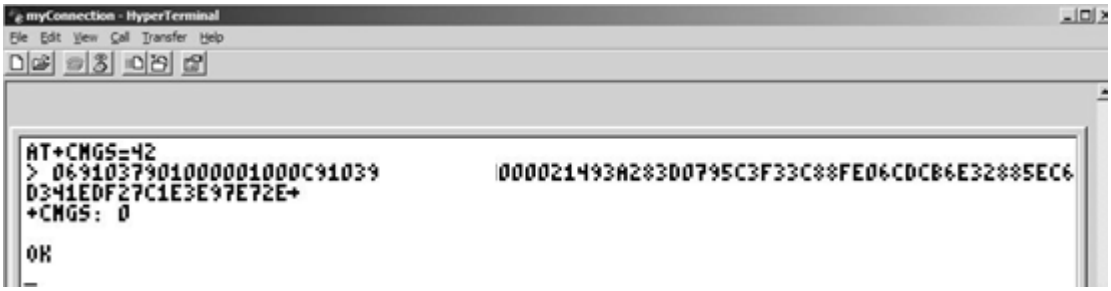
Να υπενθυμίσουμε ότι η εντολή συντάσσεται με τον ακόλουθο τρόπο:
"AT+CMGS=[μήκος_TPDU]<CR>[αριθμός_SMSC][πλαίσιο_TPDU]
<Ctrl+z>"

Οπότε επιστρέφουμε στο παράθυρο του HyperTerminal και ξεκινάμε την εισαγωγή των τιμών που υπολογίσαμε στην εντολή "AT+CMGS". Αρχικά πληκτρολογούμε "AT+CMGS=42" και μετά πατάμε το πλήκτρο Enter, το οποίο ισοδυναμεί με το Carriage Return. Στη συνέχεια ο κέρσορας πηγαίνει στην αρχή της επόμενης γραμμής και περιμένει την εισαγωγή του αριθμού SMSC και του πλαισίου TPDU. Έτσι πληκτρολογούμε τη δεκαεξαδική τιμή "06910379010000", που αντιστοιχεί στον αριθμό του SMSC.

Χωρίς να αφήσουμε κάποιο κενό, συνεχίζουμε με την πληκτρολόγηση του πλαισίου TPDU. Εισάγουμε δηλαδή τη δεκαεξαδική τιμή:
"01000C91039628ΒΑΔΓΖΕ000021493A283D0795C3F33C88FE06CDCB6E32885EC6D341EDF27C1E3E97E72E", και υποδεικνύουμε το τέλος της εισαγωγής παραμέτρων με το συνδυασμό των πλήκτρων "<Ctrl+z>".

Η απάντηση που λαμβάνεται στην περίπτωση μιας επιτυχημένης αποστολής μηνύματος έχει τη μορφή:
"+CMGS: <Αριθμός_αναφοράς_του_μηνύματος>" και συνοδεύεται από ένα "OK".

Στην εικόνα που ακολουθεί βλέπουμε το αποτέλεσμα που είχε η εκτέλεση της εντολής "AT+CMGS", όπου η συσκευή κινητού τηλεφώνου αναθέτει στο μήνυμα που εστάλη τον αριθμό αναφοράς "0" και μας ειδοποιεί ότι η αποστολή πραγματοποιήθηκε με επιτυχία.



Εικόνα 29: Το αποτέλεσμα που προκύπτει από την εκτέλεση της εντολής "AT+CMGS"

2.2.4 Η χρησιμοποίηση του PDUspy

Στην προηγούμενη υπό-ενότητα είδαμε με ποιο τρόπο μπορούμε να δημιουργήσουμε σε ένα PC ένα μήνυμα SMS, υπολογίζοντας τις τιμές των πεδίων ενός πλαισίου TPDU. Το οποίο μπορούμε στη συνέχεια να το προωθήσουμε εισάγοντας τις κατάλληλες εντολές "AT" σε ένα GSM modem που είναι συνδεδεμένο με το PC.

Έτσι είδαμε, ότι η παραπάνω διαδικασία δεν είναι δύσκολη, όμως το κομμάτι όπου υπολογίζονται τα πεδία του TPDU δεν είναι και το πιο ευχάριστο, ενώ μπορεί αρκετά εύκολα να συμβεί κάποιο λάθος που θα επηρεάσει τη μετάδοση ενός μηνύματος.

Ευτυχώς όμως υπάρχει μια εναλλακτική λύση, που προσφέρει μιαν ελκυστική διεπαφή, με την οποία μπορούμε να στείλουμε ένα SMS μέσω ενός GMS modem αποφεύγοντας το μεγαλύτερο μέρος του φόρτου που θα είχαμε αν ακολουθούσαμε την τακτική της προηγούμενης υπό-ενότητας.

Η λύση έρχεται με το πρόγραμμα PDUspy, που χρησιμοποιείται σε περιβάλλον των Microsoft Windows και λαμβάνεται ελεύθερα από τον ακόλουθο διαδικτυακό σύνδεσμο: http://www.nobbi.com/download/pduspy.zip.

Το PDUspy συνδέεται με το GSM modem και αναλαμβάνει, μετά από την εισαγωγή των κατάλληλων στοιχείων, να δομήσει το πλαίσιο ενός μηνύματος SMS και να το προωθήσει μέσω του modem προς τον προορισμό του.

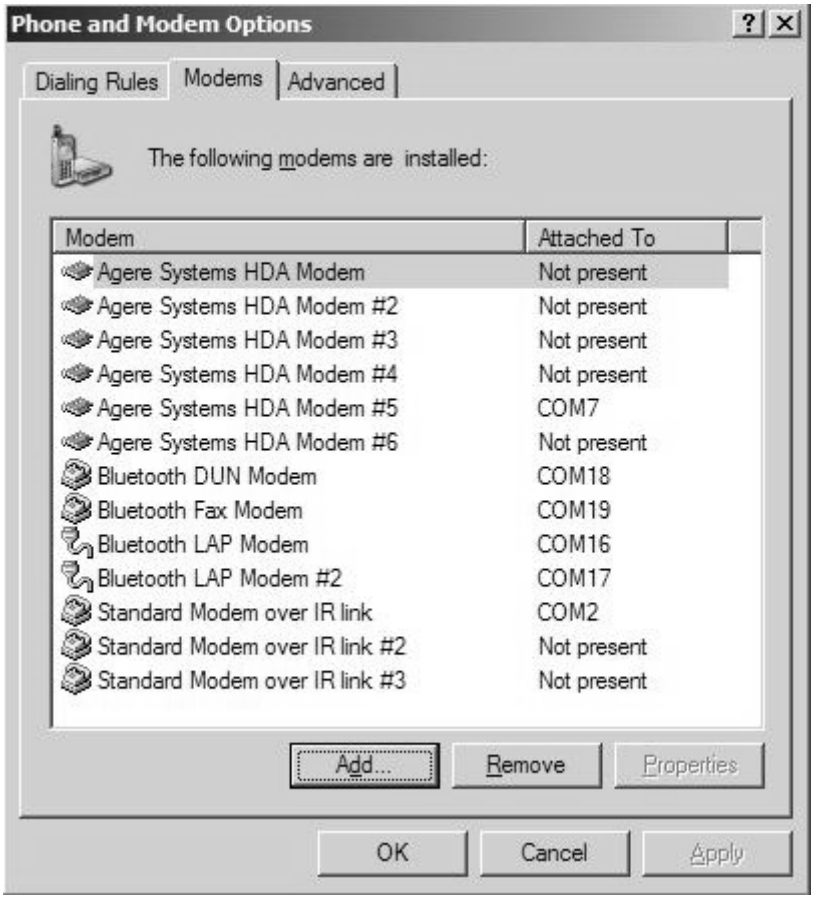
Η διαδικασία με την οποία ρυθμίζεται το PDUspy είναι άμεση και αρκετά απλή, αφού το πρόγραμμα αποτελείται από ένα εκτελέσιμο αρχείο και δεν απαιτεί κάποια σύνθετη εγκατάσταση.

Ο μόνος περιορισμός που εντοπίζεται στη λειτουργία του προγράμματος είναι το port στο οποίο συνδέεται το GSM modem ή το κινητό τηλέφωνο που χρησιμοποιείται, αφού το PDUspy χρειάζεται μία από τις θύρες COM2 μέχρι και COM16.

Μπορούμε να μάθουμε ποια θύρα εκχωρείται στη συσκευή του GSM modem / κινητού τηλεφώνου πηγαίνοντας στο "`Control Panel`", και από εκεί επιλέγουμε την κατηγορία "`Phone and Modem Options`". Στη συγκεκριμένη κατηγορία επιλέγουμε το tab "`Modems`", όπου εμφανίζονται όσα modems σχετίζονται με το PC μας.

Στην περίπτωση που το modem που χρησιμοποιούμε δεν συνδέεται σε μία από τις θύρες COM2 μέχρι και COM16, τότε μπορούμε να αλλάξουμε τις ιδιότητες που έχει και να το ρυθμίσουμε όπως μας εξυπηρετεί.

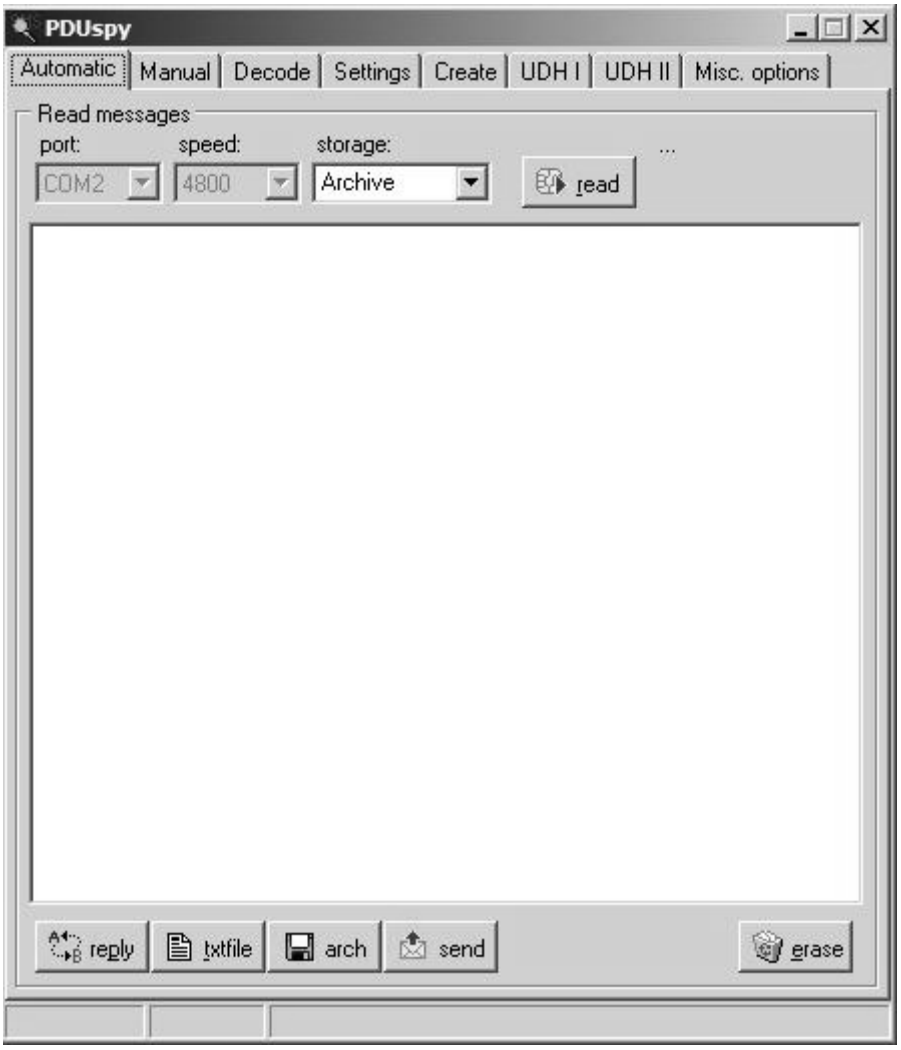
Στην εικόνα που ακολουθεί εμφανίζεται το tab "`Modems`", στο οποίο εντοπίζουμε το modem που αντιστοιχεί στη δικιά μας περίπτωση. Όπως φαίνεται, χρησιμοποιείται το modem που λειτουργεί πάνω από ζεύξη υπερύθρων και του έχει συνδεθεί στη θύρα COM2.



Εικόνα 30: Η εμφάνιση της θύρας COM που χρησιμοποιείται από το GSM modem

Αφού σιγουρευτούμε ότι το GSM modem συνδέεται σε μία από τις κατάλληλε θύρες τότε περνάμε στην εκτέλεση του PDUspy.

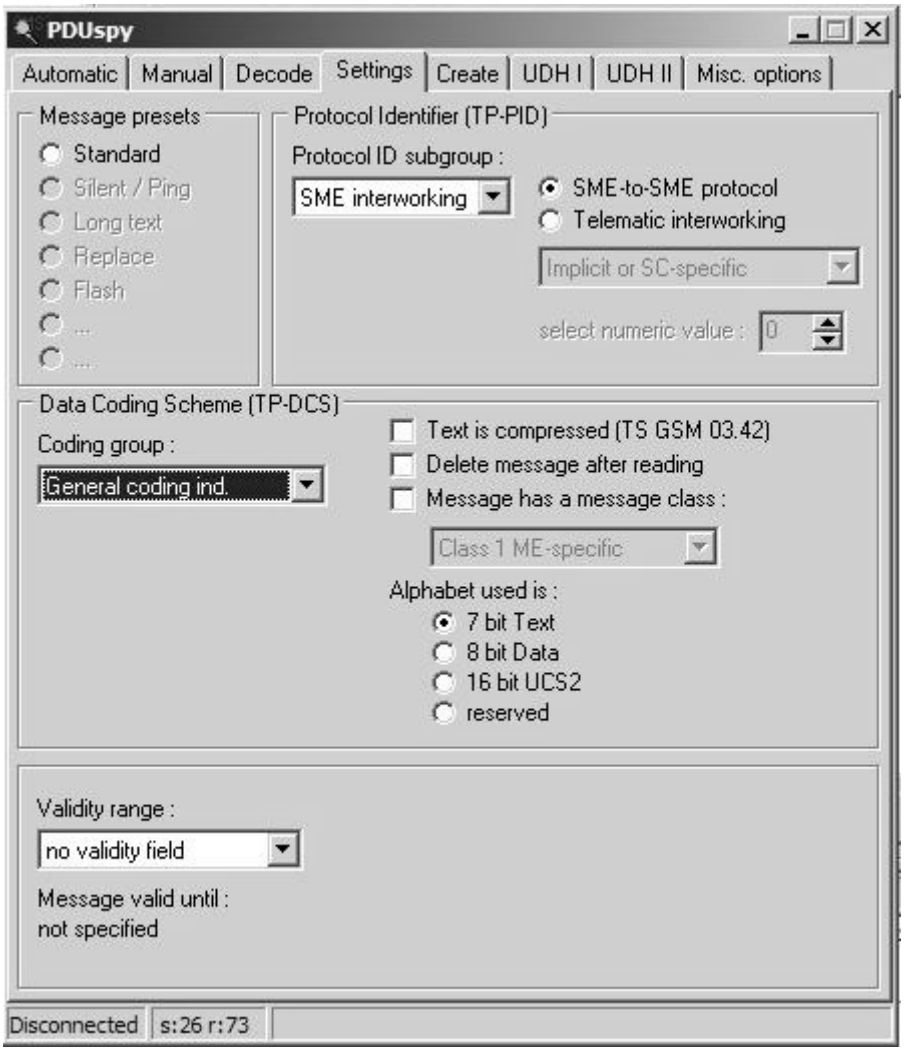
Η εικόνα που ακολουθεί είναι πρώτη συναντάμε όταν ξεκινά η εκτέλεση του PDUspy. Στην οποία βρισκόμαστε στο tab "`Automatic`", όπου δίνεται η δυνατότητα για ανάγνωση της κάρτας SIM και εύρεση μηνυμάτων που ενδέχεται να είναι αποθηκευμένα στο χρησιμοποιούμενο GSM modem ή κινητό τηλέφωνο.



Εικόνα 31: Η πρώτη οθόνη που συναντάμε κατά την εκτέλεση του προγράμματος PDUspy

Από το tab "`Automatic`" περνάμε στο tab "`Settings`" όπου, όπως απεικονίζεται και στην εικόνα που ακολουθεί, παρέχονται επιλογές για τα πεδία `Protocol Identifier`, `Data Coding Scheme` και `Validity Period Range`.

Αφήνουμε τις επιλογές ως έχουν, αφού επιθυμούμε να στείλουμε ένα απλό μήνυμα κειμένου.



Εικόνα 32: Το tab με τις ρυθμίσεις του προγράμματος PDUspy

Στη συνέχεια, επιλέγουμε το tab "`Create`", στο οποίο μπορούμε να συμπληρώσουμε τις απαιτούμενες τιμές για τα υπόλοιπα πεδία του πλαισίου ενός μηνύματος SMS.

Στο πεδίο με το `Destination Address` εισάγουμε τον αριθμό του παραλήπτη. Από το drop-down menu με τον τύπο του αριθμού, όπως φαίνεται και στην εικόνα που ακολουθεί, αλλάζουμε από την επιλογή "`Unknown`" σε "`International`".

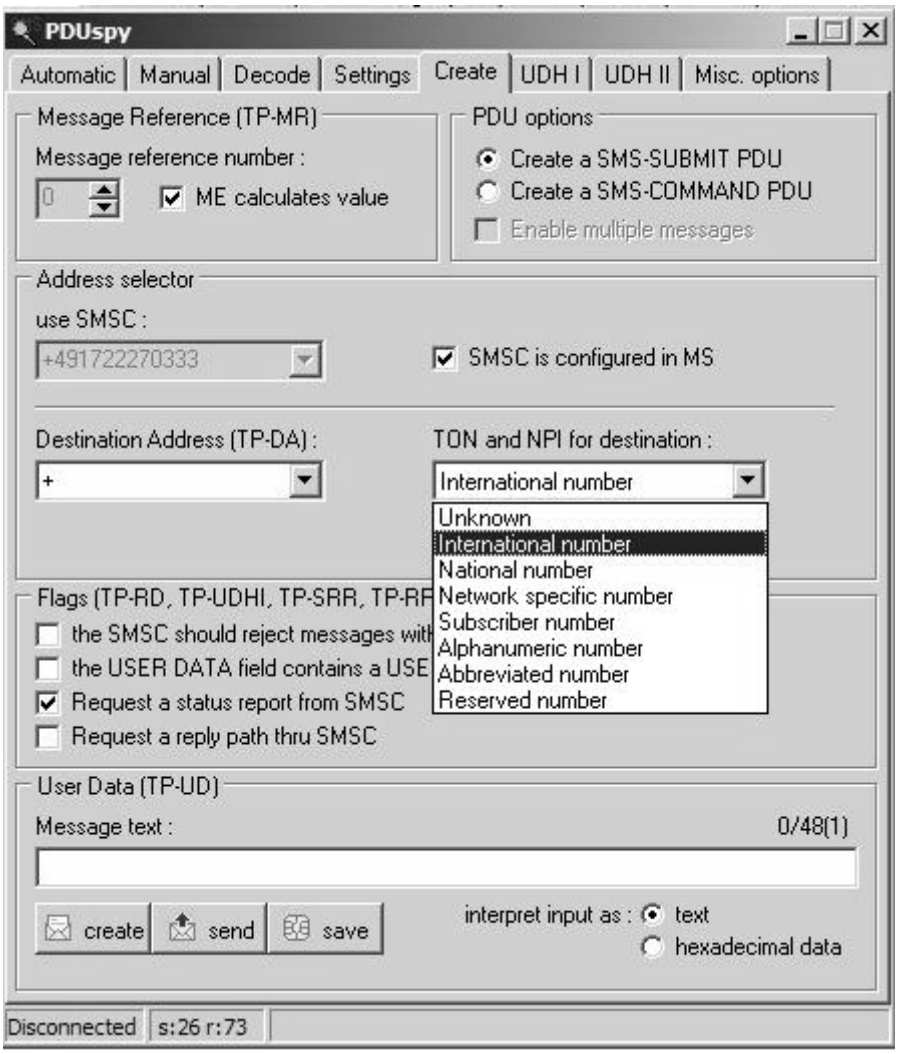
Έτσι, ο αριθμός του παραλήπτη αποτελείται από το χαρακτήρα "+", ακολουθούμενο από το διεθνή κωδικό κλήσης της χώρας του παραλήπτη και τον τηλεφωνικό αριθμό του παραλήπτη.

Στο πεδίο `User Data` εισάγουμε το μήνυμα:
"`It is easy to send text messages`".

Στα υπόλοιπα πεδία δεν χρειάζεται να κάνουμε κάποια αλλαγή, οπότε περνάμε στις επιλογές που έχουμε στη διάθεση μας.

Όπως φαίνεται και από την εικόνα που ακολουθεί, μετά από τη σύνταξη του μηνύματος μπορούμε να πιέσουμε ένα από τα κουμπιά:

- "`create`", με το οποίο δημιουργείται το πλαίσιο, το οποίο στη συνέχεια μπορούμε να προωθήσουμε από το χειροκίνητα πηγαίνοντας στο tab "`Manual`".
- "`send`", μέσω του οποίου το μήνυμα προωθείται στο GSM modem και από εκεί αποστέλλεται στην τηλεφωνική συσκευή του παραλήπτη.
- "`save`", για την αποθήκευση του μηνύματος στην κάρτα SIM του συνδεδεμένου GSM modem ή κινητού τηλεφώνου.

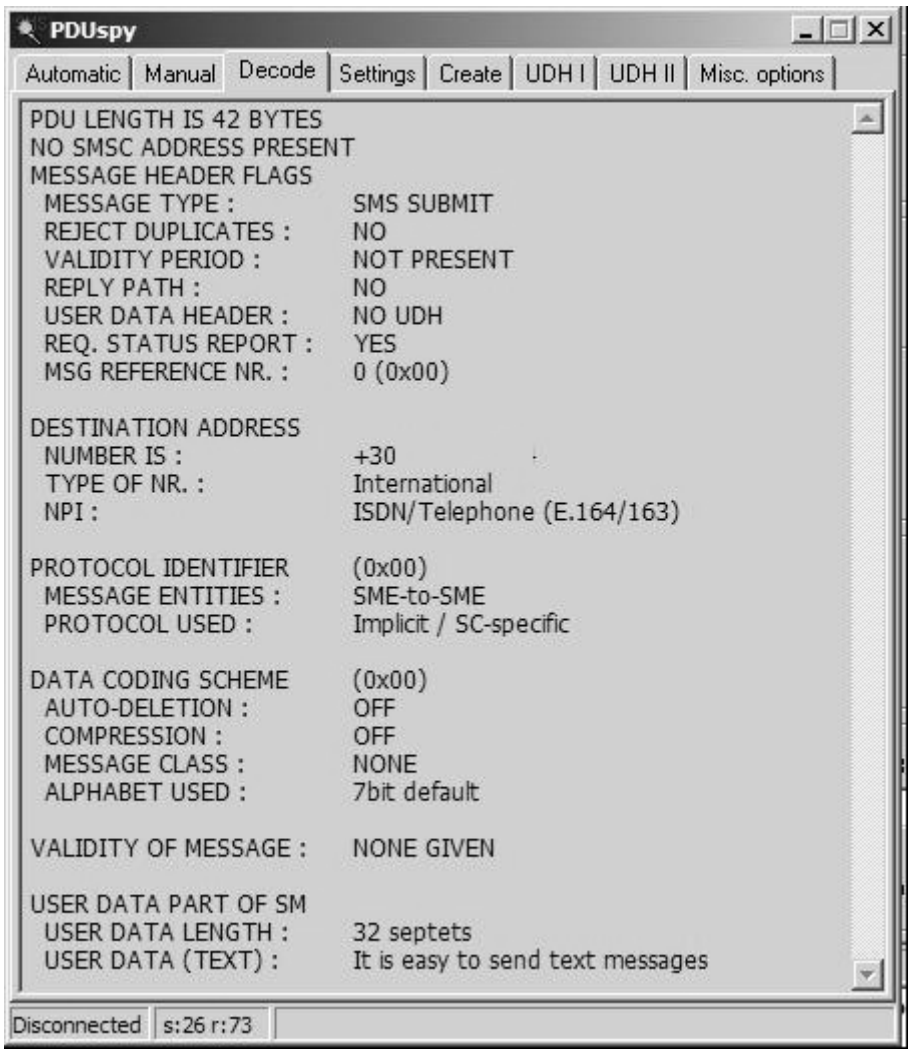


Εικόνα 33: Το tab "Create" και η αλλαγή του τύπου του αριθμού του παραλήπτη

Για το παράδειγμα μας, θα πιέσουμε το πλήκτρο "`create`". Το πρόγραμμα μας μεταφέρει αυτόματα στο tab "`Manual`", όμως εμείς θα επιλέξουμε το γειτονικό tab με τον τίτλο "`Decode`", το οποίο παρουσιάζει αρκετό ενδιαφέρον.

Στο tab "`Decode`" πραγματοποιείται η αποκωδικοποίηση του πλαισίου που δημιουργήθηκε και παρουσιάζονται αναλυτικά και με κάθε λεπτομέρεια όλες οι πληροφορίες που σχετίζονται με αυτό. Με αυτό τον τρόπο μπορεί να παρουσιαστεί η δομή του πλαισίου που δημιουργήθηκε μαζί με τις αντίστοιχες τιμές που εισήχθησαν στα πεδία.

Στην εικόνα που ακολουθεί παρουσιάζεται το αποτέλεσμα που παράγεται στο tab "Decode" από το πλαίσιο του μηνύματος που δημιουργήσαμε νωρίτερα.



Εικόνα 34: Το αποτέλεσμα που εμφανίζεται στο tab "Decode"

Κεφάλαιο 3 - Η ασφάλεια των δικτύων κινητής τηλεφωνίας

Σε αυτό το κεφάλαιο παρουσιάζονται τα μέτρα που λαμβάνονται για την προστασία των συνδρομητών κινητής τηλεφωνίας, ενώ παράλληλα επισημαίνονται και οι πηγές κινδύνου που υπάρχουν σε αυτά τα δίκτυα.

3.1 Δίκτυα πρώτης γενιάς

Τα δίκτυα που περιλαμβάνονταν στην πρώτη γενιά κινητής τηλεφωνίας ήταν πλήρως αναλογικά. Το γεγονός αυτό τα έκανε αρκετά ευάλωτα, αφού καθιστούσε αδύνατη την εφαρμογή κάποιου κρυπτογραφικού μηχανισμού που θα προστατεύσει τους συνδρομητές.

Η ασύρματη μετάδοση κάθε συνομιλίας διεξαγόταν ανοιχτά, χωρίς κάποιο είδος κρυπτογράφησης. Έτσι δινόταν η δυνατότητα παρακολούθησης των επικοινωνιών σε όποιον χρησιμοποιούσε ένα δέκτη υψηλών συχνοτήτων, αφού με το συντονισμό του δέκτη στη κατάλληλη συχνότητα γινόταν η συνακρόαση των συνομιλιών μιας κυψέλης.

Ένα άλλο σημαντικό πρόβλημα που υπήρχε ήταν η αδυναμία προφύλαξης της ταυτότητας των συνδρομητών, γεγονός που οφείλεται κατά ένα μέρος και στις συσκευές που ήταν συμβατές με τα δίκτυα πρώτης γενιάς.

Οι περισσότερες συσκευές ήταν ευάλωτες σε επιθέσεις αντίστροφης μηχανικής(reverse engineering) και έδιναν σε έναν κακόβουλο χρήστη τη δυνατότητα να τις επαναπρογραμματίσει.

Μέσω του επαναπρογραμματισμού μπορούσε να δημιουργηθεί ο κλώνος μιας συσκευής που ανήκει σε κάποιον νόμιμο συνδρομητή και μέσα από τη χρήση μιας συσκευής αυτού του τύπου δινόταν σε έναν κακόβουλο χρήστη η δυνατότητα να εμφανίζεται σαν νόμιμος συνδρομητής και να πραγματοποιεί τις κλήσεις που επιθυμεί, χρεώνοντας όμως τις δικές του κλήσεις στο λογαριασμό του νόμιμου συνδρομητή.

3.2 Δίκτυα δεύτερης γενιάς

Η εμπειρία από τα δίκτυα πρώτης γενιάς έδειξε ξεκάθαρα ότι έπρεπε να ληφθεί μια σειρά από μέτρα για την προστασία των συνδρομητών.

Το γεγονός ότι τα δίκτυα της δεύτερης γενιάς κινητής τηλεφωνίας ήταν ψηφιακά διευκόλυνε την εφαρμογή μιας σειράς από κρυπτογραφικούς μηχανισμούς, μέσω των οποίων μπορούσαν να εξαλειφθούν τα φαινόμενα που παρατηρήθηκαν στην προηγούμενη γενιά κινητής τηλεφωνίας.

Σε αυτό το σημείο θα εξετάσουμε την περίπτωση του δικτύου GSM, και θα δούμε αναλυτικότερα τη διαχείριση των θεμάτων ασφαλείας και τους κινδύνους που υπάρχουν για τους συνδρομητές.

3.2.1 Το μοντέλο ασφάλειας του δικτύου GSM

Οι προδιαγραφές που σχετίζονται με την ασφάλεια του δικτύου GSM αναφέρουν χαρακτηριστικά ότι, «ένα δίκτυο GSM πρέπει να είναι όσο ασφαλές είναι και ένα δίκτυο PSTN». Με άλλα λόγια το δίκτυο GSM πρέπει να παρέχει ένα ελεγχόμενο περιβάλλον παρόμοιο με αυτό που παρέχεται από το δίκτυο σταθερής τηλεφωνίας.

Ο κορμός του δικτύου PSTN θεωρείται ένα ελεγχόμενο περιβάλλον, καθώς το δίκτυο ελέγχεται πλήρως από τις εταιρείες παροχής τηλεφωνίας. Η ασφάλεια του δικτύου σταθερής τηλεφωνίας βασίζεται στον περιορισμό της φυσικής πρόσβασης ενός συνδρομητή στον κορμό του δικτύου.

Η συγκεκριμένη φιλοσοφία μεταφέρεται και στο δίκτυο GSM, όπου δημιουργείται ένα ελεγχόμενο περιβάλλον μέσω του περιορισμού της φυσικής πρόσβασης των συνδρομητών στο τμήμα του δικτύου που βρίσκεται μετά από κάθε σταθμό βάσης.

Όμως το τμήμα της ασύρματης διασύνδεσης, δηλαδή από τη συσκευή κινητού τηλεφώνου προς το σταθμό βάσης, παρέχει φυσική πρόσβαση σε όλους τους συνδρομητές και θεωρείται μη ελεγχόμενο.

Γι’ αυτό το λόγο οι μηχανισμοί ασφάλειας του δικτύου GSM εστιάζουν στο τμήμα ασύρματης διασύνδεσης και διευθετούν τα ακόλουθα θέματα:

- Τον έλεγχο αυθεντικότητας ενός συνδρομητή
- Τη διατήρηση της ανωνυμίας ενός συνδρομητή
- Τη διαφύλαξη της εμπιστευτικότητας των επικοινωνιών

3.2.2 Αυθεντικοποίηση στο GSM

Ο έλεγχος αυθεντικότητας ή αυθεντικοποίηση(Authentication) του δικτύου GSM είναι μια απολύτως αναγκαία διαδικασία που έχει ως στόχο την πιστοποίηση της ταυτότητας ενός συνδρομητή του δικτύου.

Ο εν λόγω έλεγχος πραγματοποιείται κάθε φορά που ενεργοποιείται μια συσκευή κινητού τηλεφώνου και επιχειρεί να αποκτήσει πρόσβαση στο δίκτυο. Με αυτό τον τρόπο αποκλείεται η είσοδος στο δίκτυο σε μη εξουσιοδοτημένους χρήστες, αποτρέποντας έτσι περιπτώσεις εξαπάτησης, όπου κάποιος «μεταμφιέζεται» σε νόμιμο συνδρομητή.

Η συγκεκριμένη διαδικασία πραγματοποιείται με την αποστολή ενός μηνύματος «πρόκλησης» από το δίκτυο προς τον συνδρομητή και τη λήψη μιας προβλεπόμενης απάντησης από τη μεριά του συνδρομητή.

Ο ρόλος της κάρτας SIM

Πριν περάσουμε στην παρουσίαση της διαδικασίας αυθεντικοποίησης πρέπει να αναφέρουμε ότι η συγκεκριμένη διαδικασία, και γενικότερα η ασφάλεια του δικτύου GSM, βασίζεται σε πολύ μεγάλο βαθμό στην παρουσία της κάρτας SIM.

Στην κάρτα SIM εκτελούνται αλγόριθμοι που σχετίζονται με την ασφάλεια του συστήματος, όπως είναι οι αλγόριθμοι A3 και A8, ενώ παράλληλα αποθηκεύονται εκεί πολύ σημαντικά στοιχεία, όπως τα IMSI και K_i, που είναι απαραίτητα για τη λειτουργία των συγκεκριμένων αλγορίθμων.

Ο αριθμός IMSI

Ο αριθμός International Mobile Subscriber Identity(IMSI) είναι μια ακολουθία 15 ψηφίων που χωρίζεται σε τρία πεδία:

- το Mobile Country Code(MCC), που έχει μήκος τριών ψηφίων και αντιπροσωπεύει την χώρα στην οποία εκδόθηκε η κάρτα SIM
- το Mobile Network Code(MNC), που έχει μήκος δύο ψηφίων και αντιπροσωπεύει το δίκτυο στο οποίο ανήκει η κάρτα SIM
- το Mobile Subscriber Identification Number(MSIN), που καταλαμβάνει τα δέκα τελευταία ψηφία και αντιπροσωπεύει τον αριθμό ταυτότητας που είναι μοναδικός για κάθε συνδρομητή

Η αντιστοιχία ένα-προς-ένα μεταξύ του IMSI και του συνδρομητή καθιστά τον αριθμό ιδανικό για να ξεχωρίζει ένας συνδρομητής.

Γι’ αυτό και ο IMSI χρησιμοποιείται από το δίκτυο για τη δρομολόγηση των κλήσεων προς τον κινητό σταθμό του συνδρομητή, ενώ κατά τη διαδικασία του ελέγχου αυθεντικότητας ο συνδρομητής υποβάλλει το IMSI του στο δίκτυο, για να αποδείξει με αυτό τον τρόπο ότι είναι πράγματι αυτός που ισχυρίζεται ότι είναι.

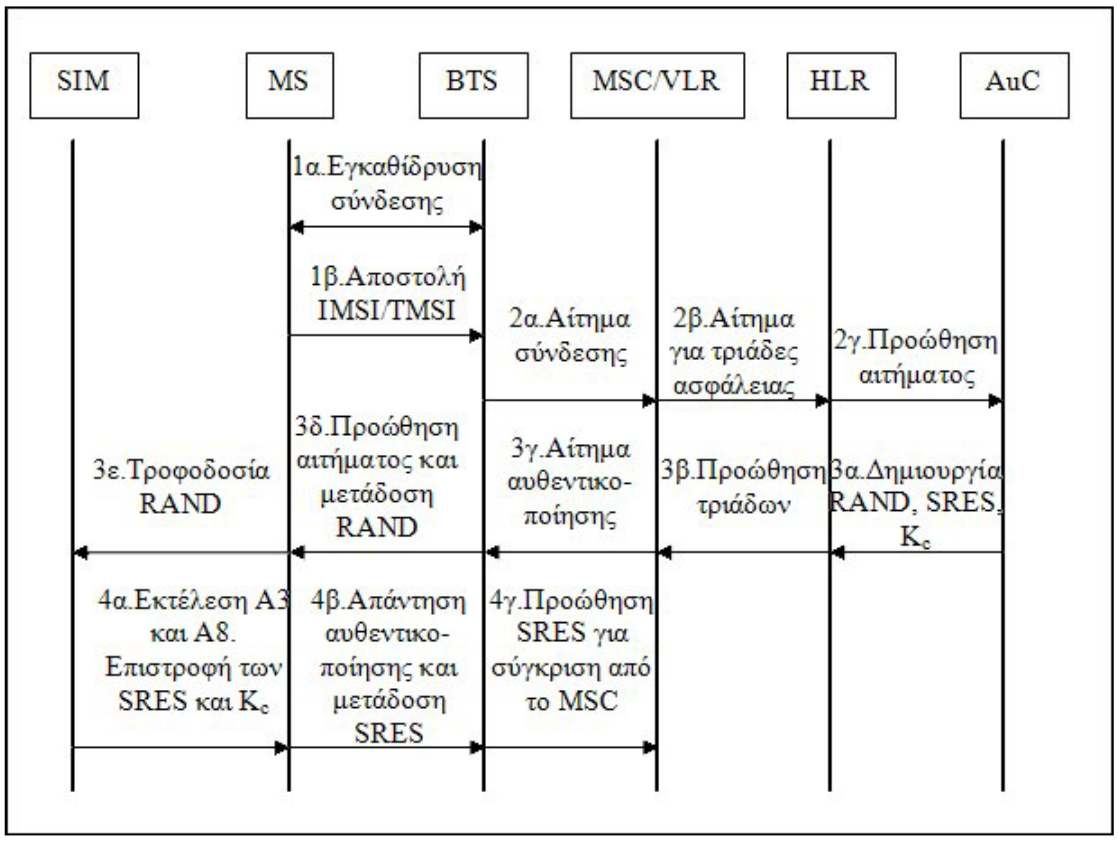
Το κλειδί K_i

Το K_i είναι το βασικό κλειδί που εισάγεται στους αλγόριθμους που εκτελούνται στην κάρτα SIM. Έχει μήκος 128 bit, είναι μοναδικό για κάθε συνδρομητή, ενώ η τιμή του διατηρείται μυστική και είναι γνωστή μόνο στην κάρτα SIM και το κέντρο AuC του δικτύου GSM.

Η συσκευή κινητής τηλεφωνίας δεν μαθαίνει ποτέ την τιμή του K_i, αλλά μόνο τροφοδοτεί την κάρτα SIM με πληροφορίες που απαιτούνται για την εκτέλεση αλγορίθμων.

Η διαδικασία της αυθεντικοποίησης του συνδρομητή

Σε αυτό το σημείο, έχοντας σαν επιπρόσθετη βοήθεια το σχήμα που ακολουθεί, θα παρουσιάσουμε βήμα προς βήμα τον τρόπο με τον οποίο γίνεται η διαδικασία της αυθεντικοποίησης ενός συνδρομητή που επιχειρεί να συνδεθεί στο δίκτυο.



Εικόνα 35: Η διαδικασία αυθεντικοποίησης ενός συνδρομητή του δικτύου GSM

1. Ο κινητός σταθμός(MS) ενεργοποιείται και συνδέεται με τον πομποδέκτη σταθμό βάσης(BTS) της κυψέλης του.

   Η διαδικασία της αυθεντικοποίησης ξεκινά με την υποβολή της ταυτότητας του συνδρομητή στο δίκτυο.

   Σε αυτή τη φάση δεν υπάρχει κάποιο κρυπτογραφημένο κανάλι επικοινωνίας, για αυτό προτιμάται η αποστολή κάποιας προσωρινής ταυτότητας TMSI που πιθανότατα υπάρχει αποθηκευμένη από προηγούμενο έλεγχο αυθεντικότητας.

   Σε διαφορετική περίπτωση αποστέλλεται η μόνιμη ταυτότητα IMSI, αν και γενικότερα αποφεύγεται η έκθεση της μόνιμης ταυτότητας και η μη κρυπτογραφημένη μετάδοση της, καθώς με την αποκάλυψη της μόνιμης ταυτότητας τίθεται σε κίνδυνο η ανωνυμία του συνδρομητή.

2. Το BTS στέλνει στο αρμόδιο κέντρο MSC ένα αίτημα για να επιτραπεί η σύνδεση στο συνδρομητή.

Αν έχει σταλεί κάποιο TMSI τότε, με τη βοήθεια του VLR, γίνεται αντιστοίχηση με το πραγματικό IMSI.

Στη συνέχεια, το MSC ζητά από το HLR να του χορηγηθεί ένα σύνολο από πέντε τριάδες ασφάλειας(security triplets) που προορίζονται για το συγκεκριμένο IMSI.

Από την πλευρά του, το HLR προωθεί το αίτημα για τη δημιουργία των απαιτούμενων τριάδων στο AuC, καθώς αυτό είναι υπεύθυνο για όλες τις λειτουργίες που σχετίζονται με την ασφάλεια.

3. Το AuC βρίσκει το κλειδί K_i που αντιστοιχεί στο IMSI.

   Να θυμίσουμε ότι το K_i είναι γνωστό μόνο στην SIM του συνδρομητή και στο AuC, γι' αυτό και θα χρησιμοποιηθεί σαν πειστήριο της ταυτότητας του συνδρομητή.

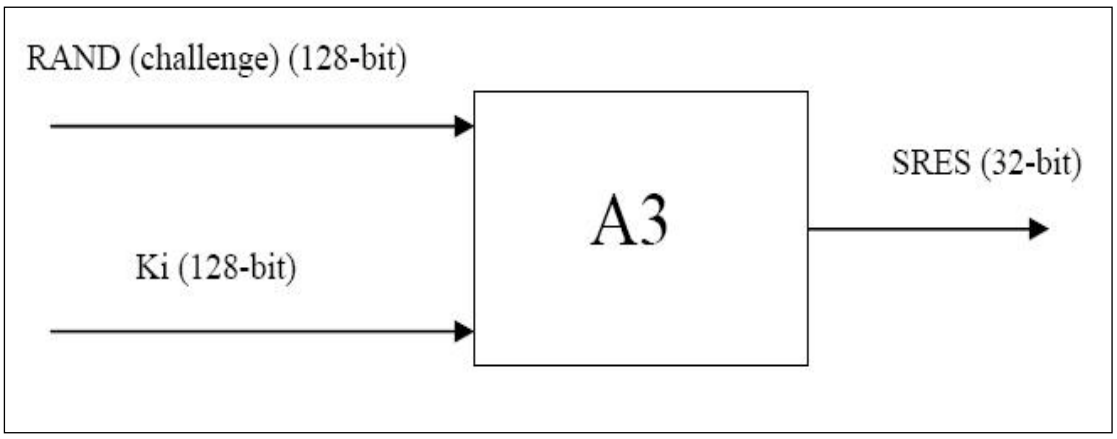
   Ο καταρτισμός των πέντε τριάδων ασφάλειας ξεκινά με τη δημιουργία πέντε αριθμών RAND.

   Ο αριθμός RAND είναι το πρώτο στοιχείο μιας τριάδας ασφάλειας. Πρόκειται για έναν τυχαίο αριθμό μήκους 128 bit που εισάγεται μαζί με το κλειδί K_i στους αλγόριθμους με ονόματα αναφοράς A3 και A8, από την εκτέλεση των οποίων δημιουργούνται τα δύο εναπομείναντα στοιχεία μιας τριάδας ασφάλειας.

   Για ένα δεδομένο συνδυασμό τιμών RAND και K_i δημιουργούνται στοιχεία που είναι μοναδικά. Έτσι, έχοντας το ίδιο K_i και πέντε διαφορετικούς αριθμούς RAND δημιουργούμε πέντε διαφορετικές τριάδες.

   Από την εκτέλεση του A3 προκύπτει το δεύτερο στοιχείο της τριάδας. Είναι ένας αριθμός μήκους 32 δυαδικών ψηφίων, που λέγεται SRES, και μέσω αυτού γίνεται η επιβεβαίωση της ταυτότητας του συνδρομητή.

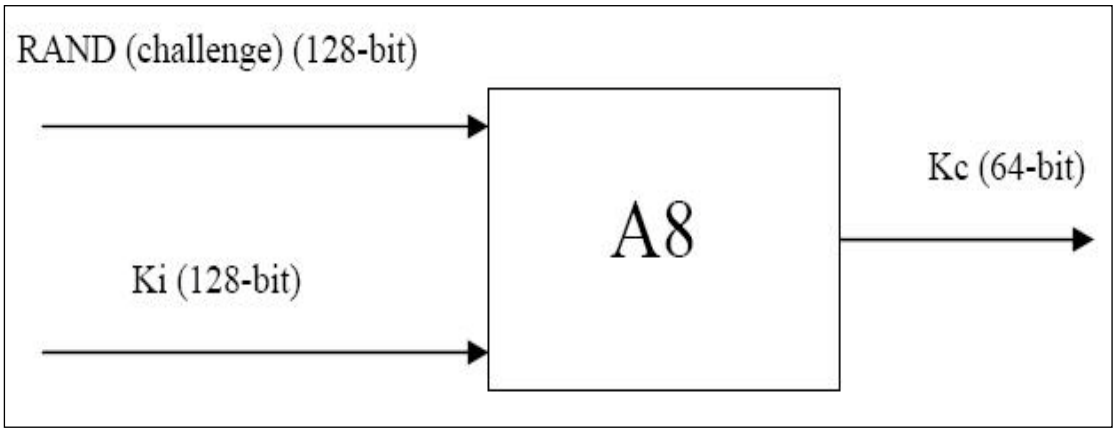
   Η κάρτα SIM του συνδρομητή καλείται και αυτή να υπολογίσει την τιμή του SRES, σύμφωνα με το RAND που λαμβάνει από το δίκτυο και το K_i που υπάρχει στη SIM.



Εικόνα 36: Η εκτέλεση του αλγορίθμου A3

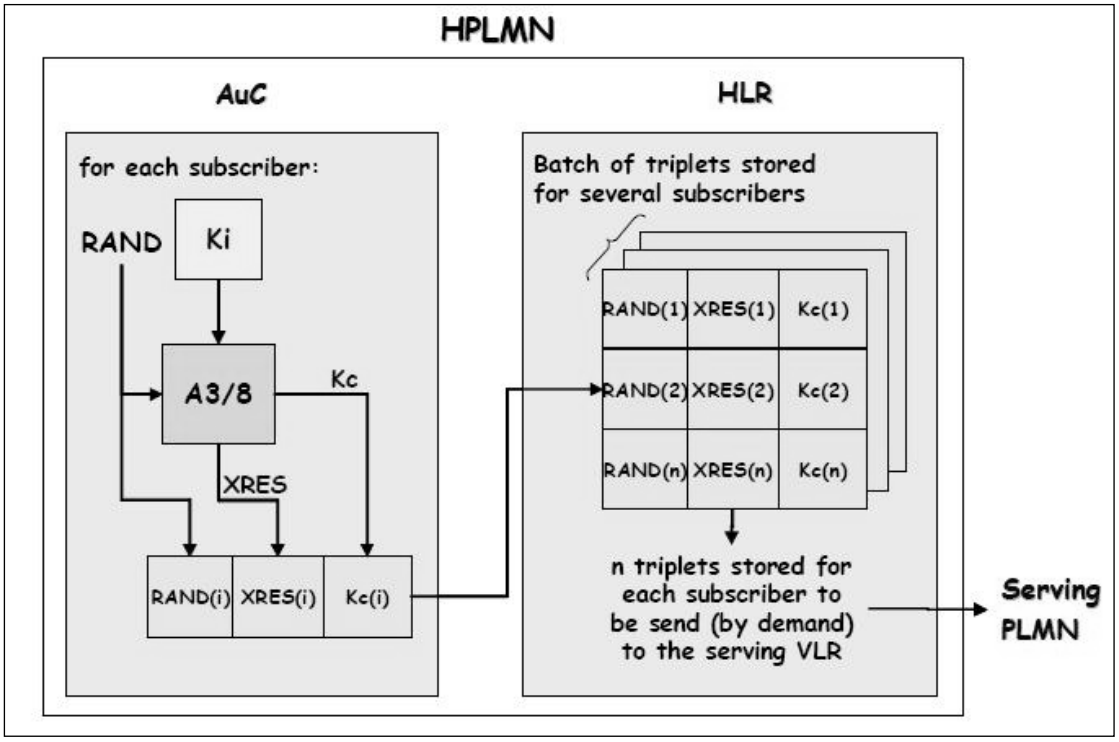
Το τελευταίο στοιχείο της τριάδας δημιουργείται από την εκτέλεση του αλγόριθμου A8. Είναι το κλειδί κρυπτογράφησης K_c, που έχει μήκος 64 bit και συμβάλλει τα μέγιστα για τη διατήρηση της εμπιστευτικότητας των επικοινωνιών.

Το K_c χρησιμοποιείται για την κρυπτογράφηση των επικοινωνιών ανάμεσα στον κινητό σταθμό του συνδρομητή και το υποσύστημα σταθμού βάσης.



Εικόνα 37: Η εκτέλεση του αλγορίθμου A8

Μετά την ολοκλήρωση της δημιουργίας των πέντε τριάδων(με τα RAND, SRES και K_c) το AuC τις προωθεί στο HLR και από εκεί οι τριάδες πηγαίνουν προς το αρμόδιο MSC.



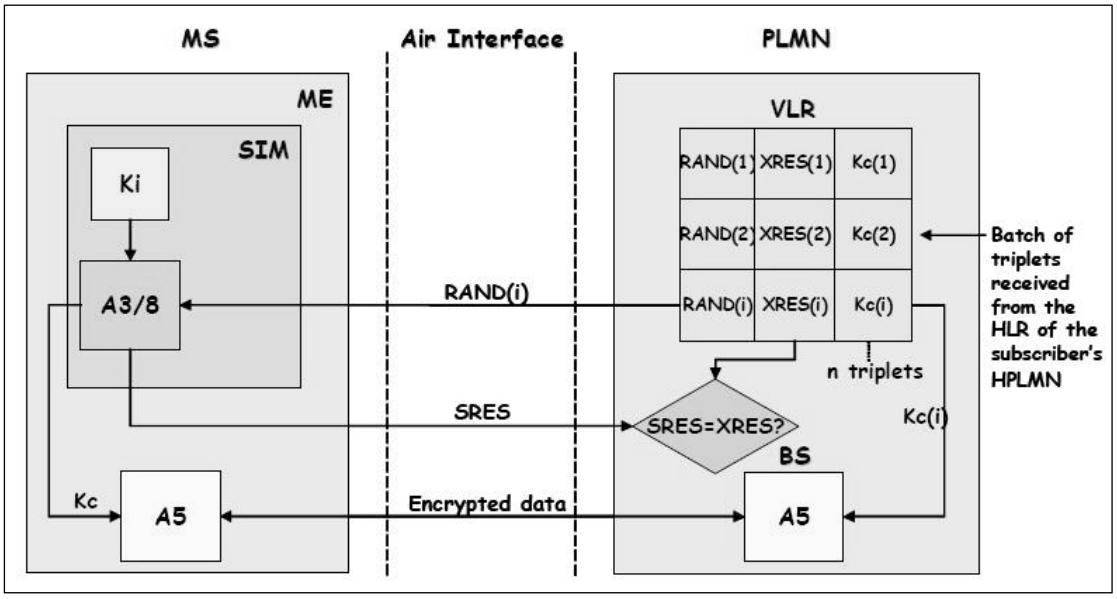
Εικόνα 38: Η δημιουργία των τριάδων ασφάλειας του GSM στο περιβάλλον του οικείου δικτύου ενός συνδρομητή και η προώθηση τους στο δίκτυο εξυπηρέτησης

Το MSC επιλέγει μία από τις διαθέσιμες τριάδες, ενώ οι υπόλοιπες τέσσερις αποθηκεύονται για κάποια μελλοντική διαδικασία αυθεντικοποίησης του ίδιου συνδρομητή.

Το MSC προωθεί το αίτημα, ή αλλιώς την «πρόκληση», για αυθεντικοποίηση του συνδρομητή. Το συγκεκριμένο αίτημα προωθείται μέσω του BTS στη συσκευή του συνδρομητή και ακολούθως, η τιμή του RAND που περιέχεται στο αίτημα, προωθείται στην κάρτα SIM.

Όλα αυτά παρουσιάζονται συνοπτικά στο σχήμα που ακολουθεί, όπου από το MSC/VLR επιλέγεται μια από τις τριάδες ασφάλειας και η τιμή RAND προωθείται προς τον κινητό σταθμό του συνδρομητή.

Ακόμη, παρουσιάζεται το πέρασμα της τιμής RAND στην κάρτα SIM και η εκτέλεση των αλγορίθμων ασφάλειας που βρίσκονται στο περιβάλλον της κάρτας SIM και του κινητού τηλεφώνου.

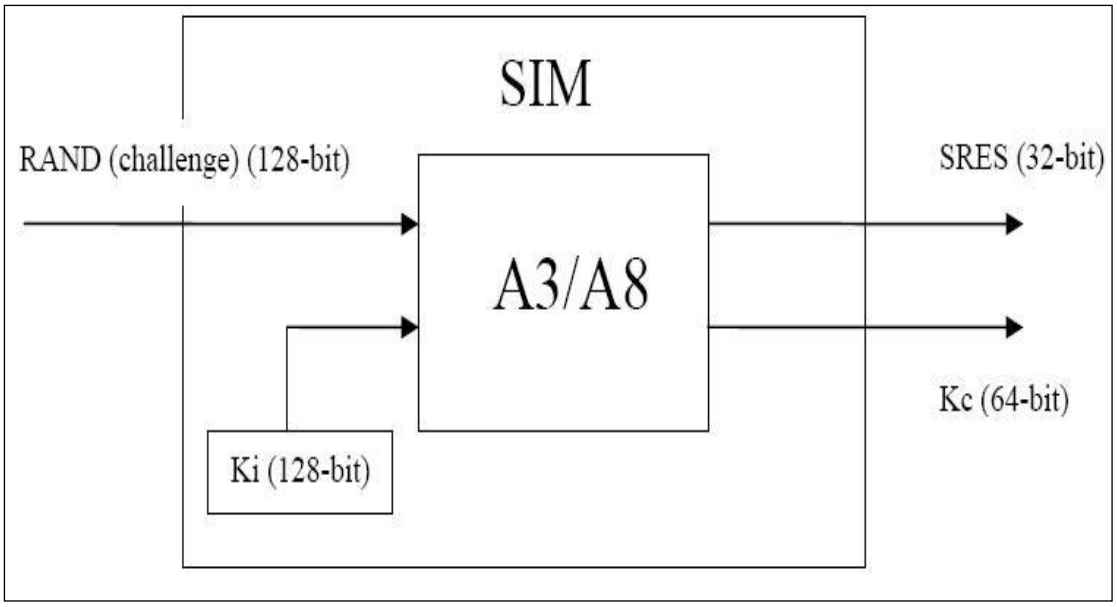


Εικόνα 39: Επιλογή μιας από τις τριάδες ασφάλειας που βρίσκονται στο MSC/VLR και προώθηση της τιμής RAND προς τον κινητό σταθμό του συνδρομητή.

4. Στο περιβάλλον της κάρτας SIM εκτελούνται οι αλγόριθμοι A3 και A8, στους οποίους εισάγονται τα απαραίτητα στοιχεία, δηλαδή το K_i που βρίσκεται ενσωματωμένο στην κάρτα SIM και η τιμή του RAND που ήρθε από το δίκτυο.

   Μετά την εκτέλεση των A3 και A8 δημιουργούνται τα δύο στοιχεία που απομένουν, τα SRES και K_c, από την τριάδα που αντιστοιχεί στο RAND.

   Το SRES που δημιουργήθηκε αποστέλλεται στο MSC σαν απάντηση για την «πρόκληση» που έχει θέσει το δίκτυο.



Εικόνα 40: Η εκτέλεση των αλγορίθμων A3 και A8 στο περιβάλλον της κάρτας SIM

Το MSC λαμβάνει το SRES που δημιουργήθηκε στον κινητό σταθμό του συνδρομητή και το συγκρίνει με το SRES που προέρχεται από το AuC.

Το MSC επιτρέπει την πρόσβαση στο δίκτυο μόνο αν τα δύο SRES είναι ίσα, διότι μόνο τότε επιβεβαιώνεται ότι ο συνδρομητής είναι πράγματι αυτός που ισχυρίζεται ότι είναι, αφού σε περίπτωση που ήταν άλλος συνδρομητής, είχαμε δηλαδή διαφορετικό K_i από αυτό του AuC, τότε οι τιμές των SRES δεν θα ήταν ίσες, οπότε το MSC θα απαγόρευε την πρόσβαση στο δίκτυο.

Μετά από έναν επιτυχημένο έλεγχο αυθεντικότητας ενός συνδρομητή πιστοποιείται εμμέσως ότι και το κλειδί K_c που απομένει από την τριάδα ασφάλειας είναι αληθινό.

Οπότε ο συνδρομητής και το δίκτυο θα μοιράζονται το συγκεκριμένο K_c, με το οποίο θα κρυπτογραφούν και θα αποκρυπτογραφούν τη μεταξύ τους επικοινωνία.

3.2.3 Η ανωνυμία στο GSM

Η διατήρηση της ανωνυμίας ενός συνδρομητή του δικτύου GSM επιτυγχάνεται όταν δεν έχουμε αρκετά συχνή έκθεση της μόνιμης ταυτότητας IMSI.

Όπως είδαμε νωρίτερα, κατά τη διάρκεια του ελέγχου αυθεντικότητας του συνδρομητή, είναι πιθανή η μετάδοση του IMSI πάνω από το τμήμα ασύρματης διασύνδεσης χωρίς να υπάρχει κρυπτογράφηση.

Σε αυτή την περίπτωση υπάρχει το ενδεχόμενο της υποκλοπής της τιμής του IMSI από κάποιον κακόβουλο χρήστη που παρακολουθεί το μέσο μετάδοσης.

Το IMSI, εκτός από το ρόλο ταυτότητας, χρησιμοποιείται και για τη δρομολόγηση κλήσεων από και προς το συνδρομητή.

Αυτό το γεγονός δίνει στον κακόβουλο χρήστη τη δυνατότητα να παρακολουθεί τις κινήσεις του συνδρομητή και να γνωρίζει πότε γίνεται κάποια κλήση από ή προς αυτόν, ενώ παράλληλα είναι γνωστή και η ευρύτερη περιοχή στην οποία βρίσκεται ο συνδρομητής.

Η λύση στο πρόβλημα έρχεται με τη χρήση των προσωρινών ταυτοτήτων TMSI(Temporary Mobile Subscriber Identity). Κάθε μια από τις προσωρινές ταυτότητες έχει μήκος 32 δυαδικών ψηφίων και εκδίδεται από το VLR κατά τη διάρκεια του ελέγχου αυθεντικότητας ενός συνδρομητή.

Για την αποφυγή υποκλοπών από κάποιον τρίτο συνηθίζεται η κρυπτογραφημένη μετάδοση της ταυτότητας, με την κρυπτογράφηση να γίνεται χρησιμοποιώντας το κλειδί K_c που κοινό ανάμεσα στο δίκτυο και τον συνδρομητή.

Έτσι, η αντιστοιχία IMSI-TMSI είναι γνωστή μόνο στο δίκτυο και στη συσκευή του συνδρομητή, γεγονός που επιτρέπει τη δρομολόγηση κλήσεων και μηνυμάτων μέσω του TMSI.

Μετά από μια επιτυχημένη διαδικασία ελέγχου αυθεντικότητας του συνδρομητή το δίκτυο αποστέλλει μια νέα κρυπτογραφημένη ταυτότητας TMSI.

Σε αντίθεση με το IMSI, που είναι μοναδικό για κάθε συνδρομητή στον κόσμο, το TMSI έχει τοπική ισχύ και μπορεί να αλλάζει τιμές, αφού χρησιμοποιείται αποκλειστικά στις κυψέλες που ανήκουν στην περιοχή ευθύνης του MSC/VLR που βρίσκεται ο συνδρομητής.

Σε κάθε περιοχή ευθύνης ενός MSC/VLR αντιστοιχεί ένα μοναδικό χαρακτηριστικό ταυτότητας που ονομάζεται Local Area Identity(LAI) και έχει βοηθητικό ρόλο όταν χρειάζεται να ανανεωθεί η προσωρινή ταυτότητα ενός συνδρομητή.

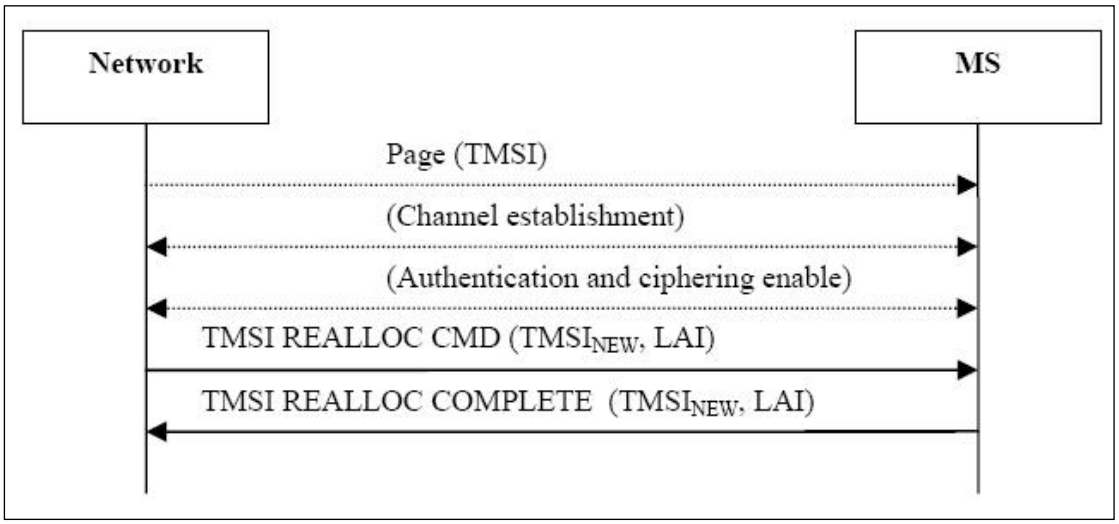
Η προσωρινή ταυτότητα του συνδρομητή ενδέχεται να αλλάξει σύμφωνα με δύο σενάρια, όταν:

- Η πλευρά του δικτύου κρίνει ότι είναι αναγκαία η ανανέωση του TMSI
- Ο συνδρομητής μετακινηθεί και περάσει στην περιοχή ευθύνης ενός νέου MSC και κατ' επέκταση ζητήσει μια νέα ταυτότητα από το νέο VLR

Στο σχήμα που ακολουθεί παρουσιάζεται πως πραγματοποιείται η ανανέωση του TMSI, όταν από την πλευρά του δικτύου θεωρηθεί ότι είναι απαραίτητη η συγκεκριμένη ενέργεια.

Όπως βλέπουμε και από το σχήμα, η διαδικασία είναι άμεση, αφού θεωρούμε ότι έχει πραγματοποιηθεί η αυθεντικοποίηση του συνδρομητή και η κρυπτογράφηση είναι ενεργοποιημένη, έτσι η νέα προσωρινή ταυτότητα μεταδίδεται κρυπτογραφημένη στον κινητό σταθμό του συνδρομητή μαζί με την ταυτότητα της περιοχής στην οποία βρίσκεται.

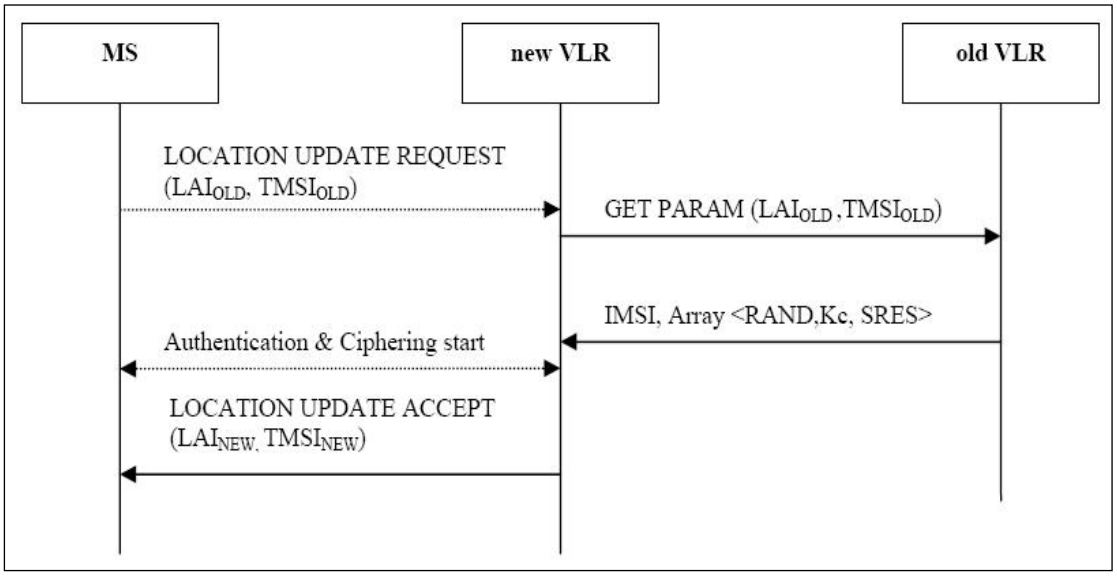
Από την πλευρά του συνδρομητή εκπέμπεται ένα σήμα που ενημερώνει το δίκτυο ότι η ανανέωση του TMSI ολοκληρώθηκε.



Εικόνα 41: Η ανανέωση της ταυτότητας TMSI όταν δεν υπάρχει μετακίνηση του συνδρομητή

Στην περίπτωση που ο συνδρομητής περάσει στην ευθύνη ενός νέου MSC/VLR, τότε έχουμε μια περισσότερο πολύπλοκη διαδικασία ανανέωσης της προσωρινής ταυτότητας.

Στο σχήμα που ακολουθεί παρατηρούμε τι συμβαίνει σε αυτή την περίπτωση ανανέωσης του TMSI.



Εικόνα 42: Η ανανέωση της ταυτότητας TMSI όταν ο συνδρομητής μετακινείται σε νέα περιοχή

Αρχικά, ο κινητός σταθμός του συνδρομητή στέλνει στο VLR της νέας περιοχής ένα αίτημα για ανανέωση της προσωρινής του ταυτότητας και μαζί με αυτό υποβάλλει στο δίκτυο επιπλέον στοιχεία, όπως είναι το TMSI που χρησιμοποιεί και το LAI της περιοχής στην οποία βρισκόταν.

Στη συνέχεια, το VLR της νέας περιοχής προωθεί το TMSI που του υποβλήθηκε στο VLR που βρίσκεται στην περιοχή απ' όπου προήλθε ο συνδρομητής, ενώ παράλληλα, ζητάει να μάθει τη συσχέτιση που υπάρχει ανάμεσα σε IMSI-TMSI και να λάβει τα διανύσματα αυθεντικοποίησης που ενδέχεται να έχει στην κατοχή του το VLR της παλαιότερης περιοχής.

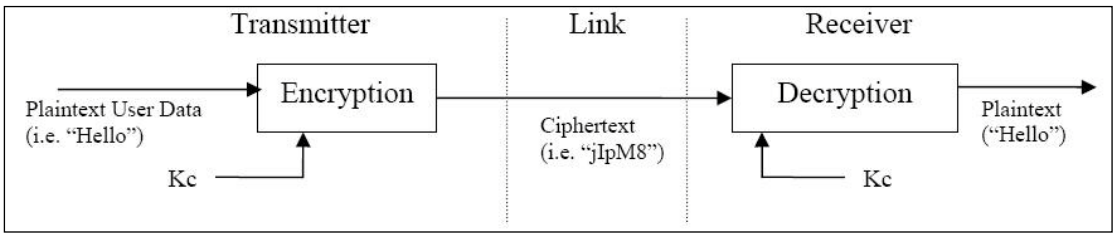
Το παλιό VLR προωθεί προς το νέο VLR την ταυτότητα IMSI που αντιστοιχεί μαζί με τις διαθέσιμες τριάδες αυθεντικοποίησης.

Τώρα το νέο VLR γνωρίζει όλα τα απαραίτητα στοιχεία για να ξεκινήσει μια νέα διαδικασία αυθεντικοποίησης, και αργότερα, ανάλογα με την έκβαση της αυθεντικοποίησης, να μεταδώσει κρυπτογραφημένη τη νέα προσωρινή ταυτότητα TMSI που αντιστοιχεί στο συνδρομητή.

3.2.4 Η εμπιστευτικότητα των επικοινωνιών στο GSM

Η εμπιστευτικότητα στο GSM διασφαλίζεται με την κρυπτογράφηση των επικοινωνιών στο τμήμα της ασύρματης διασύνδεσης του δικτύου.

Η κρυπτογράφηση γίνεται μέσω της πράξης XOR πάνω σε δύο ροές(streams) μήκους 114 bit, οι οποίες αντιπροσωπεύουν τη μεταδιδόμενη πληροφορία και ένα ψευδό-τυχαίο κλειδί, και από εκεί προκύπτει η κρυπτογραφημένη ροή, που έχει και αυτή μήκος 114 δυαδικών ψηφίων.



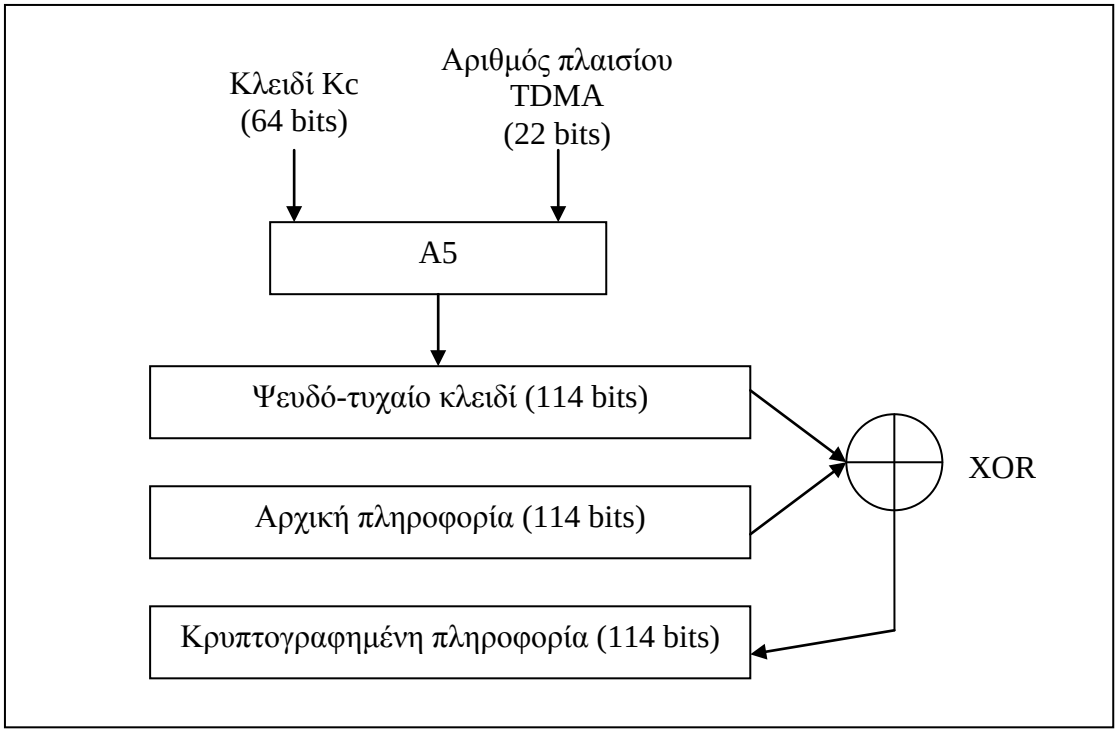
Εικόνα 43: Η κρυπτογράφηση των επικοινωνιών στο GSM

Η ροή των ψευδό-τυχαίων κλειδιών γεννάται από τον αλγόριθμο A5, ο οποίος παρέχεται με τη μορφή υλικού και βρίσκεται ενσωματωμένος στον κινητό σταθμό.

Για την εκτέλεση του A5 απαιτείται η είσοδος δύο στοιχείων. Πρόκειται για:

- το κλειδί K_c, που παράγεται από τον αλγόριθμο A8 και δημιουργείται κατά τη διάρκεια του ελέγχου αυθεντικότητας του συνδρομητή
- το αριθμό ακολουθίας που έχει το πλαίσιο TDMA που χρησιμοποιείται εκείνη τη στιγμή

Στο σχήμα που ακολουθεί παρουσιάζεται συνοπτικά η διαδικασία κρυπτογράφησης των ροών πληροφορίας που μεταδίδονται στο GSM.



Εικόνα 44: Η διαδικασία κρυπτογράφησης στο GSM

3.2.5 Τρωτά σημεία στην ασφάλεια του GSM

Τα ακόλουθα είναι μερικά από τα τρωτά σημεία που εντοπίζονται στην ασφάλεια του δικτύου GSM:

- Οι υλοποιήσεις των αλγορίθμων A3 και A8 περιέχουν ορισμένα ψεγάδια, που μπορούν να οδηγήσουν σε προσδιορισμό του K_i που βρίσκεται στην κάρτα SIM
- Ο έλεγχος αυθεντικότητας πραγματοποιείται μόνο από την πλευρά του συνδρομητή προς το δίκτυο, έτσι μπορούν να δημιουργηθούν ψεύτικα στοιχεία του δικτύου
- Ο αλγόριθμος A5 έχει ελαττώματα με τα οποία δίνεται η δυνατότητα αποκρυπτογράφησης των επικοινωνιών

Οι υλοποιήσεις των αλγορίθμων A3 και A8

Λίγοι άνθρωποι στον κόσμο γνωρίζουν πραγματικά πως υλοποιούνται οι αλγόριθμοι A3 και A8. Αυτό οφείλεται στο γεγονός ότι οι αλγόριθμοι σχεδιάστηκαν με άκρα μυστικότητα.

Οι δημιουργοί των A3 και A8 θεώρησαν ότι θα παρείχαν ασφαλέστερους αλγόριθμους αν διατηρούσαν κρυφές τις λεπτομέρειες που αφορούν το σχεδιασμό τους(Security by Obscurity). Με αυτό τον τρόπο όμως απέκρυψαν και τα ελαττώματα και τις ατέλειες σχεδιασμού.

Οι περισσότερες εταιρείες παροχής κινητής τηλεφωνίας χρησιμοποιούν την COMP128, μια υλοποίηση που συνδυάζει την εκτέλεση των A3 και A8 και δημιουργεί μια σειρά από 128 bit, από την οποία εξάγονται τα SRES και K_c. Όμως αποδείχθηκε ότι η συγκεκριμένη υλοποίηση έχει σημαντικότατες ατέλειες.

Μια από αυτές τις ατέλειες βρίσκεται στη δημιουργία του κλειδιού K_c, το οποίο έχει μήκος 64 bit. Η COMP128 δημιουργεί αδύναμα κλειδιά, αφού σε κάθε εκτέλεση της θέτει την τιμή 0 στα 10 τελευταία bit, με αποτέλεσμα το κλειδί που δημιουργείται να έχει την ισχύ των 54 bit.

Ένα άλλο σημαντικό ελάττωμα της COMP128 εντοπίζεται όταν εισάγονται σε αυτήν επιλεγμένες τιμές RAND. Όπως απέδειξαν οι Wagner και Goldberg[*], με τη δοκιμή 160000 επιλεγμένων τιμών RAND σε μια κάρτα SIM λαμβάνονται αρκετές πληροφορίες, με τις οποίες είναι δυνατόν να προσδιοριστεί η τιμή του K_i και να δημιουργηθούν κάρτες-κλώνοι.

Η συγκεκριμένη τεχνική βελτιώθηκε σημαντικά από τον Dejan Kaljevic, ο οποίος εκτιμά ότι με το δικό του τρόπο απαιτούνται κατά μέσο όρο 18000 δοκιμές RAND μέχρι την πλήρη αποκάλυψη του K_i.

[*] Περισσότερα στοιχεία, μαζί με μια υλοποίηση των A3 και A8, βρίσκονται στο URL www.scard.org/gsm/a3a8.txt

Η εύρεση του K_i πραγματοποιείται ταχύτερα στην περίπτωση όπου ο κακόβουλος χρήστης έχει φυσική πρόσβαση στην κάρτα SIM. Βεβαίως, για να γίνει αυτό, πρέπει ο κακόβουλος χρήστης να βρει ένα τρόπο και να αποσπάσει τη συσκευή κινητού τηλεφώνου ενός νόμιμου συνδρομητή.

Όμως, όπως θα δούμε και παρακάτω, υπάρχει μια μέθοδος με την οποία δεν απαιτείται φυσική πρόσβαση στην κάρτα SIM, καθώς ο επιτιθέμενος δοκιμάζει επιλεγμένες τιμές RAND από απόσταση, χωρίς να γίνεται αντιληπτός από το συνδρομητή.

Ο αλγόριθμος A5

Ο αλγόριθμος A5 κυκλοφορεί στις ακόλουθες δύο εκδόσεις:

- Την έκδοση A5/1, που είναι η ισχυρότερη και είναι διαθέσιμη στις χώρες που είναι μέλη του CEPT
- Την έκδοση A5/2, που προορίζεται για τις χώρες όπου εξάγεται το GSM και είναι αρκετά ασθενέστερη, καθώς έχει ισχύ 16 δυαδικών ψηφίων

Οι Biryukov και Shamir έχουν δημοσιεύσει μια τεχνική[*] με την οποία μπορούν να προσδιορίσουν το κλειδί K_c που χρησιμοποιείται από τον A5/1.

Για την επίτευξη του στόχου απαιτείται η δημιουργία μιας τεράστιας βάσης δεδομένων, με μέγεθος που προσεγγίζει τα 300GB, η οποία περιλαμβάνει διάφορες τιμές που προκύπτουν από την εκτέλεση του A5 και υποακολουθίες των κλειδιών με τα οποία σχετίζονται.

Ο προσδιορισμός του K_c επιτυγχάνεται μετά από παρακολούθηση της κίνησης για ένα διάστημα 2 λεπτών, όπου τότε αναζητείται στη βάση κάποια ακολουθία δυαδικών ψηφίων που ταιριάζει με τις ροές που έρχονται από την παρακολουθούμενη κίνηση. Μετά από μια πετυχημένη αναζήτηση είναι ευκολότερος ο υπολογισμός του K_c και είναι εφικτή η αποκρυπτογράφηση μιας συνομιλίας.

Ο Wagner κατάφερε αργότερα να βελτιστοποιήσει την τεχνική των Biryukov και Shamir[*]. Έτσι, η παρακολούθηση της κίνησης για ένα διάστημα 2 δευτερολέπτων είναι αρκετή για να ξεκινήσει ο υπολογισμός του K_c.

Στις προαναφερθείσες τεχνικές έρχεται να προστεθεί και εκείνη που αναπτύχθηκε από τον Karsten Nohl[*] και αφορά, σε πρώτη φάση, μια κατανεμημένη παθητική επίθεση που εκμεταλλεύεται την παράλληλη αρχιτεκτονική που παρουσιάζουν οι κάρτες γραφικών της νέας γενιάς, και εν συνεχεία, με τη χρησιμοποίηση του κατάλληλου λογισμικού(OpenBTS και Asterisk) και την επιλογή ενός πομποδέκτη δημιουργεί ένα «σπιτικό» IMSI Catcher με κόστος που προσεγγίζει τα 1500 δολάρια.

[*] Η δημοσίευση βρίσκεται στην ιστοσελίδα http://cryptome.org/a5.ps
[*] Περισσότερες πληροφορίες στο URL http://cryptome.info/0001/a51-bsw/a51-bsw.htm
[*] Περισσότερα στην ιστοσελίδα http://www.h-online.com/open/news/item/26C3-GSM-hacking-made-easy-893245.html

Μονόπλευρος έλεγχος αυθεντικότητας

Σε προηγούμενη υποενότητα παρουσιάστηκε η διαδικασία με την οποία ελέγχεται η αυθεντικότητα των συνδρομητών που επιθυμούν την πρόσβαση στο δίκτυο GSM. Όπως φάνηκε από τη συγκεκριμένη διαδικασία, ελέγχεται μόνο η ταυτότητα των συνδρομητών και δεν απαιτείται κάτι ανάλογο από το δίκτυο.

Χωρίς την παροχή κάποιου μηχανισμού ασφαλείας, που θα εγγυάται στο συνδρομητή ότι συνδέθηκε με το σωστό δίκτυο, δίνεται σε κάποια κακόβουλη πηγή η δυνατότητα να «υποδυθεί» πραγματικές οντότητες του δικτύου.

Η δημιουργία ψεύτικων δικτυακών στοιχείων επιτρέπει σε ένα κακόβουλο χρήστη να πραγματοποιεί την ενεργή επίθεση του ενδιάμεσου(Man-in-the-middle attack).

Μια εκδοχή αυτής της επίθεσης περιλαμβάνει τη δημιουργία ενός ψεύτικου σταθμού βάσης, τον οποίο ο κακόβουλος χρήστης εγκαθιστά ανάμεσα στον κινητό σταθμό του συνδρομητή και έναν αληθινό σταθμό βάσης.

Όποιος κινητός σταθμός επιθυμεί να συνδεθεί στο δίκτυο τότε επιχειρεί τη σύνδεση με το σταθμό βάσης που βρίσκεται πιο κοντά και έχει ισχυρότερο σήμα. Στην προκειμένη περίπτωση ο ψεύτικος σταθμός βάσης ικανοποιεί αυτή τη συνθήκη, οπότε, με την εγκαθίδρυση αυτής της σύνδεσης, ο κακόβουλος χρήστης αποκτά μια πληθώρα επιλογών με τις οποίες είναι σε θέση να επηρεάζει τις επικοινωνίες του συνδρομητή.

Ειδικότερα αν γνωρίζει και τα ελαττώματα των A3 και A8, τότε ο κακόβουλος χρήστης είναι σε θέση:

- Να απενεργοποιήσει την κρυπτογράφηση και να παρακολουθεί τις κλήσεις ενός συνδρομητή
- Να αποσπάσει το IMSI
- Να δοκιμάσει από απόσταση επιλεγμένες τιμές RAND και να προσδιορίσει το K_i
- Να χρησιμοποιήσει τα IMSI και K_i για να δημιουργήσει τον κλώνο μιας νόμιμης κάρτας SIM
- Να συνδεθεί στο δίκτυο χρησιμοποιώντας τον κλώνο μιας SIM και ταυτόχρονα να εμποδίσει την πρόσβαση στον νόμιμο κάτοχο της κάρτας

Στην περίπτωση που ο κακόβουλος χρήστης ενδιαφέρεται για το περιεχόμενο των επικοινωνιών ενός συνδρομητή, τότε μπορεί να ξεκινήσει τη διαδικασία αυθεντικοποίησης, να στείλει ένα RAND, να αγνοήσει την τιμή του SRES που θα λάβει από τον κινητό σταθμό και να επιτρέψει την πραγματοποίηση κλήσεων απενεργοποιώντας την κρυπτογράφηση τους.

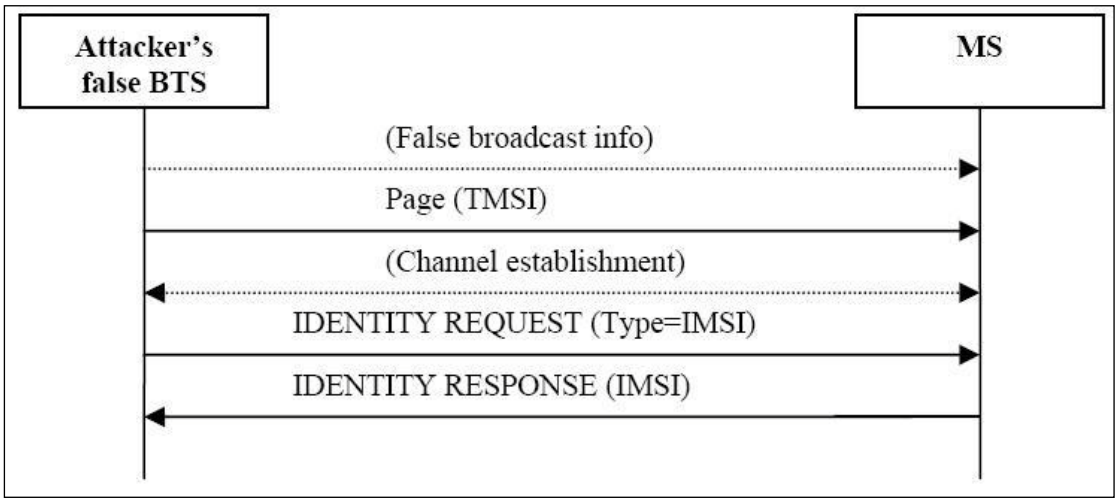
Ο συνδρομητής δεν είναι σε θέση να ελέγξει αν η κρυπτογράφηση είναι ενεργοποιημένη. Άλλωστε, υπάρχουν περιπτώσεις χωρών όπου βάσει νόμου απενεργοποιείται η κρυπτογράφηση. Με αυτό τον τρόπο ο κακόβουλος χρήστης, που βρίσκεται ανάμεσα στον συνδρομητή και το αληθινό δίκτυο, παρακολουθεί όλη τη μη κρυπτογραφημένη κίνηση από και προς τον συνδρομητή.

Αν ο κακόβουλος χρήστης επιθυμεί την κλωνοποίηση μια κάρτας SIM, τότε χρειάζεται να γνωρίζει τη μόνιμη ταυτότητα IMSI και το κλειδί K_i.

Ο ψεύτικος σταθμός βάσης μπορεί να λάβει το IMSI με δύο τρόπους.

Στην πρώτη περίπτωση ειδοποιεί τον κινητό σταθμό του συνδρομητή ότι μετακινήθηκε σε νέα κυψέλη ή ότι χάθηκε η αντιστοιχία IMSI-TMSI από το VLR. Έτσι υποχρεώνει τον κινητό σταθμό να υποβάλει την ταυτότητα IMSI.

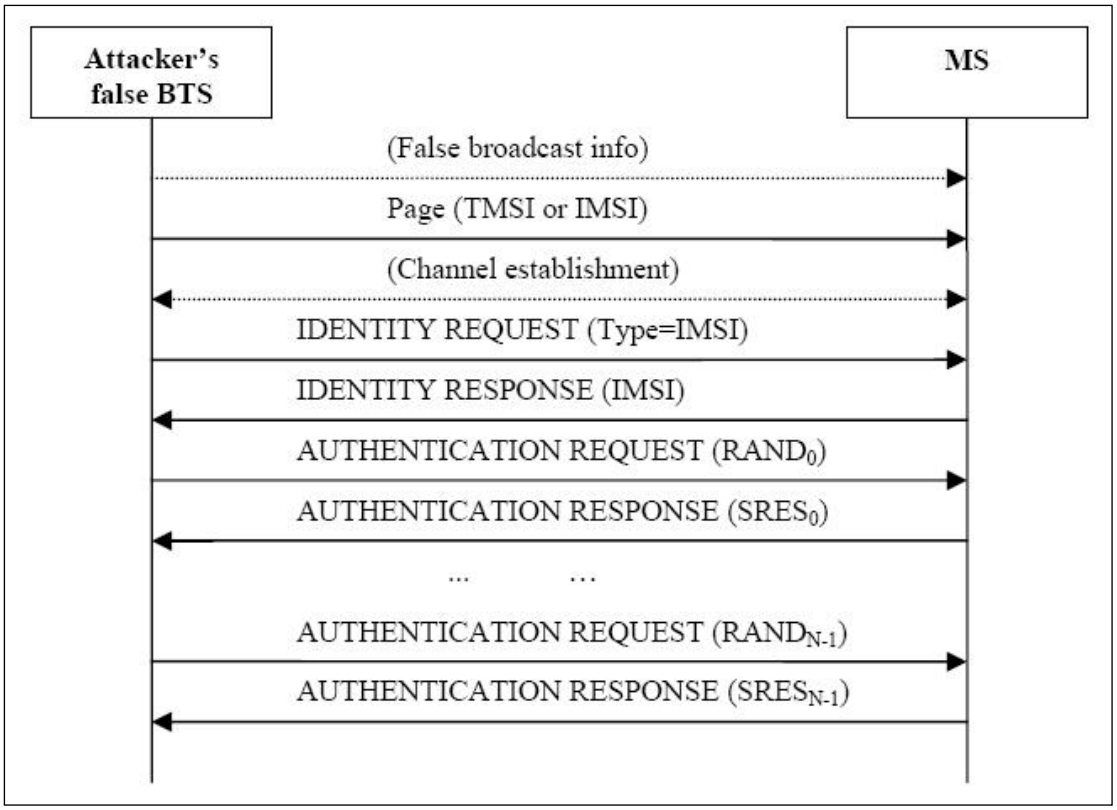
Στη δεύτερη περίπτωση ξεκινά τη διαδικασία αυθεντικοποίησης, όπου είναι πιθανό ο κινητός σταθμός να μην έχει αποθηκευμένο κάποιο TMSI, έτσι αναγκάζεται να αποστείλει το IMSI χωρίς να είναι κρυπτογραφημένο.



Εικόνα 45: Η λήψη του IMSI ενός συνδρομητή στέλνοντας ψευδείς πληροφορίες από ένα ψεύτικο σταθμό βάσης

Ο προσδιορισμός του κλειδιού K_i πραγματοποιείται μέσα από τη διαδικασία αυθεντικοποίησης. Ο ψεύτικος σταθμός βάσης στέλνει στον κινητό σταθμό τις επιλεγμένες τιμές RAND και δέχεται για ανάλυση τα SRES που δημιουργούνται από το κινητό τηλέφωνο.

Η δοκιμή χιλιάδων διαφορετικών τιμών RAND είναι μια ενεργοβόρα διαδικασία και υπάρχει πάντα το ενδεχόμενο να τελειώσει η μπαταρία της συσκευής του συνδρομητή. Γι' αυτό, η συγκεκριμένη επίθεση μπορεί να χωριστεί και να πραγματοποιηθεί τμηματικά(partitioning attack).



Εικόνα 46: Ο προσδιορισμός του Ki μέσα από τη χρήση ενός ψεύτικου σταθμού βάσης και την πραγματοποίηση διαδοχικών αυθεντικοποιήσεων

Με το IMSI και το K_i να είναι διαθέσιμα στον κακόβουλο χρήστη, είναι εύκολο γι' αυτόν να δημιουργήσει τον κλώνο μιας κάρτας SIM που ανήκει σε έναν νόμιμο συνδρομητή. Η επόμενη ενέργεια του επιτιθέμενου είναι να συνδεθεί στο δίκτυο.

Υπάρχει όμως το ενδεχόμενο να είναι ήδη συνδεδεμένος ο νόμιμος συνδρομητής. Τότε, ο κακόβουλος χρήστης χρησιμοποιεί τον ψεύτικο σταθμό βάσης για να στείλει ένα πολύ δυνατό σήμα στον κινητό σταθμό του συνδρομητή.

Με αυτό τον τρόπο αρνείται στο συνδρομητή την πρόσβαση στις υπηρεσίες του δικτύου(Denial of Service - DoS). Έτσι ο επιτιθέμενος έχει όλη την ευχέρεια να συνδεθεί στο δίκτυο και να επικοινωνεί ενώ υποδύεται τον πραγματικό συνδρομητή.

3.3 Δίκτυα τρίτης γενιάς

Η ανάπτυξη της τρίτης γενιάς των δικτύων κινητής τηλεφωνίας επηρεάστηκε σε μεγάλο βαθμό από τα δίκτυα που υπήρχαν νωρίτερα και αποτελούσαν τη δεύτερη γενιά.

Στον τομέα της ασφάλειας των δικτύων είχαμε τη διατήρηση της φιλοσοφίας που υπήρχε στη δεύτερη γενιά, με την παρουσία των απαραίτητων βελτιώσεων που στόχευαν στη διευθέτηση θεμάτων ασφαλείας που προϋπήρχαν.

Σε αυτή την υποενότητα θα δούμε αναλυτικότερα τη διαχείριση των θεμάτων ασφάλειας στο δίκτυο τρίτης γενιάς UMTS.

3.3.1 Η ασφάλεια του δικτύου UMTS

Το δίκτυο UMTS ήταν ο διάδοχος του αρκετά επιτυχημένου δικτύου δεύτερης γενιάς GSM. Έτσι ήταν απολύτως λογικό η ασφάλεια του δικτύου UMTS να διέπεται από παρόμοια φιλοσοφία με το GSM.

Από το μοντέλο ασφάλειας του GSM ξεχώριζαν τα ακόλουθα σημεία:

- Η χρήση της κάρτας SIM
- Η αυθεντικοποίηση του χρήστη
- Η χρήση προσωρινών ταυτοτήτων
- Η κρυπτογράφηση των επικοινωνιών στο τμήμα της ασύρματης διασύνδεσης

Όπως είδαμε στην προηγούμενη υποενότητα, υπάρχουν ορισμένες απειλές για την ασφάλεια του GSM. Πιο συγκεκριμένα:

- Το δίκτυο επιτρέπει τις ενεργές επιθέσεις, όπως γίνεται με τη χρησιμοποίηση ενός ψεύτικου σταθμού βάσης
- Δεν δημοσιεύτηκαν οι αλγόριθμοι που χρησιμοποιούνται για την αυθεντικοποίηση και την κρυπτογράφηση, έτσι κρατήθηκαν μυστικές και οι αδυναμίες τους
- Η ταυτότητα του συνδρομητή μεταδίδεται χωρίς κρυπτογράφηση πάνω από το τμήμα ασύρματης διασύνδεσης

Η ασφάλεια του UMTS αναπτύχθηκε έχοντας σαν στόχο την αντιμετώπιση αυτών των απειλών, άρα, τα σημεία που ξεχωρίζουν στο μοντέλο ασφάλειας του UMTS είναι:

- Η χρήση της κάρτας USIM
- Η αμοιβαία αυθεντικοποίηση του χρήστη και του δικτύου
- Η χρήση προσωρινών ταυτοτήτων
- Η κρυπτογράφηση του UTRAN
- Η διαφύλαξη της ακεραιότητας των σημάτων που μεταδίδονται στο UTRAN

Για την κρυπτογράφηση και τη διαφύλαξη της ακεραιότητας των δεδομένων των χρηστών του UMTS χρησιμοποιούνται νέοι και ισχυρότεροι αλγόριθμοι, όπου σε αντίθεση με την περίπτωση των μυστικών αλγορίθμων του GSM, είναι δημοσιευμένοι και βρίσκονται στη διάθεση της κρυπτογραφικής κοινότητας για αξιολόγηση.

3.3.2 Η αυθεντικοποίηση ανάμεσα στο χρήστη και το δίκτυο UMTS

Ο μηχανισμός αυθεντικοποίησης των χρηστών του δικτύου UMTS είναι γνωστός και ως Authentication and Key Agreement(AKA). Η αυθεντικοποίηση είναι αμοιβαία και πραγματοποιείται, όπως και στο GSM, με τη χρησιμοποίηση της τεχνικής «πρόκληση-και- απάντηση».

Στο μηχανισμό αυθεντικοποίησης του δικτύου UMTS εμπλέκονται τρεις οντότητες:

- Το AuC από το οικείο δίκτυο(Home Network – HN)
- Το VLR/SGSN από το δίκτυο εξυπηρέτησης(Serving Network – SN)
- Ο τερματικός εξοπλισμός που περιλαμβάνει την κάρτα USIM του συνδρομητή

Η κάρτα USIM, σε απόλυτη αντιστοιχία με την κάρτα SIM στο GSM, αποτελεί τον ακρογωνιαίο λίθο της ασφάλειας του δικτύου UMTS, αφού περιέχει στοιχεία που είναι μοναδικά για κάθε συνδρομητή και είναι γνωστά μόνο σε αυτήν και την οντότητα AuC, όπως είναι η μόνιμη ταυτότητα IMSI και το βασικό κλειδί K_i.

Το οικείο δίκτυο είναι το δίκτυο στο οποίο ανήκει ο συνδρομητής, οπότε, κατά την εκκίνηση της διαδικασίας αυθεντικοποίησης, η πρόκληση για τον συνδρομητή θα δημιουργηθεί από το AuC εκείνου του δικτύου.

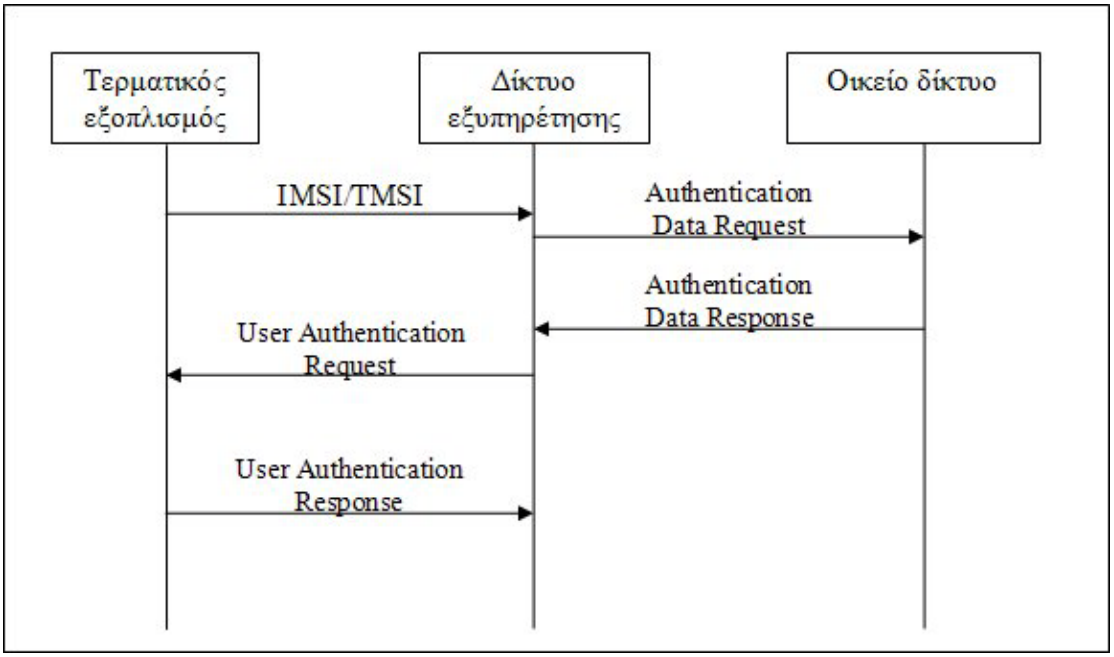
Το δίκτυο εξυπηρέτησης είναι το δίκτυο στο οποίο βρίσκεται ο περιπλανώμενος συνδρομητής τη στιγμή που πραγματοποιείται η αυθεντικοποίηση. Το VLR/SGSN εκείνου του δικτύου χρησιμοποιείται για την επικοινωνία με το οικείο δίκτυο.

Η βασική ιδέα είναι ότι το SN πρέπει να ελέγξει την ταυτότητα του συνδρομητή, ενώ την ίδια στιγμή ο συνδρομητής διαπιστώνει αν το HN εξουσιοδότησε τον έλεγχο ταυτότητας από το συγκεκριμένο SN.

Με την αμοιβαία αυθεντικοποίηση δεν είναι σίγουρο ότι αποτρέπεται μια ενεργή επίθεση, όπως θα ήταν η περίπτωση Man-in-the-middle, όπου θα χρησιμοποιείτο ένας ψεύτικος σταθμός βάσης. Πάντως, είναι σίγουρο ότι σε συνδυασμό με τους υπόλοιπους μηχανισμούς ασφαλείας ο επιτιθέμενος δεν θα είχε κανένα όφελος από αυτή την επίθεση.

Σε αυτό το σημείο θα περιγράψουμε το μηχανισμό αυθεντικοποίησης σε ένα γενικότερο επίπεδο, χωρίς να εισέλθουμε στα ενδότερα των αλγορίθμων που εκτελούνται κατά τη συγκεκριμένη φάση.

Η διαδικασία θυμίζει σε ένα μεγάλο βαθμό την αυθεντικοποίηση του συνδρομητή στο δίκτυο GSM. Το σχήμα που ακολουθεί θα μας βοηθήσει σε αυτή την περιγραφή.



Εικόνα 47: Η αυθεντικοποίηση ενός συνδρομητή στο δίκτυο UMTS

Η διαδικασία ξεκινά με τον συνδρομητή να υποβάλει την προσωρινή(TMSI) ή τη μόνιμη(IMSI) του ταυτότητα στο δίκτυο που τον εξυπηρετεί.

Η ταυτότητα του συνδρομητή καταλήγει στο VLR ή το SGSN του δικτύου εξυπηρέτησης και από εκεί δημιουργείται ένα αίτημα για τη δημιουργία δεδομένων αυθεντικοποίησης (Authentication Data Request), το οποίο αποστέλλεται μαζί με την ταυτότητα του συνδρομητή στο οικείο δίκτυο του εν λόγω συνδρομητή.

Στο οικείο δίκτυο αναλαμβάνει δράση η οντότητα, είτε αυτή είναι το AuC ή το HSS, που είναι επιφορτισμένη με τις λειτουργίες ασφάλειας.

Το AuC είναι παρόν στην περίπτωση που το δίκτυο έχει υλοποιηθεί σύμφωνα με τις προδιαγραφές Release 4, ή με κάποιες παλαιότερες προδιαγραφές.

Η οντότητα του AuC, ακολουθώντας τη λογική που υπήρχε στο δίκτυο GSM, είναι η μόνη που γνωρίζει το μυστικό κλειδί K_i που αντιστοιχεί στο συνδρομητή. Οπότε, το AuC γεννά μια σειρά από τυχαίους αριθμούς τους οποίους εισάγει μαζί με την τιμή του K_i σε ένα σύνολο από αλγόριθμους ασφάλειας.

Με την ολοκλήρωση της εκτέλεσης των αλγορίθμων δημιουργείται μια δέσμη από διανύσματα αυθεντικοποίησης (Authentication Vectors - AVs), τα οποία από εκεί προωθούνται στο HLR του οικείου δικτύου, το οποίο στέλνει στο δίκτυο εξυπηρέτησης τα AVs σαν την απάντηση στο αίτημα για δεδομένα αυθεντικοποίησης (Authentication Data Response).

Αν το δίκτυο UMTS έχει στηθεί ακολουθώντας τις προδιαγραφές Release 5 ή κάποιες νεότερες, τότε έχουμε την οντότητα HSS, να αντικαθιστά τα AuC και HLR, και να διεκπεραιώνει τη διαδικασία της δημιουργίας και προώθησης των διανυσμάτων αυθεντικοποίησης προς το δίκτυο που εξυπηρετεί το συνδρομητή.

Στη συνέχεια, το δίκτυο εξυπηρέτησης λαμβάνει από το οικείο δίκτυο τη δέσμη με τα διανύσματα αυθεντικοποίησης, έτσι το VLR/SGSN που αντιστοιχεί σε αυτό επιλέγει ένα από τα διανύσματα και απομονώνει δύο παραμέτρους του. Τον τυχαίο αριθμό RAND(Random Number) και τη σκυτάλη αυθεντικοποίησης AUTN(Authentication Token).

Το δίκτυο εξυπηρέτησης αποστέλλει στον τερματικό εξοπλισμό του συνδρομητή ένα αίτημα αυθεντικοποίησης (User Authentication Request) που περιέχει τις παραμέτρους RAND και AUTN.

Το αίτημα αυθεντικοποίησης λαμβάνεται από τον τερματικό εξοπλισμό του συνδρομητή και ξεκινά η δημιουργία της απάντησης σε αυτό. Ο στόχος είναι να υπολογιστούν εκ νέου οι παράμετροι του διανύσματος.

Στο περιβάλλον της κάρτας USIM εκτελούνται οι ίδιοι αλγόριθμοι με αυτούς που εκτελέστηκαν στο AuC/HSS κατά τη φάση της δημιουργίας διανυσμάτων αυθεντικοποίησης.

Για την εκτέλεση των αλγορίθμων απαιτείται η παρουσία του RAND, του AUTN και του K_i. Το μυστικό κλειδί K_i βρίσκεται ήδη αποθηκευμένο στην κάρτα USIM, μένει λοιπόν να τροφοδοτηθεί η κάρτα USIM με τις παραμέτρους που περιέχονται στο αίτημα αυθεντικοποίησης.

Μετά την εκτέλεση των αλγορίθμων δημιουργούνται τα υπόλοιπα στοιχεία του διανύσματος αυθεντικοποίησης. Η κάρτα USIM είναι σε θέση να κρίνει από τα στοιχεία που δημιουργήθηκαν αν το AUTN πράγματι δημιουργήθηκε στο AuC/HSS που βρίσκεται στο οικείο δίκτυο.

Στην περίπτωση που ο έλεγχος του AUTN έχει θετικό αποτέλεσμα, τότε αποστέλλεται από τον τερματικό εξοπλισμό του συνδρομητή προς το δίκτυο εξυπηρέτησης η απάντηση του αιτήματος αυθεντικοποίησης.

Στην απάντηση περιέχεται ένα από τα στοιχεία που προέκυψαν από την εκτέλεση των αλγορίθμων στην κάρτα USIM, είναι το RES(Result).

Σε αυτό το σημείο το VLR/SGSN συγκρίνει το RES με την τιμή της προβλεπόμενης απάντησης που υπήρχε στο αρχικό διάνυσμα αυθεντικοποίησης. Με αυτό τον τρόπο μπορεί να αποφανθεί αν ο συνδρομητής είναι πράγματι αυτός που ισχυρίζεται ότι είναι.

Αν η τιμή του RES είναι σωστή τότε έχουμε την επιτυχημένη ολοκλήρωση της διαδικασίας αυθεντικοποίησης, έτσι επιτρέπεται η πρόσβαση του συνδρομητή στο δίκτυο.

Για την ασφαλή επικοινωνία ανάμεσα στο δίκτυο και το συνδρομητή χρησιμοποιούνται δύο προσωρινά κλειδιά, τα οποία περιέχονται στα AVs και είναι παράγωγα της διαδικασίας αυθεντικοποίησης.

Πρόκειται για το κλειδί κρυπτογράφησης CK(Cipher Key) και το κλειδί διαφύλαξης της ακεραιότητας IK(Integrity Key).

3.3.3 Η δημιουργία των διανυσμάτων αυθεντικοποίησης

Είδαμε νωρίτερα ότι ο μηχανισμός της αμοιβαίας αυθεντικοποίησης, μεταξύ του συνδρομητή και του δικτύου, βασίζεται στη σύγκριση των τιμών ορισμένων παραμέτρων, οι οποίες περιέχονται στα διανύσματα αυθεντικοποίησης που δημιουργούνται στις δύο πλευρές.

Σε αυτή την υποενότητα θα δούμε αναλυτικότερα τις συναρτήσεις με τις οποίες δημιουργούνται οι παράμετροι των διανυσμάτων.

Ξεκινάμε με την πλευρά του οικείου δικτύου, το οποίο μόλις δεχθεί ένα αίτημα για αυθεντικοποίηση κάποιου συνδρομητή, τότε αναθέτει στο AuC/HSS τη δημιουργία μιας δέσμης διανυσμάτων αυθεντικοποίησης.

Ένα διάνυσμα δημιουργείται μετά από την εκτέλεση πέντε συναρτήσεων, των f1, f2, f3, f4 και f5, οι οποίες παρέχονται μέσα από μια κοινή υλοποίηση που είναι γνωστή ως MILENAGE.

Όλη η ασφάλεια της δημιουργίας διανυσμάτων βασίζεται στο γεγονός ότι αυτές οι συναρτήσεις είναι μονόδρομες(one way functions).

Χάρη σε αυτό το χαρακτηριστικό των συναρτήσεων καθίσταται αδύνατος ο υπολογισμός των στοιχείων που εισήχθησαν σε αυτές κατά τη δημιουργία των διανυσμάτων.

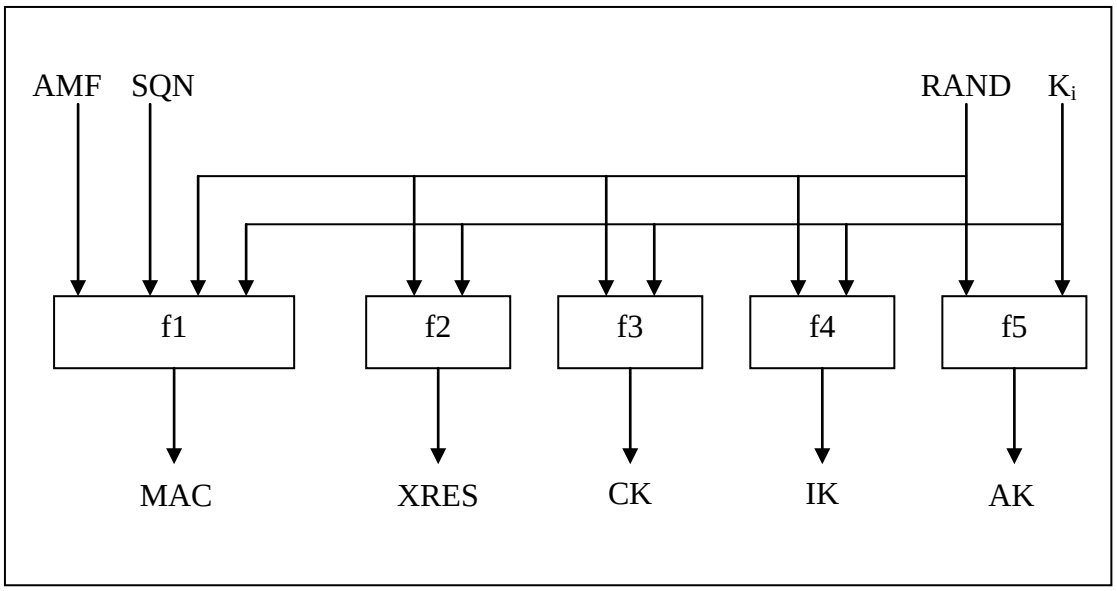
Για την εκτέλεση των f2, f3, f4 και f5 απαιτείται η είσοδος δύο στοιχείων, πρόκειται για τον τυχαίο αριθμό RAND και το μυστικό κλειδί K_i.

Το AuC/HSS γνωρίζει ήδη την τιμή του Ki που αντιστοιχεί σε κάθε συνδρομητή που ανήκει στο οικείο δίκτυο, ενώ ο RAND, που έχει μήκος 128 δυαδικών ψηφίων, δημιουργείται από μια γεννήτρια τυχαίων αριθμών που υπάρχει στο AuC/HSS.

Για τη συνάρτηση f1 εισάγονται πάλι το RAND και το K_i, μαζί τους όμως εισάγονται και δύο επιπρόσθετα στοιχεία, τα οποία είναι ο αριθμός ακολουθίας SQN(Sequence Number) και το πεδίο διαχείρισης της αυθεντικοποίησης AMF(Authentication Management Field).

Ο SQN, που είναι ένας αύξων αριθμός με μήκος 48 δυαδικών ψηφίων, επιλέγεται από το AuC/HSS έχοντας σαν σκοπό να αποδειχθεί στον συνδρομητή ότι το διάνυσμα αυθεντικοποίησης είναι σχετικά πρόσφατο, και ότι δεν έχει χρησιμοποιηθεί ξανά σε κάποια διαδικασία αυθεντικοποίησης.

Το πεδίο AMF έχει μήκος 16 δυαδικών ψηφίων. Η τιμή του είναι σταθερή και καθορίζεται ανάλογα με την υλοποίηση των συναρτήσεων f1, f2, f3, f4 και f5. Η επιλογή της υλοποίησης των συναρτήσεων είναι ευθύνη της εταιρείας παροχής κινητής τηλεφωνίας, αφού εκείνη ελέγχει το περιβάλλον εκτέλεσης των συναρτήσεων, με άλλα λόγια το AuC/HSS και οι κάρτες USIM.



Εικόνα 48: Η εκτέλεση των πέντε συναρτήσεων ασφαλείας στο περιβάλλον του Auc/HSS

Από τη συνάρτηση f1 παράγεται ο μήκους 64 bit κωδικός αυθεντικοποίησης μηνύματος MAC(Message Authentication Code).

Ο MAC έχει παρόμοια χρήση με το SRES που υπάρχει στο GSM, με τη διαφορά ότι ο συγκεκριμένος κωδικός χρησιμοποιείται για την αυθεντικοποίηση του δικτύου από τον χρήστη, όπου στην κάρτα USIM εξετάζεται αν πράγματι το δίκτυο κατέχει την τιμή του αντίστοιχου μυστικού κλειδιού K_i.

Η συνάρτηση f2 δημιουργεί την παράμετρο της αναμενόμενης απάντησης XRES(Expected Response).

Η XRES έχει μεταβλητό μήκος, από 32 ως 128 bit, και χρησιμοποιείται για την αυθεντικοποίηση του χρήστη από το δίκτυο, αφού είναι η απάντηση που πρέπει να δοθεί από το συνδρομητή ανάλογα με την πρόκληση που έχει τεθεί από το δίκτυο.

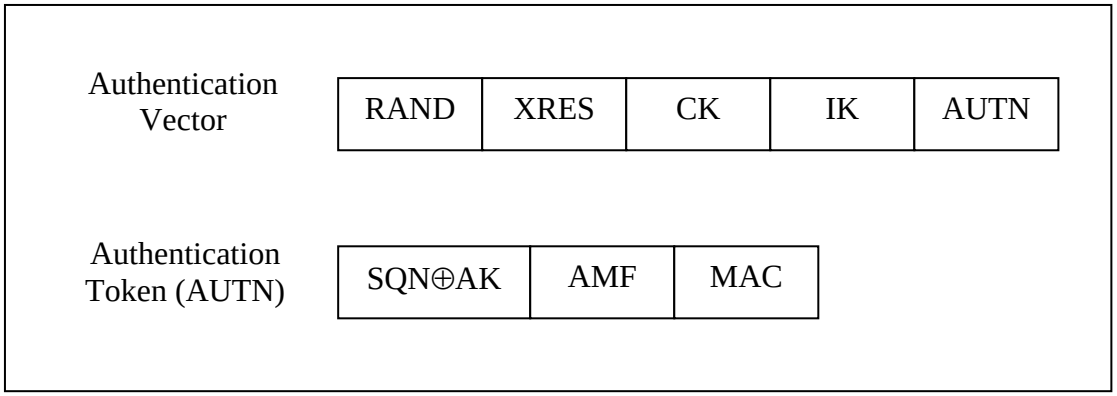
Δίνοντας τη σωστή απάντηση ο συνδρομητής αποδεικνύει ότι κατέχει το σωστό κλειδί K_i και είναι πράγματι αυτός που ισχυρίζεται ότι είναι.

Με τη συνάρτηση f3 δημιουργείται το κλειδί κρυπτογράφησης CK(Cipher Key), το οποίο έχει μήκος 128 bit.

Το αποτέλεσμα της συνάρτησης f4 είναι το κλειδί ακεραιότητας IK(Integrity Key), που έχει μήκος 128 δυαδικών ψηφίων.

Από τη συνάρτηση f5 λαμβάνεται το μήκους 64 bit κλειδί ανωνυμίας AK(Anonymity Key).

Ένα διάνυσμα αυθεντικοποίησης περιέχει τις τιμές των RAND, XRES, CK, IK και AUTN.



Εικόνα 49: Η δομή ενός διανύσματος αυθεντικοποίησης και του πεδίου AUTN

Η τιμή του AUTN σχηματίζεται από τη συνένωση των ακόλουθων τριών παραμέτρων:

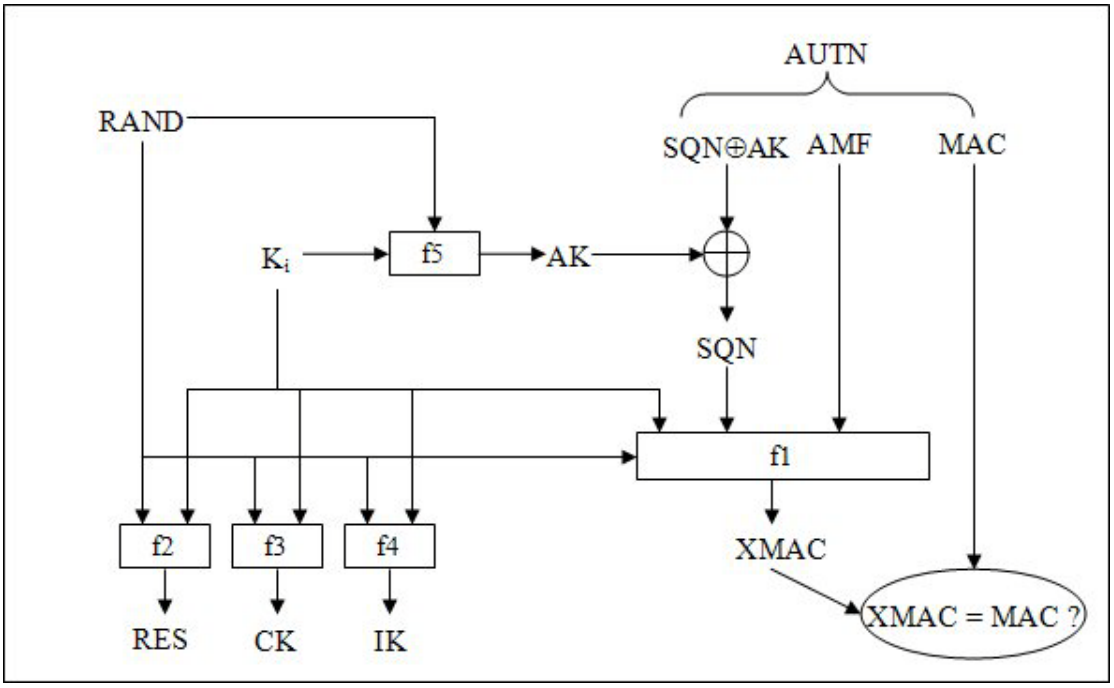
- Το αποτέλεσμα που προκύπτει από την πράξη XOR ανάμεσα στο SQN και το AK
- Το πεδίο AMF
- Την τιμή του MAC

Αυτός ήταν ο τρόπος με τον οποίο δημιουργείται ένα διάνυσμα αυθεντικοποίησης στο οικείο δίκτυο. Η συγκεκριμένη διαδικασία επαναλαμβάνεται μέχρι να σχηματιστεί μια δέσμη διανυσμάτων, την οποία το AuC/HSS προωθεί στο δίκτυο που εξυπηρετεί το συνδρομητή.

Το VLR/SGSN του δικτύου εξυπηρέτησης επιλέγει ένα από τα διανύσματα και στέλνει στον κινητό σταθμό του συνδρομητή ένα αίτημα αυθεντικοποίησης που περιέχει τις τιμές των RAND και AUTN.

Οι τιμές που περιέχονται στο αίτημα τροφοδοτούνται στην κάρτα USIM, όπου «κατοικούν» οι συναρτήσεις f1, f2, f3, f4 και f5. Μέσα από την εκτέλεση των πέντε συναρτήσεων έχουμε την αναγέννηση του διανύσματος που επιλέχθηκε από το δίκτυο εξυπηρέτησης.

Οι συναρτήσεις εκτελούνται με διαφορετική σειρά από αυτή που ακολουθήθηκε στο AuC/HSS. Στο σχήμα που ακολουθεί εμφανίζεται η πορεία που ακολουθείται κατά την εκτέλεση των συναρτήσεων.



Εικόνα 50: Η εκτέλεση των πέντε συναρτήσεων ασφαλείας στο περιβάλλον της κάρτας USIM

Η συνάρτηση f5 πρέπει να εκτελεστεί πρώτη, γιατί το κλειδί ανωνυμίας AK που προκύπτει από αυτήν χρησιμοποιείται για να αποκαλυφθεί ο αριθμός ακολουθίας από το AUTN.

Να θυμίσουμε ότι το AUTN δημιουργείται από το AuC/HSS και σχηματίζεται από τη συνένωση τριών παραμέτρων. Η πρώτη παράμετρος, που μας ενδιαφέρει περισσότερο σε αυτό το σημείο, προκύπτει από την πράξη XOR ανάμεσα στο SQN και το AK.

Με αυτό τον τρόπο δεν αποκαλύπτεται η τιμή του SQN σε κάποιον που μπορεί να παρακολουθεί τη μετάδοση του διανύσματος από το δίκτυο προς τον συνδρομητή και έτσι αποτρέπεται η αποκάλυψη της ταυτότητας του συνδρομητή.

Με την εφαρμογή της πράξης XOR ανάμεσα στο πρώτο πεδίο του AUTN, που περιέχει την τιμή SQN⊕AK, και το AK που δημιουργήθηκε από την f5 λαμβάνουμε την τιμή του SQN που χρησιμοποιήθηκε από το AuC/HSS του οικείου δικτύου.

Αμέσως μετά την αποκάλυψη του SQN εκτελείται η συνάρτηση f1, από την οποία προκύπτει η παράμετρος XMAC(Expected Message Authentication Code). Με αυτή την παράμετρο αντιπροσωπεύεται η τιμή που αναμένεται να έχει ο κώδικας MAC που έχει αποσταλεί από το δίκτυο.

Αν το MAC που στάλθηκε από το δίκτυο και το XMAC συμπίπτουν, τότε η κάρτα USIM συμπεραίνει ότι το RAND και το AUTN πράγματι δημιουργήθηκαν στο AuC/HSS του οικείου δικτύου, καθώς μόνο αυτή η οντότητα γνωρίζει την τιμή K_i που αντιστοιχεί στην κάρτα USIM του συνδρομητή.

Σε διαφορετική περίπτωση, αν δηλαδή το XMAC δεν είναι ίσο με το MAC, η αυθεντικοποίηση στην κάρτα USIM τερματίζεται και ο τερματικός εξοπλισμός του συνδρομητή ενημερώνει το δίκτυο εξυπηρέτησης για το σφάλμα που παρουσιάστηκε.

Από το δίκτυο εξυπηρέτησης επιλέγεται ένα άλλο διάνυσμα από αυτά που βρίσκονται στη δέσμη και επιχειρείται ξανά η διαδικασία της αυθεντικοποίησης στην κάρτα USIM του συνδρομητή.

Οι συναρτήσεις f2,f3 και f4 εκτελούνται μόνο μετά από τον επιτυχημένο έλεγχο της τιμής MAC.

Με τη συνάρτηση f3 παράγεται το κλειδί CK, που χρησιμοποιείται για την εκτέλεση του αλγορίθμου κρυπτογράφησης των επικοινωνιών που διεξάγονται στο UTRAN.

Από την εκτέλεση της f4 δημιουργείται το κλειδί IK, που εισάγεται σε μια συνάρτηση διαφύλαξης της ακεραιότητας των επικοινωνιών του UTRAN.

Από την f2 προκύπτει η παράμετρος RES(Result), η οποία αποτελεί την απάντηση του συνδρομητή στην πρόκληση που τέθηκε από το δίκτυο.

Ο συνδρομητής στέλνει το RES στο δίκτυο εξυπηρέτησης. Εκεί γίνεται η σύγκριση με την προβλεπόμενη τιμή XRES και ανάλογα με το αποτέλεσμα αποφασίζεται αν θα επιτραπεί στο συνδρομητή η πρόσβαση στο δίκτυο.

Στην περίπτωση που το RES και το XRES ταυτίζονται, τότε έχουμε την επιτυχημένη ολοκλήρωση της αμοιβαίας αυθεντικοποίησης.

3.3.4 Η χρησιμοποίηση προσωρινών ταυτοτήτων

Η μόνιμη ταυτότητα του χρήστη του δικτύου UMTS είναι το IMSI, όπως ήταν άλλωστε και στην περίπτωση του δικτύου GSM. Όμως, για την αναγνώριση του χρήστη από το δίκτυο χρησιμοποιείται πάντα κάποια από τις προσωρινές ταυτότητες.

Υπάρχουν δύο τύποι προσωρινών ταυτοτήτων με τις οποίες αναγνωρίζεται ο χρήστης από το δίκτυο:

- η προσωρινή ταυτότητα TMSI, που χρησιμοποιείται στον τομέα του δικτύου με μεταγωγή κυκλώματος
- η ταυτότητα P-TMSI, που χρησιμοποιείται στον τομέα του δικτύου μεταγωγής πακέτου

Με τη χρησιμοποίηση των προσωρινών ταυτοτήτων αποφεύγεται η έκθεση της μόνιμης ταυτότητας στο UTRAN. Με αυτό τον τρόπο προστατεύεται η εμπιστευτικότητα της μόνιμης ταυτότητας, αφού είναι πιθανό να παρακολουθείται η κίνηση στο UTRAN από κάποιον κακόβουλο χρήστη.

Μια περίπτωση όπου ο συνδρομητής δεν έχει κάποια προσωρινή ταυτότητα στη διάθεση του είναι την πρώτη φορά που θα επιχειρήσει να συνδεθεί στο δίκτυο, οπότε θα πραγματοποιηθεί για πρώτη φορά και η διαδικασία της αυθεντικοποίησης.

Το δίκτυο δεν είναι σίγουρο για την ταυτότητα του συνδρομητή, γι’ αυτό απαιτείται η χρήση της μόνιμης ταυτότητας. Μετά την επιτυχημένη αυθεντικοποίηση

ενεργοποιείται η κρυπτογράφηση στο UTRAN, ενώ ταυτόχρονα εκδίδεται από το δίκτυο μια προσωρινή ταυτότητα για τον συνδρομητή.

Η προσωρινή ταυτότητα κρυπτογραφείται και προωθείται από το δίκτυο προς το συνδρομητή.

Από εκείνο το σημείο και μετά ο συνδρομητής χρησιμοποιεί πάντα μια προσωρινή ταυτότητα, ακόμα και στις αυθεντικοποιήσεις που θα ακολουθήσουν. Αφού το δίκτυο διατηρεί συνεχώς μια συσχέτιση ανάμεσα στο IMSI του συνδρομητή και την προσωρινή ταυτότητα που του έχει ανατεθεί, ακόμα και αν ο συνδρομητής απενεργοποιήσει της συσκευή του.

Μια προσωρινή ταυτότητα έχει ισχύ για το χρονικό διάστημα που ένας συνδρομητής κινείται στην περιοχή ευθύνης του δικτύου που του έχει αναθέσει την προσωρινή ταυτότητα. Την ίδια χρονική περίοδο το VLR/SGSN προσέχει μην αναθέσει ταυτόχρονα την ίδια ταυτότητα σε κάποιον άλλο συνδρομητή.

Όταν ο συνδρομητής μετακινηθεί σε διαφορετική περιοχή, και βρεθεί υπό τον έλεγχο ενός άλλου VLR/SGSN, θα πρέπει με κάποιο τρόπο να είναι σίγουρο ότι η προσωρινή ταυτότητα του συνδρομητή συνεχίζει να είναι μοναδική.

Αυτή η απαίτηση ικανοποιείται με την προσάρτηση της ταυτότητας της περιοχής στην οποία κινείται ο συνδρομητής.

Γι’ αυτό το λόγο στην περίπτωση του δικτύου μεταγωγής κυκλώματος, όπου χρησιμοποιείται η προσωρινή ταυτότητα TMSI, έχουμε την προσάρτηση της ταυτότητας περιοχής Local Area Identity(LAI).

Ενώ, στο δίκτυο μεταγωγής πακέτου, όπου χρησιμοποιείται η ταυτότητα P-TMSI, προσαρτάται η ταυτότητα Routing Area Identity(RAI).

Η ταυτότητα της περιοχής στην οποία κινείται ο συνδρομητής είναι σημαντική και για έναν επιπλέον λόγο. Το VLR/SGSN που έχει υπό την ευθύνη του τον συνδρομητή πρέπει να γνωρίζει ανά πάσα στιγμή τη συσχέτιση που υπάρχει ανάμεσα στο IMSI και την προσωρινή ταυτότητα που έχει ο συνδρομητής.

Έτσι, όταν ο συνδρομητής μετακινείται σε μια νέα περιοχή, πρέπει και το νέο VLR/SGSN να ενημερωθεί για τη συσχέτιση της μόνιμης και της προσωρινής ταυτότητας.

Γι’ αυτό το νέο VLR/SGSN εξετάζει την ταυτότητα της περιοχής που έχει προσαρτηθεί στην προσωρινή ταυτότητα του συνδρομητή, βρίσκει ποιο ήταν το VLR/SGSN που είχε αναθέσει την προσωρινή ταυτότητα, και ζητά από εκεί τη συσχέτιση μόνιμης-προσωρινής ταυτότητας και όποια αχρησιμοποίητα διανύσματα αυθεντικοποίησης υπάρχουν.

Στην περίπτωση που το νέο VLR/SGSN δεν λάβει απάντηση από το προηγούμενο, τότε ζητά το IMSI του συνδρομητή και ξεκινά εκ νέου τη διαδικασία αυθεντικοποίησης.

Με την επιτυχημένη ολοκλήρωση της αυθεντικοποίησης εκδίδεται νέα προσωρινή ταυτότητα και δημιουργείται νέα συσχέτιση μόνιμης και προσωρινής ταυτότητας.

3.3.5 Κρυπτογράφηση του UTRAN

Η κρυπτογράφηση στο UTRAN ενεργοποιείται μετά την επιτυχημένη ολοκλήρωση της διαδικασίας αυθεντικοποίησης, καθώς τότε επιβεβαιώνεται ότι ο συνδρομητής και το δίκτυο χρησιμοποιούν το ίδιο διάνυσμα αυθεντικοποίησης.

Μια από τις παραμέτρους του διανύσματος που μοιράζονται ο συνδρομητής και το δίκτυο είναι το κλειδί κρυπτογράφησης CK.

Το συγκεκριμένο κλειδί είναι το σημαντικότερο απ' όσα στοιχεία εισάγονται στη συνάρτηση με το όνομα αναφοράς f8, η οποία χρησιμοποιείται για την κρυπτογράφηση του περιεχομένου των επικοινωνιών.

Οι οντότητες στις οποίες λαμβάνει χώρα η συγκεκριμένη λειτουργία είναι ο τερματικός εξοπλισμός του συνδρομητή και το RNC του δικτύου εξυπηρέτησης. Αυτό σημαίνει, ότι το VLR/SGSN που έχει το διάνυσμα αυθεντικοποίησης που χρησιμοποιήθηκε, πρέπει να προωθήσει την τιμή CK στο αρμόδιο RNC.

Όμως πριν να ξεκινήσει η κρυπτογράφηση, πρέπει τα δύο επικοινωνούντα μέρη να αποφασίσουν από κοινού ποιον αλγόριθμο κρυπτογράφησης θα χρησιμοποιούν.

Για την ώρα υπάρχει ένας αλγόριθμος κρυπτογράφησης για το UMTS. Στις προδιαγραφές αναφέρεται με την ονομασία UEA1(UMTS Encryption Algorithm 1) και βασίζεται στον αλγόριθμο KASUMI.

Ο εν λόγω αλγόριθμος πραγματοποιεί κρυπτογράφηση ανά τμήματα(block cipher) των 64 δυαδικών ψηφίων.

Από την εκτέλεση του KASUMI δημιουργείται μια μάσκα με μήκος 64 bit, η οποία χρησιμοποιείται σε μια πράξη XOR με 64 bit από την ροή της αρχικής πληροφορίας.

Από την πράξη XOR προκύπτει κάθε φορά ένα block των 64 δυαδικών ψηφίων που περιέχει την κρυπτογραφημένη πληροφορία.

Η αποκρυπτογράφηση πραγματοποιείται με τον ίδιο τρόπο. Έχουμε σαν δεδομένο ότι και οι δύο πλευρές που επικοινωνούν μοιράζονται το ίδιο CK.

Έτσι η πλευρά που δέχεται την κρυπτογραφημένη πληροφορία εκτελεί και αυτή τον αλγόριθμο KASUMI με σκοπό να δημιουργήσει τη μάσκα που χρησιμοποιήθηκε κατά την κρυπτογράφηση.

Η μάσκα γίνεται XOR με την κρυπτογραφημένη πληροφορία και το αποτέλεσμα που προκύπτει είναι η αρχική πληροφορία.

Για την εκτέλεση του KASUMI και τη δημιουργία της μάσκας χρειάζεται βεβαίως το κλειδί κρυπτογράφησης CK, που θυμίζουμε ότι έχει μήκος 128 bit.

Υπάρχει όμως η περίπτωση συγκρουόμενων αποτελεσμάτων, να δημιουργηθούν δηλαδή δύο όμοιες μάσκες από δύο διαφορετικές τιμές του κλειδιού CK.

Για αυτό το σκοπό μαζί με το CK εισάγονται στον αλγόριθμο και κάποια επιπλέον στοιχεία που μειώνουν τις πιθανότητες σύγκρουσης. Τα στοιχεία αυτά είναι:

- το μήκος(LENGTH) της εισαγόμενης πληροφορίας
- η ταυτότητα του ράδιο-φορέα(BEARER)
- η κατεύθυνση(DIRECTION) της μεταδιδόμενης πληροφορίας
- ο μετρητής COUNT-C

Το στοιχείο LENGTH καταλαμβάνει 16 δυαδικά ψηφία και η τιμή του εξαρτάται από το μήκος που έχει η πληροφορία που εισέρχεται στη συνάρτηση f8.

Η ταυτότητα του ράδιο-φορέα, που έχει μήκος 5 δυαδικών ψηφίων, έχει μοναδική τιμή και αντιπροσωπεύει το κανάλι που χρησιμοποιείται για τη μεταφορά της κίνησης.

Η παράμετρος DIRECTION έχει μήκος 1 bit και η τιμή της μεταβάλλεται ανάλογα με την κατεύθυνση της μεταδιδόμενης πληροφορίας, αν είναι δηλαδή εισερχόμενη(downlink) ή εξερχόμενη(uplink).

Ο μετρητής COUNT-C έχει μήκος 32 δυαδικών ψηφίων και αυξάνεται για κάθε block πληροφορίας που περνά από τη συνάρτηση f8. Η τιμή του COUNT-C προκύπτει από το συνδυασμό δύο άλλων μετρητών.

Η κρυπτογράφηση και η αποκρυπτογράφηση πραγματοποιείται στο επίπεδο ελέγχου πρόσβασης μέσου(Medium Access Control - MAC) ή στο επίπεδο ελέγχου ράδιο-ζεύξης(Radio Link Control - RLC).

Σε κάθε περίπτωση, διατηρείται ένας μετρητής που μεταβάλλεται για κάθε μονάδα δεδομένων(Protocol Data Unit - PDU) που δημιουργείται. Στο MAC υπάρχει ένας αριθμός διόρθωσης πλαισίου(Correction Frame Number - CFN), ενώ στο RLC χρησιμοποιείται ένας αριθμός ακολουθίας(RLC Sequence Number – RLC-SN).

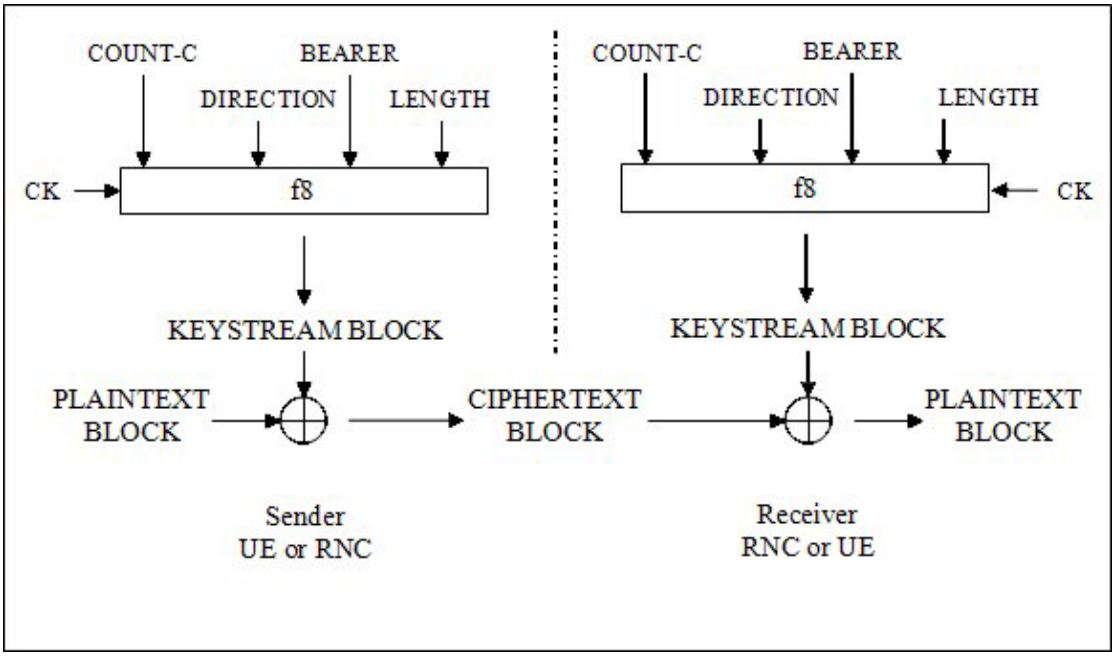
Όμως ο CFN και ο RLC-SN φτάνουν πολύ εύκολα στη μέγιστη τιμή τους, οπότε αναγκαστικά μηδενίζονται και αρχίζουν ξανά την αρίθμηση από το μηδέν.

Το γεγονός αυτό θα οδηγούσε στη δημιουργία της ίδιας μάσκας, αν βέβαια οι αριθμοί εισάγονταν ως έχουν στη συνάρτηση. Γι’ αυτό το λόγο, χρησιμοποιείται ένας επιπλέον μετρητής που ονομάζεται Hyper Frame Number(HFN), ο οποίος αυξάνεται κατά ένα κάθε φορά που ο CFN ή ο RLC-SN ξεκινούν από την αρχή.

Ο συνδυασμός του HFN με το μετρητή που χρησιμοποιείται, είτε είναι ο CFN είτε είναι ο RLC-SN, σχηματίζουν τον αριθμό COUNT-C που εισάγεται κατά την παραγωγή της μάσκας. Ο αριθμός COUNT-C μηδενίζεται κάθε φορά που ο συνδρομητής πραγματοποιεί τη διαδικασία αυθεντικοποίησης.

Αυτό έχει ως αποτέλεσμα να είναι σπάνια η περίπτωση που στη συνάρτηση f8 θα εισαχθούν διαφορετικές παράμετροι και θα δημιουργηθούν δύο όμοιες μάσκες.

Στο σχήμα που ακολουθεί παρουσιάζεται η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης δύο πλευρών που επικοινωνούν μέσω του δικτύου UMTS.



Εικόνα 51: Η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης στο UTRAN

Μέχρι τις αρχές του 2010 δεν είχε βρεθεί κάποιο τρωτό σημείο που να αφορούσε τη λειτουργία του αλγορίθμου KASUMI, όμως η έρευνα των Orr Dunkelmann, Nathan Keller και Adi Shamir απέδειξε ότι υπάρχει, περισσότερο σε θεωρητικό επίπεδο, η δυνατότητα αποκάλυψης των περιεχομένων μιας συνομιλίας που μέχρι πρότινος θεωρούνταν εμπιστευτικά.

Με τη συγκεκριμένη επίθεση[*], που ονομάστηκε από τους ερευνητές ως "sandwich attack", αποκαλύπτεται πλήρως το κλειδί μήκους 128 δυαδικών ψηφίων που χρησιμοποιείται από τον αλγόριθμο KASUMI.

Για την πρακτική επίτευξη της παραπάνω επίθεσης απαιτούνται 4 συσχετιζόμενα κλειδιά του αλγορίθμου KASUMI όπου σε συνδυασμό με μια ποσότητα 2^{26} αρχικής πληροφορίας θα χρησιμοποιούν 2^{30} bytes μνήμης σε ένα βάθος χρόνου 2^{32}.

Σύμφωνα με τους ερευνητές, η επίθεση παρουσιάζει χαμηλό βαθμό πολυπλοκότητας και δεν έχει υψηλές απαιτήσεις σε υλικό, γεγονός που επιτρέπει την ολοκλήρωση μιας προσομοίωσης από έναν απλό υπολογιστή σε χρονικό διάστημα λιγότερο από 2 ώρες.

Το ουσιαστικότερο πρόβλημα που μπορεί να συναντήσει που πραγματοποιεί τη συγκεκριμένη επίθεση έγκειται στη συλλογή των τεσσάρων συσχετιζόμενων κλειδιών

[*] Περισσότερες πληροφορίες σχετικά με την επίθεση στον αλγόριθμο KASUMI παρέχονται μέσω της επιστημονικής δημοσίευσης που βρίσκεται στην ιστοσελίδα http://eprint.iacr.org/2010/013

του αλγορίθμου, αφού μια σωστή υλοποίηση του KASUMI δεν επιτρέπει την ανάκτηση ούτε δύο τέτοιων κλειδιών.

Πάντως, η απάντηση στις προαναφερθείσες προκλήσεις είναι έτοιμη, καθώς από τις προδιαγραφές του UMTS προβλέπεται η χρησιμοποίηση του δευτέρου αλγόριθμου κρυπτογράφησης που είναι γνωστός και ως UEA2. Ενώ, δεν πρέπει να παραβλέπουμε το γεγονός ότι παρουσιάζονται συνεχώς αλγόριθμοι με ενισχυμένη ασφάλεια, τόσο σε νέα επικοινωνιακά standards, όπως είναι το 4^{ης} γενιάς δίκτυο LTE, όσο και σε ήδη υπάρχοντα, όπως είναι τα δίκτυα 2G και 3G.

Σαν παράδειγμα αλγορίθμου της συγκεκριμένης κατηγορίας, αναφέρουμε τον SNOW 3G, που πρωτοεμφανίστηκε για τις ανάγκες του δευτέρου αλγορίθμου κρυπτογράφησης του UMTS, γνωστού και ως UEA2.

Ο SNOW 3G θα αποτελέσει τη βάση και για την ανάπτυξη της ασφάλειας του δικτύου LTE, ενώ είναι πολύ πιθανό να χρησιμοποιηθεί και για την αναθεώρηση του αλγορίθμου A5 που χρησιμοποιείται από τα δίκτυα δεύτερης γενιάς.

3.3.6 Η διαφύλαξη της ακεραιότητας των σημάτων στο UTRAN

Ο μηχανισμός για τη διαφύλαξη της ακεραιότητας είναι αρκετά χρήσιμος, αφού έτσι επιτυγχάνεται η αυθεντικοποίηση κάθε μηνύματος που μεταδίδεται στο UTRAN.

Η διαδικασία της αμοιβαίας αυθεντικοποίησης του συνδρομητή και του δικτύου δεν μπορεί να εγγυηθεί για την ταυτότητα των δύο επικοινωνούντων πλευρών μετά τη λήξη διαδικασίας.

Με αυτό τον τρόπο μπορεί να πραγματοποιεί η επίθεση Man-in-the-Middle, όπου ο επιτιθέμενος χρησιμοποιώντας ένα ψεύτικο σταθμό βάσης θα προωθεί αμετάβλητα τα μηνύματα που ανταλλάσσονται κατά τη διαδικασία αυθεντικοποίησης, όμως με τη λήξη της διαδικασίας θα μπορεί να τροποποιεί τα μηνύματα όπως εκείνος επιθυμεί.

Αν όμως κάθε μήνυμα προστατεύεται από έναν κωδικό αυθεντικοποίησης MAC-I (Message Authentication Code), τότε είναι δυνατή η αναγνώριση ενός ψεύτικου ή ενός τροποποιημένου μηνύματος, οπότε κρίνεται αν το μήνυμα θα ληφθεί υπόψη ή θα απορριφθεί.

Ο μηχανισμός διαφύλαξης της ακεραιότητας πραγματοποιείται ανάμεσα στον τερματικό εξοπλισμό του συνδρομητή και το RNC, όπως συμβαίνει και με την περίπτωση της κρυπτογράφησης.

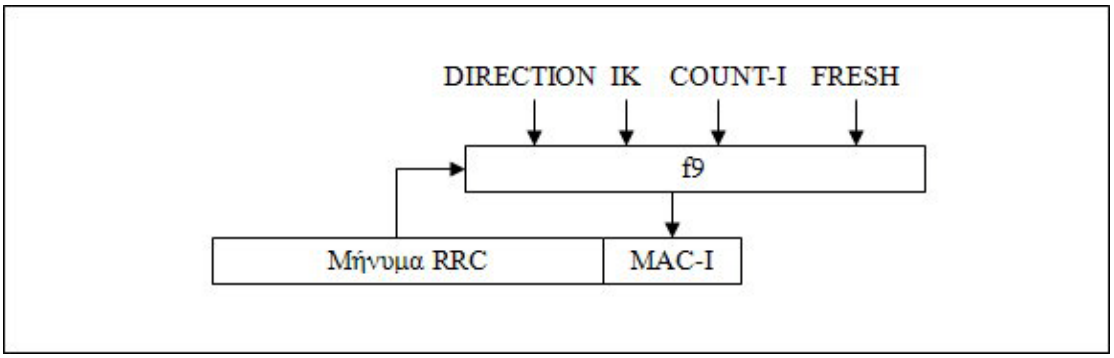
Με τη διαφορά ότι η διαφύλαξη ακεραιότητας εφαρμόζεται στο επίπεδο του ελέγχου ράδιο-σημάτων RRC(Radio Resource Control).

Ο κωδικός αυθεντικοποίησης ενός μηνύματος έχει τη μορφή μιας ψευδό-τυχαίας αλφαριθμητικής ακολουθίας 32 δυαδικών ψηφίων και παράγεται από την εκτέλεση μιας μονόδρομης συνάρτησης με την ονομασία αναφοράς f9.

Σε κάθε RRC μήνυμα επισυνάπτεται ο αντίστοιχος κωδικός που δημιουργείται από την f9. Η πλευρά που λαμβάνει το μήνυμα καλείται να υπολογίσει ξανά τον κωδικό αυθεντικοποίησης και εξετάζει αν είναι ο ίδιος με αυτόν που έχει επισυναφθεί στο μήνυμα.

Για την εκτέλεση της f9 απαιτείται η εισαγωγή του κλειδιού ακεραιότητας IK, μαζί με το υπό μετάδοση μήνυμα RRC και τις παραμέτρους DIRECTION, FRESH και COUNT-I.

Στο σχήμα που ακολουθεί παρουσιάζεται η εκτέλεση της συνάρτησης f9 και η παραγωγή του κωδικού αυθεντικοποίησης MAC-I.



Εικόνα 52: Η διαδικασία με την οποία δημιουργείται ο κωδικός αυθεντικοποίησης MAC-I

Οι δύο πλευρές που επικοινωνούν μοιράζονται το κλειδί IK, που έχει μήκος 128 δυαδικών ψηφίων, και έχει δημιουργηθεί κατά τη διαδικασία αυθεντικοποίησης.

Η παράμετρος DIRECTION καταλαμβάνει 1 bit και δηλώνει αν το μήνυμα RRC είναι εισερχόμενο ή εξερχόμενο.

Ο μετρητής COUNT-I θυμίζει αρκετά το μετρητή COUNT-C που χρησιμοποιείται κατά την κρυπτογράφηση. Ο μετρητής έχει μήκος 32 δυαδικών ψηφίων, με τα 28 περισσότερο σημαντικά να προέρχονται από τον αριθμό HFN και τα 4 λιγότερο σημαντικά ψηφία να αποτελούνται από τον αριθμό ακολουθίας RRC-SQN.

Ο COUNT-I χρησιμοποιείται για την προστασία από την επανεκπομπή(replay) κωδικών MAC, καθώς εγγυάται ότι η συνάρτηση f9 θα παράγει διαφορετικές τιμές κάθε φορά που εκτελείται.

Η παράμετρος FRESH επιλέγεται από το RNC και στέλνεται στον τερματικό εξοπλισμό του συνδρομητή. Είναι ένας τυχαίος αριθμός μήκους 32 bit, που χρησιμοποιείται για να αποτρέψει το ενδεχόμενο της αποστολής των ίδιων κωδικών MAC-I από κάποιον κακόβουλο χρήστη.

Η υλοποίηση της συνάρτησης f9 παρέχεται μέσα από μια ειδική λειτουργία του αλγορίθμου KASUMI. Αν προκύψει η ανάγκη αντικατάστασης του KASUMI, τότε θα χρησιμοποιηθεί ο αλγόριθμος SNOW 3G.

3.4 Η ασφάλεια που παρέχεται σε πραγματικές συνθήκες

Είδαμε προηγουμένως τους μηχανισμούς που χρησιμοποιούνται για την ασφάλεια των επικοινωνιών και των δεδομένων όσων χρηστών συμπεριλαμβάνονται στο δίκτυο GSM και τι συμβαίνει με τους αντίστοιχους χρήστες του δικτύου UMTS.

Ακόμη, είδαμε ότι η ασφάλεια που παρέχεται από το δίκτυο UMTS είναι σε μεγάλο βαθμό ενισχυμένη, ιδιαίτερα σε σχέση με τον πρόγονο του, το δίκτυο GSM που πρέσβευε την παλαιότερη γενιά κινητής τηλεφωνίας. Οπότε, ταυτόχρονα με την εξέλιξη των δικτύων κινητής τηλεφωνίας έχουμε ανάλογη πορεία και για την ασφάλεια που παρέχεται από αυτά.

Βέβαια, μετά από αρκετά χρόνια προσπαθειών ανακαλύφθηκαν κάποιες αδυναμίες στον αλγόριθμο KASUMI, όπου χρησιμοποιείται ως βάση για τον αλγόριθμο UEA1 με τον οποίο πραγματοποιείται η κρυπτογράφηση του UTRAN.

Αλλά και πριν από αυτό το γεγονός υπήρχαν μεμονωμένες περιπτώσεις χωρών όπου, η καθεμία για τους δικούς της λόγους, ακολουθούσε διαφορετική πολιτική στον τομέα της παροχής υπηρεσιών κινητής τηλεφωνίας.

Μερικές από αυτές τις χώρες είναι:

- Η Σαουδική Αραβία[1], που θέτει φραγμούς στην εισαγωγή συσκευών κινητών τηλεφώνων που έχουν ενσωματωμένες περισσότερες δυνατότητες ασφάλειας.
- Η Ουγκάντα[2], όπου μετά από πολλά έτη παράνομων καταγραφών συνομιλιών νομιμοποιεί την παρακολούθηση των πολιτών της χωρίς να απαιτείται η έκδοση εντάλματος.
- Οι Ηνωμένες Πολιτείες της Αμερικής[3], όταν αποκαλύφθηκε ότι κατά τη διάρκεια της προεδρίας του George Bush του νεότερου διενεργούνταν παρακολουθήσεις εις βάρος των αμερικανών πολιτών δίχως την έκδοση του αντίστοιχου εντάλματος έρευνας.

Ενώ μια άλλη ιδιάζουσα περίπτωση αποτελεί η Βόρεια Κορέα, όπου ο τομέας της κινητής τηλεφωνίας έχει περάσει από πληθώρα αλλαγών.

Αρχικά, το 2002, τέθηκε σε λειτουργία το πρώτο δίκτυο κινητής τηλεφωνίας. Όμως η κυβέρνηση, σε μια προσπάθεια περιορισμού των ελευθεριών που παρέχονταν στους πολίτες της, σταμάτησε τη λειτουργία του δικτύου το 2004[4], αφήνοντας ανοιχτό μόνο ένα δίκτυο στην πρωτεύουσα για να το χρησιμοποιούν κυβερνητικοί αξιωματούχοι.

Έκτοτε, συσκευές κινητών τηλεφώνων κυκλοφορούσαν λαθραία στην αγορά, με τους κατόχους αυτών να συνδέονται σε κινέζικα δίκτυα που κάλυπταν εν μέρει τη χώρα. Μέχρι το 2008[5], όπου τέθηκε σε λειτουργία ένα νέο δίκτυο 3ης γενιάς το οποίο βρήκε την άμεση ανταπόκριση από τους κατοίκους της Βόρειας Κορέας που ήθελαν να συνδεθούν σε αυτό.

[1] http://www.bbc.co.uk/news/world-middle-east-10888954
[2] http://allafrica.com/stories/200903110096.html
[3] http://www.usatoday.com/news/washington/2006-05-11-nsa-reax_x.htm
[4] http://www.google.com/hostednews/afp/article/ALeqM5hX7RvD5hXieRvD3BAXL2n4Do5N9w
[5] http://www.msnbc.msn.com/id/28224515/ns/technology_and_science-wireless/

3.4.1 Η περίπτωση των υποκλοπών στην Αθήνα

Η υπόθεση των αθηναϊκών υποκλοπών αποτελεί ένα σπουδαίο παράδειγμα της διαφορετικής λογικής που μπορεί να έχει ένας επιτιθέμενος, όπου πολύ συχνά σκέφτεται αντισυμβατικά και κάτω από μια νέα προοπτική(Thinking outside of the box). Έτσι, για τη συγκεκριμένη περίπτωση, το μοντέλο ασφάλειας του δικτύου κινητής τηλεφωνίας παρακάμφθηκε πλήρως, δίνοντας τη δυνατότητα για τη διεξαγωγή τηλεφωνικών υποκλοπών.

Οι παρακολουθήσεις των κινητών τηλεφώνων πραγματοποιήθηκαν στην Αθήνα και διήρκησαν κάποιους μήνες, ξεκινώντας από την περίοδο των Ολυμπιακών Αγώνων του 2004 μέχρι τους πρώτους μήνες του 2005.

Για όλο αυτό το διάστημα ήταν υπό παρακολούθηση τα τηλέφωνα που είχαν στην κατοχή τους μέλη του Ελληνικού πολιτικού κόσμου, στελέχη της εγχώριας στρατιωτικής ηγεσίας, καθώς και επιφανείς επιχειρηματίες που είχαν ως βάση τους την Αθήνα.

Σύμφωνα με την αναφορά των κυρίων Πρεβελάκη και Σπινέλλη[*] οι παρακολουθήσεις δεν έγιναν εστιάζοντας στο τμήμα της ασύρματης διασύνδεσης του δικτύου κινητής τηλεφωνίας. Οι υπαίτιοι για τις παρακολουθήσεις κατάφεραν με κάποιο τρόπο να αποκτήσουν φυσική πρόσβαση στο δίκτυο κορμού και να επαναπρογραμματίσουν 4 κέντρα μεταγωγής.

Τα συγκεκριμένα κέντρα μεταγωγής έδιναν σε διωκτικές αρχές ή αρχές διάσωσης τη δυνατότητα για νόμιμη παρακολούθηση τηλεφώνων, μέσω των οποίων πραγματοποιείται η εξαγωγή στοιχειών που μπορεί να αποδειχθούν πολύτιμα σε περιπτώσεις όπως είναι:

- Η διαπραγμάτευση μιας απαγωγής
- Η αποτροπή ενός τρομοκρατικού χτυπήματος
- Ο εντοπισμός ενός τραυματία που κάλεσε τον αριθμό έκτακτης ανάγκης(112)

Η νόμιμη παρακολούθηση ενός κινητού τηλεφώνου ξεκινά όταν από τις αρμόδιες αρχές κατατεθεί ένα αίτημα για την έκδοση ενός σχετικού εντάλματος, το οποίο στη συνέχεια παρουσιάζεται στην εταιρεία παροχής κινητής τηλεφωνίας.

Από την πλευρά της εταιρείας κινητής τηλεφωνίας ρυθμίζονται οι τελευταίες λεπτομέρειες για την παρακολούθηση, έτσι οι αρχές μπορούν να λαμβάνουν σε μια δική τους τηλεφωνική συσκευή στοιχεία που σχετίζονται με το υπό παρακολούθηση τηλέφωνο.

Οι υπαίτιοι για τις υποκλοπές εξακολουθούν να παραμένουν άγνωστοι, όμως είναι σίγουρο ότι κατέχουν εξαιρετικές γνώσεις, αφού εγκατέστησαν μια πολύ καλά σχεδιασμένη «διόρθωση» του λογισμικού των κέντρων μεταγωγής. Με τη διορθωμένη έκδοση μπορούσαν να εκμεταλλευτούν τη δυνατότητα για έννομες παρακολουθήσεις τηλεφώνων χωρίς να αφήνουν κανένα ίχνος για τις δραστηριότητες τους.

[*] Ολόκληρη η αναφορά μπορεί να βρεθεί στο URL http://www.spectrum.ieee.org/jul07/5280

Οι παρακολουθήσεις διεξάγονταν με την αντιγραφή των ψηφιοποιημένων ροών πληροφορίας, που αντιπροσώπευαν τις συνομιλίες που διεξάγονταν μέσα από τα τροποποιημένα κέντρα μεταγωγής. Στη συνέχεια, το αντίγραφο καθεμιάς από αυτές τις ροές επροωθείτο σε ένα από τα 14 κινητά τηλέφωνα «σκιές», τα οποία είχαν στην κατοχή τους οι υποκλοπείς και πιθανότατα να τα χρησιμοποιούσαν για να καταγράφουν τις συνομιλίες.

Ένα επιπλέον στοιχείο που είχαν στη διάθεση τους οι υπαίτιοι ήταν η δυνατότητα απομακρυσμένης πρόσβασης στα κέντρα μεταγωγής χωρίς να γίνονται αντιληπτοί. Έτσι μπορούσαν να εγκαταστήσουν κάποια νέα έκδοση του λογισμικού τους ή να τροποποιήσουν τη λίστα όσων ήταν υπό παρακολούθηση.

Σε μια από τις νεότερες εκδόσεις που εγκατέστησαν οι υπεύθυνοι για τις υποκλοπές εμφανίστηκε ένα σφάλμα, το οποίο σχετιζόταν με την παράδοση των γραπτών μηνυμάτων που προορίζονταν για την περιοχή ευθύνης των τεσσάρων τροποποιημένων κέντρων μεταγωγής.

Το γεγονός αυτό ώθησε τους τεχνικούς του δικτύου να ερευνήσουν για ποιο λόγο εμφανίστηκε το πρόβλημα με τα μηνύματα, έτσι εξέτασαν τα συγκεκριμένα κέντρα και έφθασαν στην ανακάλυψη του τροποποιημένου λογισμικού.

Όπως ήταν φυσιολογικό, ακολούθησε η απεγκατάσταση του τροποποιημένου λογισμικού από τα τέσσερα κέντρα που είχαν δεχθεί την επίθεση και οι υποκλοπές σταμάτησαν.

Η υπόθεση όμως περιπλέκεται ακόμη περισσότερο με την απώλεια της ζωής του Κωνσταντίνου Τσαλικίδη, που ήταν τμηματάρχης του σχεδιασμού του δικτύου της εταιρείας στην οποία διενεργούνταν οι υποκλοπές, και που εικάζεται ότι θα μπορούσε να δώσει απαντήσεις σχετικά με το συγκεκριμένο ζήτημα. Όμως ο Τσαλικίδης βρέθηκε απαγχονισμένος στο διαμέρισμα του μία ημέρα μετά το πέρας το υποκλοπών και μία ημέρα πριν από την ενημέρωση παραγόντων της ελληνικής κυβέρνησης σχετικά με την υπόθεση.

Με αυτό τον τρόπο δημιουργείται πλήθος ερωτημάτων. Θα συνεχίζονταν οι παράνομες παρακολουθήσεις αν είχε αποφευχθεί το σφάλμα με τις παραδόσεις των γραπτών μηνυμάτων; Είμαστε σίγουροι ότι οι υποκλοπές έγιναν μόνο στον Ελληνικό χώρο ή ακολουθήθηκε η ίδια πρακτική και σε άλλα μέρη του κόσμου; Τελικώς, μπορούμε να νιώθουμε σιγουριά όταν χρησιμοποιούμε το κινητό μας τηλέφωνο;

Μια θετική απάντηση στο τελευταίο ερώτημα δίνεται μέσω των εφαρμογών που αναπτύχθηκαν για τις ανάγκες της παρούσας πτυχιακής εργασίας. Αφού θα δούμε σε ένα από τα κεφάλαια που ακολουθούν τον τρόπο με τον οποίο διασφαλίζεται η επικοινωνία που διεξάγεται με την ανταλλαγή σύντομων γραπτών μηνυμάτων, ενώ στη συνέχεια παρουσιάζονται εφαρμογές που σχετίζονται με την πιστοποίηση της ταυτότητας ενός χρήστη και τη διατήρηση της ακεραιότητας των επικοινωνιών.

Κεφάλαιο 4 - Κρυπτογραφία

Σε αυτό το κεφάλαιο παρουσιάζονται τα κυριότερα συστατικά στοιχεία της εφαρμογής που επιτρέπει την ασφαλή μετάδοση σύντομων μηνυμάτων.

Για τις ανάγκες της εφαρμογής συνδυάζεται η λειτουργία δύο διαφορετικών ειδών κρυπτογραφικών αλγορίθμων μαζί με μια συνάρτηση κατακερματισμού. Οι αλγόριθμοι κρυπτογράφησης που έχουν επιλεγεί είναι ο συμμετρικός αλγόριθμος AES και ο αλγόριθμος δημοσίου κλειδιού RSA, ενώ η συνάρτηση SHA-256 αποτελεί την επιλογή μας για τη συνάρτηση κατακερματισμού.

Έτσι λοιπόν, στις σελίδες που ακολουθούν αναλύονται οι βασικές αρχές των χρησιμοποιούμενων ειδών κρυπτογραφίας, της κρυπτογραφίας δημοσίου κλειδιού και της συμμετρικής κρυπτογραφίας. Μαζί τους παρουσιάζεται και ο τρόπος με τον οποίο λειτουργούν και εφαρμόζονται οι αλγόριθμοι AES και RSA, που αντιπροσωπεύουν το κάθε είδος κρυπτογραφίας.

Στο τέλος του κεφαλαίου παρουσιάζεται η λογική που διέπει τις συναρτήσεις κατακερματισμού. Ακόμη, αναλύεται ο τρόπος λειτουργίας της συνάρτησης SHA-256 και παρουσιάζεται μια μέθοδος αξιοποίησης της εν λόγω συνάρτησης, όπου το αποτέλεσμα της χρησιμοποιείται για τη δημιουργία συμμετρικών κλειδιών κρυπτογράφησης.

4.1 Κρυπτογραφικά συστήματα

Η διάκριση ανάμεσα στα είδη κρυπτογραφικών συστημάτων βασίζεται σε τρία κριτήρια:

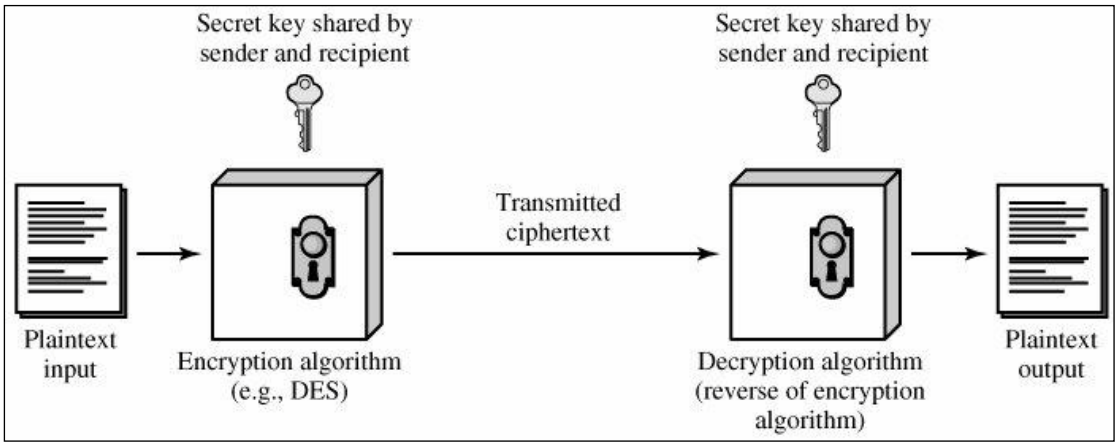
- Στον τύπο των μετασχηματισμών που χρησιμοποιούνται για την κρυπτογράφηση της αρχικής πληροφορίας. Για παράδειγμα, ορισμένα παλαιότερα συστήματα πραγματοποιούσαν την κρυπτογράφηση μέσα από την αντικατάσταση και την αντιμετάθεση κάθε στοιχείου που αποτελούσε την αρχική μορφή της πληροφορίας.
- Στον αριθμό των κλειδιών που χρησιμοποιούνται. Αν και τα δύο άκρα χρησιμοποιούν το ίδιο κλειδί, τότε το κρυπτογραφικό σύστημα αναφέρεται ως συμμετρικό ή συμβατικό ή σύστημα μυστικού κλειδιού. Στην περίπτωση που τα δύο άκρα χρησιμοποιούν διαφορετικά κλειδιά, τότε έχουμε ένα ασύμμετρο σύστημα ή σύστημα δημοσίου κλειδιού.
- Στον τρόπο με τον οποίο γίνεται η επεξεργασία της αρχικής πληροφορίας. Ένας τμηματικός αλγόριθμος κρυπτογράφησης (Block Cipher) επεξεργάζεται κάθε φορά από ένα τμήμα αρχικής πληροφορίας και παράγει το αντίστοιχο τμήμα της κρυπτογραφημένης πληροφορίας. Ενώ ένας αλγόριθμος κρυπτογράφησης ροής (Stream Cipher), επεξεργάζεται ένα προς ένα τα στοιχεία που αποτελούν την εισαγόμενη αρχική πληροφορία και εξάγει πάλι ένα προς ένα τα αντίστοιχα κρυπτογραφημένα στοιχεία.

4.2 Συμμετρική κρυπτογραφία

Η συμμετρική κρυπτογραφία(symmetric cryptography), που είναι επίσης γνωστή και ως συμβατική κρυπτογραφία(conventional cryptography) ή κρυπτογραφία μυστικού κλειδιού(secret key cryptography), αποτελεί το πλέον διαδεδομένο είδος κρυπτογραφίας. Άλλωστε, η συμμετρική κρυπτογραφία αποτελούσε το μοναδικό είδος, μέχρι να εμφανιστεί η κρυπτογραφία δημοσίου κλειδιού τη δεκαετία του 1970.

4.2.1 Βασικές αρχές της συμμετρικής κρυπτογραφίας

Στο σχήμα που ακολουθεί έχουμε ένα σύστημα επικοινωνίας στο οποίο εφαρμόζεται συμμετρική κρυπτογραφία.



Εικόνα 53: Σύστημα στο οποίο εφαρμόζεται συμμετρική κρυπτογράφηση

Το σύστημα περιλαμβάνει πέντε βασικά στοιχεία:

- Την αρχική μορφή του κειμένου ή μηνύματος(Plaintext)
- Το κοινό μυστικό κλειδί(Shared Secret Key)
- Τον αλγόριθμο κρυπτογράφησης(Encryption Algorithm)
- Την κρυπτογραφημένη μορφή του κειμένου ή μηνύματος(Cipher text)
- Τον αλγόριθμο αποκρυπτογράφησης(Decryption Algorithm)

Όταν το κείμενο βρίσκεται στην αρχική του μορφή είναι απόλυτα κατανοητό από τα δύο επικοινωνούντα άκρα. Όμως με αυτή τη μορφή είναι ακατάλληλο προς μετάδοση, αφού κάποιος που παρακολουθεί την κίνηση στο κανάλι επικοινωνιών μπορεί να έχει και αυτός πρόσβαση στο περιεχόμενο του μηνύματος.

Πριν τη μετάδοση πρέπει να τροποποιηθεί το περιεχόμενο του μηνύματος. Ο στόχος είναι να σχηματιστεί κάτι που θα είναι ακατάληπτο για οποιονδήποτε σκοπεύει να παρακολουθήσει την επικοινωνία ανάμεσα στα δύο άκρα.

Για αυτό το σκοπό, το μήνυμα εισάγεται στον αλγόριθμο κρυπτογράφησης. Μέσω του αλγορίθμου πραγματοποιείται μια σειρά από μετασχηματισμούς με τους οποίους αλλάζει η μορφή του μηνύματος, το αποτέλεσμα που λαμβάνεται είναι η κρυπτογραφημένη μορφή του μηνύματος.

Μαζί με το μήνυμα εισάγεται και ένα μυστικό κλειδί. Τα δύο επικοινωνούντα άκρα χρησιμοποιούν το ίδιο κλειδί και η τιμή του κλειδιού είναι γνωστή μόνο σε αυτά.

Το μυστικό κλειδί ελέγχει ως ένα βαθμό την εκτέλεση του αλγόριθμου κρυπτογράφησης, καθώς οι μετασχηματισμοί που πραγματοποιούνται βασίζονται στην τιμή που έχει το μυστικό κλειδί.

Έτσι είναι δυνατόν να προκύπτουν διαφορετικά αποτελέσματα από τον αλγόριθμο ακόμη και αν κρυπτογραφείται το ίδιο κείμενο, αρκεί να αλλάζει το μυστικό κλειδί που εισάγεται κάθε φορά στον αλγόριθμο.

Όταν το κρυπτογραφημένο μήνυμα φτάσει στον προορισμό του, τότε θα ξεκινήσει η διαδικασία ανάκτησης του αρχικού περιεχομένου.

Η αρχική μορφή του μηνύματος λαμβάνεται μετά από την εκτέλεση ενός αλγορίθμου αποκρυπτογράφησης, ο οποίος περιλαμβάνει μια σειρά από μετασχηματισμούς που είναι οι αντίστροφοι από αυτούς που εφαρμόστηκαν κατά την κρυπτογράφηση.

Στον αλγόριθμο αποκρυπτογράφησης εισάγονται το κρυπτογραφημένο μήνυμα μαζί με το κοινό μυστικό κλειδί που χρησιμοποιήθηκε κατά τη διάρκεια της κρυπτογράφησης.

Το αρχικό μήνυμα λαμβάνεται μόνο αν χρησιμοποιηθεί η σωστή τιμή του μυστικού κλειδιού, σε διαφορετική περίπτωση το μήνυμα που προκύπτει παραμένει ακατάληπτο.

Για την ικανοποιητική χρήση συμμετρικής κρυπτογραφίας πρέπει να εκπληρώνονται δύο απαιτήσεις:

- Θα πρέπει να χρησιμοποιείται ένας αρκετά ισχυρός αλγόριθμος κρυπτογράφησης, ο οποίος να εμποδίζει με κάθε τρόπο τον προσδιορισμό των στοιχείων που χρησιμοποιούνται. Ακόμη και στην περίπτωση όπου κάποιος γνωρίζει τη λειτουργία του αλγορίθμου και έχει στην κατοχή του μια σειρά από κρυπτογραφημένα μηνύματα, θα πρέπει να μην μπορεί να υπολογίσει τις τιμές του κλειδιού και του αρχικού μηνύματος που παρήγαγαν το κάθε ένα κρυπτογραφημένο μήνυμα.
- Τα δύο άκρα πρέπει να διατηρούν μυστική την τιμή του κλειδιού που χρησιμοποιούν, όμως ταυτόχρονα πρέπει να βρουν και έναν ασφαλή τρόπο για να αποκτήσουν το κλειδί που πρόκειται να χρησιμοποιηθεί. Αν κάποιος άλλος αποκαλύψει την τιμή του κλειδιού και γνωρίζει τον χρησιμοποιούμενο αλγόριθμο, τότε αποκτά πρόσβαση σε όλη την επικοινωνία που διεξάγεται ανάμεσα στα δύο άκρα.

Συνήθως η πρώτη απαίτηση ικανοποιείται από τις προδιαγραφές που θέτονται κατά τη φάση της σχεδίασης του αλγορίθμου, ο οποίος αργότερα μπορεί να δημοσιευτεί και να είναι ανοιχτός στο κοινό.

Έτσι δίνεται μεγαλύτερη σημασία στη δεύτερη απαίτηση, που αφορά τη διαφύλαξη της τιμής του κλειδιού. Για αυτό και το συγκεκριμένο στοιχείο ανάγεται σε ένα από

τα κυρίαρχα ζητήματα που παρουσιάζουν τα συστήματα που χρησιμοποιούν συμμετρική κρυπτογράφηση.

4.2.2 Η επιλογή του αλγορίθμου για το AES

Ο αλγόριθμος που θα αποτελούσε το AES(Advanced Encryption Standard) δημοσιεύτηκε το 2001 από το αμερικανικό ίδρυμα προτύπων και τεχνολογίας NIST(National Institute for Standards and Technology).

Ο στόχος του AES ήταν να αντικαταστήσει το DES(Data Encryption Standard), το οποίο ήταν το εγκεκριμένο πρότυπο κρυπτογράφησης για ένα μεγάλο εύρος εφαρμογών.

Το DES είχε υιοθετηθεί επισήμως το 1977, με τη θεμελίωση των προδιαγραφών FIPS PUB 46 από το NIST. Με το DES πραγματοποιείται η συμμετρική κωδικοποίηση τμημάτων πληροφορίας με μήκος 64 bit. Για να πραγματοποιηθεί η εν λόγω διαδικασία απαιτείται η εισαγωγή κλειδιού μήκους 56 bit.

Όμως από την πρώτη στιγμή γίνονταν συζητήσεις σχετικά με τη λειτουργία του DES και την ασφάλεια που παρέχει.

Το κυρίαρχο ζήτημα σχετικά με την ασφάλεια του DES εντοπίζεται στο μήκος του κλειδιού κρυπτογράφησης που χρησιμοποιείται. Αφού από τα πρώτα χρόνια, έστω και σε θεωρητικό επίπεδο, ήταν εφικτή η εύρεση του κλειδιού μέσω της εξαντλητικής δοκιμής όλων των πιθανών κλειδιών(brute-force attack).

Τη δεκαετία του 1990 οι θεωρητικές επιθέσεις μετουσιώθηκαν σε πρακτικές, όπου με τη χρησιμοποίηση ειδικού hardware, του "DES Cracker" που κατασκευάστηκε από το ίδρυμα Electronic Frontier Foundation(EFF), κατέστη δυνατή η εύρεση του κλειδιού σε ένα διάστημα λιγότερο των 48 ωρών.

Μια προσωρινή λύση στο πρόβλημα δόθηκε με την χρησιμοποίηση του τριπλού DES(Triple DES), επίσης γνωστό και ως 3DES ή TDES.

Για τη λειτουργία του 3DES χρειάζεται ένα κλειδί μήκους 168 δυαδικών ψηφίων. Κάθε τμήμα της αρχικής πληροφορίας υπόκειται σε τρεις διαδοχικές κρυπτογραφήσεις μέσω του DES.

Με αυτό τον τρόπο καλύπτεται το τρωτό σημείο που υπάρχει με το μήκος του κλειδιού, όμως παράλληλα υπάρχει κατακόρυφη πτώση της αποδοτικότητας, αφού για την κρυπτογράφηση ενός τμήματος πληροφορίας απαιτείται ο τριπλάσιος επεξεργαστικός φόρτος.

Οι λόγοι που σχετίζονται με την ασφάλεια και την αποδοτικότητα του DES ώθησαν το NIST στην αναζήτηση του διαδόχου του. Γι' αυτό το 1997 τέθηκαν οι απαιτήσεις για το AES και ξεκίνησε η διαδικασία αξιολόγησης των υποψήφιων αλγορίθμων που προορίζονταν για χρήση σε αυτό.

Σύμφωνα με τις απαιτήσεις, ο αλγόριθμος που θα χρησιμοποιείται στο AES θα πρέπει:

- Να παρέχει αυξημένη ασφάλεια σε σχέση με τον προκάτοχο του
- Να έχει υψηλή απόδοση χωρίς να απαιτεί αρκετή επεξεργαστική ισχύ
- Να εφαρμόζεται εύκολα και με χαμηλό κόστος
- Να έχει ευελιξία υλοποίησης, τόσο σε υλικό όσο και σε λογισμικό
- Να πραγματοποιεί συμμετρική κρυπτογράφηση σε block πληροφορίας, με κάθε block να έχει μέγεθος 128 bit
- Να υποστηρίζει τρεις διαφορετικούς τρόπους λειτουργίας, με αντίστοιχα κλειδιά με μήκος 128,192 και 256 bit

Στην πρώτη φάση της αξιολόγησης πέρασαν 15 υποψήφιοι αλγόριθμοι, στη δεύτερη φάση πέρασαν οι 5 από αυτούς, μέχρι να φτάσουμε στο Νοέμβριο του 2001, όπου με την έκδοση των προδιαγραφών FIPS PUB 197 αποφασίστηκε η υιοθέτηση του αλγορίθμου Rijndael ως του καταλληλότερου για χρήση στο AES.

4.2.3 Ο τρόπος λειτουργίας του AES

Η αρχική πρόταση του αλγορίθμου Rijndael αφορούσε τη χρησιμοποίηση μεταβλητού μεγέθους block και υποστήριζε τη λειτουργία με κλειδί μήκους 128,192 και 256 bit. Οι απαιτήσεις για το AES περιόρισαν το μέγεθος του block στα 128 bit. Όμως τα διαφορετικά μεγέθη κλειδιών επηρεάζουν ένα πλήθος παραμέτρων και μεταβάλλουν τον τρόπο λειτουργίας του αλγορίθμου.

Έκδοση του αλγορίθμου	Μέγεθος κλειδιού (σε λέξεις/ bytes/ bits)	Μέγεθος block (σε λέξεις/ bytes/ bits)	Αριθμός γύρων με μετασχηματισμούς	Μέγεθος κλειδιού μετά την επέκταση (σε λέξεις/ bytes/ bits)
AES-128	4/16/128	4/16/128	10	44/176/1408
AES-192	6/24/192	4/16/128	12	52/208/1664
AES-256	8/32/256	4/16/128	14	60/240/1920

Πίνακας 4: Η σχέση ανάμεσα στο μέγεθος του κλειδιού και τις παραμέτρους του αλγορίθμου

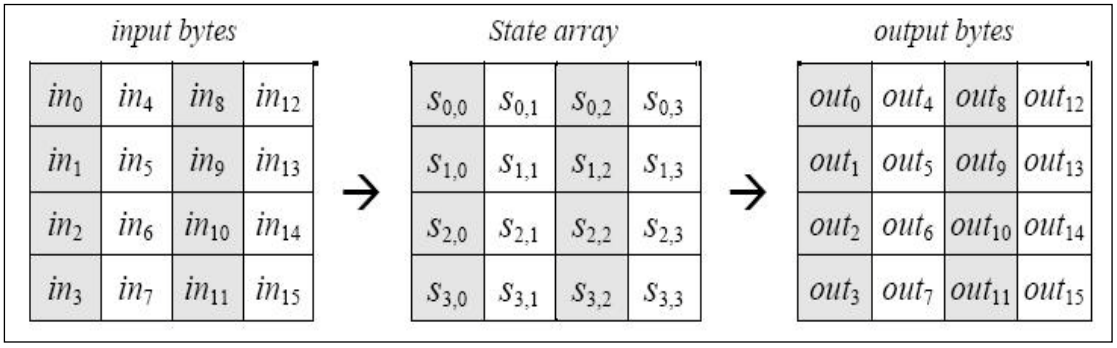
Στις σελίδες που ακολουθούν θα περιγράψουμε τη λειτουργία του αλγορίθμου όταν χρησιμοποιείται κλειδί με μήκος 128 δυαδικών ψηφίων. Σε αυτή την περίπτωση πραγματοποιούνται δέκα γύροι με μετασχηματισμούς οι οποίοι εφαρμόζονται στο block με την πληροφορία που εισάγεται στον αλγόριθμο.

Το block έχει μέγεθος 128 δυαδικών ψηφίων και στις προδιαγραφές του αλγορίθμου απεικονίζεται με τη μορφή ενός τετραγωνικού πίνακα που περιέχει 16 bytes.

Το block που εισάγεται στον αλγόριθμο αντιγράφεται σε έναν πίνακα που ονομάζεται State, ο οποίος περιέχει την ενδιάμεση κατάσταση στην οποία βρίσκεται η πληροφορία καθώς υπόκειται στους προβλεπόμενους μετασχηματισμούς.

Με την ολοκλήρωση των μετασχηματισμών έχουμε την αντιγραφή της τελικής τιμής του State σε έναν πίνακα εξόδου. Να τονίσουμε ότι η διάταξη των bytes που περιέχονται σε κάθε πίνακα γίνεται ανά στήλη, δηλαδή, τα πρώτα τέσσερα bytes καταλαμβάνουν την πρώτη στήλη του πίνακα, η επόμενη τετράδα από bytes αντιστοιχεί στη δεύτερη στήλη και ούτω καθ' εξής.

Στο σχήμα που ακολουθεί παρουσιάζεται η μορφή που έχει ο πίνακας εισόδου, ο πίνακας State και ο πίνακας εξόδου.

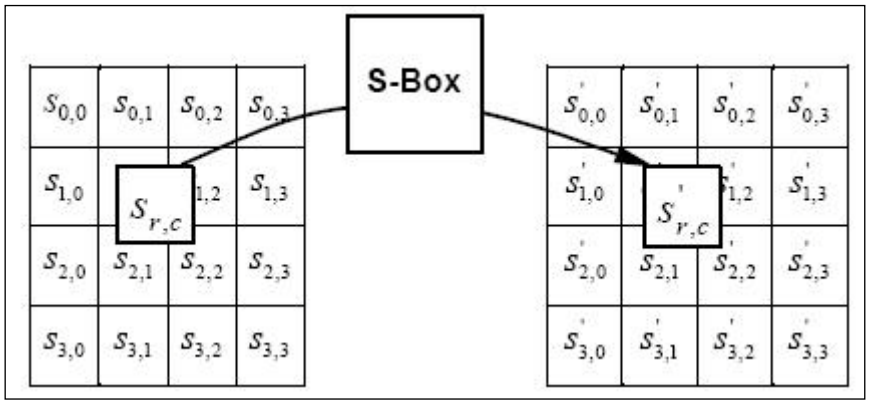


Εικόνα 54: Η μορφή των πινάκων εισόδου, εσωτερικής κατάστασης και εξόδου

Κατά τη διάρκεια εκτέλεσης του αλγορίθμου έχουμε τέσσερις τύπους μετασχηματισμών οι οποίοι εφαρμόζονται στον πίνακα State:

- **Αντικατάσταση των bytes (SubBytes)**

Ο συγκεκριμένος μετασχηματισμός αναφέρεται στις προδιαγραφές του αλγορίθμου ως SubBytes, είναι μη γραμμικός, πλήρως αντιστρέψιμος, και εφαρμόζεται ανεξάρτητα σε κάθε byte του πίνακα State. Στην τιμή που έχει το byte αντιστοιχεί μια τιμή που προκύπτει από έναν πίνακα αντικατάστασης(S-Box).



Εικόνα 55: Εφαρμογή S-Box

Για παράδειγμα, αν ο πίνακας του State περιέχει ένα byte με την τιμή `0x95`, τότε το συγκεκριμένο byte θα λάβει την τιμή που βρίσκεται στο σημείο τομής της γραμμής '`9`' και της στήλης '`5`'. Στην προκειμένη περίπτωση η τιμή `0x95` μετατρέπεται σε `0x2A`.

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Πίνακας 5: Οι τιμές του πίνακα S-Box

Ο πίνακας του S-Box έχει διαστάσεις 16 επί 16 και κατασκευάζεται ακολουθώντας μια συγκεκριμένη λογική.

Σε πρώτη φάση, ο πίνακας γεμίζει με τις τιμές των bytes που αντιστοιχούν στα στοιχεία που απαρτίζουν το πεπερασμένο πεδίο $GF(2^8)$, το οποίο είναι αυτό που χρησιμοποιείται από το AES. Τα bytes εισάγονται με αύξουσα σειρά, έτσι η πρώτη γραμμή περιλαμβάνει τις δεκαεξαδικές τιμές από `0x00` ως και `0x0F`, η δεύτερη γραμμή ξεκινά από το στοιχείο `0x10` και τελειώνει στο `0x1F`, και συνεχίζουμε μέχρι την τελευταία γραμμή που περιέχει τα στοιχεία με τιμές από `0xF0` μέχρι και `0xFF`.

Στη συνέχεια πραγματοποιείται η αντιστοίχηση κάθε στοιχείου με τον πολλαπλασιαστικό αντίστροφο που ορίζεται στο πεδίο $GF(2^8)$. Σε κάθε στοιχείο του πεδίου αντιστοιχεί ένας μοναδικός αντίστροφος, εκτός από την τιμή `0x00` όπου αντιστοιχίζεται στον εαυτό της.

Οι τελικές τιμές του πίνακα προκύπτουν από το μετασχηματισμό που ακολουθεί:

$$b_i^{'} = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

Εικόνα 56: Ο μετασχηματισμός μέσω του οποίου προκύπτει η τιμή του κάθε bit του S-Box

Ο μετασχηματισμός εφαρμόζεται ξεχωριστά σε κάθε ένα από τα δυαδικά ψηφία του πολλαπλασιαστικού αντιστρόφου, έτσι, για τον υπολογισμό ενός byte του S-Box χρειάζεται να εκτελεστεί 8 φορές ο παραπάνω μετασχηματισμός.

Το *i* λειτουργεί σαν μετρητής, οπότε λαμβάνει τιμές από `0` ως `7`, ταυτόχρονα όμως καθορίζει ποια δυαδικά ψηφία συμμετέχουν στο μετασχηματισμό και που πρόκειται να αποθηκευτεί το αποτέλεσμα του μετασχηματισμού.

Το b_i αντιπροσωπεύει το bit που ανήκει στην τιμή του πολλαπλασιαστικού αντιστρόφου και βρίσκεται στη θέση που ορίζει η τιμή του *i*.

Το byte *c* περιέχει τη δεκαεξαδική τιμή `0x63`, έτσι το c_i ορίζει ποιο δυαδικό ψηφίο επιλέγεται από το byte *c*.

Το αποτέλεσμα του μετασχηματισμού, που προκύπτει από μια σειρά από XOR ανάμεσα στα bits που συμμετέχουν, αποθηκεύεται στο b_i'.

Ας δούμε όμως ένα παράδειγμα του τρόπου με τον οποίο δημιουργείται ο πίνακας αντικατάστασης. Ο πίνακας έχει γεμίσει με τις τιμές των bytes που ανήκουν στο πεδίο GF(2^8). Στο σημείο τομής της γραμμής '`9`' και της στήλης '`5`' βρίσκεται το byte με την τιμή `0x95`.

Ο πολλαπλασιαστικός αντίστροφος του συγκεκριμένου byte έχει την τιμή `0x8A`. Μετά από την ξεχωριστή εφαρμογή του μετασχηματισμού πάνω στα 8 δυαδικά ψηφία που αποτελούν το byte `0x95` εξάγονται τα δυαδικά ψηφία $(00101010)_2$, τα οποία ισοδυναμούν με την τελική δεκαεξαδική τιμή `0x2A`. Η τιμή αυτή αποθηκεύεται στη θέση που βρισκόταν το byte `0x95`.

Με αυτό τον τρόπο λειτουργεί ο μετασχηματισμός όταν θέλουμε να κρυπτογραφήσουμε την αρχική πληροφορία που εισάγεται στον αλγόριθμο. Στην περίπτωση που επιθυμούμε να ανακτήσουμε την αρχική πληροφορία από ένα κρυπτογράφημα, τότε έχουμε την εμπλοκή του αντίστροφου μετασχηματισμού αντικατάστασης bytes.

Η αντίστροφη αντικατάσταση των bytes, μέσω του μετασχηματισμού InvSubBytes, πραγματοποιείται με τη βοήθεια ενός πίνακα που περιέχει τις αντίστροφες τιμές από αυτές που ορίζει ο πίνακας του S-Box. Έτσι για παράδειγμα, αν ένα byte από την κρυπτογραφημένη πληροφορία περιέχει την τιμή `0x2A`, τότε το συγκεκριμένο byte λαμβάνει την τιμή `0x95`, που βρίσκεται στην τομή της γραμμής '`2`' και της στήλης '`A`'.

Ο αντίστροφος πίνακας του S-Box έχει τις ακόλουθες τιμές:

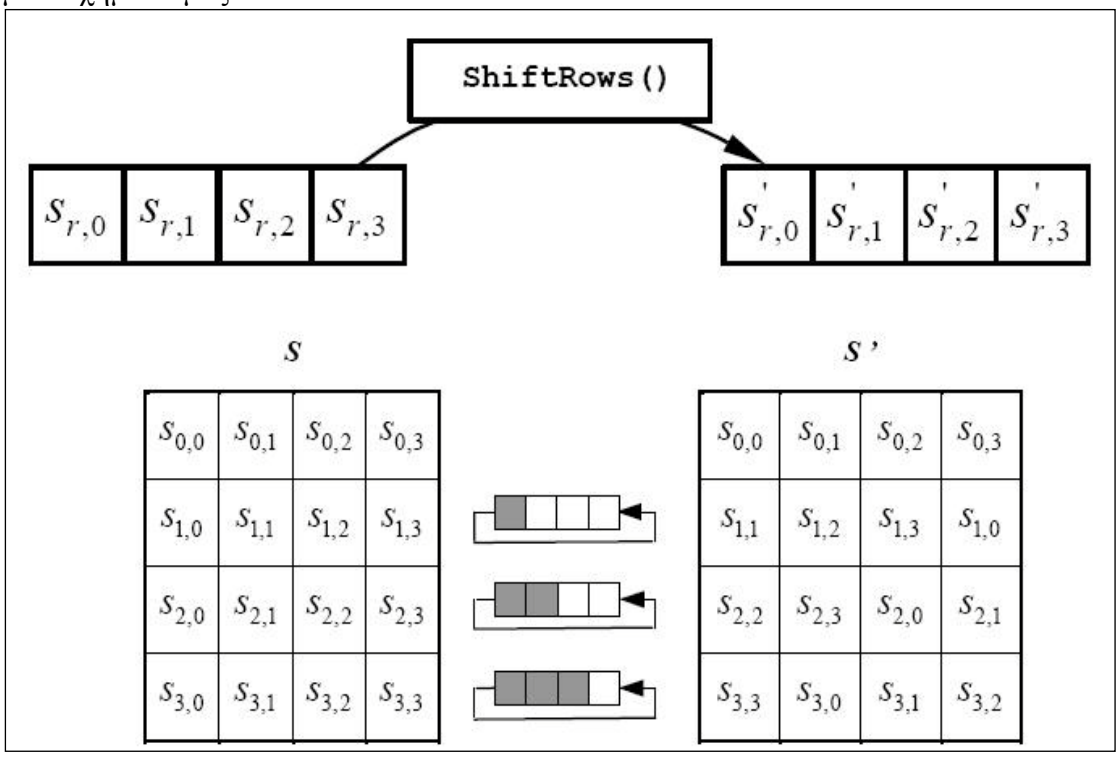
		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Πίνακας 6: Οι τιμές του αντιστρόφου του πίνακα S-Box

- **Ολίσθηση των γραμμών (ShiftRows)**

Με τον μετασχηματισμό ShiftRows πραγματοποιείται η κυκλική ολίσθηση των bytes που βρίσκονται στις τρεις τελευταίες γραμμές του State. Η πρώτη γραμμή διατηρείται ανέπαφη, ενώ στις υπόλοιπες τρεις γραμμές ολισθαίνει διαφορετικός αριθμός από bytes. Στη δεύτερη γραμμή έχουμε την ολίσθηση του πρώτου byte μια θέση προς τα αριστερά, στην τρίτη γραμμή ολισθαίνουν τα δύο πρώτα bytes προς τα αριστερά, και στην τέταρτη γραμμή ολισθαίνουν τα τρία πρώτα bytes προς τα αριστερά.

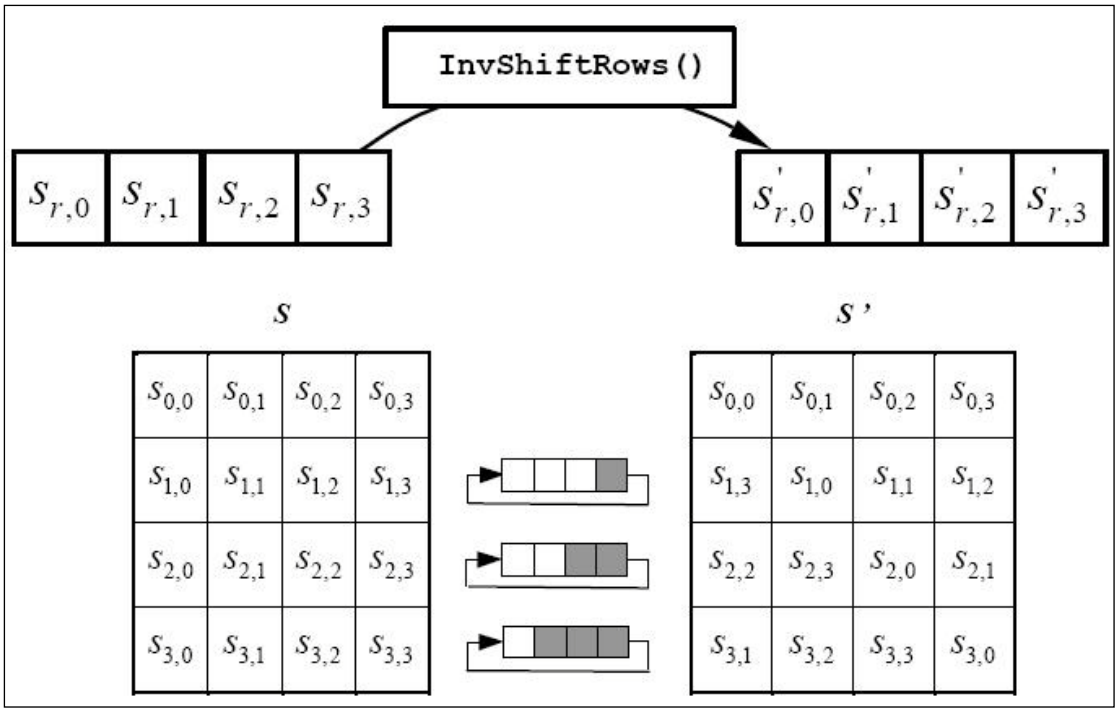
Στο σχήμα που ακολουθεί εμφανίζεται ο τρόπος με τον οποίο εφαρμόζεται ο μετασχηματισμός ShifRows πάνω στον πίνακα State.



Εικόνα 57: Εφαρμογή του μετασχηματισμού ShiftRows

Ο μετασχηματισμός ShiftRows είναι αντιστρέψιμος. Μέσω του μετασχηματισμού InvShiftRows πραγματοποιείται η ολίσθηση των bytes του State με φορά αντίθετη από αυτή που χρησιμοποιήθηκε κατά το μετασχηματισμό ShiftRows. Η πρώτη γραμμή διατηρείται ως έχει. Το τελευταίο byte της δεύτερης γραμμής ολισθαίνει μια θέση προς τα δεξιά και τοποθετείται στην αρχή της γραμμής του. Στην τρίτη γραμμή έχουμε τα δύο τελευταία bytes να ολισθαίνουν προς τα δεξιά και να καταλαμβάνουν τις δύο πρώτες θέσεις της γραμμής τους. Στην τέταρτη γραμμή τα τρία τελευταία bytes ολισθαίνουν και οδηγούνται στις τρεις πρώτες θέσεις.

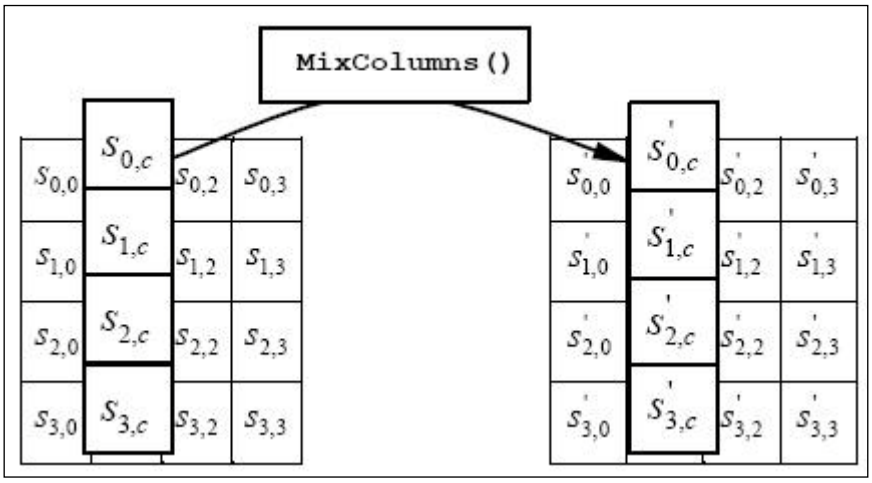
Η εφαρμογή του μετασχηματισμού InvShiftRows απεικονίζεται στο σχήμα που ακολουθεί.



Εικόνα 58: Εφαρμογή του μετασχηματισμού InvShiftRows

- **Ανάμιξη των στηλών (MixColumns)**

Ο μετασχηματισμός MixColumns εφαρμόζεται ξεχωριστά σε κάθε μια από τις τέσσερις στήλες του πίνακα State.



Εικόνα 59: Εφαρμογή του μετασχηματισμού MixColumns

Οι στήλες του State θεωρούνται ως πολυώνυμα του πεδίου GF(2^8). Οι τιμές των bytes που αποτελούν κάθε στήλη αντικαθίστανται από τις τιμές που προκύπτουν μετά από τον πολλαπλασιασμό της στήλης με ένα σταθερό πολυώνυμο ακολουθούμενο από την πράξη modulo με το πολυώνυμο (x^4+1).

Το σταθερό πολυώνυμο που χρησιμοποιείται στο μετασχηματισμό MixColumns κατά την κρυπτογράφηση έχει τη μορφή:

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$$

Εικόνα 60: Το σταθερό πολυώνυμο που χρησιμοποιείται κατά το μετασχηματισμό MixColumns

Ο μετασχηματισμός μπορεί να παρουσιαστεί ισοδύναμα και σαν πολλαπλασιασμός πινάκων στο πεπερασμένο πεδίο $GF(2^8)$. Τα bytes κάθε στήλης του State πολλαπλασιάζονται με τον πίνακα που ισοδυναμεί με το σταθερό πολυώνυμο, στη συνέχεια τα bytes που προκύπτουν από τον πολλαπλασιασμό παίρνουν τη θέση των bytes που υπήρχαν στη στήλη του State.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Πίνακας 7: Ο πολλαπλασιασμός των πινάκων που πραγματοποιείται κατά τον μετασχηματισμό MixColumns

Ο πολλαπλασιασμός ενός byte με την τιμή `0x01` δεν αλλάζει κάτι, αφού η τιμή `0x01` είναι το ουδέτερο στοιχείο της πράξης του πολλαπλασιασμού. Έτσι, το αποτέλεσμα του πολλαπλασιασμού ανάμεσα στον πίνακα του πολυωνύμου και τη στήλη του State ισοδυναμεί με τα ακόλουθα:

$$\begin{aligned} s'_{0,c} &= (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\ s'_{3,c} &= (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) \end{aligned}$$

Εικόνα 61: Το αποτέλεσμα του πολλαπλασιασμού των πινάκων κατά το μετασχηματισμό MixColumns

Ο μετασχηματισμός MixColumns είναι και αυτός αντιστρέψιμος, με την αντίστροφη μορφή του μετασχηματισμού να αναφέρεται στις προδιαγραφές του αλγορίθμου με το όνομα InvMixColumns.

Ο τρόπος με τον οποίο εφαρμόζεται ο μετασχηματισμός InvMixColumns είναι παρόμοιος με αυτόν του MixColumns. Η διαφορά βρίσκεται στο σταθερό πολυώνυμο που χρησιμοποιείται, όπου στην προκειμένη περίπτωση είναι το αντίστροφο του σταθερού πολυωνύμου *a(x)* που χρησιμοποιείται στο μετασχηματισμό MixColumns. Το πολυώνυμο που χρησιμοποιείται είναι το ακόλουθο:

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

Εικόνα 62: Το σταθερό πολυώνυμο που χρησιμοποιείται στο μετασχηματισμό InvMixColumns

Η μορφή που έχει ο μετασχηματισμός όταν περιγράφεται σαν πολλαπλασιασμός πινάκων είναι αυτή που ακολουθεί:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Πίνακας 8: Ο πολλαπλασιασμός των πινάκων που πραγματοποιείται κατά τον μετασχηματισμό InvMixColumns

Τα bytes της στήλης State αντικαθίστανται από το αποτέλεσμα των πράξεων που ακολουθούν:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$

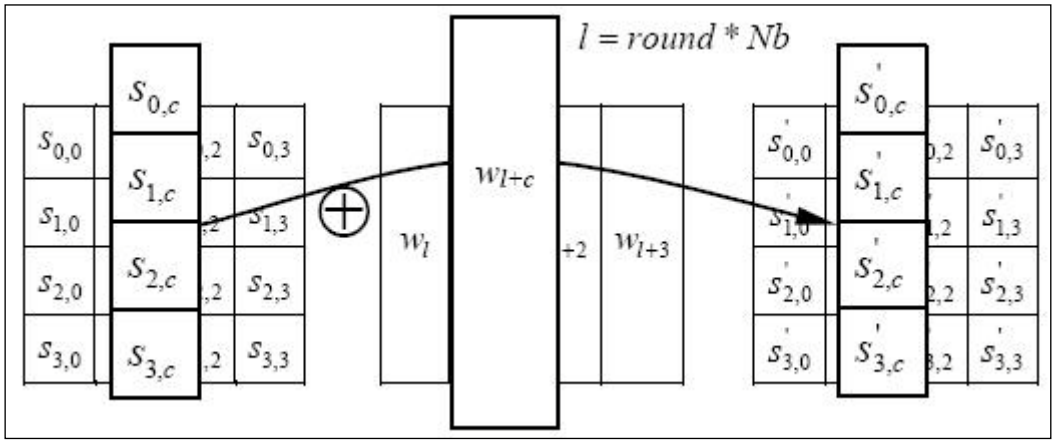
$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

Εικόνα 63: Το αποτέλεσμα του πολλαπλασιασμού πινάκων κατά το μετασχηματισμό InvMixColumns

- **Προσθήκη υπό-κλειδιού (AddRoundKey)**

Με το μετασχηματισμό AddRoundKey προστίθεται ο πίνακας του State με ένα block μεγέθους 16 bytes που περιέχει το υπό-κλειδί Round Key. Η πρόσθεση πραγματοποιείται με την εφαρμογή της πράξης XOR σε κάθε μία από τις 4 στήλες των δύο πινάκων. Οι στήλες αποτελούνται από 4 bytes, τα οποία αναφέρονται και ως λέξη(word).



Εικόνα 64: Η εφαρμογή του μετασχηματισμού AddRoundKey

Ο μετασχηματισμός AddRoundKey έχει την ιδιαιτερότητα να εκτελείται μια φορά παραπάνω από ότι οι υπόλοιποι μετασχηματισμοί που εμπλέκονται στον αλγόριθμο AES. Η μοναδική περίπτωση όπου ο μετασχηματισμός AddRoundkey εφαρμόζεται μόνος του είναι πριν να ξεκινήσει ο πρώτος γύρος κρυπτογράφησης ή αποκρυπτογράφησης.

Σε εκείνο το σημείο, δηλαδή πριν από την εκκίνηση των μετασχηματισμών, εκτελείται μια διαδικασία με την οποία επεκτείνεται το αρχικό κλειδί κρυπτογράφησης. Η εν λόγω διαδικασία αναφέρεται στις προδιαγραφές του AES με το όνομα Key Expansion Schedule ή πιο σύντομα Key Schedule. Χάρη σε αυτή διαδικασία παράγεται ένα διευρυμένο κλειδί, το οποίο αποθηκεύεται σε έναν πίνακα που ονομάζεται *w* και αποτελείται από ένα ευρύ πλήθος από λέξεις.

Το πλήθος των λέξεων που περιέχονται στο διευρυμένο κλειδί εξαρτάται άμεσα από το πλήθος των μετασχηματισμών που απαιτούνται για να ολοκληρωθεί η εκτέλεση του αλγορίθμου. Ο γενικός τύπος που δίνει το πλήθος των λέξεων είναι `4*(Πλήθος_Μετασχηματισμών + 1)`, οπότε για την περίπτωση του AES-128 που έχουμε 10 γύρους μετασχηματισμών το διευρυμένο κλειδί έχει μήκος 44 λέξεις.

Έτσι το Round Key μπορεί να έχει διαφορετική τιμή σε κάθε γύρο μετασχηματισμών, καθώς αντλείται ένα διαφορετικό τμήμα μήκους 4 λέξεων από το διευρυμένο κλειδί κρυπτογράφησης, το οποίο χρησιμοποιείται για να πραγματοποιείται ο μετασχηματισμός AddRoundKey.

Η διαδικασία επέκτασης του κλειδιού κρυπτογράφησης ξεκινά με την αντιγραφή των λέξεων οι οποίες το αποτελούν στις αντίστοιχες πρώτες λέξεις του διευρυμένου κλειδιού. Για παράδειγμα, ένα κλειδί των 128 bit αποτελείται από 4 λέξεις των 32 bit, οπότε οι πρώτες 4 από τις 44 λέξεις που αποτελούν το διευρυμένο κλειδί λαμβάνουν αυτές τις τιμές.

Οι υπόλοιπες λέξεις του διευρυμένου κλειδιού προκύπτουν από την εφαρμογή της πράξης XOR ανάμεσα σε δύο προηγούμενες λέξεις. Η μία από αυτές βρίσκεται στην αμέσως προηγούμενη θέση από αυτή της λέξης που πρόκειται να υπολογιστεί, ενώ η δεύτερη λέξη βρίσκεται πίσω για έναν αριθμό από λέξεις που είναι ίσος με το πλήθος των λέξεων του αρχικού κλειδιού κρυπτογράφησης. Ας δούμε σαν παράδειγμα πως υπολογίζεται η λέξη που βρίσκεται στη θέση 5 όταν το κλειδί κρυπτογράφησης έχει 128 bit, τότε από στον πίνακα *w* έχουμε $w[5]=w[5-1] \oplus w[5-4]$ που ισοδυναμεί με $w[5]=w[4] \oplus w[1]$.

Όμως οι λέξεις που βρίσκονται σε θέσεις που είναι ακέραια πολλαπλάσια του πλήθους των λέξεων του αρχικού κλειδιού κρυπτογράφησης υπολογίζονται με διαφορετικό τρόπο, αφού εμπλέκονται τρεις επιπλέον μετασχηματισμοί οι οποίοι εφαρμόζονται στη λέξη που βρίσκεται στην προηγούμενη θέση. Οι συγκεκριμένοι μετασχηματισμοί αναφέρονται στις προδιαγραφές ως:

- RotWord
- SubWord
- Rcon

Με το μετασχηματισμό RotWord πραγματοποιείται μια κυκλική ολίσθηση ανάμεσα στα bytes της λέξης, έτσι το πρώτο byte μετακινείται μια θέση προς τα αριστερά με αποτέλεσμα να εισαχθεί στο τέλος της λέξης.

Στη συνέχεια εφαρμόζεται ο μετασχηματισμός SubWord, όπου τα bytes της λέξης αντικαθίστανται από τις τιμές που αντιστοιχούν από το S-Box.

Μετά την εφαρμογή των μετασχηματισμών RotWord και SubWord έρχεται η ώρα να εφαρμοστεί ο μετασχηματισμός Rcon. Σε αυτή τη φάση πραγματοποιείται μια πράξη XOR ανάμεσα σε μια προκαθορισμένη λέξη και στη λέξη που έχει προκύψει μετά τους προηγούμενους μετασχηματισμούς.

Η προκαθορισμένη λέξη αποτελείται από τα bytes x^{i-1} 00 00 00, με το byte x^{i-1} να αντιπροσωπεύονται οι δυνάμεις στις οποίες υψώνεται το byte 02 στο πεδίο GF(2^8). Οι δυνάμεις που χρησιμοποιούνται βρίσκονται στον πίνακα που ακολουθεί.

i	1	2	3	4	5	6	7	8	9	10
02^{i-1}	01	02	04	08	10	20	40	80	1B	36

Πίνακας 9: Οι δυνάμεις του byte 02 στο πεδίο GF(2^8)

Σε αυτό το σημείο θα παρουσιάσουμε σαν παράδειγμα ένα μέρος από τη διαδικασία του Key Expansion Schedule όταν χρησιμοποιείται ένα κλειδί μήκους 128 bit. Το κλειδί που θα χρησιμοποιήσουμε έχει την τιμή:

2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C,

το οποίο χωρίζεται σε 4 λέξεις που αντιγράφονται στις αντίστοιχες θέσεις του πίνακα *w*.

Οπότε έχουμε:

1. *w*[0]=2B7E1516
2. *w*[1]=28AED2A6
3. *w*[2]=ABF71588
4. *w*[3]=09CF4F3C

Οι λέξεις *w*[0], *w*[1], *w*[2] και *w*[3] σχηματίζουν την τιμή που θα έχει το πρώτο Round Key που θα χρησιμοποιηθεί στο μετασχηματισμό AddRoundKey.

Οι υπόλοιπες λέξεις, από τη θέση 4 μέχρι τη θέση 43, υπολογίζονται με τον τρόπο που αναφέρθηκε νωρίτερα. Να θυμίσουμε ότι το κλειδί κρυπτογράφησης των 128 bit αποτελείται από 4 λέξεις, οπότε στις λέξεις του διευρυμένου κλειδιού που βρίσκονται σε θέσεις πολλαπλάσιες του 4 εφαρμόζονται οι μετασχηματισμοί RotWord, SubWord και Rcon. Το πλήθος των λέξεων του αρχικού κλειδιού κρυπτογράφησης αντιπροσωπεύεται από τη μεταβλητή Nk.

Στον πίνακα που ακολουθεί εμφανίζεται ένα τμήμα από τις λέξεις που περιέχονται στο διευρυμένο κλειδί, πιο συγκεκριμένα οι λέξεις από τη θέση 4 ως 15. Η θέση που έχει η κάθε λέξη στον πίνακα *w* αντιπροσωπεύεται με τη μεταβλητή i.

i	w[i-1]	RotWord	SubWord	Rcon[i/Nk]	w[i-1] XOR Rcon[i/Nk]	w[i-Nk]	w[i] = w[i-1] XOR

							w[i-Nk]
4	09CF4F3C	CF4F3C09	8A84EB01	01000000	8B84EB01	2B7E1516	A0FAFE17
5	A0FAFE17					28AED2A6	88542CB1
6	88542CB1					ABF71588	23A33939
7	23A33939					09CF4F3C	2A6C7605
8	2A6C7605	6C76052A	50386BE5	02000000	52386BE5	A0FAFE17	F2C295F2
9	F2C295F2					88542CB1	7A96B943
10	7A96B943					23A33939	5935807A
11	5935807A					2A6C7605	7359F6F7
12	7359F6F7	59F6F773	CB42D28F	04000000	CF42D28F	F2C295F2	3D80477D
13	3D80477D					7A96B943	4716FE3E
14	4716FE3E					5935807A	1E237E44
15	1E237E44					7359F6F7	6D7A883D

Πίνακας 10: Ένα απόσπασμα από τη διαδικασία Key Expansion Schedule

Το τμήμα των λέξεων που περιέχεται στον παραπάνω πίνακα χωρίζεται σε τετράδες, από 4 ως 7, 8 ως 11, και 12 ως 15. Με αυτό τον τρόπο σχηματίζονται αντίστοιχες τιμές Round Key που είναι έτοιμες για να χρησιμοποιηθούν στο μετασχηματισμό AddRoundKey.

Όταν ο αλγόριθμος χρησιμοποιείται για την κρυπτογράφηση ενός block πληροφορίας τότε κάθε Round Key που χρησιμοποιείται στο μετασχηματισμό AddRoundKey ακολουθεί την ορθή φορά, δηλαδή πρώτα χρησιμοποιείται το Round Key που έχει τις λέξεις $w[0]$ ως $w[3]$, στη συνέχεια χρησιμοποιείται το Round Key με τις λέξεις από $w[4]$ ως $w[7]$, μέχρι να φτάσουμε στο τελευταίο Round Key που περιλαμβάνει τις λέξεις από $w[40]$ ως $w[43]$.

Για την αποκρυπτογράφηση ενός block πληροφορίας χρησιμοποιείται ο αντίστροφος του μετασχηματισμού AddRoundKey. Όμως ο αντίστροφος δεν είναι κάποιος άλλος πέρα από τον ίδιο το μετασχηματισμό AddRoundKey. Αυτό οφείλεται στο γεγονός ότι ο συγκεκριμένος μετασχηματισμός βασίζεται στην πράξη XOR, η οποία είναι εκ των πραγμάτων αντιστρέψιμη.

Ακόμη, η διαδικασία Key Expansion Schedule εκτελείται με τον ίδιο τρόπο, με τη διαφορά ότι κάθε Round Key χρησιμοποιείται με αντίστροφη φορά. Δηλαδή, το πρώτο Round Key που χρησιμοποιείται περιλαμβάνει τις λέξεις από $w[40]$ ως $w[43]$, το δεύτερο έχει τις λέξεις από $w[36]$ ως $w[39]$, και ούτω καθεξής, μέχρι να φτάσουμε στο Round Key με τις λέξεις από $w[0]$ ως $w[3]$.

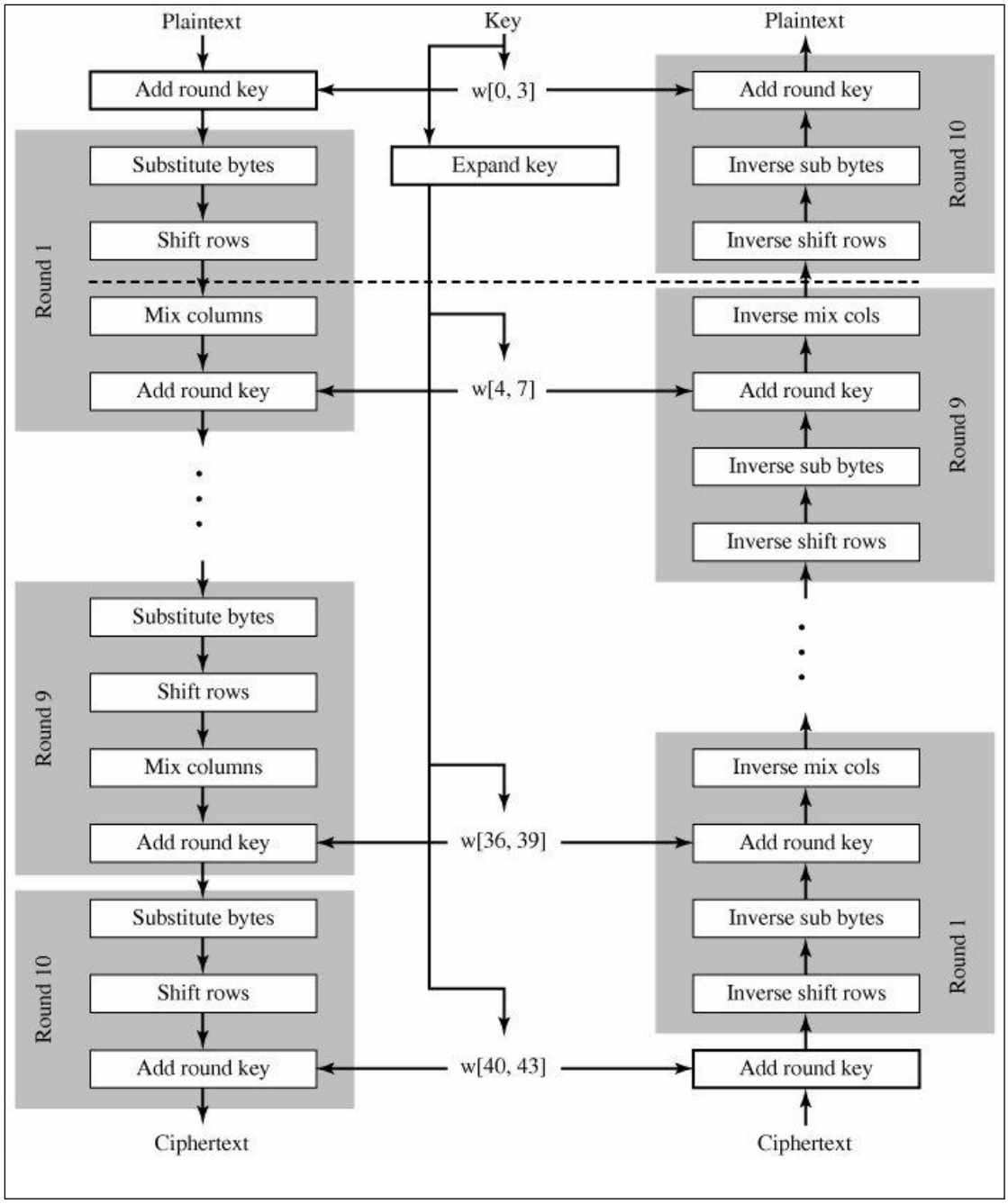
Στο σχήμα που ακολουθεί απεικονίζεται συνοπτικά ένας πλήρης κύκλος εκτέλεσης του αλγορίθμου AES, με την κρυπτογράφηση και την αποκρυπτογράφηση να έχουν τοποθετηθεί με αντίθετη κάθετη κατεύθυνση.

Στο κέντρο του σχήματος βρίσκεται η διαδικασία για την επέκταση του κλειδιού κρυπτογράφησης(Key Expansion Schedule).

Στην αριστερή στήλη έχουμε τη διαδικασία κρυπτογράφησης της αρχικής πληροφορίας που εισάγεται στον αλγόριθμο. Παρατηρούμε την εφαρμογή του AddRoundKey μια φορά πριν να αρχίσουν οι 10 γύροι των μετασχηματισμών από τους οποίους προκύπτει η κρυπτογραφημένη πληροφορία. Επίσης, γίνεται εμφανές

ότι ο μετασχηματισμός MixColumns εφαρμόζεται μόνο στους 9 πρώτους γύρους μετασχηματισμών.

Στη δεξιά στήλη παρουσιάζεται η διαδικασία αποκρυπτογράφησης. Παρατηρούμε ότι πριν από την εκκίνηση των 10 γύρων με αντίστροφους μετασχηματισμούς εφαρμόζεται ο μετασχηματισμός AddRoundKey χρησιμοποιώντας την τελευταία τιμή του Round Key. Μετά από τους 10 γύρους μετασχηματισμών θα ανακτηθεί η αρχική πληροφορία, όμως ο InvMixColumns εκτελείται μόνο κατά τους πρώτους 9.



Εικόνα 65: Ολοκληρωμένος κύκλος εκτέλεσης του αλγορίθμου AES

4.2.4 Κρυπτανάλυση του AES

Η ευρεία χρησιμοποίηση του αλγορίθμου συμμετρικής κρυπτογράφησης AES έχει προκαλέσει το ενδιαφέρον από ένα πολύ μεγάλο μέρος της κοινότητας των ερευνητών που ασχολούνται με θέματα κρυπτογραφίας.

Έτσι, μετά από εκτεταμένες προσπάθειες κρυπτανάλυσης του εν λόγω αλγορίθμου, βρέθηκαν κάποιες μέθοδοι, όπου μέσω συσχετιζόμενων κλειδιών(related-key attack), μπορούν να προσδιοριστούν οι τιμές από τα ενδιάμεσα states και να οδηγηθούμε στην αποκάλυψη κρυπτογραφημένων δεδομένων.

Η μέθοδος επίθεσης[*] που αναπτύχθηκε από τους Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich και Adi Shamir εφαρμόζεται σε διαφοροποιημένες εκδοχές των AES-192 και AES-256 οι οποίες περιλαμβάνουν μικρότερο αριθμό γύρων με μετασχηματισμούς απ' ότι έχουν οι κανονικές εκδοχές των αλγορίθμων.

Η προαναφερθείσα επίθεση, παρά το γεγονός ότι δεν βρίσκει πρακτική εφαρμογή στις εκδοχές του AES με πλήρη αριθμό μετασχηματισμών, αποδεικνύει ότι το θεωρητικό περιθώριο ασφάλειας που παρέχεται από τον αλγόριθμο είναι μικρότερο απ' ότι είχε αρχικά εκτιμηθεί και ίσως στο μέλλον καταστήσει επιτακτική την ανάγκη αναθεώρησης ορισμένων σημείων της σχεδίασης του αλγορίθμου AES.

4.3 Κρυπτογραφία δημοσίου κλειδιού

Μέχρι τη δεκαετία του 1970 η συμμετρική κρυπτογραφία αποτελούσε το μοναδικό, και κατ' επέκταση το πλέον διαδεδομένο, είδος κρυπτογραφίας. Η συγκεκριμένη μορφή κρυπτογραφίας είχε δύο σημεία για τα οποία αναζητούνταν πιθανές βελτιώσεις.

Το πρώτο από αυτά τα σημεία αφορούσε την ασφαλή ανταλλαγή του συμμετρικού κλειδιού που επρόκειτο να χρησιμοποιηθεί από τις δύο πλευρές που επιθυμούν να επικοινωνήσουν. Άλλωστε, η διαφύλαξη της τιμής του συμμετρικού αποτελεί διαχρονικά το κυρίαρχο ζήτημα για τη λειτουργία της συμμετρικής κρυπτογραφίας, διότι στην περίπτωση που διαρρεύσει το κλειδί τότε αυτομάτως η επικοινωνία εκτίθεται σε οποιονδήποτε το γνωρίζει.

Το δεύτερο σημείο για το οποίο γίνονταν αρκετές προσπάθειες σχετιζόταν με τις «ψηφιακές υπογραφές» ή την εξακρίβωση ταυτότητας. Αφού με τη διάδοση της κρυπτογραφίας κρίθηκε απαραίτητη η ύπαρξη ενός μηχανισμού, μέσω του οποίου θα εξετάζεται αν πράγματι ένα κρυπτογραφημένο μήνυμα έχει σταλεί από μια συγκεκριμένη πλευρά.

[*] Η επίθεση βρίσκεται δημοσιευμένη στην ιστοσελίδα http://eprint.iacr.org/2009/374.pdf

Κατά την προσπάθεια επίλυσης αυτών των ζητημάτων οδηγηθήκαμε στη σύλληψη της ιδέας για την κρυπτογραφία δημοσίου κλειδιού(Public Key Cryptography) ή ασύμμετρη κρυπτογραφία(Asymmetric Cryptography).

4.3.1 Βασικές αρχές της κρυπτογραφίας δημοσίου κλειδιού

Οι πρώτοι που δημοσίευσαν κάτι σχετικό με την κρυπτογραφία δημοσίου κλειδιού ήταν οι Whitfield Diffie και Martin Hellman[*], οι οποίοι το 1976 παρουσίασαν την πρόταση τους για ασφαλή ανταλλαγή κλειδιών. Αν και θεωρείται σχεδόν σίγουρο ότι σε επίπεδο μυστικών υπηρεσιών υπήρχαν ήδη ορισμένες προτάσεις και λύσεις κρυπτογραφίας δημοσίου κλειδιού.

Το σημαντικότερο νέο στοιχείο που εισήχθη με τον αλγόριθμο που πρότειναν οι Diffie και Hellman σχετίζεται με τον αριθμό των κλειδιών που χρησιμοποιούνται. Ο κάθε χρήστης έχει στην κατοχή του ένα ζεύγος ασύμμετρων κλειδιών, σε αντίθεση με το ένα κλειδί που χρησιμοποιείται σε συμμετρικούς αλγόριθμους.

Το ένα τμήμα του ζεύγους κλειδιών αποτελείται από το δημόσιο κλειδί (public key), ενώ το άλλο αντιστοιχεί στο ιδιωτικό κλειδί(private key). Το δημόσιο κλειδί, όπως υποδηλώνει άλλωστε και το όνομα του, είναι προσβάσιμο και γνωστό σε όλους τους χρήστες, ενώ το ιδιωτικό κλειδί διατηρείται μυστικό και χρησιμοποιείται μόνο από τον κάτοχο του.

Ανάμεσα στο δημόσιο και το ιδιωτικό κλειδί υπάρχει μια σχέση που αποτελεί θεμελιώδες συστατικό για τη λειτουργία του αλγορίθμου δημοσίου κλειδιού. Τα δύο τμήματα του ζεύγους λειτουργούν συμπληρωματικά, με το ένα τμήμα να εκτελεί μιαν ενέργεια και το άλλο τμήμα να είναι σε θέση να αναιρέσει αυτή την ενέργεια.

Σαν παράδειγμα αναφέρουμε την κρυπτογράφηση ενός μηνύματος, ας υποθέσουμε ότι χρησιμοποιείται πρώτα το δημόσιο κλειδί και το αρχικό μήνυμα κρυπτογραφείται.

Η αρχική πληροφορία μπορεί να ανακτηθεί από το κρυπτογραφημένο μήνυμα που έχει δημιουργηθεί μόνο όταν χρησιμοποιηθεί το συμπλήρωμα του δημοσίου κλειδιού που ενεπλάκη κατά την κρυπτογράφηση. Με άλλα λόγια, μόνο το ιδιωτικό κλειδί που συμπληρώνει το ζεύγος των κλειδιών είναι σε θέση να αντιστρέψει την κρυπτογράφηση.

Επιπλέον, οι Diffie και Hellman καθόρισαν τις προϋποθέσεις που πρέπει να ικανοποιεί ένας αλγόριθμος δημοσίου κλειδιού. Έτσι, πρέπει:

- Η δημιουργία του ζεύγους δημόσιου και ιδιωτικού κλειδιού να είναι εύκολη και να μην απαιτεί αρκετή υπολογιστική ισχύ.
- Οι διαδικασίες της κρυπτογράφησης και της αποκρυπτογράφησης να είναι πραγματοποιήσιμες χωρίς να απαιτούνται αρκετοί υπολογιστικοί πόροι.
- Να είναι πρακτικά ανέφικτος ο προσδιορισμός του ιδιωτικού κλειδιού από το αντίστοιχο δημόσιο κλειδί. Με άλλα λόγια να απαιτείται τεράστια

[*] Η δημοσίευση των Diffie και Hellman είναι διαθέσιμη από την ιστοσελίδα http://www-ee.stanford.edu/~hellman/publications/24.pdf

υπολογιστική ισχύς για την επίτευξη αυτού του στόχου. Επίσης, θα πρέπει να είναι δύσκολος ο προσδιορισμός του ιδιωτικού κλειδιού ακόμα και στην περίπτωση που κάποιος γνωρίζει το δημόσιο κλειδί και έχει στην κατοχή του ένα κρυπτογραφημένο μήνυμα που έχει παραχθεί από το συγκεκριμένο δημόσιο κλειδί.

4.3.2 Εφαρμογές της κρυπτογραφίας δημοσίου κλειδιού

Ένας αλγόριθμος δημοσίου κλειδιού βρίσκει εφαρμογή σε τουλάχιστον μία από τις περιπτώσεις που ακολουθούν:

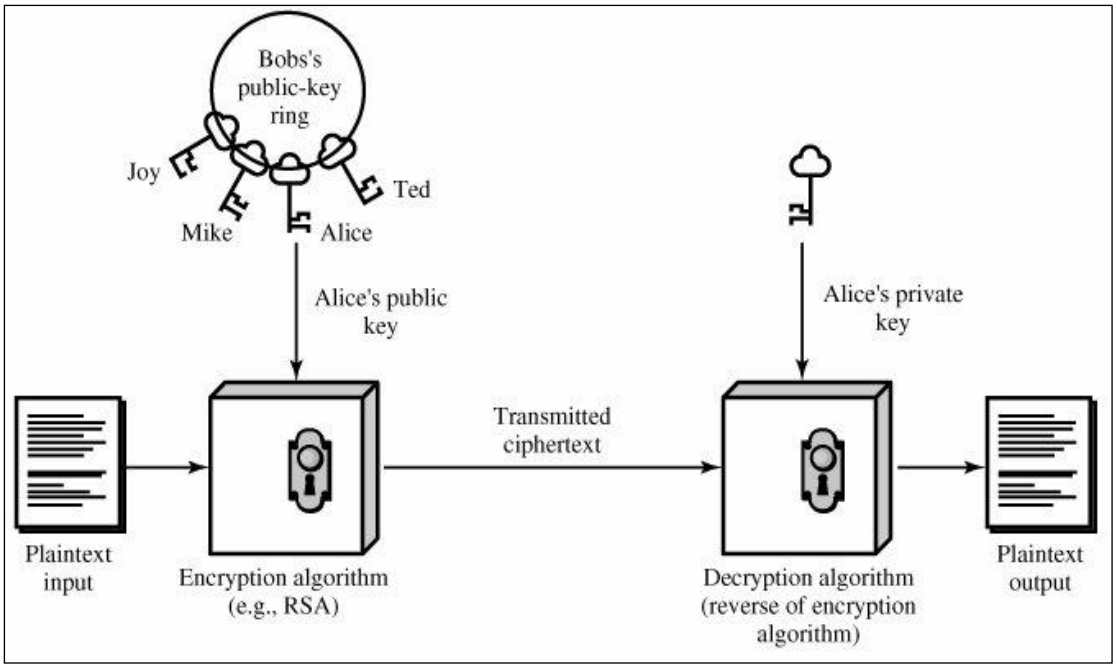
- Κρυπτογράφηση/Αποκρυπτογράφηση μηνυμάτων
- Ασφαλής ανταλλαγή κλειδιών
- Δημιουργία ψηφιακών υπογραφών ή ψηφιακών πιστοποιητικών

Στον πίνακα που ακολουθεί αναφέρονται μερικοί από τους πλέον διαδεδομένους αλγόριθμους κρυπτογραφίας δημοσίου κλειδιού, καθώς και οι εφαρμογές για τις οποίες είναι κατάλληλοι.

Αλγόριθμος	Κρυπτογράφηση Μηνυμάτων	Ανταλλαγή κλειδιών	Ψηφιακές υπογραφές
RSA	☑	☑	☑
Diffie-Hellman	☒	☑	☒
DSS	☒	☒	☑
Ελλειπτικής Καμπύλης	☑	☑	☑

Πίνακας 11: Αλγόριθμοι δημοσίου κλειδιού και οι εφαρμογές που παρέχουν

Στα δύο σχήματα που ακολουθούν απεικονίζονται δύο από τις πιθανές περιπτώσεις στις οποίες εφαρμόζεται η κρυπτογραφία δημοσίου κλειδιού, η μία περίπτωση αφορά την κρυπτογράφηση ενός μηνύματος και η άλλη αναφέρεται στη δημιουργία ψηφιακών υπογραφών.



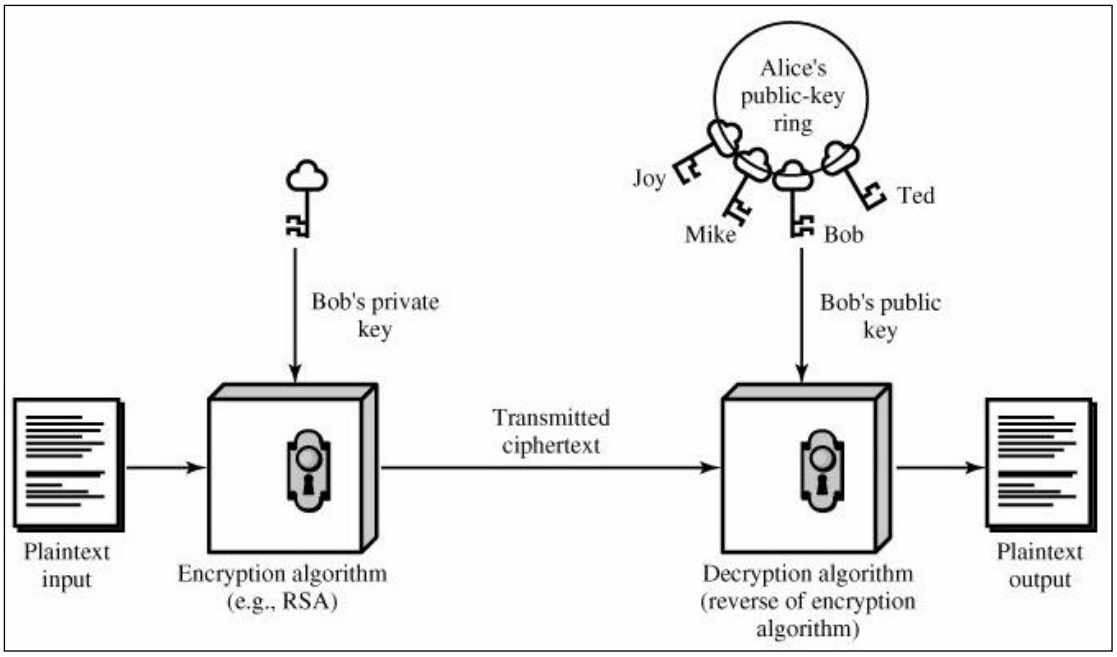
Εικόνα 66: Η κρυπτογράφηση ενός μηνύματος με έναν αλγόριθμο δημοσίου κλειδιού

Υποθέτουμε ότι η Alice και ο Bob είναι τα δύο άκρα που επικοινωνούν και χρησιμοποιούν τον ίδιο αλγόριθμο κρυπτογράφησης δημοσίου κλειδιού. Ο Bob θέλει να στείλει ένα μήνυμα στην Alice και ενδιαφέρεται κυρίως για την εμπιστευτικότητα του περιεχομένου του μηνύματος.

Η διασφάλιση της εμπιστευτικότητας επιτυγχάνεται από τον Bob με τη δημιουργία ενός κρυπτογραφημένου μηνύματος, το οποίο προκύπτει από την εισαγωγή του αρχικού μηνύματος σε έναν αλγόριθμο κρυπτογράφησης και τη χρησιμοποίηση του δημοσίου κλειδιού της Alice.

Η αρχική πληροφορία μπορεί να ανακτηθεί από το κρυπτογραφημένο μήνυμα μόνο από τον κάτοχο του ιδιωτικού κλειδιού που συσχετίζεται με το δημόσιο κλειδί της Alice.

Με άλλα λόγια, το κρυπτογραφημένο μήνυμα είναι «χρήσιμο» μόνο για την Alice, καθώς μόνο εκείνη έχει πρόσβαση στο ιδιωτικό της κλειδί, το οποίο εισάγει μαζί με το κρυπτογραφημένο μήνυμα στον αλγόριθμο που αντιστρέφει τη διαδικασία κρυπτογράφησης και έτσι λαμβάνει την αρχική πληροφορία.



Εικόνα 67: Η επαλήθευση μιας ηλεκτρονικής υπογραφής

Ας υποθέσουμε ότι ο Bob επικοινωνεί με την Alice, μόνο που τώρα οι δύο πλευρές ενδιαφέρονται πρωτίστως για την αυθεντικοποίηση της προέλευσης των μηνυμάτων και όχι τόσο για την εμπιστευτικότητα του περιεχομένου τους.

Η μια πλευρά αποδεικνύει την ταυτότητα της στην άλλη κρυπτογραφώντας τα μηνύματα με το αντίστοιχο ιδιωτικό κλειδί. Έτσι ο Bob κρυπτογραφεί ένα μήνυμα χρησιμοποιώντας το ιδιωτικό του κλειδί και προωθεί το κρυπτογραφημένο μήνυμα στην Alice, η οποία αποκρυπτογραφεί το μήνυμα χρησιμοποιώντας το δημόσιο κλειδί του Bob.

Μέσω της επιτυχημένης αποκρυπτογράφησης η Alice ελέγχει αν πραγματικά ο Bob της έστειλε το μήνυμα, καθώς τότε αποδεικνύεται ότι χρησιμοποιήθηκε το σωστό ιδιωτικό κλειδί, οπότε το μήνυμα πράγματι προέρχεται από τον Bob.

Αυτός είναι ο πλέον απλός τρόπος για τη δημιουργία μιας «ψηφιακής υπογραφής» ή για την εξακρίβωση της ταυτότητας του αποστολέα ενός μηνύματος, αφού με αυτό τον τρόπο δεν παρέχεται εμπιστευτικότητα του περιεχομένου κάθε μηνύματος.

Όμως υπάρχουν και άλλες μέθοδοι, στις οποίες εμπλέκονται και συναρτήσεις κατακερματισμού, με τις οποίες συνδυάζεται η παροχή της εμπιστευτικότητας των περιεχομένων και η δυνατότητα εξακρίβωσης της προέλευσης ενός μηνύματος.

4.3.3 Ο αλγόριθμος RSA

Το 1977, ένα χρόνο μετά τη δημοσίευση του αλγορίθμου Diffie-Hellman, τρεις ερευνητές του MIT, οι Ronald Rivest, Adi Shamir και Leonard Adelman, ολοκλήρωσαν την ανάπτυξη του αλγορίθμου Rivest-Shamir-Adelman(RSA).

Η πρόταση των Rivest-Shamir-Adelman εξελίχθηκε στον πλέον διαδεδομένο και ευρύτατα χρησιμοποιούμενο αλγόριθμο δημοσίου κλειδιού, αφού υποστηρίζει όλες τις πιθανές εφαρμογές της κρυπτογραφίας αυτής της μορφής.

Στην τρέχουσα υπό-ενότητα θα παρουσιάσουμε μόνο τον τρόπο με τον οποίο λειτουργεί ο αλγόριθμος RSA όταν χρησιμοποιείται για την κρυπτογράφηση ενός μηνύματος. Έτσι θα δούμε με την ακόλουθη σειρά πως πραγματοποιείται:

- Η δημιουργία του ζεύγους κλειδιών
- Η κρυπτογράφηση ενός μηνύματος
- Η αποκρυπτογράφηση ενός μηνύματος

Στο τέλος της υπό-ενότητας θα επιχειρήσουμε μέσα από ένα παράδειγμα ολοκληρωμένης λειτουργίας να καταστήσουμε περισσότερο κατανοητό τον τρόπο με τον οποίο εφαρμόζεται ο αλγόριθμος RSA.

Η δημιουργία ενός ζεύγους κλειδιών για τον αλγόριθμο RSA

Η δημιουργία του ζεύγους που αποτελείται από το δημόσιο και το ιδιωτικό κλειδί είναι ένα προαπαιτούμενο βήμα για όποιον επιθυμεί να χρησιμοποιήσει τον αλγόριθμο RSA. Αφού χωρίς την ύπαρξη του ζεύγους κλειδιών είναι αδύνατη η εκτέλεση των λειτουργιών της κρυπτογράφησης και της αποκρυπτογράφησης.

Για τη δημιουργία του ζεύγους κλειδιών, αλλά και γενικότερα για τη λειτουργία του αλγορίθμου RSA, απαιτείται η χρήση πολύ μεγάλων ακεραίων αριθμών και η εμπλοκή τους σε πράξεις της αριθμητικής modulo.

Η διαδικασία ξεκινά με την τυχαία επιλογή δύο πολύ μεγάλων πρώτων αριθμών, οι οποίοι έχουν περίπου τον ίδιο αριθμό δεκαδικών ψηφίων και συμβολίζονται ως p και q. Αυτοί οι δύο αριθμοί συμμετέχουν σε μια σειρά από πράξεις με τις οποίες προσδιορίζονται οι τιμές του δημοσίου και του ιδιωτικού κλειδιού.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι για λόγους που σχετίζονται με την ασφάλεια που παρέχεται από τον αλγόριθμο RSA συνήθως προτιμώνται p και q που έχουν ελάχιστο μήκος τα 150 δεκαδικά ψηφία ή είναι τουλάχιστον 512 bits.

Από τον πολλαπλασιασμό ανάμεσα στο p και το q παράγεται ο αριθμός n, που αντιπροσωπεύει το modulo που πρόκειται να χρησιμοποιηθεί στο σύστημα ακεραίων, ενώ την ίδια στιγμή αποτελεί τμήμα του δημοσίου κλειδιού.

Έχοντας σαν δεδομένο ότι οι αριθμοί p και q έχουν τουλάχιστον 150 δεκαδικά ψηφία, προκύπτει ότι το modulo n έχει μήκος κάτι παραπάνω από 300 δεκαδικά ψηφία ή περίπου 1024 δυαδικά ψηφία.

Ένας ακέραιος αυτής της τάξης είναι πολύ δύσκολο να αναλυθεί σε γινόμενο πρώτων παραγόντων, αφού ακόμα και με τη χρήση ηλεκτρονικού υπολογιστή χρειάζεται ένας μεγάλος αριθμός από χρονοβόρους υπολογισμούς για να επιτευχθεί αυτός ο στόχος.

Έτσι λοιπόν, η δυσκολία παραγοντοποίησης του modulo *n*, όπου συχνά η συγκεκριμένη περίπτωση αναφέρεται και ως το «RSA πρόβλημα», είναι το στοιχείο με το οποίο δομείται η ασφάλεια που παρέχεται από τον αλγόριθμο RSA. Αφού με αυτό τον τρόπο διατηρούνται μυστικές οι τιμές των *p* και *q*, από τις οποίες μπορούμε να οδηγηθούμε στον προσδιορισμό του ιδιωτικού κλειδιού.

Στη συνέχεια, μέσω του πολλαπλασιασμού του $(p - 1)$ με το $(q - 1)$, υπολογίζεται η τιμή $\varphi(n)$. Η συγκεκριμένη τιμή αντιπροσωπεύει το πλήθος των ακεραίων που είναι μικρότεροι από τον αριθμό *n*, ενώ παράλληλα είναι πρώτοι σε σχέση με αυτόν.

Με τη βοήθεια της τιμής $\varphi(n)$ πραγματοποιείται η επιλογή του αριθμού *e*, ο οποίος ονομάζεται εκθέτης δημοσίου κλειδιού(public key exponent). Για τον εκθέτη *e* απαιτείται να ισχύουν ταυτόχρονα δύο συνθήκες, θα πρέπει $1<e<\varphi(n)$ και ο μέγιστος κοινός διαιρέτης του *e* και της τιμής $\varphi(n)$ να είναι ίσος με 1.

Μια αρκετά δημοφιλής περίπτωση για την τιμή του εκθέτη δημοσίου κλειδιού είναι αυτή όπου $e = 2^{16} + 1 = 65537$.

Μέσω του εκθέτη δημοσίου κλειδιού και της τιμής $\varphi(n)$ είναι δυνατός ο προσδιορισμός του εκθέτη ιδιωτικού κλειδιού(private key exponent), ο οποίος συμβολίζεται ως *d*. Ο συγκεκριμένος εκθέτης θα πρέπει να έχει τιμή $1<d<\varphi(n)$, ενώ ταυτόχρονα πρέπει να αποτελεί τον πολλαπλασιαστικό αντίστροφο του $e \bmod \varphi(n)$. Το οποίο ισοδύναμα εκφράζεται ως $e \cdot d \equiv 1 \bmod (p - 1)(q - 1)$.

Ο εκθέτης δημοσίου κλειδιού *e* μαζί με το modulo *n* σχηματίζουν το δημόσιο κλειδί, που συμβολίζεται και ως $KU = (e,n)$.

Οι πρώτοι αριθμοί *p,q* και ο ιδιωτικός εκθέτης *d* πρέπει να παραμείνουν γνωστοί μόνο στον κάτοχο τους. Αυτός είναι και ο λόγος για τον οποίο το ιδιωτικό κλειδί διατηρείται μυστικό και είναι γνωστό μόνο στον κάτοχο του, καθώς περιέχει τα προαναφερθέντα στοιχεία και συμβολίζεται ως $KR = (d,p,q)$.

Η κρυπτογράφηση ενός μηνύματος με τον αλγόριθμο RSA

Υποθέτουμε ότι έχουμε δύο πλευρές που επιθυμούν να επικοινωνήσουν χρησιμοποιώντας τον αλγόριθμο RSA. Σε αυτή την περίπτωση, ο αποστολέας ενός μηνύματος πρέπει να το κρυπτογραφήσει πριν προχωρήσει στη μετάδοση του.

Η κρυπτογράφηση είναι εφικτή όταν ο αποστολέας γνωρίζει το πραγματικό δημόσιο κλειδί του παραλήπτη, στο οποίο περιλαμβάνονται ο δημόσιος εκθέτης *e* και το modulo *n*. Όταν η κρυπτογράφηση ενός μηνύματος πραγματοποιείται με το δημόσιο κλειδί του παραλήπτη, τότε μόνο αυτός μπορεί να ανακτήσει το αρχικό μήνυμα.

Μια επιπρόσθετη υποχρέωση του αποστολέα είναι η μετατροπή του αρχικού μηνύματος, το οποίο θεωρούμε ότι συμβολίζεται ως *m*, σε έναν ακέραιο αριθμό που είναι μικρότερος από το *n* που περιλαμβάνεται στο δημόσιο κλειδί του παραλήπτη.

Η μετατροπή του *m* σε ακέραιο γίνεται σύμφωνα με κάποιο προσυμφωνημένο πρωτόκολλο και είναι απολύτως αναγκαία, αφού η κρυπτογράφηση πραγματοποιείται μέσα από μια σειρά με πράξεις ανάμεσα σε μεγάλους ακέραιους αριθμούς.

Το κρυπτογραφημένο μήνυμα, που θεωρούμε ότι συμβολίζεται ως *c*, προκύπτει σε δύο βήματα. Αρχικά, έχουμε την ύψωση του *m* στο δημόσιο εκθέτη *e*. Στη συνέχεια, το αποτέλεσμα αυτό υποβάλλεται στην πράξη mod με το συντελεστή *n* και από εκεί λαμβάνεται ο ακέραιος αριθμός που αντιπροσωπεύει το κρυπτογραφημένο μήνυμα. Όλα τα παραπάνω συνοψίζονται στον τύπο που ακολουθεί:

$$c = m^e \bmod n$$

Βέβαια, η επιλογή ενός μεγάλου εκθέτη *e* είναι ένα πολύ συχνό φαινόμενο, το οποίο καθιστά τη διαδικασία κρυπτογράφησης χρονοβόρα. Αφού από την πράξη m^e προκύπτει ένας πολύ μεγάλος αριθμός ο οποίος στη συνέχεια συμμετέχει στην πράξη mod με τον επίσης μεγάλο συντελεστή *n*.

Υπάρχει όμως η δυνατότητα να επιταχυνθεί η διαδικασία της κρυπτογράφησης μέσω της μεθόδου των διαδοχικών εκθετικών τετραγώνων(exponentiation by squaring). Με αυτό τον τρόπο η ύψωση του *m* στον εκθέτη *e* αντικαθίσταται από μια σειρά από ευκολότερους και συντομότερους υπολογισμούς.

Η αποκρυπτογράφηση ενός μηνύματος με τον αλγόριθμο RSA

Υποθέτουμε ξανά ότι έχουμε δύο πλευρές που επικοινωνούν χρησιμοποιώντας τον αλγόριθμο RSA. Ο αποστολέας ενός μηνύματος προχωρά στην κρυπτογράφηση του χρησιμοποιώντας το δημόσιο κλειδί της πλευράς για την οποία προορίζεται το μήνυμα.

Μετά από την επιτυχημένη λήψη του μηνύματος από τον παραλήπτη ξεκινά η διαδικασία της αποκρυπτογράφησης. Ο παραλήπτης επιχειρεί να ανακτήσει από το κρυπτογραφημένο μήνυμα τον ακέραιο αριθμό *m* που αντιπροσωπεύει το αρχικό μήνυμα. Οι δύο πλευρές έχουν συμφωνήσει εκ των προτέρων για τον τρόπο με τον οποίο το αρχικό μήνυμα θα μετατρέπεται στον ακέραιο αριθμό *m*, οπότε ο παραλήπτης του κρυπτογραφημένου μηνύματος γνωρίζει τι απαιτείται για να αντιστρέψει τη μετατροπή του *m*.

Η ανάκτηση του ακεραίου *m* από την κρυπτογραφημένη πληροφορία *c* επιτυγχάνεται μέσα από μια σειρά πράξεων στις οποίες έχουμε τη χρησιμοποίηση του ιδιωτικού εκθέτη *d* και του συντελεστή *n*. Πιο συγκεκριμένα, ο ακέραιος *c* υψώνεται στον ιδιωτικό εκθέτη *d* και το αποτέλεσμα που προκύπτει εισάγεται σε μια πράξη mod με το modulo *n*.
Συνοπτικά λοιπόν έχουμε:

$$m = c^d \bmod n$$

Όμως, ο ιδιωτικός εκθέτης *d* είναι συνήθως πολύ μεγαλύτερος σε σχέση με το δημόσιο εκθέτη *e*. Για αυτόν ακριβώς το λόγο η αποπεράτωση της διαδικασίας αποκρυπτογράφησης χρειάζεται περισσότερο χρόνο και περισσότερους υπολογιστικούς πόρους από ότι απαιτείται για τη διαδικασία κρυπτογράφησης.

Ευτυχώς όμως υπάρχει μια μέθοδος με την οποία επιταχύνεται η διαδικασία, ενώ παράλληλα εξοικονομούνται πολύτιμοι πόροι. Η εν λόγω βελτιστοποίηση είναι εφικτή μέσω του συνδυασμού του αλγορίθμου του Garner και του Κινέζικου Θεωρήματος για το Υπόλοιπο των διαιρέσεων(Chinese Remainder Theorem - CRT).

Παράδειγμα εφαρμογής του αλγορίθμου RSA

Σε αυτό το σημείο θα δούμε μέσα από ένα πρακτικό παράδειγμα τον τρόπο με τον οποίο εφαρμόζεται ο αλγόριθμος RSA. Οι αριθμοί που θα χρησιμοποιηθούν είναι αρκετά μικροί και σε καμία περίπτωση δεν είναι αντιπροσωπευτικοί μιας αληθινής περίπτωσης. Άλλωστε, ο κύριος στόχος του παραδείγματος είναι να βοηθήσει στην κατανόηση της λειτουργίας του αλγορίθμου RSA.

Η δημιουργία του ζεύγους που αποτελείται από το δημόσιο και το ιδιωτικό κλειδί πραγματοποιείται πριν από οποιαδήποτε άλλη λειτουργία.

Ξεκινάμε λοιπόν με την επιλογή των p και q, των δύο πρώτων αριθμών στους οποίους βασίζεται η δημιουργία ενός ζεύγους κλειδιών. Οι τιμές που επιλέγουμε είναι οι εξής:

$$p=17$$
$$q=11$$

Οπότε, ο συντελεστής n έχει την τιμή $n = p \cdot q = 17 \text{ x } 11 = 187$.

Αντίστοιχα, η τιμή που δίνεται από τη συνάρτηση $\varphi()$ είναι $\varphi(n) = (p\text{-}1) \text{ x } (q\text{-}1)$, δηλαδή είναι $\varphi(187) = (17\text{-}1) \text{ x } (11\text{-}1) = 16 \text{ x } 10 = 160$.

Ακολουθεί η επιλογή του δημοσίου εκθέτη e. Να υπενθυμίσουμε ότι για τον εκθέτη e πρέπει να ισχύουν ταυτόχρονα δύο συνθήκες, να έχει τέτοια τιμή ώστε $1<e<\varphi(n)$ και ο μέγιστος κοινός διαιρέτης του e και της τιμής $\varphi(n)$ να είναι ίσος με 1. Για το παράδειγμα μας επιλέγουμε $e = 7$.

Με τη βοήθεια του δημόσιου εκθέτη e μπορούμε να υπολογίσουμε τον ιδιωτικό εκθέτη d.

Ο εκθέτης d πρέπει να είναι μικρότερος της τιμής $\varphi(n)$, ενώ ταυτόχρονα πρέπει να αποτελεί τον πολλαπλασιαστικό αντίστροφο του e mod $\varphi(n)$.

Οπότε ισχύει $e \cdot d \equiv 1 \bmod \varphi(n)$, δηλαδή $7 \cdot d \equiv 1 \bmod 160$ που ισοδυναμεί με τη Διοφαντική εξίσωση $(7 \cdot d) = (160 \cdot k) + 1$.

Η εξίσωση λύνεται για k=1 και d=23 και έτσι έχουμε υπολογίσει όλα τα απαραίτητα στοιχεία για το σχηματισμό του ζεύγους κλειδιών του αλγορίθμου RSA.

Έτσι, έχουμε το δημόσιο κλειδί $KU = (e,n) = (7, 187)$. Ενώ το ιδιωτικό κλειδί που σχηματίζεται είναι $KR = (d,p,q) = (23, 17, 11)$.

Αφού έχουμε στη διάθεση μας το δημόσιο και το ιδιωτικό κλειδί μπορούμε να περάσουμε στην παρουσίαση των διαδικασιών της κρυπτογράφησης και της αποκρυπτογράφησης ενός μηνύματος.

Ας δούμε πρώτα τι γίνεται κατά την κρυπτογράφηση ενός μηνύματος. Το μήνυμα πρέπει να έχει μετασχηματιστεί σε ακέραιο αριθμό. Για το παράδειγμα μας χρησιμοποιούμε θεωρούμε ότι ένα μήνυμα έχει μετατραπεί σε ακέραιο αριθμό με την τιμή $m = 88$.

Η κρυπτογράφηση πραγματοποιείται με τον υπολογισμό του ακεραίου αριθμού c που αντιπροσωπεύει το κρυπτογραφημένο μήνυμα. Ο συγκεκριμένος αριθμός προκύπτει από τη σχέση:

$$c = m^e \bmod n$$

Υπάρχουν δύο μέθοδοι υπολογισμού του c. Στην πρώτη περίπτωση πραγματοποιείται κανονικά η ύψωση του m στον εκθέτη e, οπότε έχουμε:

$$c = 88^7 \bmod 187 = 40867559636992 \bmod 187 = 11.$$

Όμως, όπως αναφέραμε και νωρίτερα αυτή η πράξη είναι στις περισσότερες περιπτώσεις αρκετά χρονοβόρα.

Γι' αυτό προτιμάμε τη δεύτερη μέθοδο που έχουμε τη συμμετοχή των διαδοχικών εκθετικών τετραγώνων. Σε αυτή την περίπτωση ο εκθέτης e αναλύεται σε δυνάμεις του 2, στην περίπτωση μας είναι $7 = 2^0 + 2^1 + 2^2$. Υπολογίζουμε κάθε μια από αυτές τις δυνάμεις και πραγματοποιούμε την πράξη mod με το συντελεστή n.

Οπότε για τη δύναμη για το 2^0 έχουμε: $88^1 \bmod 187 = 88$.

Η επόμενη δύναμη προκύπτει από την τιμή της προηγούμενης όταν υψώνεται στο τετράγωνο, οπότε για τη δύναμη 2^1 έχουμε:
$88^2 \bmod 187 = (88^1)^2 \bmod 187 = 88 \times 88 \bmod 187 = 7744 \bmod 187 = 77$.

Με τον ίδιο τρόπο για τη δύναμη 2^2 έχουμε:
$88^4 \bmod 187 = (88^2)^2 \bmod 187 = 77 \times 77 \bmod 187 = 5929 \bmod 187 = 132$.

Έτσι, το c προκύπτει με τον ακόλουθο τρόπο:
$c = 88^7 \bmod 187 = [(88^1 \bmod 187) \times (88^2 \bmod 187) \times (88^4 \bmod 187)] \bmod 187$
$= [88 \times 77 \times 132] \bmod 187 = 894432 \bmod 187 = 11$.

Η αποκρυπτογράφηση γίνεται με την ανάκτηση του ακεραίου m από τον ακέραιο c. Αυτό είναι εφικτό μέσα από τη σχέση:

$$m = c^d \bmod n = 11^{23} \bmod 187$$

Ο ιδιωτικός εκθέτης που αντιστοιχεί στο συγκεκριμένο παράδειγμα δεν είναι πολύ μεγαλύτερος από τον δημόσιο εκθέτη, γεγονός που επιτρέπει τον υπολογισμό της τιμής 11^{23} ακόμα και με τη μέθοδο των διαδοχικών εκθετικών τετραγώνων.

Όμως, θα προτιμήσουμε να δείξουμε τον τρόπο με τον οποίο πραγματοποιείται η αποκρυπτογράφηση όταν χρησιμοποιείται ο αλγόριθμος του Garner μαζί με το θεώρημα CRT.

Με τη συγκεκριμένη μέθοδο εκτελείται δύο φορές η σχέση με την οποία υπολογίζεται ο αριθμός *m*, με τη μόνη διαφορά ότι δεν χρησιμοποιείται ο εκθέτης *d*, αλλά χρησιμοποιούνται δύο μικρότεροι εκθέτης που προέρχονται από το υπόλοιπο της διαίρεσης του *d* με τις τιμές των (*p* - 1) και (*q* - 1).

Με αυτό τον τρόπο η αποκρυπτογράφηση πραγματοποιείται 4 φορές ταχύτερα από ότι θα γινόταν με την άλλη μέθοδο.

Για την εφαρμογή της συγκεκριμένης βελτιστοποίησης χρειάζεται να υπολογίσουμε δύο εκθέτες και ένα συντελεστή για το θεώρημα CRT.

Οι δύο εκθέτες συμβολίζονται ως *dP, dQ* και είναι θετικοί ακέραιοι αριθμοί για τους οποίου ισχύουν οι σχέσεις:

$$e \cdot dP \equiv 1 \bmod (p - 1)$$
$$e \cdot dQ \equiv 1 \bmod (q - 1)$$

Ο συντελεστής CRT συμβολίζεται ως *qInv*, είναι θετικός ακέραιος μικρότερος από τον αριθμό p. Για το συντελεστή *qInv* ισχύει η σχέση:

$$q \cdot qInv \equiv 1 \bmod p$$

Οι αριθμοί *dP, dQ* και *qInv* συνήθως υπολογίζονται μια φορά, αποθηκεύονται, και αποτελούν τμήμα του ιδιωτικού κλειδιού, το οποίο λαμβάνει τη μορφή:

$$KR = (p, q, dP, dQ, qInv)$$

Στο δικό μας παράδειγμα το *dP* δίνεται από τη σχέση:
$e \cdot dP \equiv 1 \bmod (p - 1)$, δηλαδή $7 \cdot dP \equiv 1 \bmod (17 - 1)$, το οποίο ισοδυναμεί με τη Διοφαντική εξίσωση $(7 \cdot dP) = (16 \cdot k) + 1$, η οποία λύνεται για k=3 και *dP*=7.

Με παρόμοιο τρόπο έχουμε για το *dQ*:
$e \cdot dQ \equiv 1 \bmod (q - 1)$, δηλαδή $7 \cdot dQ \equiv 1 \bmod (11 - 1)$, το οποίο ισοδυναμεί με τη σχέση $(7 \cdot dQ) = (10 \cdot k) + 1$, με τις λύσεις να είναι k=2 και *dQ*=3.

Ενώ το qInv είναι:
$q \cdot qInv \equiv 1 \bmod p$, δηλαδή $11 \cdot qInv \equiv 1 \bmod 17$, το οποίο ισοδυναμεί με την εξίσωση $(11 \cdot qInv) = (17 \cdot k) + 1$, με τις λύσεις να είναι k=9 και qInv=14.

Έχοντας πια στη διάθεση μας όλα τα απαραίτητα στοιχεία μπορούμε να αρχίσουμε τη διαδικασία με την οποία αντιστρέφεται η κρυπτογράφηση και ανακτάται ο ακέραιος αριθμός *m*.

Θεωρούμε ότι υπάρχουν δύο ακέραιοι m_1 και m_2 για τους οποίους ισχύει:

$$m_1 = c^{dP} \bmod p$$
$$m_2 = c^{dQ} \bmod q$$

Οπότε, για την περίπτωση μας έχουμε:
$m_1 = c^{dP} \bmod p = 11^7 \bmod 17 = [(11^1 \bmod 17) \text{ x } (11^2 \bmod 17) \text{ x } (11^4 \bmod 17)] \bmod 17$
$= [11 \text{ x } 2 \text{ x } 4] \bmod 17 = 88 \bmod 17 = 3$

$m_2 = c^{dQ} \bmod q = 11^3 \bmod 11 = [(11^1 \bmod 17) \text{ x } (11^2 \bmod 17)] \bmod 11$
$= [11 \text{ x } 2] \bmod 11 = 22 \bmod 11 = 0$

Στη συνέχεια υπολογίζουμε το συντελεστή h, για τον οποίο ισχύει η σχέση:

$$h = [(m_1 - m_2) \cdot qInv] \bmod p$$

Οπότε για το παράδειγμα μας ο συντελεστής ισούται με την τιμή:
$h = [(m_1 - m_2) \cdot qInv] \bmod p = [(3 - 0) \cdot 14] \bmod 17 = 3 \cdot 14 \bmod 17 = 42 \bmod 17 = 8$

Μετά από αυτό φτάνουμε στο τελευταίο βήμα, με το οποίο ολοκληρώνεται η ανάκτηση του ακεραίου m που αντιπροσωπεύει το αρχικό μήνυμα. Η τιμή του συγκεκριμένου αριθμού δίνεται από τη σχέση:

$$m = m_2 + q \cdot h$$

Έτσι, για το τρέχον παράδειγμα έχουμε:
$m = m_2 + q \cdot h = 0 + (8 \cdot 11) = 88$

4.3.4 Κρυπτανάλυση του RSA

Ο αλγόριθμος RSA αποτελεί την πλέον διαδεδομένη περίπτωση κρυπτοσυστήματος δημοσίου κλειδιού αφού εκτός από την κρυπτογράφηση και τη δημιουργία ψηφιακών υπογραφών χρησιμοποιείται και σε περιπτώσεις όπως:

- Η θωράκιση ηλεκτρονικού ταχυδρομείου
- Η διασφάλιση ηλεκτρονικού εμπορίου
- Η υλοποίηση εικονικών ιδιωτικών δικτύων(virtual private networks)
- Η εξακρίβωση της αυθεντικότητας ηλεκτρονικών εγγράφων

Έτσι λοιπόν δεν θα αποτελούσε υπερβολή αν λέγαμε ότι ο αλγόριθμος RSA, χάρις στις ιδιότητες του, αποτελεί την καρδιά των περισσοτέρων ενεργειών που εκτελούνται με ηλεκτρονική μορφή μέσα από ανοιχτά δίκτυα όπως είναι το Internet.

Το συγκεκριμένο γεγονός, σε συνδυασμό με το ότι ο RSA είναι γνωστός από το 1977, έχει οδηγήσει σε εκτεταμένες προσπάθειες κρυπτανάλυσης και εύρεσης αδυναμιών. Παρόλο που δεν έχει βρεθεί κάποια επίθεση που θα ήταν καταστροφική για τον αλγόριθμο, μέσα από την κρυπτανάλυση έχουν αναδειχθεί στοιχεία που μπορούν να χρησιμοποιηθούν ως κατευθυντήριες γραμμές που θα οδηγήσουν στην ορθότερη χρήση και υλοποίηση του RSA.

Οι ακόλουθες είναι μερικές από τις μεθόδους που χρησιμοποιούνται ενάντια στον αλγόριθμο RSA:

- Η παραγοντοποίηση του *n*, μέσω του τετραγωνικού κόσκινου(quadratic sieve) ή μέσω του κόσκινου πεδίου αριθμών(number field sieve)
- Οι δοκιμαστικές διαιρέσεις του *n* με μεγάλους πρώτους αριθμούς
- Επιθέσεις που εστιάζουν στην υλοποίηση του αλγορίθμου(side-channel attacks)

Οι τελευταίες εξελίξεις πάνω στην κρυπτανάλυση του αλγορίθμου RSA προήλθαν μέσα από τη συντονισμένη προσπάθεια που ξεκίνησε το 2007 μια ομάδα με 13 ερευνητές από 6 διαφορετικά πανεπιστημιακά ιδρύματα και ερευνητικά εργαστήρια.

Η συγκεκριμένη ομάδα, μέσα από την παράλληλη εκτέλεση της μεθόδου του κόσκινου πεδίου αριθμών σε κάποιες εκατοντάδες επεξεργαστών κατόρθωσαν το 2009 να παραγοντοποιήσουν έναν αριθμό *n* με μήκος 232 ψηφίων που αντιστοιχεί σε περίπου 768 bits[*].

Το γεγονός αυτό μας οδηγεί στην υποχρεωτική υιοθέτηση ενός modulo *n* που θα έχει μήκος τουλάχιστον 1024 δυαδικά ψηφία ή, εναλλακτικά κατά τη δημιουργία του ζεύγους κλειδιών, στην επιλογή πρώτων αριθμών *p* και *q* με ελάχιστο μήκος 512 bits.

Η αναπτυχθείσα εφαρμογή δεν θα μπορούσε να συμπεριφέρεται διαφορετικά, γι' αυτό και στο τμήμα όπου εμπλέκεται ο αλγόριθμος RSA επιλέγεται η χρησιμοποίηση πρώτων αριθμών *p* και *q* με μήκος 512 δυαδικών ψηφίων.

4.4 Συναρτήσεις Κατακερματισμού

Μια συνάρτηση κατακερματισμού(Hash Function), ακολουθεί μια συγκεκριμένη διαδικασία που καλείται συμπίεση(Compression). Μέσω της συμπίεσης πραγματοποιείται η αντιστοίχηση μιας μεταβλητής ποσότητας δεδομένων σε μια προκαθορισμένη σταθερή ποσότητα που καλείται σύνοψη(Digest) ή τιμή κατακερματισμού(Hash Value).

Πιο σύντομα μπορούμε να πούμε ότι ισχύει:

$$h = H(M),$$

όπου το *h* αντιπροσωπεύει τη σύνοψη που αντιστοιχεί στην πληροφορία *M* όταν αυτή εισάγεται σε μια συνάρτηση κατακερματισμού *H( )*.

Μια συνάρτηση κατακερματισμού έχει τις ακόλουθες ιδιότητες:

- Εφαρμόζεται σε οποιαδήποτε πληροφορία *M* ανεξαρτήτως του μεγέθους της.
- Η σύνοψη που παράγεται έχει πάντα το μήκος δυαδικών ψηφίων που ορίζεται σύμφωνα με τη συνάρτηση κατακερματισμού που χρησιμοποιείται.
- Η πράξη *h = H (M)* είναι εύκολη υπολογιστικά, χωρίς να επηρεάζεται από το μέγεθος του *M*.
- Λειτουργεί σύμφωνα με την τιμή που έχουν τα δυαδικά ψηφία της εισερχόμενης πληροφορίας *M*. Για μια δεδομένη πληροφορία *M* παράγεται

[*] Η μέθοδος είναι δημοσιευμένη στην ιστοσελίδα http://eprint.iacr.org/2010/006.pdf

πάντα η ίδια τιμή κατακερματισμού. Μια αλλαγή σε ένα ή περισσότερα από τα εισερχόμενα bits αρκεί για να φέρει ένα πλήθος από αλλαγές στην έξοδο, παρόμοιο με το αποτέλεσμα μιας χιονοστιβάδας(the avalanche effect). Με αυτό τον τρόπο γίνεται εύκολα αντιληπτή οποιαδήποτε αλλαγή πραγματοποιείται σε μια ποσότητα πληροφορίας, αφού η τιμή κατακερματισμού που παράγεται είναι διαφοροποιημένη κατά το μέγιστο δυνατό βαθμό.

- Είναι μονόδρομη(one-way function). Δηλαδή, για μια δεδομένη τιμή κατακερματισμού είναι αδύνατο να αντιστραφεί η συνάρτηση και να προσδιοριστεί η πληροφορία από την οποία προήλθε η συγκεκριμένη σύνοψη.
- Είναι ανθεκτική σε ασθενείς συγκρούσεις(weak collision resistant function). Δηλαδή, για μια γνωστή πληροφορία *X* είναι αδύνατο να βρεθεί μια διαφορετική πληροφορία *Y*, τέτοια ώστε *H (X) = H (Y).*
- Είναι ανθεκτική σε ισχυρές συγκρούσεις(strong collision resistant function). Δηλαδή, είναι αδύνατο να βρεθεί οποιοδήποτε ζεύγος πληροφοριών *X, Y* που να παράγει την ίδια τιμή σύνοψης.

Οι συναρτήσεις κατακερματισμού χρησιμοποιούνται συνήθως:

- Για την επαλήθευση της ακεραιότητας ενός μηνύματος ή αρχείου που ανταλλάσσεται ανάμεσα σε δύο πλευρές. Η επαλήθευση γίνεται με τη σύγκριση ανάμεσα στην τιμή σύνοψης πριν τη μετάδοση και την τιμή σύνοψης μετά τη μετάδοση.

- Για την αναγνώριση ενός αρχείου. Όπως γίνεται για παράδειγμα στα ομότιμα δίκτυα διαμοίρασης αρχείων, όπου η τιμή σύνοψης αποτελεί το διαχωριστικό στοιχείο ενός αρχείου.

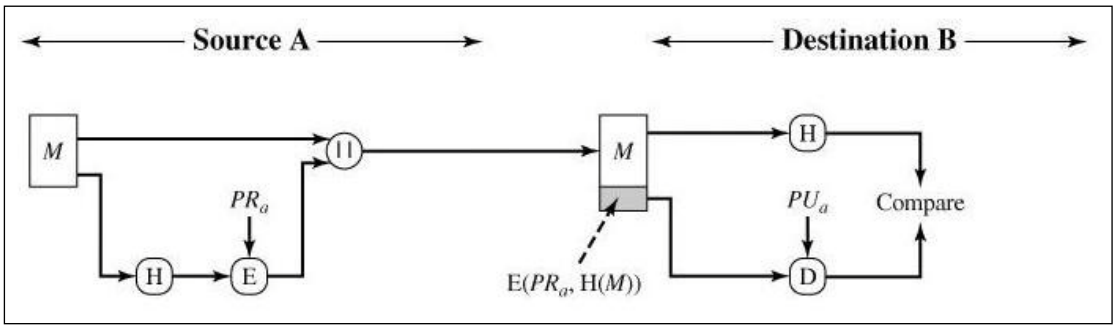
- Για τον εντοπισμό αλλαγών σε ένα αρχείο. Μια διαφορετική τιμή σύνοψης υποδηλώνει ότι ένα αρχείο έχει δεχτεί ορισμένες αλλαγές.

- Για την αποθήκευση κωδικών. Όπως γίνεται σε διάφορες διανομές του λειτουργικού συστήματος Linux, όπου για λόγους ασφαλείας οι κωδικοί δεν αποθηκεύονται με τη μορφή κειμένου.

- Για την αυθεντικοποίηση χρηστών. Αυτή η περίπτωση συνδυάζεται με την ακριβώς προηγούμενη. Η αυθεντικοποίηση ενός χρήστη πραγματοποιείται με τη σύγκριση μεταξύ του αποθηκευμένου hash που του αντιστοιχεί και την τιμή κατακερματισμού που δημιουργείται εκείνη τη στιγμή από την εισαγωγή του κωδικού.

- Για την παραγωγή κλειδιών τα οποία χρησιμοποιούνται σε συμμετρικούς κρυπτογραφικούς αλγόριθμους. Όπως γίνεται με τη μέθοδο Password Based Key Derivation Function(PBKDF), την οποία αναλύουμε παρακάτω.

- Για τη δημιουργία και την εξακρίβωση ψηφιακών υπογραφών. Σε αυτή την περίπτωση μια συνάρτηση κατακερματισμού συνδυάζεται με έναν αλγόριθμο δημοσίου κλειδιού που υποστηρίζει τη λειτουργία ψηφιακών υπογραφών. Στο σχήμα που ακολουθεί απεικονίζεται το απλούστερο σενάριο αυτής της εφαρμογής.



Εικόνα 68: Η δημιουργία και η επαλήθευση μιας ψηφιακής υπογραφής

Η πλευρά Α συντάσσει το μήνυμα *M*, υπολογίζει τη σύνοψη *H(M)* και την κρυπτογραφεί χρησιμοποιώντας το ιδιωτικό κλειδί PR_a που ανήκει σε αυτή την πλευρά.

Το μήνυμα και η κρυπτογραφημένη σύνοψη συνενώνονται και προωθούνται στην πλευρά Β. Ο παραλήπτης του μηνύματος υπολογίζει εκ νέου τη σύνοψη του *M*, ενώ παράλληλα χρησιμοποιεί το δημόσιο κλειδί PU_a του αποστολέα για να αποκρυπτογραφήσει τη σύνοψη που είχε επισυναφθεί στο *M*.

Ακολουθεί σύγκριση των δύο τιμών κατακερματισμού, σε περίπτωση που είναι ίσες, τότε η πλευρά Β μπορεί να ισχυριστεί ότι το μήνυμα πράγματι προήλθε από την πλευρά Α χωρίς να έχει υποστεί οποιαδήποτε τροποποίηση.

Ο εν λόγω ισχυρισμός βασίζεται στο γεγονός ότι η πλευρά Α «υπέγραψε» τη σύνοψη χρησιμοποιώντας το ιδιωτικό της κλειδί, το οποίο είναι γνωστό μόνο σε εκείνη.

4.4.1 Οι συναρτήσεις κατακερματισμού SHA

Μερικές από τις πιο γνωστές συναρτήσεις κατακερματισμού είναι μέλη της οικογένειας Secure Hash Algorithm(SHA).

Η πρώτη συνάρτηση κατακερματισμού που εμφανίστηκε από τη συγκεκριμένη οικογένεια είναι η SHA-1, η οποία έγινε γνωστή το 1993, μέσα από τις προδιαγραφές FIPS PUB 180 που εκδόθηκαν το ινστιτούτο NIST.

Η συνάρτηση SHA-1 δέχεται ένα μήνυμα με μήκος μέχρι $(2^{64} - 1)$ δυαδικά ψηφία και παράγει μια σύνοψη του μηνύματος με μήκος 160 δυαδικών ψηφίων.

Το 2002 πραγματοποιήθηκε αναθεώρηση των προδιαγραφών FIPS PUB 180. Μέσα από τις αναθεωρημένες προδιαγραφές FIPS PUB 180-2 ορίζονται τρεις επιπλέον συναρτήσεις κατακερματισμού που είναι παραλλαγές της συνάρτησης SHA-2.

Οι νέες συναρτήσεις που εισήχθησαν ήταν οι SHA-256, SHA-384 και SHA-512. Όλες τους έχουν κοινή δομή και παρόμοια λογική, όμως διαφέρουν σε ορισμένες πράξεις που πραγματοποιούνται ανάμεσα στα δυαδικά ψηφία που επεξεργάζονται. Αλλά η σημαντικότερη διαφορά εντοπίζεται στο μήκος που έχει η σύνοψη που

παράγουν, αφού το πλήθος των δυαδικών ψηφίων της παραγόμενης σύνοψης υποδηλώνεται από τον αριθμό που ακολουθεί το αντίστοιχο όνομα κάθε συνάρτησης.

Μετά από ένα σύντομο χρονικό διάστημα προστέθηκε και η συνάρτηση SHA-224, η οποία λειτουργεί όπως η SHA-256 και παράγει την ανάλογη σύνοψη, με τη διαφορά όμως ότι «αγνοεί» τα τελευταία 32 δυαδικά ψηφία και έτσι παράγει σύνοψη η οποία έχει μήκος 224 δυαδικά ψηφία.

4.4.2 Η συνάρτηση κατακερματισμού SHA-256

Σε αυτό το σημείο θα εστιάσουμε στον τρόπο λειτουργίας της συνάρτησης κατακερματισμού SHA-256, η οποία άλλωστε αποτελεί και ένα από τα συστατικά στοιχεία της εφαρμογής που έχει αναπτυχθεί για τις ανάγκες της παρούσας πτυχιακής εργασίας.

Η συνάρτηση SHA-256 είναι ικανή να δεχθεί ένα μήνυμα με μέγιστο μήκος που φτάνει μέχρι και τα $(2^{64} - 1)$ δυαδικά ψηφία, και από αυτό να παράγει μια σύνοψη των 256 δυαδικών ψηφίων.

Η επεξεργασία του μηνύματος γίνεται σε τμήματα των 512 δυαδικών ψηφίων. Όμως δεν παρέχεται καμία εγγύηση ότι το μήνυμα θα έχει μήκος που να είναι ακέραιο πολλαπλάσιο των 512 δυαδικών ψηφίων. Για αυτό και πριν από την επεξεργασία του μηνύματος προηγείται ένα στάδιο προετοιμασίας, κατά το οποίο το μήνυμα φτάνει στο επιθυμητό μήκος και στη συνέχεια χωρίζεται σε τμήματα των 512 bits.

Αφού ολοκληρωθεί το στάδιο προεπεξεργασίας του μηνύματος περνάμε στην κεντρική επεξεργασία, όπου το κάθε τμήμα του μηνύματος εισάγεται σύμφωνα με τη σειρά του στη συνάρτηση, περνάει από 64 γύρους πράξεων και μετασχηματισμών και παράγει μιαν ενδιάμεση τιμή σύνοψης. Μετά από την επεξεργασία όλων των τμημάτων του μηνύματος προκύπτει η τελική τιμή που θα λάβει η σύνοψη.

Με τη βοήθεια ενός παραδείγματος που περιέχεται στις προδιαγραφές FIPS PUB 180-2, όπου παρουσιάζεται ο υπολογισμός της σύνοψης του μηνύματος "abc", θα δούμε περισσότερο αναλυτικά τα δύο στάδια επεξεργασίας που περιλαμβάνονται στη συνάρτηση SHA-256.

<u>Η προεπεξεργασία του μηνύματος</u>

Το μήνυμα αποτελείται από τους τρεις χαρακτήρες "abc". Σε κάθε χαρακτήρα αντιστοιχούν 8 δυαδικά ψηφία, με την τιμή τους να διαμορφώνεται σύμφωνα με αυτή που αναλογεί από τον κώδικα ASCII.

Άρα το μήνυμα έχει μήκος $\ell = 3 * 8 = 24$ δυαδικά ψηφία, οπότε είναι απαραίτητη η χρησιμοποίηση ενισχυτικών δυαδικών ψηφίων(padding bits), με τα οποία το μήνυμα θα αποκτήσει μήκος που είναι ακέραιο πολλαπλάσιο των 512 bits.

Στο συγκεκριμένο παράδειγμα το αρχικό μήνυμα είναι κατά πολύ μικρότερο από ένα block των 512 bits, οπότε πρέπει να χρησιμοποιηθούν τα ανάλογα padding bits που θα φέρουν το μήνυμα στο μήκος του ενός block.

Η προσθήκη των padding bits γίνεται με τον ακόλουθο τρόπο:

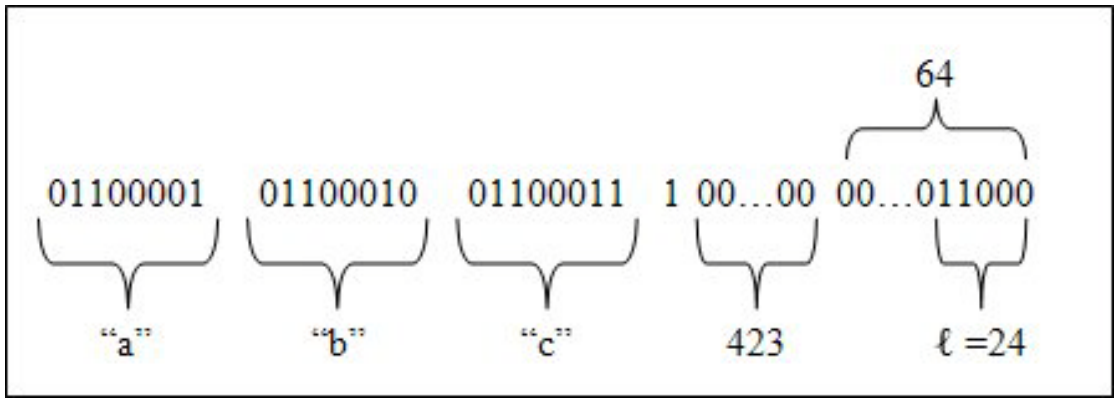
- Ένα δυαδικό ψηφίο με την τιμή "1" συνενώνεται στο τελευταίο block του αρχικού μηνύματος.

- Ακολουθεί μια σειρά από k δυαδικά ψηφία με την τιμή "0", όπου ο αριθμός k είναι η μικρότερη θετική λύση της εξίσωσης:
  $\ell + 1 + k \equiv 448 \mod 512$. Με αυτό τον τρόπο το τελευταίο block θα φτάσει στο μήκος των 448 bits.

  Για το συγκεκριμένο παράδειγμα έχουμε $k = 448 - (24 + 1) = 423$. Οπότε στο τέλος του μηνύματος επικολλώνται 423 bits με την τιμή "0".

- Σε αυτή τη φάση το τελευταίο τμήμα του μηνύματος υπολείπεται 64 bits από το επιθυμητό μήκος. Το συγκεκριμένο κενό καλύπτεται με τη δυαδική τιμή του ℓ καταχωρημένη 64 δυαδικά ψηφία.

Στο σχήμα που ακολουθεί έχουμε τη μορφή που λαμβάνει το μήνυμα μετά από την προσθήκη των ενισχυτικών δυαδικών ψηφίων.



Εικόνα 69: Η προσθήκη message padding

Στο παράδειγμα μας το μήνυμα είναι ίσο με 1 block των 512 bits, το οποίο συμβολίζεται ως $M^{(1)}$. Σε αυτή την περίπτωση δεν απαιτείται κάποια επιπλέον ενέργεια. Αν όμως το μήνυμα ήταν N φορές μεγαλύτερο, τότε θα χρειαζόταν η διαίρεση του σε αντίστοιχα N τμήματα των 512 bits, τα οποία θα συμβολίζονταν ως $M^{(1)}, M^{(2)}, \ldots, M^{(N)}$.

Το κάθε block μπορεί να εκφραστεί και ως μια σειρά από 16 λέξεις των 32 δυαδικών ψηφίων. Η πρώτη λέξη ενός τυχαίου block i συμβολίζεται ως $M_0^{(i)}$, τα αμέσως επόμενα 32 bits συμβολίζονται ως $M_1^{(i)}$, και συνεχίζουμε με αυτό το συμβολισμό για όλες τις υπόλοιπες λέξεις του block, μέχρι να φτάσουμε στην τελευταία λέξη που συμβολίζεται ως $M_{15}^{(i)}$.

Το τελευταίο βήμα της προεπεξεργασίας περιλαμβάνει την απόδοση της αρχικής τιμής $H^{(0)}$ στη σύνοψη, η οποία αποτελείται από 8 λέξεις των 32 bits με δεκαεξαδικές τιμές:

$$H_0^{(0)} = 6A09E667$$
$$H_1^{(0)} = BB67AE85$$
$$H_2^{(0)} = 3C6EF372$$
$$H_3^{(0)} = A54FF53A$$
$$H_4^{(0)} = 510E527F$$
$$H_5^{(0)} = 9B05688C$$
$$H_6^{(0)} = 1F83D9AB$$
$$H_7^{(0)} = 5BE0CD19$$

Οι τιμές των παραπάνω λέξεων προέρχονται από τις τετραγωνικές ρίζες των οκτώ πρώτων αριθμών από το 2 ως και το 19. Τα πρώτα 32 δυαδικά ψηφία του δεκαδικού μέρους της τετραγωνικής ρίζας απομονώνονται, πολλαπλασιάζονται με την τιμή 2^{32}, και το αποτέλεσμα μετατρέπεται σε δεκαεξαδικό αριθμό.

Ο υπολογισμός της σύνοψης

Αφού ολοκληρωθεί το στάδιο της προεπεξεργασίας έχει σχηματιστεί ένα σύνολο από N blocks των 512 δυαδικών ψηφίων που αντιπροσωπεύουν το μήνυμα.

Το κάθε ένα block, από το $M^{(1)}$ ως το $M^{(N)}$, εισέρχεται ξεχωριστά στη συνάρτηση και παράγει μιαν ενδιάμεση τιμή σύνοψης. Οπότε σύμφωνα με το μέγεθος του μηνύματος εκτελούνται από 1 ως N φορές τα ακόλουθα τέσσερα βήματα:

- Το τυχαίο block $M^{(i)}$ που εισάγεται στη συνάρτηση ακολουθεί ένα σχέδιο επέκτασης, σύμφωνα με το οποίο οι 16 λέξεις των 32 bits από τις οποίες αποτελείται τετραπλασιάζονται και γίνονται 64. Οι λέξεις του διευρυμένου τμήματος πληροφορίας συμβολίζονται ως W_0, W_1, …, W_{63}.

  Οι πρώτες 16 λέξεις, δηλαδή οι W_0, W_1, …, W_{15}, δημιουργούνται με την απευθείας αντιγραφή των λέξεων $M_0^{(i)}$, $M_1^{(i)}$, …, $M_{15}^{(i)}$, από τις οποίες αποτελείται το block $M^{(i)}$ που εισάγεται στη συνάρτηση.

  Οι υπόλοιπες 48 λέξεις, δηλαδή από W_{16} ως και W_{63}, προκύπτουν από μια σειρά πράξεις ανάμεσα στις τιμές των λέξεων που βρίσκονται σε προηγούμενες θέσεις. Έτσι, η λέξη που βρίσκεται σε μια τυχαία θέση t υπολογίζεται σύμφωνα με τον τύπο:

  $$W_t = \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16}$$

  Με το "+" συμβολίζεται η πρόσθεση ανάμεσα στα δυαδικά ψηφία των λέξεων, με το αποτέλεσμα που προκύπτει να υποβάλλεται στην πράξη mod με την τιμή 2^{32}.

  Με $\sigma_1^{\{256\}}(\ )$ και $\sigma_0^{\{256\}}(\ )$ συμβολίζονται δύο συναρτήσεις που εφαρμόζονται σε μια λέξη x και ορίζονται ως εξής:

  $$\sigma_0^{\{256\}}(x) = \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{SHR}^3(x)$$

$$\sigma_1^{\{256\}}(x) = \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{SHR}^{10}(x)$$

Μέσω της συνάρτησης $\mathrm{ROTR}^n(x)$ πραγματοποιείται κυκλική ολίσθηση των bits της λέξης *x* κατά n θέσεις προς τα δεξιά.

Η συνάρτηση $\mathrm{SHR}^n(x)$ πραγματοποιεί δεξιά ολίσθηση, όπου τα n bits που βρίσκονται δεξιά στη λέξη *x* απορρίπτονται, τα περισσότερο σημαντικά bits που απομένουν κινούνται προς τα δεξιά και το κενό που δημιουργείται στις θέσεις που κατείχαν πριν συμπληρώνεται με n bits με την τιμή "0".

Ανάμεσα στις λέξεις που επιστρέφουν από τις συναρτήσεις $\mathrm{ROTR}^n(\ )$ και $\mathrm{SHR}^n(\ )$ πραγματοποιείται η πράξη XOR, που συμβολίζεται με "⊕". Από εκεί προκύπτει μια τελική λέξη που είναι και η τιμή που αποδίδεται στις συναρτήσεις $\sigma_0^{\{256\}}(\ )$ και $\sigma_1^{\{256\}}(\ )$.

- Δίνονται αρχικές τιμές σε μια οκτάδα μεταβλητών λέξεων των 32 bits που θα βοηθήσουν στον υπολογισμό της σύνοψης.

  Σε κάθε μία από τις μεταβλητές, που συμβολίζονται ως *a, b, c, d, e, f, g* και *h*, καταχωρείται και μία από τις λέξεις που αποτελούν την τιμή σύνοψης $H^{(i-1)}$, με το *i* να λαμβάνει τιμές από 1 ως *N* σύμφωνα με το πλήθος των block που αντιστοιχούν στο μέγεθος του μηνύματος.

  Σύμφωνα με αυτό, οι μεταβλητές *a, b, …, h* λαμβάνουν τις τιμές που ακολουθούν:

  $$a = H_0^{(i-1)}$$
  $$b = H_1^{(i-1)}$$
  $$c = H_2^{(i-1)}$$
  $$d = H_3^{(i-1)}$$
  $$e = H_4^{(i-1)}$$
  $$f = H_5^{(i-1)}$$
  $$g = H_6^{(i-1)}$$
  $$h = H_7^{(i-1)}$$

  Οπότε για την πρώτη εκτέλεση, όπου έχουμε $i = 1$, οι μεταβλητές παίρνουν από μια από τις λέξεις της τιμής σύνοψης $H^{(0)}$ που συναντήσαμε στο προηγούμενο βήμα.

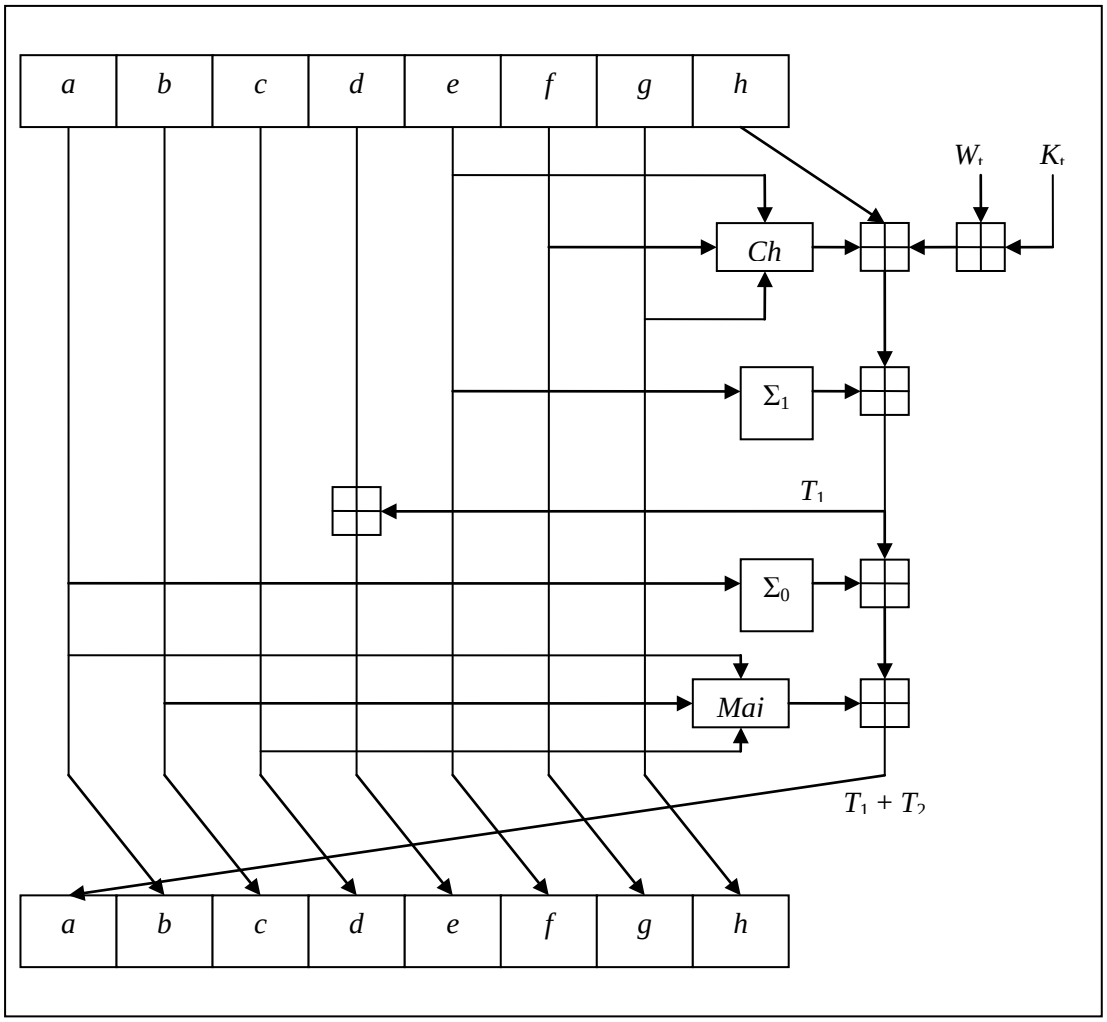
- Εκτελούνται 64 γύροι μετασχηματισμών, μέσα από τους οποίους συμπιέζεται το block που έχει εισαχθεί στη συνάρτηση.

  Οι μετασχηματισμοί εφαρμόζονται στα περιεχόμενα των οκτώ μεταβλητών που συναντήσαμε στο προηγούμενο βήμα, ενώ χρησιμοποιούνται και δύο επιπλέον προσωρινές μεταβλητές, που συμβολίζονται ως T_1 και T_2 και υποδέχονται τα αποτελέσματα μιας σειράς υπολογισμών.

  Συνοπτικά αναφέρουμε ότι σε κάθε γύρο πραγματοποιούνται σε 10 βήματα οι παρακάτω πράξεις:

1. $T_1 = h + \sum_1^{\{256\}}(e) + Ch(e, f, g) + K_t^{\{256\}} + W_t$
2. $T_2 = \sum_0^{\{256\}}(a) + Maj(a, b, c)$
3. $h = g$
4. $g = f$
5. $f = e$
6. $e = d + T_1$
7. $d = c$
8. $c = b$
9. $b = a$
10. $a = T_1 + T_2$

Στο σχήμα που ακολουθεί εμφανίζονται οι πράξεις που λαμβάνουν χώρα κατά τη διάρκεια μιας από τις επαναλήψεις της συμπίεσης.



Εικόνα 70: Μια επανάληψη της διαδικασίας συμπίεσης της SHA-256

Ας δούμε όμως αναλυτικότερα, βήμα προς βήμα, τη διαδικασία που ακολουθείται σε κάθε γύρο.

1. Ξεκινάμε με τον υπολογισμό της λέξης που θα αποθηκευτεί στην προσωρινή μεταβλητή T_1. Όπως είδαμε και παραπάνω, η τιμή της T_1 προκύπτει από τη σχέση που ακολουθεί:

$$T_1 = h + \sum_1^{\{256\}}(e) + Ch(e, f, g) + K_t^{\{256\}} + W_t$$

Με το σύμβολο "+" υποδηλώνεται η δυαδική πρόσθεση των λέξεων, ακολουθούμενη από την πράξη mod με την τιμή 2^{32}.

Η μεταβλητή *h* απλά περιέχει την τιμή της λέξης που της είχε αποδοθεί σε προηγούμενη φάση.

Η συνάρτηση $\sum_1^{\{256\}}( )$, εφαρμόζεται σε μια λέξη, έστω την *x*, στην οποία πραγματοποιεί μια τριάδα από δεξιές κυκλικές ολισθήσεις σύμφωνα με τη σχέση που ακολουθεί:

$$\sum_1^{\{256\}}(x) = \text{ROTR}^6(x) \oplus \text{ROTR}^{11}(x) \oplus \text{ROTR}^{25}(x)$$

Στις λέξεις που επιστρέφουν από τη συνάρτηση $\text{ROTR}^n( )$ εφαρμόζεται η πράξη XOR, έτσι καταλήγουμε στην ύπαρξη μιας λέξης που θα επιστρέψει από τη συνάρτηση $\sum_1^{\{256\}}( )$.

Η συνάρτηση *Ch*( ), εφαρμόζεται σε τρεις λέξεις, έστω ότι αυτές είναι οι *x*, *y* και z, στις οποίες πραγματοποιεί μια σειρά πράξεων ανάμεσα στα δυαδικά ψηφία τους ακολουθώντας τον τύπο:

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z)$$

Με το σύμβολο "^" αντιπροσωπεύεται η δυαδική πράξη AND ανάμεσα σε δύο λέξεις, ενώ με το "¬" συμβολίζεται το δυαδικό συμπλήρωμα μιας λέξης.

Οι δύο λέξεις που προκύπτουν από τις πράξεις AND εισάγονται σε μία πράξη XOR. Με αυτό τον τρόπο η συνάρτηση *Ch*( ) επιστρέφει μία λέξη.

Η μεταβλητή $K_t^{\{256\}}$ περιέχει την τιμή μίας λέξης που αντλείται από μια ακολουθία με 64 σταθερές τιμές. Οι τιμές αυτές συμβολίζονται ως $K_0^{\{256\}}$, $K_1^{\{256\}}$, …, $K_{63}^{\{256\}}$ και υπολογίζονται από την κυβική ρίζα των 64 πρώτων αριθμών που βρίσκονται ανάμεσα στο 2 ως και το 311. Τα πρώτα 32 δυαδικά ψηφία του δεκαδικού μέρους της κυβικής ρίζας απομονώνονται, πολλαπλασιάζονται με την τιμή 2^{32}, το αποτέλεσμα μετατρέπεται σε δεκαεξαδική μορφή και αποθηκεύεται στην $K_t^{\{256\}}$ που του αντιστοιχεί.

Η μεταβλητή W_t είναι ίση με μία από τις 64 λέξεις, οι οποίες συμβολίζονται ως W_0, W_1, …, W_{63} και υπολογίστηκαν κατά τη διάρκεια της διαδικασίας επέκτασης του block του μηνύματος, που θυμίζουμε ότι έλαβε χώρα κατά το πρώτο βήμα του υπολογισμού της σύνοψης.

2. Σε αυτή τη φάση υπολογίζεται η τιμή της προσωρινής μεταβλητής T_2, η λέξη που αποθηκεύεται στην T_2 προκύπτει από τη σχέση:

$$T_2 = \Sigma_0^{\{256\}}(a) + Maj(a, b, c)$$

Η συνάρτηση $\Sigma_0^{\{256\}}( )$, εφαρμόζεται σε μια λέξη, έστω την x, στην οποία πραγματοποιεί μια τριάδα από δεξιές κυκλικές ολισθήσεις σύμφωνα με τη σχέση που ακολουθεί:

$$\Sigma_0^{\{256\}}(x) = \text{ROTR}^2(x) \oplus \text{ROTR}^{13}(x) \oplus \text{ROTR}^{22}(x)$$

Ανάμεσα στις λέξεις που επιστρέφουν από τη συνάρτηση $\text{ROTR}^n( )$ εφαρμόζεται η πράξη XOR. Με αυτή την ενέργεια δημιουργείται μια λέξη που επιστρέφεται με τη συνάρτηση $\Sigma_0^{\{256\}}( )$.

Η συνάρτηση $Maj( )$, εφαρμόζεται σε τρεις λέξεις, έστω ότι αυτές είναι οι x, y και z, στις οποίες πραγματοποιεί τις πράξεις AND και XOR σύμφωνα με τον ακόλουθο τύπο:

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

Από τη συνάρτηση $Maj( )$ επιστρέφεται μια λέξη που εισάγεται σε δυαδική πρόσθεση με τη λέξη που επιστρέφεται από τη συνάρτηση $\Sigma_0^{\{256\}}( )$. Στο αποτέλεσμα της δυαδικής πρόσθεσης πραγματοποιείται η πράξη mod με την τιμή 2^{32}. Ότι απομένει αποθηκεύεται στην προσωρινή μεταβλητή T_2.

3. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή g αποθηκεύεται στη μεταβλητή h.

4. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή f αποθηκεύεται στη μεταβλητή g.

5. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή e αποθηκεύεται στη μεταβλητή f.

6. Πραγματοποιείται η πράξη της δυαδικής πρόσθεσης ανάμεσα στη λέξη της μεταβλητής d και την τιμή της προσωρινής μεταβλητής T_1. Στο αποτέλεσμα εφαρμόζεται η πράξη mod με την τιμή 2^{32} και το υπόλοιπο της πράξης αποθηκεύεται στη μεταβλητή e.

7. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή c αποθηκεύεται στη μεταβλητή d.

8. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή b αποθηκεύεται στη μεταβλητή c.

9. Η λέξη που βρίσκεται αποθηκευμένη στη μεταβλητή *a* αποθηκεύεται στη μεταβλητή *b*.

10. Πραγματοποιείται η πράξη της δυαδικής πρόσθεσης ανάμεσα στις λέξεις των προσωρινών μεταβλητών T_1 και T_2. Στο αποτέλεσμα εφαρμόζεται η πράξη mod με την τιμή 2^{32} και το υπόλοιπο της πράξης αποθηκεύεται στη μεταβλητή *a*.

- Υπολογίζεται η ενδιάμεση τιμή της σύνοψης που αντιστοιχεί στο τμήμα πληροφορίας που έχει εισαχθεί στη συνάρτηση.

  Η ενδιάμεση τιμή σύνοψης του τυχαίου block *i* ισούται με την τιμή που προκύπτει από τη συνένωση των τιμών των λέξεων $H_0^{(i)}$, $H_1^{(i)}$, …, $H_7^{(i)}$. Η συνένωση συμβολίζεται με "||", οπότε η σύνοψη αναπαρίσταται με τον εξής τρόπο:

  $$H_0^{(i)} \,||\, H_1^{(i)} \,||\, H_2^{(i)} \,||\, H_3^{(i)} \,||\, H_4^{(i)} \,||\, H_5^{(i)} \,||\, H_6^{(i)} \,||\, H_7^{(i)}$$

  Ο υπολογισμός αυτών των λέξεων γίνεται με τον τρόπο που ακολουθεί:

  $$\begin{aligned} H_0^{(i)} &= a + H_0^{(i-1)} \\ H_1^{(i)} &= b + H_1^{(i-1)} \\ H_2^{(i)} &= c + H_2^{(i-1)} \\ H_3^{(i)} &= d + H_3^{(i-1)} \\ H_4^{(i)} &= e + H_4^{(i-1)} \\ H_5^{(i)} &= f + H_5^{(i-1)} \\ H_6^{(i)} &= g + H_6^{(i-1)} \\ H_7^{(i)} &= h + H_7^{(i-1)} \end{aligned}$$

  Οι μεταβλητές a, b, …, h συμμετέχουν σε δυαδική πρόσθεση με τις αντίστοιχες λέξεις $H_0^{(i-1)}$, $H_1^{(i-1)}$, …, $H_7^{(i-1)}$, που προέρχονται από την ενδιάμεση σύνοψη του προηγούμενου block του μηνύματος. Τα αποτέλεσμα τους ακολουθείται από την πράξη mod με την τιμή 2^{32} και το υπόλοιπο αποθηκεύεται στις αντίστοιχες λέξεις $H_0^{(i)}$, $H_1^{(i)}$, …, $H_7^{(i)}$.

Μετά από *N* επαναλήψεις των παραπάνω τεσσάρων βημάτων έχουν επεξεργαστεί όλα τα τμήματα του μηνύματος *M*. Οι λέξεις $H_0^{(N)}$, $H_1^{(N)}$, …, $H_7^{(N)}$, περιέχουν την ενδιάμεση τιμή σύνοψης του τελευταίου block του μηνύματος. Η τελική τιμή σύνοψης με τα 256 δυαδικά ψηφία προκύπτει από τη συνένωση της συγκεκριμένης οκτάδας λέξεων των 32 bits. Η συνένωση συμβολίζεται με "||", οπότε η σύνοψη αναπαρίσταται με τον εξής τρόπο:

$$H^{(N)} = H_0^{(N)} \,||\, H_1^{(N)} \,||\, H_2^{(N)} \,||\, H_3^{(N)} \,||\, H_4^{(N)} \,||\, H_5^{(N)} \,||\, H_6^{(N)} \,||\, H_7^{(N)}$$

Στο παράδειγμα με το μήνυμα "abc" έχουμε 1 block των 512 δυαδικών ψηφίων. Οπότε τα τέσσερα βήματα που απαιτούνται για τον υπολογισμό της σύνοψης θα εκτελεστούν 1 φορά.

Άρα, η τελική τιμή της σύνοψης $H^{(1)}$ προκύπτει από τη συνένωση των οκτώ λέξεων που ακολουθούν:

$$H_0^{(1)} = a + H_0^{(0)}$$

$$H_1^{(1)} = b + H_1^{(0)}$$
$$H_2^{(1)} = c + H_2^{(0)}$$
$$H_3^{(1)} = d + H_3^{(0)}$$
$$H_4^{(1)} = e + H_4^{(0)}$$
$$H_5^{(1)} = f + H_5^{(0)}$$
$$H_6^{(1)} = g + H_6^{(0)}$$
$$H_7^{(1)} = h + H_7^{(0)}$$

Που ισοδυναμούν με:

$$H_0^{(1)} = 506E3058 + 6A09E667 = BA7816BF$$
$$H_1^{(1)} = D39A2165 + BB67AE85 = 8F01CFEA$$
$$H_2^{(1)} = 04D24D6C + 3C6EF372 = 414140DE$$
$$H_3^{(1)} = B85E2CE9 + A54FF53A = 5DAE2223$$
$$H_4^{(1)} = 5EF50F24 + 510E527F = B00361A3$$
$$H_5^{(1)} = FB121210 + 9B05688C = 96177A9C$$
$$H_6^{(1)} = 948D25B6 + 1F83D9AB = B410FF61$$
$$H_7^{(1)} = 961F4894 + 5BE0CD19 = F20015AD$$

Έτσι έχουμε την τελική τιμή σύνοψης του μηνύματος "abc", που ισούται με:

$$H^{(1)} = BA7816BF\ 8F01CFEA\ 414140DE\ 5DAE2223\ B00361A3\ 96177A9C\ B410FF61\ F20015AD$$

4.4.3 Η χρήση συναρτήσεων κατακερματισμού για την παραγωγή συμμετρικών κλειδιών

Όπως είδαμε και νωρίτερα, η παραγωγή συμμετρικών κρυπτογραφικών κλειδιών είναι μία από τις πολλές εφαρμογές που αξιοποιούν την ύπαρξη των συναρτήσεων κατακερματισμού.

Ένας συνηθισμένος τρόπος παραγωγής κλειδιών είναι με μία από τις συναρτήσεις της οικογένειας Password Based Key Derivation Function(PBKDF), με το παραγόμενο κλειδί(Derived Key) να προκύπτει από την τιμή σύνοψης που δίνει μια συνάρτηση κατακερματισμού που λειτουργεί στο εσωτερικό της συνάρτησης PBKDF.

Στην οικογένεια των συναρτήσεων PBKDF ανήκουν οι:

- PBKDF1. Είναι η παλαιότερη συνάρτηση αυτής της οικογένειας. Οι συναρτήσεις κατακερματισμού που χρησιμοποιούνται στο εσωτερικό της είναι οι MD2, MD5 και SHA-1. Το παραγόμενο κλειδί είναι δυνατό να έχει μήκος είτε 128 δυαδικών ψηφίων, στην περίπτωση των MD2 και MD5, είτε 160 δυαδικών ψηφίων, στην περίπτωση που χρησιμοποιείται η SHA-1.

  Η συνάρτηση PBKDF1 δεν συνίσταται για τη χρήση της σε σύγχρονες εφαρμογές, αφού τα κλειδιά που παράγονται μέσω αυτής έχουν περιορισμένο μήκος. Έτσι, διατηρείται μόνο για λόγους συμβατότητας με ήδη υπάρχουσες εφαρμογές στις οποίες χρησιμοποιείται η PBKDF1.

- PBKDF2. Είναι το νεότερο μέλος της οικογένειας των συναρτήσεων PBKDF. Παρέχει αρκετές επιλογές σε σχέση με την ενσωματωμένη συνάρτηση κατακερματισμού, καθώς δίνεται η δυνατότητα επιλογής μιας εκ των SHA-1, SHA-224, SHA-256, SHA-384 και SHA-512.

  Το μήκος του παραγόμενου κλειδιού εξαρτάται άμεσα από τη συνάρτηση κατακερματισμού που έχει επιλεγεί για την εσωτερική λειτουργία της συνάρτησης PBKDF2.

Η συνάρτηση παραγωγής κλειδιών PBKDF2, σε συνδυασμό με τη συνάρτηση κατακερματισμού SHA-256, αποτελούν τις επιλογές μας για την εφαρμογή που αναπτύχθηκε για τις ανάγκες της παρούσας πτυχιακής εργασίας. Γι' αυτό και σε αυτό το σημείο θα εστιάσουμε στον τρόπο με τον οποίο λειτουργεί η συνάρτηση PBKDF2, όπου αρχικά θα δούμε το γενικό τρόπο λειτουργίας της συνάρτησης, ενώ στη συνέχεια θα δούμε πως λειτουργεί με τη συνάρτηση κατακερματισμού SHA-256.

Ο τρόπος λειτουργίας της συνάρτησης PBKDF2

Η διαδικασία παραγωγής κλειδιών μέσω της συνάρτησης PBKDF2 ξεκινά με την εισαγωγή μιας σειράς παραμέτρων σε αυτήν.

Η είσοδος της συνάρτησης PBKDF2 αποτελείται από τις εξής παραμέτρους:

- τη συνθηματική λέξη(Password) ή συνθηματική φράση(Pass Phrase), που συμβολίζεται ως P.
- την αριθμητική τιμή Salt, με το χαρακτήρα S να αποτελεί τον τρόπο με τον οποίο συμβολίζεται η συγκεκριμένη παράμετρος.
- το μετρητή επαναλήψεων(Iteration Count), ο οποίος συμβολίζεται με το χαρακτήρα c.
- το επιθυμητό μήκος του παραγόμενου κλειδιού, εκφρασμένο σε octets, συμβολίζεται ως $dkLen$.

Συνηθίζεται σε μια εφαρμογή να διατηρούνται σταθερές οι παράμετροι με τις τιμές των επαναλήψεων και με το επιθυμητό μήκος του παραγόμενου κλειδιού. Αντιθέτως, η συνθηματική λέξη και το Salt έχουν τη δυνατότητα αλλαγής, όταν αυτό βέβαια θεωρείται αναγκαίο.

Η συνάρτηση PBKDF2 εκτός από το προφανές, που είναι η δημιουργία κλειδιών, παρέχει δύο πολύ χρήσιμα χαρακτηριστικά που δεν θα ήταν εφικτά χωρίς τις παραμέτρους εισόδου:

- Την επέκταση του παραγόμενου κλειδιού(Key Stretching). Οι παράμετροι εισόδου της PBKDF2 έχουν μήκος μερικών δεκάδων δυαδικών ψηφίων και διαμορφώνουν την τιμή του παραγόμενου κλειδιού, το οποίο όμως φτάνει σε μήκος μερικών εκατοντάδων δυαδικών ψηφίων.

- Την ενδυνάμωση του παραγόμενου κλειδιού(Key Strengthening). Οι παράμετροι που εισάγονται στη συνάρτηση PBKDF2 οδηγούν στη δημιουργία ανθεκτικών κλειδιών. Ενδεικτικά να αναφέρουμε, ότι με την εκτέλεση ενός

μεγάλου αριθμού επαναλήψεων μια επίθεση τύπου Brute-Force γίνεται υπολογιστικά ανέφικτη. Ενώ η προσθήκη του Salt, αποτρέπει τις επιθέσεις όπου έχει σχηματιστεί ένα «λεξικό» από συνθηματικά (Dictionary Attack), αφού μια μικρή αλλαγή της τιμής του Salt καθιστά το «λεξικό» άχρηστο.

Συνήθως, και πάντα ανάλογα με τις απαιτήσεις, πραγματοποιούνται τουλάχιστον 1000 επαναλήψεις. Η μέγιστη τιμή που μπορεί να τεθεί ως αριθμός επαναλήψεων είναι κοντά στις 65000.

Μετά από την εισαγωγή των παραμέτρων περνάμε στον προσδιορισμό της τιμής του παραγόμενου κλειδιού. Η συγκεκριμένη διαδικασία πραγματοποιείται στα πέντε βήματα που ακολουθούν και εμφανίζουν το γενικό τρόπο προσδιορισμού του παραγόμενου κλειδιού:

1. Ελέγχεται αν ισχύει η παρακάτω ανισότητα:
$$dkLen < (2^{32} - 1) \times hLen$$
Με $hLen$ αντιπροσωπεύεται το πλήθος των octets από τα οποία αποτελείται η σύνοψη που παράγεται από τη χρησιμοποιούμενη συνάρτηση κατακερματισμού.

Στην περίπτωση που η παραπάνω ανισότητα δεν ισχύει, τότε η διαδικασία δημιουργίας ενός κλειδιού δεν προχωρά.

2. Θεωρούμε ως l τον αριθμό, με προσέγγιση στον επόμενο ακέραιο, των $hLen$ blocks από τα οποία αποτελείται το $dkLen$, ενώ ως r συμβολίζεται το πλήθος των octets που βρίσκονται στο τελευταίο block. Με άλλα λόγια, έχουμε:
$$l = dkLen \,/\, hLen$$
$$r = dkLen - [(l - 1) \times hLen]$$

3. Σε κάθε block εφαρμόζεται η συνάρτηση F, με το αποτέλεσμα της αποθηκεύεται σε μια προσωρινή μεταβλητή T:
$$T_1 = F\ (P,\ S,\ c,\ 1),$$
$$T_2 = F\ (P,\ S,\ c,\ 2),$$
$$\ldots,$$
$$T_l = F\ (P,\ S,\ c,\ l).$$
Όπου η συνάρτηση F, ορίζεται ως η πράξη XOR ανάμεσα στις c επαναλήψεις της χρησιμοποιούμενης συνάρτησης hash που εφαρμόζεται στο P και τη συνένωση του S με το δείκτη του block, τον οποίο συμβολίζουμε ως i:
$$F\ (P,\ S,\ c,\ i) = U_1 \text{ XOR } U_2 \text{ XOR } \ldots \text{ XOR } U_c,$$
Με την προσωρινή μεταβλητή U να δίνεται μέσα από τη σχέση:
$$U_1 = \text{hash}\ (P,\ S \,||\, \text{INT}(i)),$$
$$U_2 = \text{hash}\ (P,\ U_1),$$
$$\ldots,$$
$$U_c = \text{hash}\ (P,\ U_{c-1}).$$

Όπου με INT(*i*) συμβολίζεται η κωδικοποίηση του δείκτη *i* σε τέσσερα octets.

4. Οι τιμές που έχουν προκύψει από την εφαρμογή της *F* σε κάθε block συνενώνονται. Με την απόσπαση των πρώτων *dkLen* octets σχηματίζεται το παραγόμενο κλειδί, το οποίο συμβολίζεται ως *DK*. Οπότε έχουμε:

$$DK = T_1 \,||\, T_2 \,||\, \ldots \,||\, T_l\langle 0 \ldots r-1 \rangle.$$

Όπου με <0…*r*-1> συμβολίζεται η απόσπαση των octets από 0 μέχρι και *r*-1.

5. Ο προσδιορισμός του παραγόμενου κλειδιού έχει ολοκληρωθεί, οπότε η τιμή του *DK* επιστρέφεται από την κλήση της συνάρτησης PBKDF2 και το κλειδί είναι στη διάθεση του χρήστη.

Ας δούμε όμως πως θα εκτελεστούν τα παραπάνω πέντε βήματα όταν η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η SHA-256, και το παραγόμενο κλειδί έχει μήκος 256 bits. Έτσι λοιπόν, έχουμε:

1. Τον έλεγχο της παρακάτω ανισότητας:

$$dkLen < (2^{32} - 1) \times hLen$$

Η οποία ισχύει, καθώς $dkLen = hLen = 32$

2. $l = dkLen / hLen = 32 / 32 = 1$
$r = dkLen - [(l - 1) \times hLen] = 32 - [(1-1) \times 32] = 32$

3. Η συνάρτηση *F* εφαρμόζεται σε μόνο ένα block, και το αποτέλεσμα της αποθηκεύεται στην προσωρινή μεταβλητή T_1:

$$T_1 = F\ (P,\ S,\ c,\ 1),$$

Όπου για τη συνάρτηση *F* έχουμε:

$$F\ (P,\ S,\ c,\ 1) = U_1 \text{ XOR } U_2 \text{ XOR } \ldots \text{ XOR } U_c$$

Με την προσωρινή μεταβλητή *U* να δίνεται μέσα από τη σχέση:

$$U_1 = \text{sha256}\ (P,\ S \,||\, \text{INT}\ (1)),$$
$$U_2 = \text{sha256}\ (P,\ U_1),$$
$$\ldots,$$
$$U_c = \text{sha256}\ (P,\ U_{c-1}).$$

4. Το παραγόμενο κλειδί έχει τη μορφή:

$$DK = T_1\ \langle 0 \ldots r-1 \rangle.$$

5. Η τιμή του *DK* επιστρέφεται από την κλήση της συνάρτησης PBKDF2 και το κλειδί είναι στη διάθεση του χρήστη.

4.4.4 Κρυπτανάλυση των συναρτήσεων κατακερματισμού

Όπως αναφέραμε και νωρίτερα, οι συναρτήσεις κατακερματισμού χρησιμοποιούνται ευρύτατα σε μια πληθώρα διαφορετικών εφαρμογών, αποτελώντας κατά αυτό τον τρόπο ένα από τα αντικείμενα μελέτης της ερευνητικής κοινότητας.

Τα μέλη της οικογένειας των συναρτήσεων κατακερματισμού SHA έχουν συγκεντρώσει και αυτά με τη σειρά τους αρκετό ενδιαφέρον από ερευνητές που εξετάζουν τις δυνατότητες κρυπτανάλυσης του συγκεκριμένου τομέα.

Έτσι, ξεκινώντας με βάση τη χρονολογία εμφάνισης μιας συνάρτησης της εν λόγω οικογένειας, να αναφέρουμε ότι η SHA-1 έχει αποδειχθεί ότι έχει κάποιες αδυναμίες. Πιο συγκεκριμένα, είναι δυνατή η εύρεση συγκρούσεων για την SHA-1 μετά από 2^{69} υπολογισμούς[1].

Το συγκεκριμένο στοιχείο οδήγησε το NIST στην πρόταση της σταδιακής εγκατάλειψης της συνάρτησης κατακερματισμού SHA-1 και την αντικατάσταση της με τις παραλλαγές της νεώτερης συνάρτησης SHA-2.

Όπου για την εν λόγω συνάρτηση δεν έχει βρεθεί κάποια επίθεση που να αποκαλύπτει σε πλήρη βαθμό κάποιο αδύναμο σημείο της, όμως έχουν αναφερθεί δύο περιπτώσεις όπου βρέθηκαν συγκρούσεις σε τροποποιημένες εκδοχές της αρχικής συνάρτησης.

Η πρώτη περίπτωση όπου αναφέρθηκαν συγκρούσεις σε τροποποιημένες εκδοχές της SHA-2 προήλθε από τους Jian Guo και Krystian Matusiewicz[2], οι οποίοι ανακάλυψαν συγκρούσεις σε μια συνάρτηση SHA-256 που είχε 42 γύρους μετασχηματισμών έναντι των 64 γύρων που έχει η πλήρης εκδοχή της. Σύμφωνα τους δύο ερευνητές, η ίδια επίθεση μπορεί να εφαρμοστεί και στη συνάρτηση SHA-512.

Η δεύτερη αναφορά συγκρούσεων σε τροποποιημένες συναρτήσεις SHA-2 αποτελεί έργο των Yu Sasaki, Lei Wang και Kazumaro Aoki[3]. Η συγκεκριμένη ομάδα ερευνητών παρουσίασε τη δυνατότητα επίθεσης σε μια συνάρτηση SHA-256 με 41 γύρους μετασχηματισμών, αλλά παράλληλα έδειξε ότι είναι εφικτή και η επίθεση σε μια συνάρτηση SHA-512 προσαρμοσμένη σε 46 γύρους μετασχηματισμών, όταν οι πλήρεις εκδοχές των δύο συναρτήσεων κατακερματισμού έχουν, αντίστοιχα, 64 και 80 γύρους μετασχηματισμών.

Το γεγονός αυτό μας αποδεικνύει ότι οι συναρτήσεις κατακερματισμού SHA-2 συνεχίζουν να θεωρούνται ασφαλείς για χρήση και να λειτουργούν ικανοποιητικά, όμως παράλληλα είναι εμφανές ότι το θεωρητικό περιθώριο της παρεχόμενης ασφάλειας είναι μικρότερο από αυτό που είχε αρχικά εκτιμηθεί κατά τη φάση της δημιουργίας των συγκεκριμένων συναρτήσεων κατακερματισμού.

[1] Η επίθεση βρίσκεται δημοσιευμένη στην ιστοσελίδα http://people.csail.mit.edu/yiqun/SHA1AttackProceedingVersion.pdf
[2] Η επίθεση βρίσκεται δημοσιευμένη στην ιστοσελίδα http://eprint.iacr.org/2009/477.pdf
[3] Η επίθεση βρίσκεται δημοσιευμένη στην ιστοσελίδα http://eprint.iacr.org/2009/479.pdf

Πάντως, εδώ και κάποια χρόνια έχει ξεκινήσει από το NIST μια διαδικασία, που είναι παρόμοια με αυτή που ακολουθήθηκε για το AES, κατά την οποία κατατίθενται υποψήφιες προτάσεις για την ανάπτυξη μιας νέας συνάρτησης κατακερματισμού, της SHA-3.

Η αξιολόγηση και η τελική επιλογή του αλγορίθμου που θα υιοθετηθεί ως η συνάρτηση κατακερματισμού τρίτης γενιάς SHA-3 αναμένεται, σύμφωνα με το διαδικτυακό χώρο της εν λόγω διαδικασίας[*], να ολοκληρωθεί το 2012.

[*] http://csrc.nist.gov/groups/ST/hash/index.html

Κεφάλαιο 5 - Ασφαλής μετάδοση σύντομων μηνυμάτων

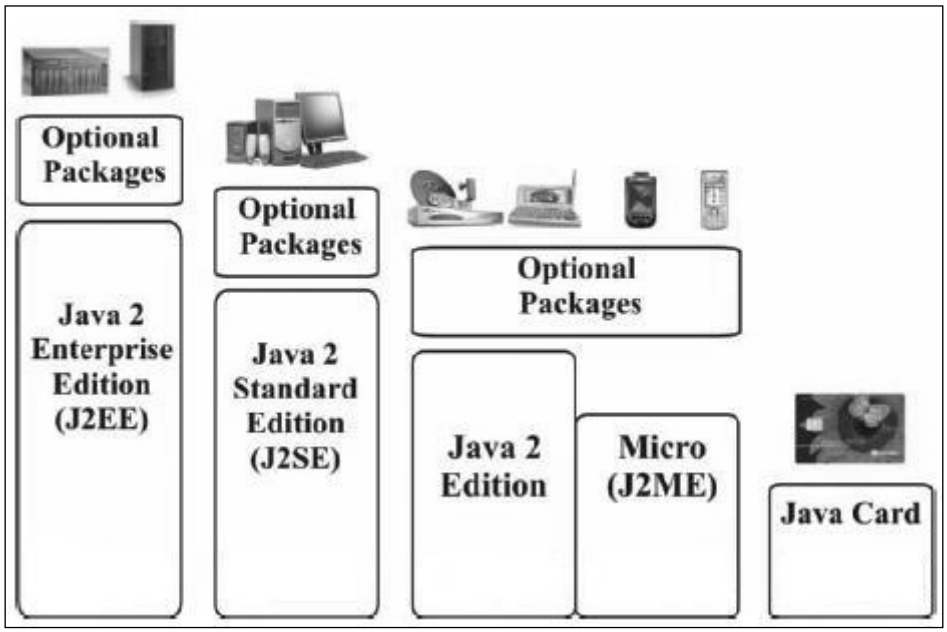
Σε αυτό το κεφάλαιο παρουσιάζεται το περιβάλλον Java Micro Edition, μέσω του οποίου δίνεται η δυνατότητα εκτέλεσης της εφαρμογής που έχει αναπτυχθεί. Στη συνέχεια καθορίζονται οι απαιτήσεις που υπάρχουν από το περιβάλλον εκτέλεσης της εφαρμογής. Ενώ μετά παρουσιάζεται αναλυτικά ο τρόπος με τον οποίο λειτουργεί η εφαρμογή και καθιστά εφικτή την ασφαλή μετάδοση σύντομων μηνυμάτων.

5.1 Java Micro Edition

Η τεχνολογία Java παρέχει ένα αντικειμενοστραφές περιβάλλον ανάπτυξης για εφαρμογές που είναι ικανές να εκτελούνται σε μια πληθώρα από διαφορετικές συσκευές, από εξυπηρετητές μέχρι και έξυπνες κάρτες, ενώ εκτελούνται ανεξάρτητα από την πλατφόρμα με την οποία λειτουργεί κάθε συσκευή. Άλλωστε, το σύνθημα που υπηρετεί η Java είναι το "write once, run anywhere", δηλαδή, μια εφαρμογή πρέπει να δημιουργείται μια φορά και να εκτελείται σε οποιαδήποτε συσκευή.

Όμως κάθε συσκευή έχει ξεχωριστές δυνατότητες, για παράδειγμα ένας server είναι σίγουρα πιο ισχυρός από ένα κινητό τηλέφωνο. Γι' αυτό και είναι αρκετά χρήσιμος ο διαχωρισμός της Java σε τέσσερα επίπεδα, ανάλογα με τις δυνατότητες που παρέχονται από κάθε συσκευή. Με αυτό τον τρόπο παράγονται εφαρμογές που είναι προσαρμοσμένες στην εκάστοτε συσκευή, είναι αποδοτικότερες, και ικανοποιούν καλύτερα τις διαφορετικές ανάγκες που υπάρχουν σε κάθε περίσταση.

Στην εικόνα που ακολουθεί παρουσιάζεται όλο το εύρος των εκδόσεων της Java, ενώ παράλληλα απεικονίζονται και μερικές από τις συσκευές για τις οποίες προορίζονται οι παραγόμενες εφαρμογές κάθε μιας από τις εκδόσεις.



Εικόνα 71: Η οικογένεια της Java

Η Java χωρίζεται στις εκδόσεις:

- Java Enterprise Edition (Java EE ή J2EE). Η συγκεκριμένη έκδοση έχει σαν στόχο την ανάπτυξη κατανεμημένων εταιρικών εφαρμογών και δίνει έμφαση στην server-side ανάπτυξη και στις εφαρμογές web.

- Java Standard Edition (Java SE ή J2SE). Προορίζεται για την ανάπτυξη συμβατικών desktop εφαρμογών.

- Java Micro Edition (Java ME ή J2ME). Αποτελεί υποσύνολο της J2SE, προορίζεται για συσκευές με περιορισμένους πόρους που δεν υποστηρίζουν πλήρως μιαν υλοποίηση J2SE, τέτοιες περιπτώσεις είναι οι φορητές συσκευές και τα ενσωματωμένα συστήματα(Embedded Systems).

- Java Card. Παρέχει ένα περιβάλλον ανάπτυξης εφαρμογών που θα εκτελούνται σε «έξυπνες κάρτες» (Smart Cards).

Όμως στην περίπτωση της J2ME η κατάσταση είναι περισσότερο πολύπλοκη, αφού η συγκεκριμένη εκδοχή της Java στοχεύει σε πολλές συσκευές με ανόμοια χαρακτηριστικά και σημαντικές διαφορές σε τομείς όπως η επεξεργαστική ισχύς και η διαθέσιμη μνήμη.

Αυτό το γεγονός καθιστά απαραίτητη την ύπαρξη μιας μεθόδου διαχωρισμού των συσκευών που υποστηρίζουν την J2ME σε δύο κατηγορίες, με τη μια από αυτές να περιλαμβάνει τις ισχυρές συσκευές, και την άλλη να αντιπροσωπεύει τις λιγότερο ισχυρές.

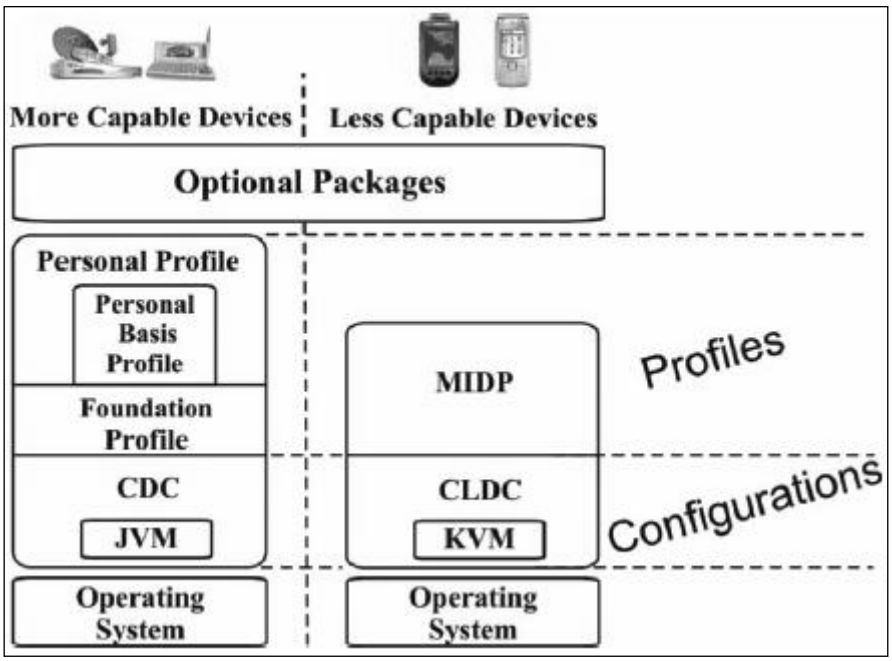
Ο διαχωρισμός των συσκευών εφαρμόζεται στην πράξη μέσα από τη χρησιμοποίηση δύο διαρρυθμίσεων(configurations) της J2ME, οι οποίες λειτουργούν πάνω από το λειτουργικό σύστημα(operating system) της εκάστοτε συσκευής, και παρέχουν ένα βασικό σύνολο με βιβλιοθήκες κλάσεων και APIs. Ακόμη, ένα configuration παρέχει μιαν εικονική μηχανή(Virtual Machine - VM) που ταιριάζει στις δυνατότητες της συσκευής και βοηθά στην ερμηνεία και την εκτέλεση των εφαρμογών.

Σίγουρα τα configurations της J2ME είναι αρκετά σημαντικά, όμως δεν παύουν να είναι αρκετά γενικά και συχνά να μην αρκούν για την ανάπτυξη κάποιας εφαρμογής. Έτσι, με βάση αυτά τα configurations δημιουργούνται περισσότερο εξειδικευμένες κλάσεις και APIs που ομαδοποιούνται σε συγκεκριμένα προφίλ(profiles), με τα οποία παρέχονται πρόσθετα στοιχεία που ταιριάζουν περισσότερο με τις δυνατότητες κάθε συσκευής και είναι αρκετά χρήσιμα για την ανάπτυξη αποδοτικότερων εφαρμογών.

Αν και μετά από την προσθήκη κάποιου profile υπάρχει η ανάγκη για κάποια επιπλέον λειτουργία, τότε η λύση δίνεται με την προσθήκη προαιρετικών πακέτων(optional packages) κλάσεων και APIs πάνω από το συνδυασμό configuration – profile που χρησιμοποιείται.

Ο συνδυασμός ενός configuration με ένα ή περισσότερα profiles και μια σειρά optional packages θεωρείται ότι σχηματίζει μια στοίβα πάνω από το λειτουργικό σύστημα της εκάστοτε συσκευής, η οποία αντιπροσωπεύει το ολοκληρωμένο περιβάλλον εκτέλεσης εφαρμογών J2ME(J2ME Runtime Environment).

Στην εικόνα που ακολουθεί παρουσιάζεται ο τρόπος με τον οποίο δομείται η αρχιτεκτονική J2ME και σχηματίζεται το αντίστοιχο Runtime Environment για τις περισσότερο και τις λιγότερο ικανές συσκευές.



Εικόνα 72: Η αρχιτεκτονική J2ME

Όπως φαίνεται και από το παραπάνω σχήμα, μέσω της πλατφόρμας J2ME παρέχονται δύο configurations που συνεργάζονται με συγκεκριμένα profiles, πρόκειται για τα:

- Connected Limited Device Configuration (CLDC). Το συγκεκριμένο configuration συνεργάζεται με το profile Mobile Information Device Profile(MIDP) και προορίζεται για τις λιγότερο ικανές συσκευές που χαρακτηρίζονται από:
    - Πολύ απλό User Interface
    - Ελάχιστη ποσότητα διαθέσιμης μνήμης για εφαρμογές Java, από 160Kb μέχρι 512Kb
    - Ασύρματη επικοινωνία, χαμηλού εύρους ζώνης και χωρίς να είναι πάντα σταθερή
    - Επεξεργαστές 16 ή 32 bit
    - Περιορισμένη ενέργεια, συνήθως τροφοδοτούνται από κάποια μπαταρία

  Η εικονική μηχανή που παρέχεται από το CLDC είναι η Kilo Virtual Machine(KVM). Πρόκειται για μια πολύ ελαφρύτερη έκδοση της Java Virtual Machine(JVM) που συναντάται και στην J2SE.

  Παραδείγματα τέτοιων συσκευών αποτελούν τα κινητά τηλέφωνα και τα PDA αρχικού επιπέδου.

- Connected Device Configuration (CDC). Στοχεύει σε περισσότερο ικανές συσκευές. Έτσι συνεργάζεται με μια σειρά από αρκετά πολύπλοκα profiles, όπως είναι τα Foundation Profile, Personal Basis Profile και Personal Profile, μέσω των οποίων παρέχονται επιπλέον λειτουργίες σε συσκευές που έχουν:

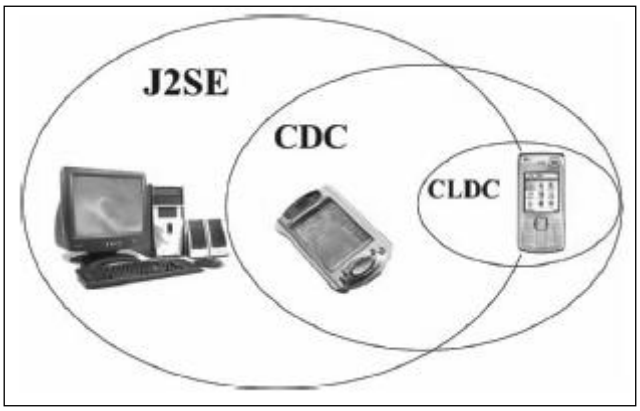
- Σύνθετο User Interface
- Μνήμη ανάμεσα σε 2-16MB αφιερωμένη σε εκτέλεση εφαρμογών Java
- Σύνδεση σε κάποιο δίκτυο
- Επεξεργαστές 32 bit

Στο configuration CDC χρησιμοποιείται η εικονική μηχανή JVM που είναι ίδια με αυτή που χρησιμοποιείται στην J2SE.

Σε αυτές τις συσκευές συγκαταλέγονται οι Διαδικτυακές τηλεοράσεις, οι δορυφορικοί δέκτες και τα περισσότερο προηγμένα PDA.

Ανάμεσα στην J2SE και τα δύο configurations της J2ME υπάρχει μια πολύ στενή σχέση. Πιο συγκεκριμένα, τα CDC και CLDC κληρονομούν ένα μεγάλο μέρος από τις βασικές λειτουργίες από μιαν υλοποίηση της J2SE.

Για την ακρίβεια, η διαρρύθμιση CDC βρίσκεται πιο κοντά στην J2SE, με την οποία άλλωστε χρησιμοποιούν κοινή εικονική μηχανή, την JVM. Αντίθετα, η διαρρύθμιση CLDC αποτελεί ένα ελαφρύτερο υποσύνολο της CDC, όπως συμβαίνει άλλωστε και με την εικονική μηχανή KVM, η οποία είναι μια ελαφρύτερη εκδοχή της JVM.



Εικόνα 73: Η σχέση κληρονομικότητας ανάμεσα σε J2SE και J2ME

Ο συνδυασμός CLDC – MIDP είναι αυτός που εντοπίζεται συχνότερα σε συσκευές που κυκλοφορούν στην αγορά, αφού χρησιμοποιείται σχεδόν σε όλα τα σύγχρονα κινητά τηλέφωνα. Μια εφαρμογή που έχει δημιουργηθεί για το profile MIDP αναφέρεται ως MIDlet.

Κάθε εφαρμογή MIDlet που αναπτύσσεται οργανώνεται σε μορφή πακέτου ακολουθώντας έναν κοινό τρόπο, ο οποίος ονομάζεται σουίτα MIDlet(MIDlet suite). Μια σουίτα MIDlet αποτελείται από:

- Ένα αρχείο Java Archive(JAR).

  Αυτό το αρχείο περιλαμβάνει σε συμπιεσμένη μορφή όλα τα στοιχεία που είναι απαραίτητα για την εκτέλεση της εφαρμογής.

Έτσι, μέσα στο αρχείο JAR βρίσκονται οι κλάσεις που έχουν παραχθεί για την εφαρμογή, μαζί με κάποιο προαιρετικό αρχείο εικόνας που ενδεχομένως να χρησιμοποιείται από το MIDlet.

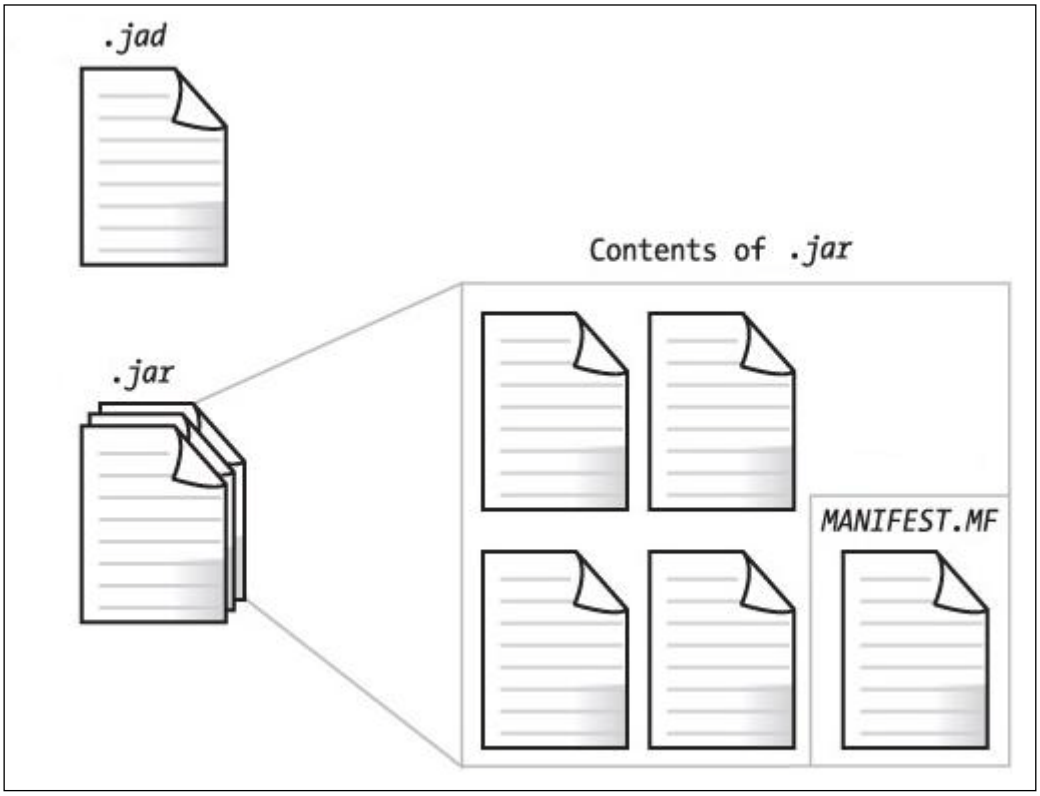
Ακόμη, συμπεριλαμβάνεται ένα αρκετά σημαντικό αρχείο για το Runtime Environment, πρόκειται για το αρχείο MANIFEST.MF, που περιγράφει τα περιεχόμενα του αρχείου JAR, ενώ αναφέρει και άλλες πολύτιμες πληροφορίες, όπως είναι το όνομα της κλάσης του MIDlet, το όνομα του πάροχου του MIDlet, η έκδοση του MIDlet και οι εκδόσεις του CLDC και του MIDP που χρειάζονται από αυτό.

- Ένα αρχείο Java Application Descriptor(JAD).

  Το αρχείο JAD αποτελεί μια περιγραφή του MIDlet και περιέχει περίπου τις ίδιες πληροφορίες με αυτές που συναντάμε στο αρχείο MANIFEST.MF.

  Το γεγονός ότι το συγκεκριμένο αρχείο βρίσκεται εκτός του JAR δίνει τη δυνατότητα λήψης πληροφοριών σχετικά με ένα MIDlet χωρίς να είναι απαραίτητη η εγκατάσταση του MIDlet σε κάποια συσκευή. Την ίδια στιγμή όμως δίνει τη δυνατότητα να περάσουν στο εκτελούμενο MIDlet ορισμένες παράμετροι από το JAD χωρίς να χρειάζεται κάποια αλλαγή στις κλάσεις που βρίσκονται στο JAR.

Μια σουίτα MIDlet έχει τη μορφή που απεικονίζεται στην εικόνα που ακολουθεί.



Εικόνα 74: Η οργάνωση μιας σουίτας MIDlet

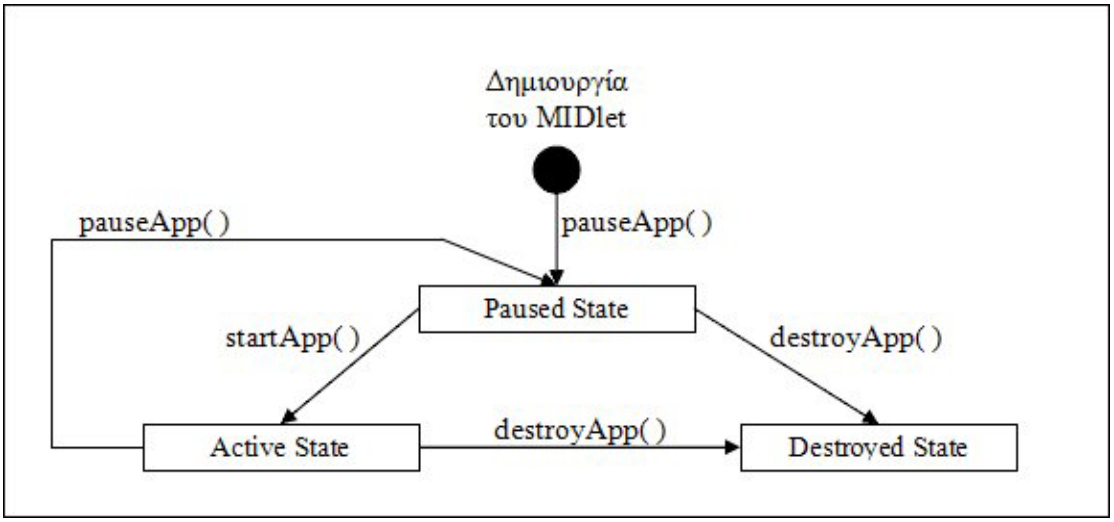
Με το συγκεκριμένο τρόπο οργάνωσης γίνεται ευκολότερη η δουλειά του Java Application Manager (JAM).

Η εφαρμογή JAM έχει το ρόλο του διαχειριστή του Runtime Environment, καθώς αναλαμβάνει μεταξύ άλλων, την εγκατάσταση των MIDlets σε μια συσκευή, ενώ ρυθμίζει και τον κύκλο εκτέλεσης ενός MIDlet.

Για την εγκατάσταση ενός MIDlet αρκεί στην εφαρμογή JAM να εξετάσει το εξωτερικό αρχείο JAD, και από αυτό να κρίνει αν το MIDlet που περιγράφεται από το JAD μπορεί να εγκατασταθεί σε μια συγκεκριμένη συσκευή.

Μετά από την επιτυχημένη εγκατάσταση έρχεται η στιγμή που ένα MIDlet θα αρχίσει να εκτελείται. Το JAM είναι αυτό που κρίνει την κατάσταση ενός MIDlet και αποφασίζει πότε θα είναι σε ενεργή κατάσταση(active state), πότε θα πραγματοποιήσει μια παύση(paused state) και θα περιμένει να ενεργοποιηθεί ξανά ή πότε θα καταστραφεί(destroyed state) και θα απομακρυνθεί από τη μνήμη της συσκευής.

Η μετάβαση ανάμεσα στις τρεις καταστάσεις ενός MIDlet γίνεται με το JAM να καλεί μια από τις μεθόδους startApp( ), pauseApp( ) και destroyApp( ), οι οποίες υλοποιούνται στο MIDlet και οδηγούν στην κατάσταση που τους αντιστοιχεί. Στην εικόνα που ακολουθεί παρουσιάζεται ο κύκλος εκτέλεσης ενός MIDlet που μόλις έχει κατασκευαστεί από τη μέθοδο constructor( ).

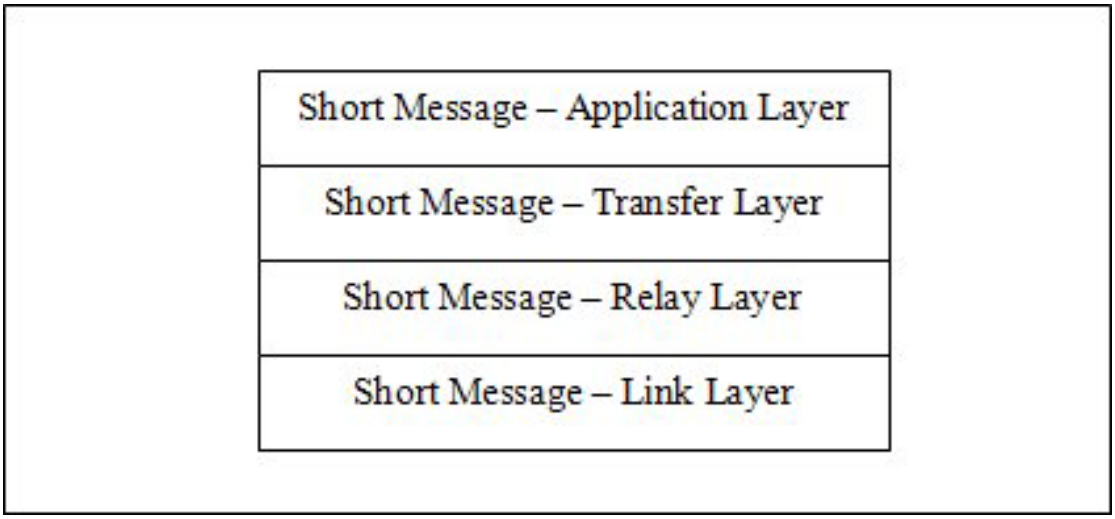


Εικόνα 75: Ο κύκλος εκτέλεσης ενός MIDlet

5.2 Ανάλυση της αναπτυχθείσας εφαρμογής

Η εφαρμογή που έχει αναπτυχθεί για την ασφαλή μετάδοση μηνυμάτων λειτουργεί στο στρώμα εφαρμογής για σύντομα μηνύματα(Short Message-Application Layer – SM-AL), το οποίο συναντάται στη στοίβα του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων.

Η στοίβα του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων αποτελείται από τέσσερα στρώματα και έχει τη μορφή που παρουσιάζεται στην ακόλουθη εικόνα.



Εικόνα 76: Η στοίβα του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων

Στο στρώμα Short Message – Application Layer ένα σύντομο μήνυμα παρουσιάζεται με μορφή κειμένου(text-mode), η οποία είναι κατανοητή από κάθε άνθρωπο.

Μέσα από την εφαρμογή που έχει αναπτυχθεί παρέχεται η κατάλληλη διεπαφή με την οποία δημιουργείται ένα αρχικό μήνυμα, το οποίο στη συνέχεια με τη βοήθεια ενός αλγόριθμου κρυπτογραφείται και λαμβάνει μια δυσνόητη μορφή που δεν επιτρέπει τον προσδιορισμό του αρχικού κειμένου.

Το κρυπτογραφημένο μήνυμα μεταφέρεται στα κατώτερα στρώματα του πρωτοκόλλου μεταφοράς σύντομων μηνυμάτων. Όπου εκεί μετατρέπεται σε μονάδα δεδομένων του πρωτοκόλλου μεταφοράς(Transfer Protocol Data Unit - TPDU) και προωθείται στον παραλήπτη του.

Κατά τη διάρκεια της μετάδοσης του TPDU που περιλαμβάνει το κρυπτογραφημένο μήνυμα υπάρχει η βεβαιότητα ότι το περιεχόμενο του μηνύματος είναι απόλυτα ασφαλές.

Αφού το αρχικό μήνυμα μπορεί να διαβαστεί μόνο από ομότιμες οντότητες σύντομων μηνυμάτων που έχουν και αυτές εγκατεστημένη την ίδια εφαρμογή, αλλά γνωρίζουν και το κλειδί που χρησιμοποιήθηκε για την κρυπτογράφηση του αρχικού μηνύματος. Μόνο σε αυτή την περίπτωση μπορεί να αντιστραφεί η κρυπτογράφηση και να ανακτηθεί το αρχικό μήνυμα.

Όμως, πριν να περάσουμε σε περισσότερες λεπτομέρειες πρέπει να αναφέρουμε τις απαιτήσεις που πρέπει να ικανοποιούνται από το περιβάλλον στο οποίο εκτελείται η εφαρμογή.

Μετά από αυτό το βήμα, θα είμαστε σε θέση να παρουσιάσουμε τον τρόπο με τον οποίο δύο πλευρές επικοινωνούν με ασφάλεια, τις επιλογές που βρίσκονται στη διάθεση ενός χρήστη κατά την εκτέλεση της εφαρμογής και ταυτόχρονα να παρουσιάσουμε τη λογική που ακολουθήθηκε για την υλοποίηση των συγκεκριμένων λειτουργιών.

5.2.1 Οι απαιτήσεις της εφαρμογής σε σχέση με το περιβάλλον εκτέλεσης

Η εφαρμογή που αναπτύχθηκε για την παρούσα πτυχιακή εργασία έχει τη μορφή ενός MIDlet, το οποίο έχει τη δυνατότητα εκτέλεσης από κάθε συσκευή κινητής τηλεφωνίας που έχει εγκατεστημένο το J2ME Runtime Environment που σχηματίζεται από το συνδυασμό του configuration CLDC 1.1 και του profile MIDP 2.0.

Από το CLDC παρέχονται βασικά πακέτα κλάσεων που βοηθούν στην επικοινωνία μεταξύ του MIDlet και της συσκευής κινητού τηλεφώνου, ενώ παρέχεται και η κλάση Thread με την οποία δημιουργούνται νήματα διεργασιών που βοηθούν σημαντικά κατά την εκτέλεση του MIDlet.

Από το MIDP παρέχονται τα πακέτα javax.microedition.lcdui και javax.microedition.rms, με τα οποία υλοποιούνται κλάσεις για τη δημιουργία του γραφικού περιβάλλοντος του MIDlet και τη δυνατότητα μόνιμης αποθήκευσης δεδομένων που έχουν ιδιαίτερη σημασία για τη λειτουργία της εφαρμογής.

Για τη σωστή λειτουργία του MIDlet απαιτείται η παρουσία ενός προαιρετικού πακέτου στο Runtime Environment, πρόκειται για την έκδοση 1.1 του πακέτου Wireless Messaging API (WMA), μέσω του οποίου δίνεται η δυνατότητα δημιουργίας και ανταλλαγής μηνυμάτων SMS.

Αν και το συγκεκριμένο πακέτο είναι προαιρετικό δεν παύει να είναι αρκετά σημαντικό, αφού επεκτείνει κατά πολύ τις δυνατότητες επικοινωνίας που παρέχονται σε κάθε εκτελούμενο MIDlet. Έτσι όλοι οι κατασκευαστές κινητών τηλεφώνων συνήθως ενσωματώνουν το πακέτο WMA στο J2ME Runtime Environment των συσκευών τους. Οπότε θεωρείται ότι το συγκεκριμένο πακέτο γίνεται de facto υποχρεωτικό, καθώς συναντάται σε όλες τις συσκευές κινητής τηλεφωνίας.

Ένα ακόμη πολύ σημαντικό προαιρετικό πακέτο είναι το Security and Trust Services API(SATSA), μέσω του οποίου υλοποιούνται κρυπτογραφικοί αλγόριθμοι που θα ήταν πολύτιμοι για την εφαρμογή.

Όμως το πακέτο SATSA, σε αντίθεση με την περίπτωση του πακέτου WMA, δεν συναντάται στις περισσότερες από τις συσκευές κινητής τηλεφωνίας που κυκλοφορούν στην αγορά. Οπότε θα ήταν καλύτερο η εφαρμογή να ήταν ανεξάρτητη από το πακέτο SATSA και έτσι να μπορεί να χρησιμοποιηθεί από το μεγαλύτερο δυνατό μερίδιο των κινητών τηλεφώνων που υπάρχουν.

Γι’ αυτό λοιπόν, όλοι οι αλγόριθμοι που χρησιμοποιούνται από την εφαρμογή αναπτύχθηκαν ξεχωριστά και ειδικά για την περίσταση. Οι μόνες υλοποιήσεις που ήταν έτοιμες και χρησιμοποιήθηκαν με ελάχιστες τροποποιήσεις προήλθαν από την κρυπτογραφική βιβλιοθήκη ανοιχτού κώδικα Bouncy Castle[*], πρόκειται για τις υλοποιήσεις του αλγορίθμου RSA και των κλάσεων BigInteger και SecureRandom, οι οποίες είναι πολύ σημαντικές για τη λειτουργία του RSA.

[*] Η βιβλιοθήκη είναι διαθέσιμη στην ιστοσελίδα http://www.bouncycastle.org/

5.2.2 Η δημιουργία ενός διαύλου ασφαλούς επικοινωνίας

Δύο πλευρές που χρησιμοποιούν την εφαρμογή που έχει αναπτυχθεί οφείλουν πριν από την εκτέλεση της βασικής λειτουργίας της, που δεν είναι άλλη από την ανταλλαγή κρυπτογραφημένων σύντομων μηνυμάτων, να προβούν στις κατάλληλες ενέργειες με τις οποίες θα προετοιμάσουν το έδαφος για την ασφαλή επικοινωνία.

Κάθε πλευρά έχει στα χέρια της τα ακόλουθα εργαλεία:

- Τον αλγόριθμο RSA, με κάθε πλευρά να έχει δημιουργήσει το δικό της ζεύγος κλειδιών για το συγκεκριμένο αλγόριθμο.
- Τον αλγόριθμο AES.
- Τη συνάρτηση PBKDF2, σε συνεργασία με τη συνάρτηση SHA256.
- Έναν κατάλογο επαφών, στον οποίο περιέχονται ο αριθμός τηλεφώνου και το αντίστοιχο δημόσιο κλειδί RSA άλλων χρηστών της εφαρμογής.

Ας δούμε τώρα το σενάριο που ακολουθείται όταν θέλουν να επικοινωνήσουν δύο πλευρές.

Θεωρούμε ότι η Alice και ο Bob επιθυμούν τη διεξαγωγή μιας κρυπτογραφημένης συνεδρίας σύντομων μηνυμάτων.

Οι δύο πλευρές χρησιμοποιούν τον αλγόριθμο AES για να διασφαλίσουν το περιεχόμενο των μηνυμάτων που ανταλλάσσουν. Η Alice και ο Bob έχουν τη δυνατότητα να χρησιμοποιούν διαφορετικό κλειδί(session key) για κάθε μια συνεδρία.

Όμως υπάρχουν δύο ζητήματα σχετικά με το session key. Κατά πρώτον, ο τρόπος με τον οποίο θα δημιουργείται το κλειδί, και κατά δεύτερον η μέθοδος με την οποία θα διατηρείται εμπιστευτικό.

Η λύση στο πρώτο ζήτημα έρχεται με τη χρησιμοποίηση της συνάρτησης PBKDF2 και της συνάρτησης SHA256. Ο χρήστης, μέσα από την κατάλληλη διεπαφή, θα εισάγει μια συνθηματική λέξη και μια τιμή salt στη συνάρτηση PBKDF2 και από την εκτέλεση της θα λαμβάνει το session key που θα έχει μήκος 256 δυαδικών ψηφίων.

Το δεύτερο ζήτημα αντιμετωπίζεται με τη χρησιμοποίηση του αλγορίθμου RSA. Έτσι, πριν από χρησιμοποίηση του AES και την ανταλλαγή κρυπτογραφημένων μηνυμάτων, είναι επιτακτική η πραγματοποίηση δύο ενεργειών με τις οποίες ο Bob και η Alice θα αποκτήσουν τη δυνατότητα επικοινωνίας μέσω του αλγορίθμου RSA.

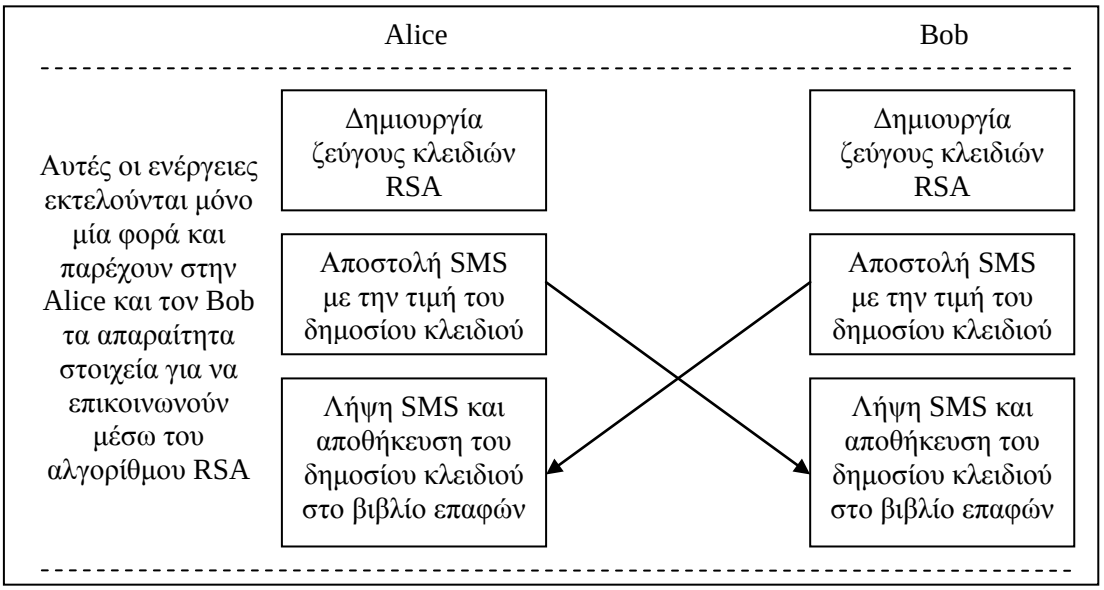
Έτσι σε πρώτη φάση, η κάθε πλευρά οφείλει να δημιουργήσει το δικό της ζεύγος κλειδιών RSA. Αυτή η διαδικασία πραγματοποιείται μόνο μία φορά και το παραγόμενο ζεύγος αποθηκεύεται και είναι έτοιμο για τις επόμενες φορές που θα χρησιμοποιηθεί ο αλγόριθμος RSA.

Αμέσως μετά η Alice και ο Bob ανταλλάσσουν τα δημόσια κλειδιά τους. Η ανταλλαγή των δημοσίων κλειδιών πραγματοποιείται μόνο μια φορά. Το κλειδί που λαμβάνεται από μια πλευρά εισάγεται στον αντίστοιχο κατάλογο επαφών, για

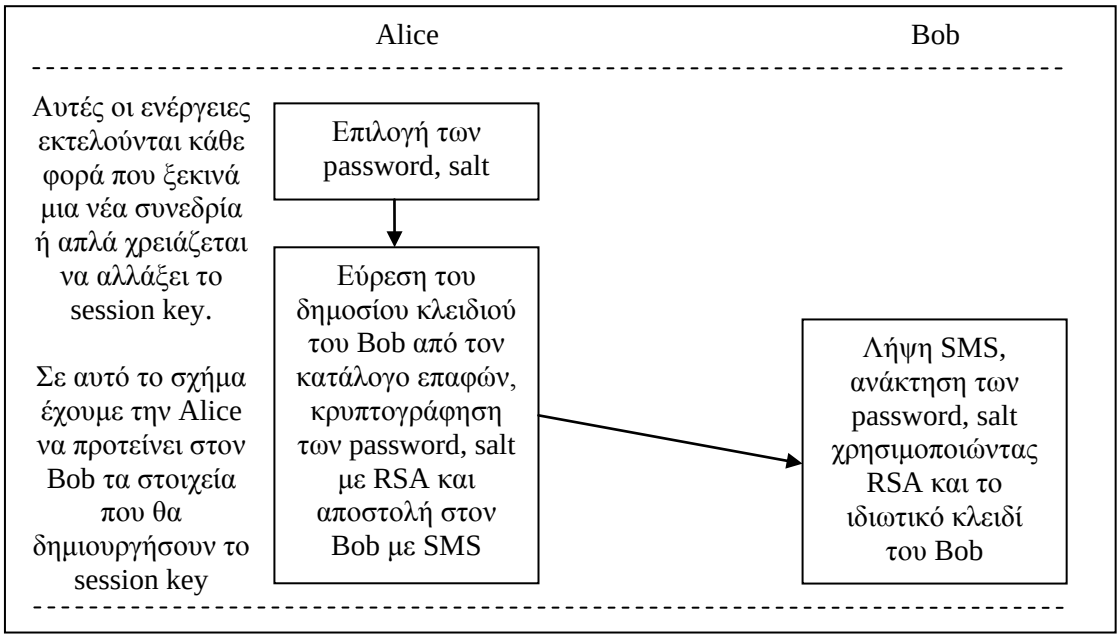
παράδειγμα, η Alice λαμβάνει το δημόσιο κλειδί του Bob και το αποθηκεύει σε μία εγγραφή στον κατάλογο επαφών της, κάνοντας έτσι εύκολη τη μελλοντική εύρεση του δημοσίου κλειδιού του Bob.

Με το πέρας των δύο παραπάνω ενεργειών οι δύο πλευρές έχουν τη δυνατότητα, όποτε εκείνες το θεωρούν απαραίτητο, να χρησιμοποιούν τον αλγόριθμο RSA και μέσω αυτού να ανταλλάσσουν τη συνθηματική λέξη και την τιμή salt που παράγουν το session key που θα χρησιμοποιείται σε κάθε συνεδρία κρυπτογραφημένων μηνυμάτων.

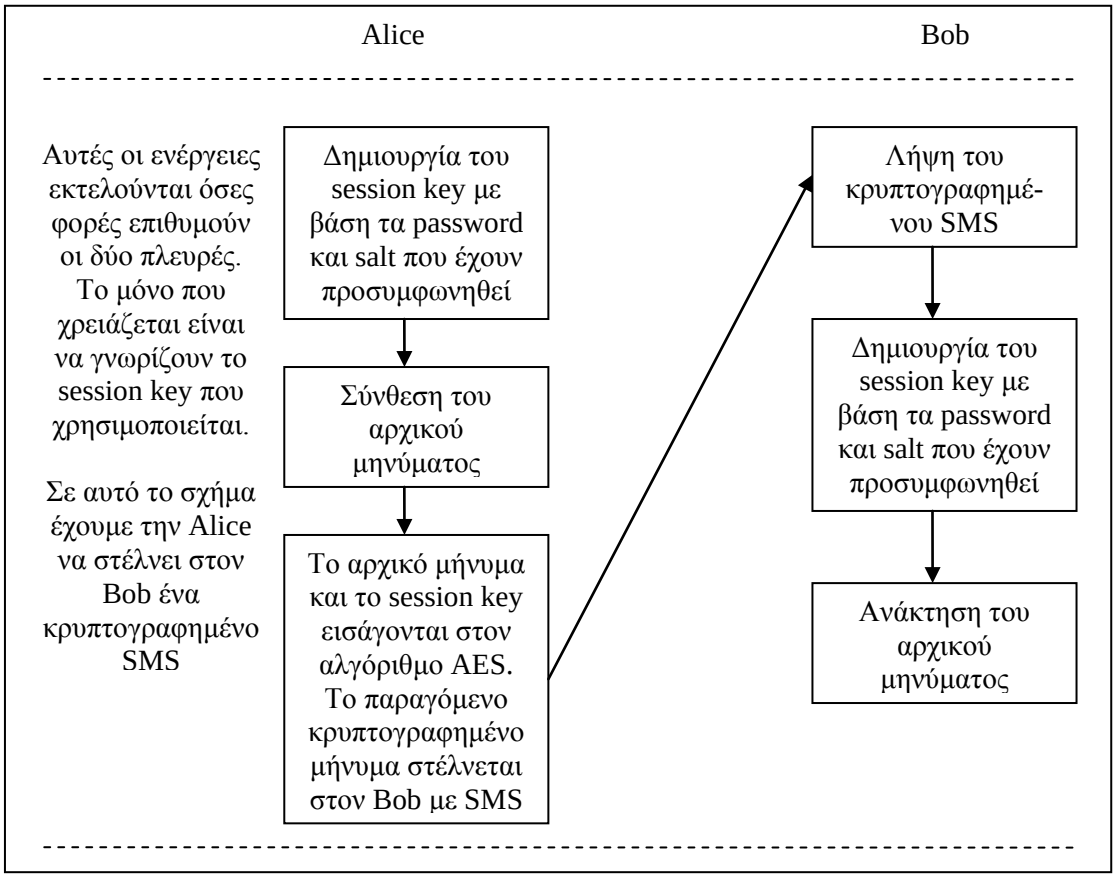
Στα τρία ακόλουθα σχήματα, και τα αντίστοιχα τρία βήματα, απεικονίζεται ολοκληρωμένο το σενάριο που ακολουθείται κατά την ασφαλή ανταλλαγή σύντομων μηνυμάτων.



Εικόνα 77: Το πρώτο βήμα, όπου εγκαθίσταται η επικοινωνία μέσω του RSA



Εικόνα 78: Το δεύτερο βήμα, όπου ανταλλάσσεται το session key που θα χρησιμοποιηθεί



Εικόνα 79: Το τρίτο βήμα, όπου πραγματοποιείται η ασφαλής ανταλλαγή σύντομων μηνυμάτων

5.2.3 Οι επιλογές που παρέχονται στο χρήστη κατά την εκτέλεση της εφαρμογής

Ένας χρήστης της εφαρμογής που έχει αναπτυχθεί για την ασφαλή μετάδοση σύντομων μηνυμάτων έχει στη διάθεση του μια σειρά από επιλογές, με τις οποίες έχει τη δυνατότητα, εκτός από την εκτέλεση της βασικής λειτουργίας, να προετοιμάσει το έδαφος για την ασφαλή επικοινωνία με κάποιον άλλο χρήστη της εφαρμογής.

Κατά την εκκίνηση της εφαρμογής, και πριν από την εισαγωγή στο κεντρικό μενού επιλογών, ελέγχεται αν βρίσκεται αποθηκευμένο κάποιο ζεύγος κλειδιών RSA που έχει δημιουργηθεί σε κάποια προηγούμενη εκτέλεση.

Στην περίπτωση που ο χρήστης δεν έχει στην κατοχή του κάποιο ζεύγος κλειδιών RSA, τότε καλείται να ακολουθήσει τη διαδικασία δημιουργίας ενός νέου ζεύγους κλειδιών.

Κατόπιν αυτού, ο χρήστης έρχεται αντιμέτωπος με το κεντρικό μενού που του δίνει τις ακόλουθες επιλογές:

- Να αποστείλει ένα SMS με το οποίο να κάνει γνωστή σε κάποιον άλλο χρήστη την τιμή του δημοσίου κλειδιού RSA που έχει στην κατοχή του.

- Να αποστείλει με μήνυμα SMS τα στοιχεία(password, salt) που πρόκειται να εισαχθούν στη συνάρτηση PBKDF2, μέσω της οποίας θα παραχθεί ένα κλειδί για χρήση σε μια επικοινωνιακή συνεδρία(session key).

- Να συνθέσει ένα σύντομο μήνυμα, να το κρυπτογραφήσει χρησιμοποιώντας τον αλγόριθμο AES και στη συνέχεια να το προωθήσει με SMS σε κάποιον άλλο χρήστη της εφαρμογής.

- Να διαχειριστεί έναν κατάλογο επαφών, όπου περιέχονται στοιχεία για όσους χρήστες χρησιμοποιούν την εφαρμογή ασφαλούς μετάδοσης SMS.

- Να εγκαταλείψει το περιβάλλον της εφαρμογής.

Ας δούμε όμως αναλυτικότερα τι συμβαίνει σε κάθε μια από τις φάσεις της εφαρμογής, ενώ παράλληλα παραθέτουμε και εικόνες από την προσομοίωση της εκτέλεσης της εφαρμογής.

Εκκίνηση της εφαρμογής και έλεγχος για την ύπαρξη κλειδιών

Η εφαρμογή ξεκινά μέσα από την επιλογή που δίνεται από το μενού όπου περιέχονται και οι υπόλοιπες εφαρμογές Java που βρίσκονται εγκατεστημένες στη συσκευή του κινητού τηλεφώνου.

Στην περίπτωση μας, όπου εκτελείται η προσομοίωση της εφαρμογής, θεωρείται ότι στο κινητό βρίσκεται μόνο η εφαρμογή που έχει αναπτυχθεί, οπότε το μενού με τις εφαρμογές έχει παρόμοια μορφή με την εικόνα που ακολουθεί. Όπως βλέπουμε και από την εν λόγω εικόνα, η εφαρμογή είναι ήδη επιλεγμένη και το μόνο που μένει είναι ο χρήστης να πιέσει το πλήκτρο που αντιστοιχεί στην εντολή της εκκίνησης(Launch).

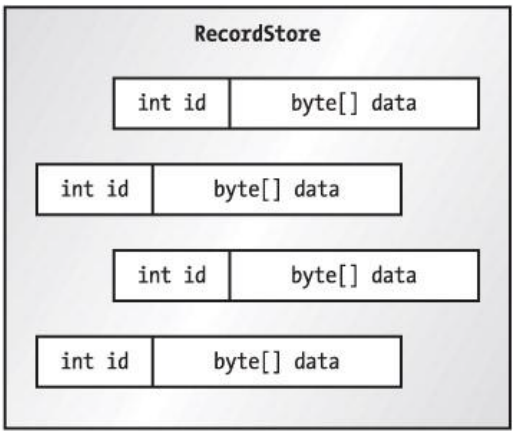
Εικόνα 80: Το μενού όπου επιλέγεται η εφαρμογή που θα αρχίσει την εκτέλεση της

Η πρώτη ενέργεια που πραγματοποιείται κατά την εκκίνηση της εφαρμογής σχετίζεται με τον έλεγχο για την ύπαρξη τυχών αποθηκευμένων κλειδιών για τον αλγόριθμο RSA.

Η αποθήκευση των κλειδιών γίνεται χάρις στο profile MIDP, το οποίο δίνει τη δυνατότητα μόνιμης διατήρησης αποθηκευμένων εγγραφών(Record Store).

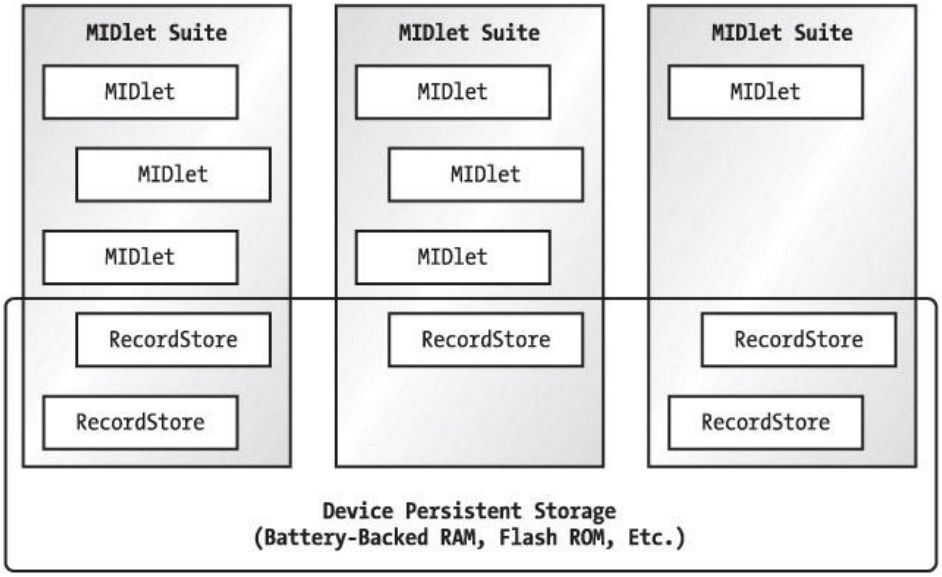
Κάθε μία από τις εγγραφές αποτελείται από δύο τμήματα. Το πρώτο τμήμα περιέχει έναν ακέραιο αριθμό ταυτότητας που αποδίδεται αυτόματα από το σύστημα διαχείρισης εγγραφών(Record Management System), ενώ το δεύτερο, που ουσιαστικά αποτελεί το κύριο μέρος μιας εγγραφής, περιέχει έναν πίνακα bytes που μεταβάλλεται ανάλογα με τα δεδομένα που εισάγονται σε κάθε εγγραφή.

Οπότε το εσωτερικό ενός Record Store μοιάζει με αυτό που απεικονίζεται στην εικόνα που ακολουθεί.



Εικόνα 81: Η μορφή που έχει το εσωτερικό ενός Record Store

Ένα Record Store φέρει ένα διακριτικό όνομα και δίνει τη δυνατότητα πρόσβασης μόνο στο MIDlet Suite από το οποίο δημιουργήθηκε. Στην περίπτωση που ένα MIDlet δημιουργεί πολλαπλά Record Stores, τότε το κάθε Record Store θα πρέπει να έχει διαφορετικό όνομα.



Εικόνα 82: Η σχέση που υπάρχει ανάμεσα σε MIDlet Suite και RecordStore

Στην περίπτωση της δικής μας εφαρμογής διατηρούνται δύο Record Stores, το ένα ονομάζεται `myKeyPair` και το άλλο ονομάζεται `contacts`.

Το Record Store που ονομάζεται `contacts` αποτελεί τον κατάλογο επαφών του κάθε χρήστη, καθώς εκεί αποθηκεύονται στοιχεία όπως είναι το όνομα, ο αριθμός τηλεφώνου και η τιμή του δημοσίου κλειδιού RSA που αντιστοιχούν σε άλλους χρήστες της εφαρμογής. Μια εγγραφή που βρίσκεται στο Record Store contacts μοιάζει κάπως έτσι.

Αριθμός_Ταυτότητας	Όνομα	Αριθμός_Τηλεφώνου	Δημόσιο_Κλειδί_RSA

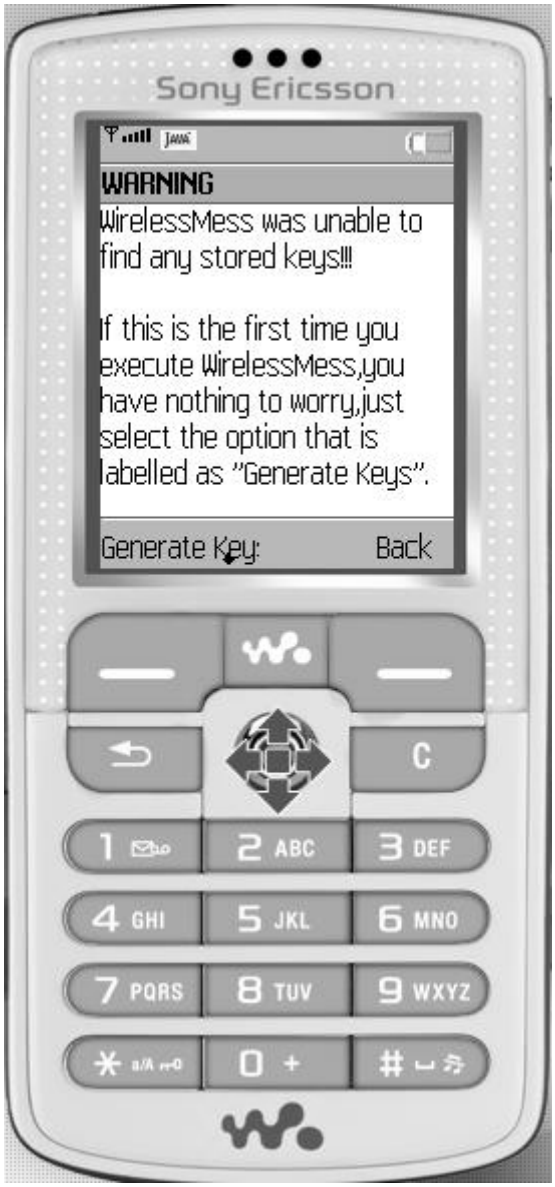
Εικόνα 83: Παράδειγμα εγγραφής στο Record Store contacts

Στο Record Store με την ονομασία `myKeyPair` φυλάσσονται όλα τα στοιχεία που προκύπτουν από τη διαδικασία παραγωγής του ζεύγους κλειδιών για τον αλγόριθμο RSA.

Κατά την εκκίνηση της εφαρμογής εξετάζεται αν υπάρχει ήδη αποθηκευμένο κάποιο Record Store με την ονομασία `myKeyPair`. Αν το εν λόγω Record Store υπάρχει, τότε σίγουρα θα περιέχει ένα ζεύγος κλειδιών που έχει δημιουργηθεί σε προηγούμενη εκτέλεση της εφαρμογής.

Σε διαφορετική περίπτωση, όπου το Record Store `myKeyPair` δεν υπάρχει, τότε εμφανίζεται η σχετική προειδοποίηση ενώ παράλληλα συστήνεται στο χρήστη να ξεκινήσει τη διαδικασία παραγωγής ενός νέου ζεύγους κλειδιών για τον αλγόριθμο RSA.

Η διαδικασία παραγωγής κλειδιών ξεκινά όταν ο χρήστης πιέσει το πλήκτρο του κινητού που αντιστοιχεί στην επιλογή "`Generate Keys`", όπως φαίνεται και στην εικόνα που ακολουθεί.



Εικόνα 84: Προειδοποίηση σχετικά με το ζεύγος των κλειδιών για τον αλγόριθμο RSA και σύσταση για τη δημιουργία ενός νέου ζεύγους

Όμως, δεν πρέπει να ξεχνάμε ότι η αναπτυχθείσα εφαρμογή προορίζεται για συσκευές κινητών τηλεφώνων, οι οποίες παρουσιάζουν ορισμένους περιορισμούς σχετικά με την επεξεργαστική τους ισχύ και τη διαθέσιμη μνήμη.

Αυτό το γεγονός εντείνει τις δυσκολίες που παρουσιάζονται κατά την αποκρυπτογράφηση με τον αλγόριθμο RSA, η οποία εκ των προτέρων είναι μια αρκετά απαιτητική διαδικασία.

Γι' αυτό και κρίνεται απαραίτητη η χρησιμοποίηση του Κινεζικού Θεωρήματος των Υπολοίπων, μέσω του οποίου η αποκρυπτογράφηση RSA διεξάγεται αποδοτικότερα, καθώς επιταχύνεται 4 φορές σε σχέση με την απλή μορφή της.

Οπότε στο Record Store `myKeyPair` αποθηκεύονται και οι συντελεστές που χρησιμοποιούνται για το Κινεζικό Θεώρημα των Υπολοίπων.

Το μόνο στοιχείο που δεν αποθηκεύεται στο Record Store, αλλά παρέχεται έτοιμο από την εφαρμογή, είναι η τιμή `65537` που χρησιμοποιείται για τον δημόσιο εκθέτη *e*.

Η διαδικασία με την οποία παράγονται τα κλειδιά για τον αλγόριθμο RSA πραγματοποιείται μια φορά, έτσι το Record Store `myKeyPair` αποτελείται από μόνο μία εγγραφή που έχει την ακόλουθη μορφή.

Αριθμός_Ταυτότητας	*p*	*q*	*dP*	*dQ*	*qInv*	*mod*	*d*

Εικόνα 85: Η εγγραφή που πραγματοποιείται στο Record Store myKeyPair

Ο χρήστης της εφαρμογής οφείλει να αποκτήσει ένα ζεύγος κλειδιών για τον αλγόριθμο RSA, αφού σε διαφορετική περίπτωση δεν μπορεί να εισέλθει σε αυτήν, έτσι είναι σίγουρο ότι αποδέχεται την πρόταση που του γίνεται, όπως εμφανίζεται και στην παραπάνω εικόνα, και αφορά τη δημιουργία ενός νέου ζεύγους κλειδιών.

Με αυτή την ενέργεια του χρήστη αρχίζει η εκτέλεση ενός thread το οποίο αναλαμβάνει να φέρει εις πέρας τη διαδικασία γέννησης του νέου ζεύγους κλειδιών.

Κατά τη διάρκεια εκτέλεσης του thread ο χρήσης δεν έχει καμία επιλογή στη διάθεση του, αφού η διαδικασία της παραγωγής κλειδιών είναι αρκετά απαιτητική.

Έτσι, το μόνο αντικείμενο που εμφανίζεται στην οθόνη του κινητού είναι μια μπάρα που απεικονίζει την πρόοδο που έχει σημειωθεί σε σχέση με τη διαδικασία της δημιουργίας των κλειδιών.

Με αυτό τον τρόπο ξεκαθαρίζεται στο χρήστη ότι η δημιουργία των κλειδιών είναι σε εξέλιξη και η εφαρμογή δεν έχει αντιμετωπίσει κάποιο σφάλμα που θα μπορούσε να διακόψει τη λειτουργία της.

Εικόνα 86: Η ένδειξη της προόδου που σημειώνεται κατά τη δημιουργία των κλειδιών RSA

Κατά τη διαδικασίας ανάπτυξης της εφαρμογής δοκιμάστηκαν τρεις πιθανές τιμές, με 512 bits, 768 bits και 1024 bits, σχετικά με το μήκος που θα έχει το δημόσιο κλειδί ενός χρήστη της εφαρμογής.

Η τιμή που επιλέχθηκε ήταν αυτή των 768 bits, καθώς αποτελούσε τη χρυσή τομή μεταξύ της παρεχόμενης ασφάλειας και της γενικότερης απόδοσης της εφαρμογής.

Όμως, οι εξελίξεις που πραγματοποιήθηκαν πάνω στην παραγοντοποίηση των μεγάλων ακεραίων, και πιο συγκεκριμένα η ανάλυση αριθμών που προσεγγίζουν τα 768 δυαδικά ψηφία και χρησιμοποιούνται από τον αλγόριθμο RSA, μας ώθησαν στην τελική επιλογή ακεραίων που θα προσεγγίζουν τα 1024 bits.

Εξαιτίας αυτού του γεγονότος η δημιουργία ενός νέου ζεύγους κλειδιών επιβραδύνεται σημαντικά, όμως μπορούμε με βεβαιότητα να ισχυριστούμε ότι η

ασφάλεια που παρέχεται από τον αλγόριθμο RSA, και γενικότερα από την εφαρμογή, είναι σε πολύ ικανοποιητικό επίπεδο.

Στον πίνακα που ακολουθεί παρουσιάζουμε ενδεικτικές τιμές για το χρόνο που απαιτείται, όπως προέκυψε από δοκιμές που πραγματοποιήθηκαν, για τη δημιουργία ενός καινούργιου ζεύγους κλειδιών για τον αλγόριθμο RSA.

Μήκος του δημοσίου κλειδιού	**Απαιτούμενος χρόνος για τη δημιουργία του ζεύγους κλειδιών**
512	Περίπου 20 δευτερόλεπτα
768	Περίπου 2 λεπτά
1024	Περίπου 3 λεπτά

Πίνακας 12: Σύγκριση ανάμεσα στο μήκος του δημοσίου κλειδιού και τον απαιτούμενο χρόνο για να δημιουργηθεί το ζεύγος κλειδιών RSA

Ενδεικτικά να αναφέρουμε ότι ένα δημόσιο κλειδί με μήκος που προσεγγίζει τα 1024 δυαδικά ψηφία αντιστοιχεί σε έναν ακέραιο αριθμό που αποτελείται από περίπου 300 ψηφία.

Για παράδειγμα, ένα από τα δημόσια κλειδιά που δημιουργήθηκαν κατά την προσομοίωση της εκτέλεσης της εφαρμογής έχει 308 ψηφία και η τιμή του είναι:
`91076521486999317298337404287835408579630467007532126110762164626056614682178019803114201003238373192894873279315042204639752354841521279849208563468051846912177296952376981205507697117235510557064123499217389024657016115766213061411020162262394759421627657863721113420099231632687668035801269685532253423 9`

Για τη μετάδοση ενός κλειδιού με αυτό το μήκος απαιτείται η αποστολή δύο μηνυμάτων SMS, αφού με ένα μήνυμα SMS μπορούν να μεταδοθούν μέχρι και 160 χαρακτήρες.

Διαχείριση των επαφών

Από το βασικό μενού επιλογών της εφαρμογής δίνεται η δυνατότητα διαχείρισης των εγγραφών που βρίσκονται αποθηκευμένες στο Record Store contacts και αντιπροσωπεύουν τον κατάλογο επαφών ενός χρήστη.

Η συγκεκριμένη ενέργεια είναι αρκετά σημαντική για τη σωστή λειτουργία της εφαρμογής, αφού σε κάθε απόπειρα του χρήστη να στείλει κάποιο μήνυμα εμφανίζεται μια λίστα με τις καταχωρημένες επαφές, από την οποία ο χρήστης επιλέγει τον προορισμό του μηνύματος.

Έτσι λοιπόν, ο χρήστης οφείλει να καταχωρήσει εκ των προτέρων ορισμένα από τα στοιχεία που αντιστοιχούν σε κάποιον άλλο χρήστη με τον οποίο επιθυμεί να επικοινωνήσει. Μόνο μετά από αυτή την ενέργεια μπορεί να ξεκινήσει η διαδικασία για την εγκατάσταση ενός διαύλου ασφαλούς επικοινωνίας.

Η είσοδος στη διαχείριση των επαφών γίνεται από την επιλογή που έχει τον τίτλο "`Manage your Contacts`", την οποία συναντάμε στο κεντρικό μενού της εφαρμογής.

Εικόνα 87: Το κεντρικό μενού της εφαρμογής, όπου επιλέγεται η διαχείριση των επαφών

Μέσα από την επιλογή της διαχείρισης των επαφών ο χρήστης έχει τη δυνατότητα να εκτελέσει τρεις ενέργειες:

- Να προσθέσει μια νέα επαφή
- Να τροποποιήσει μιαν ήδη υπάρχουσα επαφή
- Να διαγράψει μια παλιότερη επαφή

Σε αυτό το σημείο θα δούμε με ποιο τρόπο πραγματοποιείται η προσθήκη μιας νέας επαφής στον κατάλογο.

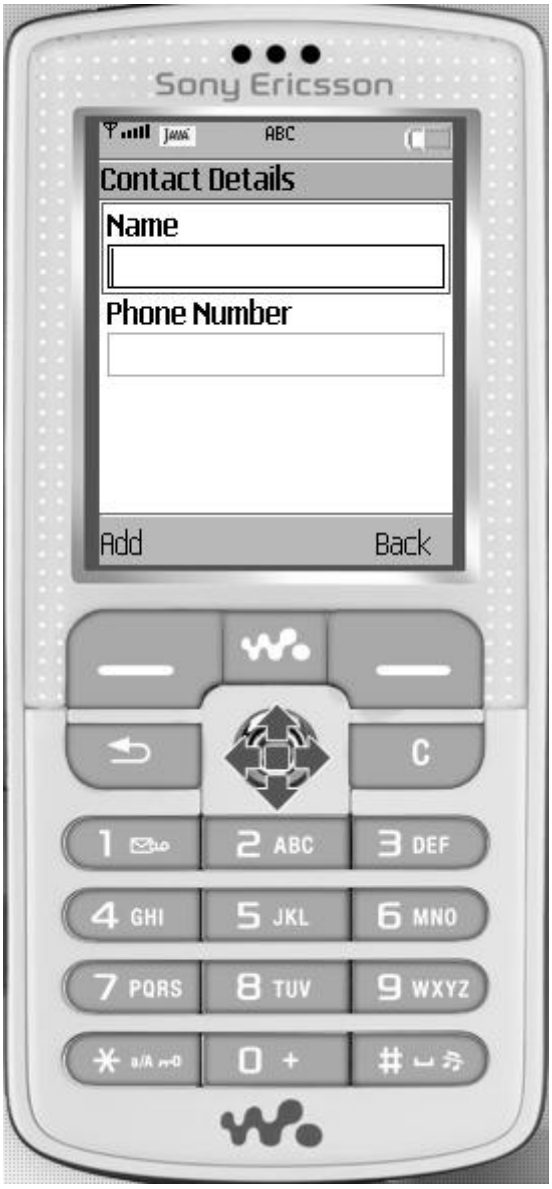
Αρχικά, ο χρήστης εισάγεται στην οθόνη όπου πραγματοποιείται η διαχείριση επαφών, όπου εκεί έρχεται αντιμέτωπος με τις τρεις επιλογές που αντιστοιχούν στις ενέργειες που έχει στη διάθεση του.

Η κεντρική οθόνη διαχείρισης των επαφών μαζί με τις επιλογές που παρέχονται μοιάζει με την οθόνη που εμφανίζεται στην ακόλουθη εικόνα.

Εικόνα 88: Το κεντρικό μενού που παρέχεται από τη διαχείριση επαφών

Ο χρήστης που επιθυμεί την προσθήκη μιας νέας επαφής μετακινεί τον επιλογέα πάνω από το label "Add" και πιέζει το πλήκτρο με το οποίο θα ξεκινήσει η εκτέλεση της συγκεκριμένης ενέργειας.

Στη συνέχεια, ο χρήστης προωθείται σε μια οθόνη, η οποία είναι παρόμοια με την εικόνα που ακολουθεί, στην οποία καλείται να εισάγει τα στοιχεία της επαφής που επιθυμεί να προσθέσει στον κατάλογο.



Εικόνα 89: Εισαγωγή ονόματος και τηλεφωνικού αριθμού κατά τη διάρκεια της προσθήκης μιας νέας επαφής

Όπως είδαμε νωρίτερα, μια εγγραφή του Record Store contacts έχει τέσσερα στοιχεία:

- Τον ακέραιο αριθμό ταυτότητας
- Το όνομα με το οποία καταχωρείται η επαφή
- Τον αριθμό τηλεφώνου
- Την τιμή του δημοσίου κλειδιού RSA

Ο χρήστης της εφαρμογής εισάγει μόνο το όνομα της καταχώρησης και τον τηλεφωνικό αριθμό που αντιστοιχεί.

Τα δύο εναπομείναντα στοιχεία εξαρτώνται από άλλους παράγοντες, όπως είναι το Record Management System που παρέχεται από το profile MIDP και ο χρήστης στον οποίο αντιστοιχεί η υπό δημιουργία επαφή.

Ο ακέραιος αριθμός ταυτότητας εισάγεται αυτόματα από το σύστημα διαχείρισης όλων των Record Stores, το οποίο παρέχεται από το profile MIDP, οπότε δεν είναι αρμοδιότητα του χρήστη της εφαρμογής.

Η τιμή του δημοσίου κλειδιού, ακόμα και αν είναι εκ των προτέρων γνωστή στο χρήστη, είναι αρκετά μεγάλη και πολύπλοκη για να εισαχθεί μέσω πληκτρολόγησης, ενώ είναι και αρκετά εύκολο να γίνει κάποιο λάθος κατά τη χειροκίνητη εισαγωγή της.

Έχοντας σαν δεδομένο αυτό το στοιχείο, θεωρήθηκε προτιμότερο να δίνεται σκόπιμα η ίδια αρχική τιμή για το δημόσιο κλειδί κάθε νέας επαφής. Για την ακρίβεια, σε κάθε νέα επαφή εισάγεται η τιμή "22".

Με αυτό τον τρόπο αποφεύγεται η πιθανότητα να καταχωρηθεί μια λανθασμένη τιμή δημοσίου κλειδιού και ταυτόχρονα είναι αρκετά εύκολος ο εντοπισμός των επαφών από τις οποίες απουσιάζει το πραγματικό δημόσιο κλειδί τους.

Η πραγματική τιμή του δημοσίου κλειδιού μιας εκ των επαφών του καταλόγου λαμβάνεται κατά τη φάση στην οποία πραγματοποιείται η ανταλλαγή των δημοσίων κλειδιών RSA. Στη συνέχεια, και για την αποφυγή πιθανών λαθών, πραγματοποιείται αυτόματη τροποποίηση στην εγγραφή της επαφής που έστειλε το κλειδί της και η αρχική ψευδής τιμή του κλειδιού αντικαθίσταται από την πραγματική.

Μετά και από αυτή την ενέργεια ουσιαστικά ολοκληρώνεται η διαδικασία με την οποία προστίθεται μια νέα επαφή, και όπως είναι προφανές, η προσθήκη μιας νέας επαφής εξαρτάται ως ένα βαθμό και από τον χρήστη στον οποίο αντιστοιχεί η υπό δημιουργία επαφή.

Υπάρχει βεβαίως και η πιθανότητα να φθάσει κάποιο μήνυμα που να περιέχει ένα δημόσιο κλειδί και η επαφή στην οποία αντιστοιχεί να μην έχει δημιουργηθεί από το χρήστη που υποδέχεται το δημόσιο κλειδί.

Σε αυτή την περίπτωση το δημόσιο κλειδί και ο αριθμός τηλεφώνου αποθηκεύονται κανονικά, ενώ το πεδίο του ονόματος λαμβάνει τον αριθμό τηλεφώνου από τον οποίο ήρθε το μήνυμα. Ονομάζοντας την επαφή με αυτό τον τρόπο δίνεται στο χρήστη η δυνατότητα να την εντοπίσει εύκολα, και αργότερα να επιλέξει την τροποποίηση της και να προβεί στην αλλαγή του ονόματος.

Αποστολή του δημοσίου κλειδιού του χρήστη

Η αποστολή του δημοσίου κλειδιού του αλγορίθμου RSA είναι μια από τις πρώτες ενέργειες που εκτελούνται όταν δύο χρήστες της εφαρμογής επιθυμούν την εγκατάσταση ενός διαύλου ασφαλούς επικοινωνίας.

Η διαδικασία για την αποστολή του δημοσίου κλειδιού μπορεί να προχωρήσει μόνο αν έχουμε εξασφαλίσει ότι ο χρήστης έχει δημιουργήσει το δικό του ζεύγος κλειδιών και έχει καταχωρήσει τις επαφές που επιθυμεί στον κατάλογο του.

Είναι αυτονόητο ότι για να στείλει ένας χρήστης το δημόσιο κλειδί του πρέπει πιο πριν να έχει ολοκληρώσει τη δημιουργία ενός ζεύγους κλειδιών RSA. Βέβαια, η εφαρμογή υποχρεώνει με τον τρόπο της το χρήστη να ολοκληρώσει τη δημιουργία κλειδιών, ιδιαίτερα όταν αυτός επιχειρεί την πρώτη του πρόσβαση στην εφαρμογή, οπότε είναι σίγουρο ότι ο χρήστης έχει στη διάθεση του ένα ζεύγος κλειδιών RSA.

Η καταχώρηση του χρήστη για τον οποίο προορίζεται το δημόσιο κλειδί είναι απολύτως απαραίτητη, αφού εκτός από την ευκολία με την οποία επιλέγεται μια επαφή από τον κατάλογο, παράλληλα προετοιμάζεται το έδαφος για την υποδοχή του δημοσίου κλειδιού από την πλευρά του άλλου χρήστη.

Η αποστολή του δημοσίου κλειδιού πραγματοποιείται μέσα από την επιλογή που παρέχεται στο κεντρικό μενού της εφαρμογής. Το κεντρικό μενού της εφαρμογής είναι παρόμοιο με αυτό που απεικονίζεται στην εικόνα που ακολουθεί.

Εικόνα 90: Το κεντρικό μενού της εφαρμογής και η επιλογή για την αποστολή του δημοσίου κλειδιού

Όλα ξεκινούν με το χρήστη να εντοπίζει στο κεντρικό μενού την επιλογή με την ετικέτα "`Public Key Exchange`" και ακολούθως να πιέζει το πλήκτρο του κινητού τηλεφώνου που αντιστοιχεί στην εντολή για την εκτέλεση της συγκεκριμένης επιλογής.

Από εκεί προχωρά στην επόμενη οθόνη, όπου εμφανίζονται οι επαφές που βρίσκονται καταχωρημένες στον κατάλογο.

Σε αυτό το σημείο ο χρήστης της εφαρμογής επιλέγει τον προορισμό που θα έχει το μήνυμα με το δημόσιο κλειδί του, όπως στη εικόνα που ακολουθεί, όπου ο χρήστης επιλέγει την επαφή που έχει το όνομα "`Bob`".

Εικόνα 91: Η λίστα με τις καταχωρημένες επαφές στις οποίες μπορεί να σταλεί το δημόσιο κλειδί

Στη συνέχεια, ο χρήστης πιέζει το πλήκτρο του κινητού που αντιστοιχεί στην εντολή "`Select`", με την οποία υποδηλώνει ότι επιλέγει αυτό τον προορισμό, και με αυτό τον τρόπο ολοκληρώνει από την πλευρά του τη διαδικασία αποστολής του δημοσίου κλειδιού.

Την ίδια στιγμή εκτελούνται δύο απαραίτητες ενέργειες, που δεν τις γνωρίζει ο χρήστης, και με τις οποίες προετοιμάζεται το μήνυμα SMS που θα αποσταλεί στην επιλεχθείσα επαφή και θα περιέχει το δημόσιο κλειδί του χρήστη.

Η μία ενέργεια αφορά την εύρεση του αριθμού τηλεφώνου του παραλήπτη και η άλλη σχετίζεται με τη φόρτωση του δημοσίου κλειδιού στο πεδίο με τα δεδομένα που βρίσκονται στο SMS.

Ο τηλεφωνικός αριθμός του παραλήπτη βρίσκεται μετά από αναζήτηση στο Record Store contacts, όπου χρησιμοποιείται σαν φίλτρο αναζήτησης του όνομα που επιλέχθηκε από τη λίστα με τις επαφές, στην προκειμένη περίπτωση έχουμε το όνομα του χρήστη "`Bob`".

Η τιμή του δημοσίου κλειδιού του χρήστη ανακτάται από το Record Store `myKeyPair`, μετατρέπεται σε string και φορτώνεται ως payload, σύμφωνα με τη μέθοδο που παρέχεται από το WMA, στο μήνυμα που δημιουργείται εκείνη τη στιγμή.

Μετά από αυτές τις ενέργειες η αποστολή ενός σύντομου μηνύματος μέσα από το WMA γίνεται ακόμη ευκολότερη και πιο άμεση.

Ο αριθμός του παραλήπτη συνδυάζεται με το χρησιμοποιούμενο πρωτόκολλο και με ένα προσυμφωνημένο port, για την εφαρμογή χρησιμοποιείται το port "`6666`". Από το συνδυασμό αυτό σχηματίζεται ένα URL, στο οποίο προωθείται το μήνυμα που έχει δημιουργηθεί.

Με την προσθήκη του port υποδεικνύεται στη συσκευή του παραλήπτη ότι το μήνυμα πρέπει να κατευθυνθεί στο MIDlet που έχει δηλώσει το ανάλογο port. Στην περίπτωση που το μήνυμα έφτανε χωρίς port, τότε είναι πιθανό να κατευθυνόταν στο φάκελο με τα εισερχόμενα μηνύματα του κινητού τηλεφώνου, ενώ υπάρχει και το ενδεχόμενο απόρριψης του μηνύματος από τη συσκευή.

Η λήψη ενός σύντομου μηνύματος μέσω του WMA, σε αντίθεση με την αποστολή, είναι περισσότερο πολύπλοκη και απαιτεί την παρουσία ενός νήματος που εκτελείται στο παρασκήνιο. Με το εκτελούμενο νήμα συνδέεται στο port ένας MessageListener και ειδοποιεί το MIDlet για τα νέα μηνύματα που φθάνουν σε αυτό, ενώ χάρις σε αυτό είναι εφικτή η ασύγχρονη αποστολή και λήψη μηνυμάτων χωρίς να παρατηρούνται διακοπές στη διεπαφή του χρήστη.

Στις εικόνες που ακολουθούν παρουσιάζεται ότι θα γινόταν στην περίπτωση που ο παραλήπτης του δημοσίου κλειδιού δεν έχει τον αποστολέα καταχωρημένο στον κατάλογο επαφών.

Αρχικά, ο παραλήπτης ειδοποιείται ότι έλαβε ένα νέο μήνυμα με την τιμή ενός δημοσίου κλειδιού.

Ο αριθμός του αποστολέα χρησιμοποιείται σαν φίλτρο για αναζήτηση στο Record Store contacts και εύρεση της επαφής στην οποία αντιστοιχεί. Στην προκειμένη περίπτωση δεν υπάρχει καταχωρημένη κάποια επαφή με το συγκεκριμένο αριθμό τηλεφώνου.

Οπότε στην οθόνη της συσκευής εμφανίζεται ο αριθμός του αποστολέα του μηνύματος, ενώ ακόμη αναφέρεται ότι το νέο μήνυμα προήλθε από κάποιον αριθμό που δεν είναι καταχωρημένος στον κατάλογο επαφών.

Την ίδια στιγμή, δημιουργείται μια νέα επαφή που περιέχει όλα τα στοιχεία που μπορούν να αξιοποιηθούν από το νεοεισελθόν μήνυμα. Με αυτό τον τρόπο αποθηκεύονται το δημόσιο κλειδί και ο τηλεφωνικός αριθμός από τον οποίο προήλθε το μήνυμα.

Εικόνα 92: Λήψη του δημοσίου κλειδιού από μια μη καταχωρημένη επαφή

Ο χρήστης που παρέλαβε το μήνυμα γνωρίζει τον αριθμό του αποστολέα, έτσι έχει τη δυνατότητα να μάθει ποιος έστειλε το μήνυμα και αργότερα να τροποποιήσει την επαφή που δημιουργήθηκε.

Η τροποποίηση μιας επαφής παρέχεται μέσα από τη διαχείριση των επαφών, οπότε ο χρήστης που επιθυμεί να προβεί στη συγκεκριμένη ενέργεια θα πρέπει αρχικά να εισέλθει στο κεντρικό μενού της εφαρμογής, να επιλέξει την επιλογή με την ετικέτα "`Manage your Contacts`" και να εισέλθει στη διαχείριση των επαφών.

Εικόνα 93: Η επιλογή για τη διαχείριση επαφών

Με την προώθηση του χρήστη στην οθόνη για διαχείριση των επαφών είναι διαθέσιμο το μενού με τις επιλογές για προσθήκη, τροποποίηση ή διαγραφή μιας επαφής.

Ο χρήστης της εφαρμογής επιθυμεί την τροποποίηση μιας επαφής οπότε, όπως φαίνεται και στη ακόλουθη εικόνα, σημαδεύει την επιλογή με το όνομα "`Edit`" και πιέζει το πλήκτρο του κινητού με το οποίο θα μεταφερθεί σε μιαν άλλη οθόνη για να συνεχίσει για τη διαδικασία τροποποίησης.

Εικόνα 94: Η επιλογή για την τροποποίηση μιας επαφής

Στη νέα οθόνη που μεταφέρεται ο χρήστης της εφαρμογής εμφανίζεται μια λίστα με όλες τις καταχωρημένες επαφές. Ο χρήστης μετακινείται ανάμεσα στις επαφές, εντοπίζει εκείνη που θέλει να τροποποιήσει, και πιέζει το πλήκτρο που αντιστοιχεί στην ενέργεια "`Edit`".

Στην εικόνα που ακολουθεί τυχαίνει η επαφή που είναι υπό τροποποίηση να είναι η μοναδική που περιλαμβάνεται στον κατάλογο του χρήστη.

Εικόνα 95: Η λίστα όπου παρουσιάζεται μία μη καταχωρημένη επαφή

Μετά από το πάτημα του "`Edit`", ο χρήστης προωθείται σε μια οθόνη που θυμίζει αρκετά εκείνη που χρησιμοποιήθηκε κατά την εισαγωγή των στοιχείων μιας υπό προσθήκη επαφής, όπου εμφανίζονταν δύο πεδία για την εισαγωγή του ονόματος και του τηλεφωνικού αριθμού της επαφής.

Όμως στην περίπτωση της τροποποίησης μιας επαφής υπάρχει μια διαφορά, όπως φαίνεται άλλωστε και στην εικόνα που ακολουθεί, τα δύο πεδία ήδη περιέχουν τις τιμές που έχουν βρεθεί από την εγγραφή που υπήρχε στο Record Store contacts και περιμένουν από τον χρήστη να πραγματοποιήσει τις αλλαγές που επιθυμεί.

Εικόνα 96: Η τροποποίηση μιας επαφής

Μόλις ο χρήστης τελειώσει με τις αλλαγές πιέζει το πλήκτρο που αντιστοιχεί στο "`Edit`", έτσι δίνει την εντολή για να ανανεωθούν όσα από τα στοιχεία της εγγραφής υπέστησαν αλλαγές. Ο χρήστης δεν έχει άμεση πρόσβαση στην τιμή του δημοσίου κλειδιού μιας επαφής, οπότε το συγκεκριμένο στοιχείο δεν επηρεάζεται.

Αποστολή των στοιχείων που θα παράγουν το κλειδί συνεδρίας

Με αυτή την ενέργεια δύο χρήστες της εφαρμογής ανταλλάσσουν ένα μήνυμα κρυπτογραφημένο με τον αλγόριθμο RSA, στο οποίο περιέχονται τα στοιχεία από τα οποία θα παραχθεί το κλειδί που θα χρησιμοποιηθεί κατά την κρυπτογραφημένη συνεδρία που θα ακολουθήσει.

Το παραγόμενο κλειδί προκύπτει από τη συνάρτηση παραγωγής κλειδιών PBKDF2, όταν αυτή χρησιμοποιείται μαζί με τη συνάρτηση κατακερματισμού SHA-256, οπότε το μήκος του κλειδιού είναι 256 δυαδικά ψηφία.

Για την εκτέλεση της συγκεκριμένης ενέργειας είναι απαραίτητο να έχει προηγηθεί ανάμεσα στους δύο χρήστες η ανταλλαγή των δημοσίων κλειδιών RSA που έχουν στην κατοχή τους.

Η διαδικασία με την οποία αποστέλλονται τα στοιχεία που παράγουν το session key ξεκινά από το κεντρικό μενού της εφαρμογής, όπου, όπως εμφανίζεται και στην εικόνα που ακολουθεί, ο χρήστης επιλέγει την επιλογή με την ετικέτα "`Session Key Exchange`".

Εικόνα 97: Το κεντρικό μενού της εφαρμογής και η επιλογή για την αποστολή των στοιχείων που δημιουργούν το session key

Μετά από αυτή την επιλογή ο χρήστης μετακινείται σε μιαν άλλη οθόνη, όπου εισάγει τα απαραίτητα στοιχεία για την εκτέλεση της συνάρτησης PBKDF2.

Έτσι, όπως φαίνεται και στην ακόλουθη εικόνα, ο χρήστης πρέπει να συμπληρώσει ένα πεδίο για τη συνθηματική λέξη Password και άλλο ένα πεδίο για την αριθμητική τιμή Salt.

Αφού ολοκληρωθεί η εισαγωγή των στοιχείων στα δύο πεδία, ο χρήστης πιέζει το πλήκτρο που αντιστοιχεί στην εντολή "Proceed", και συνεχίζει στο επόμενο βήμα της διαδικασίας που περιλαμβάνει την επιλογή του χρήστη για τον οποίο προορίζεται το μήνυμα.

Εικόνα 98: Η συμπλήρωση των πεδίων με το password και την τιμή salt

Με την επιλογή μιας από τις επαφές, όπως απεικονίζεται στην επόμενη εικόνα, ανασύρονται από το Record Store contacts η τιμή του δημοσίου κλειδιού και ο αριθμός τηλεφώνου που αντιστοιχούν στην επιλεχθείσα επαφή.

Η τιμή του δημοσίου κλειδιού εξετάζεται πριν να συνεχιστεί η διαδικασία, αφού αν η τιμή ισούται με την αρχική τιμή "22", που αποδίδεται κατά την προσθήκη μιας νέας επαφής, τότε η πλευρά για την οποία προορίζεται το μήνυμα δεν έχει αποστείλει το δημόσιο κλειδί της.

Το γεγονός αυτό καθιστά ανέφικτη την κρυπτογράφηση του μηνύματος, για αυτό εμφανίζεται η σχετική προειδοποίηση στο χρήστη που του γνωστοποιεί ότι η συγκεκριμένη επαφή του δεν έχει πραγματική τιμή δημοσίου κλειδιού.

Αν η τιμή του δημοσίου κλειδιού διαφέρει από την αρχική τιμή, τότε η διαδικασία συνεχίζεται κανονικά, με το δημόσιο κλειδί να χρησιμοποιείται για την κρυπτογράφηση του password και της τιμής salt.

Το παραγόμενο κρυπτογραφημένο μήνυμα προωθείται με SMS στον αριθμό τηλεφώνου που αντιστοιχεί στην επιλεχθείσα επαφή.

Σε αυτό το σημείο θα αναδείξουμε την πολυπλοκότητα που υπάρχει, παρουσιάζοντας ότι πραγματικά συμβαίνει σε αυτή την ενέργεια, ενώ μόνο ένα μικρό μέρος είναι φανερό στο χρήστη. Θα χρησιμοποιήσουμε ως password τη λέξη "`test`" και ως salt την τιμή "`123`".

Αυτά τα δύο στοιχεία πρέπει να κρυπτογραφηθούν με τον αλγόριθμο RSA, οπότε πρέπει να τα προετοιμάσουμε για να εισαχθούν στον αλγόριθμο.

Έτσι, μετατρέπονται σε strings και συνενώνονται χρησιμοποιώντας το χαρακτήρα "`:`" ως διαχωριστικό, οπότε έχουμε το string "`test:123`".

Στη συνέχεια, οι χαρακτήρες που σχηματίζουν το string αντικαθίστανται από τις δεκαεξαδικές τιμές που τους αντιστοιχούν από τον κώδικα ASCII. Άρα, το string παίρνει τη μορφή "`746573743a313233`". Από εκεί μετατρέπεται σε πίνακα από bytes και είναι έτοιμο για να εισαχθεί στον αλγόριθμο RSA.

Επιλέγεται ο προορισμός του μηνύματος, όπου στην προκειμένη περίπτωση είναι η "`Alice`", και βρίσκεται το αντίστοιχο δημόσιο κλειδί που είναι καταχωρημένο στον κατάλογο επαφών. Για την περίπτωση της "`Alice`", το δημόσιο κλειδί που έχει καταχωρηθεί έχει 308 ψηφία με την τιμή:

```
91076521486999317298337404287835408579630467007532126110762164626056614682178019803114201003238373192894873279315042204639752354841521279849208563468051846912177296952376981205507697117235510557064123499217389024657016115766213061411020162262394759421627657863721113420099231632687668680358012696855322534239
```

Ακολουθεί η εκτέλεση του αλγορίθμου και παράγεται ένα πίνακας τύπου byte που περιέχει το κρυπτογραφημένο αποτέλεσμα που προκύπτει. Ο πίνακας μετατρέπεται σε string και εισάγεται στο πεδίο με τα δεδομένα που μεταφέρονται μέσω SMS.

Για το παράδειγμα μας, το string που είναι αποτέλεσμα του αλγορίθμου RSA έχει την τιμή:

```
608488fbce3c69e32ccb0487eb2f31b2ff950262c38e894e0012ca901
786ce2b5f986897cde056f2da7850d5dc7c5243312a2a898e6be129a5
ac5e827017d7f84577203fd6250c5704f14431e13544b82c63b83ac5f
1d80238302b882d5b79e4e05b416d17779a20c59e76c6e608ae008b3e
dc97104f2c99de409c763a4a4926
```

Το παραπάνω string έχει μήκος 256 χαρακτήρων, οπότε χρειάζονται 2 μηνύματα SMS για τη μεταφορά του.

Εικόνα 99: Αποστολή των password και salt

Τα πράγματα είναι περισσότερο απλά στην πλευρά του χρήστη που λαμβάνει το κρυπτογραφημένο μήνυμα.

Σε πρώτη φάση, εξετάζεται ο αριθμός του τηλεφώνου από τον οποίο προήλθε το κρυπτογραφημένο μήνυμα. Με βάση αυτό τον αριθμό πραγματοποιείται μια

αναζήτηση στον κατάλογο των επαφών, όπου εντοπίζεται ποια από τις επαφές έστειλε το μήνυμα.

Έχοντας πια όλα τα στοιχεία της επαφής διαθέσιμα περνάμε στην ανάκτηση των password και salt, η οποία πραγματοποιείται με την αποκρυπτογράφηση του μηνύματος χρησιμοποιώντας τον αλγόριθμο RSA μαζί με το Κινέζικο Θεώρημα των Υπολοίπων.

Αφού ολοκληρωθεί η αποκρυπτογράφηση, ακολουθεί η παρουσίαση των αποτελεσμάτων που προκύπτουν στην οθόνη του κινητού τηλεφώνου.

Έτσι, έχουμε στην οθόνη μιαν εικόνα που μοιάζει αρκετά με την εικόνα που ακολουθεί, στην οποία αναγράφονται το όνομα και ο αριθμός τηλεφώνου του αποστολέα, μαζί με το κείμενο που αντιστοιχεί στο password και την αριθμητική τιμή που χρησιμοποιείται ως salt.

Εικόνα 100: Ανάκτηση των password και salt

Ας δούμε όμως τι διαδραματίζεται στο παρασκήνιο. Ο χρήστης που δέχεται τα στοιχεία για τη δημιουργία του session key ουσιαστικά, στην προκειμένη περίπτωση είναι η "`Alice`", λαμβάνει σε δύο μηνύματα SMS το παρακάτω κείμενο:

```
608488fbce3c69e32ccb0487eb2f31b2ff950262c38e894e0012ca901
786ce2b5f986897cde056f2da7850d5dc7c5243312a2a898e6be129a5
ac5e827017d7f84577203fd6250c5704f14431e13544b82c63b83ac5f
1d80238302b882d5b79e4e05b416d17779a20c59e76c6e608ae008b3e
dc97104f2c99de409c763a4a4926
```

Από τον τηλεφωνικό αριθμό του αποστολέα ξεκινά η διαδικασία αναζήτησης του ονόματος με το οποίο είναι καταχωρημένος στον κατάλογο επαφών.

Έτσι, στο παράδειγμα μας βρίσκεται ότι το μήνυμα που είχε αριθμό αποστολέα "`5550001`" στάλθηκε από το χρήστη "`Bob`".

Στη συνέχεια η "`Alice`" μετατρέπει το μήνυμα που έλαβε σε byte, το εισάγει στον αλγόριθμο RSA και το αποκρυπτογραφεί χρησιμοποιώντας τη βελτιστοποίηση που παρέχεται με τους συντελεστές του Κινεζικού Θεωρήματος για τα Υπόλοιπα. Το αποτέλεσμα είναι στη μορφή ενός πίνακα τύπου byte και στη συνέχεια μετατρέπεται στην αντίστοιχη δεκαεξαδική τιμή.

Για το παράδειγμα μας, προκύπτει η δεκαεξαδική τιμή "`746573743a313233`", η οποία χωρίζεται σε ζεύγη χαρακτήρων, με κάθε ζεύγος να αντιπροσωπεύει μια τιμή του κώδικα ASCII. Οι τιμές κάθε ζεύγους αντικαθίστανται από το χαρακτήρα που αντιστοιχεί, έτσι, με αυτό τον τρόπο λαμβάνουμε το string "`test:123`".

Χρησιμοποιώντας το χαρακτήρα "`:`" ως διαχωριστικό απομονώνουμε το string που βρίσκεται πριν από αυτό τον χαρακτήρα και το θεωρούμε ως password, ενώ το string που ακολουθεί το "`:`" θεωρείται ότι είναι το salt.

<u>Αποστολή κρυπτογραφημένου σύντομου μηνύματος</u>

Με τη συγκεκριμένη επιλογή πραγματοποιείται η ουσιαστικότερη λειτουργία που παρέχεται από την εφαρμογή, αφού δίνεται σε δύο χρήστες της εφαρμογής η δυνατότητα να ανταλλάξουν μια σειρά από τα δικά τους κρυπτογραφημένα μηνύματα.

Για να πραγματοποιηθεί η συγκεκριμένη διαδικασία, θα πρέπει σε κάποια προηγούμενη φάση να έχουν συμφωνήσει οι δύο πλευρές στα στοιχεία που θα χρησιμοποιηθούν για την παραγωγή του session key.

Το παραγόμενο session key έχει μήκος 256 δυαδικών ψηφίων και τροφοδοτείται στην αντίστοιχη έκδοση του αλγορίθμου AES που χρησιμοποιεί κλειδί με αυτό το μήκος.

Η διαδικασία για την κρυπτογράφηση ενός μηνύματος ξεκινά με τον χρήστη να επιλέγει την εντολή που φέρει τον τίτλο "`Send an encrypted message`", την

οποία βρίσκουμε στο κεντρικό μενού της εφαρμογής, όπως εμφανίζεται και στην ακόλουθη εικόνα.

Εικόνα 101: Επιλογή για σύνταξη ενός κρυπτογραφημένου μηνύματος

Με αυτή την επιλογή ο χρήστης προωθείται σε μιαν άλλη οθόνη, όπου σε δύο πεδία θα εισάγει το password και το salt από τα οποία θα παραχθεί το session key.

Η εν λόγω οθόνη είναι παρόμοια με αυτή που χρησιμοποιήθηκε κατά την ανταλλαγή των στοιχείων με τα οποία παράγεται το κλειδί συνεδρίας, με τη μόνη διαφορά όμως ότι εδώ τα στοιχεία εισάγονται στη συνάρτηση PBKDF2 και στην έξοδο της λαμβάνουμε το παραγόμενο κλειδί.

Η εκτέλεση της συνάρτησης PBKDF2 πραγματοποιείται αφού ο χρήστης ολοκληρώσει την εισαγωγή των στοιχείων και στη συνέχεια πιέσει το πλήκτρο που αντιστοιχεί στην εντολή "`Proceed`".

Εικόνα 102: Εισαγωγή των password και salt από τα οποία δημιουργείται το session key που θα χρησιμοποιηθεί

Με την ολοκλήρωση της παραγωγής του κλειδιού έχει δημιουργηθεί το ένα από τα δύο στοιχεία που εισάγονται στον αλγόριθμο AES. Αυτό που απομένει είναι να δημιουργηθεί και η αρχική πληροφορία την οποία θα επεξεργαστεί ο αλγόριθμος, γι' αυτό και ο χρήστης προωθείται στην επόμενη οθόνη όπου θα συνθέσει το μήνυμα που θα αντιπροσωπεύει τη αρχική πληροφορία.

Εικόνα 103: Πληκτρολόγηση του αρχικού μηνύματος

Μετά από την ολοκλήρωση της κρυπτογράφησης της αρχική πληροφορίας ο χρήστης προωθείται στη λίστα με τις επαφές του, απ' όπου επιλέγει τον χρήστη με τον οποίο επιθυμεί να επικοινωνήσει.

Εικόνα 104: Επιλογή του χρήστη για τον οποίο προορίζεται το κρυπτογραφημένο μήνυμα

Ας δούμε όμως τι εκτελείται χωρίς την απόλυτη γνώση από την πλευρά του χρήστη.

Με τη χρησιμοποίηση των τιμών "`test`" και "`123`" παράγεται το κλειδί μήκους 256 δυαδικών ψηφίων που έχει την ακόλουθη τιμή:

```
88a3c877d60b0c24c68564b034ad5a7b32882048af62d23bc78d7d2a0
17cc19e
```

Το μήνυμα "`Attack at midnight`" που έχει συνθέσει ο χρήστης μετατρέπεται σε δεκαεξαδική μορφή, με κάθε χαρακτήρα να αντικαθίσταται σύμφωνα με την τιμή ASCII που του αντιστοιχεί.

Το μήνυμα πρέπει να έχει μήκος που να είναι πολλαπλάσιο του 32. Γι' αυτό, προστίθεται στο τέλος του μηνύματος, όσες φορές είναι απαραίτητο, ο χαρακτήρας

του κενού ή ισοδύναμα η δεκαεξαδική τιμή 20. Έτσι, το μήνυμα λαμβάνει τη μορφή που ακολουθεί:

```
41747461636b206174206d69646e6967687420202020202020202020202
0202020
```

Το μήνυμα χωρίζεται σε δύο block, με το καθένα από αυτά να υποβάλλεται σε 14 γύρους μετασχηματισμών.

Το πρώτο block έχει την τιμή: “41747461636b206174206d69646e6967”.

Στην αρχή κάθε γύρου μετασχηματισμών το block έχει μια από τις ακόλουθες τιμές:

```
Round 01: c9 d7 bc 16 b5 60 2c 45 b2 a5 09 d9 50 c3 33 1c
Round 02: 65 71 90 c6 90 a2 55 14 d0 27 1c e4 f1 95 21 7e
Round 03: 22 ae 99 87 89 9b 45 cd 8d 2b 24 aa 8f 72 ab 91
Round 04: fe a9 f2 b4 cf 44 f8 55 09 6f 28 18 32 e4 4b f2
Round 05: d8 5c 6f 06 b8 56 3b 46 2f 2e 3d 46 9a 89 2c 14
Round 06: c7 f4 3d c1 62 26 eb 2b fd 0b dc b5 fb e3 e7 50
Round 07: a3 c3 2f 32 57 c5 2e c7 27 89 54 18 cc 81 f8 7f
Round 08: 1f e2 ee 9b d9 37 91 f0 6c 06 fa 4d a6 6a 8e 90
Round 09: 64 87 21 85 58 4d c1 c5 e6 5e 4f 36 49 94 bc ce
Round 10: cf 6e 96 20 49 d7 73 84 95 6e d4 bf 43 05 de 85
Round 11: 8b 71 a2 fc a1 bf dd 74 05 c7 d8 1a fa 92 fc 69
Round 12: 8f 8e 07 e8 c7 12 8d 46 f2 fb 5c 54 eb c5 46 44
Round 13: 1a 06 bc 5b 6d a5 f5 9c 6e 17 2d c3 65 f4 33 f7
Round 14: bd d7 83 ba 23 0e e7 51 b3 f7 cf 30 38 bf 77 fb
```

Το κρυπτογράφημα που παράγεται από το πρώτο block έχει την τιμή:

“141eaccd62d5ed8320728932b09ce834”

Ακολουθώντας παρόμοια λογική για το δεύτερο block, με τιμή “68742020202020202020202020202020”, έχουμε:

```
Round 01: e0 d7 e8 57 f6 2b 2c 04 e6 a5 44 90 14 8d 7a 5b
Round 02: c1 84 4d e8 a0 02 d4 97 4e fd 8e 9a ed 69 77 34
Round 03: f1 7e d6 dd 06 07 3c 98 ac 25 23 ff 06 73 b8 67
Round 04: e6 16 3d 2b d5 cf 83 61 23 f5 0d 6d 0c f8 84 2b
Round 05: 81 16 9e db f3 10 bb 9c a4 a3 40 8f a0 1a 8c 68
Round 06: 01 a2 ad 46 2e d8 fa 11 6d 27 4f e4 7d 90 d9 ef
Round 07: e3 c4 88 70 cb 51 6d 73 a5 bd a7 85 db 8b c1 56
Round 08: af f0 df 7b 66 eb ec 3c 28 8f 63 7a 38 d1 1a d6
Round 09: d8 2f fd c7 fe 90 e9 02 bb e4 c9 de 45 77 5e 82
Round 10: d4 22 36 b7 cf 72 1a 42 47 8e 46 d4 91 e6 6d 2e
Round 11: 72 bf fb aa 76 64 8a 27 d8 4b 13 09 ae 26 f1 7e
Round 12: be 4b 17 2c 41 dd f4 04 5e 9b 66 7b 98 38 bf 4d
Round 13: 22 b8 88 bd 72 ca 2a d5 b2 68 d4 74 26 28 ea 57
Round 14: ea 9a ae 6d 18 8c 63 a2 02 77 c9 58 ed ba 17 4d
```

Το κρυπτογράφημα που παράγεται από το δεύτερο block έχει την τιμή:

“e9d1fb21e948e84b3a8e81d9e22a875a”

Τα δύο κρυπτογραφημένα blocks συνενώνονται και σχηματίζουν το κρυπτογραφημένο μήνυμα με την τιμή:

“141eaccd62d5ed8320728932b09ce834e9d1fb21e948e84b3a8e81d9e22a875a”

Η συγκεκριμένη τιμή εισάγεται στο SMS που πρόκειται να μεταδοθεί στο χρήστη που επιλέχθηκε από τη λίστα των επαφών, στην προκειμένη περίπτωση είναι ο χρήστης “Bob”.

Στην πλευρά του χρήστη ο οποίος λαμβάνει το κρυπτογραφημένο μήνυμα, δηλαδή την πλευρά του “Bob”, ξεκινά η διαδικασία με την οποία θα αποκρυπτογραφηθεί το νεοεισελθόν μήνυμα και θα ανακτηθεί η αρχική πληροφορία.

Αρχικά, ο “Bob” ειδοποιείται ότι έλαβε ένα νέο κρυπτογραφημένο μήνυμα έτσι στην οθόνη του, όπως φαίνεται και στην εικόνα που ακολουθεί, αναγράφεται η σχετική ειδοποίηση μαζί με τα στοιχεία του αποστολέα του μηνύματος.

Τα στοιχεία του αποστολέα εμφανίζονται μετά από αναζήτηση που γίνεται στο Record Store contacts, όπου το τηλέφωνο του αποστολέα χρησιμοποιείται σαν φίλτρο αναζήτησης.

Στην περίπτωση μας, το κρυπτογραφημένο μήνυμα προήλθε από τον αριθμό “5550000”, όπου μετά από αναζήτηση στις εγγραφές βρέθηκε ότι ο αριθμός αντιστοιχεί στην “Alice”.

Εικόνα 105: Ειδοποίηση για τη λήψη ενός νέου κρυπτογραφημένου μηνύματος

Ο χρήστης πιέζει το πλήκτρο του κινητού του που αντιστοιχεί στο "`Ok`" και προωθείται στην επόμενη οθόνη.

Εκεί ο χρήστης συναντά τα δύο πεδία, στα οποία εισάγει τις τιμές του password και του salt που θα παράγουν το session key. Εννοείται ότι ο χρήστης πρέπει να έχει συμφωνήσει εκ των προτέρων για τις τιμές που θα χρησιμοποιεί από κοινού με τον αποστολέα κατά τη διάρκεια μιας συνεδρίας κρυπτογραφημένων μηνυμάτων.

Μετά από τη συμπλήρωση των δύο πεδίων, ο χρήστης πιέζει το πλήκτρο του κινητού τηλεφώνου που αντιστοιχεί στην εντολή "`Proceed`", ξεκινώντας με αυτό τον τρόπο την εκτέλεση της συνάρτησης PBKDF2 που θα οδηγήσει στην παραγωγή του session key.

Για το παράδειγμα μας, ο "`Bob`" εισάγει τις τιμές "`test`" και "`123`" αναπαράγοντας με αυτό τον τρόπο το κλειδί με την τιμή:

“88a3c877d60b0c24c68564b034ad5a7b32882048af62d23bc78d7d2a017cc19e”, το οποίο άλλωστε χρησιμοποιήθηκε και κατά την κρυπτογράφηση του μηνύματος.

Εικόνα 106: Εισαγωγή των παραμέτρων που θα δημιουργήσουν το session key

Το παραγόμενο session key εισάγεται μαζί με το κρυπτογραφημένο μήνυμα στον αλγόριθμο AES και αρχίζει η διαδικασία της αποκρυπτογράφησης. Με το πέρας της συγκεκριμένης διαδικασίας λαμβάνεται το μήνυμα που είχε δημιουργηθεί αρχικά από την πλευρά του αποστολέα.

Ο χρήστης της εφαρμογής προωθείται σε μια νέα οθόνη που παρουσιάζει το περιεχόμενο του αρχικού μηνύματος. Η εν λόγω οθόνη μοιάζει με αυτή που παρουσιάζεται στην εικόνα που ακολουθεί.

Εικόνα 107: Ανάκτηση του αρχικού μηνύματος

Αξίζει όμως να δούμε πως πραγματοποιείται η αποκρυπτογράφηση που εκτελείται στο παρασκήνιο.

Το κρυπτογραφημένο μήνυμα με την τιμή

“141eaccd62d5ed8320728932b09ce834e9d1fb21e948e84b3a8e81d9e22a875a”

χωρίζεται σε δύο blocks, όπου το καθένα από αυτά υποβάλλεται σε 14 γύρους μετασχηματισμών.

Για το πρώτο block, που έχει τιμή:

“141eaccd62d5ed8320728932b09ce834”

Λαμβάνονται οι ακόλουθες τιμές στην αρχή κάθε γύρου μετασχηματισμών

```
Round 01: 7a ab 8a 0f 26 68 f5 f4 6d 08 ec d1 07 0e 94 04
Round 02: a2 06 d8 68 3c f0 c3 39 9f bf 65 de 4d 6f e6 2e
Round 03: 73 c9 4a 1b c6 0f 5a 9b 89 a6 c5 5a e9 19 5d 20
Round 04: 3d 08 61 f9 32 c6 b0 b0 6b 4f 3a 92 2d a3 c1 a2
Round 05: 8a 0e 48 97 3b 9f 1d b7 2a 6b 90 5f 1a 9f 8f 08
Round 06: 43 e3 84 8b 6a 58 65 97 8e 22 fd a6 3b 17 78 05
Round 07: c0 9a 2d 60 35 6f 19 14 50 02 28 8c 24 98 81 e3
Round 08: 0a a6 20 d2 5b a7 41 23 cc 0c 15 c6 4b 2e 31 ad
Round 09: c6 f7 86 53 aa 2b 94 78 54 11 27 f1 0f bf e9 d5
Round 10: 61 b1 27 fa 6c 31 71 6f 15 a7 a8 5a b8 4a e2 5a
Round 11: bb 1b 34 89 8a a8 b3 8d 01 69 89 fc 23 d3 41 ad
Round 12: 93 14 36 81 a7 f1 62 17 5d 40 ee bd 73 e4 6e ac
Round 13: 4d 3a 9c f3 60 cc fd b4 70 2a 60 fa a1 a3 fc 69
Round 14: dd d0 01 9c d5 06 c3 47 37 2e 65 6e 53 0e 71 35
```

Το αρχικό κείμενο που λαμβάνεται για το πρώτο block έχει την τιμή:

“41747461636b206174206d69646e6967”

Με τον ίδιο τρόπο, για το δεύτερο block που είναι
“e9d1fb21e948e84b3a8e81d9e22a875a”, έχουμε

```
Round 01: 87 64 dd e3 ad f5 f0 3c 77 f4 e4 3a 55 b8 fb 6a
Round 02: 93 74 48 5b 40 45 87 7a 37 34 c4 03 f7 6c e5 92
Round 03: ae c1 33 e3 83 14 08 71 58 07 f0 f2 46 b3 bf 21
Round 04: 40 43 7d f3 38 b3 a1 ac 61 f7 0f cc e4 08 7e 01
Round 05: 48 40 5a 31 8a 19 3c a9 a0 8e 05 2c 81 93 a2 48
Round 06: 61 60 dd 13 bb 69 58 c6 ea f5 54 77 6e 15 1e 1d
Round 07: 79 e9 fb f6 33 73 a2 21 34 3e 9e eb 07 8c ce da
Round 08: 11 d1 5c b1 1f 7a 78 51 06 3d c4 8f b9 1c 3c 97
Round 09: 7c 61 84 df 31 cc 35 5a 3c 60 95 82 ff 3a 2d 69
Round 10: 0c ca 09 45 0d 0a 64 b9 49 a2 0b de e0 47 ea 73
Round 11: 8e 8a d7 f1 03 e6 5f f1 26 41 27 ef fe 47 ec 3c
Round 12: a1 c5 26 85 6f 3f 6c c1 91 8f f6 46 6f f3 eb 16
Round 13: 78 77 19 18 e0 54 f5 9b 2f f9 e3 88 55 5f 48 b8
Round 14: e1 f1 1b 39 42 06 da 5b 8e 5d 9b f2 fa 0e 71 60
```

Με την αντίστοιχη αρχική πληροφορία να έχει την τιμή:
“68742020202020202020202020202020”

Οι τιμές της αρχικής πληροφορίας που προκύπτουν από τη συνένωση των δύο blocks και το σχηματισμό του μηνύματος:

“41747461636b206174206d69646e6967687420202020202020202020
20202020”

Κάθε ένα από τα ζεύγη χαρακτήρων αντικαθίσταται από την τιμή ASCII που αντιστοιχεί, οπότε λαμβάνεται το μήνυμα “Attack at midnight” μαζί με τα κενά που είχαν προστεθεί πριν από την κρυπτογράφηση.

Αυτός ήταν ο τρόπος με τον οποίο δύο πλευρές αξιοποιούν την εφαρμογή που τους παρέχει τη δυνατότητα να επικοινωνούν ανταλλάσσοντας σύντομα μηνύματα, ενώ παράλληλα να διατηρούν εμπιστευτικό το περιεχόμενο της επικοινωνίας τους.

Έξοδος από την εφαρμογή

Η τελευταία επιλογή που παρέχεται από την εφαρμογή είναι αυτή που δίνει στο χρήστη τη δυνατότητα να εγκαταλείψει το περιβάλλον της και να στραφεί σε άλλες λειτουργίες του κινητού τηλεφώνου του.

Εικόνα 108: Το κεντρικό μενού της εφαρμογής, όπου διακρίνεται η επιλογή της εξόδου

Όπως φαίνεται και στην παραπάνω εικόνα, η έξοδος από την εφαρμογή πραγματοποιείται όταν ο χρήστης πιέσει το πλήκτρο της συσκευής που αντιστοιχεί στο "`Exit`".

Μπορεί με αυτή την ενέργεια να σταματά η εκτέλεση της εφαρμογής, όμως εκμεταλλευόμαστε την παρουσία του μηχανισμού Push Registry και μέσω αυτού πάμε την ιδέα της ασύγχρονης ανταλλαγής μηνυμάτων ένα βήμα πιο μπροστά.

Ο μηχανισμός Push Registry αποτελεί τμήμα του Java Application Manager και δίνει σε ένα MIDlet τη δυνατότητα να ενεργοποιείται αυτόματα, χωρίς την εμπλοκή του χρήστη, αλλά σύμφωνα με κάποια εξωτερικά γεγονότα, όπως είναι η ανάγκη για εκτέλεση του MIDlet σε μια συγκεκριμένη χρονική στιγμή ή η ανάγκη για επεξεργασία μιας εισερχόμενης σύνδεσης που σχετίζεται με το MIDlet.

Αυτό που ουσιαστικά κάνει το Push Registry είναι να διατηρεί ένα αρχείο με εγγραφές των εξωτερικών γεγονότων που θα οδηγήσουν τα εγκατεστημένα MIDlets μιας συσκευής σε αυτόματη ενεργοποίηση. Το JAM έχει πρόσβαση στο εν λόγω αρχείο, οπότε γνωρίζει ποια MIDlets χρειάζεται να εκκινούν αυτόματα και έτσι προσέχει αν πραγματοποιείται κάποιο από τα γεγονότα που έχει σχέση με αυτά.

Στην περίπτωση της εφαρμογής μας έχουμε το JAM να ελέγχει αν έχει ληφθεί κάποιο SMS στη θύρα “6666”. Όταν πραγματοποιείται αυτό το γεγονός, τότε το JAM αποθηκεύει προσωρινά το SMS και καλεί το MIDlet για να ξεκινήσει να εκτελείται και να διαχειριστεί το μήνυμα. Έτσι, δεν χρειάζεται να ενεργοποιήσει ο χρήστης το MIDlet και να περιμένει για τη στιγμή που θα λάβει κάποιο μήνυμα.

5.3 Αποτίμηση της παρεχόμενης ασφάλειας

Για την καλύτερη δυνατή αποτίμηση της ασφάλειας που παρέχεται από το MIDlet που επιτρέπει την ανταλλαγή κρυπτογραφημένων μηνυμάτων χρειάζεται να δούμε σε βάθος τις περιπτώσεις όπου ανταλλάσσονται δεδομένα ανάμεσα στους χρήστες της εφαρμογής.

Σε κάθε μια από αυτές τις περιπτώσεις χρειάζεται να εξετάσουμε τη ροή της μετακινούμενης πληροφορίας, ενώ παράλληλα πρέπει να δούμε κατά πόσο αυτή είναι ευάλωτη σε επιθέσεις, αλλά και σε τι βαθμό επηρεάζει τη συνολική ασφάλεια των επικοινωνιών.

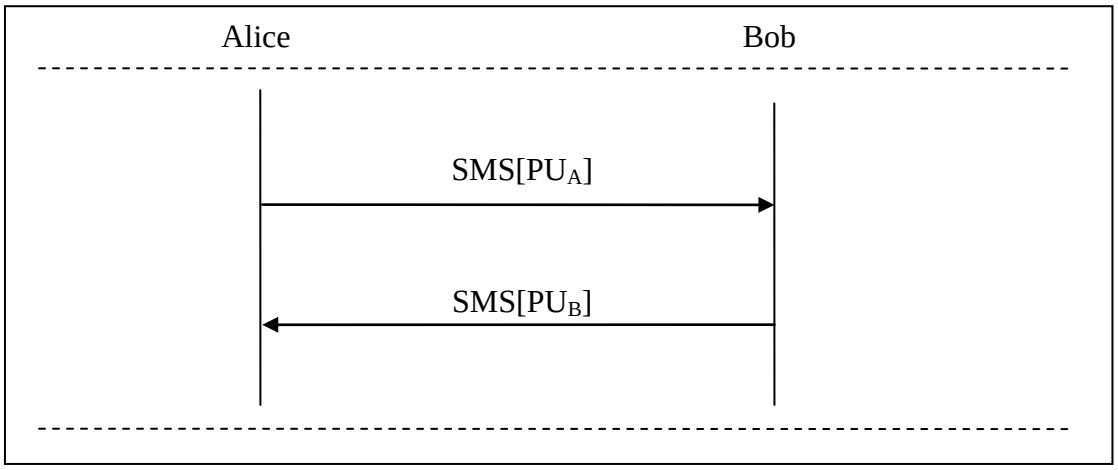
Οι χρήστες που συμμετέχουν σε μια συνεδρία ανταλλαγής κρυπτογραφημένων μηνυμάτων ενδέχεται να εμπλακούν μέχρι και σε τρεις πιθανές περιπτώσεις όπου μεταδίδονται δεδομένα. Πρόκειται για τις τρεις ακόλουθες φάσεις:

- Ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA
- Ανταλλαγή κλειδιών για την τρέχουσα συνεδρία ανταλλαγής κρυπτογραφημένων μηνυμάτων
- Ανταλλαγή μηνυμάτων που έχουν κρυπτογραφηθεί με τον αλγόριθμο AES

5.3.1 Ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA

Αρχικά, θα εξετάσουμε τι γίνεται κατά την ανταλλαγή των δημοσίων κλειδιών που χρησιμοποιούνται από τον αλγόριθμο RSA, έτσι στο σχήμα που ακολουθεί

απεικονίζεται η ροή των δεδομένων ανάμεσα σε δύο χρήστες της εφαρμογής που επιθυμούν να δημιουργήσουν ένα μονοπάτι ασφαλούς επικοινωνίας.



Εικόνα 109: Η ροή δεδομένων κατά την ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA

Στην παρούσα φάση θεωρούμε ότι και οι δύο χρήστες της εφαρμογής έχουν περάσει από το στάδιο της δημιουργίας ενός ζεύγους κλειδιών για τον αλγόριθμο RSA, οπότε στη συνέχεια ανταλλάσσουν, μέσα από μηνύματα SMS, το δημόσιο κλειδί RSA που τους αντιστοιχεί.

Έτσι, από την πλευρά του ενός χρήστη, όπου θεωρούμε ότι είναι η Alice, αποστέλλεται το δημόσιο κλειδί PU_A, ενώ από την πλευρά του άλλου χρήστη, όπου στην προκειμένη περίπτωση είναι ο Bob, αποστέλλεται το δημόσιο κλειδί PU_B.

Μετά την ανταλλαγή των δύο SMS μηνυμάτων το περιεχόμενο τους αποθηκεύεται στο record store όπου περιλαμβάνονται τα δημόσια κλειδιά των υπολοίπων επαφών που έχει ο εκάστοτε χρήστης της εφαρμογής.

Στα συγκεκριμένα μηνύματα δεν εφαρμόζεται κάποιου είδους κρυπτογράφηση, όμως το περιεχόμενο των μηνυμάτων είναι τέτοιο που δεν επηρεάζει την επικοινωνία ανάμεσα στις δύο πλευρές.

Η μόνη περίπτωση όπου θα υπήρχε κάποιο πρόβλημα θα ήταν κατά τη χρησιμοποίηση ενός ζεύγους RSA με μικρό μήκος δυαδικών ψηφίων. Τότε θα είχαμε ένα δημόσιο κλειδί το οποίο θα ήταν ευάλωτο στην παραγοντοποίηση, αποκαλύπτοντας έτσι τους δύο πρώτους αριθμούς που χρησιμοποιήθηκαν κατά τη δημιουργία ολόκληρου του ζεύγους κλειδιών.

Το προαναφερθέν γεγονός είναι που μας οδήγησε στη χρησιμοποίηση κλειδιών με μήκος περίπου ίσο με 1024 δυαδικά ψηφία, αν και κατά τις πρώτες φάσεις ανάπτυξης της εφαρμογής προτιμήθηκαν, περισσότερο για λόγους αποδοτικότητας, τα κλειδιά με μήκος 768 bits.

Αφού όμως αποδείχθηκε ότι τα 768 bits μπορούν να παραγοντοποιηθούν τότε κατέστη απολύτως απαραίτητη η χρησιμοποίηση κλειδιών των 1024 bits που θα παρέχουν την αναγκαία ασφάλεια.

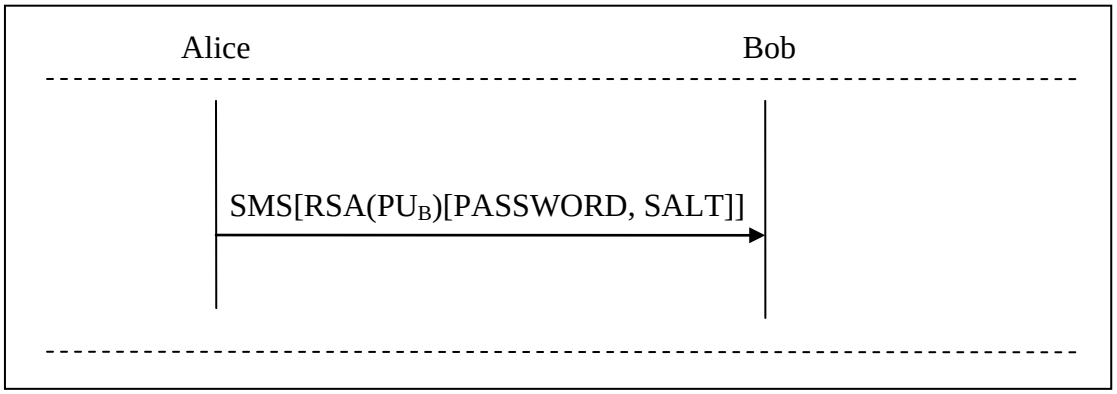
5.3.2 Ανταλλαγή κλειδιού για μια συνεδρία κρυπτογραφημένων μηνυμάτων

Στην επόμενη φάση της επικοινωνίας ανάμεσα σε δύο χρήστες του MIDlet βρίσκεται το σημείο όπου, ένας εξ αυτών, αποφασίζει ποια στοιχεία(Password, Salt) θα χρησιμοποιηθούν κατά την τρέχουσα συνεδρία ανταλλαγής κρυπτογραφημένων μηνυμάτων.

Να θυμίσουμε ότι, η συμβολοσειρά Password και η αριθμητική τιμή Salt έχουν ιδιαίτερη σημασία για την επικοινωνία ανάμεσα στους δύο χρήστες του πρωτοκόλλου, καθώς χρησιμοποιούνται από τη συνάρτηση PBKDF2, μέσω της οποίας προκύπτει το κλειδί μήκους 256 δυαδικών ψηφίων που χρησιμοποιείται στο συμμετρικό αλγόριθμο κρυπτογράφησης AES.

Εξαιτίας του ρόλου που προαναφέραμε για τα Password και Salt δεν είναι δυνατόν να τα αφήσουμε απροστάτευτα σε επιθέσεις που ενδέχεται να διενεργηθούν στο κανάλι επικοινωνιών, οπότε τα συγκεκριμένα στοιχεία κρυπτογραφούνται με τον αλγόριθμο RSA.

Όπως φαίνεται και στο σχήμα που ακολουθεί η μία πλευρά, σε αυτή την περίπτωση η Alice, αποστέλλει με κρυπτογραφημένο μήνυμα SMS στην άλλη, δηλαδή τον Bob, τις τιμές των Password και Salt.



Εικόνα 110: Η ροή δεδομένων κατά την ανταλλαγή κλειδιού μιας συνεδρίας μηνυμάτων

Κατά την κρυπτογράφηση των δύο στοιχείων χρησιμοποιείται το δημόσιο κλειδί του αποδέκτη του μηνύματος, εξασφαλίζοντας με αυτό τον τρόπο ότι το μήνυμα θα διαβαστεί μόνο από εκείνον και όχι από κάποιον τρίτο που μπορεί να παρακολουθεί το κανάλι επικοινωνίας, αφού κατά την αποκρυπτογράφηση χρησιμοποιείται το ιδιωτικό κλειδί του αποδέκτη για το οποίο θεωρούμε ότι μόνον εκείνος το έχει στην κατοχή του.

Από τη χρησιμοποίηση του αλγορίθμου RSA γίνεται εμφανές ότι η διασφάλιση των τιμών του Password και του Salt, και κατ’ επέκταση όλων των συνεδριών ανταλλαγής

κρυπτογραφημένων μηνυμάτων, βασίζεται ουσιαστικά στην ασφάλεια που παρέχεται από τον αλγόριθμο RSA και την ανθεκτικότητα που επιδεικνύει σε διάφορες επιθέσεις που έχουν επιχειρηθεί σε αυτόν κατά καιρούς.

Γι' αυτόν ακριβώς το σκοπό οι δύο πλευρές επιλέγουν τη χρησιμοποίηση κλειδιών που διαθέτουν το κατάλληλο μήκος δυαδικών ψηφίων, τέτοιο ώστε να αποτρέπεται η απόπειρα παραγοντοποίησης του τμήματος που αποτελεί το δημόσιο κλειδί του κάθε χρήστη.

Σε διαφορετική περίπτωση, αφού οι τιμές των Password και Salt μπορούν να ανακτηθούν από κάποιον τρίτο, τότε ο συμμετρικός αλγόριθμος AES που χρησιμοποιείται στην ανταλλαγή κρυπτογραφημένων μηνυμάτων καθίσταται εντελώς άχρηστος, αφού μπορεί και η τρίτη πλευρά μπορεί να εισάγει τα στοιχεία στη συνάρτηση PBKDF2 και να ανακτήσει το κλειδί που θα χρησιμοποιηθεί όχι μόνο στην τρέχουσα συνεδρία, αλλά και σε κάθε άλλη απόπειρα επικοινωνίας που θα γίνει στο μέλλον.

5.3.3 Ανταλλαγή κρυπτογραφημένων μηνυμάτων

Η τρίτη και τελική φάση της επικοινωνίας των δυο πλευρών που χρησιμοποιούν το πρωτόκολλο ασφαλούς ανταλλαγής σύντομων μηνυμάτων έχει με διαφορά τη μεγαλύτερη σημασία, αφού σε αυτό το στάδιο ικανοποιείται η κυριότερη απαίτηση που υπάρχει από το πρωτόκολλο, που δεν είναι άλλη από την ανταλλαγή μηνυμάτων με περιεχόμενο που έχει κρυπτογραφηθεί με τη βοήθεια του συμμετρικού αλγορίθμου AES.

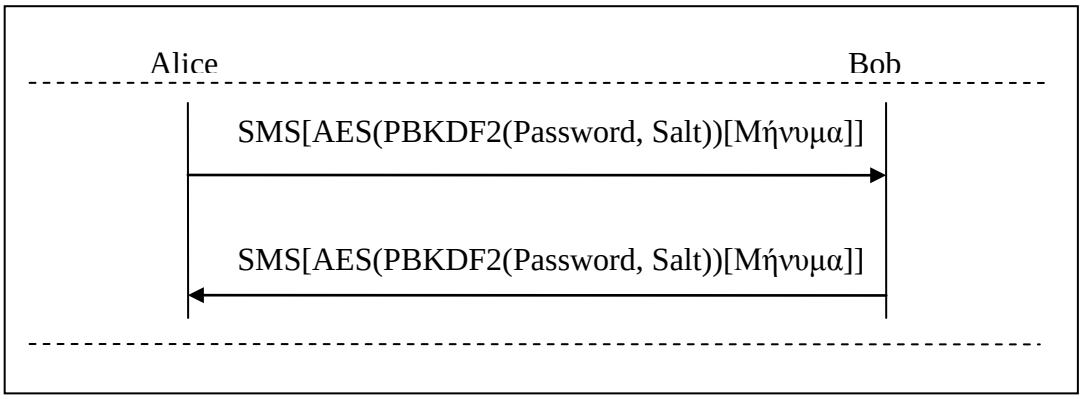
Για την εκτέλεση της κρυπτογράφησης θεωρούμε ότι οι δύο πλευρές έχουν προετοιμάσει το έδαφος, πραγματοποιώντας ενέργειες που περιγράφηκαν και νωρίτερα, όπως:

- Η δημιουργία ζεύγους κλειδιών για τον αλγόριθμο δημοσίου κλειδιού RSA
- Η ανταλλαγή δημοσίων κλειδιών για τον αλγόριθμο RSA και η αποθήκευση τους
- Η επιλογή των Password και Salt που θα χρησιμοποιηθούν σε μια συνεδρία μηνυμάτων
- Η γνωστοποίηση των Password και Salt στην άλλη πλευρά που συμμετέχει στο πρωτόκολλο

Έτσι, φτάνουμε στο σημείο όπου η μία από τις δύο πλευρές επιθυμεί να αποστείλει ένα κρυπτογραφημένο μήνυμα. Σε αυτή την περίπτωση η συγκεκριμένη πλευρά θα πρέπει να πραγματοποιήσει τις ακόλουθες ενέργειες:

- Να δημιουργήσει το κλειδί που θα χρησιμοποιηθεί στον αλγόριθμο AES κατά την τρέχουσα συνεδρία. Το συγκεκριμένο κλειδί προκύπτει όταν τα προσυμφωνηθέντα Password και Salt εισαχθούν από το χρήστη στη συνάρτηση PBKDF2.
- Να συνθέσει το σύντομο μήνυμα που επιθυμεί να αποστείλει στην άλλη πλευρά.
- Να κρυπτογραφήσει το σύντομο μήνυμα με το συμμετρικό αλγόριθμο κρυπτογράφησης AES.

Στην εικόνα που ακολουθεί παρουσιάζεται η περίπτωση όπου δύο πλευρές επικοινωνούν χρησιμοποιώντας το MIDlet που επιτρέπει την κρυπτογράφηση του περιεχομένου μηνυμάτων SMS.



Εικόνα 111: Η ανταλλαγή κρυπτογραφημένων μηνυμάτων ανάμεσα στις δύο επικοινωνούσες πλευρές

Κάθε ένα από τα μηνύματα κρυπτογραφείται με τον αλγόριθμο AES και ως κλειδί κρυπτογράφησης χρησιμοποιείται η σύνοψη που παράγεται από τη συνάρτηση PBKDF2, όταν βεβαίως σε αυτήν εισαχθούν το Password και το Salt που έχουν επιλέξει οι δύο πλευρές για την τρέχουσα συνεδρία μηνυμάτων.

Το περιεχόμενο των μηνυμάτων που ανταλλάσσονται ανάμεσα στις δύο πλευρές παραμένει εμπιστευτικό, καθώς ο αλγόριθμος κρυπτογράφησης AES διατηρεί ένα υψηλό επίπεδο ασφάλειας και παρά τις προσπάθειες κρυπτανάλυσης που έχουν επιχειρηθεί δεν υπάρχει κάποια μέθοδος που να επιτρέπει την αποκάλυψη των κρυπτογραφημένων μηνυμάτων.

Ακόμη, το κλειδί που χρησιμοποιείται σε κάθε συνεδρία ανταλλαγής μηνυμάτων έχει γίνει γνωστό, όπως είδαμε και σε προηγούμενο βήμα, μόνο στις δύο πλευρές που συμμετέχουν στο πρωτόκολλο επικοινωνίας, ανεβάζοντας έτσι ακόμα περισσότερο το επίπεδο της παρεχόμενης ασφάλειας.

5.3.4 Τελικό συμπέρασμα

Από τις προαναφερθείσες περιπτώσεις ανταλλαγής δεδομένων ανάμεσα στις δύο πλευρές που συμμετέχουν στο πρωτόκολλο ασφαλούς ανταλλαγής μηνυμάτων γίνεται εμφανές ότι η γενικότερη ασφάλεια του πρωτοκόλλου εξασφαλίζεται μέσα από τη συμπληρωματική λειτουργία των αλγορίθμων RSA και AES.

Βέβαια, σημαντικότερη θα πρέπει να θεωρείται η συμβολή του αλγορίθμου δημοσίου κλειδιού RSA, καθώς χάρις σε αυτόν επιλύεται το ζήτημα της ασφαλούς ανταλλαγής του κλειδιού για τον συμμετρικό αλγόριθμο AES, επιτρέποντας έτσι την ασφαλέστερη λειτουργία τόσο του συμμετρικού αλγορίθμου όσο και του γενικότερου πρωτοκόλλου.

5.4 Αποτελέσματα Εργασίας - Μελλοντική Έρευνα

Με αυτό το κεφάλαιο, όπου παρουσιάστηκε η εφαρμογή που επιτρέπει την ασφαλή ανταλλαγή μηνυμάτων SMS, ολοκληρώθηκε το πρώτο μέρος της πτυχιακής εργασίας.

Η εν λόγω εφαρμογή αναπτύχθηκε για το περιβάλλον Java Micro Edition ικανοποιώντας τις απαιτήσεις που υπήρχαν κατά τη φάση του σχεδιασμού του έργου. Έτσι το τελικό αποτέλεσμα, δηλαδή το MIDlet που εγκαθίσταται σε μια συσκευή κινητού τηλεφώνου, λειτουργεί άκρως ικανοποιητικά και διασφαλίζει το περιεχόμενο των μηνυμάτων που ανταλλάσσονται ανάμεσα σε δύο χρήστες της εφαρμογής.

Ακόμη, η εφαρμογή εκμεταλλεύεται το κυριότερο πλεονέκτημα της υλοποίησης σε Java Micro Edition, που είναι το γεγονός ότι η συγκεκριμένη πλατφόρμα είναι εγκατεστημένη σε ένα πολύ μεγάλο ποσοστό των συσκευών κινητής τηλεφωνίας που κυκλοφορούν στην αγορά, επιτρέποντας με αυτό τον τρόπο στην εφαρμογή να χρησιμοποιηθεί από ένα μεγάλο αριθμό χρηστών χωρίς να απαιτείται η κατοχή ενός πολύ ισχυρού και ταυτοχρόνως ακριβού κινητού τηλεφώνου.

Όμως από την άλλη πλευρά, το γεγονός ότι η εφαρμογή δομήθηκε σύμφωνα με όσα ορίζονται από το περιβάλλον Java Micro Edition, μας οδηγεί σε κάποια σημεία, όπως είναι για παράδειγμα οι κλάσσεις των κρυπτογραφικών αλγορίθμων, ή το γραφικό περιβάλλον και το σύστημα διαχείρισης των εγγραφών, που ενδεχομένως στο μέλλον να αποτελέσουν μιαν ευκαιρία για επανεξέταση και ίσως και την εκ νέου υλοποίηση τους με έναν άλλο τρόπο.

Γι’ αυτό λοιπόν κάποιες εναλλακτικές λύσεις, πέραν του περιβάλλοντος Java Micro Edition, θα ήταν ιδανικές για την περαιτέρω διερεύνηση του θέματος της ασφαλούς ανταλλαγής σύντομων μηνυμάτων. Οι ακόλουθες είναι μερικές από τις πλατφόρμες στις οποίες θα μπορούσε να εξαχθεί η εφαρμογή:

- Android Operating System, υπάρχει σε πληθώρα ισχυρών Smart-Phones και αποτελεί μια Open-Source υλοποίηση πλατφόρμας όπου οι εφαρμογές που προορίζονται γι’ αυτήν γράφονται κυρίως σε Java, περιλαμβάνει API με κλάσσεις που σχετίζονται με την ασφάλεια, ενώ διαθέτει και RDBMS SQL Lite.
- iOS, βρίσκεται εγκατεστημένο αποκλειστικά στις συσκευές iPhone της Apple, καθώς αποτελεί δικό της προϊόν, και οι εφαρμογές για το συγκεκριμένο λειτουργικό γράφονται στη γλώσσα προγραμματισμού Objective-C.
- BlackBerry Operating System, παράγεται από την εταιρεία Research In Motion και βρίσκεται στη δικιά της σειρά από Smart-Phones που ονομάζονται BlackBerry. Το συγκεκριμένο λειτουργικό σύστημα βασίζεται στη γλώσσα προγραμματισμού Java κι έτσι οι εφαρμογές που εκτελούνται σε αυτό γράφονται στην ίδια γλώσσα.

Μέρος Δεύτερο

Στο κεφάλαιο που ακολουθεί περιγράφεται αναλυτικά ο τρόπος με τον οποίο δομείται ένα πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας που βασίζεται στη χρησιμοποίηση μιας σκυτάλης ασφάλειας.

Βεβαίως, πριν από την ανάλυση του πρωτοκόλλου, οφείλουμε να καλύψουμε το θεωρητικό υπόβαθρο στο οποίο βασιστήκαμε κατά τη διαδικασία ανάπτυξης του μηχανισμού απομακρυσμένης πιστοποίησης ταυτότητας.

Έτσι, παρουσιάζονται οι διάφορες μορφές που μπορεί να έχει μια σκυτάλη ασφάλειας αλλά και η γενικότερη φιλοσοφία που χαρακτηρίζει την αυθεντικοποίηση που βασίζεται σε πολλαπλούς παράγοντες.

Ακόμη, παρουσιάζονται τα δομικά στοιχεία που επέτρεψαν το σχηματισμό του εν λόγω πρωτοκόλλου, μεταξύ των οποίων είναι:

- Η πλατφόρμα Java Enterprise Edition
- Το πρωτόκολλο SSL-TLS
- Το Java Communications API
- Το RDBMS MySQL

Κεφάλαιο 6 - Απομακρυσμένη πιστοποίηση ταυτότητας

Σε αυτό το κεφάλαιο θα παρουσιάσουμε μια τροποποιημένη εκδοχή της εφαρμογής ασφαλούς ανταλλαγής σύντομων μηνυμάτων, η οποία θα δίνει σε ένα κινητό τηλέφωνο τη δυνατότητα να λειτουργεί σαν σκυτάλη ασφάλειας(Security Token) ή σκυτάλη αυθεντικοποίησης(Authentication Token), αποτελώντας με αυτό τον τρόπο έναν από τους παράγοντες που συμμετέχουν σε ένα πρωτόκολλο απομακρυσμένης πιστοποίησης της ταυτότητας ενός χρήστη.

Όμως, το νέο MIDlet που προκύπτει, δεν είναι από μόνο του αρκετό για να αναδείξει σφαιρικά τον τρόπο με τον οποίο θα λειτουργούσε ένα πρωτόκολλο αυθεντικοποίησης που βασίζεται σε δύο διαφορετικούς παράγοντες.

Έτσι κρίθηκε αναγκαίο να αναπτυχθεί μια νέα εφαρμογή η οποία, σύμφωνα με τις προδιαγραφές που ορίζονται από την πλατφόρμα Java Enterprise Edition, θα εκτελείται στο περιβάλλον ενός εξυπηρετητή, αποτελώντας με αυτό τον τρόπο το ιδανικό συμπλήρωμα για το νέο MIDlet.

6.1 Αυθεντικοποίηση δύο παραγόντων

Σαν παράγοντας αυθεντικοποίησης λογίζεται η πληροφορία που χρησιμοποιείται ως αποδεικτικό στοιχείο από μια οντότητα, είτε αυτή είναι άνθρωπος είτε είναι υπολογιστής, όταν κρίνεται απαραίτητη η πιστοποίηση ταυτότητας της εν λόγω οντότητας.

Υπάρχουν διάφορα στοιχεία με τα οποία μια οντότητα μπορεί να πιστοποιήσει την εγκυρότητα της ταυτότητας της. Ανάμεσα στους πιθανούς παράγοντες αυθεντικοποίησης συγκαταλέγονται οι:

- Ανθρώπινοι παράγοντες που είναι άρρηκτα συνδεδεμένοι με ένα άτομο, όπως είναι τα βιομετρικά στοιχεία. Με αυτό τον τρόπο ένα άτομο αποδεικνύει κάτι που είναι δικό του όταν, για παράδειγμα, υποβάλλει ένα δακτυλικό αποτύπωμα.
- Προσωπικοί παράγοντες, οι οποίοι είναι με κάποιο τρόπο γνωστοί αποκλειστικά σε ένα άτομο. Με αυτό τον τρόπο ένα άτομο αποδεικνύει κάτι που γνωρίζει όταν, για παράδειγμα, χρησιμοποιεί ένα μυστικό κωδικό.
- Τεχνικοί παράγοντες, οι οποίοι παρέχονται με κάποια φυσική μορφή και σχετίζονται με το άτομο. Έτσι, ένα άτομο αποδεικνύει την ταυτότητα του με κάτι που έχει στην κατοχή του όπως, για παράδειγμα, είναι μια κάρτα ταυτότητας ή μια σκυτάλη ασφάλειας.

Θεωρητικά, παρέχεται ένα υψηλότερο επίπεδο ασφάλειας στην περίπτωση που χρησιμοποιούνται δύο ή παραπάνω παράγοντες αυθεντικοποίησης.

Η επιλογή των χρησιμοποιούμενων παραγόντων αυθεντικοποίησης γίνεται πάντοτε ανάλογα με τις απαιτήσεις που υπάρχουν, αλλά και σύμφωνα με το πρωτόκολλο που

ακολουθείται για την πιστοποίηση της ταυτότητας των χρηστών ενός συστήματος με ελεγχόμενη πρόσβαση.

Η περίπτωση της τραπεζικής κάρτας και του κωδικού PIN είναι πιθανότατα ο πλέον γνωστός και ταυτόχρονα ο συχνότερα χρησιμοποιούμενος συνδυασμός δυο παραγόντων αυθεντικοποίησης. Ένας χρήστης του συστήματος αποκτά πρόσβαση στον τραπεζικό του λογαριασμό μόνο αν έχει στην κατοχή του την τραπεζική του κάρτα, ενώ παράλληλα γνωρίζει τον κωδικό PIN που αντιστοιχεί στη συγκεκριμένη κάρτα.

6.2 Σκυτάλες ασφάλειας

Μια σκυτάλη ασφάλειας είναι μια φυσική συσκευή, την οποία έχει στην κατοχή του ο εξουσιοδοτημένος χρήστης μιας ηλεκτρονικής υπηρεσίας, και την χρησιμοποιεί σε συνδυασμό με κάποιον παράγοντα αυθεντικοποίησης για να αποδείξει ότι είναι πράγματι αυτός που ισχυρίζεται ότι είναι.

Συνήθως, μια σκυτάλη περιέχει αποθηκευμένο κάποιο κρυπτογραφικό κλειδί ή/και έχει τη δυνατότητα εκτέλεσης κάποιας συνάρτησης παραγωγής κωδικών. Ακόμη, είναι πολύ πιθανό να απαιτείται από την πλευρά του χρήστη η εισαγωγή κάποιου κωδικού ή ενός PIN με το οποίο ενεργοποιείται η σκυτάλη. Με αυτό τον τρόπο η σκυτάλη προστατεύεται σε περίπτωση κλοπής και, κατ' επέκταση, προστατεύεται και ο κάτοχος της σκυτάλης.

Υπάρχουν αρκετοί πιθανοί τρόποι με τους οποίους υλοποιείται μια σκυτάλη ασφάλειας, με πιο συνηθισμένες μορφές αυτές της:

- Έξυπνης Κάρτας(Smart Card). Η κάρτα συνδυάζεται με μια συσκευή ανάγνωσης καρτών, δίνοντας έτσι τη δυνατότητα σύνδεσης με το PC του χρήστη. Σε ορισμένες υλοποιήσεις του αναγνώστη δίνεται στο χρήστη η δυνατότητα να επικυρώσει τις συναλλαγές του, καθώς μπορεί να εισάγει το χρηματικό ποσό που απαιτείται για κάθε μια από τις αυτές.
- Συσκευής USB Stick.
- Θήκης Κλειδιών(Key Fob), που λειτουργεί ως γεννήτρια κωδικών.

Στην εικόνα που ακολουθεί παρουσιάζονται συγκεντρωμένες οι πιο συνηθισμένες υλοποιήσεις με τις οποίες συναντάμε τις σκυτάλες ασφάλειας.

Εικόνα 112: Διάφορες υλοποιήσεις μιας σκυτάλης ασφάλειας

Όμως, σε όλες τις ανωτέρω περιπτώσεις ο χρήστης υποχρεώνεται να έχει μαζί του μία παραπάνω συσκευή, ενώ και το κόστος της συσκευής περνά τελικώς στο χρήστη και τον επιβαρύνει οικονομικά.

Γι' αυτό και τον τελευταίο καιρό εμφανίζονται ολοένα και πιο συχνά σκυτάλες ασφάλειας που λαμβάνουν τη μορφή λογισμικού, με μια προτίμηση να εμφανίζεται προς τις εφαρμογές που εκτελούνται στην πλατφόρμα J2ME που συναντάμε σε συσκευές κινητών τηλεφώνων.

Η πλατφόρμα J2ME υποστηρίζεται ευρέως από την πλειοψηφία των κατασκευαστών κινητών τηλεφώνων, οι οποίοι και την εγκαθιστούν σε όλες τις συσκευές που παράγουν. Με αυτό τον τρόπο, σχεδόν όλα τα κινητά τηλέφωνα μπορούν να εκτελέσουν το κατάλληλο πρόγραμμα και να ενσωματώσουν τη λειτουργία τη σκυτάλης ασφάλειας.

Πέρα όμως από την ευκολία που παρέχεται από το περιβάλλον J2ME πρέπει να αναφέρουμε ότι η υλοποίηση μια σκυτάλης ασφάλειας μέσω ενός κινητού τηλεφώνου παρουσιάζει και τα ακόλουθα θετικά στοιχεία:

- Είναι αρκετά οικονομικότερη από τις άλλες υλοποιήσεις, αφού υποτίθεται ότι μια συσκευή κινητού τηλεφώνου βρίσκεται ήδη στην κατοχή του χρήστη και έτσι δεν απαιτείται η αγορά επιπλέον υλικού.
- Ο χρήστης έχει σχεδόν πάντα το κινητό τηλέφωνο μαζί του, έτσι γλιτώνει από τη μεταφορά μιας επιπλέον συσκευής ή μιας επιπλέον κάρτας που θα έχει στο πορτοφόλι του.

- Το κινητό τηλέφωνο έχει περισσότερη επεξεργαστική ισχύ απ' ότι οι άλλες υλοποιήσεις και σε συνδυασμό με την πλατφόρμα J2ME παρέχει μεγαλύτερη ευελιξία για την ανάπτυξη μιας εξειδικευμένης εφαρμογής.
- Οι εφαρμογές που προορίζονται για την πλατφόρμα J2ME εκτελούνται σε ένα δικό τους sandbox, δηλαδή έχουν μια ποσότητα μνήμης και μια περιοχή αποθήκευσης ειδικά για αυτές. Με αυτό τον τρόπο αποτρέπεται η ανεξέλεγκτη πρόσβαση από και προς άλλες εφαρμογές J2ME, ενώ παράλληλα διασφαλίζονται και οι υπόλοιπες λειτουργίες του τηλεφώνου.
- Η συσκευή του τηλεφώνου είναι ανεξάρτητη από το PC του χρήστη, έτσι μένει ανεπηρέαστη σε μια περίπτωση όπου το PC θα μολυνθεί από κακόβουλο λογισμικό.
- Το δίκτυο κινητής τηλεφωνίας παρέχει ένα ξεχωριστό κανάλι επικοινωνίας, δίνοντας έτσι μια εναλλακτική λύση στην περίπτωση όπου η επικοινωνία μέσα από το Διαδίκτυο κρίνεται ανασφαλής. Ακόμη, πρέπει να αναφέρουμε ότι συνήθως υπάρχει ικανοποιητική κάλυψη από το δίκτυο κινητής τηλεφωνίας και παράλληλα παρατηρούνται μηδαμινά ποσοστά λανθασμένης δρομολόγησης κλήσεων ή μηνυμάτων SMS, έτσι υπάρχει η βεβαιότητα ότι ο χρήστης θα έχει πρόσβαση στο δίκτυο και θα λαμβάνει ότι προορίζεται για αυτόν.

Τα δύο τελευταία στοιχεία είναι ιδιαιτέρως σημαντικά και θα αναδείξουμε τη χρησιμότητα τους σε ένα από τα ακόλουθα υποκεφάλαια, όπου και θα εξετάσουμε πόσο ασφαλής είναι η υλοποίηση του πρωτοκόλλου απομακρυσμένης πιστοποίησης ταυτότητας.

Αλλά, εκτός από τα πλεονεκτήματα που αναφέραμε, υπάρχουν και συγκεκριμένα μειονεκτήματα που εμφανίζονται σε μια σκυτάλη ασφάλειας που υλοποιείται μέσω ενός κινητού τηλεφώνου. Τα πιο σημαντικά από αυτά είναι:

- Η αυτονομία της μπαταρίας που τροφοδοτεί τη συσκευή του κινητού τηλεφώνου. Το συγκεκριμένο στοιχείο αποφορτίζεται γρηγορότερα, σε σύγκριση πάντα με ότι συμβαίνει σε σκυτάλες ασφάλειας διαφορετικής μορφής. Αυτό το γεγονός οφείλεται στους πολλαπλούς ρόλους που διαδραματίζει το κινητό τηλέφωνο, αφού δεν χρησιμοποιείται αποκλειστικά ως σκυτάλη ασφάλειας.
- Η ανθεκτικότητα του κινητού τηλεφώνου, καθώς είναι περισσότερο ευπαθές απ' ότι οι άλλες υλοποιήσεις μιας σκυτάλης. Μετά από πολυετή χρήση ενδέχεται ένα κινητό τηλέφωνο να μην λειτουργεί σωστά και να αποτρέπει τη χρησιμοποίηση του ως σκυτάλη ασφάλειας, σε αντίθεση με μια έξυπνη κάρτα που εκτίθεται λιγότερο και έχει ελάχιστες φθορές.

Βέβαια, τα δύο προαναφερθέντα μειονεκτήματα εξαλείφονται σε μεγάλο βαθμό όταν οι κάτοχοι των κινητών τηλεφώνων προνοούν και αυτοί για τη σωστή λειτουργία των συσκευών τους, είτε μέσω της έγκαιρης φόρτισης της μπαταρίας ή με την επιδιόρθωση πιθανών βλαβών που θα παρουσιαστούν.

6.3 Πρωτόκολλο για την απομακρυσμένη πιστοποίηση ταυτότητας

Το πρωτόκολλο που υλοποιήθηκε βασίζεται στο μοντέλο πελάτη-εξυπηρετητή(Client-Server Model), έτσι όλα εκτυλίσσονται σύμφωνα με ένα σενάριο στο οποίο συμμετέχουν δύο οντότητες:

- Η μια οντότητα επιθυμεί αρχικά να πιστοποιήσει την ταυτότητα της και εν συνεχεία να εισέλθει σε ένα περιβάλλον με περιορισμούς πρόσβασης. Για παράδειγμα, ο πελάτης μιας τράπεζας που επιχειρεί να εισέλθει σε μια ιστοσελίδα για τη διαχείριση τραπεζικού του λογαριασμού.
- Η δεύτερη οντότητα είναι εκείνη που φιλοξενεί ένα περιβάλλον στο οποίο επιτρέπει την πρόσβαση μόνο σε όσους έχουν την ανάλογη εξουσιοδότηση. Ένα τυπικό παράδειγμα είναι ο εξυπηρετητής που χρησιμοποιείται για τις ηλεκτρονικές συναλλαγές των πελατών μιας τράπεζας.

Στο αναπτυχθέν πρωτόκολλο η πιστοποίηση ταυτότητας γίνεται με δύο παράγοντες, οπότε η πλευρά του πελάτη καλείται να αποδείξει στην πλευρά του εξυπηρετητή ότι γνωρίζει το σωστό μυστικό κωδικό και παράλληλα έχει στην κατοχή της την κατάλληλη σκυτάλη ασφάλειας.

Η επικοινωνία ανάμεσα στις οντότητες που συμμετέχουν πραγματοποιείται μέσα από δύο διαφορετικά κανάλια επικοινωνίας. Το ένα εκ των δύο καναλιών είναι το Διαδίκτυο και το άλλο είναι το δίκτυο κινητής τηλεφωνίας.

Η πιστοποίηση της ταυτότητας του πελάτη επιτυγχάνεται με μια σειρά από βήματα που βασίζονται σε ένα συνδυασμό των τεχνικών One-Time Password και Challenge-Response. Όμως θα αφήσουμε για αργότερα την ανάλυση της συγκεκριμένης διαδικασίας, αφού για την ώρα προέχει η παρουσίαση των προϋποθέσεων που απαιτούνται για την ομαλή λειτουργία του πρωτοκόλλου.

Ξεκινάμε από την πλευρά ενός πελάτη που επιθυμεί την πρόσβαση σε ένα φυλασσόμενο περιβάλλον, η οποία οφείλει:

- Να έχει πρόσβαση στο διαδίκτυο.
- Να γνωρίζει το σωστό μυστικό κωδικό, τον οποίο εισάγει στο web interface που παρέχεται από τον εξυπηρετητή.
- Να έχει στην κατοχή της μια σκυτάλη ασφάλειας, όπου στην προκειμένη περίπτωση είναι ένα κινητό τηλέφωνο στο οποίο εκτελείται μια εφαρμογή για το J2ME Runtime Environment.
- Να αποδείξει την κατάλληλη στιγμή, όταν δηλαδή ζητηθεί από τον εξυπηρετητή, ότι πράγματι κατέχει τη χρησιμοποιούμενη σκυτάλη ασφάλειας.

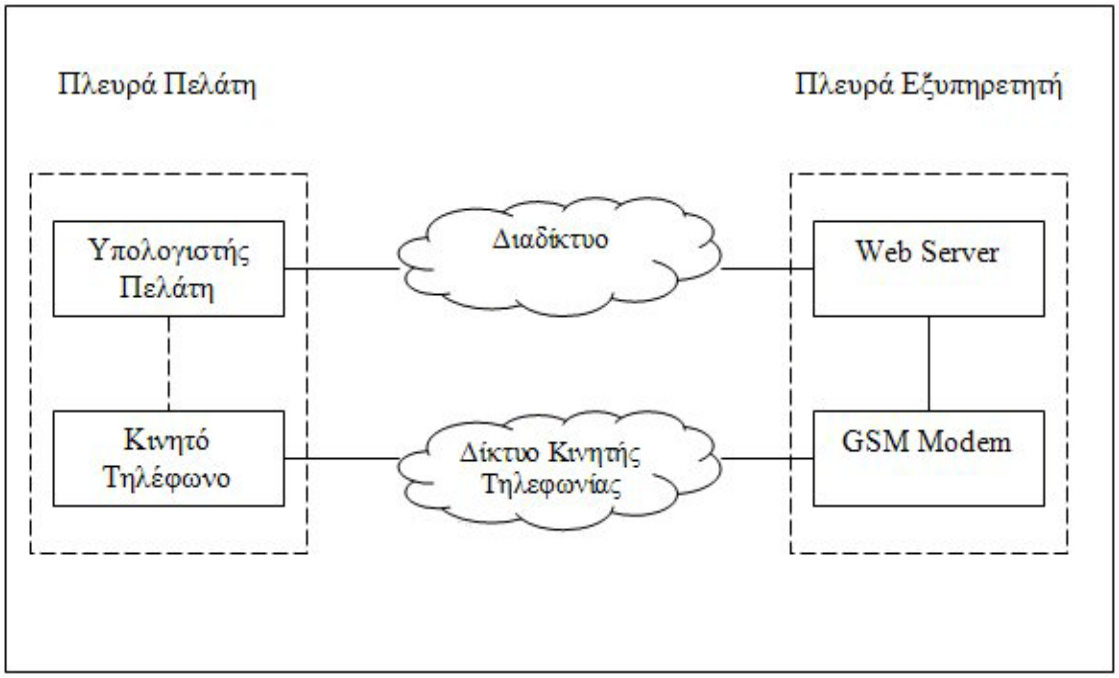
Ενώ από την άλλη, η πλευρά του εξυπηρετητή οφείλει:

- Να έχει πρόσβαση στο Διαδίκτυο
- Να φιλοξενεί την κατάλληλη διαδικτυακή εφαρμογή, μέσα από την οποία θα παρέχει το web interface για την επικοινωνία με την πλευρά του πελάτη. Ακόμη, η διαδικτυακή εφαρμογή θα επεξεργάζεται τα δεδομένα που θα εισέρχονται από την πλευρά του πελάτη και στη συνέχεια θα παράγει τα ανάλογα αποτελέσματα.

- Να παρέχει ένα μηχανισμό για την αποθήκευση των χρησιμοποιούμενων δεδομένων.
- Να υποστηρίζει πλήρως τις συναρτήσεις που εκτελούνται στη σκυτάλη ασφάλειας που έχει στην κατοχή του ο πελάτης
- Να έχει στη διάθεση της κάποια μέθοδο για την ανταλλαγή σύντομων μηνυμάτων με την πλευρά του πελάτη

Η αποστολή και η λήψη μηνυμάτων SMS μέσω του server είναι εφικτή όταν συνδέσουμε σε αυτόν μια συσκευή GSM modem ή ένα κινητό τηλέφωνο που έχει ενσωματωμένη τη συγκεκριμένη λειτουργία.

Σε αυτό το σημείο θα ήταν καλό να δείξουμε συνοπτικά τη μορφή που λαμβάνει το πρωτόκολλο. Έτσι, στην εικόνα που ακολουθεί παρουσιάζονται αρκετά απλά, αποκρύπτοντας την εσωτερική τους πολυπλοκότητα, όλα τα στοιχεία που συμμετέχουν στο πρωτόκολλο για την απομακρυσμένη πιστοποίηση ταυτότητας.



Εικόνα 113: Συνοπτική παρουσίαση του πρωτοκόλλου

Το πιο πολύπλοκο από όλα τα κομμάτια του πρωτοκόλλου είναι η διαδικτυακή εφαρμογή που φιλοξενείται από τον web server.

Αυτός είναι και ο λόγος για τον οποίο θα αφιερώσουμε την ακόλουθη υποενότητα στο συγκεκριμένο τμήμα, για να παρουσιάσουμε πρώτα τον τρόπο με τον οποίο αυτό δομείται και να αναλύσουμε εκτενέστερα ότι λαμβάνει χώρα εκεί. Στη συνέχεια, θα παρουσιάσουμε σε ξεχωριστές υποενότητες και τα υπόλοιπα τμήματα του πρωτοκόλλου.

6.3.1 Η αναπτυχθείσα Διαδικτυακή Εφαρμογή

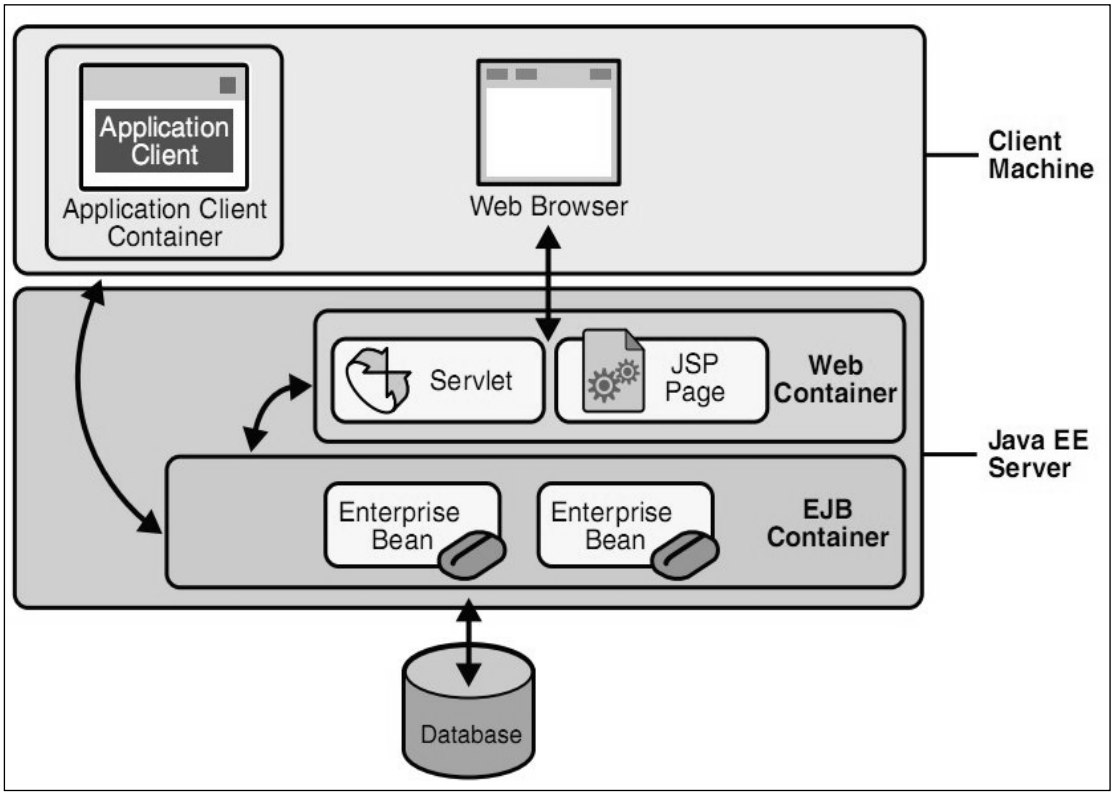
Είδαμε σε προηγούμενο κεφάλαιο ότι υπάρχει μια ειδική εκδοχή της γλώσσας προγραμματισμού Java, η οποία ονομάζεται Java Enterprise Edition (Java EE), και μέσω αυτής δημιουργούνται εφαρμογές πολλαπλών επιπέδων (multi-tiered) που εγκαθίστανται σε κάποιον application server και δίνουν τη δυνατότητα κατανεμημένης εκτέλεσης.

Είναι αρκετά συνηθισμένη η περίπτωση κατά την οποία αναπτύσσονται εφαρμογές που ακολουθούν το μοντέλο πελάτη – εξυπηρετητή(Client – Server Model) και οργανώνονται σε τρία ξεχωριστά επίπεδα, σχηματίζοντας τις λεγόμενες three-tier applications.

Η αρχιτεκτονική μιας three-tier Java EE εφαρμογής περιλαμβάνει:

- Το πρώτο επίπεδο, όπου φιλοξενείται η πλευρά του πελάτη(Client).
- Το δεύτερο επίπεδο, όπου παρέχεται ένα ενδιάμεσο στρώμα λογισμικού από την πλευρά του εξυπηρετητή(Server).
- Το τρίτο επίπεδο, όπου βρίσκονται αποθηκευμένα τα δεδομένα μιας διαδικτυακής εφαρμογής.

Η σχηματική απεικόνιση της αρχιτεκτονικής μιας Java EE three-tiered εφαρμογής είναι παρόμοια με την ακόλουθη εικόνα.



Εικόνα 114: Η αρχιτεκτονική ενός three-tiered application σύμφωνα με την πλατφόρμα Java EE

Σε αυτό το σημείο θα ήταν καλύτερα να δούμε περισσότερο αναλυτικά τι περιλαμβάνεται σε κάθε ένα από τα επίπεδα της εφαρμογής που έχει αναπτυχθεί.

Πρώτο Επίπεδο

Το πρώτο επίπεδο αποτελεί το σημείο όπου δίνεται στον πελάτη η δυνατότητα πρόσβασης στις υπηρεσίες που παρέχει ο εξυπηρετητής.

Ο πελάτης χρησιμοποιεί ένα πρόγραμμα πλοήγησης στο διαδίκτυο(web browser) είτε κάποιο άλλο πρόγραμμα που εκτελείται στον υπολογιστή του, μέσω του οποίου έχει στη διάθεση του ένα γραφικό περιβάλλον με μια σειρά από επιλογές.

Ανάλογα με τις ενέργειες που εκτελούνται από την πλευρά του πελάτη έχουμε την αντίστοιχη αποστολή αιτημάτων(requests) για την παροχή υπηρεσιών από την πλευρά του εξυπηρετητή.

Στην περίπτωση της εφαρμογής που έχουμε αναπτύξει δίνεται σε πρώτη φάση η δυνατότητα πρόσβασης μέσω ενός web browser, όμως στη συνέχεια απαιτείται η παρουσία του κινητού τηλεφώνου του πελάτη και η εκτέλεση του MIDlet, με το οποίο η φορητή συσκευή θα μετατραπεί σε σκυτάλη αυθεντικοποίησης.

Δεύτερο Επίπεδο

Εδώ συναντάμε το Java EE Server, με άλλα λόγια βρίσκουμε το περιβάλλον εκτέλεσης της πλατφόρμας Java EE, το οποίο περιέχει τα απαραίτητα APIs και εργαλεία διαχείρισης-υποδοχής(containers) που επιτρέπουν την εκτέλεση κλάσεων, όπως είναι τα Java Servlets και τα Enterprise Beans, αλλά και την παροχή σελίδων όπως οι Java Server Pages(JSP).

Ως περιβάλλον εκτέλεσης επιλέχθηκε ο Apache Tomcat Server ή Apache Tomcat ή απλούστερα Tomcat.

Ο Apache Tomcat παρέχεται δωρεάν από το Apache Software Foundation και αποτελεί ένα open source servlet container που παράλληλα ενσωματώνει έναν HTTP Server υλοποιημένο σε Java.

Ανάλογα με τις απαιτήσεις και το σχεδιασμό κάθε εφαρμογής επιλέγεται και η παρουσία διαφορετικών δομικών στοιχείων. Η αναπτυχθείσα εφαρμογή εκμεταλλεύεται τα πλεονεκτήματα που παρέχονται από το συνδυασμό των σελίδων JSP και των κλάσεων Java Servlets ή συντομότερα Servlets.

Σαν Servlet λογίζεται μια κλάση Java που έχει δημιουργηθεί σύμφωνα με το Java Servlet API και δρα ανάμεσα στο πρώτο και το τρίτο επίπεδο μιας διαδικτυακής εφαρμογής.

Ίσως η σημαντικότερη δυνατότητα που έχει ένα Servlet είναι να δημιουργεί δυναμικές ιστοσελίδες, καθώς κατά την εκτέλεση του μπορεί και παράγει κώδικα HTML. Συνοπτικά, στις δυνατότητες που έχει ένα Servlet περιλαμβάνονται:

- Η ανάγνωση δεδομένων που έρχονται από τον πελάτη. Συνήθως πρόκειται για τις πληροφορίες με τις οποίες ο πελάτης συμπληρώνει μια φόρμα HTML που βρίσκεται σε κάποια ιστοσελίδα. Υπάρχει όμως και η πιθανότητα τα δεδομένα να προέρχονται από κάποιο άλλο πρόγραμμα που εκτελείται στον υπολογιστή του πελάτη.

- Η ανάγνωση αιτημάτων που αποστέλλονται από ένα πρόγραμμα web browser. Πρόκειται για δεδομένα που είναι περισσότερο κατανοητά σε ένα πρόγραμμα αυτού του τύπου, όπως είναι οι επικεφαλίδες του πρωτοκόλλου HTTP και πληροφορίες όπως τα web cookies.

- Η δημιουργία αποτελεσμάτων. Σε αυτή την περίπτωση ένα Servlet λειτουργεί ως συνδετικός κρίκος ανάμεσα στο πρώτο και το τρίτο επίπεδο, αφού για κάθε αίτημα που δέχεται κάνει ότι είναι απαραίτητο για να σχηματίσει την αντίστοιχη απάντηση.

  Από την πολυπλοκότητα κάθε αιτήματος εξαρτάται και το πλήθος των ενεργειών που απαιτούνται από ένα Servlet.

  Για παράδειγμα, σε μια απλή περίπτωση, ένα Servlet ενδέχεται να χρησιμοποιείται για τον υπολογισμό μιας αριθμητικής πράξης.

  Όμως, σε μια περισσότερο πολύπλοκη περίπτωση, αρχικά το Servlet δέχεται ένα αίτημα που περιέχει τα δεδομένα από μια HTML φόρμα που βρίσκεται στο πρώτο επίπεδο, στη συνέχεια δημιουργεί ένα ερώτημα SQL και επικοινωνεί με μια βάση δεδομένων που βρίσκεται στο τρίτο επίπεδο. Κατόπιν δέχεται το αποτέλεσμα του ερωτήματος SQL, το μετατρέπει σε σελίδα HTML και το προωθεί ως απάντηση στο πρώτο επίπεδο.

- Η αποστολή αρχείων προς την πλευρά του πελάτη. Σε αυτή την περίπτωση ένα Servlet αναλαμβάνει την αποστολή ενός συγκεκριμένου format αρχείου, το οποίο μπορεί να είναι κείμενο με τη μορφή HTML ή XML, ή να είναι ένα λογιστικό φύλλο ή ένα αρχείο εικόνας.

- Η αποστολή απαντήσεων προς την πλευρά ενός προγράμματος web browser. Πρόκειται για την αποστολή δεδομένων όπως είναι οι επικεφαλίδες του πρωτοκόλλου HTTP και ρυθμίσεις σχετικές με τα cookies και το caching σελίδων.

Όμως, το γεγονός ότι τα Servlets είναι καθαρές κλάσεις Java, καθιστά δύσχρηστη τη σχεδίαση σελίδων με μεγάλα στατικά τμήματα HTML και γενικότερα δεν ευνοεί την παρουσίαση μιας διαδικτυακής εφαρμογής. Το συγκεκριμένο θέμα καλύπτεται με όσα προσφέρουν οι σελίδες JSP.

Οι σελίδες JSP είναι κατά βάση σελίδες HTML που έχουν τη δυνατότητα ενσωμάτωσης κώδικα Java και μιας σειράς από tags που είναι φτιαγμένα ειδικά για αυτές. Αυτό το χαρακτηριστικό επιτρέπει την ευκολότερη σχεδίαση ιστοσελίδων και οδηγεί τελικά στην καλύτερη παρουσίαση μιας διαδικτυακής εφαρμογής.

Αυτός είναι και ο κυριότερος λόγος για τον οποίο προτιμάται η δημιουργία διαδικτυακών εφαρμογών μέσα από το συνδυασμό των δύο τεχνολογιών, ενώ παράλληλα διαχωρίζονται ως ένα βαθμό τα καθήκοντα που αναλαμβάνονται από το κάθε τμήμα της εφαρμογής.

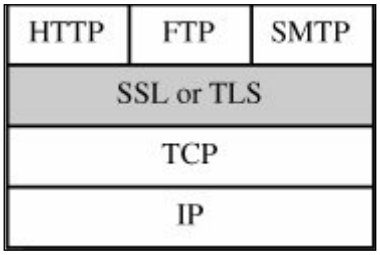
Έτσι έχουμε τις σελίδες JSP να αναλαμβάνουν κυρίως την παρουσίαση όπου, για παράδειγμα τους ανατίθεται η δημιουργία σελίδων HTML για την προβολή απαντήσεων στα ερωτήματα των πελατών. Ενώ από την πλευρά τους τα Servlets αναλαμβάνουν τις ενέργειες που απαιτούνται για την επεξεργασία, όπως είναι η ανάγνωση αιτημάτων και η επικοινωνία με βάσεις δεδομένων.

Βέβαια, για να είμαστε περισσότερο ακριβείς, ο web container διαχειρίζεται με παρόμοιο τρόπο και τα δύο συστατικά της εφαρμογής μας. Καθώς στο παρασκήνιο του Apache Tomcat οι σελίδες JSP μετατρέπονται και αυτές σε Servlets, οπότε στην ουσία έχουμε να κάνουμε με μια εφαρμογή που αποτελείται από Servlets.

Επιπλέον, αξίζει να αναφέρουμε μια ακόμα πολύ σημαντική δυνατότητα που παρέχεται από τον Apache Tomcat, πέρα από το γεγονός ότι αποτελεί το ιδανικό περιβάλλον για την εκτέλεση Servlets και JSP.

Η εν λόγω δυνατότητα είναι η δημιουργία, με έναν πολύ εύκολο τρόπο όπως θα δούμε παρακάτω, και η διαχείριση ψηφιακών ασφαλών συνδέσεων που βασίζονται στο πρωτόκολλο Secure Socket Layer(SSL) ή, όπως αργότερα μετονομάστηκε, Transport Layer Security(TLS).

Το συγκεκριμένο πρωτόκολλο είναι διαθέσιμο ως ένα επιπλέον επίπεδο στη στοίβα με τα πρωτόκολλα TCP/IP. Όπως παρουσιάζεται και στην εικόνα που ακολουθεί, το SSL/TLS εισάγεται μεταξύ του επιπέδου των εφαρμογών και του επιπέδου ελέγχου μετάδοσης.



Εικόνα 115: Η προσθήκη του πρωτοκόλλου SSL/TLS στη στοίβα των πρωτοκόλλων TCP/IP

Το SSL/TLS αποτελεί το πιο δημοφιλές πρότυπο για τη διασφάλιση των επικοινωνιών που διεξάγονται μέσα από ανοικτά δίκτυα, όπως είναι το Internet. Οι διαδικτυακές εφαρμογές που εκμεταλλεύονται τη χρήση του SSL/TLS, μέσω του πρωτοκόλλου HTTPS ή HTTP πάνω από SSL, παρέχουν συνδέσεις που εγγυώνται την ασφάλεια και την ακεραιότητα των πληροφοριών που διακινούνται σε αυτές.

Απαραίτητη προϋπόθεση για τη δημιουργία συνδέσεων SSL/TLS είναι η ύπαρξη ψηφιακών πιστοποιητικών, όπου αναγκαστικά από την πλευρά του server συναντούμε κάποιο πιστοποιητικό, ενώ σε ορισμένες περιπτώσεις απαιτείται και από την πλευρά του client να παρέχει το δικό της ψηφιακό πιστοποιητικό.

Στην περίπτωση της δικής μας διαδικτυακής εφαρμογής δημιουργούμε, υπογράφουμε ψηφιακά και εγκαθιστούμε στην πλευρά του server ένα δικό μας πιστοποιητικό. Η συγκεκριμένη πρακτική πρέπει να αποφεύγεται σε πραγματικές συνθήκες, όπου σύμφωνα με την ορθή λογική πρέπει να αγοράσουμε το πιστοποιητικό από μιαν έμπιστη ανεξάρτητη αρχή που εκδίδει πιστοποιητικά, αλλά για τις ανάγκες της εργασίας και μόνο καταφεύγουμε σε αυτή την εύκολη, γρήγορη και οικονομική λύση.

Για την εγκαθίδρυση μιας σύνδεσης SSL/TLS απαιτείται η ολοκλήρωση μιας διαδικασίας που ονομάζεται handshake(χειραψία), μέσω της οποίας οι δύο πλευρές που πρόκειται να χρησιμοποιήσουν τη σύνδεση συμφωνούν σε ορισμένες παραμέτρους που σχετίζονται με αυτήν.

Έτσι λοιπόν, ένα handshake συμβαίνει όταν από την πλευρά του client, συνήθως μέσω ενός web browser, υπάρχει η επιθυμία για σύνδεση με ασφαλή τρόπο με ένα συγκεκριμένο site. Τότε συμβαίνουν οι ακόλουθες ενέργειες:

- Ο browser στέλνει ένα αίτημα για μιαν ασφαλή συνεδρία, συνήθως ζητώντας ένα URL που ξεκινά με HTTPS αντί για το συνηθισμένο HTTP.
- Ο server απαντά με την αποστολή του ψηφιακού του πιστοποιητικού, όπου εκεί περιέχεται και το δημόσιο κλειδί του.
- Ο browser επιβεβαιώνει ότι το πιστοποιητικό του server είναι έγκυρο και είναι υπογεγραμμένο από μιαν έμπιστη αρχή έκδοσης πιστοποιητικών.
- Εφόσον το πιστοποιητικό του server είναι έγκυρο, τότε ο browser είναι βέβαιος για την ταυτότητα του server, οπότε προχωρά στη δημιουργία ενός κλειδιού μόνο για την τρέχουσα συνεδρία, το οποίο το κρυπτογραφεί με το δημόσιο κλειδί του server και στη συνέχεια του το αποστέλλει.
- Ο server αποκρυπτογραφεί το μήνυμα χρησιμοποιώντας το ιδιωτικό του κλειδί και ανακτά το κλειδί που πρόκειται να χρησιμοποιηθεί για την τρέχουσα συνεδρία.
- Από αυτό το σημείο και ύστερα μόνο ο client και ο server κατέχουν αντίγραφα του κλειδιού συνεδρίας, οπότε θεωρούμε ότι η μεταξύ τους επικοινωνία μπορεί να διεξαχθεί με ασφάλεια.

Σε ένα από τα παραρτήματα που ακολουθούν θα δούμε αναλυτικότερα τον τρόπο με τον οποίο δημιουργούμε ένα ψηφιακό πιστοποιητικό και πως, στη συνέχεια, ρυθμίζονται οι ασφαλείς συνδέσεις μέσω του πρωτοκόλλου SSL/TLS.

Τέλος, οφείλουμε να τονίσουμε για μια ακόμη φορά τη βαρύνουσα σημασία που έχει το ρητό που αντικατοπτρίζει τη γενικότερη λογική της γλώσσας Java. Ο λόγος βέβαια για το “Write Once, Run Anywhere”, το οποίο εκμεταλλευόμαστε στην πράξη και ενσωματώνουμε στη web εφαρμογή τα εξής στοιχεία:

- Το σύνολο των κρυπτογραφικών κλάσεων που αναπτύχθηκαν σε προηγούμενη φάση, κατά τη δημιουργία του MIDlet που επιτρέπει την ασφαλή ανταλλαγή σύντομων μηνυμάτων.
- Το Java Communications API, με βάση το οποίο αναπτύχθηκε μια ακόμη εφαρμογή που αναλαμβάνει το ρόλο της γέφυρας επικοινωνίας ανάμεσα στο web application και σε ένα κινητό τηλέφωνο που λειτουργεί ως GSM Modem.
- Το driver MySQL Connector/J, που χρησιμοποιείται σε συνδυασμό με το API Java Data Base Connectivity(JDBC) και επιτρέπει την καλύτερη επικοινωνία του δευτέρου επιπέδου με το σύστημα διαχείρισης βάσεων δεδομένων που έχουμε επιλέξει για το τρίτο επίπεδο.

Τρίτο Επίπεδο

Το τρίτο επίπεδο είναι με σιγουριά το πιο σημαντικό και αποτελεί τη βάση για τις διαδικτυακές εφαρμογές πολλαπλών επιπέδων, καθώς εκεί αποθηκεύονται όλα τα δεδομένα που χρησιμοποιούνται από κάθε εφαρμογή αυτού του τύπου. Άλλωστε, χωρίς την ύπαρξη των δεδομένων δεν θα υπήρχε η ανάγκη για τη γενικότερη δημιουργία εφαρμογών.

Η αποθήκευση των δεδομένων του τρίτου επιπέδου δεν πραγματοποιείται με κάποιον αυθαίρετο τρόπο, αντιθέτως, χρησιμοποιείται μια συγκεκριμένη μέθοδος με την οποία τα δεδομένα συλλέγονται και καταχωρούνται σε πίνακες. Ο πιο εύκολος τρόπος για να αντιληφθούμε τη μορφή μιας βάσης δεδομένων είναι αν θεωρήσουμε ότι αυτή αποτελείται από μια συλλογή από πίνακες.

Ο κάθε πίνακας αποτελείται από έναν αριθμό στηλών και γραμμών.

Κάθε στήλη του πίνακα, που ονομάζεται εναλλακτικά και πεδίο, αντιπροσωπεύει και μια ξεχωριστή ποσότητα πληροφορίας, όπως είναι για παράδειγμα το μικρό όνομα ενός πελάτη ή η τιμή πώλησης ενός προϊόντος.

Κάθε γραμμή περιλαμβάνει όλα τα απαιτούμενα δεδομένα που θα σχηματίσουν μιαν εγγραφή της βάσης δεδομένων, όπως θα ήταν για παράδειγμα ένα πεδίο με το όνομα ενός προϊόντος και ένα πεδίο με το κόστος του συγκεκριμένου προϊόντος.

Στην περίπτωση κατά την οποία οι πίνακες μιας βάσης μοιράζονται δεδομένα μέσα από μερικά κοινά πεδία λέμε ότι υπάρχει μεταξύ τους ένας συσχετισμός, και κατ' επέκταση δημιουργείται μια Σχεσιακή Βάση Δεδομένων(Relational Data Base).

Για την καλύτερη λειτουργία και την ευκολότερη διαχείριση των σχεσιακών βάσεων προτιμάται η χρησιμοποίηση ενός Συστήματος Διαχείρισης Σχεσιακών Βάσεων Δεδομένων(Relational Data Base Management System - RDBMS).

Το RDBMS περιλαμβάνει ένα σύνολο προγραμμάτων που δίνουν τη δυνατότητα οργάνωσης, αποθήκευσης, διαχείρισης και ανάκτησης των δεδομένων μιας βάσης.

Όλες αυτές οι λειτουργίες πραγματοποιούνται με την αποστολή των κατάλληλων εντολών προς το RDBMS. Οι εν λόγω εντολές ονομάζονται ερωτήματα(queries) και συντάσσονται σύμφωνα με όσα ορίζονται από τη γλώσσα δομημένων ερωτημάτων Standard Query Language(SQL).

Για την αναπτυχθείσα διαδικτυακή εφαρμογή επιλέχθηκε το σύστημα διαχείρισης MySQL, το οποίο διατίθεται ελεύθερα από την ομώνυμη εταιρεία λογισμικού.

Η επικοινωνία του MySQL με το δεύτερο επίπεδο πραγματοποιείται με τη βοήθεια του driver MySQL Connector/J και του API JDBC, μέσω των οποίων επιτρέπεται η αποστολή SQL queries προς το RDBMS και η λήψη των αντίστοιχων απαντήσεων.

6.3.2 Ο ρόλος του GSM Modem

Ένα GSM Modem είναι στην ουσία μια συσκευή που συνδέεται με έναν προσωπικό υπολογιστή, όπως όλα τα modems άλλωστε, με τη διαφορά ότι στη συγκεκριμένη συσκευή εγκαθίσταται μια κάρτα SIM, επιτρέποντας έτσι την πρόσβαση σε ένα δίκτυο κινητής τηλεφωνίας.

Η επικοινωνία ανάμεσα στον υπολογιστή και το συνδεδεμένο σε αυτόν GSM Modem διεξάγεται μέσω των εντολών AT. Η πλευρά του υπολογιστή στέλνει τις επιθυμητές εντολές στο GSM Modem και εκείνο με τη σειρά του επιστρέφει στον υπολογιστή το αποτέλεσμα κάθε εντολής.

Το κυριότερο πλεονέκτημα που παρέχεται από ένα GSM Modem είναι η πλήρης υποστήριξη των εντολών AT, όμως η συγκεκριμένη δυνατότητα συνοδεύεται από το ανάλογο τίμημα, καθώς το κόστος απόκτησης μιας τέτοιας συσκευής κυμαίνεται από μερικές εκατοντάδες ευρώ, για τις φθηνότερες υλοποιήσεις, μέχρι μερικές χιλιάδες ευρώ με τις οποίες κοστολογούνται οι ακριβότερες λύσεις.

Μια εναλλακτική και αρκετά οικονομικότερη λύση είναι η χρησιμοποίηση ενός κινητού τηλεφώνου. Σχεδόν όλοι οι κατασκευαστές κινητών τηλεφώνων ενσωματώνουν στις συσκευές που παράγουν τη δυνατότητα εκτέλεσης ενός υποσυνόλου των εντολών AT.

Οπότε, κυρίως για λόγους οικονομίας, αλλά και για την αξιοποίηση όσων μάθαμε στο δεύτερο κεφάλαιο, προτιμήσαμε για τη συγκεκριμένη εφαρμογή τη χρησιμοποίηση δύο κινητών τηλεφώνων που έχουν ενσωματωμένη τη λειτουργία του GSM Modem.

Η επιλογή των δύο τηλεφώνων έγινε αναγκαστικά αφού καμία από τις συσκευές δεν υποστήριζε όλες τις απαιτούμενες εντολές AT.

Πιο συγκεκριμένα, η μία συσκευή δεν επέτρεπε την εκτέλεση εντολών για την ανάγνωση του γραμματοκιβωτίου της, οπότε δεν μπορούσαμε να ελέγξουμε αν υπήρχαν καινούργια εισερχόμενα μηνύματα, ενώ η δεύτερη δεν επέτρεπε την εκτέλεση εντολών για την αποστολή τροποποιημένων μηνυμάτων που είχαν παρούσα την επικεφαλίδα User Data Header(UDH).

Η ύπαρξη της τροποποιημένης επικεφαλίδας χρησιμεύει για την αποστολή μηνυμάτων προς τη σκυτάλη ασφάλειας, καθώς τα στοιχεία της επικεφαλίδας υποδεικνύουν τη θύρα στην οποία πρέπει να προωθηθεί το μήνυμα, κάνοντας το έτσι αντιληπτό από την εφαρμογή που εκτελείται στο κινητό τηλέφωνο του πελάτη.

Επιπλέον, με αυτό τον τρόπο αποδείχθηκε ότι είναι προτιμότερη η κατανομή των ρόλων, με τη μία συσκευή να αναλαμβάνει τη λήψη μηνυμάτων και την άλλη να λειτουργεί συμπληρωματικά έχοντας τη ευθύνη για την αποστολή μηνυμάτων.

Για να πραγματοποιηθεί η σύνδεση ανάμεσα στα δύο τηλέφωνα και τον εξυπηρετητή απαιτούνται δύο πράγματα, το ένα από αυτά είναι τα κατάλληλα καλώδια και το άλλο είναι οι modem drivers που αντιστοιχούν στο λειτουργικό σύστημα του εξυπηρετητή.

Οι δύο τηλεφωνικές συσκευές διαθέτουν θύρες Mini-B USB με πέντε ακροδέκτες, έτσι η σύνδεση τους με τον server προκύπτει μέσω καλωδίων που πραγματοποιούν τη μετατροπή από τη θύρα Mini-B στην τυπική μορφή USB Type-A με τους τέσσερις ακροδέκτες.

Οι drivers των δύο συσκευών είναι διαθέσιμοι από τις επίσημες ιστοσελίδες των κατασκευαστών και για την εγκατάσταση τους δεν απαιτείται κάτι πολύπλοκο.

Μετά από την εγκατάσταση του συγκεκριμένου λογισμικού τα δύο τηλέφωνα εμφανίζονται ως modems που είναι συνδεδεμένα σε κάποιες από τις θύρες επικοινωνίας του υπολογιστή.

Από αυτό το σημείο και ύστερα έχει επιτευχθεί η αναγνώριση των συσκευών και, μέσα από ένα πρόγραμμα προσομοίωσης τερματικού, είναι δυνατή η εκτέλεση εντολών AT.

Όμως, η χρησιμοποίηση ενός προγράμματος προσομοίωσης τερματικού μας δίνει τη δυνατότητα για χειροκίνητη εκτέλεση των εντολών AT, ενώ στην πραγματικότητα απαιτείται ένας αυτοματοποιημένος τρόπος εκτέλεσης των εντολών μέσα από την ίδια τη διαδικτυακή εφαρμογή. Η συγκριμένη ανάγκη καλύπτεται με όσα παρέχονται από το API Java Communications.

Χάρις στο συγκεκριμένο API αναπτύχθηκε μια ακόμη εφαρμογή Java, η οποία ανταλλάσει ροές δεδομένων(data streams) με τις θύρες όπου συνδέονται τα κινητά τηλέφωνα, λειτουργώντας έτσι ως γέφυρα επικοινωνίας ανάμεσα στις τηλεφωνικές συσκευές και την ήδη υπάρχουσα διαδικτυακή εφαρμογή.

Η γέφυρα επικοινωνίας ενσωματώνεται στην ευρύτερη διαδικτυακή εφαρμογή και, ανάλογα με μια σειρά από εξωτερικά γεγονότα, εκτελεί τις ακόλουθες λειτουργίες:

- Εξετάζει αν έφτασε κάποιο νέο μήνυμα στο τηλέφωνο που αναλαμβάνει την εισερχόμενη κίνηση. Η συγκεκριμένη ενέργεια ρυθμίζεται από ένα thread που εκτελείται περιοδικά, με ένα χρονικό κενό των 10 δευτερολέπτων, από την πλευρά της διαδικτυακής εφαρμογής.
- Κατά την άφιξη ενός νέου μηνύματος η γέφυρα επικοινωνιών αναλαμβάνει, σε πρώτη φάση, τη μετατροπή της εισερχόμενης κίνησης σε μια μορφή που είναι κατανοητή για την πλευρά της διαδικτυακής εφαρμογής, ενώ στη συνέχεια προωθεί το αποτέλεσμα προς την κατεύθυνση της διαδικτυακής εφαρμογής.
- Μετατρέπει τις πληροφορίες που επιθυμεί να αποστείλει η διαδικτυακή εφαρμογή στη μορφή με την οποία μπορεί να τις αντιληφθεί ένα κινητό τηλέφωνο. Η συγκεκριμένη ενέργεια πραγματοποιείται αφού έχει προηγηθεί μια επιτυχημένη προσπάθεια εισόδου μέσω της διεπαφής της διαδικτυακής εφαρμογής.
- Προωθεί τα δεδομένα που επεξεργάστηκε από την πλευρά της διαδικτυακής εφαρμογής προς το κινητό τηλέφωνο που αναλαμβάνει την εξερχόμενη κίνηση. Στην ουσία, αυτός είναι ο τρόπος με τον οποίο αποστέλλεται ένα SMS από την πλευρά του εξυπηρετητή προς το κινητό τηλέφωνο του πελάτη.

6.3.3 Η σκυτάλη ασφάλειας

Όπως αναφέραμε νωρίτερα, η σκυτάλη ασφάλειας που συμμετέχει στο αναπτυχθέν πρωτόκολλο έχει τη μορφή λογισμικού, πιο συγκεκριμένα, πρόκειται για ένα MIDlet που εκτελείται στο περιβάλλον J2ME που συναντάμε στο κινητό τηλέφωνο της πλευράς του πελάτη.

Η εν λόγω εφαρμογή είναι, κατά βάση, μια τροποποίηση της εφαρμογής που αναπτύχθηκε για τη διεξαγωγή ασφαλούς μετάδοσης σύντομων μηνυμάτων. Έτσι, τα δομικά στοιχεία από τα οποία αποτελείται είναι:

- Η δυνατότητα για αποστολή και λήψη σύντομων μηνυμάτων.
- Η αποθήκευση δεδομένων μέσω του συστήματος Record Management System.
- Ο αλγόριθμος κρυπτογράφησης δημοσίου κλειδιού RSA.
- Η συνάρτηση παραγωγής κλειδιών PBKDF2, σε συνεργασία με τη συνάρτηση κατακερματισμού SHA-256.
- Ο συμμετρικός αλγόριθμος κρυπτογράφησης AES.

Η ουσιαστική διαφορά που εντοπίζεται στο αναπτυχθέν MIDlet, πέρα από τις επιλογές που παρέχονται από το γραφικό του περιβάλλον, είναι το γεγονός ότι οι κρυπτογραφικές συναρτήσεις που ενσωματώνει έχουν ως στόχο τη δημιουργία κωδικών μιας χρήσης.

Ο πελάτης χρησιμοποιεί τους παραγόμενους κωδικούς ως ένα επιπρόσθετο αποδεικτικό στοιχείο και τους αποστέλλει στην πλευρά του εξυπηρετητή κάθε φορά που συμμετέχει στη διαδικασία της απομακρυσμένης πιστοποίησης ταυτότητας.

Καθότι στο πρωτόκολλο συμμετέχουν δύο ξεχωριστοί δίαυλοι επικοινωνίας, όπως είναι το Διαδίκτυο και το δίκτυο κινητής τηλεφωνίας, κατ' επέκταση έχουμε δύο εναλλακτικές οδούς μέσω των οποίων η πλευρά του εξυπηρετητή μπορεί να λάβει τους κωδικούς των πελατών.

Στη μία περίπτωση, όπου χρησιμοποιείται το Διαδίκτυο, ο πελάτης πληκτρολογεί τον κωδικό που παράγεται από το κινητό του στη φόρμα μιας ιστοσελίδας που παρέχεται από τον server. Ενώ στη δεύτερη περίπτωση, όπου χρησιμοποιείται το δίκτυο κινητής τηλεφωνίας, ο πελάτης στέλνει από το κινητό του τηλέφωνο ένα SMS, που περιέχει βεβαίως τον παραγόμενο κωδικό, προς το GSM modem που είναι συνδεδεμένο με τον server.

6.3.4 Ο υπολογιστής του πελάτη

Είδαμε νωρίτερα, όταν και αναλύσαμε τη δομή των εφαρμογών τριών επιπέδων, ότι το τμήμα όπου συναντάμε τον υπολογιστή του πελάτη θεωρείται ως το πρώτο επίπεδο μιας εφαρμογής αυτού του τύπου.

Η ίδια λογική ακολουθήθηκε και για τη διαδικτυακή εφαρμογή που αναπτύχθηκε από την πλευρά μας.

Κατά το σχεδιασμό του πρωτοκόλλου δεν κρίθηκε αναγκαία η ανάπτυξη κάποιας εξειδικευμένης εφαρμογής που θα εκτελείτο στον υπολογιστή του πελάτη, όπως θα ήταν ένα Java Applet ή ένα τυπικό πρόγραμμα Java SE.

Η μόνη ουσιαστική απαίτηση που υπάρχει από το πρωτόκολλο, σχετικά με το λογισμικό που είναι εγκατεστημένο στον υπολογιστή του πελάτη, αφορά την παρουσία ενός προγράμματος web browser.

Όμως ιδιαίτερη προσοχή πρέπει να δοθεί στην παλαιότητα του εγκατεστημένου προγράμματος και να προτιμάται η χρησιμοποίηση της όσο το δυνατόν νεότερης έκδοσης. Καθώς, σε διαφορετική περίπτωση, μια παλιότερη έκδοση ενδέχεται να παρουσιάσει προβλήματα συμβατότητας με τις συνδέσεις TLS που χρησιμοποιούνται για την ασφαλή επικοινωνία με την πλευρά του εξυπηρετητή.

6.3.5 Τι απαιτείται πριν από τη συμμετοχή στο πρωτόκολλο;

Οι οντότητες που θα επιθυμούσαν να συμμετάσχουν στο πρωτόκολλο οφείλουν πιο πριν να ολοκληρώσουν μια σειρά από ενέργειες που θα τους επιτρέψουν την ομαλή τους ένταξη σε αυτό.

Ο εξυπηρετητής αποτελεί το σημαντικότερο στοιχείο που συμμετέχει στο πρωτόκολλο, οπότε θα ξεκινήσουμε με τις ενέργειες που πρέπει να εκτελεστούν από την πλευρά του.

Η πλευρά του εξυπηρετητή οφείλει:

1. Να εγκαταστήσει όλα αρχεία από τα οποία αποτελείται η διαδικτυακή εφαρμογή.

2. Να πραγματοποιήσει τη σύνδεση με το GSM Modem.

3. Να εκκινήσει μια φορά τη διεργασία του Apache Tomcat για να δημιουργήσει τη βάση δεδομένων που θα χρησιμοποιηθεί από τη διαδικτυακή εφαρμογή.

4. Να δημιουργήσει μιαν επικαιροποιημένη έκδοση του MIDlet, στην οποία θα περιέχονται το δημόσιο κλειδί που θα αντιστοιχεί στον εξυπηρετητή και ο αριθμός του τηλεφώνου που θα χρησιμοποιείται από το GSM modem.

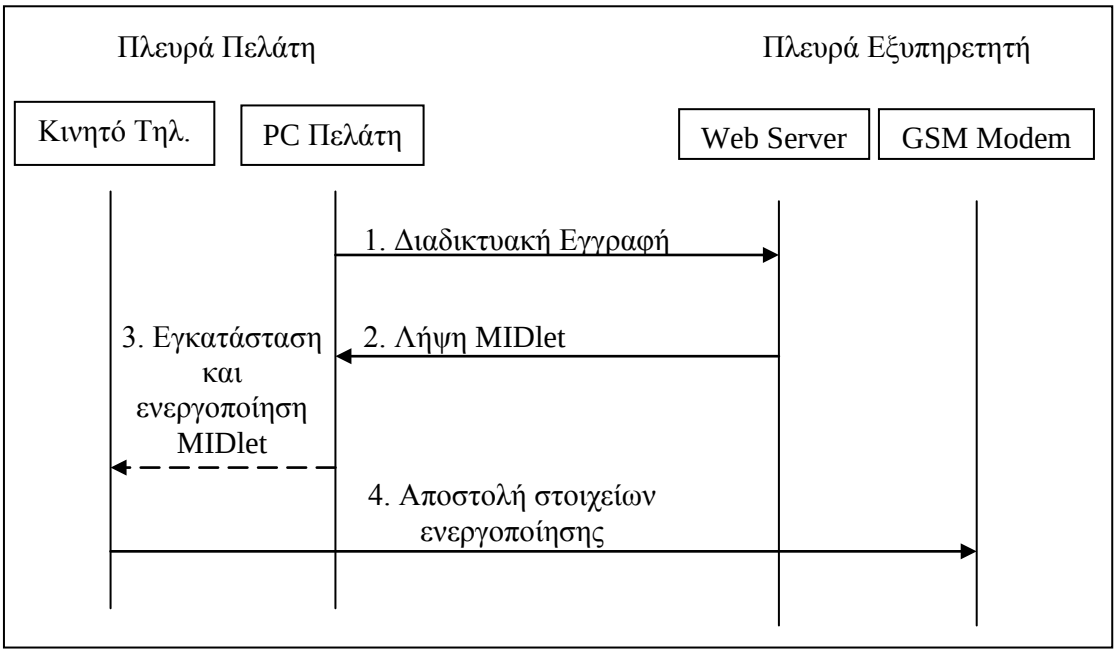
5. Να διαθέσει, μέσα από τη σχετική ιστοσελίδα της εφαρμογής, τα αρχεία που θα αποτελούν το επικαιροποιημένο MIDlet.

6. Να εκκινήσει τη διεργασία του Apache Tomcat, θέτοντας έτσι το πρωτόκολλο σε λειτουργία.

Σε αυτό το σημείο να αναφέρουμε ότι η αναλυτική παρουσίαση των παραπάνω ενεργειών βρίσκεται στο Παράρτημα 1, όπου αναφέρεται το λογισμικό που πρέπει να

εγκαταστήσουμε και οι ρυθμίσεις που απαιτούνται, ούτως ώστε να προσομοιώσουμε τη λειτουργία του εξυπηρετητή σε ένα δικό μας υπολογιστή.

Αφού λοιπόν ολοκληρωθούν επιτυχώς οι παραπάνω ενέργειες η πλευρά του εξυπηρετητή είναι σε θέση να δεχθεί και να επεξεργαστεί όσα αιτήματα έρχονται από τον πελάτη, ο οποίος με τη σειρά του θεωρείται έτοιμος για συμμετοχή στο πρωτόκολλο αφού φέρει εις πέρας τις ενέργειες που εμφανίζονται συνοπτικά στο ακόλουθο σχήμα.



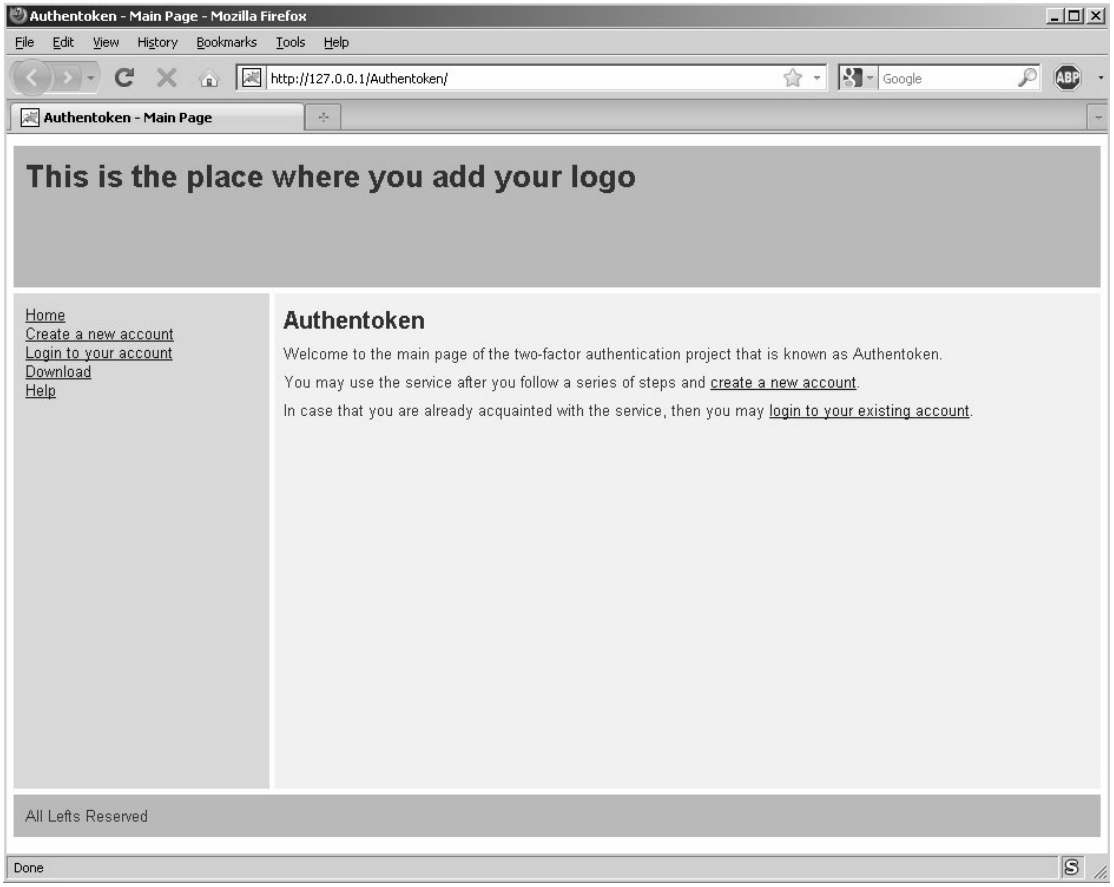
Εικόνα 116: Ενέργειες που πραγματοποιούνται από την πλευρά του πελάτη

Ας δούμε όμως σε αυτό το σημείο πιο αναλυτικά τι συμβαίνει σε κάθε ένα από τα βήματα που ακολουθούνται από τον χρήστη πριν από τη συμμετοχή του στο πρωτόκολλο:

1. Η εγγραφή στη διαδικτυακή υπηρεσία.

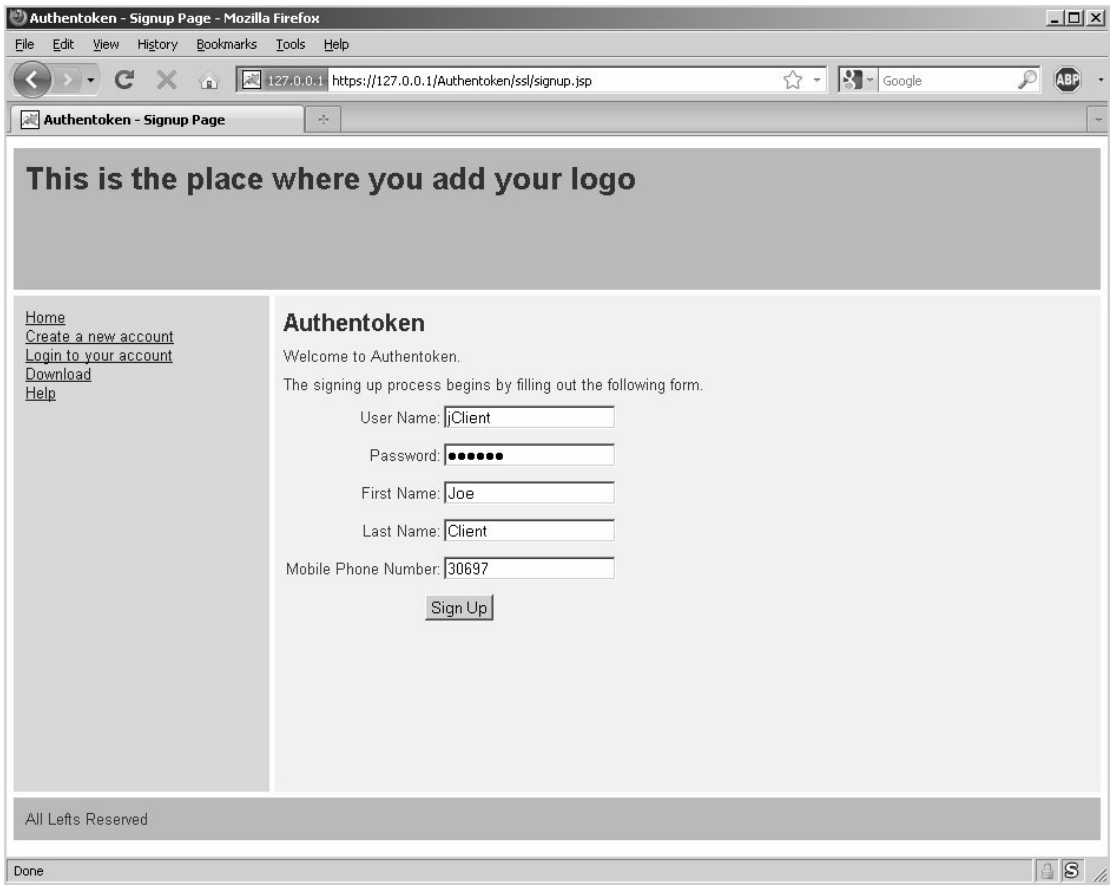
   Ο πελάτης επισκέπτεται το διαδικτυακό τόπο, οπότε και του εμφανίζεται η αρχική σελίδα, στην οποία περιέχονται δύο σύνδεσμοι που θα τον οδηγήσουν στην κατάλληλη σελίδα όπου θα εγγραφεί στη διαδικτυακή υπηρεσία.

   Το ένα link βρίσκεται στο κύριο μέρος της αρχικής σελίδας, ενώ το δεύτερο link βρίσκεται σε μία πλευρική μπάρα που περιέχει όλα τα links και εμφανίζεται μόνιμα στο αριστερό μέρος όλων των σελίδων της διαδικτυακής εφαρμογής.



Εικόνα 117: Η αρχική σελίδα της διαδικτυακής εφαρμογής

Για την εγγραφή του πελάτη στη διαθέσιμη υπηρεσία απαιτείται η συμπλήρωση μιας φόρμας με στοιχεία όπως είναι το username και το password με τα οποία θα συνδέεται στο website, το ονοματεπώνυμο του, καθώς και τον αριθμό που αντιστοιχεί στο κινητό του τηλέφωνο.



Εικόνα 118: Σελίδα εγγραφής στη διαδικτυακή υπηρεσία

Όλα τα δεδομένα του πελάτη υποβάλλονται με πλήρη ασφάλεια, καθώς η συγκεκριμένη διαδικασία πραγματοποιείται με τη βοήθεια του πρωτοκόλλου ασφαλών συνδέσεων TLS. Άλλωστε, το παράθυρο του web browser μας ενημερώνει ότι τα δεδομένα κρυπτογραφούνται όταν στην κάτω δεξιά γωνία του εμφανίζεται το εικονίδιο ενός λουκέτου, ενώ την ίδια στιγμή αλλάζει και ο χρωματισμός που έχει η μπάρα με τη διαδικτυακή διεύθυνση.

Στην τιμή του password που εισάγεται από τον χρήστη στο website εφαρμόζεται η συνάρτηση κατακερματισμού SHA-256. Η σύνοψη που υπολογίζεται είναι αυτή που τελικώς αποθηκεύεται στον server αντί για το password σε clear-text μορφή.

Με αυτό τον τρόπο προφυλάσσεται η τιμή του password ακόμα και στην περίπτωση που κάποιος τρίτος αποκτήσει πρόσβαση στη λίστα με τους κωδικούς των χρηστών που βρίσκονται αποθηκευμένοι στο server.

Η πλευρά του εξυπηρετητή κρίνει αν ο πελάτης έχει εισάγει το σωστό password βασιζόμενη στο γεγονός ότι αν οι δύο τιμές hash ταυτίζονται τότε έχει χρησιμοποιηθεί η ίδια είσοδος στη συνάρτηση κατακερματισμού. Οπότε, κάθε φορά που ο χρήστης θα εισάγει τον κωδικό του μέσω web, τότε θα υπολογίζεται εκ νέου το hash του συγκεκριμένου κωδικού και θα συγκρίνεται με το hash που ήταν ήδη αποθηκευμένο στον server.

Στην εικόνα που ακολουθεί μπορούμε να δούμε τα μηνύματα που εμφανίζονται στην κονσόλα του Apache Tomcat και παράλληλα μπορούμε να δούμε τον τρόπο με τον οποίο η πλευρά του εξυπηρετητή διαχειρίζεται τα δεδομένα που υποβάλλονται κατά την εγγραφή του νέου πελάτη.

```
session signup: null
plain Password: s3cret
pA1 pass2bytes:  733363726574
pA1 hashed pass  8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572
Inserting new Client parameters
================================
```

Εικόνα 119: Η εισαγωγή των στοιχείων ενός νέου πελάτη στη βάση δεδομένων

Αρχικά, ο κωδικός που έχει εισαχθεί από τον πελάτη μετατρέπεται σε πίνακα από bytes, στον οποίο στη συνέχεια εφαρμόζεται η συνάρτηση SHA-256 και έτσι προκύπτει η σύνοψη του κωδικού. Μετά από όλα αυτά δημιουργείται το κατάλληλο ερώτημα SQL με το οποίο όλα τα στοιχεία θα αποθηκευτούν στη βάση δεδομένων.

Σε αυτό το σημείο, για λόγους παρουσίασης και μόνο, μπορούμε να επιβεβαιώσουμε ότι η εγγραφή στη βάση δεδομένων έχει όντως πραγματοποιηθεί. Έτσι, συνδεόμαστε στο MySQL Server και με το κατάλληλο ερώτημα SQL εμφανίζουμε τις εγγραφές που έχουν καταχωρηθεί στον πίνακα που περιέχει τα στοιχεία των πελατών.

```
mysql> select * from clients;
+-----------+------------+-----------+--------------+----------+
| client_id | first_name | last_name | mobile_phone | username |
+-----------+------------+-----------+--------------+----------+
|        17 | Joe        | Client    | 3069         | jClient  |
+-----------+------------+-----------+--------------+----------+
+------------------------------------------------------------------+
| stored_pass                                                      |
+------------------------------------------------------------------+
| 8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572 |
+------------------------------------------------------------------+
+---------+-----------+
| counter | init_vect |
+---------+-----------+
|       0 | 21        |
+---------+-----------+
```

Εικόνα 120: Εμφάνιση της εγγραφής ενός πελάτη στη βάση δεδομένων

Μιας και παρουσιάσαμε τον τρόπο με τον οποίο οργανώνεται η εγγραφή ενός πελάτη στη βάση δεδομένων αξίζει να επισημάνουμε ότι σε αυτή τη φάση τα πεδία counter και init_vect έχουν λάβει τις αρχικές τιμές που τοποθετούνται σε κάθε καινούργια εγγραφή.

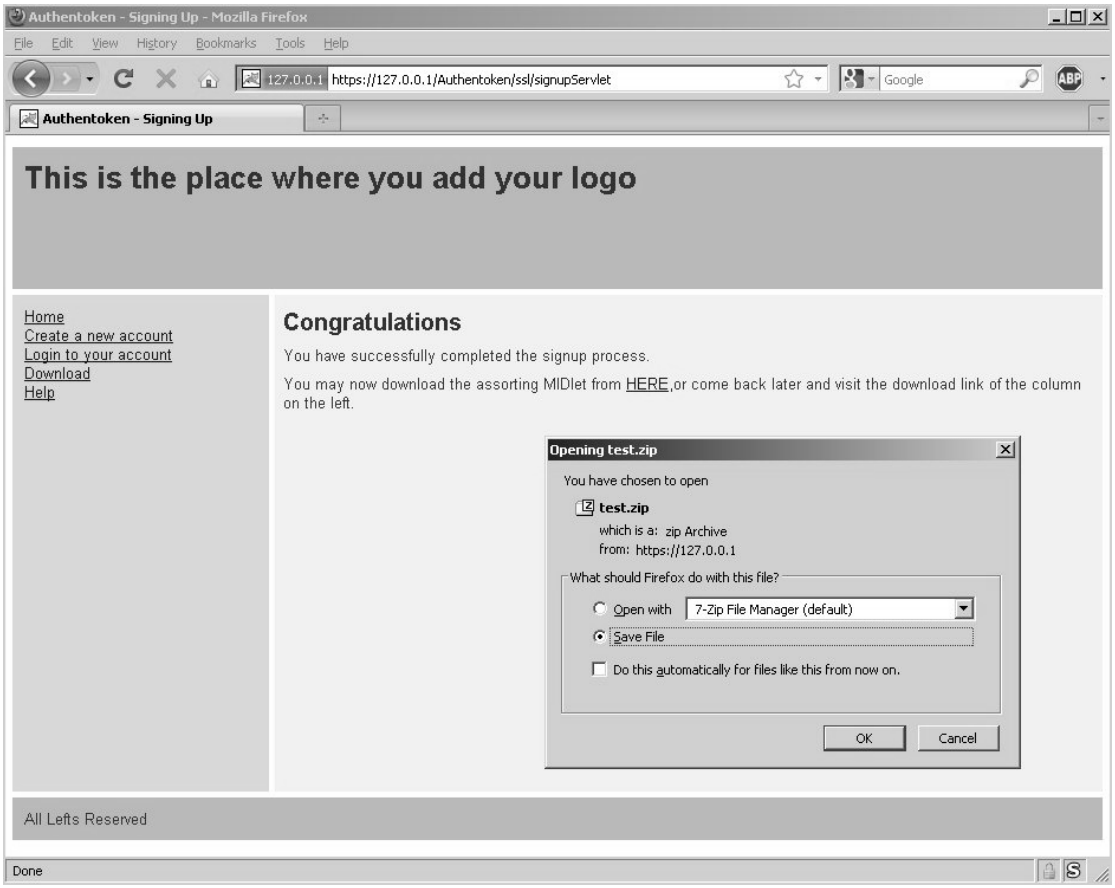
Αργότερα, καθώς ο πελάτης χρησιμοποιεί το πρωτόκολλο, θα δούμε πως μεταβάλλονται οι τιμές των δύο πεδίων και τι ρόλο διαδραματίζουν στη λειτουργία του πρωτοκόλλου.

2. Η διαδικασία λήψης του MIDlet.

Μετά από την εγγραφή, ο πελάτης αποκτά το δικαίωμα να κατεβάσει στον προσωπικό του υπολογιστή την εφαρμογή που θα εγκατασταθεί στο κινητό του τηλέφωνο και θα λειτουργήσει ως σκυτάλη αυθεντικοποίησης.

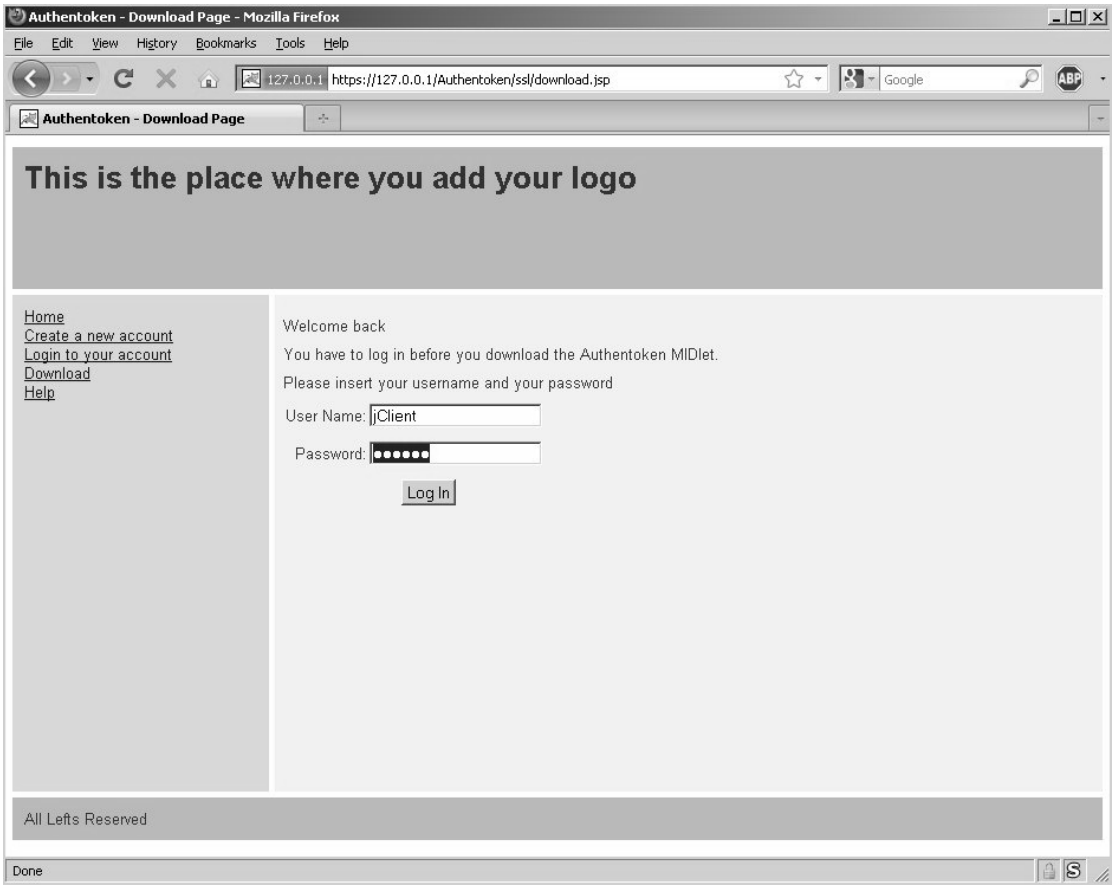
Η λήψη του συγκεκριμένου MIDlet μπορεί να πραγματοποιηθεί αμέσως μετά την εγγραφή του πελάτη, καθώς εκείνος προωθείται στη σελίδα που περιέχει το ανάλογο link.

Βέβαια, στην περίπτωση που ο πελάτης επιθυμεί να κατεβάσει αργότερα το MIDlet, μπορεί να επιλέξει το κατάλληλο link που εμφανίζεται στην πλευρική στήλη με συνδέσμους.



Εικόνα 121: Λήψη του MIDlet αμέσως μετά την εγγραφή του πελάτη

Όταν ένας πελάτης επιθυμεί να λάβει αργότερα το MIDlet, τότε αναγκάζεται να πραγματοποιήσει login και εν συνεχεία αποκτά πρόσβαση σε αυτό.



Εικόνα 122: Login πριν από τη λήψη του MIDlet

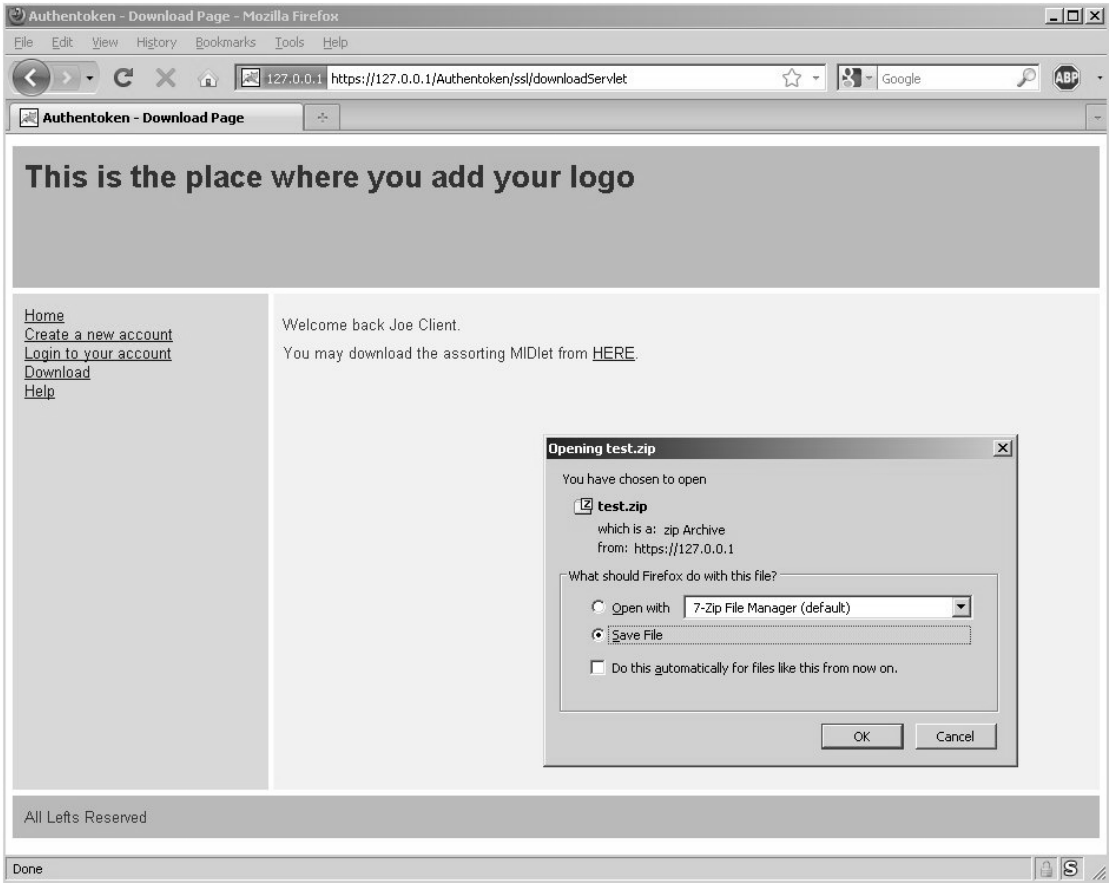
Τα δεδομένα που εισάγει ο πελάτης πριν από τη λήψη του MIDlet ελέγχονται με τον ίδιο τρόπο όπως θα γινόταν σε μια προσπάθεια login στη διαδικτυακή εφαρμογή.

Με άλλα λόγια, από τον κωδικό που εισάγει ο πελάτης δημιουργείται μια τιμή hash και στη συνέχεια ελέγχεται αν στη βάση δεδομένων υπάρχει αποθηκευμένος ο συνδυασμός username και τιμής hash που να ταιριάζει με τα στοιχεία του πελάτη.

```
session login: null
plain Password: s3cret
pA1 pass2bytes:  733363726574
pA1 hashed pass  8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572
Retrieving Client parameters
===========
stored_pass: 8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572
allClear: true
```

Εικόνα 123: Έλεγχος των στοιχείων που εισάγει ο πελάτης πριν από τη λήψη του MIDlet

Αν υπάρχει ταύτιση των στοιχείων, τότε ο πελάτης προωθείται στην κατάλληλη σελίδα απ' όπου μπορεί να λάβει το συμπιεσμένο αρχείο που περιέχει το αναπτυχθέν MIDlet.



Εικόνα 124: Λήψη MIDlet μετά από login

3. Η μεταφορά της J2ME εφαρμογής στο κινητό τηλέφωνο του πελάτη και η ενεργοποίηση αυτής.

   Ο πελάτης χρησιμοποιεί έναν από τους διαθέσιμους τρόπους διασύνδεσης ενός κινητού τηλεφώνου και ενός ηλεκτρονικού υπολογιστή για να μεταφέρει τελικώς το MIDlet στο τηλέφωνο του.

   Κατά την πρώτη εκτέλεση της J2ME εφαρμογής ο πελάτης συναντά μια οθόνη που τον προτρέπει να ακολουθήσει μια διαδικασία ρύθμισης της εν λόγω εφαρμογής.

Εικόνα 125: Οθόνη από την πρώτη εκτέλεση του MIDlet

Στη συνέχεια ο πελάτης καλείται να εισάγει τα στοιχεία που θα χρησιμοποιεί κάθε φορά που θα κάνει login στο κινητό του, πρόκειται για ένα password και μια τιμή salt, δηλαδή τα στοιχεία που απαιτούνται για την εκτέλεση της συνάρτησης PBKDF2.

Εικόνα 126: Οθόνη εισαγωγής των στοιχείων με τα οποία θα γίνεται login στο MIDlet

Μετά την εισαγωγή των στοιχείων παράγεται μια τιμή hash η οποία αποθηκεύεται σε ένα RecordStore που χρησιμοποιείται από το MIDlet.

```
sms thread
sms port ok
No rs found
pAl Derived Key 1f637163c029ded38ef0a0d2691dee038feb959cdbalb548f3a70d24c381bd32
```

Εικόνα 127: Τιμή hash που παράγεται από τα στοιχεία του login

Η τιμή του αποθηκευμένου hash χρησιμοποιείται σε δύο λειτουργίες που πραγματοποιούνται από το MIDlet.

Η μία από αυτές τις λειτουργίες σχετίζεται με τη διαδικασία του login που πραγματοποιείται κάθε φορά που εκκινεί το MIDlet. Πιο συγκεκριμένα, το αποθηκευμένο hash συγκρίνεται με αυτό που παράγεται από τα στοιχεία που εισάγονται σε κάθε login. Αν τα δύο hashes ταυτίζονται τότε είμαστε σίγουροι ότι έχουν εισαχθεί τα σωστά στοιχεία.

Η άλλη λειτουργία του αποθηκευμένου hash αφορά τη δημιουργία του One-Time Password που χρησιμοποιείται από τις δύο οντότητες που συμμετέχουν σε ένα session πιστοποίησης ταυτότητας. Γι' αυτό και είναι απολύτως αναγκαίο το επόμενο βήμα.

4. Η αποστολή των στοιχείων που παράχθηκαν κατά την ενεργοποίηση του MIDlet.

   Αμέσως μετά από την ενεργοποίηση του MIDlet ο χρήστης της εφαρμογής υποχρεώνεται να στείλει με μήνυμα SMS το hash που παράγεται κατά το login στο κινητό του τηλέφωνο. Γι' αυτό το σκοπό, στην οθόνη του κινητού του πελάτη εμφανίζεται η ανάλογη επιλογή.

Εικόνα 128: Οθόνη αποστολής του παραγόμενου hash προς την πλευρά του εξυπηρετητή

Η τιμή του hash που αποστέλλεται στην πλευρά του server κρυπτογραφείται με τον αλγόριθμο RSA, χρησιμοποιώντας το δημόσιο κλειδί που βρίσκεται ήδη εγκατεστημένο στην εφαρμογή και αντιστοιχεί στο ζεύγος των κλειδιών του server.

Με αυτό τον τρόπο, η τιμή του παραγόμενου hash διαφυλάσσεται και γίνεται γνωστή μόνο στον server και όχι σε κάποια τρίτη πλευρά που μπορεί να παρακολουθεί το κανάλι επικοινωνιών.

```
msg sent

pAl Plain Data 31663633373136336330323964656433386566306130643236393164656530333866656239353963646261316235343866336137
pAl Ciphered Data: 81d6382127658c20001728422bb73e2ce329393ad5c9ce8e28977b9628222d49432b553365c5fd9eelf5c9e87fa2ad7416b9
Segments 2
```

Εικόνα 129: Η κρυπτογράφηση του hash που παράχθηκε από το MIDlet

Το κρυπτογραφημένο μήνυμα που αποστέλλεται από τον πελάτη φτάνει στο κινητό τηλέφωνο - GSM Modem που αναλαμβάνει την εισερχόμενη κίνηση, το οποίο λειτουργεί σύμφωνα με το Text Mode που αναφέραμε νωρίτερα κατά την ανάλυση των μηνυμάτων SMS.

Η πλευρά του εξυπηρετητή είναι εκ των προτέρων έτοιμη, οπότε η ανάκτηση και η αποθήκευση του hash πραγματοποιείται με τον εξής τρόπο:

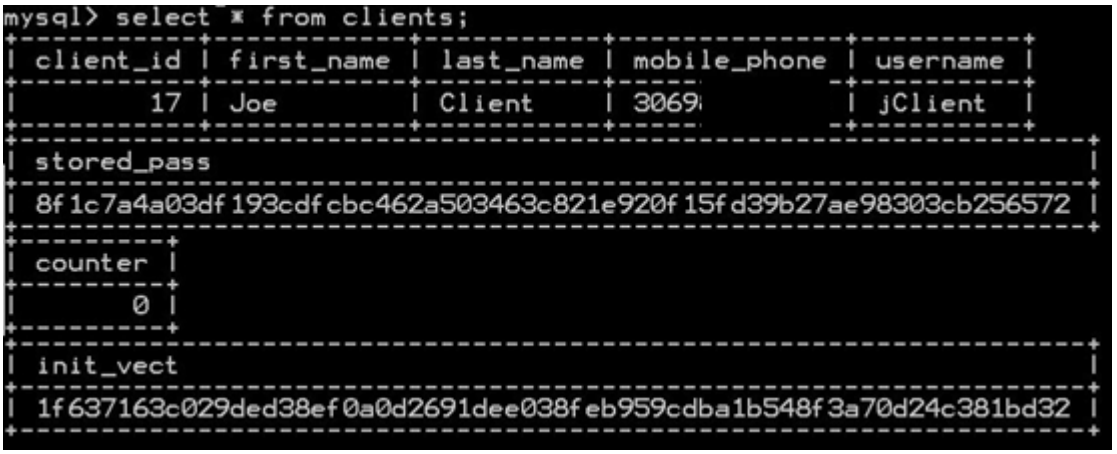
- Το thread που εξετάζει την εισερχόμενη κίνηση στέλνει στο αρμόδιο κινητό τηλέφωνο που λειτουργεί ως GSM Modem την εντολή AT+CMGL=″REC UNREAD″, με την οποία ελέγχει για την ύπαρξη νέων μηνυμάτων. Από την εντολή επιστρέφεται το πλήθος των μη επεξεργασμένων μηνυμάτων μαζί με τους αντίστοιχους αριθμούς αναφοράς.

Εικόνα 130: Εκτέλεση εντολής AT+CMGL για την εμφάνιση νέων μηνυμάτων

- Κάθε ένα μήνυμα διαβάζεται ξεχωριστά μέσω της εντολής AT+CMGR και του αντίστοιχου αριθμού αναφοράς. Από την εκτέλεση κάθε εντολής επιστρέφονται όλα τα δεδομένα που περιέχονται σε κάθε μήνυμα, όμως από αυτά απομονώνουμε τον αριθμό τηλεφώνου του αποστολέα και το κρυπτογραφημένο hash.
- Η πλευρά του εξυπηρετητή έχει στη διάθεση της τα στοιχεία που αποτελούν το ιδιωτικό της κλειδί για τον αλγόριθμο RSA, οπότε αποκρυπτογραφεί το περιεχόμενο κάθε μηνύματος, λαμβάνοντας έτσι την καθαρή τιμή hash που στέλνει κάθε πελάτης.
- Ελέγχεται αν ο αριθμός τηλεφώνου του αποστολέα αντιστοιχεί στα στοιχεία κάποιου πελάτη που βρίσκονται αποθηκευμένα στη βάση δεδομένων. Αν βρεθεί αντιστοιχία, τότε εξετάζεται η τιμή του πεδίου init_vect.
- Στην περίπτωση που το πεδίο init_vect έχει αποθηκευμένη την αρχική του τιμή, που είναι "21", τότε εκτελείται το κατάλληλο ερώτημα SQL με το οποίο πραγματοποιείται η ανανέωση της εν λόγω τιμής με την τιμή hash του πελάτη που μόλις ελήφθη.

Εικόνα 131: Η αποθήκευση της τιμής hash που δημιουργήθηκε στο κινητό τηλέφωνο του πελάτη

Για να επιβεβαιώσουμε την ανανέωση των δεδομένων του πελάτη συνδεόμαστε στο MySQL Server και επιλέγουμε την εμφάνιση των εγγραφών της βάσης. Τότε μπορούμε πράγματι να δούμε ότι στο πεδίο init_vect έχει αποθηκευτεί η τιμή hash, με μήκος 256 δυαδικών ψηφίων, που δημιουργήθηκε νωρίτερα στο κινητό τηλέφωνο του πελάτη.



Εικόνα 132: Εμφάνιση της ανανεωμένης εγγραφής ενός πελάτη στη βάση δεδομένων

6.3.6 Παρουσίαση του πρωτοκόλλου σε λειτουργία

Η διαδικασία με την οποία πιστοποιείται η ταυτότητα ενός πελάτη βασίζεται, όπως αναφέραμε και νωρίτερα, σε ένα συνδυασμό των τεχνικών One-Time Password και Challenge-Response.

Το πρωτόκολλο τίθεται σε λειτουργία όταν από την πλευρά του πελάτη έχει πραγματοποιηθεί ένα επιτυχημένο login στο διαδικτυακό τόπο που παρέχεται από τον server.

Τότε στην πλευρά του εξυπηρετητή δημιουργείται ένα Challenge, το οποίο προωθείται με ένα μήνυμα SMS στο κινητό τηλέφωνο του πελάτη.

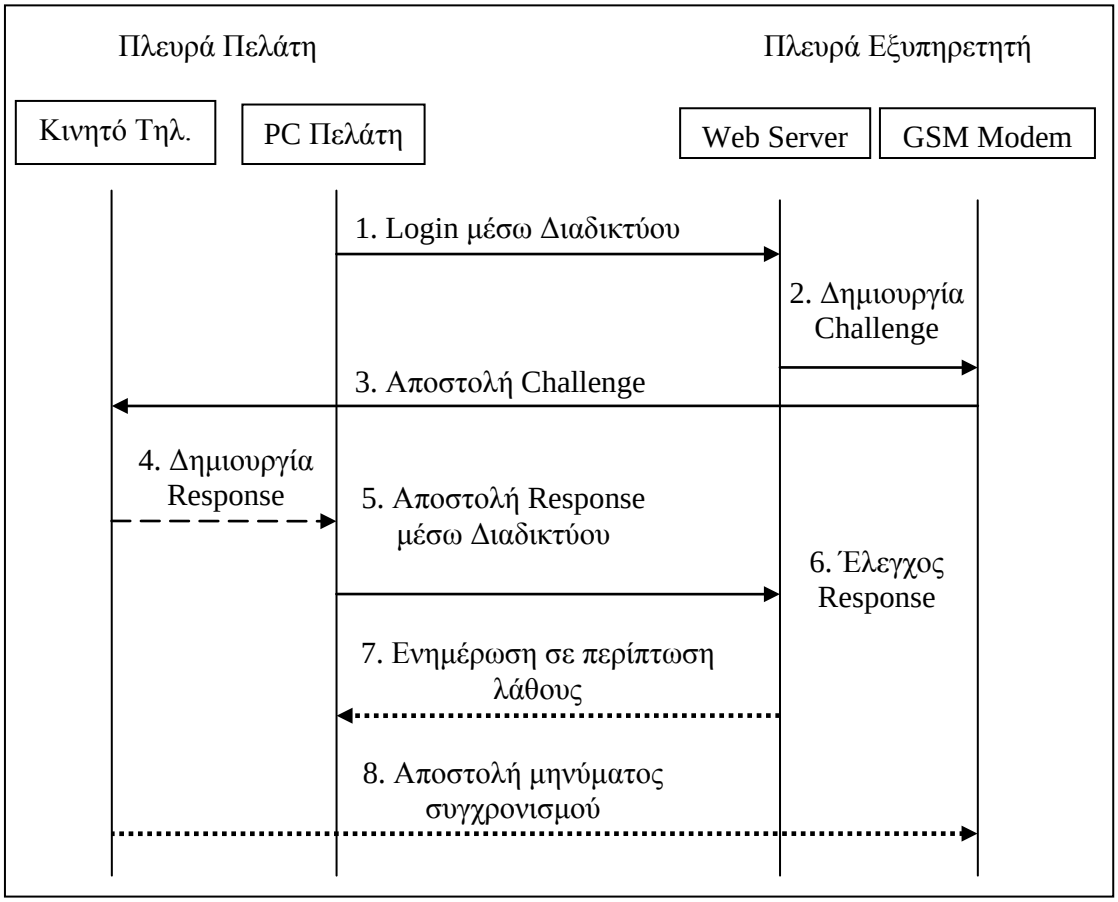
Στη συνέχεια οι δύο πλευρές δημιουργούν ένα One-Time Password, το οποίο χρησιμοποιούν κατά την εκτέλεση μιας συνάρτησης πάνω στην τιμή του Challenge.

Το Response του πελάτη γνωστοποιείται με κάποιο τρόπο στον εξυπηρετητή, όπως θα γινόταν με τη συμπλήρωση ενός πεδίου που συναντάται σε μια διαδικτυακή φόρμα. Με αυτό τον τρόπο ο εξυπηρετητής είναι σε θέση να συγκρίνει την απάντηση του πελάτη με το Response που έχει υπολογιστεί από την δική του πλευρά.

Αν τα δύο Responses είναι ίσα, τότε θεωρείται ότι ο πελάτης είναι πράγματι αυτός που ισχυρίζεται ότι είναι, εκπληρώνοντας έτσι ένα από τα κριτήρια με τα οποία αποφασίζεται αν θα επιτραπεί η είσοδος του σε ένα περιβάλλον με περιορισμένη πρόσβαση.

Όμως πριν να περάσουμε σε περισσότερο τεχνικές λεπτομέρειες θα πρέπει να δούμε τις ενέργειες που πραγματοποιούνται, τόσο στην πλευρά του εξυπηρετητή όσο και σε εκείνη του πελάτη.

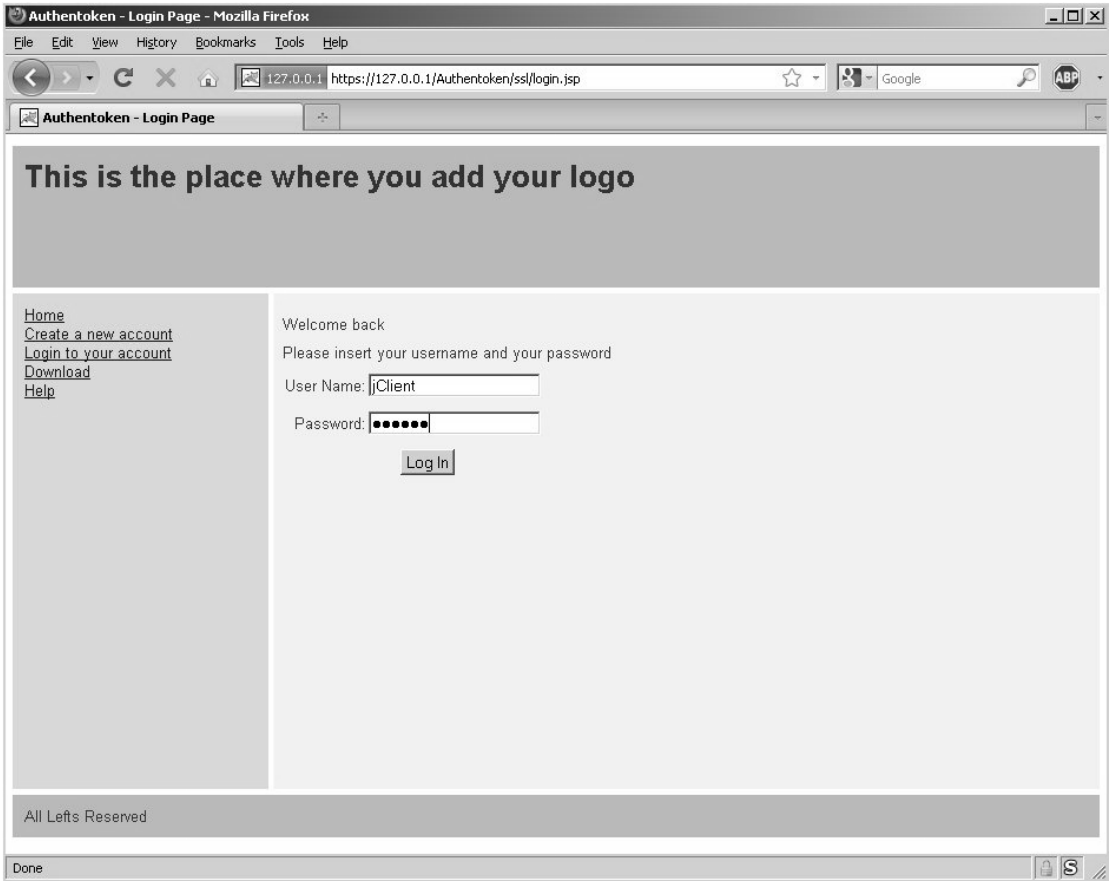
Στο σχήμα που ακολουθεί παρουσιάζονται συνοπτικά οι ενέργειες που πραγματοποιούνται από τις δύο πλευρές που συμμετέχουν στο πρωτόκολλο πιστοποίησης ταυτότητας.



Εικόνα 133: Οι ενέργειες που περιλαμβάνονται στο πρωτόκολλο πιστοποίησης ταυτότητας

1. *Πραγματοποιείται login μέσω του διαδικτύου.*

   Ο πελάτης επισκέπτεται την αρχική σελίδα της διαδικτυακής εφαρμογής και από εκεί επιλέγει ένα από τα links για το login, έτσι προωθείται στο κατάλληλο web interface που παρέχεται από τον server και εισάγει τα στοιχεία(username, password) με τα οποία έχει εγγραφεί στην παρεχόμενη υπηρεσία.



Εικόνα 134: Η σελίδα όπου πραγματοποιείται το login στη διαδικτυακή υπηρεσία

Κατά τη διαδικασία του login ξεκινά ένα νέο session ασφαλούς σύνδεσης που βασίζεται στο πρωτόκολλο TLS, έτσι, στο παράθυρο του web browser εμφανίζεται το διακριτικό λουκέτο που υποδεικνύει ότι τα δεδομένα που υποβάλλονται από τη φόρμα του login αποστέλλονται κρυπτογραφημένα.

Την ίδια στιγμή, ο πελάτης μπορεί να κάνει login στην εφαρμογή που έχει εγκατεστημένη στο κινητό του τηλέφωνο και να περιμένει για το Challenge που θα σταλεί από την πλευρά του εξυπηρετητή.

Εικόνα 135: Οθόνη για Login στο MIDlet που βρίσκεται στο κινητό τηλέφωνο του πελάτη

Το login του πελάτη στο MIDlet πραγματοποιείται μετά από σύγκριση δύο τιμών hash. Η μία εξ αυτών, που είναι ήδη αποθηκευμένη στο κινητό τηλέφωνο, έχει δημιουργηθεί από τα στοιχεία που εισήγαγε ο πελάτης κατά την εγκατάσταση του MIDlet, ενώ η άλλη τιμή δημιουργείται μετά από το username και το password που εισάγει ο πελάτης κατά το τρέχον login.

Αν τα δύο συγκρινόμενα hashes ταυτίζονται, τότε είμαστε σίγουροι ότι ο πελάτης έχει εισάγει τα ίδια στοιχεία με αυτά που εισήχθησαν κατά την ενεργοποίηση του MIDlet, οπότε του επιτρέπεται η πρόσβαση σε αυτό.

Στην εικόνα που ακολουθεί παρουσιάζεται ότι εκτυπώνεται στην κονσόλα του IDE NetBeans κατά την εκτέλεση του MIDlet, και ειδικότερα κατά τη φάση του login, όπου παράγεται η τιμή hash από τα στοιχεία που εισάγονται από τον πελάτη και στη συνέχεια συγκρίνεται με το hash που είναι αποθηκευμένο σε ένα RecordStore που σχετίζεται με το συγκεκριμένο MIDlet.

```
sms thread
sms port ok
loginNFO
pAl Derived Key 1f637163c029ded38ef0a0d2691dee038feb959cdbalb548f3a70d24c38lbd32
Length: 70
1f637163c029ded38ef0a0d2691dee038feb959cdbalb548f3a70d24c38lbd32
0
```

Εικόνα 136: Περιεχόμενα της κονσόλας του NetBeans κατά το login στο MIDlet

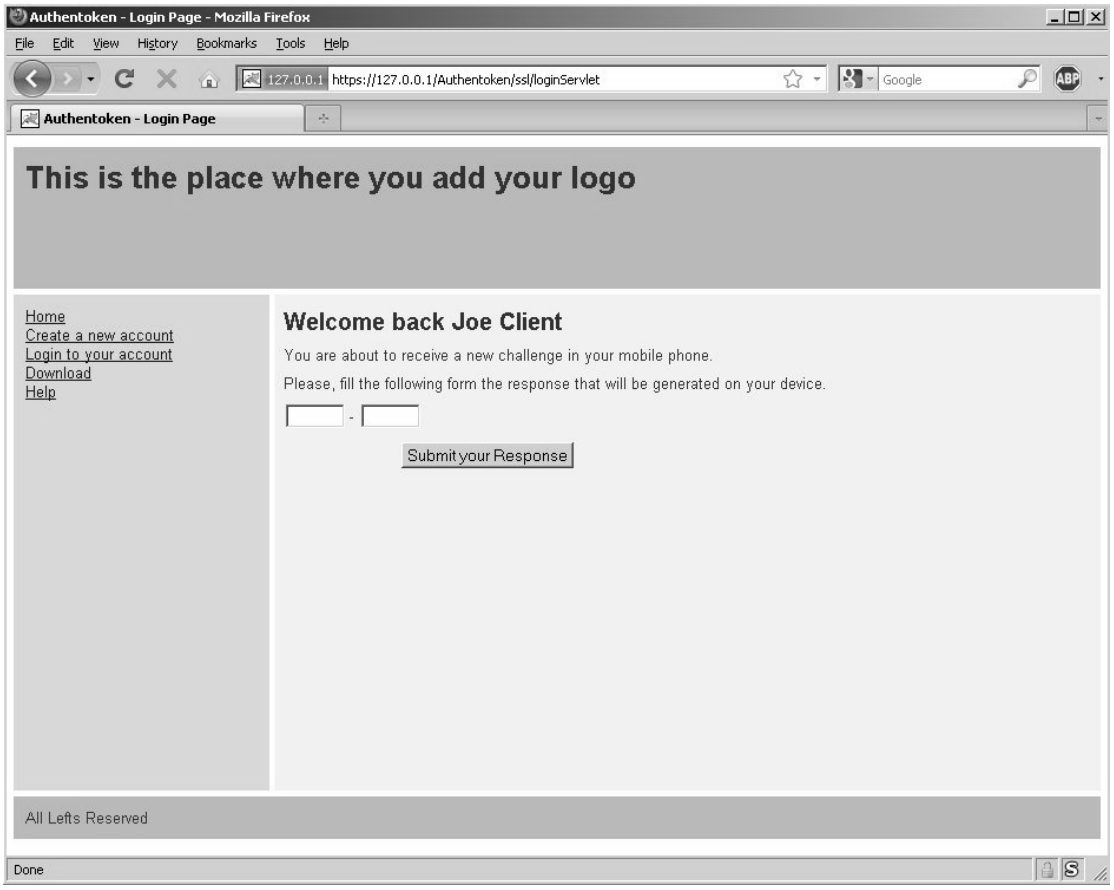
Εν τω μεταξύ, στην πλευρά του εξυπηρετητή, εφαρμόζεται η συνάρτηση κατακερματισμού SHA-256 στο password που εισήχθη από τον πελάτη και ακολούθως ελέγχεται αν υπάρχει καταχωρημένος κάποιος πελάτης με το username που μόλις χρησιμοποιήθηκε. Αφού επιβεβαιωθεί ότι πράγματι υπάρχει το συγκεκριμένο username στη βάση δεδομένων, τότε το παραγόμενο hash συγκρίνεται με την τιμή που είχε αποθηκευτεί νωρίτερα κατά το sign-up του πελάτη στη διαδικτυακή υπηρεσία.

Αν τα δύο hashes είναι ίσα, τότε ο πελάτης έχει αποδείξει ότι πληροί τον ένα από τους δύο χρησιμοποιούμενους παράγοντες αυθεντικοποίησης, αφού γνωρίζει τη σωστή τιμή του κωδικού για το login.

```
plain Password: s3cret
pA1 pass2bytes:  733363726574
pA1 hashed pass  8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572
Retrieving Client parameters
==========
stored_pass: 8f1c7a4a03df193cdfcbc462a503463c821e920f15fd39b27ae98303cb256572
allClear: true
```

Εικόνα 137: Επαλήθευση του username και του password που χρησιμοποιήθηκαν σε ένα login

Έτσι, σε αυτή την περίπτωση, ο χρήστης περνά από τη σελίδα του login σε μια άλλη σελίδα, όπως φαίνεται και στην εικόνα που ακολουθεί, όπου θα επιβεβαιώσει ότι έχει στην κατοχή του και τη σκυτάλη ασφάλειας που χρησιμοποιεί.



Εικόνα 138: Σελίδα που εμφανίζεται έπειτα από επιτυχημένο login

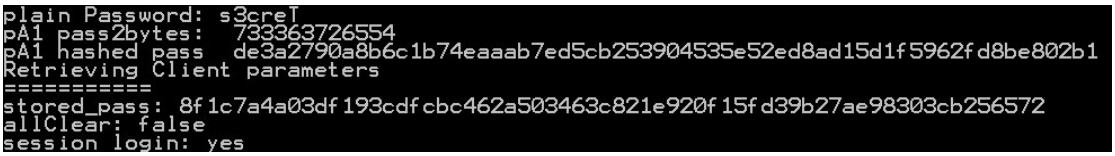
Για την πραγματοποίηση αυτής της ενέργειας απαιτείται η ανάκτηση επιπρόσθετων δεδομένων που αφορούν τον πελάτη, όπως είναι η τιμή του hash που στάλθηκε από το κινητό του τηλέφωνο και η τιμή του μετρητή που τρέχει παράλληλα ανάμεσα στη συγκεκριμένη συσκευή και στην πλευρά του εξυπηρετητή.

Βέβαια, για λόγους καλύτερης παρουσίασης και μόνο, ανακτούμε και τυπώνουμε στην κονσόλα του Apache Tomcat και άλλα στοιχεία που αντιστοιχούν στον πελάτη που μόλις επιχείρησε να κάνει login μέσω του web interface. Ανάμεσα σε αυτά περιλαμβάνονται:

- Το όνομα και το επώνυμο του πελάτη
- Ο αριθμός του κινητού του τηλεφώνου
- Η τιμή του μετρητή που παρακολουθείται από τον εξυπηρετητή
- Η τιμή hash που στάλθηκε από το κινητό τηλέφωνο του πελάτη

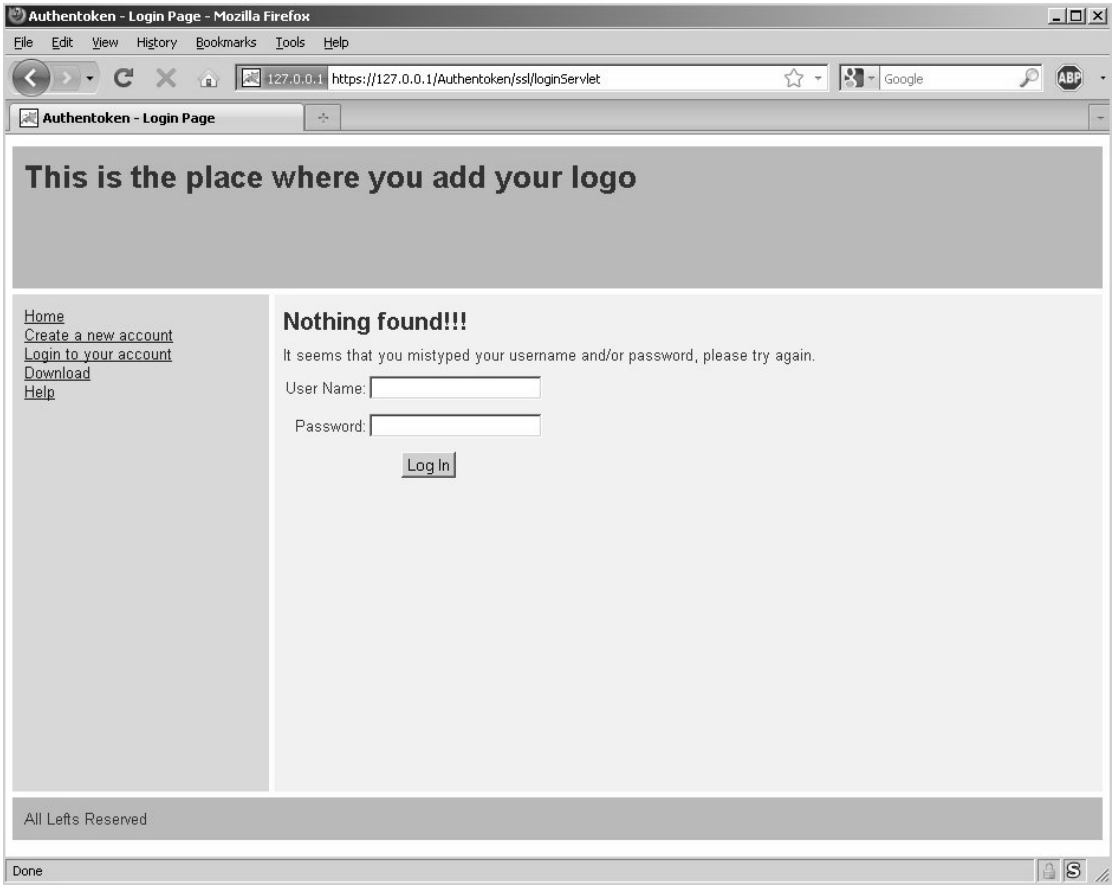
Εικόνα 139: Εκτύπωση των στοιχείων του πελάτη στην κονσόλα του Apache Tomcat

Αν όμως η σύγκριση των δύο hashes αποδείξει ότι αυτά διαφέρουν ή το username που εισήχθη δεν είναι καταχωρημένο στη βάση δεδομένων, τότε στην κονσόλα του Apache Tomcat που φιλοξενείται από την πλευρά του εξυπηρετητή θα εμφανιστεί το κατάλληλο μήνυμα που θα υποδηλώνει τη λανθασμένη εισαγωγή δεδομένων από την πλευρά του πελάτη.



Εικόνα 140: Επιβεβαίωση της εισαγωγής λανθασμένων στοιχείων από την πλευρά του πελάτη

Σε αυτή την περίπτωση, ο πελάτης θα ειδοποιηθεί ότι υπήρξε κάποιο λάθος κατά τη διαδικασία του login και θα προωθηθεί στην ανάλογη σελίδα όπου θα καλείται να επαναλάβει τη συγκεκριμένη διαδικασία.



Εικόνα 141: Η σελίδα που εμφανίζεται στην περίπτωση ενός αποτυχημένου login

2. *Δημιουργία του Challenge.*

Όπως αναφέραμε και νωρίτερα, ο έλεγχος του δεύτερου παράγοντα αυθεντικοποίησης βασίζεται στην τεχνική του Challenge – Response.

Γι' αυτό το σκοπό, δημιουργείται στην πλευρά του εξυπηρετητή το κατάλληλο Challenge, με την τιμή του να προκύπτει από τη συνάρτηση που επιστρέφει το συνολικό αριθμό των milliseconds που έχουν περάσει από την $1^{η}$ Ιανουαρίου του 1970.

Η τιμή του Challenge πρέπει να φτάσει στο κινητό τηλέφωνο που συναντάμε στην πλευρά του πελάτη, αυτό είναι το σημείο όπου αναλαμβάνει δράση η εφαρμογή που λειτουργεί σαν γέφυρα επικοινωνίας ανάμεσα στην πλευρά του εξυπηρετητή και το GSM Modem που έχει την ευθύνη για τη δρομολόγηση της εξωτερικής κίνησης.

Το συγκεκριμένο GSM Modem λειτουργεί σύμφωνα με το PDU Mode, το οποίο και αναφέραμε νωρίτερα κατά την παρουσίαση των μηνυμάτων SMS, γι' αυτό και η εφαρμογή που αναλαμβάνει το σχηματισμό ενός νέου μηνύματος SMS θα πρέπει να το κωδικοποιήσει με τον κατάλληλο τρόπο ώστε αυτό να εναρμονίζεται με το PDU Mode.

Στο σχηματισμό του νέου μηνύματος, που δημιουργείται σύμφωνα με το PDU Mode, συμμετέχουν τα ακόλουθα στοιχεία:

- Ο αριθμός τηλεφώνου του πελάτη, ο οποίος κωδικοποιείται σύμφωνα με όσα ορίζονται από το PDU Mode, δηλαδή χωρίζεται ανά ζεύγη αριθμών τα οποία αντιστρέφονται.
- Η τιμή των milliseconds που αντιπροσωπεύουν το Challenge, που στην ουσία αποτελεί το payload του μηνύματος μας, οπότε συμπιέζεται σύμφωνα με το σχήμα συμπίεσης GSM-7 και τοποθετείται στο πεδίο User Data.
- Ο αριθμός του port όπου έχει προσδεθεί ο listener νέων μηνυμάτων που σχετίζεται με το MIDlet που είναι εγκατεστημένο στο κινητό τηλέφωνο του πελάτη. Το συγκεκριμένο στοιχείο αφού μετατραπεί σε δεκαεξαδική μορφή, όπου στην περίπτωση μας ο αριθμός του port είναι "`16666`" και μετατρέπεται σε "`411A`", επισυνάπτεται στο πεδίο User Data ως επικεφαλίδα User Data Header.

Το τελευταίο στοιχείο έχει ιδιαίτερη σημασία για τη λειτουργία του πρωτοκόλλου απομακρυσμένης πιστοποίησης ταυτότητας, αφού χωρίς την παρουσία του port με το οποίο σχετίζεται το MIDlet θα είχαμε την ουσιαστική αχρήστευση όλων των τηλεφωνικών συσκευών που θα λειτουργούσαν ως σκυτάλες αυθεντικοποίησης.

Αυτό θα γινόταν διότι τα μηνύματα SMS δεν θα δρομολογούνταν προς το αρμόδιο MIDlet, αλλά προς τον εξ ορισμού φάκελο με τα εισερχόμενα μηνύματα, έχοντας έτσι ελάχιστη σημασία για τον ίδιο τον πελάτη.

3. *Αποστολή του Challenge.*

Αφού ολοκληρωθεί η δημιουργία του νέου μηνύματος τότε αυτό αποστέλλεται με την κατάλληλη εντολή AT, όπου στην περίπτωση μας είναι η εντολή "`AT+CMGS`", προς το κινητό τηλέφωνο του πελάτη.

```
millis 1269209815328
pA1 converted number:  0396
pA1 compressed msg:  31992d2783e570b1da4c8603
Num length: 6 MSG length 12
pA1 challenge:  0041000c91039628998544000015060504411a000031992d2783e570b1da4c8603
length 32
writing string with carriage return to port: AT+CMGS=32

reading from port: AT+CMGS=32
>

writing string to port: 0041000c91039628998544000015060504411a000031992d2783e570b1da4c8603→

reading from port: 0041000c91039628998544000015060504411a000031992d2783e570b1da4c8603→
```

Εικόνα 142: Η κονσόλα του Apache Tomcat κατά τη δημιουργία και την αποστολή ενός SMS που περιέχει ένα Challenge

4. *Δημιουργία του Response που θα στείλει ο χρήστης.*

Το MIDlet που εκτελείται στο κινητό του πελάτη τον ειδοποιεί για το νέο μήνυμα που μόλις έχει λάβει.

Εικόνα 143: Οθόνη ειδοποίησης για τη λήψη ενός νέου μηνύματος

Μετά από αυτό το γεγονός, ξεκινά η δημιουργία του One-Time Password που πρόκειται να χρησιμοποιηθεί κατά την τρέχουσα αυθεντικοποίηση.

Η τιμή του μετρητή που βρίσκεται αποθηκευμένη στο κινητό τηλέφωνο του πελάτη αυξάνεται κατά μία μονάδα και εισάγεται ως salt στη συνάρτηση παραγωγής κλειδιών PBKDF2, η οποία θυμίζουμε ότι λειτουργεί σε συνεργασία με τη συνάρτηση κατακερματισμού SHA-256.

Σαν το password που απαιτείται από τη συνάρτηση PBKDF2 χρησιμοποιείται η τιμή hash που βρίσκεται και αυτή αποθηκευμένη στο κινητό τηλέφωνο του πελάτη.

Από την εκτέλεση της συνάρτησης PBKDF2 παράγεται μια σύνοψη των 256 δυαδικών ψηφίων, η οποία αντιπροσωπεύει το One-Time Password που πρόκειται να χρησιμοποιηθεί κατά την τρέχουσα εκτέλεση του πρωτοκόλλου απομακρυσμένης πιστοποίησης ταυτότητας.

Στη συνέχεια, το Challenge που μόλις έχει ληφθεί από την πλευρά του πελάτη εισάγεται στον αλγόριθμο κρυπτογράφησης AES-256, ενώ ως κλειδί κρυπτογράφησης χρησιμοποιείται το One-Time Password που έχει προκύψει από την εκτέλεση της συνάρτησης PBKDF2.

Από την εκτέλεση του αλγορίθμου AES-256 προκύπτει το Response, το οποίο όμως έχει μήκος 32 χαρακτήρων, γεγονός που καθιστά επίπονη την πληκτρολόγηση από την πλευρά του πελάτη, ενώ παράλληλα υπάρχουν πολλές πιθανότητες να γίνει κάποιο λάθος κατά την εισαγωγή του Response στο κατάλληλο web interface.

Έτσι, είναι προτιμότερη η συμπίεση του Response σε μια απλούστερη μορφή που θα διευκολύνει την πληκτρολόγηση από τον πελάτη ενώ παράλληλα θα επιτρέπει τον ορθό έλεγχο από τον εξυπηρετητή.

```
Received message: 1269209815328 from tel: 5550001
sms thread
userNF0: 1f637163c029ded38ef0a0d2691dee038feb959cdba1b548f3a70d24c381bd32
stored count: 0
incremented count: 1
pA1 Derived Key d241683d769ecc76182fabe0578e039b549193c483e2659bfd7d40c4762e0147
Plain 1269209815328
Mhkos 13
Plain Hex 31323639323039383135333238
pA1 Key  :  d241683d769ecc76182fabe0578e039b549193c483e2659bfd7d40c4762e0147
Times=0 Pad=26
Padded Plain Hex 31323639323039383135333238202020
Mhkos=32
pA1 Plain:  31323639323039383135333238202020
pA2 Start round 1: e3 73 5e 04 44 ae f5 4e 29 1a 98 d2 6f ae 23 bb
pA2 Start round 2: ed 73 cf 9a 9c 3e 9d cd ec cc c0 d2 e4 07 35 bc
pA2 Start round 3: 5a a7 ef 38 c0 f2 75 bc 68 eb 77 f4 68 39 32 13
pA2 Start round 4: 82 28 43 ab 05 15 7b 7e 95 07 7a 63 a9 a9 c4 87
pA2 Start round 5: bb 09 da 44 d3 d5 32 0f 70 af 6e 76 43 c4 b2 a2
pA2 Start round 6: 3e 1e fc 21 54 a0 45 51 0b e9 47 7c 65 1d 0d b8
pA2 Start round 7: 5d 99 a7 b2 b2 8d 8f 00 99 89 29 a4 53 07 67 ab
pA2 Start round 8: 14 ec 15 0d bd 3b 32 73 93 0d e2 38 a4 5b f2 a9
pA2 Start round 9: d1 69 39 0d 7a ba ba f2 d5 03 14 f9 35 69 b5 9b
pA2 Start round 10: 4d e6 15 8b 6b 4f 94 e7 f9 70 6c 20 48 69 d2 c8
pA2 Start round 11: 6e 7f 63 89 ab 91 e3 20 31 73 47 ff b7 3d 15 50
pA2 Start round 12: e0 52 38 00 55 87 c3 91 21 a1 93 88 b0 32 c1 31
pA2 Start round 13: 3b da 32 12 40 6b 55 d8 29 57 b4 b6 1b 8c 5f 67
pA2 Start round 14: ce a8 ed 78 84 2d 1c 14 53 f1 f3 10 71 c9 bc 82
pA1 Ciphertext:  e5e8f7b5df89515146919d802c53e758
Ciphered: e5e8f7b5df89515146919d802c53e758
Resp: eb9169ce
```

Εικόνα 144: Η δημιουργία του Response όπως εμφανίζεται στην κονσόλα του IDE NetBeans

Η συγκεκριμένη διαδικασία επιτυγχάνεται με τον ακόλουθο τρόπο:

- Το Response χωρίζεται σε 8 ομάδες των 4 χαρακτήρων.
- Απομονώνονται τα τελευταία 8 ψηφία του Challenge.
- Αντιστρέφεται η σειρά των 8 ψηφίων του Challenge.
- Κάθε ψηφίο αντιστοιχεί σε μια από τις τετράδες χαρακτήρων, δηλαδή το πρώτο ψηφίο αντιστοιχεί στην πρώτη τετράδα, το δεύτερο στη δεύτερη και ούτω καθ' εξής.
- Από κάθε μια από τις τετράδες λαμβάνεται ο χαρακτήρας που βρίσκεται στη θέση που δίνεται από την πράξη (Ψηφίο mod 4). Για παράδειγμα, αν το πέμπτο ψηφίο έχει την τιμή 6, τότε από την πέμπτη τετράδα λαμβάνεται ο χαρακτήρας που βρίσκεται στη θέση 6 mod 4 = 2, όμως με την αρίθμηση να ξεκινά από το μηδέν, η παραπάνω τιμή υποδηλώνει ότι θα ληφθεί ο τρίτος χαρακτήρας από την πέμπτη τετράδα.
- Το τελευταίο βήμα εκτελείται 8 φορές, σχηματίζοντας έτσι ένα νέο Response με μήκος 8 χαρακτήρων.

Για να γίνει περισσότερο κατανοητή η παραπάνω διαδικασία θα παρουσιάσουμε τον τρόπο με τον οποίο δημιουργείται το Response της περίπτωσης που εμφανίζεται και στα screenshots που περιλαμβάνονται στο τρέχον υποκεφάλαιο. Έτσι λοιπόν, έχουμε τις ακόλουθες τιμές:

Challenge: 1269209815328

Αρχικό Response: e5e8f7b5df89515146919d802c53e758

Τετράδες Αρχικού Response	e5e8	f7b5	df89	5151	4691	9d80	2c53	e758
Τελευταία 8 ψηφία του Challenge	0	9	8	1	5	3	2	8
Αντίστροφη σειρά των ψηφίων του Challenge	8	2	3	5	1	8	9	0
Ψηφίο mod 4	8 mod 4	2 mod 4	3 mod 4	5 mod 4	1 mod 4	8 mod 4	9 mod 4	0 mod 4
Χαρακτήρας που λαμβάνεται	0	2	3	1	1	0	1	0
Τελικό Response	e	b	9	1	6	9	c	e

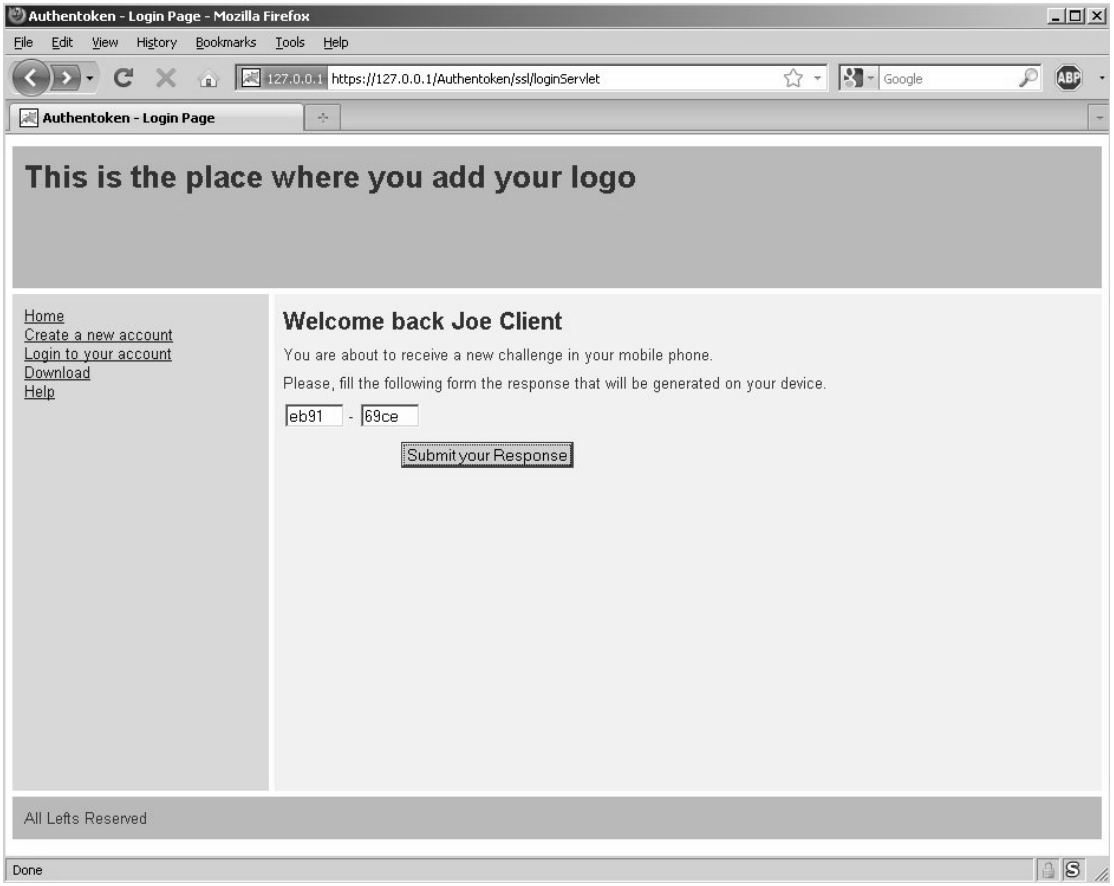
Πίνακας 13: Η δημιουργία του συμπιεσμένου Response

Το συμπιεσμένο Response που προκύπτει έχει μήκος 8 χαρακτήρων και εμφανίζεται στην οθόνη του κινητού τηλεφώνου του πελάτη χωρισμένο σε δύο τετράδες χαρακτήρων.

Εικόνα 145: Εμφάνιση του συμπιεσμένου Response με μήκος 8 χαρακτήρων στην οθόνη του κινητού τηλεφώνου του πελάτη

5. *Αποστολή του Response μέσω του Διαδικτύου*

   Στη συνέχεια ο πελάτης καλείται να πληκτρολογήσει τους 8 χαρακτήρες στη φόρμα που εμφανίζεται μετά από ένα επιτυχημένο login.



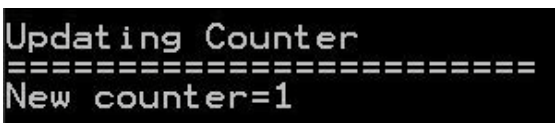
Εικόνα 146: Η ιστοσελίδα που εμφανίζεται στην πλευρά ενός πελάτη που έχει περάσει με επιτυχία το στάδιο του login

6. *Έλεγχος του ληφθέντος Response.*

   Η πλευρά του εξυπηρετητή λαμβάνει το Response που πληκτρολογήθηκε από την πλευρά του πελάτη και ξεκινά τη διαδικασία με την οποία ελέγχει αν η συγκεκριμένη τιμή είναι σωστή.

   Ο εξυπηρετητής πρέπει να δημιουργήσει και αυτός με τη σειρά του το One-Time Password που δημιουργήθηκε από την πλευρά του χρήστη.

   Γι' αυτό, αυξάνει κατά μία μονάδα τον μετρητή που διατηρεί για το συγκεκριμένο πελάτη και τον εισάγει μαζί με το αντίστοιχο αποθηκευμένο hash στη συνάρτηση PBKDF2.



Εικόνα 147: Αύξηση του μετρητή που διατηρείται από την πλευρά του εξυπηρετητή κατά μια μονάδα

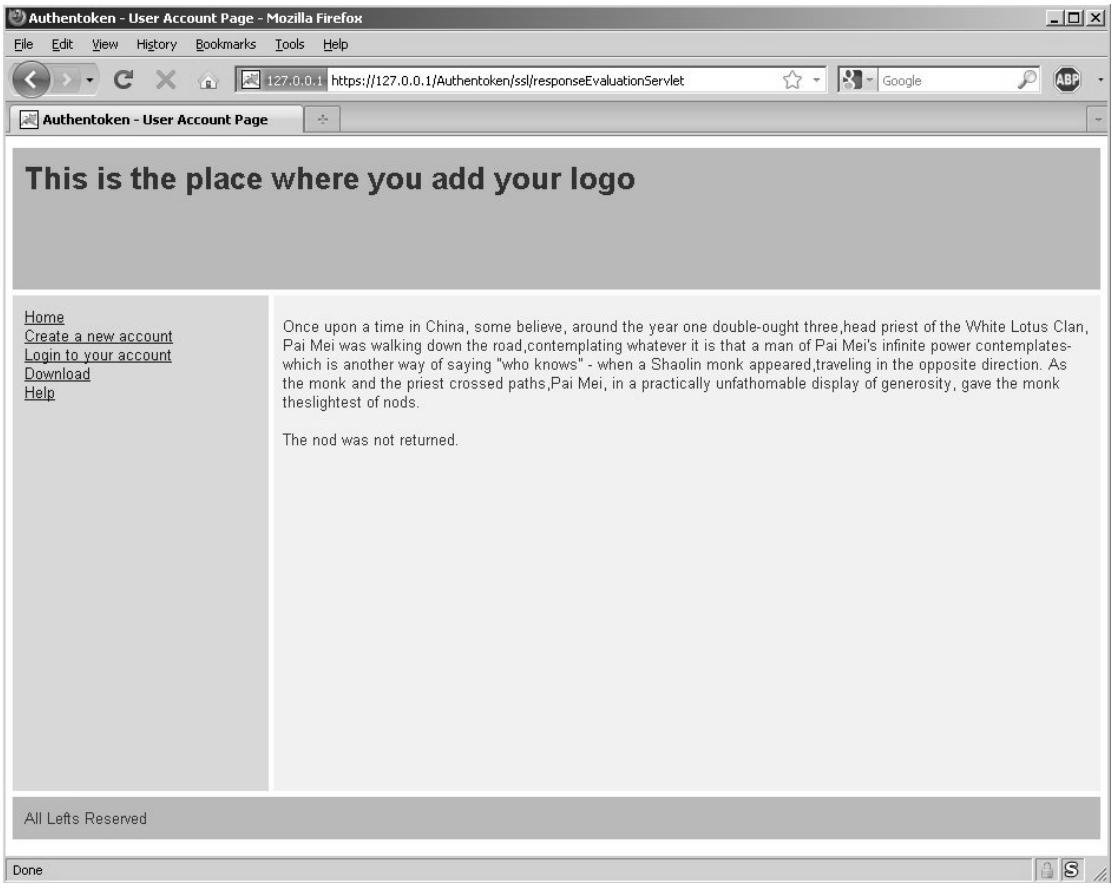
Η σύνοψη που προκύπτει από την εκτέλεση της συνάρτησης είναι, όπως έγινε και προηγουμένως στην πλευρά του πελάτη, το One-Time Password που πρόκειται να χρησιμοποιηθεί ως το κλειδί για την κρυπτογράφηση του Challenge μέσω του αλγορίθμου AES-256.

Με αυτό τον τρόπο δημιουργείται το αντίστοιχο Response από την πλευρά του εξυπηρετητή και στη συνέχεια συγκρίνεται με την τιμή του Response που ήρθε από την πλευρά του πελάτη.

Εικόνα 148: Η κονσόλα του Apache Tomcat κατά τη δημιουργία του Response από την πλευρά του εξυπηρετητή και τη σύγκριση με το Response που έχει υποβάλλει η πλευρά του πελάτη

Αν οι δύο απαντήσεις είναι ίδιες, τότε ο πελάτης έχει αποδείξει ότι πράγματι έχει στην κατοχή του την προβλεπόμενη σκυτάλη ασφάλειας, και σε συνδυασμό με το γεγονός ότι νωρίτερα πέρασε με επιτυχία και το login μέσω web, μπορούμε να αποφανθούμε με σιγουριά ότι ο πελάτης είναι πράγματι αυτός που ισχυρίζεται ότι είναι.

Οπότε σε αυτή την περίπτωση, ο πελάτης προωθείται από την ιστοσελίδα που βρισκόταν μετά το επιτυχημένο login και μεταβαίνει σε μια ιστοσελίδα με περιορισμένη πρόσβαση, όπως θα ήταν η σελίδα διαχείρισης του τραπεζικού του λογαριασμού ή μια σελίδα που περιέχει εμπιστευτικές πληροφορίες.

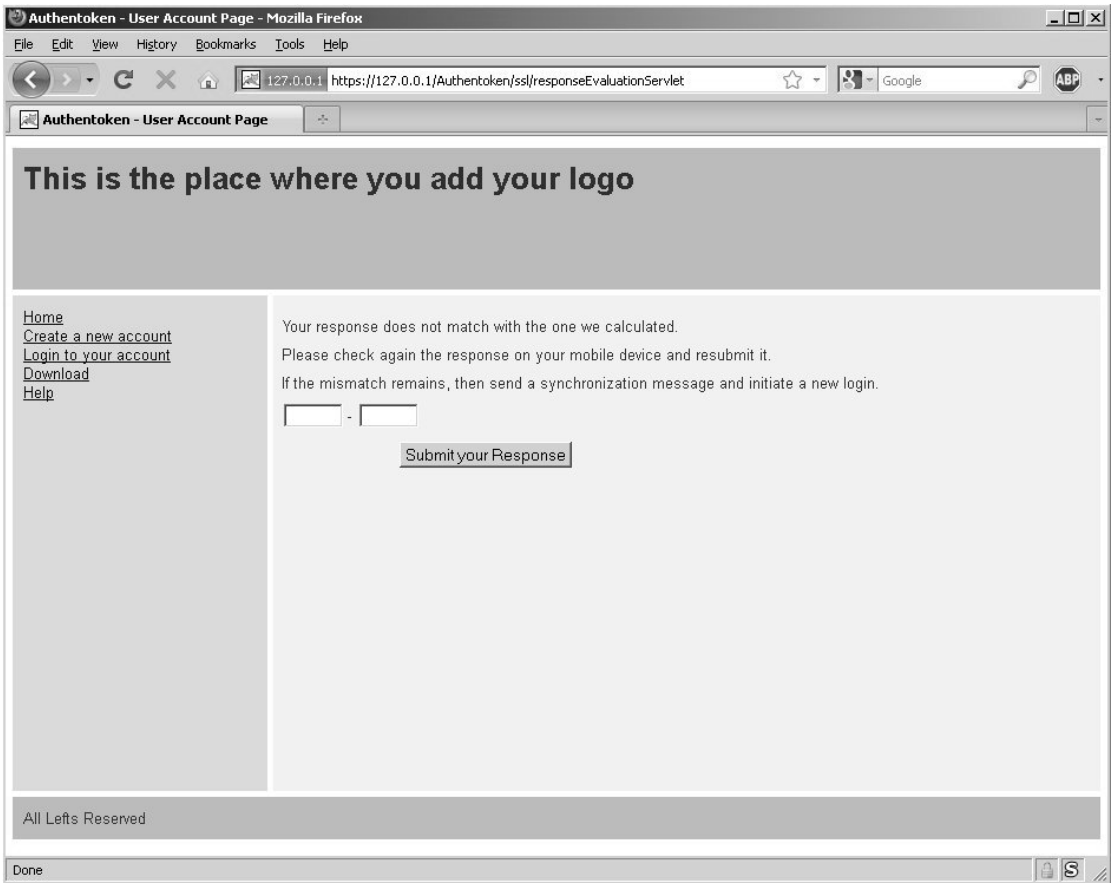


Εικόνα 149: Η ιστοσελίδα με εμπιστευτικές πληροφορίες που εμφανίζεται σε έναν πελάτη που μόλις πέρασε με επιτυχία και το δεύτερο παράγοντα αυθεντικοποίησης

7. *Ενημέρωση στην περίπτωση ενός λάθους*

   Αν το Response του πελάτη δεν είναι ίσο με το αναμενόμενο, τότε υπάρχει το ενδεχόμενο να χρησιμοποιήθηκε από την πλευρά του μια διαφορετική τιμή για το μετρητή, και κατ' επέκταση να δημιουργήθηκε διαφορετική τιμή για το One-Time Password.

   Έτσι, από την πλευρά του εξυπηρετητή εμφανίζεται ένα μήνυμα για συγχρονισμό της τιμής του μετρητή που διατηρούν οι δύο πλευρές.



Εικόνα 150: Η σελίδα που εμφανίζεται στην πλευρά του χρήστη όταν έχουμε μιαν αποτυχημένη προσπάθεια πιστοποίησης ταυτότητας.

Οι δύο πλευρές χρειάζεται να διατηρούν συγχρόνως την ίδια τιμή ενός μετρητή, αφού το συγκεκριμένο στοιχείο χρησιμοποιείται μαζί με την αποθηκευμένη τιμή hash κατά τη δημιουργία του One-Time Password που θα συμμετάσχει σε μια προσπάθεια αυθεντικοποίησης.

Αν οι δύο πλευρές χρησιμοποιήσουν διαφορετικούς μετρητές, τότε θα δημιουργηθούν One-Time Passwords που δεν θα ταυτίζονται, καθιστώντας έτσι ανέφικτη την απομακρυσμένη πιστοποίηση ταυτότητας.

Η αρχική τιμή του μετρητή αποδίδεται και στις δύο πλευρές αμέσως μετά από την αποστολή του κρυπτογραφημένου μηνύματος με το hash του χρήστη.

8. *Αποστολή μηνύματος συγχρονισμού*

Στην περίπτωση που θα παρατηρηθεί κάποια απόκλιση ανάμεσα στις τιμές που διατηρούν ο server και ο χρήστης, τότε υπάρχει η δυνατότητα συγχρονισμού σε μια νέα τιμή.

Η συγκεκριμένη διαδικασία ξεκινά όταν η πλευρά του χρήστη αποστέλλει ένα SMS στο οποίο περιέχεται κρυπτογραφημένη η τιμή του μετρητή που διατηρείται αποθηκευμένη στο κινητό του τηλέφωνο.

Η κρυπτογράφηση του μετρητή γίνεται με τον αλγόριθμο RSA, χρησιμοποιώντας, το ήδη διαθέσιμο στο χρήστη, δημόσιο κλειδί του εξυπηρετητή.

Εικόνα 151: Κεντρική οθόνη του MIDlet, όπου παρέχεται η δυνατότητα αποστολής ενός SMS συγχρονισμού

Η πλευρά του εξυπηρετητή λαμβάνει και αποθηκεύει το νέο μετρητή ακολουθώντας παρόμοια διαδικασία με αυτήν που αναφέρθηκε προηγουμένως, όταν και παρουσιάσαμε πως λαμβάνεται η τιμή hash από ένα μήνυμα.

```
message polling thread running
writing string with carriage return to port: AT+CMGL="REC UNREAD"

reading from port: AT+CMGL="REC UNREAD"
+CMGL: 4686, "REC UNREAD", "+3069         "
S1
OK


1 new message(s)
writing string with carriage return to port: AT+CMGR=4686

reading from port: AT+CMGR=4686
+CMGR: "REC READ", "+3069         ",
S1
OK

phone: 3069
payload: S1

Checking if Client Exists
=========================

Client exists
counter: 1

Updating Counter
=========================
New counter=1

Update completed
zzzzzzzzzzz
```

Εικόνα 152: Η κονσόλα του Apache Tomcat όταν πραγματοποιείται η ανανέωση του μετρητή του πελάτη

6.4 Αποτίμηση της ασφάλειας που παρέχεται από το πρωτόκολλο

Το πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας που αναπτύχθηκε για τις ανάγκες αυτής της πτυχιακής εργασίας βασίζεται στη χρησιμοποίηση δύο ξεχωριστών παραγόντων αυθεντικοποίησης, ενώ παράλληλα προϋποθέτει την πρόσβαση σε δύο διαφορετικά κανάλια επικοινωνίας, όπου στην προκειμένη περίπτωση είναι το δίκτυο κινητής τηλεφωνίας και το Διαδίκτυο.

Για την καλύτερη κατανόηση της ασφάλειας που παρέχεται από το πρωτόκολλο θα εξετάσουμε ξεχωριστά με ποιο τρόπο αλληλεπιδρούν οι παράγοντες που συμμετέχουν σε κάθε ένα από τα κανάλια επικοινωνίας, ξεκινώντας με το δίκτυο κινητής τηλεφωνίας και ολοκληρώνοντας με το Διαδίκτυο.

6.4.1 Η επικοινωνία στο δίκτυο κινητής τηλεφωνίας

Οι οντότητες που συμμετέχουν στο πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας και επικοινωνούν μέσω του δικτύου κινητής τηλεφωνίας είναι το κινητό τηλέφωνο που έχει στην κατοχή του ο πελάτης και το GSM Modem που είναι συνδεδεμένο με τον εξυπηρετητή.

Μεταξύ των δύο οντοτήτων που προαναφέραμε, υπάρχουν τρεις περιπτώσεις κατά τις οποίες επικοινωνούν απευθείας, πρόκειται για τις φάσεις όπου:

- Ο πελάτης εγγράφεται στη διαδικτυακή υπηρεσία.
- Ο πελάτης λαμβάνει το μήνυμα με την τιμή της πρόκλησης.
- Ο πελάτης αποστέλλει ένα μήνυμα για συγχρονισμό με τον εξυπηρετητή.

<u>Η εγγραφή του πελάτη στη διαδικτυακή υπηρεσία</u>

Κατά την εγγραφή ενός πελάτη στη διαδικτυακή υπηρεσία που παρέχεται από την πλευρά του εξυπηρετητή φτάνουμε στο στάδιο όπου πρέπει να ενεργοποιηθεί το MIDlet που έχει εγκαταστήσει ο πελάτης στο κινητό του τηλέφωνο.

Τότε ο πελάτης καλείται από το MIDlet, που μόλις έχει θέσει σε λειτουργία, να αποστείλει ένα μήνυμα SMS που θα περιέχει το hash που έχει δημιουργηθεί από τα στοιχεία που χρησιμοποίησε κατά την είσοδο του στο MIDlet.

Το συγκεκριμένο μήνυμα αποστέλλεται μόνο μία φορά και το περιεχόμενο του είναι διασφαλισμένο αφού έχει κρυπτογραφηθεί με τον αλγόριθμο RSA.

Για την κρυπτογράφηση του περιεχομένου έχει χρησιμοποιηθεί το δημόσιο κλειδί του εξυπηρετητή, το οποίο ήδη βρίσκεται αποθηκευμένο στο MIDlet, οπότε για την ανάκτηση του περιεχομένου απαιτείται η χρησιμοποίηση του ιδιωτικού κλειδιού του εξυπηρετητή που είναι γνωστό μόνο σε αυτόν.

Η λήψη ενός μηνύματος που περιέχει την τιμή μιας πρόκλησης

Όταν ο πελάτης που χρησιμοποιεί το πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας περάσει από το στάδιο του log-in, όπου υποβάλλει στην πλευρά του εξυπηρετητή τα διαπιστευτήρια του, δηλαδή το username και το password που του αντιστοιχούν, τότε έχουμε τη δημιουργία μιας πρόκλησης στην οποία καλείται να απαντήσει ο πελάτης.

Η πρόκληση δημιουργείται από τη διαδικτυακή εφαρμογή και προωθείται στο GSM Modem, όπου εκεί μετατρέπεται στην κατάλληλη μορφή και αποστέλλεται ως μήνυμα SMS προς την πλευρά του πελάτη.

Το περιεχόμενο του μηνύματος, όπου στην ουσία είναι το πλήθος των χιλιοστών του δευτερολέπτου από την 1^{η} Ιανουαρίου του 1970, μεταδίδεται δίχως κρυπτογράφηση.

Όμως το γεγονός αυτό δεν επηρεάζει καθόλου τη λειτουργία του πρωτοκόλλου αφού ακόμη και να υποκλέψει κάποιος τρίτος την τιμή της πρόκλησης δεν μπορεί να οδηγηθεί στην κατάλληλη απάντηση, καθώς οι δύο πλευρές διατηρούν μυστικά τα υπόλοιπα στοιχεία που χρειάζονται για τον υπολογισμό της απάντησης.

Η αποστολή ενός μηνύματος συγχρονισμού με τον εξυπηρετητή

Είδαμε νωρίτερα ότι κατά την αυθεντικοποίηση με το δεύτερο παράγοντα, δηλαδή με την απάντηση που παράγεται από το MIDlet του πελάτη, υπάρχει το ενδεχόμενο οι δύο πλευρές να δημιουργούν διαφορετικές τιμές, και κατ' επέκταση να αποτυγχάνει ο έλεγχος της ταυτότητας του πελάτη.

Ο κυριότερος παράγοντας που μπορεί να οδηγήσει στη δημιουργία διαφορετικών απαντήσεων είναι η χρησιμοποίηση διαφορετικής τιμής για τον κοινό μετρητή που διατηρούν οι δύο πλευρές που συμμετέχουν στο πρωτόκολλο.

Σε αυτή την περίπτωση δίνεται η δυνατότητα, με την αποστολή ενός SMS από την πλευρά του πελάτη, να συγχρονιστούν οι δύο πλευρές και να ξεκινήσουν να χρησιμοποιούν μια νέα τιμή για το μετρητή.

Επειδή η τιμή του μετρητή θεωρείται σημαντική, αφού συμμετέχει κατά τον υπολογισμό της απάντησης που υποβάλλει ο πελάτης, φροντίζουμε για τη διαφύλαξη της και γι' αυτό το λόγο τη μεταδίδουμε κρυπτογραφημένη.

Ο μετρητής κρυπτογραφείται με τον αλγόριθμο RSA, όπου για αυτή την ενέργεια χρησιμοποιούμε το δημόσιο κλειδί του εξυπηρετητή, διασφαλίζοντας με αυτό τον τρόπο ότι μόνο ο εξυπηρετητής είναι σε θέση να αποκαλύψει το περιεχόμενο του κρυπτογραφημένου μηνύματος.

6.4.2 Η επικοινωνία στο Διαδίκτυο

Το Διαδίκτυο αποτελεί το μέσο με το οποίο επικοινωνεί η πλευρά του πελάτη με την εφαρμογή που παρέχεται από την πλευρά του εξυπηρετητή. Για να αποκτήσει ο πελάτης πρόσβαση στην εν λόγω εφαρμογή πρέπει να έχει εγκατεστημένο στον υπολογιστή του ένα πρόγραμμα web browser, μέσω του οποίου μπορεί να πραγματοποιήσει τις ακόλουθες ενέργειες:

- Να εγγραφεί στη διαδικτυακή εφαρμογή και, αφού ρυθμίσει τις παραμέτρους με τις οποίες θα εισάγεται σε αυτήν, να κατεβάσει στον υπολογιστή του το συνοδευτικό MIDlet.
- Να εισέλθει στη διαδικτυακή εφαρμογή και να κατεβάσει εκ νέου, αν αυτό είναι απαραίτητο, το MIDlet που θα εκτελείται στο κινητό του τηλέφωνο και θα αποτελεί το δεύτερο παράγοντα πιστοποίησης ταυτότητας.
- Να εισέλθει στη διαδικτυακή εφαρμογή και να αποκτήσει πρόσβαση σε εμπιστευτικό υλικό, όπου στην περίπτωση μας είναι μια συγκεκριμένη ιστοσελίδα, αφού βεβαίως πιστοποιήσει επιτυχώς την ταυτότητα του και με τους δύο παράγοντες που απαιτούνται.

Σε κάθε μία από τις προαναφερθείσες ενέργειες του πελάτη ξεκινά ένα καινούργιο session σύνδεσης σύμφωνα με το πρωτόκολλο TLS. Θυμίζουμε ότι για την ορθή λειτουργία του πρωτοκόλλου TLS χρειάζεται να έχει ο εξυπηρετητής ένα έγκυρο ψηφιακό πιστοποιητικό στην κατοχή του. Άλλωστε, με αυτό ως βάση γίνεται ο έλεγχος της ταυτότητας του εξυπηρετητή και στη συνέχεια ξεκινά η κρυπτογράφηση των πληροφοριών που ανταλλάσσονται με τον πελάτη.

Με αυτό τον τρόπο η γενικότερη ασφάλεια που παρέχεται από το πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας βασίζεται σε μεγάλο βαθμό στην αξιόπιστη και ασφαλή λειτουργία του πρωτοκόλλου TLS, το οποίο βεβαίως αποτελεί και τη βάση για την ασφάλεια σχεδόν όλων των διαδικτυακών εφαρμογών.

Όμως το πρωτόκολλο TLS δεν λειτουργεί πάντοτε σωστά, αφού συναντάμε σε αυτό μια σειρά από προβλήματα όπως:

- Η περιορισμένη εξάπλωση της χρήσης ψηφιακών πιστοποιητικών, με πολλούς διαδικτυακούς τόπους να μην αξιολογούν σωστά την ανάγκη για διασφάλιση των δεδομένων με τα οποία σχετίζονται και έτσι να μην ενσωματώνουν το πρωτόκολλο.

- Σε πολλές περιπτώσεις το πρωτόκολλο δεν υλοποιείται σωστά από την πλευρά του εξυπηρετητή, αφού μπορεί να χρησιμοποιείται κάποιο πιστοποιητικό που έχει λήξει ή που δεν είναι υπογεγραμμένο από κάποιον έμπιστο Πάροχο Πιστοποίησης.
- Υπάρχει πάντα το ενδεχόμενο να μην υλοποιείται σωστά το πρωτόκολλο από την πλευρά του browser, δίνοντας σε έναν επιτιθέμενο τη δυνατότητα να ξεγελάσει το χρήστη. Μια περίπτωση που έχει αναφερθεί είναι όταν σε ένα domain περιέχεται ένας συγκεκριμένος χαρακτήρας διαφυγής[^], επιτρέποντας έτσι τη μεταφορά του χρήστη σε άλλο site χωρίς αυτός να αντιληφθεί το παραμικρό.
- Το γεγονός ότι λίγοι χρήστες ελέγχουν τα ψηφιακά πιστοποιητικά που χρησιμοποιούν.
- Η εμφάνιση τρωτών σημείων στη λειτουργία του πρωτοκόλλου που αγνοούσαμε την ύπαρξη τους όπως, για παράδειγμα, το πολύ σημαντικό κενό που ανακαλύφθηκε κατά την επαναδιαπραγμάτευση των παραμέτρων ενός session[*].
- Η πλειονότητα των χρηστών αγνοεί τις προειδοποιήσεις που εμφανίζει ο browser όταν υπάρχει κάποιο πρόβλημα με τη λειτουργία του πρωτοκόλλου, για παράδειγμα, όταν ένα πιστοποιητικό προορίζεται για κάποιο άλλο domain ή δεν είναι υπογεγραμμένο από κάποιον έμπιστο Πάροχο Πιστοποίησης.

Ειδικότερα το τελευταίο γεγονός οδηγεί σε πληθώρα επιθέσεων Phishing, κατά τις οποίες αρκετοί επιτήδειοι ξεγελούν τους χρήστες και αποσπούν από αυτούς εμπιστευτικά δεδομένα, τα οποία στη συνέχεια χρησιμοποιούν για το δικό τους όφελος.

Ακόμη, δεν πρέπει να παραβλέψουμε το ρόλο που θα μπορούσε να διαδραματίσει μια κακόβουλη εφαρμογή που θα ήταν εγκατεστημένη σε κάποια από τις οντότητες που συμμετέχουν στο πρωτόκολλο, αφού σε απόλυτη συνάρτηση με τις ιδιότητες της, θα έδινε μια σειρά από δυνατότητες στα χέρια ενός πιθανού επιτιθέμενου. Οι πιο συνηθισμένοι τρόποι εγκατάστασης κακόβουλου λογισμικού είναι:

- Η εκμετάλλευση κενών ασφαλείας που προϋπάρχουν στην πλατφόρμα του εξυπηρετητή ή σε κάποιο από τα νόμιμα εγκατεστημένα προγράμματα(Zero-Day Hack).
- Η άγνοια κινδύνου ή η έλλειψη εμπειρίας που μπορούμε να συναντήσουμε σε ορισμένους χρήστες, γεγονός που ενδέχεται να τους οδηγήσει στο να εγκαταστήσουν οι ίδιο το κακόβουλο λογισμικό.

Επίσης, οφείλουμε να αναφέρουμε και το γεγονός ότι η παρουσία της βάσης δεδομένων στην πλευρά του εξυπηρετητή προκαλεί το ενδιαφέρον ενός επιτιθέμενου. Αφού μια ενδεχόμενη επίθεση SQL Injection σε αυτήν θα μπορούσε να του

[^] http://www.wired.com/threatlevel/2009/07/kaminsky/
[*] http://extendedsubset.com/?p=8

αποκομίσει πολλαπλά οφέλη, με κυριότερο εξ αυτών την πρόσβαση στα δεδομένα της βάσης, ενώ μπορεί να δοθεί στον επιτιθέμενο και η δυνατότητα να εκμεταλλευτεί ήδη υπάρχοντα εργαλεία για απομακρυσμένο έλεγχο του server και να τα χρησιμοποιήσει όπως θα έκανε ο νόμιμος διαχειριστής.

Τα πιθανά αποτελέσματα από τη λειτουργία κακόβουλου λογισμικού είναι πολύ δύσκολο να καλυφθούν πλήρως αφού επηρεάζουν με πολλούς τρόπους τη λειτουργία του πρωτοκόλλου. Μόνο μερικές από αυτές τις περιπτώσεις, μαζί με τις συνέπειες τους, είναι οι ακόλουθες:

- Η ύπαρξη key logger, μέσω του οποίου καταγράφεται οποιοδήποτε πλήκτρο πατηθεί στο πληκτρολόγιο. Από το αρχείο καταγραφής μπορούν να ανακτηθούν στοιχεία όπως είναι το όνομα χρήστη και ο κωδικός πρόσβασης σε ένα site.
- Η παρουσία κάποιου προγράμματος packet sniffer, με το οποίο παρακολουθείται η κίνηση και το περιεχόμενο των πακέτων που μεταφέρονται στο δίκτυο.
- Η τροποποίηση της λειτουργίας του πρωτοκόλλου DNS, μέσω κάποιας επίθεσης DNS Spoofing ή DNS Poisoning. Το θύμα της επίθεσης δεν κατευθύνεται στο διαδικτυακό τόπο που πραγματικά επιθυμεί, αλλά πολύ συχνά, μεταφέρεται σε κάποιο κλώνο του site που λειτουργεί ως δόλωμα σε μια επίθεση Phishing. Βεβαίως, σε ένα εναλλακτικό σενάριο, το θύμα θα μπορούσε να προωθηθεί σε μια σελίδα που να μην κάνει τίποτα, με τον επιτιθέμενο να αρνείται την πρόσβαση στο νόμιμο site, κάτι δηλαδή ισοδύναμο με επίθεση Denial of Service.
- Η παρέμβαση στο περιεχόμενο που εμφανίζεται στο browser του πελάτη(Man in the Browser).
- Η βίαιη ανάληψη του ελέγχου ενός session που έχει ξεκινήσει από την πλευρά του πελάτη(Session High jacking).

6.4.3 Τελικό συμπέρασμα

Μετά από την ανάλυση των ενεργειών που λαμβάνουν χώρα στα δύο κανάλια επικοινωνίας είδαμε ότι κατά τη χρησιμοποίηση του δικτύου κινητής τηλεφωνίας η κατάσταση είναι περισσότερο ξεκάθαρη και υπάρχουν λιγότεροι κίνδυνοι για τις δύο πλευρές που συμμετέχουν στο πρωτόκολλο.

Όμως, στην περίπτωση όπου χρησιμοποιείται το Διαδίκτυο, είδαμε ότι υπάρχει ένα μεγάλο εύρος σημείων που χρήζουν ιδιαίτερης προσοχής, καθώς αν υποθέσουμε ότι μια κακόβουλη τρίτη πλευρά εκμεταλλεύεται μία αδυναμία, ή ένα συνδυασμό τρωτοτήτων, μπορεί να επηρεάσει σε μεγάλο βαθμό την επικοινωνία ανάμεσα στις δύο πλευρές και να αχρηστεύσει τη λειτουργία του πρωτοκόλλου.

Οπότε, παρά την προσθήκη του δεύτερου παράγοντα πιστοποίησης ταυτότητας, σε σχέση με την παραδοσιακή λογική της εισαγωγής ονόματος χρήστη και κωδικού πρόσβασης, δεν μπορούμε σε καμία περίπτωση να είμαστε σίγουροι ότι δεν διατρέχουμε κάποιο κίνδυνο. Διότι σε θεωρητικό, τουλάχιστον, επίπεδο παραμένει εφικτή η δυνατότητα μιας συντονισμένης και καλά οργανωμένης επίθεσης από μια τρίτη πλευρά.

6.5 Αποτελέσματα Εργασίας – Μελλοντική Έρευνα

Στο δεύτερο σκέλος της τρέχουσας πτυχιακής εργασίας παρατηρήσαμε τον τρόπο με τον οποίο δομήθηκε ένα πρωτόκολλο απομακρυσμένης πιστοποίησης της ταυτότητας ενός χρήστη που επιθυμεί να χρησιμοποιήσει μια διαδικτυακή εφαρμογή.

Η υλοποίηση του πρωτοκόλλου έγινε σύμφωνα με τις απαιτήσεις που είχαν τεθεί εξαρχής, οπότε για την πιστοποίηση της ταυτότητας ενός χρήστη που συμμετέχει σε αυτό πρέπει να χρησιμοποιηθούν δύο παράγοντες αυθεντικοποίησης, με τον ένα εξ αυτών να παρέχεται μέσα από τη λειτουργία ενός MIDlet που εκτελείται στο κινητό τηλέφωνο του χρήστη και τον άλλο να αντιπροσωπεύεται από το συνδυασμό του username και του password που υποβάλει ο χρήστης στο site.

Από το αποτέλεσμα της λειτουργίας του πρωτοκόλλου προκύπτει ότι αυτό λειτουργεί ικανοποιητικά, όμως, όπως είδαμε και στην αποτίμηση της ασφάλειας που παρέχεται από το πρωτόκολλο, υπάρχει μια πληθώρα από παράγοντες που μπορούν να επηρεάσουν την ορθή λειτουργία του.

Οπότε στο μέλλον θα μπορούσαμε να εξετάσουμε αν μπορούν να ληφθούν επιπρόσθετα μέτρα, και κατά πόσο βέβαια, θα μπορούσαν να θωρακίσουν τη λειτουργία του πρωτοκόλλου.

Ακόμη, θα μπορούσαμε να υλοποιήσουμε και ένα εναλλακτικό σενάριο, κατά το οποίο ο στόχος του πρωτοκόλλου δε θα ήταν η εξακρίβωση της ταυτότητας ενός χρήστη, αλλά η επιβεβαίωση μιας ενέργειας που εκτελείται από απόσταση, όπως θα ήταν, για παράδειγμα, η αποπληρωμή ενός λογαριασμού μιας υπηρεσίας κοινής ωφέλειας.

Σε αυτή την περίπτωση, ο χρήστης θα μπορούσε να εισέλθει σε ένα site όπου θα αναφέρονται οι λεπτομέρειες που αφορούν το λογαριασμό του και να επιλέξει την αποπληρωμή του συγκεκριμένου λογαριασμού.

Στη συνέχεια, ο χρήστης λαμβάνει, σε ένα MIDlet που εκτελείται στο κινητό του τηλέφωνο, ένα μήνυμα που τον ενημερώνει για την ενέργεια που υποβλήθηκε μέσω του site, ενώ παράλληλα διερωτάται αν είναι σύμφωνος με αυτήν.

Αν πράγματι ο χρήστης είναι σύμφωνος με την εκτέλεση της ενέργειας, τότε, με το πάτημα ενός κουμπιού στέλνει το κατάλληλο μήνυμα προς την πλευρά του εξυπηρετητή και προχωρά στην επιβεβαίωση της ενέργειας μέσα από ένα εναλλακτικό κανάλι επικοινωνίας(Out-of-Band Verification).

Παράρτημα 1 - Οδηγός Ρυθμίσεων

Το συγκεκριμένο παράρτημα έχει ως στόχο να παρουσιάσει τον τρόπο με τον οποίο θα μπορέσουμε να εκτελέσουμε όσες εφαρμογές αναπτύχθηκαν κατά το πρακτικό μέρος της παρούσης πτυχιακής εργασίας.

Για να κάνουμε περισσότερο εύκολη την κατάσταση για κάποιον που επιθυμεί να αναπαράγει κάποιο από τα υλοποιηθέντα πρωτόκολλα θα αναφέρουμε ξεχωριστά τι πρέπει να είναι εγκατεστημένο και ρυθμισμένο στον προσωπικό του υπολογιστή.

Όλα τα προγράμματα που αναφέρονται στις ακόλουθες σελίδες βρίσκονται στον ψηφιακό δίσκο που συνοδεύει την αναφορά, όμως για παν ενδεχόμενο υπάρχουν και σύνδεσμοι στους κατάλληλους διαδικτυακούς τόπους όπου μπορούμε να λάβουμε τα απαιτούμενα πακέτα λογισμικού.

Στην περίπτωση που επιθυμούμε να εκτελέσουμε το MIDlet που μας επιτρέπει την ασφαλή ανταλλαγή σύντομων μηνυμάτων τότε πρέπει να εγκαταστήσουμε:

- Το Java Standard Edition Development Kit
- Το Standard Development Kit της Sony Ericsson

Όταν θέλουμε να θέσουμε σε λειτουργία το πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας τότε χρειαζόμαστε:

- Το Java Standard Edition Development Kit
- Το Integrated Development Environment NetBeans
- Το MySQL Server
- Τον Apache Tomcat HTTP Application Server
- Ασφαλείς συνδέσεις σύμφωνα με το πρωτόκολλο TLS-SSL
- Το Java Communications API και το driver MySQL Connector/J

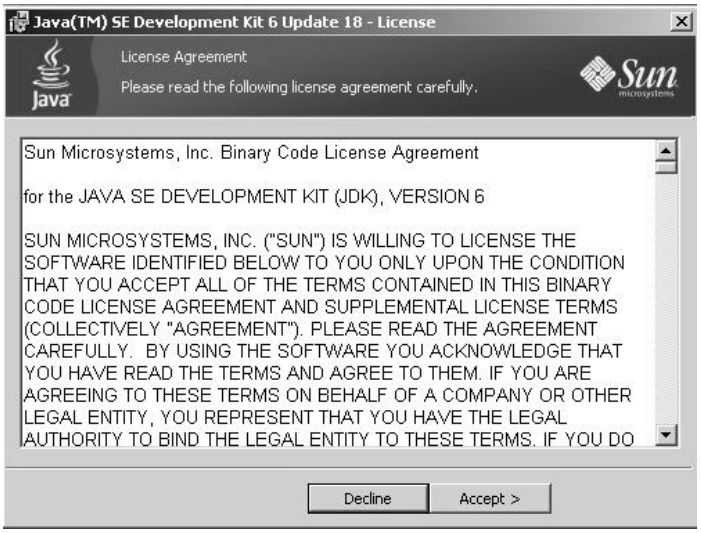
Εγκατάσταση του Java Standard Edition Development Kit

Ξεκινάμε με τη διαδικασία της εγκατάστασης του Java Standard Edition Development Kit, ή συντομότερα JDK, το οποίο μπορούμε να κατεβάσουμε ελεύθερα από το url: www.oracle.com/technetwork/java/javase/downloads/index.html

Καθώς η προαναφερθείσα ιστοσελίδα περιέχει πολλά αρχεία πρέπει να είμαστε ιδιαιτέρως προσεκτικοί και να επιλέξουμε αποκλειστικά και μόνο εκείνο που φέρει την ονομασία Java Development Kit.

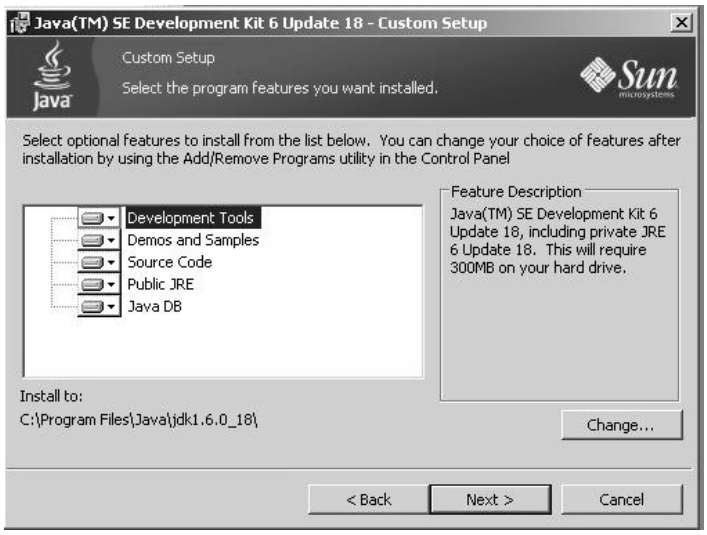
Μετά από την ολοκλήρωση της λήψης του αρχείου, αρκεί να κάνουμε διπλό κλικ σε αυτό και η διαδικασία ξεκινά.

Στο πρώτο παράθυρο που συναντάμε εμφανίζεται η άδεια χρήσης που συνοδεύει το λογισμικό και τίθεται το ερώτημα της αποδοχής των όρων που περιέχονται σε αυτήν.



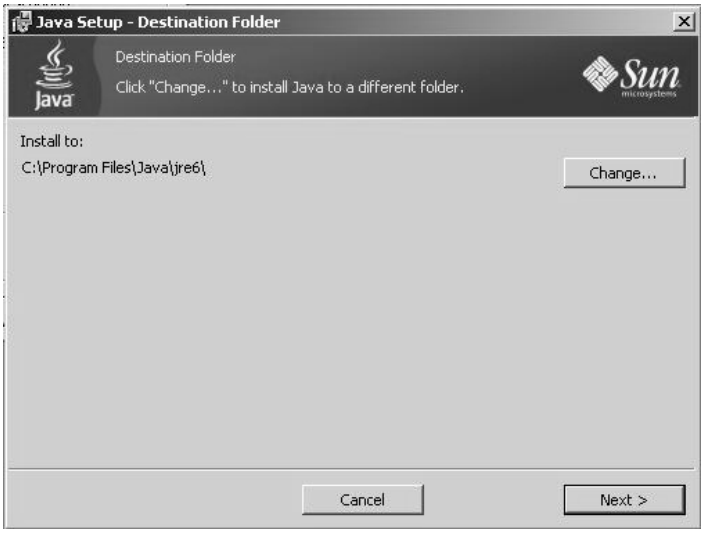
Εικόνα 153: Άδεια χρήσης του Java Standard Edition Development Kit

Μετά από την αποδοχή της άδειας, περνάμε στο επόμενο παράθυρο όπου επιλέγουμε ποια από τα στοιχεία του JDK επιθυμούμε να εγκατασταθούν καθώς και την τοποθεσία όπου θα αποθηκευθούν. Για τη δική μας περίπτωση δεν πραγματοποιούμε κάποια αλλαγή, αλλά αφήνουμε τα στοιχεία ως έχουν.



Εικόνα 154: Επιλογή στοιχείων και τοποθεσίας JDK

Στη συνέχεια, έχουμε τη δυνατότητα να αλλάξουμε το φάκελο εγκατάστασης του Java Runtime Environment, όμως εμείς δεν πραγματοποιήσαμε κάποια αλλαγή.



Εικόνα 155: Επιλογή τοποθεσίας εγκατάστασης του Java Runtime Environment

Αν όλα πάνε καλά, τότε έχουμε ολοκληρώσει με επιτυχία τη διαδικασία εγκατάστασης του JDK, κάτι που μας πιστοποιεί και το τελευταίο παράθυρο που μας εμφανίζεται και περιέχει το ανάλογο μήνυμα.

Εικόνα 156: Ολοκλήρωση της εγκατάστασης του JDK

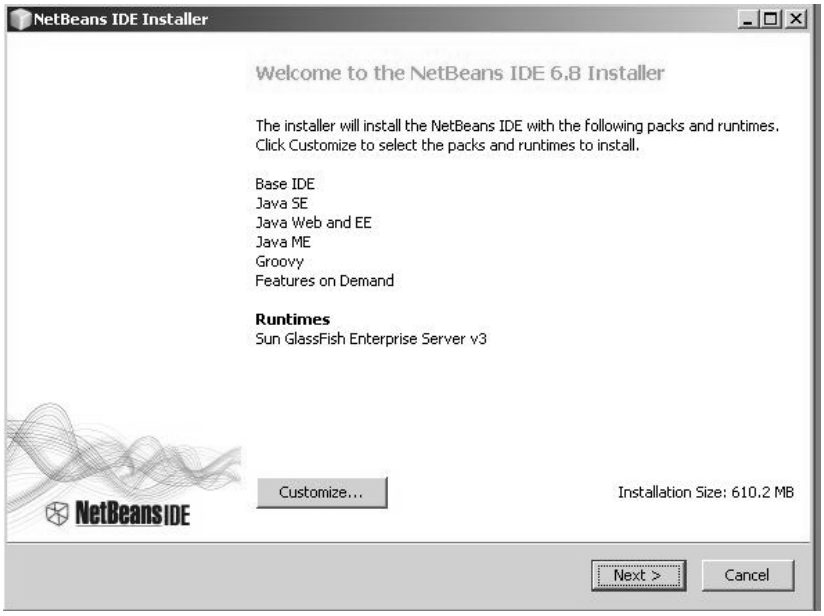
Εγκατάσταση του Integrated Development Environment NetBeans

Το περιβάλλον ανάπτυξης εφαρμογών NetBeans διατίθεται δωρεάν στο url: http://netbeans.org/downloads/index.html

Όπως είναι φανερό και από την παραπάνω ιστοσελίδα, υπάρχουν αρκετές εκδόσεις του περιβάλλοντος NetBeans, όμως εκείνη που ανταποκρίνεται περισσότερο στις ανάγκες μας είναι η full έκδοση.

Μετά από το κατέβασμα του αρχείου installer χρειάζεται ένα διπλό κλικ σε αυτό και η εγκατάσταση εκκινεί.

Το πρώτο παράθυρο που μας εμφανίζεται αναφέρει σε μια λίστα τα πακέτα που πρόκειται να εγκατασταθούν και μας δίνει τη δυνατότητα τροποποίησης αυτής.

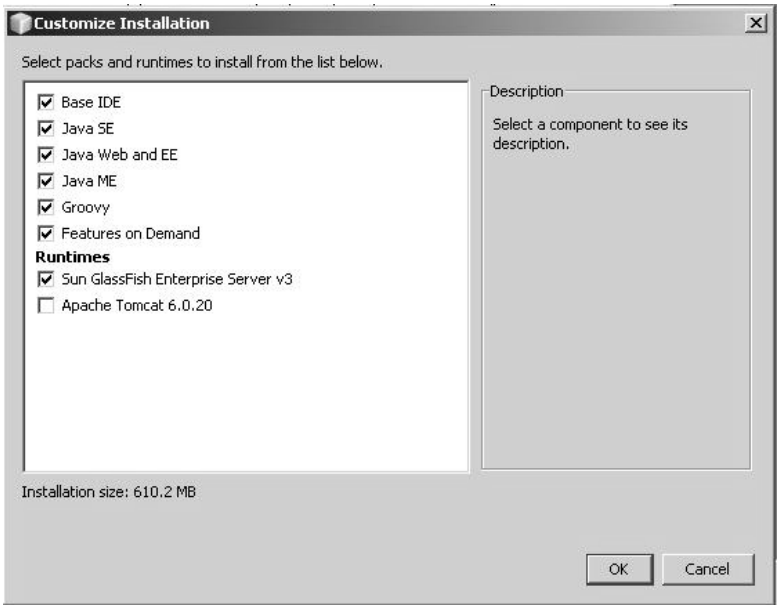


Εικόνα 157: Εμφάνιση πακέτων που θα εγκατασταθούν με το NetBeans

Εκμεταλλευόμαστε αυτή την ευκαιρία και επιλέγουμε Customize, οπότε τώρα μπορούμε να επιλέξουμε τι πρόκειται να εγκατασταθεί. Για τις δικές μας ανάγκες είναι απαραίτητη η παρουσία των:

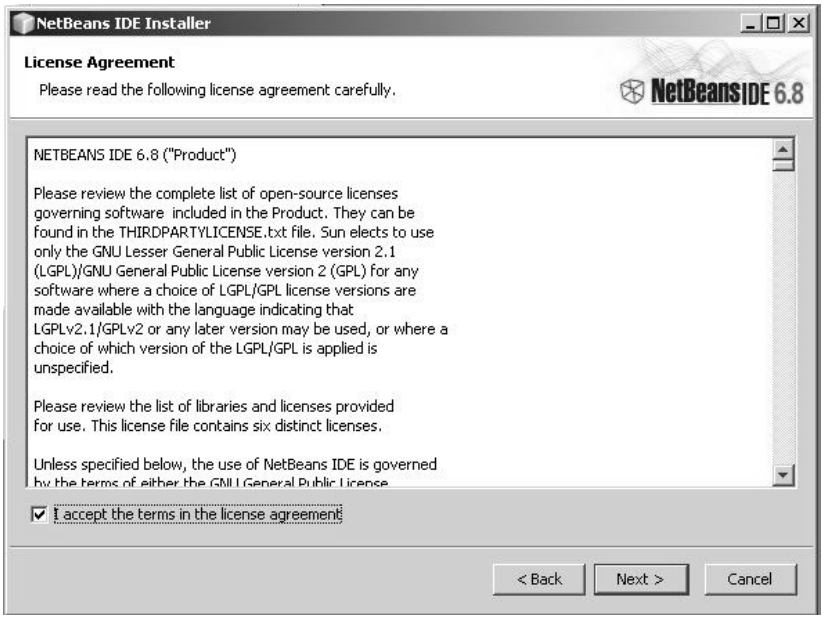
- Base IDE
- Java SE
- Java Web and EE
- Java ME

Το Runtime του Apache Tomcat, αν και μας είναι απαραίτητο, δεν θα το εγκαταστήσουμε μαζί με το NetBeans, καθώς έτσι θα έκανε κατά ένα βαθμό πιο πολύπλοκη την αναπαραγωγή του πρωτοκόλλου απομακρυσμένης πιστοποίησης ταυτότητας.



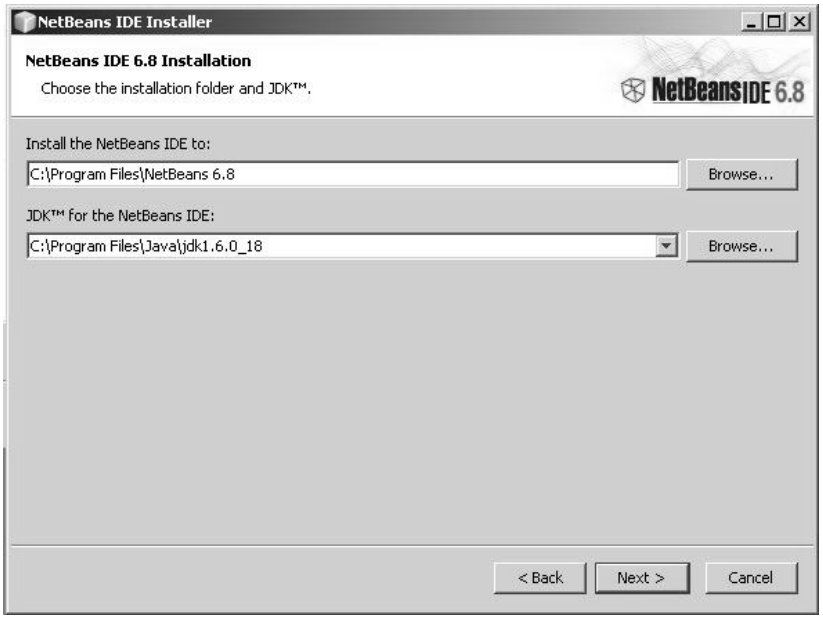
Εικόνα 158: Επιλογή των πακέτων που θα εγκατασταθούν με το NetBeans

Στη συνέχεια εμφανίζεται η άδεια που συνοδεύει το περιβάλλον NetBeans.



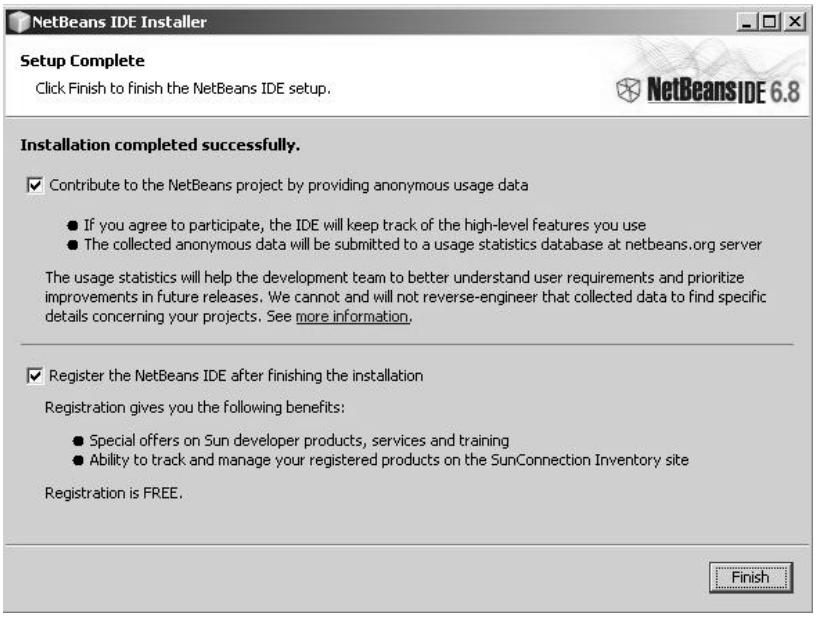
Εικόνα 159: Άδεια χρήσης του περιβάλλοντος NetBeans

Αποδεχόμαστε τους όρους της άδειας και προχωράμε στο επόμενο παράθυρο όπου επιλέγουμε το φάκελο εγκατάστασης του NetBeans και το φάκελο στον οποίο έχουμε προηγουμένως εγκαταστήσει το JDK.



Εικόνα 160: Επιλογή φακέλων εγκατάστασης

Μετά απ' όλα αυτά, και αν δεν υπάρξει κάποιο σφάλμα, έχουμε ολοκληρώσει με επιτυχία την εγκατάσταση του Integrated Development Environment NetBeans. Έτσι, στο παράθυρο που ακολουθεί εμφανίζεται το ανάλογο μήνυμα που μας ενημερώνει για την ολοκλήρωση της εγκατάστασης.



Εικόνα 161: Ολοκλήρωση εγκατάστασης NetBeans

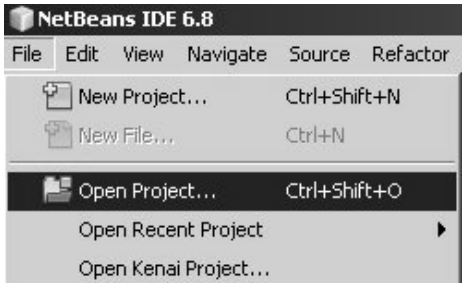
Δημιουργία και εκτέλεση ενός MIDlet μέσω του NetBeans

Η δημιουργία ενός νέου MIDlet είναι απαραίτητη κατά την περίπτωση όπου θέλουμε να εκτελέσουμε το πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας, όπου θα

πρέπει να εισάγουμε στον πηγαίο κώδικα το δημόσιο κλειδί που παράγεται από την πλευρά του εξυπηρετητή, καθώς και τον αριθμό τηλεφώνου που θα χρησιμοποιείται από το GSM Modem που διαχειρίζεται την εισερχόμενη κίνηση.

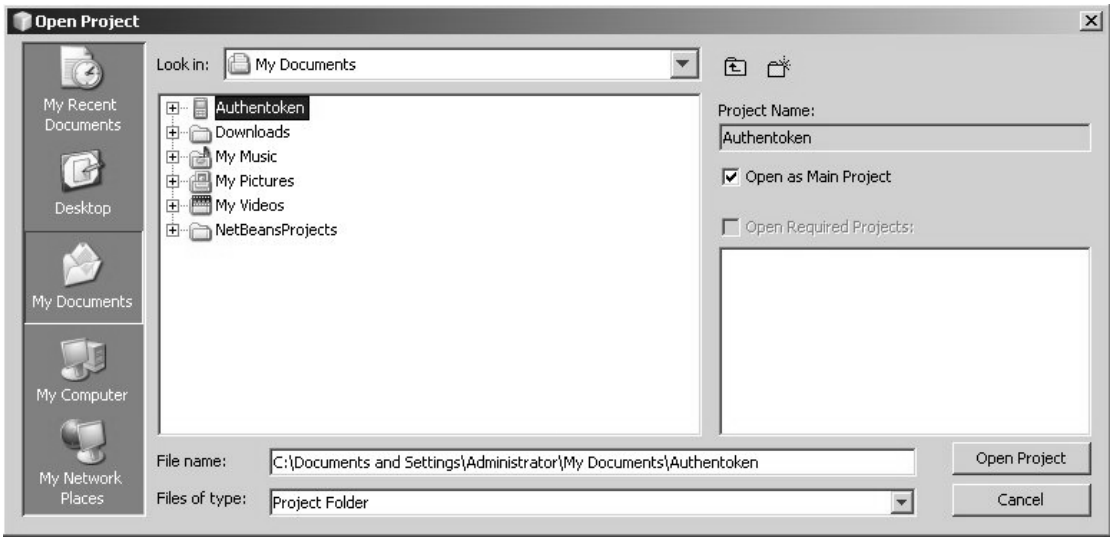
Βέβαια, εμείς δεν θα δημιουργήσουμε ένα εντελώς καινούργιο MIDlet, ξεκινώντας από μηδενική βάση, αλλά θα επεξεργαστούμε κατάλληλα ένα προϋπάρχον project για το NetBeans IDE, το οποίο βρίσκεται στο συνοδευτικό DVD.

Για το άνοιγμα του project θα εντοπίσουμε στη μπάρα του μενού του NetBeans IDE το “File” και από εκεί θα επιλέξουμε το “Open Project…”.



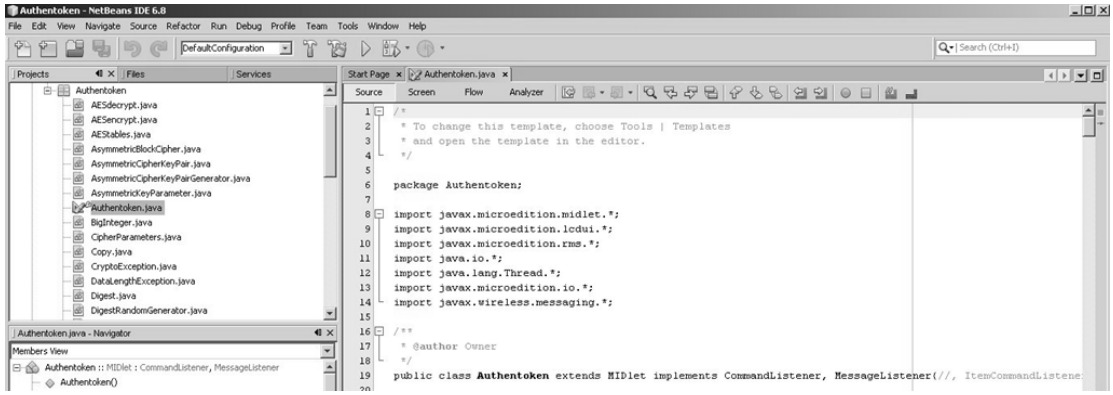
Εικόνα 162: Άνοιγμα του project από το menu του NetBeans IDE

Στη συνέχεια μας εμφανίζεται ένα παράθυρο για την επιλογή της διαδρομής στην οποία θα εντοπίσουμε το φάκελο που φιλοξενεί το ήδη δημιουργηθέν project.



Εικόνα 163: Επιλογή του project μέσω του παρεχόμενου file chooser

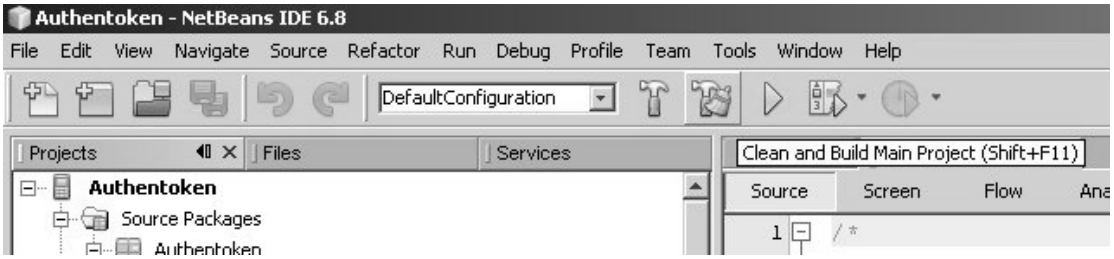
Μετά από την επιλογή του φακέλου περνάμε στο κεντρικό παράθυρο του IDE, όπου τώρα μας εμφανίζεται μια σειρά στοιχείων που αφορούν το ανοιχτό project, όπως είναι τα αρχεία πηγαίου κώδικα που συμμετέχουν σε αυτό.



Εικόνα 164: Εμφάνιση του πηγαίου κώδικα του MIDlet

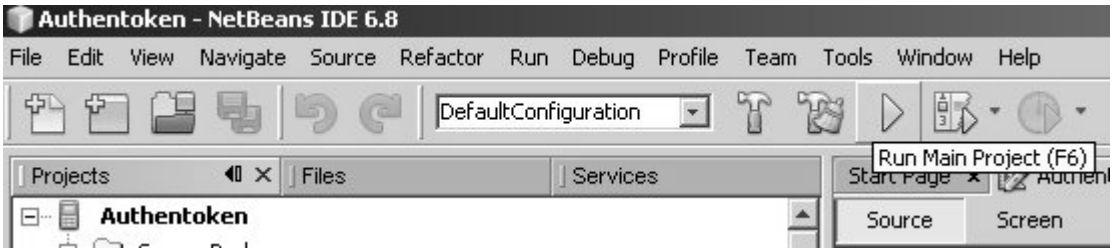
Ακόμη, από τη μπάρα εργαλείων του IDE μπορούμε να δημιουργήσουμε μια νέα υλοποίηση του project, ακολουθώντας βεβαίως το Configuration που έχουμε επιλέξει, είτε μέσω της επιλογής “Build Main Project” ή μέσω της επιλογής “Clean and Build Main Project”. Η διαφορά ανάμεσα στις δύο επιλογές είναι ότι στην περίπτωση της δεύτερης προηγείται η διαγραφή των φακέλων που περιέχουν κλάσσεις από παλαιότερες υλοποιήσεις και ξεκινά η δημιουργία νέων.

Το αποτέλεσμα που προκύπτει από τη νέα υλοποίηση του project το παραλαμβάνουμε με τη μορφή ενός αρχείου jar και ενός αρχείου jad από το φάκελο dist που βρίσκεται μέσα στο φάκελο που φιλοξενεί το project.



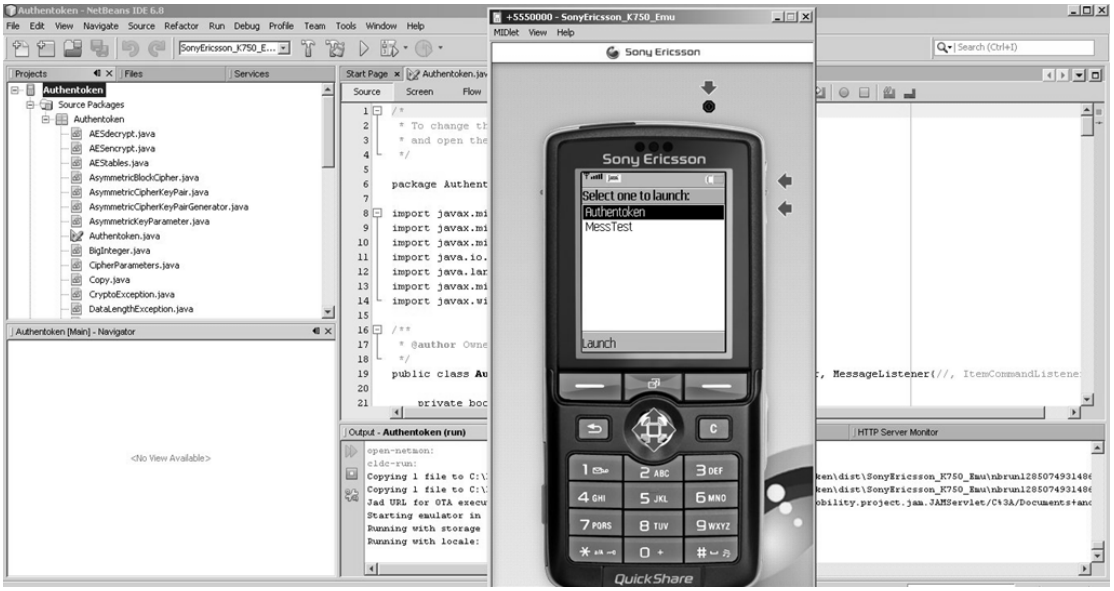
Εικόνα 165: Επιλογή για Clean and Build

Αφού περάσουμε από το στάδιο του “Build” μπορούμε, με τη βοήθεια του emulator που παρέχεται, να προσομοιώσουμε τη λειτουργία του project που μόλις δημιουργήσαμε για το τρέχον Configuration πατώντας το πλήκτρο “Run Main Project” που βρίσκεται στη μπάρα εργαλείων.



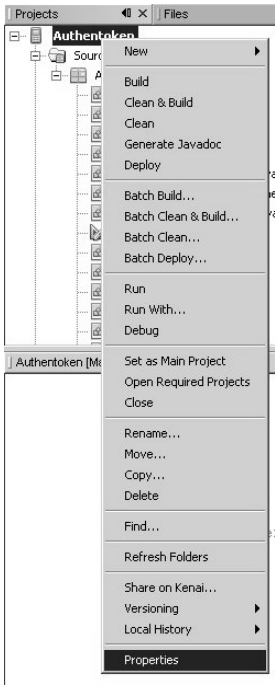
Εικόνα 166: Η επιλογή για την εκτέλεση του project

Οπότε, μετά από την επιλογή για την εκτέλεση του project μας εμφανίζεται το skin που αντιστοιχεί στη συσκευή που έχουμε επιλέξει στο τρέχον Configuration δίνοντας μας με αυτό τον τρόπο μιαν εικόνα για το πώς θα είναι το MIDlet.



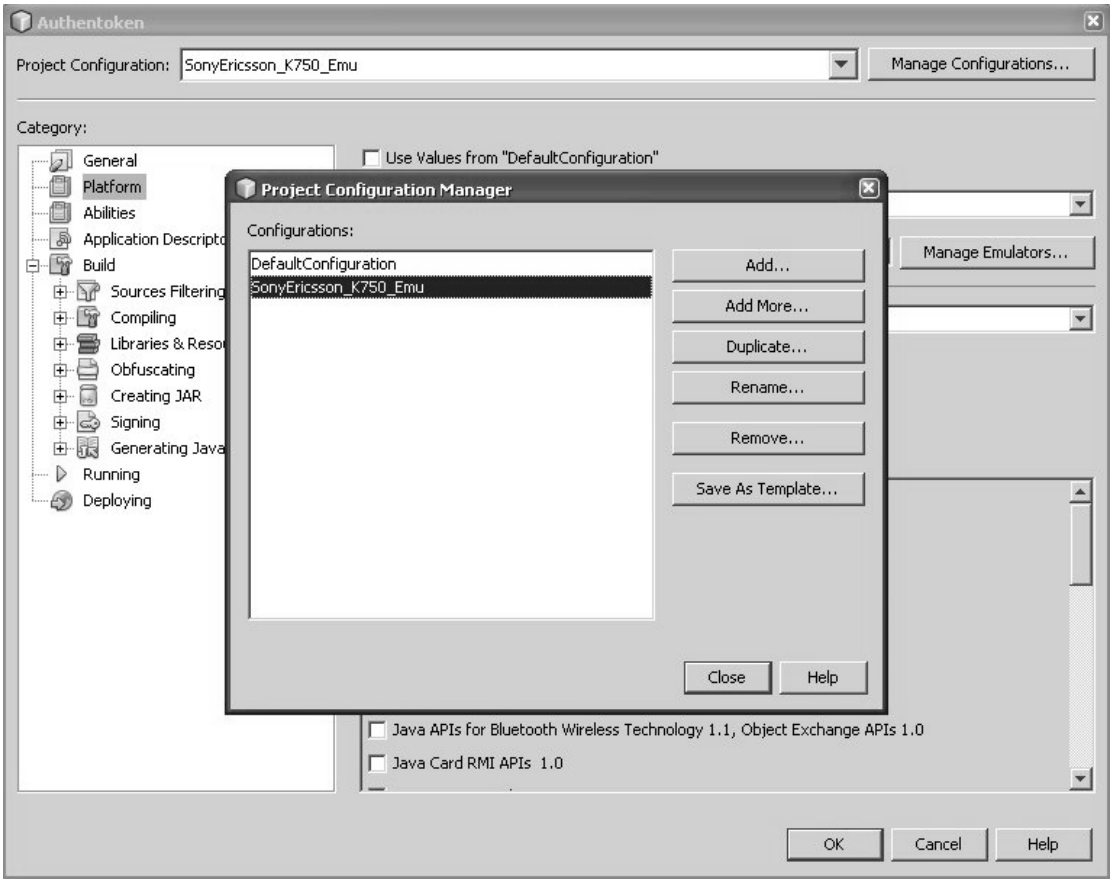
Εικόνα 167: Εκκίνηση του emulator

Στην περίπτωση που θέλουμε να τροποποιήσουμε το επιλεχθέν configuration ή να προσθέσουμε ένα καινούριο, τότε κάνουμε δεξί click στο όνομα του project και επιλέγουμε “Properties”.



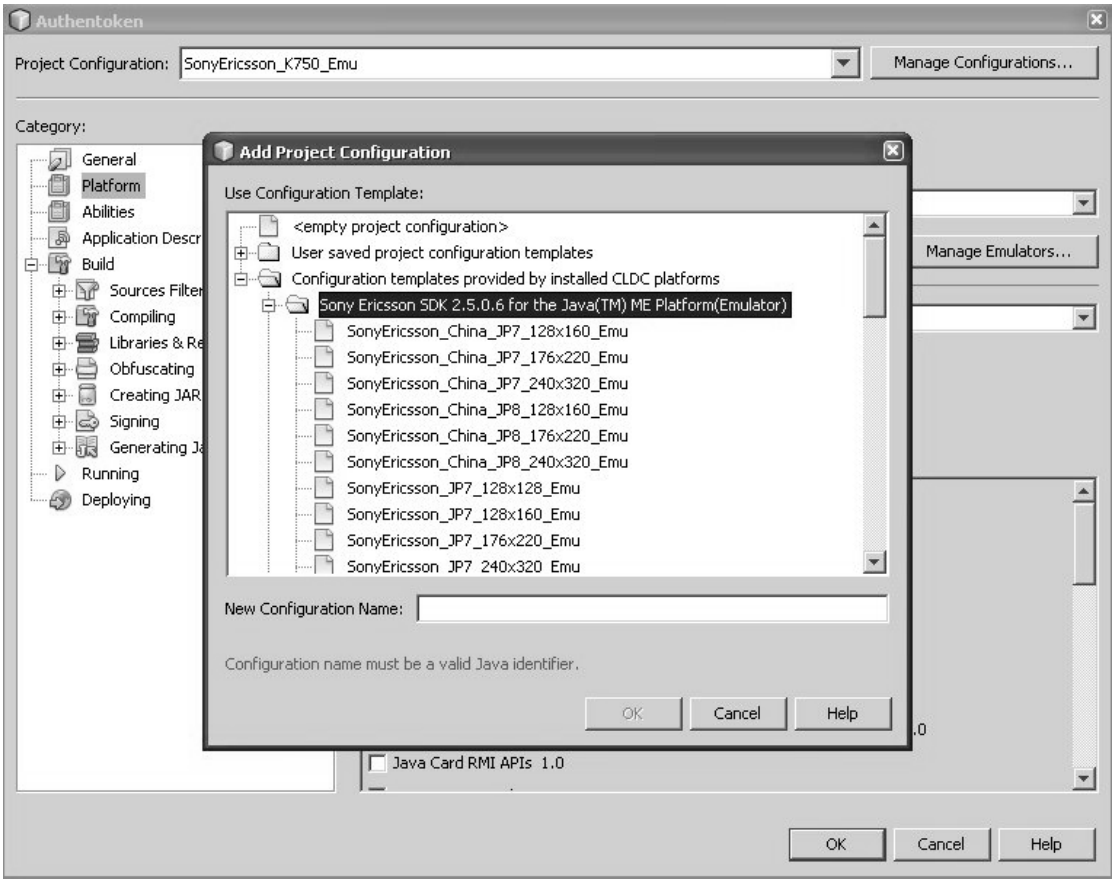
Εικόνα 168: Επιλογή των properties του project

Στο παράθυρο που μας εμφανίζεται επιλέγουμε “Manage Configurations…” και στη συνέχεια, στο νέο παράθυρο που εμφανίζεται πατάμε το πλήκτρο “Add…”.



Εικόνα 169: Προσθήκη νέου Configuration

Σε αυτή τη φάση έχουμε τη δυνατότητα να χρησιμοποιήσουμε το template κάποιου configuration, οπότε από τη λίστα με τα “Configuration templates provided by installed CLDC platforms” μπορούμε να επιλέξουμε ένα από τα διαθέσιμα templates, όπου στην περίπτωση που έχουμε εγκαταστήσει και το Sony Ericsson SDK έχουμε στη διάθεση μας έναν πολύ μεγάλο αριθμό επιλογών.

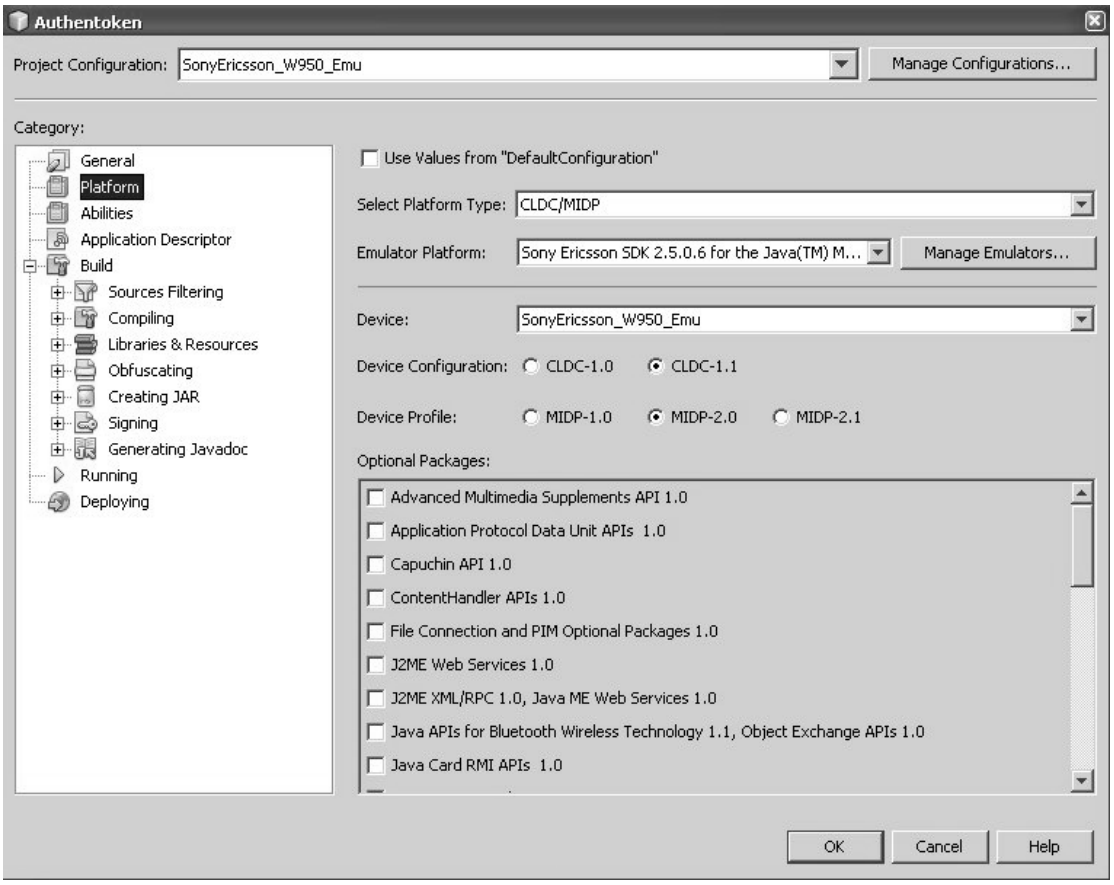


Εικόνα 170: Επιλογή του κατάλληλου template

Αφού έχουμε επιλέξει το κατάλληλο template είμαστε σε θέση να ορίσουμε την έκδοση των CLDC και MIDP με τα οποία θα είναι συμβατό το configuration που θα προκύψει. Ακόμη, μπορούμε να επιλέξουμε αν κάποιο από τα προαιρετικά APIs θα συμμετάσχει στο νέο configuration.

Τα δύο MIDlets που έχουν αναπτυχθεί για τις ανάγκες της τρέχουσας πτυχιακής εργασίας χρησιμοποιούν:

- το CLDC 1.1
- το MIDP 2.0
- το Wireless Messaging API (WMA) 1.1 ή 2.0



Εικόνα 171: Επιλογή CLDC, MIDP και προαιρετικών πακέτων του Configuration

Εγκατάσταση του Sony Ericsson SDK

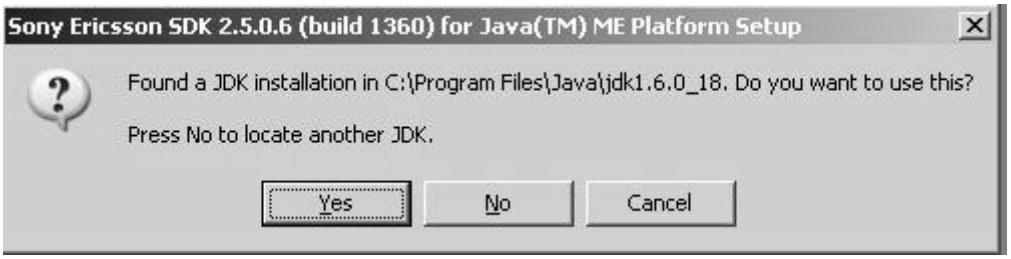
Η εγκατάσταση του SDK που παρέχεται από τη Sony Ericsson δεν είναι υποχρεωτική, αφού κατά την εγκατάσταση του IDE NetBeans έχουν προστεθεί στον υπολογιστή που χρησιμοποιούμε όλα τα απαραίτητα στοιχεία που θα επιτρέψουν την ανάπτυξη και προσομοίωση εφαρμογών Java Micro Edition.

Όμως, μέσα από το συγκεκριμένο πακέτο παρέχεται ένας πολύ μεγάλος αριθμός συσκευών, σε σχέση με αυτές που παρέχονται από το NetBeans, ενώ παράλληλα επιτρέπει την απευθείας προσομοίωση εφαρμογών που αποτελούνται από τα αρχεία JAD/JAR.

Το SDK διατίθεται δωρεάν στην ιστοσελίδα της κοινότητας των προγραμματιστών της Sony Ericsson. Το url που θα μας οδηγήσει στην κατάλληλη σελίδα είναι το εξής: http://developer.sonyericsson.com/wportal/devworld/search-downloads/docstools/sdk?cc=gb&lc=en

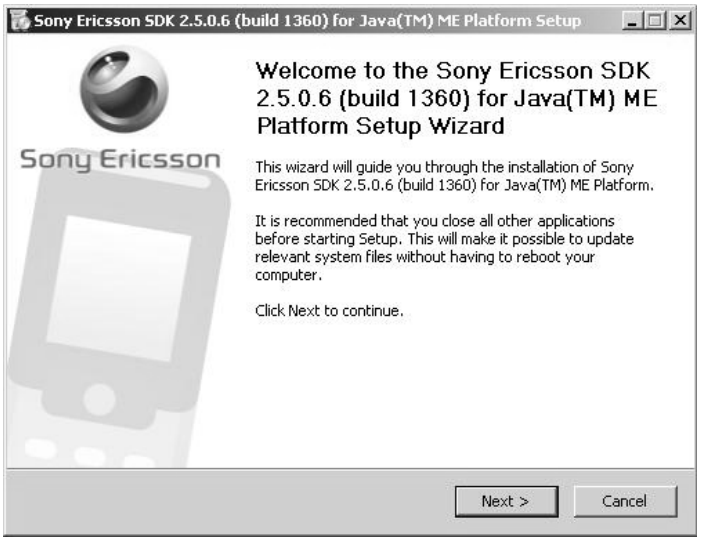
Επειδή στην παραπάνω σελίδα διατίθενται development kits για διάφορες πλατφόρμες, όπως είναι η πλατφόρμα του Android, είμαστε προσεκτικοί και επιλέγουμε το development kit που προορίζεται για την πλατφόρμα Java ME.

Με την εκκίνηση της εγκατάστασης εμφανίζεται στο χρήστη ένα μήνυμα που τον ενημερώνει σχετικά με το Java Development Kit που πρόκειται να χρησιμοποιηθεί.



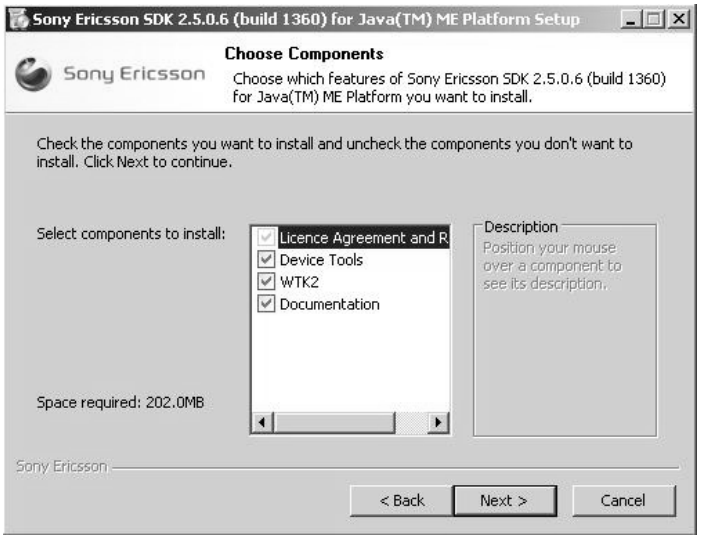
Εικόνα 172: Ενημέρωση του χρήστη σχετικά με το Java Development Kit που πρόκειται να χρησιμοποιηθεί

Αφού βεβαιωθούμε ότι το μονοπάτι εγκατάστασης του Java Development Kit που αναφέρεται είναι σωστό, τότε επιβεβαιώνουμε τη χρησιμοποίηση του από το Sony Ericsson SDK και προχωράμε στην επόμενη οθόνη, όπου στην ουσία ο χρήστης καλωσορίζεται από το SDK.



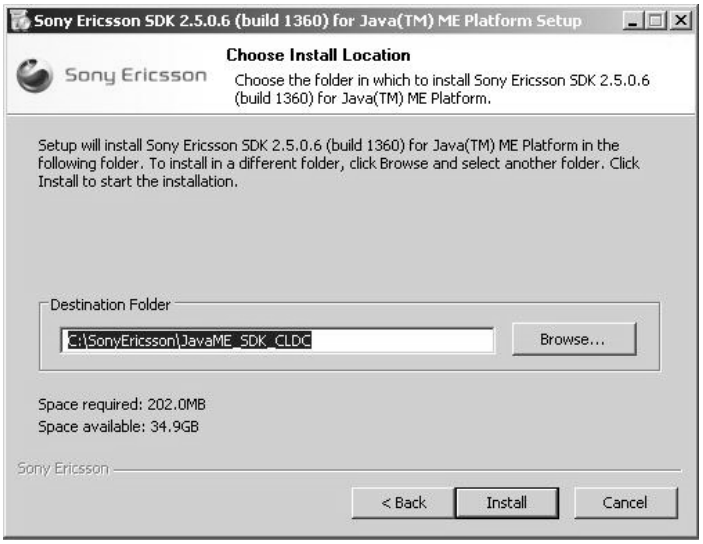
Εικόνα 173: Οθόνη υποδοχής του χρήστη στο Sony Ericsson SDK

Στην οθόνη υποδοχής επιλέγουμε το πλήκτρο “Next” και περνάμε με αυτό τον τρόπο στην επόμενη οθόνη όπου επιλέγουμε τα στοιχεία που πρόκειται να εγκατασταθούν στον υπολογιστή που χρησιμοποιούμε για την ανάπτυξη εφαρμογών Java Micro Edition.



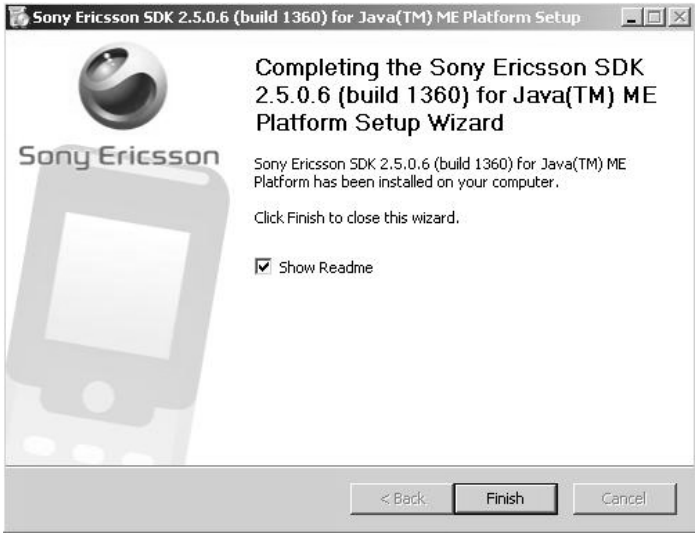
Εικόνα 174: Επιλογή των στοιχείων που πρόκειται να εγκατασταθούν από το Sony Ericsson SDK

Αφού επιβεβαιώσουμε τα στοιχεία που επιθυμούμε να εγκατασταθούν περνάμε στην επιλογή του φακέλου όπου πρόκειται αυτά να τοποθετηθούν.



Εικόνα 175: Επιλογή του φακέλου στον οποίο προορίζονται να εγκατασταθούν τα στοιχεία του Sony Ericsson SDK

Μετά και από αυτό το βήμα ολοκληρώνεται η εγκατάσταση του Sony Ericsson SDK και εμφανίζεται η οθόνη που θα μας οδηγήσει στο κλείσιμο του προγράμματος εγκατάστασης.



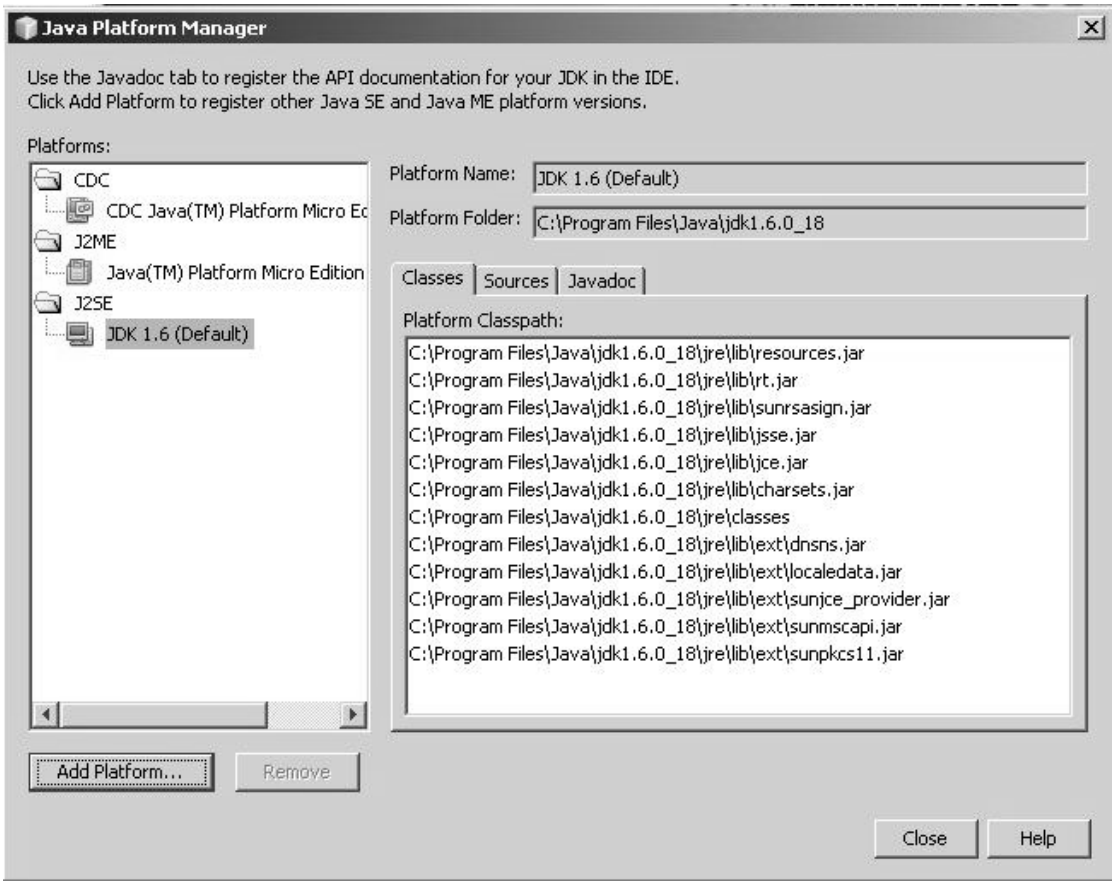
Εικόνα 176: Οθόνη με την οποία ολοκληρώνεται η εγκατάσταση του Sony Ericsson SDK

Ενσωμάτωση του Sony Ericsson SDK στο IDE NetBeans

Μετά από την εγκατάσταση του Sony Ericsson SDK προχωράμε στην ενσωμάτωση του με τις υπόλοιπες πλατφόρμες Java που αλληλεπιδρούν με το IDE NetBeans, έτσι από την μπάρα με το μενού του NetBeans επιλέγουμε “Tools” και από εκεί “Java Platforms”.

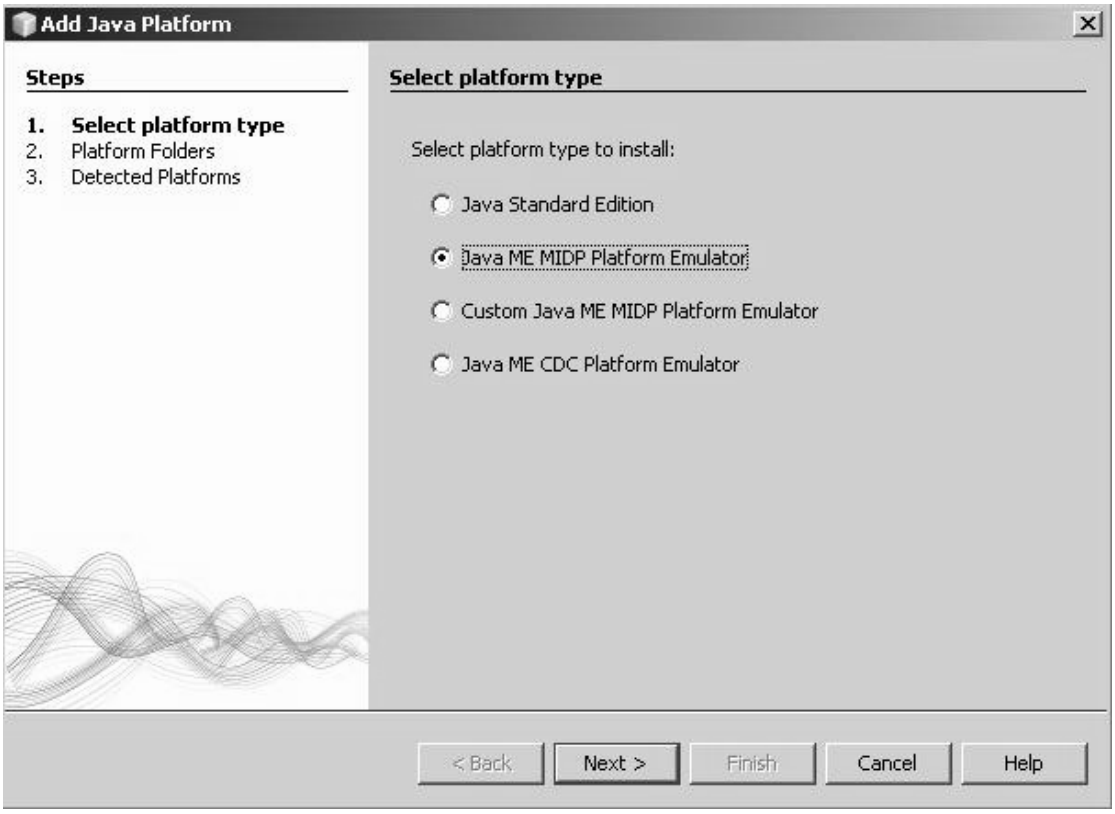
Εικόνα 177: Το πρώτο βήμα για την ενσωμάτωση μιας πλατφόρμας Java στο IDE NetBeans

Αμέσως μετά μας εμφανίζεται η οθόνη μέσω της οποίας διαχειριζόμαστε τις διαθέσιμες πλατφόρμες Java. Στην εικόνα που ακολουθεί παρατηρούμε ότι στις πλατφόρμες Java Micro Edition υπάρχει μόνο η βασική που έχει εγκατασταθεί μαζί με το IDE NetBeans.



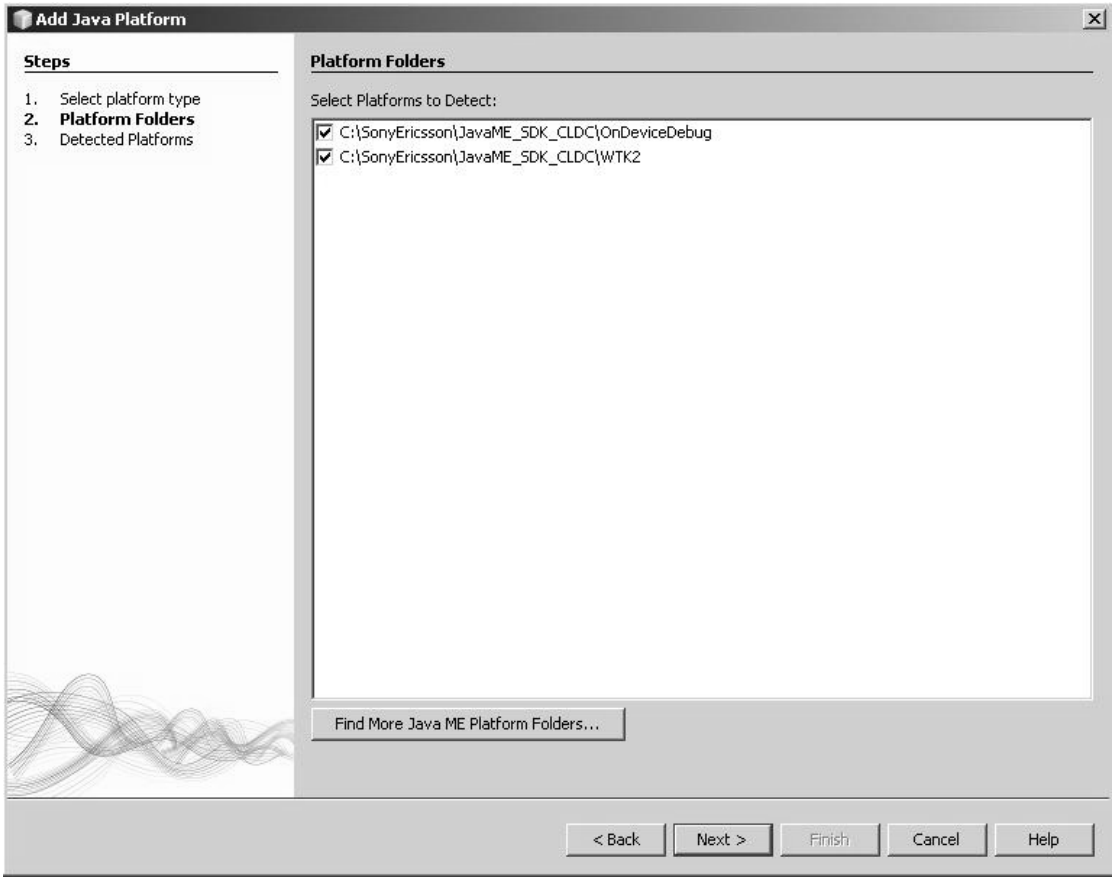
Εικόνα 178: Η διαχείριση των Java Platforms που βρίσκονται ενσωματωμένες στο NetBeans

Για να προσθέσουμε την πλατφόρμα Java που παρέχεται από το SDK της Sony Ericsson επιλέγουμε το πλήκτρο “Add Platform…”, με αυτό τον τρόπο περνάμε στην επόμενη οθόνη, όπου πρέπει να επιλέξουμε τον τύπο της νέας πλατφόρμας που πρόκειται να ενσωματωθεί στο NetBeans.



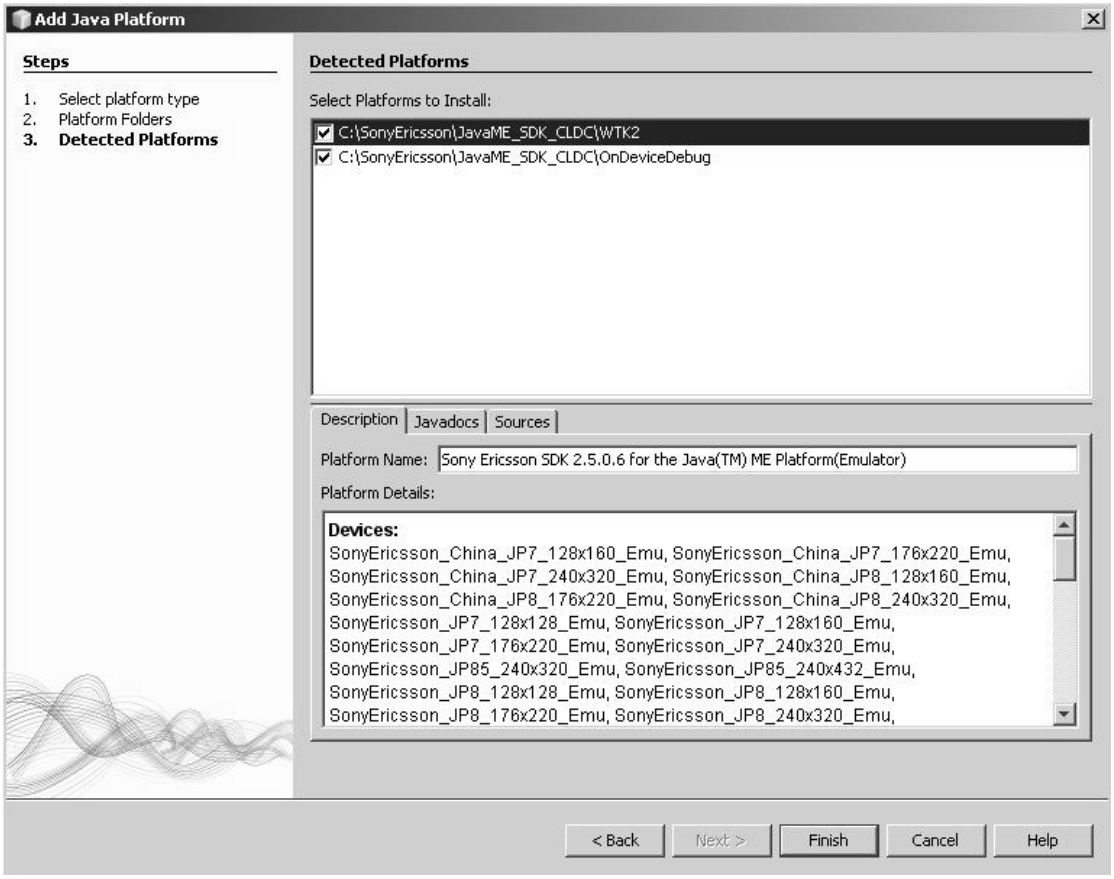
Εικόνα 179: Επιλογή του τύπου της πλατφόρμας που πρόκειται να ενσωματωθεί στο NetBeans

Για τη δική μας περίπτωση επιλέγουμε “Java ME MIDP Platform Emulator” και μετά επιλέγουμε το πλήκτρο “Next”. Από εκεί περνάμε στο επόμενο βήμα, όπου μας δίνεται η δυνατότητα να επιλέξουμε σε ποιους φακέλους να αναζητηθούν πλατφόρμες.



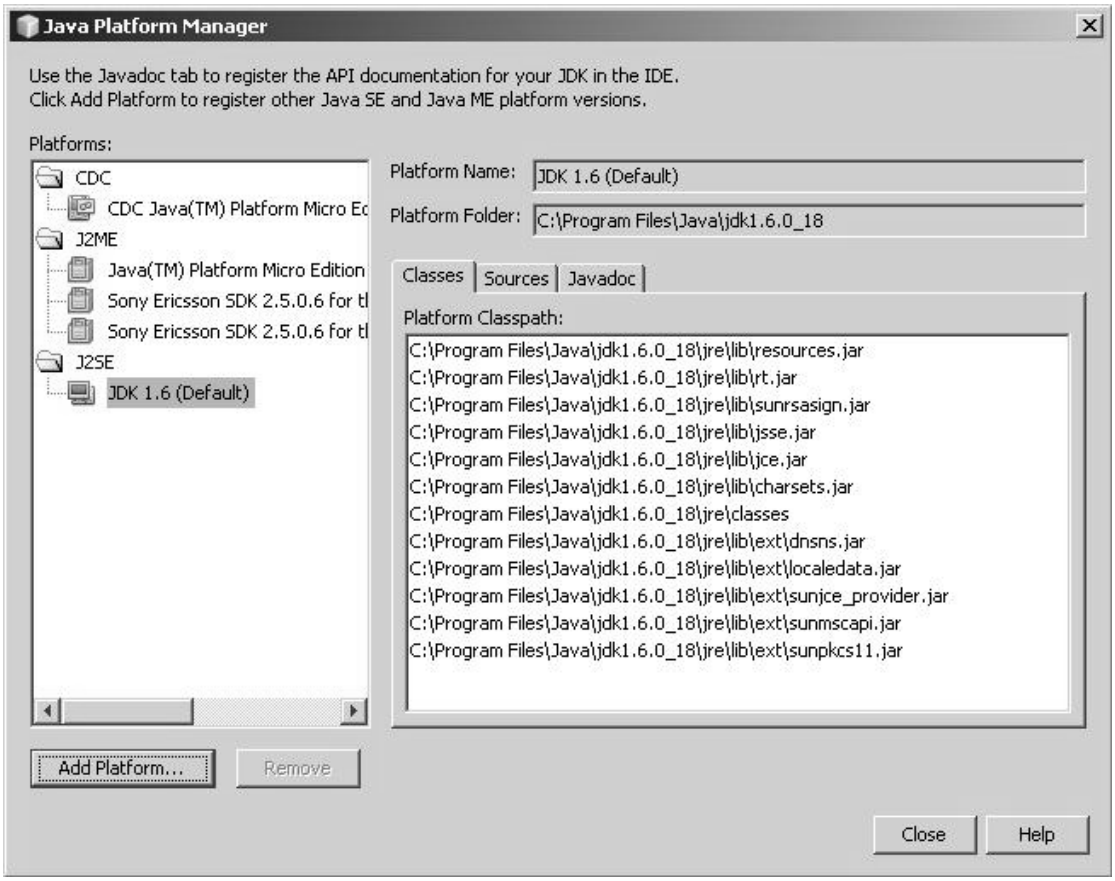
Εικόνα 180: Η επιλογή των φακέλων που πρόκειται να αναζητηθούν νέες πλατφόρμες

Αφού επιλέξουμε τους φακέλους που επιθυμούμε, στην περίπτωση μας και τους δύο όπου έχουν εγκατασταθεί από το SDK της Sony Ericsson, επιλέγουμε το πλήκτρο “Next” και προχωράμε στην ανίχνευση πλατφόρμων εντός των επιλεγμένων φακέλων. Το αποτέλεσμα αυτής της ενέργειας εμφανίζεται στην εικόνα που ακολουθεί, όπου παρουσιάζονται οι ανιχνευθείσες πλατφόρμες και εμφανίζονται οι συσκευές που υποστηρίζονται από αυτές.



Εικόνα 181: Εμφάνιση των πλατφόρμων που ανιχνεύθηκαν από το NetBeans

Σε αυτό το σημείο έχουμε τη δυνατότητα να επιλέξουμε ποιες από τις ανιχνευθείσες πλατφόρμες θα εγκατασταθούν και να ολοκληρώσουμε τη διαδικασία επιλέγοντας το πλήκτρο “Finish”. Μετά από αυτή την επιλογή επιστρέφουμε ξανά στην οθόνη διαχείρισης των πλατφόρμων Java, όπου τώρα παρατηρούμε ότι στις πλατφόρμες J2ME συμπεριλαμβάνεται το ζεύγος πλατφόρμων της Sony Ericsson που μόλις εγκαταστήσαμε.



Εικόνα 182: Η διαχείριση των πλατφόρμων Java, όπου τώρα εμφανίζονται οι νέες πλατφόρμες που εγκαταστήσαμε

Στην οθόνη της διαχείρισης των πλατφόρμων Java επιλέγουμε το πλήκτρο “Close” και με αυτό τον τρόπο κλείνουμε την οθόνη και ολοκληρώνουμε την ενσωμάτωση των πλατφόρμων στο περιβάλλον ανάπτυξης εφαρμογών NetBeans.

Εκτέλεση MIDlets μέσω του Sony Ericsson SDK

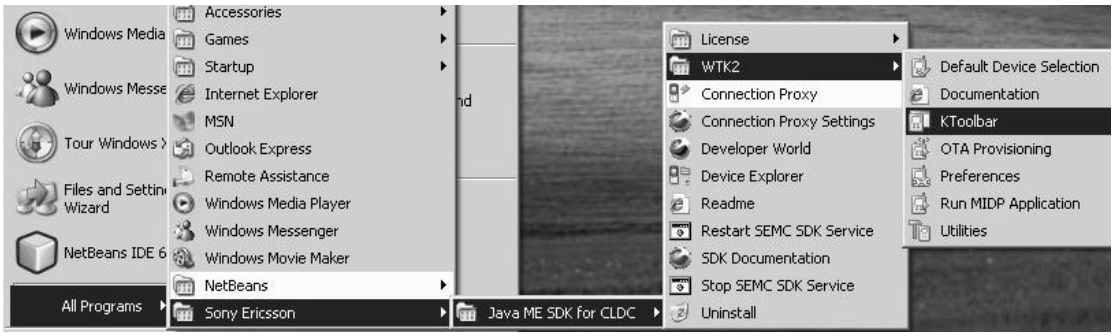
Μετά από την ενσωμάτωση του Sony Ericsson SDK στο περιβάλλον του NetBeans έχουμε τη δυνατότητα επιλογής ανάμεσα από πάρα πολλές συσκευές και να πραγματοποιήσουμε δοκιμές όσων εφαρμογών δημιουργούμε.

Υπάρχει όμως και η δυνατότητα να εκτελέσουμε ένα έτοιμο MIDlet, μόνο με τα αρχεία JAD/JAR, απευθείας από το Sony Ericsson SDK, σε αντίθεση με το NetBeans όπου θα χρειαζόταν η δημιουργία ενός εξ ολοκλήρου νέου project που θα περιέχει και τον πηγαίο κώδικα της εφαρμογής.

Με αυτό τον τρόπο, στην περίπτωση που θα θέλαμε να προσομοιώσουμε τη λειτουργία της εφαρμογής που επιτρέπει την ασφαλή ανταλλαγή σύντομων μηνυμάτων, θα εξοικονομούσαμε αρκετό χρόνο απλά εισάγοντας τα αρχεία JAD/JAR

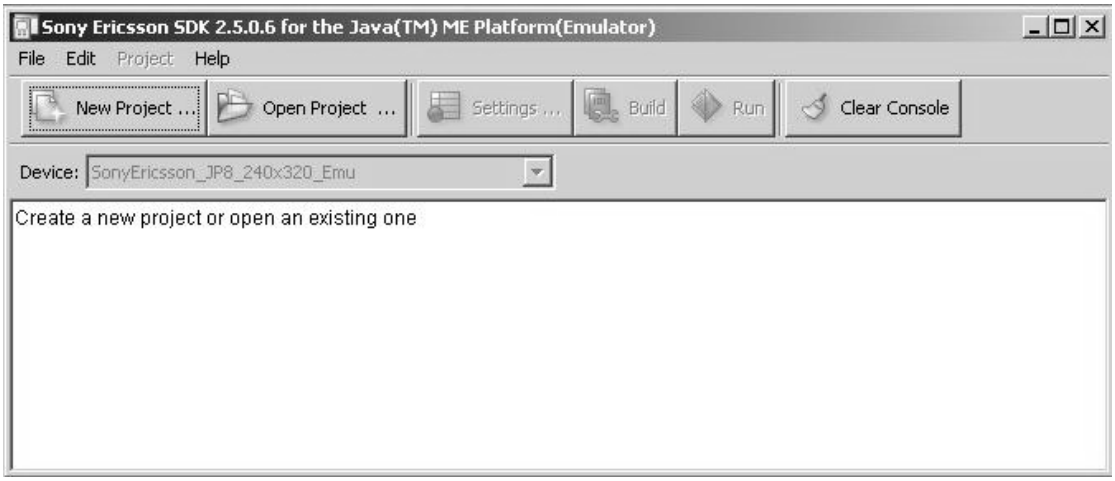
που βρίσκονται στο συνοδευτικό δίσκο απ' ότι θα χρειαζόταν με τη δημιουργία ενός νέου project μέσα από το περιβάλλον του IDE NetBeans.

Για να ξεκινήσουμε τη διαδικασία με την οποία θα εκτελέσουμε ένα MIDlet μέσω του Sony Ericsson SDK πρέπει να εκκινήσουμε το πρόγραμμα "KToolbar", το οποίο εντοπίζεται όταν από τη μπάρα των Windows ακολουθήσουμε τη διαδρομή "Start -> Programs -> Sony Ericsson -> Java ME SDK for CLDC -> WTK2 -> KToolbar".



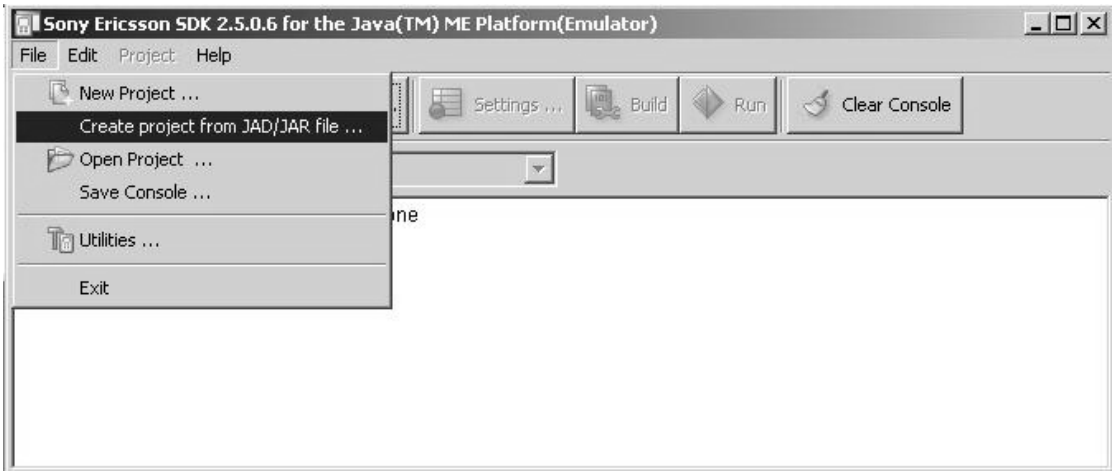
Εικόνα 183: Εντοπισμός του προγράμματος KToolbar

Μετά την εκκίνηση του προγράμματος KToolbar εμφανίζεται μια κονσόλα όπου, όπως φαίνεται και στην εικόνα που ακολουθεί, μπορούμε να δημιουργήσουμε ένα νέο project ή να ανοίξουμε ένα ήδη υπάρχον.



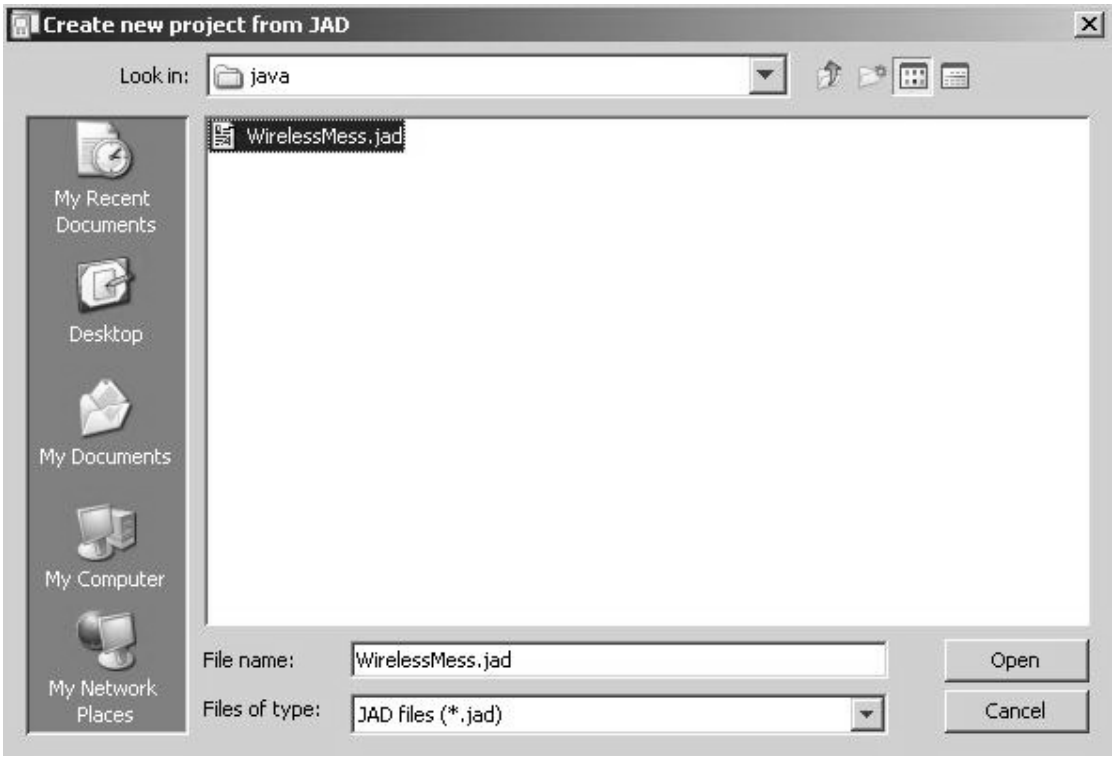
Εικόνα 184: Η κονσόλα που εμφανίζεται από το πρόγραμμα KToolbar

Για να εκτελέσουμε εν τάχει ένα MIDlet που ήδη υπάρχει θα πάμε στη μπάρα του μενού, όπου αρχικά επιλέγουμε "File" και στη συνέχεια επιλέγουμε "Create project from JAD/JAR file…". Με αυτό τον τρόπο χρησιμοποιούμε τα JAD και JAR του MIDlet για να δημιουργήσουμε ένα project.



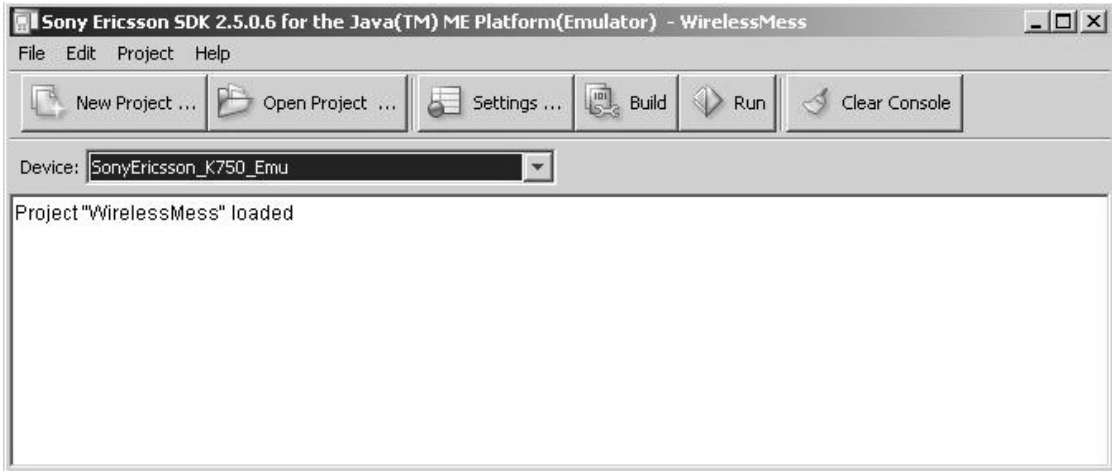
Εικόνα 185: Η επιλογή για τη δημιουργία ενός project από JAD και JAR που ήδη υπάρχουν

Στη συνέχεια καλούμαστε, με τη βοήθεια ενός επιλογέα αρχείων, να επιλέξουμε το JAD που επιθυμούμε να χρησιμοποιήσουμε για τη δημιουργία του νέου project.



Εικόνα 186: Η επιλογή του JAD από το οποίο θα δημιουργηθεί το project

Στη συνέχεια, η κονσόλα του KToolbar μας ενημερώνει ότι το project έχει φορτωθεί, οπότε εμείς από το drop-down menu με την ετικέτα “Device” επιλέγουμε ποια συσκευή θα χρησιμοποιήσουμε κατά τη δοκιμή του project. Το μόνο που μένει είναι να επιλέξουμε το πλήκτρο “Run” και να ξεκινήσει η προσομοίωση του MIDlet που έχουμε επιλέξει.



Εικόνα 187: Η επιλογή της συσκευής που πρόκειται να χρησιμοποιηθεί κατά την προσομοίωση του MIDlet

Εγκατάσταση του MySQL Server

Το RDBMS της εταιρείας MySQL, που φέρει τον τίτλο MySQL Server, παρέχεται δωρεάν από το url: www.mysql.com/downloads/mysql/

Το πρόγραμμα εγκατάστασης διατίθεται σε διάφορες εκδόσεις, όμως εμείς προτιμήσαμε εκείνη που αναφέρεται ως Windows Installer File και έχει την κατάληξη αρχείου .msi.

Αφού λάβουμε το συγκεκριμένο αρχείο κάνουμε ένα διπλό κλικ και έτσι ξεκινάμε την εγκατάσταση του RDBMS.

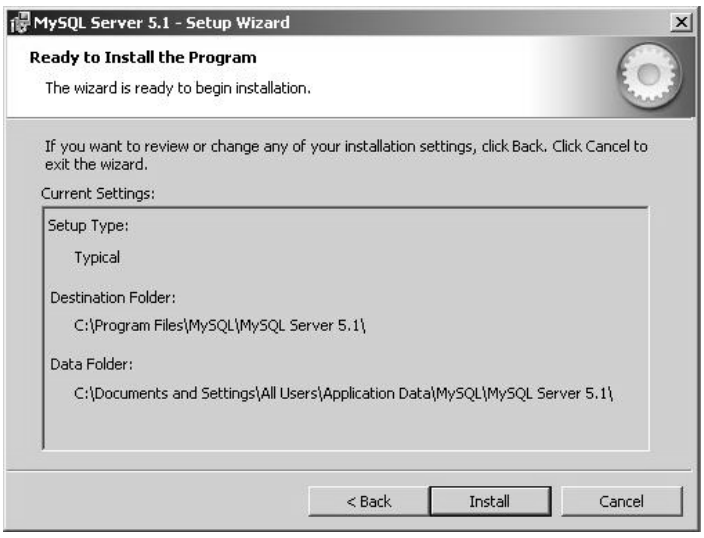
Το πρώτο παράθυρο που εμφανίζεται μας καλωσορίζει στον οδηγό εγκατάστασης του MySQL Server.

Εικόνα 188: Καλωσόρισμα στην εγκατάσταση του MySQL Server

Στη συνέχεια εμφανίζεται ένα παράθυρο στο οποίο επιλέγουμε τον τύπο της εγκατάστασης που πρόκειται να πραγματοποιηθεί. Η τυπική εγκατάσταση είναι προεπιλεγμένη και εμείς αφήνουμε την επιλογή ως έχει.

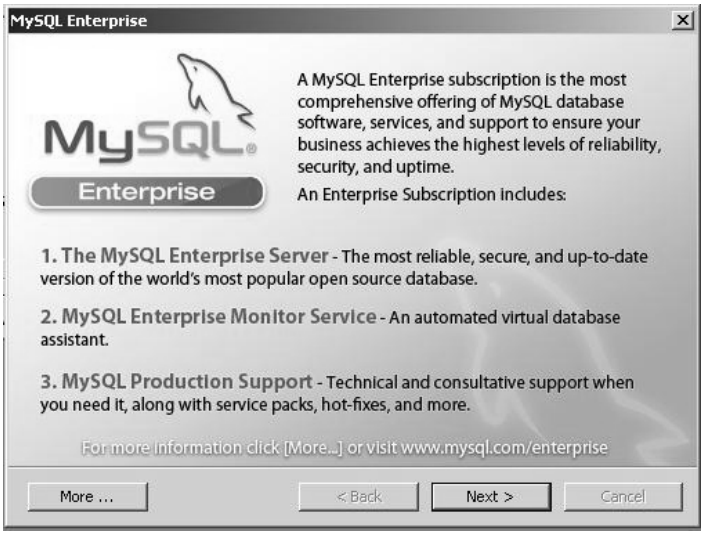
Εικόνα 189: Επιλογή του τύπου εγκατάστασης του MySQL Server

Το επόμενο παράθυρο μας πληροφορεί για το φάκελο που θα τοποθετηθούν τα αρχεία εγκατάστασης και το φάκελο όπου θα αποθηκεύονται τα δεδομένα που θα διαχειρίζεται το RDBMS της MySQL.

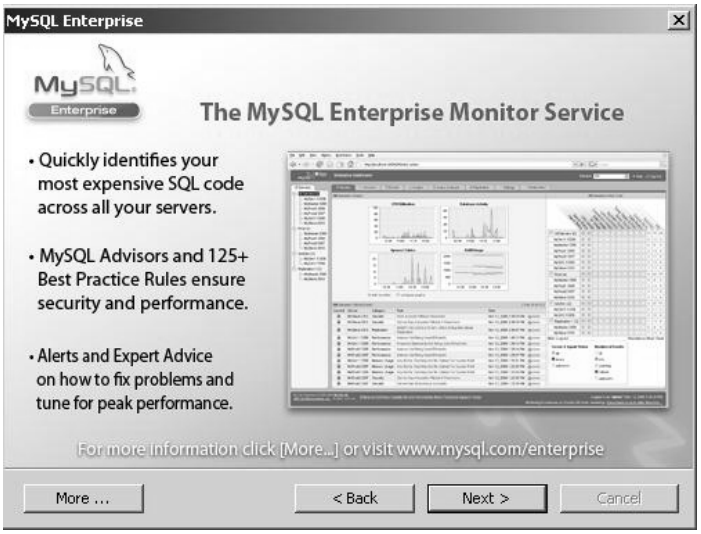


Εικόνα 190: Εμφάνιση φακέλων με τους οποίους σχετίζεται ο MySQL Server

Καθώς πραγματοποιείται η εγκατάσταση μας εμφανίζονται δύο παράθυρα με διαφημίσεις άλλων υπηρεσιών της εταιρείας MySQL. Σε κάθε ένα από αυτά είμαστε αναγκασμένοι να πατήσουμε το πλήκτρο Next ούτως ώστε να συνεχιστεί η διαδικασία της εγκατάστασης.



Εικόνα 191: Το πρώτο διαφημιστικό παράθυρο της MySQL



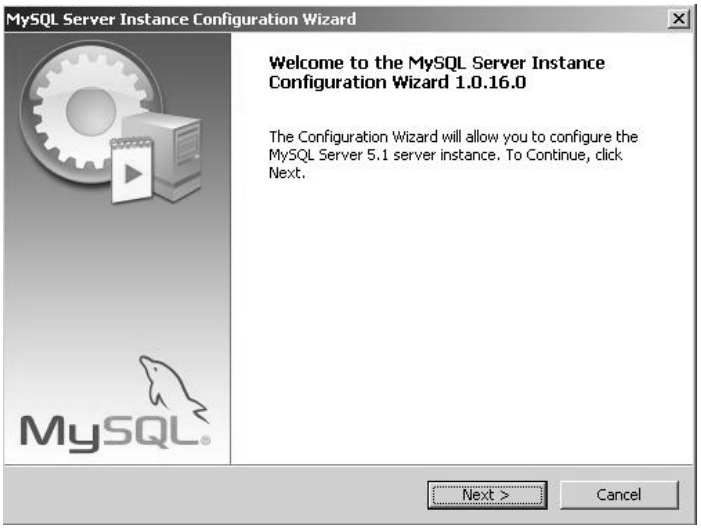
Εικόνα 192: Το δεύτερο διαφημιστικό παράθυρο της MySQL

Μετά απ' όλα αυτά, και αν δεν έχει υπάρξει κάποιο πρόβλημα, η εγκατάσταση του MySQL Server έχει ολοκληρωθεί με επιτυχία. Το παράθυρο που εμφανίζεται μας ενημερώνει γι' αυτό το γεγονός και παράλληλα μας δίνει τη δυνατότητα να ρυθμίσουμε τώρα το MySQL Server. Καλό είναι να ρυθμίσουμε αμέσως το MySQL Server, οπότε δεν αλλάζουμε κάτι στο check box που αφορά το συγκεκριμένο γεγονός.

Εικόνα 193: Ολοκλήρωση της εγκατάστασης του MySQL Server

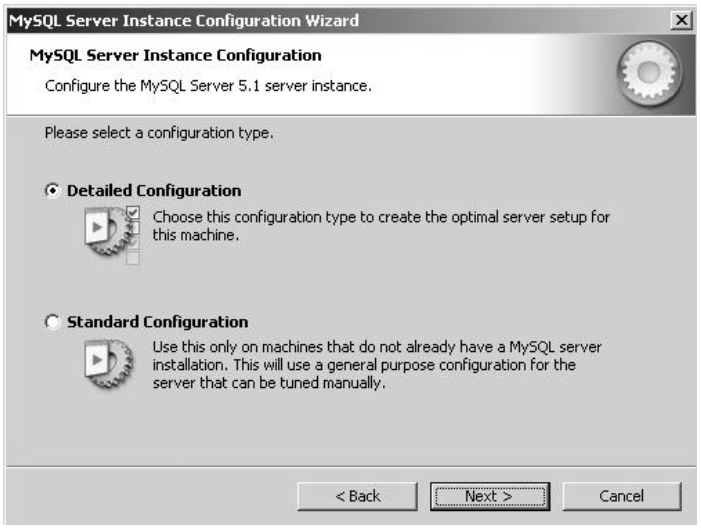
Ρύθμιση του MySQL Server

Το πρώτο παράθυρο που εμφανίζεται μας καλωσορίζει στον οδηγό ρύθμισης του MySQL Server και μας ενημερώνει ότι αν θέλουμε να συνεχίσουμε πρέπει να επιλέξουμε το πλήκτρο Next.



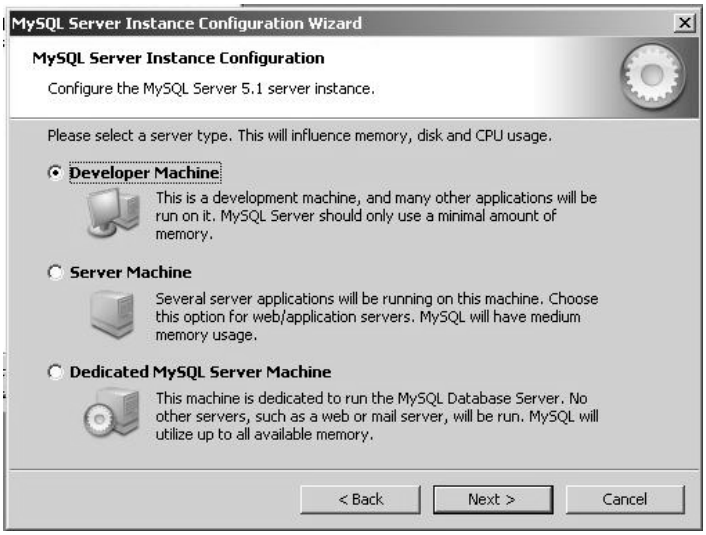
Εικόνα 194: Καλωσόρισμα στον οδηγό ρύθμισης του MySQL Server

Το επόμενο παράθυρο μας δίνει τη δυνατότητα επιλογής ανάμεσα σε τυπική ρύθμιση ή αναλυτική ρύθμιση. Επιλέγουμε το Detailed Configuration και συνεχίζουμε στο επόμενο παράθυρο.



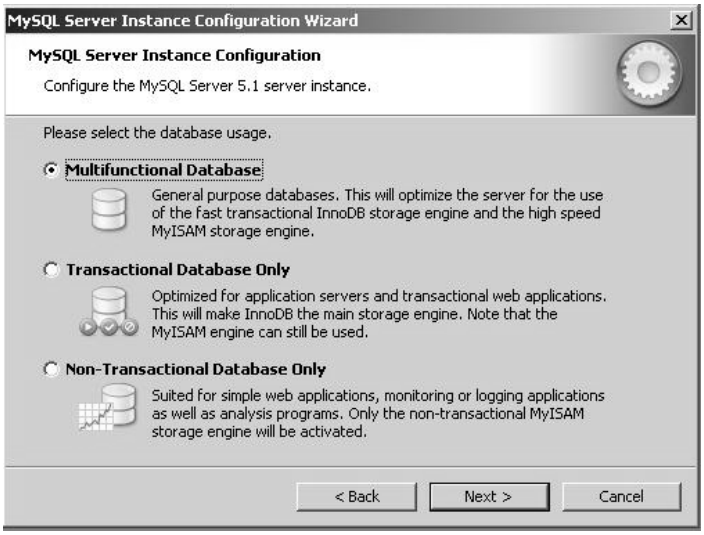
Εικόνα 195: Επιλογή του τύπου ρύθμισης του MySQL Server

Στη συνέχεια εμφανίζεται ένα παράθυρο όπου επιλέγουμε τον τύπο του MySQL Server που πρόκειται να εγκατασταθεί. Στη δική μας περίπτωση μας αρκεί η πρώτη επιλογή, που ονομάζεται Developer Machine, αφού η διαδικτυακή εφαρμογή μας έχει περισσότερο εκπαιδευτικό σκοπό.



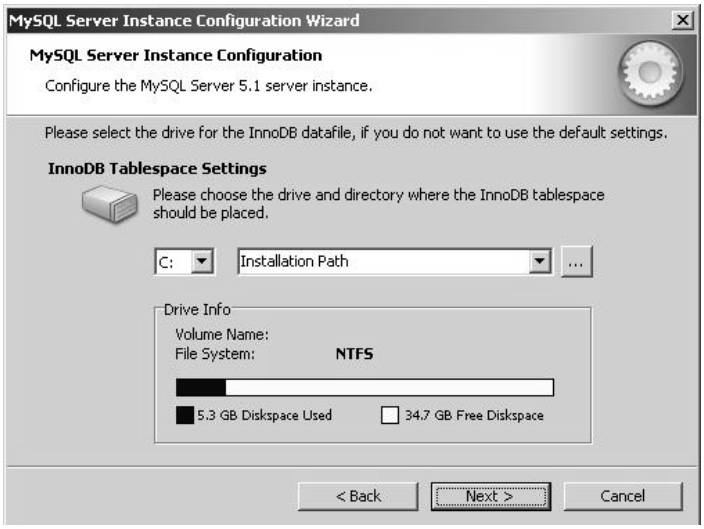
Εικόνα 196: Επιλογή του τύπου του MySQL Server

Αμέσως μετά επιλέγουμε τη χρήση που θα έχουν οι βάσεις δεδομένων που πρόκειται να δημιουργηθούν. Για τη δική μας περίπτωση αφήνουμε την πρώτη επιλογή, με την ονομασία Multifunctional Database, μέσω της οποίας θα δημιουργούνται βάσεις δεδομένων με πολλαπλές λειτουργίες.



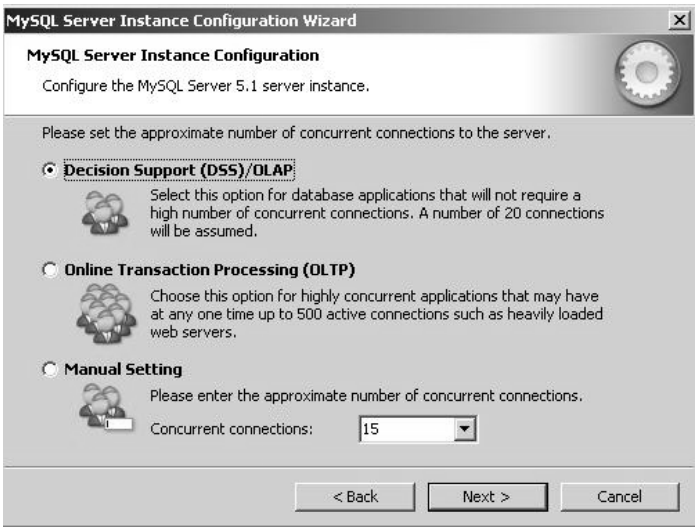
Εικόνα 197: Επιλογή της χρήσης που θα έχουν οι βάσεις δεδομένων

Ακολούθως, μας δίνεται η επιλογή να αλλάξουμε την τοποθεσία όπου θα αποθηκεύονται τα αρχεία που διαχειρίζεται η μηχανή αποθήκευσης InnoDB. Εμείς δεν αλλάξαμε κάτι και αφήσαμε την επιλογή που προϋπήρχε.



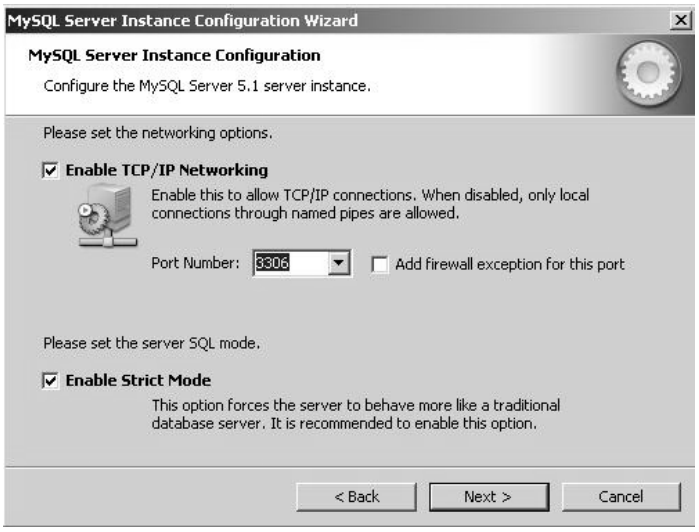
Εικόνα 198: Επιλογή της τοποθεσίας αποθήκευσης των δεδομένων του InnoDB

Στη συνέχεια επιλέγουμε τον αριθμό των ταυτόχρονων συνδέσεων που ενδέχεται να πραγματοποιούνται στο MySQL Server. Αφήνουμε την πρώτη επιλογή, με την οποία υποτίθεται ότι πραγματοποιούνται μέχρι και 20 ταυτόχρονες συνδέσεις.



Εικόνα 199: Επιλογή του αριθμού των ταυτόχρονων συνδέσεων

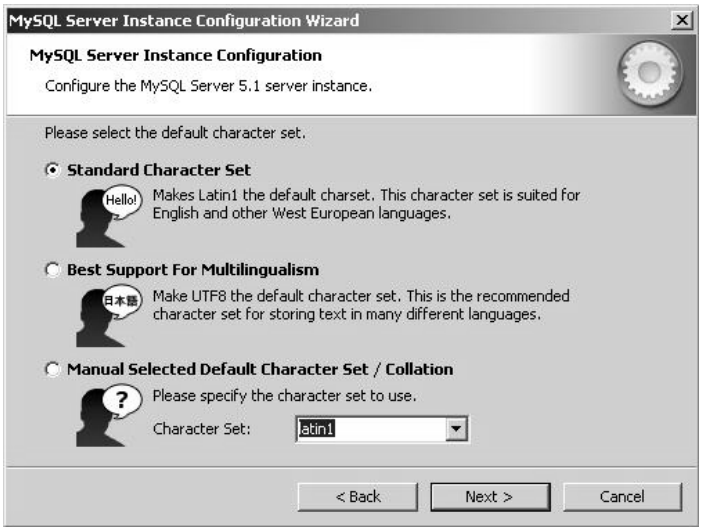
Στο επόμενο παράθυρο ρυθμίζονται οι επιλογές που αφορούν η σύνδεση που θα πραγματοποιείται μεταξύ του Client και του MySQL Server. Στη δική μας περίπτωση δεν αλλάξαμε τίποτα, έτσι αφήσαμε ενεργοποιημένες και τις δύο επιλογές που υπάρχουν μαζί με το ήδη ρυθμισμένο port που έχει την τιμή 3306.



Εικόνα 200: Ρύθμιση των επιλογών που αφορούν τη σύνδεση με το MySQL Server

Στο παράθυρο που ακολουθεί ρυθμίζουμε το default σύνολο χαρακτήρων που θα χρησιμοποιείται από κάθε βάση δεδομένων.

Από την πλευρά μας δεν πραγματοποιήθηκε κάποια αλλαγή και παρέμεινε επιλεγμένο το Default Character Set, με το οποίο το σύνολο χαρακτήρων Latin1 τοποθετείται ως πρώτο σε χρήση.



Εικόνα 201: Επιλογή του default character set

Αμέσως μετά μπορούμε να επιλέξουμε αν θέλουμε να εγκαταστήσουμε το MySQL Server και ως μια από τις υπηρεσίες των Windows.

Ακόμη, την ίδια στιγμή μπορούμε να τοποθετήσουμε στη μεταβλητή συστήματος PATH το φάκελο bin της εγκατάστασης του MySQL Server και να δώσουμε έτσι τη δυνατότητα όταν είμαστε σε command line να καλούμε άμεσα όσα εκτελέσιμα αρχεία βρίσκονται στο συγκεκριμένο φάκελο.

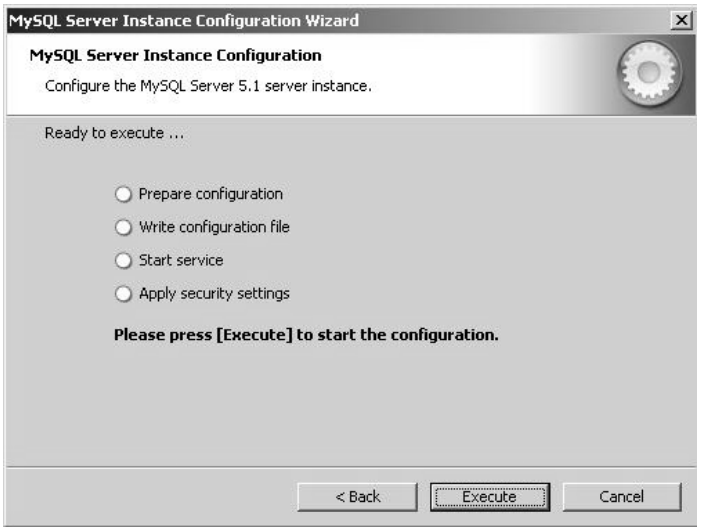
Εικόνα 202: Εγκατάσταση του MySQL Server σαν Windows Service και προσθήκη φακέλου bin στη μεταβλητή PATH

Στο ακόλουθο παράθυρο ρυθμίζονται λεπτομέρειες που αφορούν την ασφάλεια του MySQL Server. Έτσι επιλέγουμε αν θα δημιουργήσουμε λογαριασμό για χρήστη root, οπότε θα πρέπει να εισάγουμε και τον αντίστοιχο κωδικό που θα χρησιμοποιείται από το συγκεκριμένο χρήστη, ή θα επιτρέψουμε την ανώνυμη σύνδεση με το MySQL Server.

Θεωρείται ότι είναι καλύτερη η ρύθμιση του λογαριασμού ενός χρήστη root, οπότε αυτό πράξαμε και εμείς από την πλευρά μας.

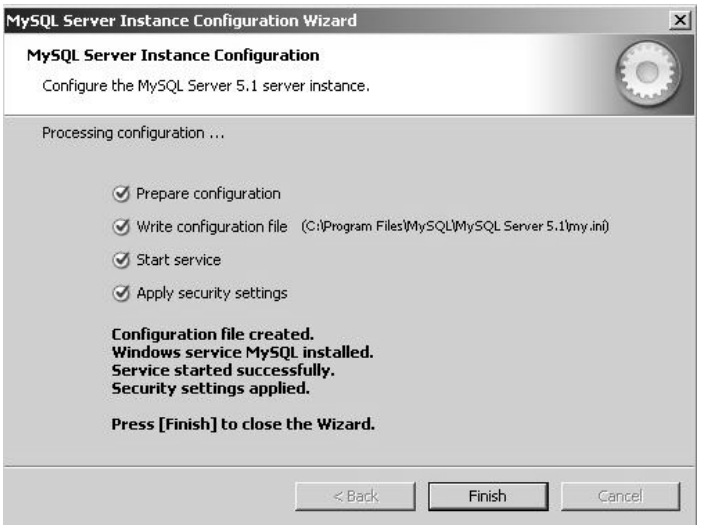
Εικόνα 203: Δημιουργία χρήστη root ή ανώνυμου χρήστη

Αφού έχουν γίνει όλα τα παραπάνω βήματα φτάνουμε σε ένα παράθυρο όπου απαιτείται η έγκριση μας για να δημιουργηθεί ένα νέο configuration file σύμφωνα με όλες τις επιλογές που κάναμε στα προηγούμενα παράθυρα.



Εικόνα 204: Παράθυρο για τη δημιουργία νέου configuration file

Πατώντας το πλήκτρο execute ξεκινά η διαδικασία δημιουργίας του νέου configuration file και το νέο παράθυρο που εμφανίζεται μας ενημερώνει για την πορεία που έχει η συγκεκριμένη ενέργεια, όπως γίνεται με την παρακάτω εικόνα που μας ενημερώνει ότι το configuration file δημιουργήθηκε με απόλυτη επιτυχία.



Εικόνα 205: Επιτυχημένη δημιουργία νέου configuration file

Εγκατάσταση του Apache Tomcat Server

Ο application server Apache Tomcat διατίθεται δωρεάν σε τρεις διαφορετικές εκδόσεις, ανάλογα με το Java Virtual Machine που έχουμε εγκατεστημένο στον υπολογιστή μας.

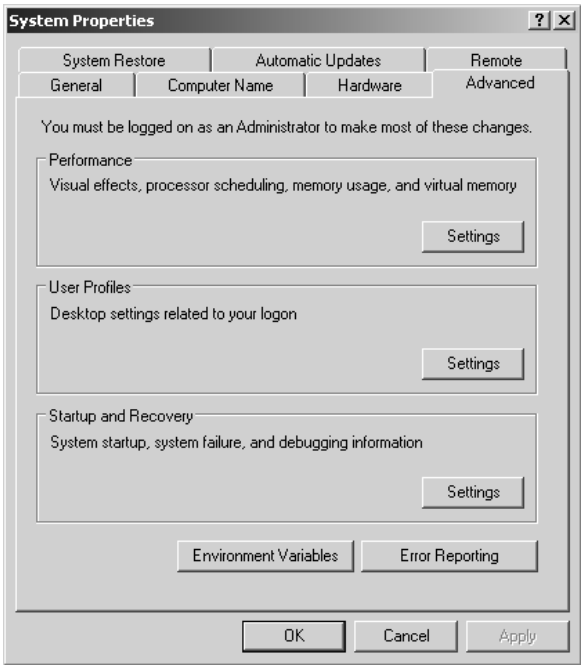
Από την ιστοσελίδα http://tomcat.apache.org/whichversion.html μπορούμε να προσδιορίσουμε ποια από τις διαθέσιμες εκδόσεις ταιριάζει για το δικό μας υπολογιστή.

Στη συνέχεια επιλέγουμε ποιο από τα διαθέσιμα binary distributions του Tomcat θα κατεβάσουμε. Για τη δική μας περίπτωση θα προτιμήσουμε το συμπιεσμένο αρχείο .zip και θα αποφύγουμε το Windows Service Installer.

Η εγκατάσταση του Tomcat γίνεται με την αποσυμπίεση του συμπιεσμένου αρχείου σε ένα φάκελο που εμείς επιθυμούμε και απομένει να ρυθμίσουμε ορισμένες παραμέτρους που αφορούν τη λειτουργία του Application Server.

Σε πρώτη φάση, πρέπει να ορίσουμε τρείς νέες οικουμενικές μεταβλητές στο λειτουργικό σύστημα που χρησιμοποιούμε.

Αν χρησιμοποιούμε μια από τις εκδόσεις του λειτουργικού συστήματος των Microsoft Windows, τότε για τη διαχείριση των μεταβλητών θα πρέπει να επιλέξουμε τα properties του My Computer και από εκεί να περάσουμε στο tab Advanced όπου και θα βρούμε το πλήκτρο Environment Variables.

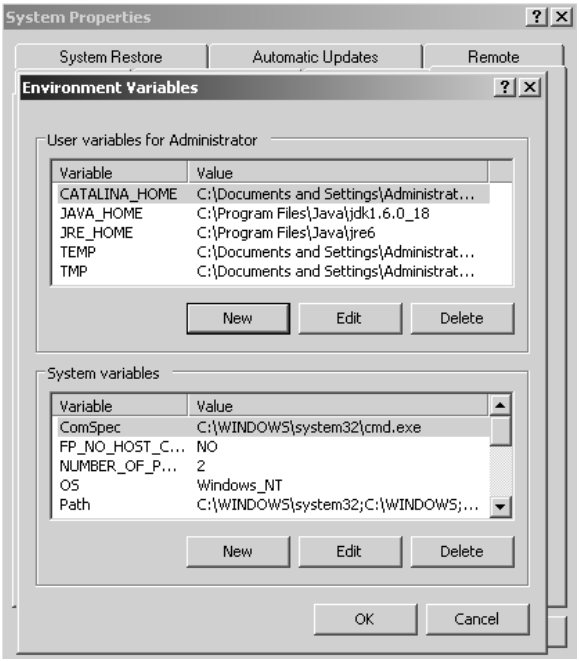


Εικόνα 206: Το tab Advanced όπου βρίσκεται η διαχείριση των μεταβλητών του λειτουργικού συστήματος

Οι τρείς μεταβλητές που χρειαζόμαστε είναι οι εξής:

- JAVA_HOME, στην οποία αντιστοιχεί ο φάκελος όπου έχει εγκατασταθεί το Java Development Kit.
- JRE_HOME, στην οποία αντιστοιχεί ο φάκελος που βρίσκεται το Java Runtime Environment.
- CATALINA_HOME, όπου υποδεικνύεται η τοποθεσία που έχει αποσυμπιεστεί το αρχείο που περιέχει τον Apache Tomcat.

Για τον ορισμό της κάθε μεταβλητής πρέπει να χρησιμοποιήσουμε το κατάλληλο όνομα από αυτά που αναφέραμε νωρίτερα και σε κάθε value να αντιγράψουμε το αντίστοιχο path.



Εικόνα 207: Μετά από τον ορισμό των τριών μεταβλητών

Στη συνέχεια, πρέπει να ρυθμίσουμε τον Tomcat, οπότε με έναν επεξεργαστή κειμένου ανοίγουμε το αρχείο που βρίσκεται στη διαδρομή:

Φάκελος_Εγκατάστασης_Apache_Tomcat/conf/server.xml

Το πρώτο στοιχείο που τροποποιούμε είναι η θύρα στην οποία προσδένεται ο Apache Tomcat, όπου εξ ορισμού για τον συγκεκριμένο application server είναι 8080, όμως είναι προτιμότερο να την αλλάξουμε στην τιμή 80 που είναι εκ των πραγμάτων δεσμευμένη για application servers.

Έτσι, στο αρχείο server.xml εντοπίζουμε την ετικέτα Connector που έχει μορφή παρόμοια με την ακόλουθη:

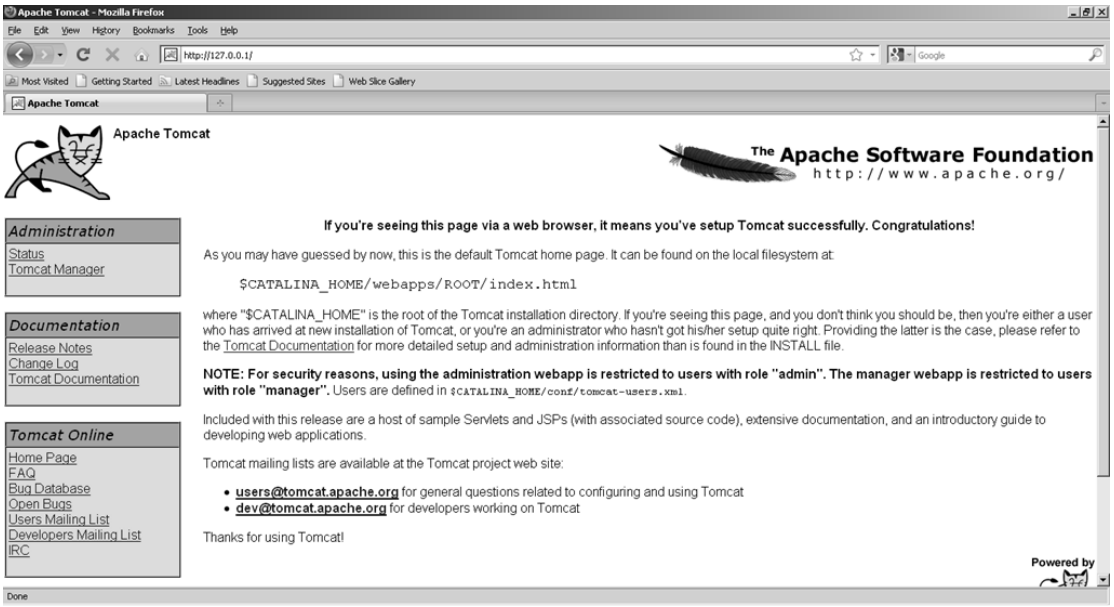
```
<Connector port="8080" protocol="HTTP/1.1"
     connectionTimeout="20000"
     redirectPort=”8443” />
```

Στην ετικέτα του Connector του Apache Tomcat βρίσκουμε και το στοιχείο redirectPort που υποδεικνύει τη θύρα στην οποία ανακατευθύνονται τα requests που αφορούν το πρωτόκολλο SSL-TLS, οπότε και εδώ αλλάζουμε την τιμή του στοιχείου με τη default του πρωτοκόλλου που είναι 443. Έτσι, η τελική ετικέτα θα μοιάζει με την ακόλουθη, με τις αλλαγές που κάναμε να τονίζονται με έντονους χαρακτήρες.

```
<Connector port="80" protocol="HTTP/1.1"
     connectionTimeout="20000"
     redirectPort=”443” />
```

Μετά από όλα τα παραπάνω μπορούμε να επιβεβαιώσουμε ότι ο Apache Tomcat λειτουργεί κανονικά, όταν μετά από την εκκίνηση της διεργασίας του application server, ανοίξουμε ένα browser και εισάγουμε τη διεύθυνση 127.0.0.1 ή πληκτρολογήσουμε localhost.

Αν όλα έχουν γίνει σωστά, τότε στο παράθυρο του browser θα πρέπει να εμφανίζεται μια σελίδα που μας καλωσορίζει στον application server Apache Tomcat και μοιάζει με την εικόνα που ακολουθεί.



Εικόνα 208: Η σελίδα που μας καλωσορίζει στον Apache Tomcat

Ρύθμιση των συνδέσεων SSL-TLS

Για τη ρύθμιση των ασφαλών συνδέσεων που υλοποιούνται σύμφωνα με το πρωτόκολλο SSL-TLS πρέπει να φέρουμε εις πέρας τις ακόλουθες ενέργειες:

- Να δημιουργήσουμε το δικό μας ψηφιακό πιστοποιητικό
- Να τροποποιήσουμε το αρχείο server.xml
- Να προσθέσουμε το πιστοποιητικό στις έμπιστες πηγές πιστοποίησης του browser που χρησιμοποιούμε

Ξεκινάμε με τη δημιουργία του ψηφιακού πιστοποιητικού, που είναι εφικτή χάρις στο keytool.exe που παρέχεται από το Java Development Kit ή/και το Runtime Environment που έχουμε εγκατεστημένο στον υπολογιστή μας.

Αφού έχουμε ορίσει τις μεταβλητές συστήματος που έχουν σχέση με τη Java, τότε αρκεί να ανοίξουμε ένα command-line shell και να πληκτρολογήσουμε την εντολή keytool μαζί με τις κατάλληλες παραμέτρους.

Με την ακόλουθη εντολή, και τα συνοδευτικά ορίσματα, ξεκινάμε τη διαδικασία με την οποία θα δημιουργηθεί το πιστοποιητικό που ταιριάζει στις ανάγκες μας:

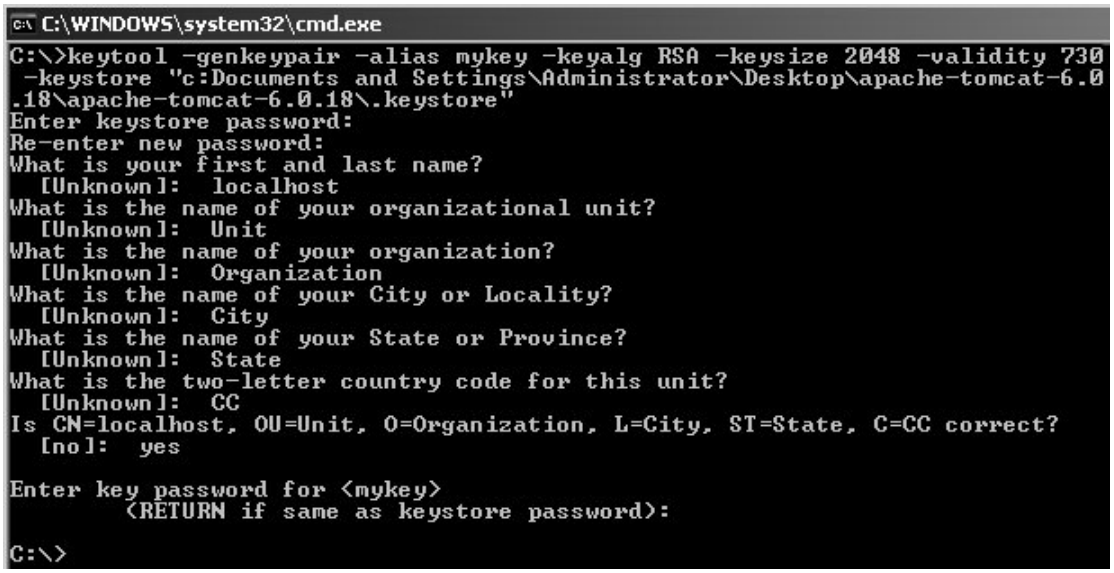
```
keytool –genkeypair –alias mykey –keyalg RSA –keysize 2048 –validity 730 –keystore "Διαδρομή\Όνομα_Αρχείου.keystore"
```

Σε αυτό το σημείο πρέπει να εξηγήσουμε τι αντιπροσωπεύει κάθε ένα από τα ορίσματα που χρησιμοποιούμε στην εντολή `keytool`, οπότε έχουμε:

- `–genkeypair`, υποδεικνύει ότι πρόκειται να δημιουργηθεί ένα ζεύγος κλειδιών.
- `–alias`, είναι προαιρετική η χρήση του και αποδίδει ένα ψευδώνυμο στο υπό δημιουργία πιστοποιητικό, στην προκειμένη περίπτωση το ονομάζει `mykey`. Αν δεν το χρησιμοποιήσουμε η εντολή ονομάζει από μόνη της το πιστοποιητικό.
- `–keyalg`, καθορίζει το δημόσιο αλγόριθμο που θα χρησιμοποιηθεί, όπου εδώ χρησιμοποιείται ο RSA.
- `–keysize`, με έναν ακέραιο αριθμό υποδεικνύεται το πλήθος των bits που θα έχει το ζεύγος κλειδιών, στην προκειμένη περίπτωση `2048`.
- `–validity`, είναι προαιρετική η χρήση του και με έναν ακέραιο αριθμό υποδεικνύεται το σύνολο των ημερών όπου θα θεωρείται έγκυρο το πιστοποιητικό, στην προκειμένη περίπτωση είναι `730` ημέρες. Αν δεν χρησιμοποιηθεί τότε αυτομάτως δίνεται μια περίοδος ισχύος των 6 μηνών.
- `–keystore`, είναι προαιρετική η χρήση του και καθορίζει σε ποιο φάκελο και με ποια ονομασία θα δημιουργηθεί το πιστοποιητικό, στην τρέχουσα εκτέλεση υποδείξαμε τη δημιουργία ενός αρχείου με την ονομασία `.keystore` στο φάκελο εγκατάστασης του Apache Tomcat. Αν δεν χρησιμοποιηθεί το όρισμα, τότε το αρχείο δημιουργείται, ανάλογα με το λειτουργικό σύστημα, στο βασικό κατάλογο του χρήστη και έχει την ονομασία `.keystore`. Πάντως, όπου και να δημιουργηθεί το αρχείο θα πρέπει τελικώς να αντιγραφεί στο βασικό φάκελο που έχουμε εγκαταστήσει τον Apache Tomcat, ενώ πρέπει να γνωρίζουμε και το όνομα του αφού χρειάζεται να το δηλώσουμε στο αρχείο server.xml που σχετίζεται με τη λειτουργία του Apache Tomcat.

Αμέσως μετά η εντολή `keytool` μας ζητάει μέσω prompt έναν κωδικό για το πιστοποιητικό, τον οποίο πρέπει να επιβεβαιώσουμε εισάγοντας τον εκ νέου. Στη συνέχεια, πάλι μέσω prompt καλούμαστε να υποβάλλουμε στοιχεία όπως, για παράδειγμα, το ονοματεπώνυμο μας και η τοποθεσία που βρισκόμαστε.

Όμως εδώ πρέπει να είμαστε προσεκτικοί και στη θέση του ονοματεπώνυμου να υποδείξουμε τη διεύθυνση IP στην οποία θα λειτουργεί ο Apache Tomcat, έτσι στην προκειμένη περίπτωση δώσαμε το ονοματεπώνυμο “localhost”.



Εικόνα 209: Η δημιουργία ενός ψηφιακού πιστοποιητικού μέσω του keytool.exe

Μετά από μια τελική επιβεβαίωση των στοιχείων που υποβάλλαμε το ψηφιακό πιστοποιητικό είναι έτοιμο, οπότε στη συνέχεια θα ασχοληθούμε με την τροποποίηση του αρχείου server.xml, το οποίο είδαμε και νωρίτερα κατά την εγκατάσταση του Apache Tomcat.

Με έναν επεξεργαστή κειμένου ανοίγουμε το αρχείο server.xml και αναζητούμε σε αυτό ένα τμήμα που περιέχει μιαν ετικέτα Connector που βρίσκεται μέσα σε σχόλια, δηλαδή σε tags <!-- και -->.

Μετά από την αφαίρεση των ετικετών που αντιπροσωπεύουν τα σχόλια έχουμε ένα tag Connector που έχει παρόμοια μορφή με αυτό που ακολουθεί.

```
<Connector port="8443" protocol="HTTP/1.1"
     SSLEnabled="true"
     maxThreads="150"
     scheme="https"
     secure="true"
     clientAuth="false"
     sslProtocol="TLS"/>
```

Οι αλλαγές που κάνουμε είναι οι ακόλουθες:

- Το `port` λαμβάνει την τιμή 443 που αντιστοιχεί στο default port όπου προσδένεται ο listener αιτημάτων για τις συνδέσεις SSL-TLS.

- Προστίθεται το στοιχείο `ciphers` με το οποίο αποκλείουμε τη χρήση κάποιων ασθενέστερων εκδόσεων κρυπτογραφικών αλγορίθμων, ενώ παράλληλα εκδηλώνουμε την προτίμηση μας σε συγκεκριμένους συνδυασμούς ισχυρότερων αλγορίθμων.
- Προστίθεται το στοιχείο `keystoreFile` με το οποίο υποδεικνύουμε το όνομα του keystore στο οποίο περιέχεται το ψηφιακό πιστοποιητικό που θα χρησιμοποιήσουμε.
- Προστίθεται το στοιχείο `keystorePass` στο οποίο εισάγουμε τον κωδικό που υποβάλλαμε κατά τη δημιουργία του ψηφιακού πιστοποιητικού.

Οπότε η ετικέτα Connector που λαμβάνουμε έχει μορφή παρόμοια με αυτήν που ακολουθεί, με τις τροποποιήσεις που κάναμε να εμφανίζονται με έντονους χαρακτήρες.

```
<Connector port="443" protocol="HTTP/1.1"
     SSLEnabled="true"
     maxThreads="150"
     scheme="https"
     secure="true"
     clientAuth="false"
     sslProtocol="TLS"
     ciphers="TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
              TLS_RSA_WITH_AES_128_CBC_SHA,
              SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA,
              SSL_RSA_WITH_3DES_EDE_CBC_SHA"
     keystoreFile=".keystore"
     keystorePass="changeit"/>
```

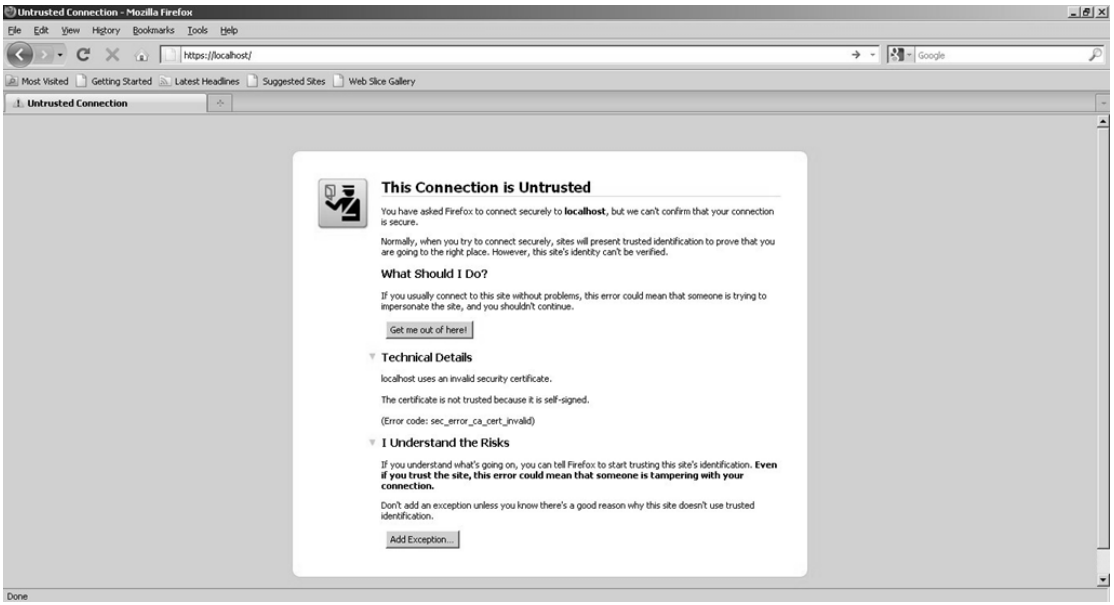
Αποθηκεύουμε το αρχείο, εκκινούμε εκ νέου τη διεργασία του server και ανοίγουμε το web browser που χρησιμοποιούμε για να επιβεβαιώσουμε ότι όντως παρέχονται ασφαλείς συνδέσεις.

Στη δική μας περίπτωση χρησιμοποιούμε το browser Mozilla Firefox, οπότε ο τρόπος διαχείρισης των ψηφιακών πιστοποιητικών ενδέχεται να διαφέρει σε κάποιο άλλο πρόγραμμα αυτού του τύπου.

Θέτοντας στη γραμμή διευθύνσεων του browser τη διεύθυνση http://localhost λαμβάνουμε ως αποτέλεσμα την αρχική σελίδα του Apache Tomcat, αλλάζοντας το http σε https ζητάμε την παροχή μιας ασφαλούς σύνδεσης με το πρωτόκολλο SSL-TLS.

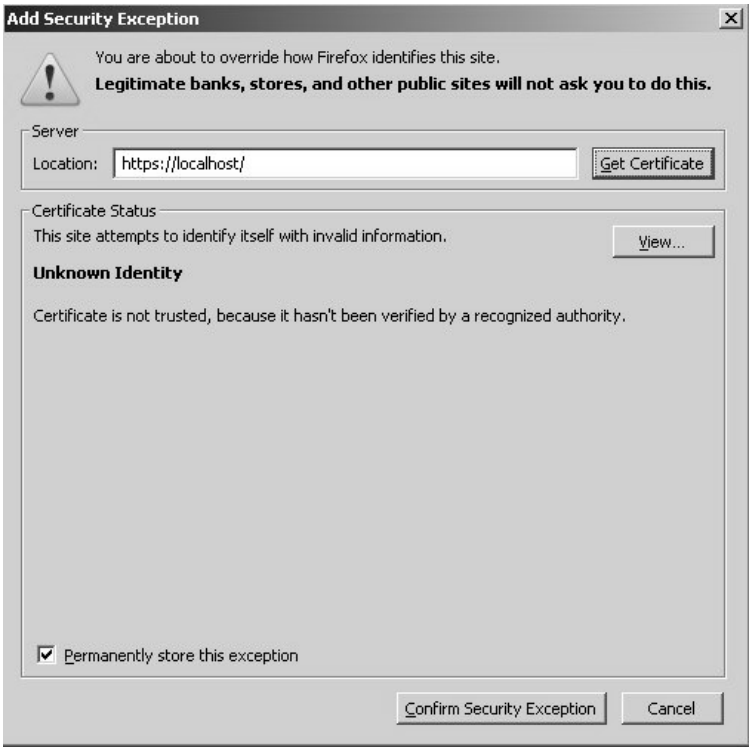
Όμως, επειδή το ψηφιακό πιστοποιητικό που χρησιμοποιείται δεν θεωρείται έμπιστο, αφού το έχουμε φτιάξει εμείς και όχι κάποια αξιόπιστη αρχή έκδοσης

πιστοποιητικών, ο browser μας εμφανίζει μιαν εικόνα παρόμοια με αυτήν που ακολουθεί.



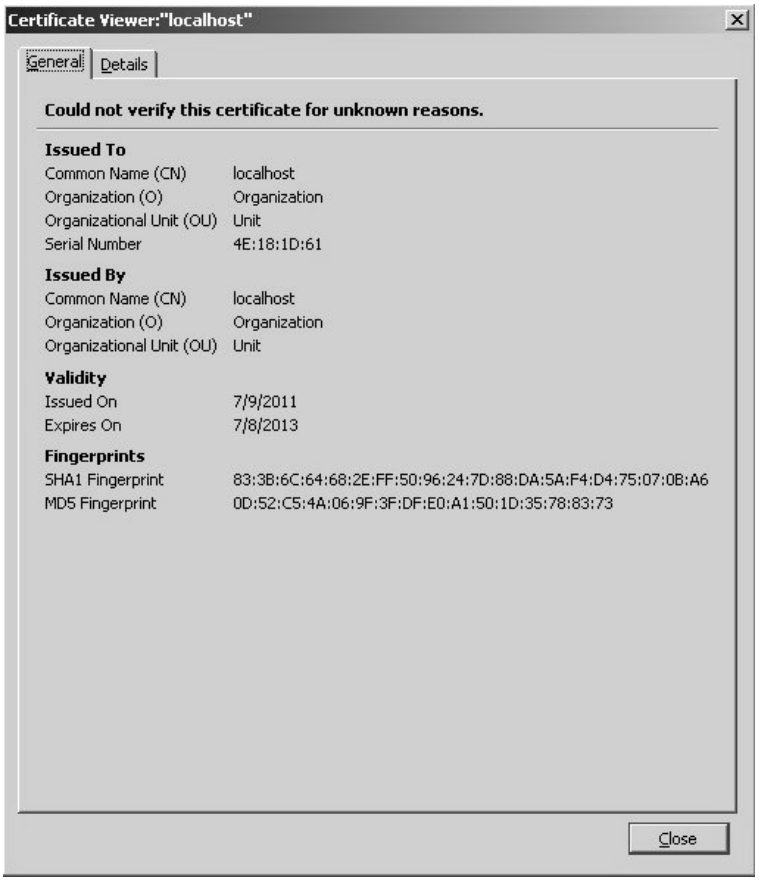
Εικόνα 210: Η οθόνη που εμφανίζεται από το Mozilla Firefox σε ένα μη έμπιστο ψηφιακό πιστοποιητικό

Προχωρούμε, ζητώντας από τον browser να προσθέσει μιαν εξαίρεση για το παρόν ψηφιακό πιστοποιητικό, οπότε μας εμφανίζεται το ακόλουθο pop-up παράθυρο με το οποίο μπορούμε, αρχικά, να δούμε περισσότερες πληροφορίες σχετικά με το ψηφιακό πιστοποιητικό, ενώ στη συνέχεια μπορούμε να επιβεβαιώσουμε την εξαίρεση ασφάλειας.

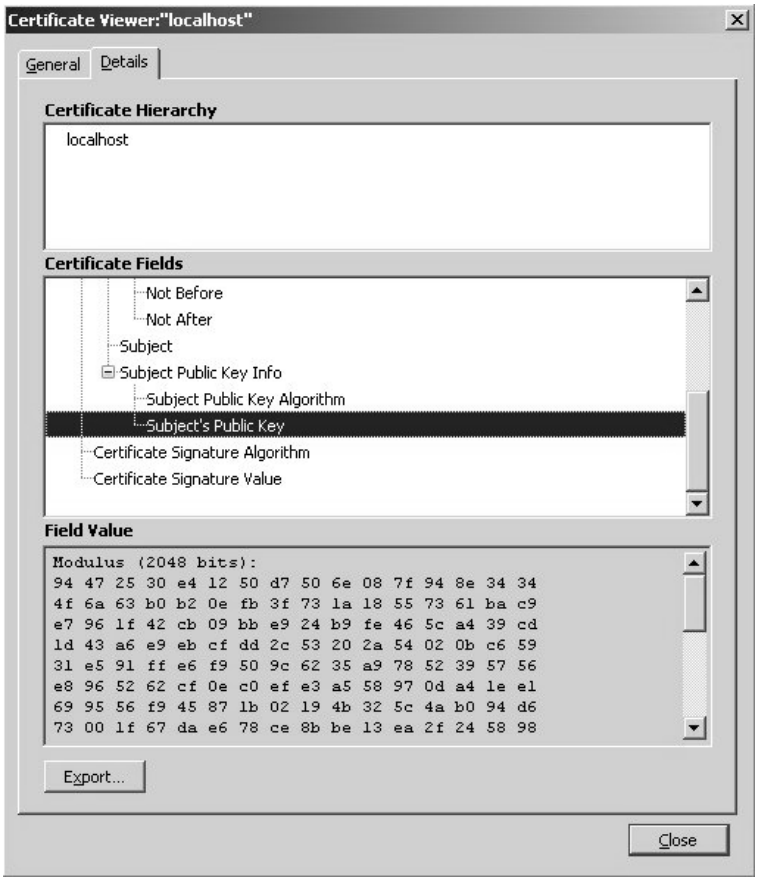


Εικόνα 211: Η προσθήκη εξαίρεσης για ένα μη έμπιστο ψηφιακό πιστοποιητικό

Πατώντας το πλήκτρο View περνάμε σε ένα άλλο παράθυρο με δύο tabs όπου στο ένα παρουσιάζονται στοιχεία όπως το όνομα του κατόχου του πιστοποιητικού και η περίοδος που θεωρείται έγκυρο, ενώ στο άλλο tab παρουσιάζονται περισσότερο ειδικές λεπτομέρειες όπως είναι η τιμή του δημοσίου κλειδιού που χρησιμοποιείται από το πιστοποιητικό.



Εικόνα 212: Η εμφάνιση του tab που περιέχει γενικά στοιχεία σχετικά με το ψηφιακό πιστοποιητικό

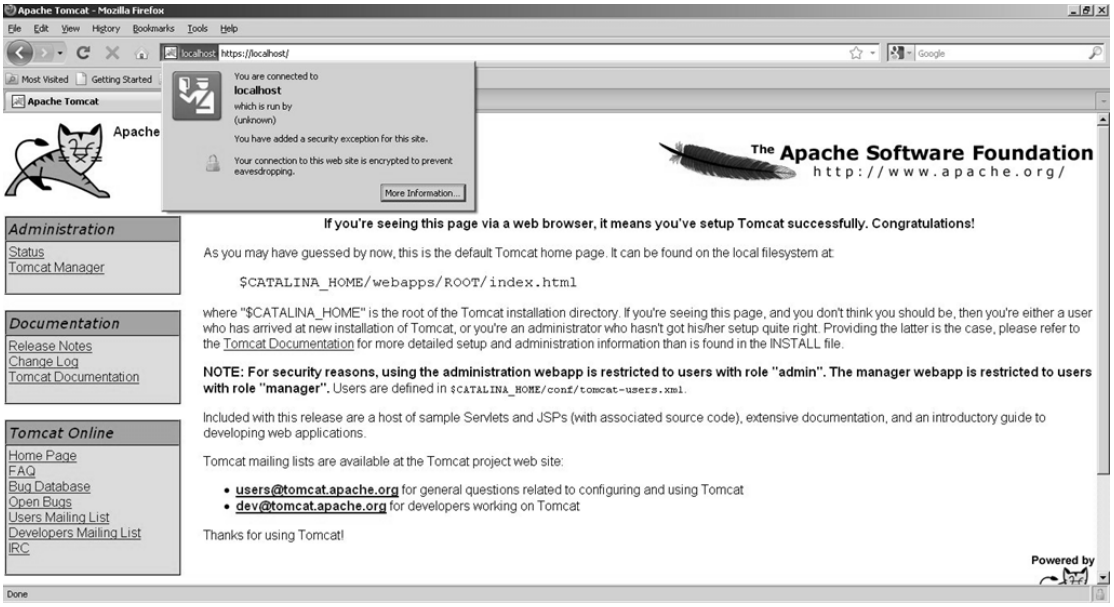


Εικόνα 213: Εμφάνιση του tab με τα ειδικά στοιχεία του χρησιμοποιούμενου ψηφιακού πιστοποιητικού

Αφού επιβεβαιώσουμε την εξαίρεση ασφάλειας το πιστοποιητικό προστίθεται στη λίστα με τις έγκυρες πηγές πιστοποίησης, οπότε, από εδώ και στο εξής, κάθε φορά που θα συναντούμε το συγκεκριμένο πιστοποιητικό θα έχουμε την παροχή ασφαλών συνδέσεων.

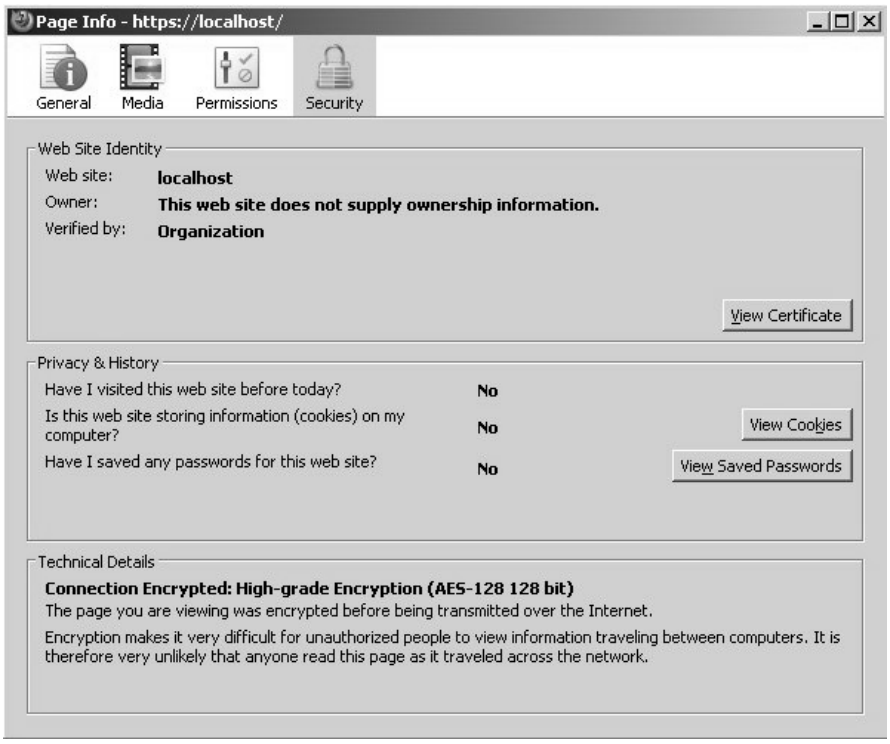
Για του λόγου το αληθές, μπορούμε να παρατηρήσουμε το παράθυρο του browser, όπου στην κάτω δεξιά γωνία είναι παρόν ένα μικρό λουκέτο και στη γραμμή της διεύθυνσης αλλάζει ο χρωματισμός.

Πατώντας επάνω στο χρωματιστό τμήμα της διεύθυνσης μπορούμε να δούμε ότι όντως η σύνδεση κρυπτογραφείται, ενώ έχουμε και τη δυνατότητα, πατώντας το πλήκτρο “More Information…”, να δούμε περισσότερες τεχνικές λεπτομέρειες σχετικά με την ασφαλή σύνδεση.



Εικόνα 214: Πατώντας στο χρωματιστό τμήμα της διεύθυνσης αποκτούμε πρόσβαση σε περισσότερο τεχνικές πληροφορίες

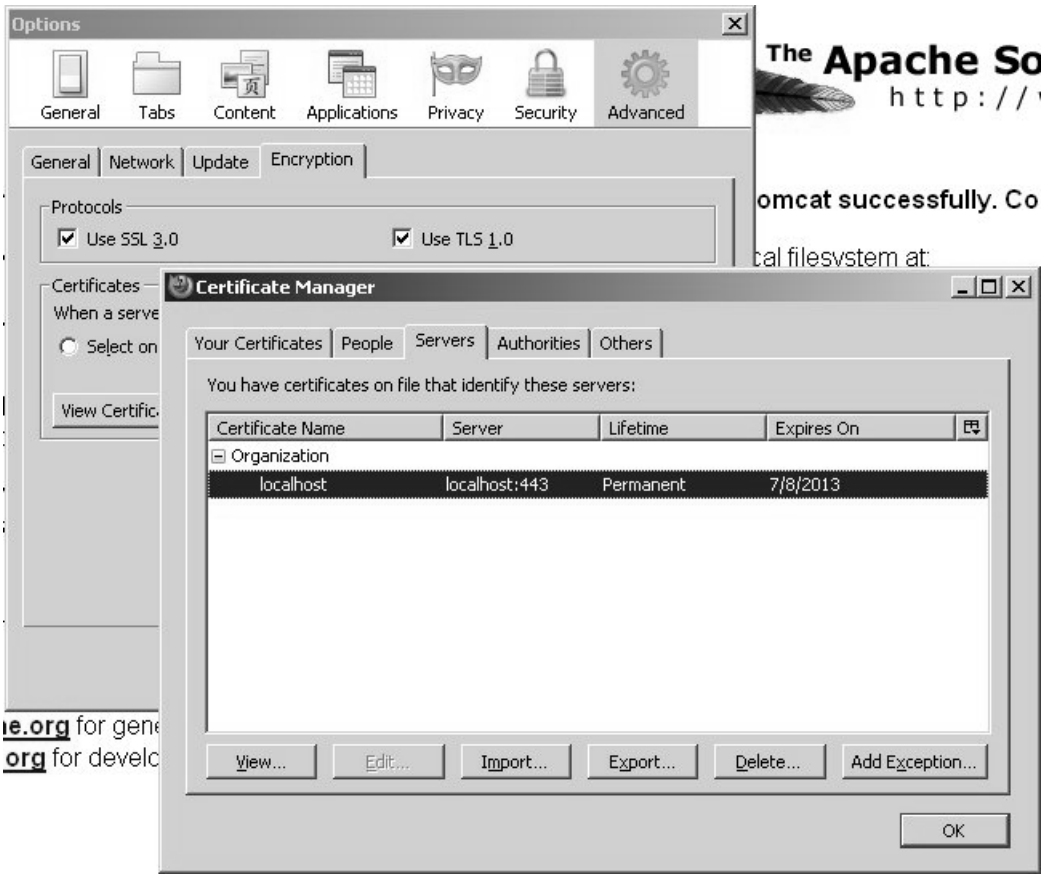
Έτσι, στο pop-up παράθυρο που εμφανίζεται μπορούμε να πληροφορηθούμε για την ταυτότητα του διαδικτυακού τόπου που επισκεπτόμαστε και παράλληλα βλέπουμε λεπτομέρειες για τον αλγόριθμο κρυπτογράφησης που χρησιμοποιείται κατά τη σύνδεση, όπου στην προκειμένη περίπτωση είναι ο AES με κλειδί 128 δυαδικών ψηφίων.



Εικόνα 215: Παράθυρο που εμφανίζεται με τεχνικές πληροφορίες

Ακόμη, ο browser παρέχει την επιλογή της διαχείρισης των ψηφιακών πιστοποιητικών που σχετίζονται με αυτόν, οπότε μπορούμε μέσω της συγκεκριμένης

επιλογής, όπως φαίνεται και στην εικόνα που ακολουθεί, να επιβεβαιώσουμε ότι το ψηφιακό πιστοποιητικό που έχουμε δημιουργήσει συμπεριλαμβάνεται στα έμπιστα.



Εικόνα 216: Η διαχείριση των ψηφιακών πιστοποιητικών μέσα από τον browser

Εγκατάσταση του Java Communications API και του MySQL Connector/J

Το API Java Communications και ο driver MySQL Connector/J είναι δύο συμπληρωματικές βιβλιοθήκες που ενσωματώνονται στην εγκατάσταση της Java που έχουμε στον υπολογιστή μας δίνοντας με αυτό τον τρόπο αυξημένες δυνατότητες.

Στην περίπτωση του Java Communications χρησιμοποιούμε μια παλιότερη έκδοση του, πιο συγκεκριμένα την έκδοση 2.0, αφού η πιο πρόσφατη δεν υποστηρίζεται από το λειτουργικό σύστημα Microsoft Windows. Στο συνοδευτικό ψηφιακό δίσκο μπορούμε να βρούμε τα κατάλληλα αρχεία, ενώ μετά από μια μικρή έρευνα έχουμε τη δυνατότητα να τα εντοπίσουμε και στο Διαδίκτυο, αφού το συγκεκριμένο API χρησιμοποιείται σε πληθώρα από projects όπως συμβαίνει με το ακόλουθο link:

http://jspoorloos.googlecode.com/files/javacomm20-win32.zip

Αντίθετα με το Java Communications API, ο τελευταίος driver της MySQL για τη γλώσσα Java υποστηρίζεται από τα Microsoft Windows και διατίθεται δωρεάν από το επίσημο site στον ακόλουθο σύνδεσμο:

http://www.mysql.com/downloads/connector/j/

Τα δύο αρχεία έχουν τη μορφή .jar και πρέπει να εγκατασταθούν με τις εξωτερικές βιβλιοθήκες που δέχεται το Java Standard Development Kit και το Java Runtime Environment.

Για το Standard Development Kit, η τοποθεσία όπου εγκαθίστανται οι εξωτερικές βιβλιοθήκες βρίσκεται στη διαδρομή:

Φάκελος_εγκατάστασης_Java_Development_Kit\JRE\LIB\EXT

Οι εξωτερικές βιβλιοθήκες για το Runtime Environment βρίσκονται στη διαδρομή:

Φάκελος_εγκατάστασης_Java_Runtime_Environment\LIB\EXT

Στο συμπιεσμένο αρχείο του Java Communications API υπάρχουν ακόμα δύο αρχεία τα οποία πρέπει να εγκαταστήσουμε στον υπολογιστή που θα εκτελεστεί η διαδικτυακή εφαρμογή. Πρόκειται για τα:

- win32com.dll, το οποίο πρέπει να αντιγραφεί στους φακέλους:
  - Java\bin
  - Java\jre\bin
  - Windows\system32
- javax.comm.properties, το οποίο πρέπει να αντιγραφεί στους φακέλους:
  - Java\lib
  - Java\jre\lib

Προετοιμασία για την εκτέλεση της Διαδικτυακής εφαρμογής

Πριν από την εκτέλεση της Διαδικτυακής εφαρμογής που συμμετέχει στο πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας οφείλουμε να ολοκληρώσουμε τις ακόλουθες ενέργειες:

1. Να εγκαταστήσουμε τα αρχεία της εφαρμογής στον κατάλληλο φάκελο της εγκατάστασης του Apache Tomcat.
2. Να συνδέσουμε τα δύο GSM-Modems που πρόκειται να χρησιμοποιηθούν.
3. Να εκκινήσουμε μια φορά τη διεργασία του Apache Tomcat, ούτως ώστε να δημιουργηθεί το ζεύγος κλειδιών που θα χρησιμοποιεί ο εξυπηρετητής. Μόλις ολοκληρωθεί η δημιουργία των κλειδιών πρέπει να σταματήσουμε τη λειτουργία του Apache Tomcat.
4. Να δημιουργήσουμε μιαν επίκαιρη έκδοση του MIDlet στην οποία ενσωματώσουμε το δημόσιο κλειδί που θα χρησιμοποιεί ο εξυπηρετητής.

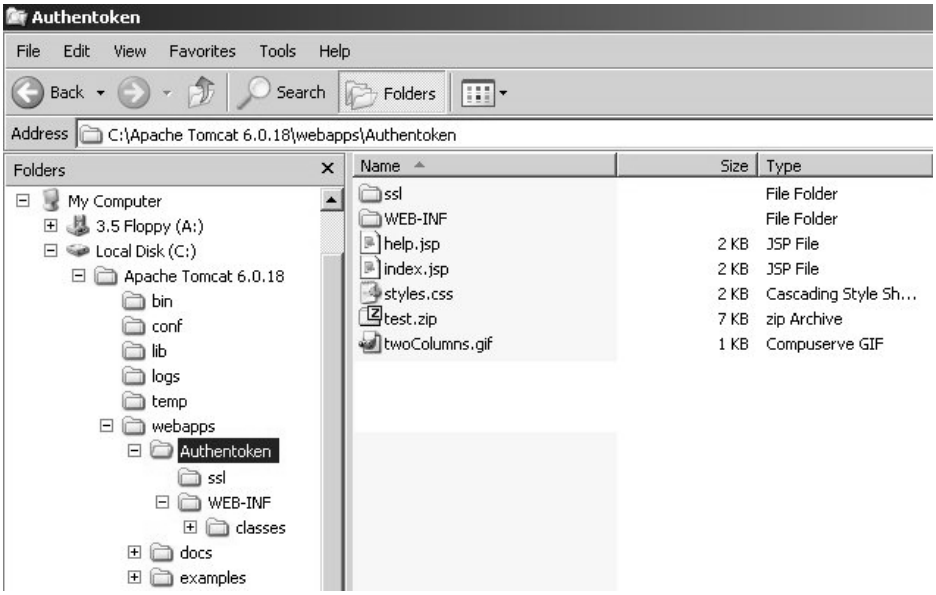
5. Να τοποθετήσουμε το δημιουργηθέν MIDlet στον κατάλληλο φάκελο του Apache Tomcat για να το διαθέσουμε στους χρήστες της Διαδικτυακής εφαρμογής.
6. Να εκκινήσουμε τη διεργασία του Apache Tomcat και είμαστε πλέον έτοιμοι να δεχθούμε τα αιτήματα που θα έρθουν από τους χρήστες της Διαδικτυακής εφαρμογής.

Οπότε, ξεκινάμε με την παρουσίαση των βημάτων που περιλαμβάνονται σε κάθε μια από τις παραπάνω ενέργειες που απαιτούνται πριν από την εκτέλεση της Διαδικτυακής εφαρμογής.

1. Εγκατάσταση των αρχείων από τα οποία αποτελείται η διαδικτυακή εφαρμογή.

   Σε αυτά τα αρχεία περιλαμβάνονται οι σελίδες JSP, που παρέχουν το web interface, οι κλάσεις των servlets, που επεξεργάζονται τα δεδομένα των πελατών και επικοινωνούν με τη βάση δεδομένων, καθώς και οι κλάσεις από τις οποίες υλοποιείται η γέφυρα επικοινωνίας.

   Όλα τα αρχεία αντιγράφονται στο φάκελο "`webapps`" του Apache Tomcat ακολουθώντας τη δομή αρχείων που υπάρχει στο φάκελο της διαδικτυακής εφαρμογής που συναντάμε στο συνοδευτικό ψηφιακό δίσκο.



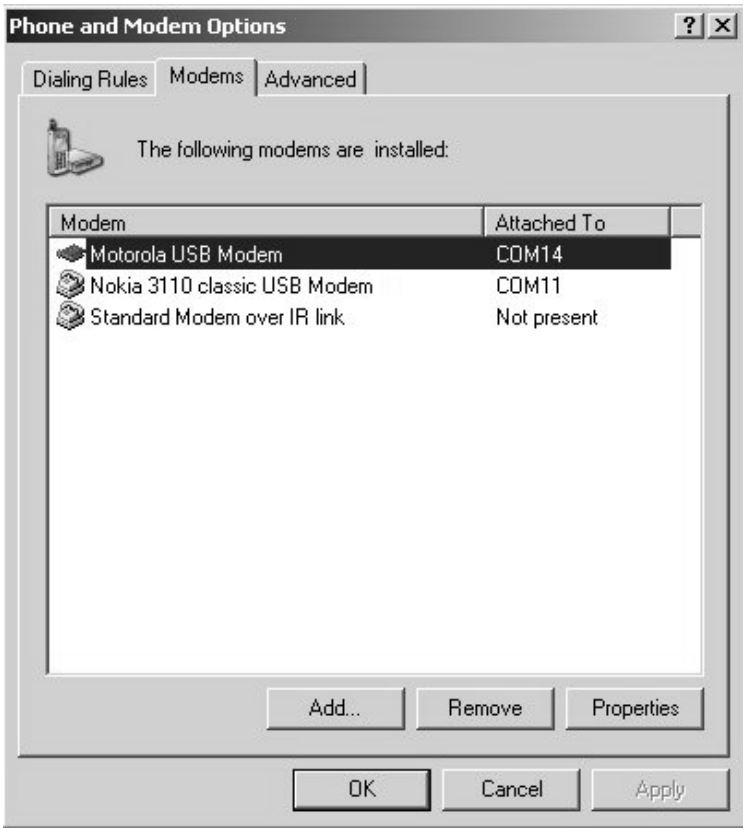
Εικόνα 217: Εμφάνιση του φακέλου webapps του Apache Tomcat

2. Σύνδεση με τα δύο GSM Modems

   Όπως αναφέραμε νωρίτερα, για τη συγκεκριμένη υλοποίηση του πρωτοκόλλου απομακρυσμένης πιστοποίησης ταυτότητας χρησιμοποιήθηκαν δύο συσκευές κινητών τηλεφώνων που συνδέονται μέσω καλωδίων με τον εξυπηρετητή.

Μετά από τη φυσική σύνδεση των δύο συσκευών, και την εγκατάσταση των αντίστοιχων drivers, είμαστε σε θέση να γνωρίζουμε τις θύρες επικοινωνιών που δεσμεύονται από τον υπολογιστή.

Αυτό γίνεται με τον εξής τρόπο, από τον πίνακα ελέγχου των Windows επιλέγουμε την κατηγορία "Phone and Modem" και από εκεί το tab "Modems" εμφανίζει τα εγκατεστημένα modems και τις θύρες όπου συνδέονται.



Εικόνα 218: Προβολή των εγκατεστημένων modems και των θυρών όπου συνδέονται

Σύμφωνα με τις κλάσεις που έχουν δημιουργηθεί για τη σύνδεση των GSM Modems και της Διαδικτυακής εφαρμογής θα πρέπει η συσκευή που αναλαμβάνει την εισερχόμενη κίνηση να είναι συνδεδεμένη στη θύρα 14, ενώ η συσκευή που αναλαμβάνει να προωθήσει τα σύντομα μηνύματα προς την πλευρά του πελάτη οφείλει να είναι συνδεδεμένη στη θύρα 11.

Οπότε, αφού κάνουμε τις όποιες απαιτούμενες αλλαγές, καλό θα ήταν να ελέγξουμε αν όντως επικοινωνούν οι δύο συσκευές και ο υπολογιστής. Έτσι, συνδεόμαστε με ένα πρόγραμμα προσομοίωσης τερματικού, όπως είναι το "PuTTY"[*], στις θύρες που αντιστοιχούν στα δύο τηλέφωνα και εκτελούμε ενδεικτικά κάποιες εντολές AT.

[*] Το πρόγραμμα προσομοίωσης τερματικού διατίθεται δωρεάν από την ιστοσελίδα http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Εικόνα 219: Ενδεικτική επικοινωνία μέσω terminal emulator με το GSM Modem

3. Δημιουργία της βάσης δεδομένων που θα χρησιμοποιείται από τη διαδικτυακή εφαρμογή.

   Κάθε μια φορά που εκκινεί η διεργασία του Apache Tomcat ελέγχεται αν υπάρχει στον εξυπηρετητή η βάση δεδομένων στην οποία περιέχονται παράμετροι για τη λειτουργία του αλγορίθμου RSA από την πλευρά του.

   Όταν ο έλεγχος πιστοποιεί ότι όντως υπάρχει η συγκεκριμένη βάση δεδομένων, τότε οι παράμετροι που περιέχονται σε αυτή προστίθενται στο context που διατηρεί η διαδικτυακή εφαρμογή δίνοντας έτσι τη δυνατότητα της γρήγορης ανάκτησης τους.

   Σε διαφορετική περίπτωση, με τη βοήθεια μιας setup κλάσης, έχουμε την αυτόματη δημιουργία της βάσης δεδομένων που θα υποδεχθεί τους δύο πίνακες που χρησιμοποιούνται από τη διαδικτυακή εφαρμογή.

   Ο ένας εκ των δύο πινάκων που δημιουργούνται περιέχει στοιχεία των πελατών, όπως είναι τα username και password, ο αριθμός του κινητού τηλεφώνου, και άλλα.

   Στον έτερο πίνακα αποθηκεύονται το δημόσιο και το ιδιωτικό κλειδί που χρησιμοποιούνται για τον αλγόριθμο RSA.

   Το ζεύγος κλειδιών δημιουργείται και αυτό κατά την εκτέλεση της κλάσης και αμέσως αποθηκεύεται στον αντίστοιχο πίνακα. Παράλληλα, το δημόσιο κλειδί εξάγεται σε ένα αρχείο κειμένου που δημιουργείται στο σκληρό δίσκο C:, κάνοντας έτσι κατά πολύ ευκολότερη την αντιγραφή του δημόσιου κλειδιού που απαιτείται στο βήμα που ακολουθεί.

4. Δημιουργία μιας ενημερωμένης έκδοσης του MIDlet.

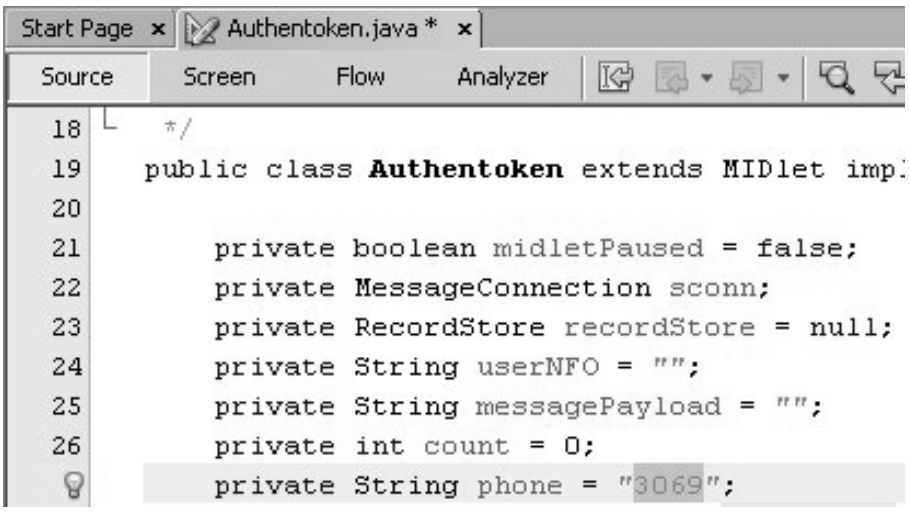
   Μετά από τη δημιουργία του ζεύγους κλειδιών πρέπει να αντιγράψουμε το δημόσιο τμήμα του στον πηγαίο κώδικα του MIDlet, έχοντας έτσι τη σιγουριά ότι σε κάθε αντίγραφο της εφαρμογής θα χρησιμοποιούνται τα ίδια στοιχεία, μειώνοντας με αυτό τον τρόπο την πιθανότητα κάποιου λάθους.

   Επιπλέον υπάρχει και μεγαλύτερη ασφάλεια για τους πελάτες, καθώς θα ήταν πολύ πιο εύκολο να εισαγόταν η τιμή του δημοσίου κλειδιού στο αρχείο JAD, όμως με αυτό τον τρόπο θα μπορούσε ένας τρίτος να αλλάξει τα στοιχεία και στη συνέχεια να πείσει τους χρήστες να χρησιμοποιήσουν την τροποποιημένη εφαρμογή.

   Ακόμη, με την ευκαιρία της πρόσβασης στον πηγαίο κώδικα, μπορούμε να εισάγουμε τον αριθμό του τηλεφώνου που θα αντιστοιχεί στο GSM Modem που πρόκειται να χρησιμοποιηθεί για τη διαχείριση της εισερχόμενης κίνησης.

   Η εισαγωγή των συγκεκριμένων δεδομένων στον πηγαίο κώδικα γίνεται μέσω του NetBeans IDE, όταν εισάγουμε σε αυτό το project και κάνουμε edit το αρχείο “Authentoken.java”.

   Ο αριθμός τηλεφώνου που θα χρησιμοποιείται από το GSM Modem εισάγεται ανάμεσα στα quotes στη γραμμή 27.

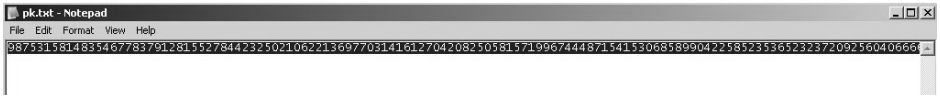


Εικόνα 220: Εντοπισμός της γραμμής κώδικα όπου εισάγεται το τηλέφωνο του GSM Modem που λαμβάνει μηνύματα SMS

   Η τιμή του δημοσίου κλειδιού έχει εξαχθεί από το MySQL Server σε ένα αρχείο κειμένου το οποίο δημιουργείται αυτόματα στο C:\ και ονομάζεται “pk.txt”.

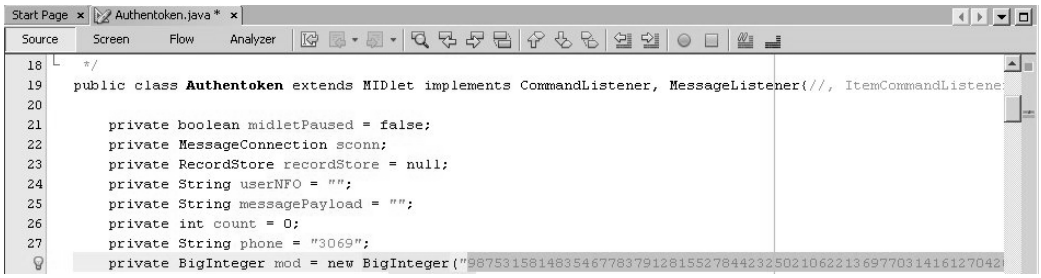
Εικόνα 221: Το αρχείο κειμένου που περιέχει το δημόσιο κλειδί που έχει δημιουργηθεί από τη διαδικτυακή εφαρμογή

Με έναν απλό επεξεργαστή κειμένου μπορούμε να αποκτήσουμε πρόσβαση στο συγκεκριμένο αρχείο και να αντιγράψουμε την τιμή που έχει το δημόσιο κλειδί.



Εικόνα 222: Αντιγραφή του δημοσίου κλειδιού μέσα από έναν επεξεργαστή αρχείων κειμένου

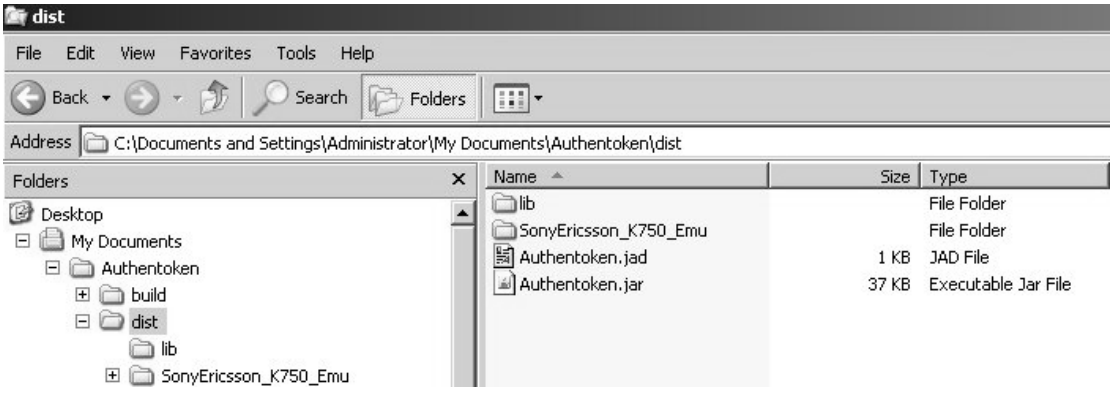
Η επικόλληση της τιμής του δημοσίου κλειδιού στον κώδικα γίνεται ανάμεσα από τα διπλά εισαγωγικά που βρίσκονται στη γραμμή 28.



Εικόνα 223: Εντοπισμός της γραμμής κώδικα όπου εισάγεται το δημόσιο κλειδί

5. Διάθεση του νέου MIDlet μέσω της διαδικτυακής εφαρμογής.

Τα αρχεία JAR και JAD, που αντιστοιχούν στο ενημερωμένο MIDlet, παράγονται από το Integrated Development Environment NetBeans και τοποθετούνται στο φάκελο "`dist`" που αντιστοιχεί στο project του MIDlet.



Εικόνα 224: Τα παραγόμενα JAR και JAD μέσα από τα projects του IDE NetBeans

Εντοπίζουμε τα δύο αρχεία και τα συμπιέζουμε σε ένα αρχείο zip, το οποίο στη συνέχεια αντιγράφουμε στο φάκελο "`ssl`" της διαδικτυακής εφαρμογής.

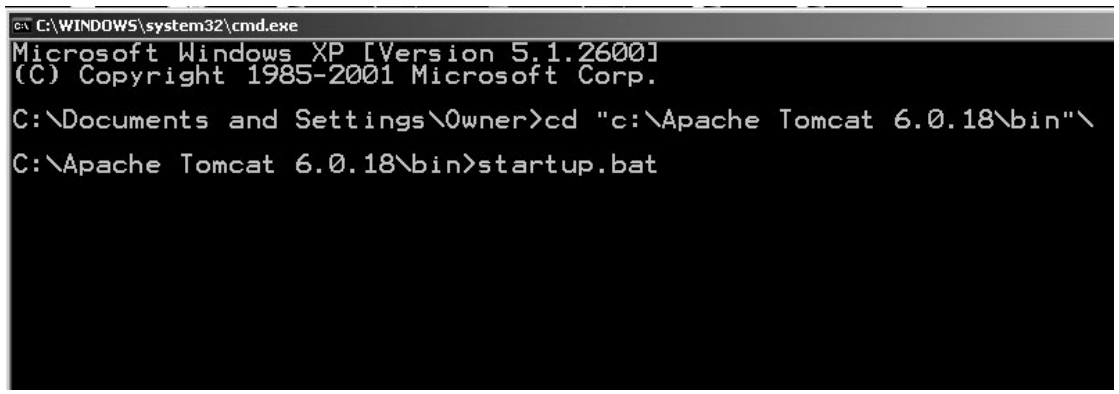
Σε περίπτωση που κάποιο από τα στοιχεία που εισάγονται στον κώδικα αλλάξει, τότε θα αλλάζει και η έκδοση της εφαρμογής που είναι διαθέσιμη από το διαδίκτυο και όλοι οι πελάτες θα υποχρεώνονται να κατεβάσουν τη νεότερη έκδοση της εφαρμογής.

Με αυτό τον τρόπο θα είμαστε σίγουροι ότι οι πελάτες που θα συμμετέχουν στο πρωτόκολλο θα έχουν τη σωστή έκδοση του MIDlet.

6. Εκκίνηση της διεργασίας του Apache Tomcat.

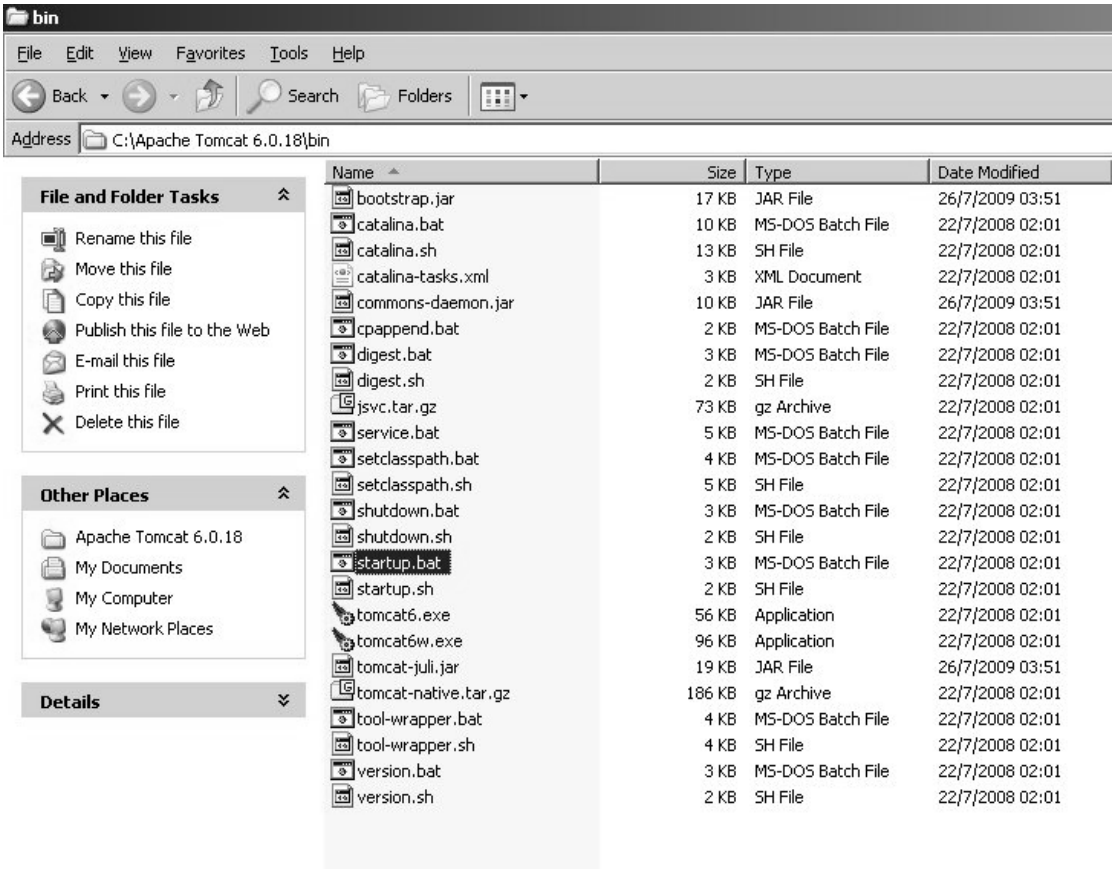
Η έναρξη του Tomcat πραγματοποιείται μέσω του αρχείου "`startup.bat`", το οποίο βρίσκεται στον κατάλογο εγκατάστασης του Tomcat και μπορούμε να το εντοπίσουμε με δύο τρόπους, είτε μέσω command line ή μέσα από τον windows explorer.

Όταν χρησιμοποιούμε command line πληκτρολογούμε "`cd`" και τη διαδρομή που αντιστοιχεί στο φάκελο "`bin`" της εγκατάστασης του Tomcat. Από εκεί πληκτρολογούμε "`startup.bat`".



Εικόνα 225: Εκκίνηση του Apache Tomcat από το command-line

Στην περίπτωση που χρησιμοποιούμε το πρόγραμμα windows explorer τα πράγματα είναι απλούστερα, οπότε επιλέγουμε από το sidebar με τους φακέλους την κατάλληλη διαδρομή και στη συνέχεια με διπλό κλικ στο αρχείο "`startup.bat`" εκκινεί η διεργασία του Tomcat.



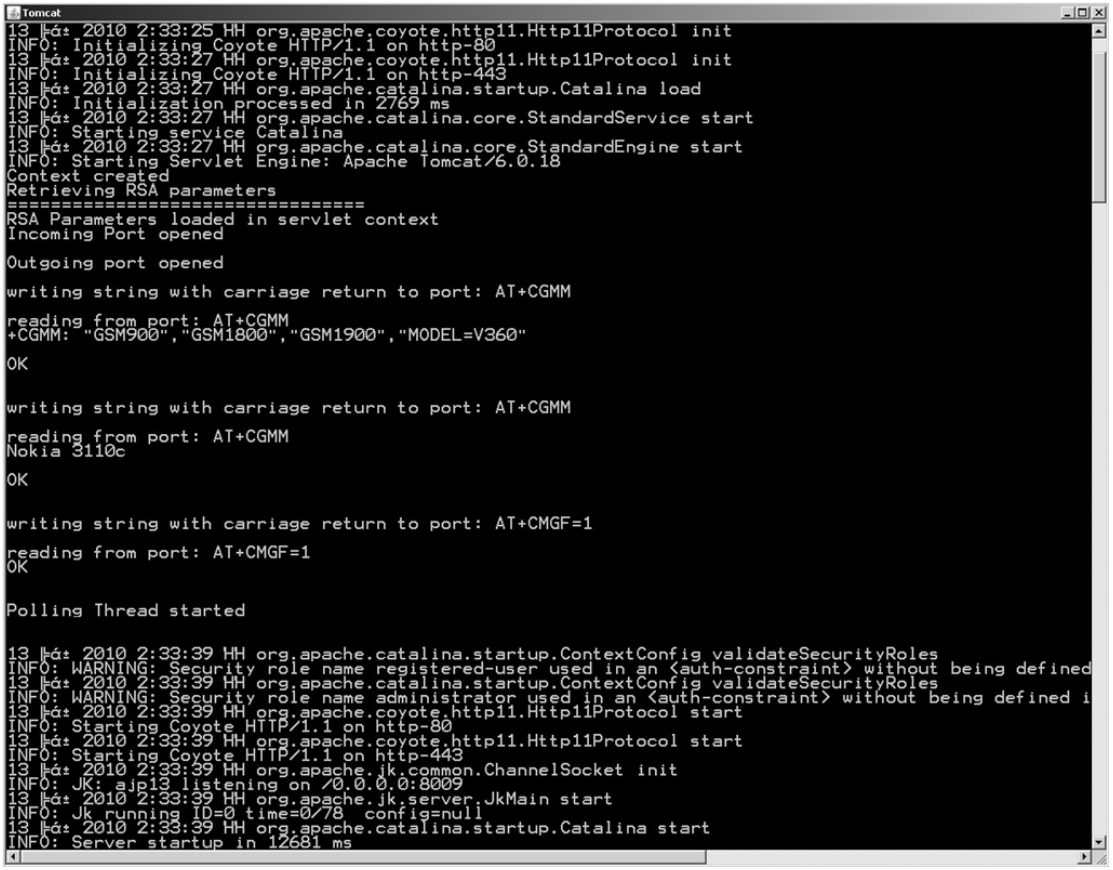
Εικόνα 226: Εκκίνηση του Apache Tomcat μέσω του Windows Explorer

Κατά την εκκίνηση της διεργασίας εμφανίζεται ένα βοηθητικό παράθυρο που χρησιμοποιείται σαν ενημερωτική κονσόλα, καθώς σε αυτό απεικονίζονται μηνύματα που σχετίζονται με τη γενικότερη λειτουργία του web server Apache Tomcat αλλά και την εκτέλεση των εφαρμογών που φιλοξενεί.

Η δική μας διαδικτυακή εφαρμογή εκμεταλλεύεται την παρουσία της συγκεκριμένης κονσόλας και έτσι εκτυπώνει σε αυτήν ένα σημαντικό αριθμό μηνυμάτων.

Με αυτό τον τρόπο διευκολύνεται η διαδικασία που θα ακολουθείτο για την εύρεση των λαθών που θα εμφανίζονταν από την διαδικτυακή εφαρμογή, ενώ παράλληλα είναι δυνατή η καλύτερη παρουσίαση όσων γεγονότων εκτελούνται παρασκηνιακά και αποκρύπτονται από ένα χρήστη της εφαρμογής.

Η ακόλουθη εικόνα περιέχει τα μηνύματα που εμφανίζονται κατά την εκκίνηση του Apache Tomcat, ενώ ταυτόχρονα ξεκινά και μέρος της εφαρμογής μας, καθώς πραγματοποιείται η σύνδεση με τα δύο κινητά τηλεφώνα – GSM Modems και ενεργοποιείται το thread του listener για τα εισερχόμενα μηνύματα.



Εικόνα 227: Η κονσόλα που εμφανίζεται κατά την εκκίνηση του Apache Tomcat

Παράρτημα 2 - Πίνακας Συντομογραφιών

0-9	**1G**	First Generation
	2G	Second Generation
	3DES	Triple DES
	3G	Third Generation
	3GPP	Third Generation Partnership Project
	3GPP 2	Third Generation Partnership Project 2
	4G	Fourth Generation
A	**ACTS**	Advanced Communications Technologies and Services
	AES	Advanced Encryption Standard
	AK	Anonymity Key
	AKA	Authentication and Key Agreement
	AMF	Authentication Management Field
	AMPS	Advanced Mobile Phone System
	API	Application Programming Interface
	ATM	Asynchronous Transfer Mode
	AuC	Authentication Centre
	AUTN	Authentication Token
	AV	Authentication Vector
B	**BS**	Base Station
	BSC	Base Station Controller
	BSS	Base Station Subsystem
	BTS	Base Transceiver Station
C	**CDC**	Connected Device Configuration
	CDMA	Code Division Multiple Access
	CEPT	Conférence Européenne des Postes et des Télécommunications
	CFN	Correction Frame Number
	CK	Cipher Key
	CLDC	Connected Limited Device Configuration
	CN	Core Network
	CRT	Chinese Remainder Theorem
	CS-MGW	Circuit Switched – Media Gateway
D	**D-AMPS**	Digital AMPS
	DECT	Digital Enhanced Cordless Telephone
	DES	Data Encryption Standard
	DoS	Denial of Service
E	**EDGE**	Enhanced Data for GSM/Global Evolution
	EFF	Electronic Frontier Foundation
	EIR	Equipment Identity Register
	ETSI	European Telecommunications Standards Institute
	EUL	Enhanced UpLink
	E-UTRAN	Evolved – UTRAN
	EV-DO	EVolution-Data Only / EVolution-Data Optimized
F	**FDMA**	Frequency Division Multiple Access
	FM	Frequency Modulation

	FOMA	Freedom Of mobile Multimedia Access
	FPLMTS	Future Public Land Mobile Telecommunications System
	FRAMES	Future Radio wideband MultiplE access Systems
G	**GGSN**	Gateway GSN
	GMSC	Gateway of MSC
	GPRS	Generic Packet Radio Service
	GSM	Global System for Mobile communications / Groupe Spéciale Mobile
	GSN	GPRS Support Node
H	**HFN**	Hyper Frame Number
	HLR	Home Location Register
	HSCSD	High Speed Circuit Switched Data
	HSDPA	High Speed Download Packet Access
	HSPA	High Speed Packet Access
	HSPA+	Evolved HSPA / HSPA Evolution
	HSS	Home Subscriber Service
	HSUPA	High Speed Upload Packet Access
	HTML	Hyper Text Markup Language
	HTTP	Hyper Text Transfer Protocol
I	**IK**	Integrity Key
	IMEI	International Mobile Equipment Identity
	IMS	IP Multimedia Subsystem
	IMSI	International Mobile Subscriber Identity
	IMT-2000	International Mobile Telecommunications standard 2000
	IMTS	Improved Mobile Telephone System
	IP	Internet Protocol
	IS-95B	Interim Standard 95B
	ISC	International Switching Centre
	ITU	International Telecommunications Union
J	**J2EE / Java EE**	Java Enterprise Edition
	J2ME / Java ME	Java Micro Edition
	J2SE / Java SE	Java Standard Edition
	JAD	Java Application Descriptor
	JAM	Java Application Manager
	JAR	Java Archive
	JDBC	Java Data Base Connectivity
	JSP	Java Server Pages
	JVM	Java Virtual Machine
K	**K_c**	Encryption Key
	K_i	Authentication Key
	KR	Private Key
	KU	Public Key
	KVM	Kilo Virtual Machine
L	**LAI**	Local Area Identity
	LTE	Long Term Evolution

M	**MAC**	Message Authentication Code
	MAC	Medium Access Control
	MCC	Mobile Country Code
	ME	Mobile Equipment
	MIDP	Mobile Information Device Profile
	MIMO	Multiple Input Multiple Output
	MNC	Mobile Network Code
	MO-SM	Mobile Originated – Short Message
	MS	Mobile Station
	MSC	Mobile services Switching Centre
	MSIN	Mobile Subscriber Identification Number
	MSISDN	Mobile Subscriber ISDN Number
	MSS	MSC Server
	MT-SM	Mobile Terminated – Short Message
N	**NIST**	National Institute for Standards and Technology
	NMT	Nordic Mobile Telephone system
	NSS	Network and Switching Subsystem
O	**OFDMA**	Orthogonal Frequency Division Multiple Access
	OMC	Operation and Maintenance Centre
	OSS	Operation Support Subsystem
P	**PBKDF**	Password Based Key Derivation Function
	PC	Personal Computer
	PDA	Palm Digital Assistant
	PDC	Personal Digital Cellular
	PDN	Packet Data Networks
	PDU	Protocol Data Unit
	PIN	Personal Identification Number
	P-TMSI	Packet TMSI
	PTT	Push To Talk
Q	**QoS**	Quality of Service
R	**RACE**	Research in Advanced Communications in Europe
	RAI	Routing Area Identity
	RAN	Radio Access Network
	RAND	Random Number
	RDBMS	Relational Data Base Management System
	RES	Result
	RLC	Radio Link Control
	RLC-SN	RLC – Sequence Number
	RNC	Radio Network Controller
	RRC	Radio Resource Control
	RSA	Rivest Shamir Adleman
S	**SAE**	System Architecture Evolution
	SATSA	Security And Trust Services API
	SC	Service Centre
	SC-FDMA	Single Carrier - FDMA
	SGSN	Serving GSN
	SHA	Secure Hash Algorithm

	SIM	Subscriber Identity Module
	SM-AL	Short Message – Application Layer
	SME	Short Message Entity
	SM-LL	Short Message – Link Layer
	SM-RL	Short Message – Relay Layer
	SMS	Short Message Service
	SMSC	SMS Centre
	SMS-GMSC	Short Message Service – Gateway for Mobile service Switching Centre
	SMS-IWMSC	Short Message Service – InterWorking Mobile service Switching Centre
	SM-TL	Short Message – Transfer Layer
	SM-TP	Short Message – Transfer Protocol
	SQL	Standard Query Language
	SQN	Sequence Number
	SSL	Secure Socket Layer
T	**TACS**	Total Area Communication System
	TDMA	Time Division Multiple Access
	TE	Terminal Equipment
	TLS	Transport Layer Security
	TMSI	Temporary Mobile Subscriber Identity
	TP-DA	Transfer Protocol – Destination Address
	TP-DCS	Transfer Protocol – Data Coding Scheme
	TPDU	Transfer Protocol Data Unit
	TP-MMS	Transfer Protocol – More Messages to Send
	TP-MR	Transfer Protocol – Message Reference
	TP-MTI	Transfer Protocol – Message Type Indicator
	TP-OA	Transfer Protocol – Originator Address
	TP-PI	Transfer Protocol – Protocol Identifier
	TP-RD	Transfer Protocol – Reject Duplicates
	TP-RP	Transfer Protocol – Reply Path
	TP-SCTS	Transfer Protocol – Service Centre Time Stamp
	TP-SRR	Transfer Protocol – Status Report Request
	TP-UD	Transfer Protocol – User Data
	TP-UDHI	Transfer Protocol – User Data Header Indicator
	TP-UDL	Transfer Protocol – User Data Length
	TP-VP	Transfer Protocol – Validity Period
	TP-VPF	Transfer Protocol – Validity Period Format
U	**UE**	User Equipment
	UEA1	UMTS Encryption Algorithm 1
	UEA2	UMTS Encryption Algorithm 2
	UMB	Ultra Mobile Broadband
	UMTS	Universal Mobile Telecommunications System
	USIM	UMTS SIM
	UTRAN	UMTS Terrestrial Radio Access Network
V	**VLR**	Visitor Location Register
	VM	Virtual Machine
	VoIP	Voice over IP
W	**WBTS**	WCDMA BTS
	WCDMA	Wideband CDMA
	WMA	Wireless Messaging API

X	**XMAC**	eXpected MAC
	XML	eXtensible Markup Language
	XOR	eXclusive OR
	XRES	eXpected Response
Y		
Z		

Παράρτημα 3 - Βιβλιογραφία

1. **Computer Networks, 4th edition**. Andrew S. Tanenbaum. Prentice Hall (2003)

2. **Computer and Communications Networks**. Nader F. Mir. Prentice Hall (2006)

3. **GSM and Personal Communications Handbook**. Siegmund M. Redl, Matthias K. Weber, Malcolm W. Oliphant. Artech House (1998)

4. **Mobile Messaging Technologies and Services: SMS, EMS and MMS, Second Edition**. Gwenaël Le Bodic. John Wiley and Sons (2005)

5. **Wireless Communications and Networking**. Vijay Garg. Morgan Kaufmann Publishers (2007)

6. **GSM switching, services and protocols, Second Edition**. Jörg Eberspächer, Hans-Jörg Vögel, Christian Bettstetter. John Wiley and Sons (2001)

7. **Convergence Technologies for 3G Networks: IP, UMTS, EGPRS and ATM**. Jeffrey Bannister, Paul Mather, Sebastian Coope. John Wiley and Sons (2004)

8. **UMTS Security**. Valtteri Niemi, Kaisa Nyberg. John Wiley and Sons (2003)

9. **UMTS Networks Architecture, Mobility and Services, Second Edition**. Heikki Kaaranen, Ari Ahtiainen, Lauri Laitinen, Siamäk Naghian, Valtteri Niemi. John Wiley and Sons (2005)

10. **Bulletproof Wireless Security GSM, UMTS, 802.11 and Ad Hoc Security**. Praphul Chandra. Elsevier (2005)

11. **Cryptography and Network Security Principles and Practices, Fourth Edition**. William Stallings. Prentice Hall (2005)

12. **Cryptography for Developers**. Tom St Denis, Simon Johnson. Syngress Publishing (2007)

13. **FIPS 180-2: Secure Hash Standard**. NIST (2002)

14. **Handbook of Applied Cryptography**. Alfred Menezes, Paul van Oorschot, Scott Vanstone. CRC Press (1997)

15. **FIPS 197: Advanced Encryption Standard**. NIST (2001)

16. **PKCS #5 v2.1: Password-Based Cryptography Standard**. RSA Laboratories (2006)

17. **Contemporary Cryptography**. Rolf Oppliger. Artech House (2005)

18. **Java How to Program, 6th edition**. Harvey M. Deitel, Paul J. Deitel. Prentice Hall (2004)

19. **Enterprise J2ME: Developing Mobile Java Applications**. Michael Juntao Yuan, Prentice Hall PTR (2003)

20. **J2ME: The Complete Reference**. James Keogh. McGraw-Hill / Osborne (2003)

21. **Learning Wireless Java**. Qusay Mahmoud. O'Reilly (2001)

22. **Java 2 Micro Edition – Java in Small Things**. James White, David Hemphill. Manning (2002)

23. **Core Servlets and Java Server Pages, Volume 1: Core Technologies, 2nd edition**. Marty Hall, Larry Brown. Prentice Hall PTR (2003)

24. **Core Servlets and Java Server Pages, Volume 2: Advanced Technologies, 2nd edition**. Marty Hall, Larry Brown, Yaakov Chaikin. Prentice Hall (2007)

25. **Java Database Programming Bible**. John O'Donahue. John Wiley and Sons (2002)

Παράρτημα 4 - Διαδικτυακές Πηγές

Πέρα από τους διαδικτυακούς τόπους που αναφέρονται μέσω παραπομπών στο κύριο τμήμα της εργασίας, υπήρξαν κάποιες επιπλέον πηγές που αξίζει να τις επισημάνουμε καθώς και αυτές συνέβαλαν στην υλοποίηση της εργασίας.

Για το δίκτυο κινητής τηλεφωνίας GSM χρησιμοποιήθηκαν τα sites:

- www.gsmworld.com
- www.gsm-security.net

Για τις εντολές AT υπήρξε πολύτιμος ο οδηγός που βρίσκεται στον ιστότοπο:
www.developershome.com/sms/

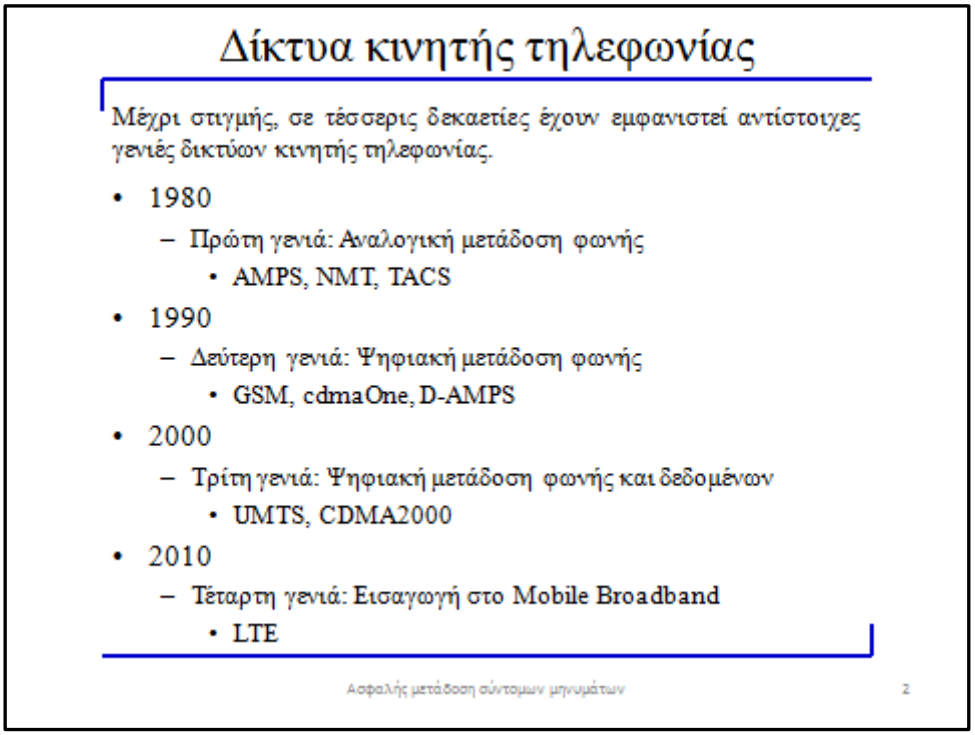
Για την κατανόηση της πλατφόρμας Java Micro Edition βοήθησε η σελίδα:
http://developers.sun.com/mobility/midp/articles/wtoolkit/

Αρκετές απορίες επάνω στη λειτουργία του AES και την άλγεβρα των πεπερασμένων πεδίων λύθηκαν μέσω του site:
www.samiam.org/rijndael.html

Για την οικογένεια των συναρτήσεων κατακερματισμού SHA βοήθησε το site:
www.quadibloc.com/crypto/mi060501.htm

Παράρτημα 5 - Παρουσιάσεις PowerPoint

Παρουσίαση εφαρμογής για την ασφαλή ανταλλαγή σύντομων μηνυμάτων

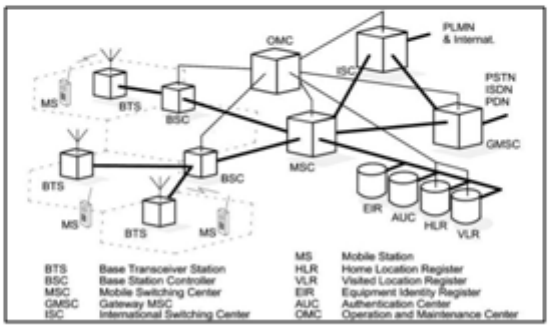


Η αρχιτεκτονική ενός δικτύου κινητής τηλεφωνίας

Ένα δίκτυο κινητής τηλεφωνίας αποτελείται από:

- Το σταθερό τμήμα, δηλαδή την υποδομή του δικτύου.
- Το κινούμενο τμήμα, δηλαδή τους κινούμενους συνδρομητές.

Η αρχιτεκτονική του GSM, το οποίο είναι το πλέον δημοφιλές δίκτυο κινητής τηλεφωνίας.



Η ασφάλεια των κινητών επικοινωνιών

Το δίκτυο κορμού θεωρείται ότι είναι ένα ασφαλές περιβάλλον.

- Ελέγχεται πλήρως από τον πάροχο κινητής τηλεφωνίας.
- Οι χρήστες δεν έχουν φυσική πρόσβαση σε αυτό.

Το δίκτυο ασύρματης διασύνδεσης θεωρείται ότι είναι ένα περιβάλλον μη ελεγχόμενο, καθώς η φυσική πρόσβαση σε αυτό δεν μπορεί να περιοριστεί.

Το γεγονός αυτό επιτρέπει σε οποιονδήποτε έχει τον κατάλληλο δέκτη να παρακολουθεί τη μεταδιδόμενη κίνηση όπως:

- Τα δεδομένα των χρηστών (data)
- Τη σηματοδοσία (signaling) μεταξύ του δικτύου και των τερματικών σταθμών των χρηστών.

Η αντιμετώπιση των ζητημάτων ασφάλειας

Τα δίκτυα κινητής τηλεφωνίας της πρώτης γενιάς ήταν πλήρως αναλογικά, γεγονός που απέτρεπε την εφαρμογή κάποιου κρυπτογραφικού μηχανισμού.

Έτσι, είχαμε τα εξής προβλήματα:

- Η ασύρματη μετάδοση κάθε συνομιλίας διεξαγόταν ανοιχτά χωρίς κανενός είδους κρυπτογράφηση.
- Δεν υπήρχε κάποια μέθοδος εξακρίβωσης της ταυτότητας του συνδρομητή που χρησιμοποιούσε το δίκτυο.
- Οι τηλεφωνικές συσκευές που ήταν συμβατές με τα πρώτα δίκτυα κινητής τηλεφωνίας ήταν ευάλωτες σε επιθέσεις reverse engineering.

Η αντιμετώπιση των ζητημάτων ασφάλειας II

Με τη δεύτερη γενιά των δικτύων κινητής τηλεφωνίας περάσαμε από τον αναλογικό στον ψηφιακό τρόπο μετάδοσης, επιτρέποντας πλέον την εφαρμογή μηχανισμών για την προστασία των επικοινωνιών.

Το μοντέλο που χρησιμοποιήθηκε για την ασφάλεια του δικτύου GSM αντιμετώπισε με τον καλύτερο τρόπο τα ζητήματα όπως:

- Η πιστοποίηση της ταυτότητας των συνδρομητών.
- Η διατήρηση της ανωνυμίας των συνδρομητών.
- Η διαφύλαξη της εμπιστευτικότητας των επικοινωνιών.

Υπήρξαν όμως και σημαντικά τρωτά σημεία, όπως:

- Η σχεδίαση των χρησιμοποιούμενων αλγορίθμων.
- Η μονόπλευρη αυθεντικοποίηση.

Η αντιμετώπιση των ζητημάτων ασφάλειας ΙΙΙ

Η ασφάλεια των δικτύων της τρίτης γενιάς κινητής τηλεφωνίας είναι σε ακόμα καλύτερο επίπεδο, αφού μετά από μια δεκαετία λειτουργίας έχουν βρεθεί ορισμένα τρωτά σημεία, από τα οποία όμως δεν προκύπτουν πρακτικές επιθέσεις.

Όμως δεν παύουν να υπάρχουν περιπτώσεις όπου παραβιάζεται η ιδιωτικότητα των χρηστών κινητής τηλεφωνίας, καθώς:

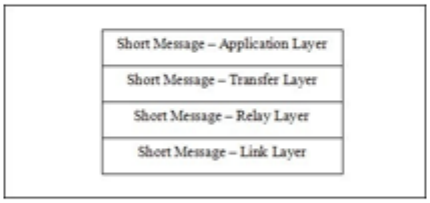
- Τα δίκτυα τρίτης γενιάς λειτουργούν παράλληλα με αυτά της δεύτερης.
- Δεν ακολουθείται παντού η ίδια πολιτική στην διασφάλιση των κινητών επικοινωνιών.

Έτσι δημιουργείται η ανάγκη ύπαρξης λύσεων με τις οποίες κρυπτογραφούνται η φωνή και τα σύντομα μηνύματα πριν από τη μετάδοση τους στο δίκτυο.

Ασφαλής μετάδοση σύντομων μηνυμάτων

Για τις ανάγκες της πτυχιακής εργασίας αναπτύχθηκε μια εφαρμογή που παρέχει την από άκρο εις άκρο κρυπτογράφηση (end-to-end encryption) ενός σύντομου μηνύματος.

Για τη μετάδοση σύντομων μηνυμάτων χρησιμοποιείται η αντίστοιχη στοίβα πρωτοκόλλων.



Η δικιά μας λύση πάνω στον τομέα της ασφάλειας των κινητών επικοινωνιών λειτουργεί στο στρώμα εφαρμογής, όπου εκτελείται μια εφαρμογή που προορίζεται για την πλατφόρμα Java Micro Edition.

Java Micro Edition

Το περιβάλλον εκτέλεσης εφαρμογών Java Micro Edition συναντάται στο μεγαλύτερο ποσοστό των συσκευών κινητής τηλεφωνίας που κυκλοφορούν στην αγορά και δομείται από ένα σύνολο APIs που υποστηρίζονται:

- Υποχρεωτικά από όλες τις συμβατές συσκευές, όπως:
  - Ο σχεδιασμός του γραφικού περιβάλλοντος.
  - Η αποθήκευση δεδομένων.
- Προαιρετικά, από τον εκάστοτε κατασκευαστή, όπως:
  - Οι υλοποιήσεις κρυπτογραφικών αλγορίθμων.
  - Η αποστολή σύντομων μηνυμάτων.

Η αναπτυχθείσα εφαρμογή απαιτεί μόνο την υποστήριξη του προαιρετικού Wireless Messaging API(WMA), γεγονός που την επιτρέπει την εκτέλεση της σε σχεδόν όλες τις συσκευές που είναι συμβατές με το περιβάλλον Java Micro Edition.

Ασφαλής μετάδοση σύντομων μηνυμάτων 9

Η λειτουργία της αναπτυχθείσας εφαρμογής

Η εφαρμογή που αναπτύξαμε διασφαλίζει το περιεχόμενο SMS μηνυμάτων μέσω της συνεργασίας των εξής στοιχείων:

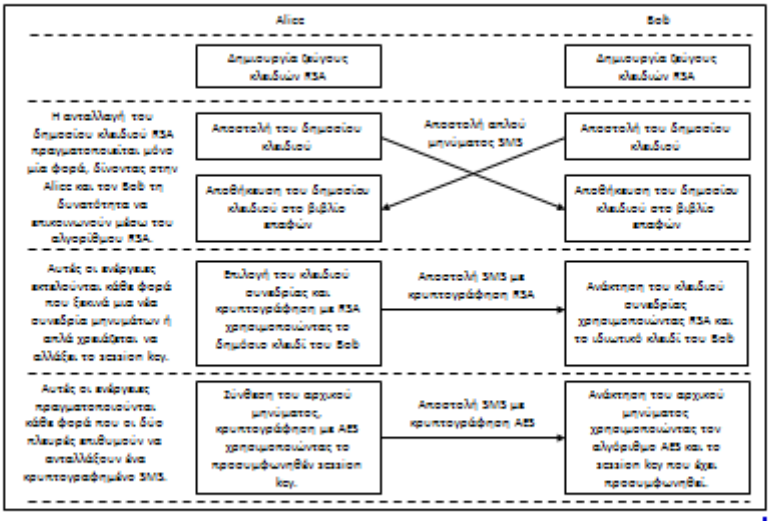
- Το συμμετρικό αλγόριθμο κρυπτογράφησης AES
- Τον αλγόριθμο κρυπτογράφησης δημοσίου κλειδιού RSA
- Τη συνάρτηση κατακερματισμού SHA-256
- Τη συνάρτηση παραγωγής κλειδιών PBKDF-2

Δύο χρήστες της εφαρμογής, έστω ότι ονομάζονται Alice και Bob, συμμετέχουν στις ακόλουθες φάσεις λειτουργίας:

- Δημιουργία ζεύγους κλειδιών για τον αλγόριθμο RSA
- Διαχείριση της λίστας επαφών
- Ανταλλαγή δημοσίου κλειδιού για τον αλγόριθμο RSA
- Ανταλλαγή κλειδιού συνεδρίας
- Ανταλλαγή κρυπτογραφημένων σύντομων μηνυμάτων

Ασφαλής μετάδοση σύντομων μηνυμάτων 10

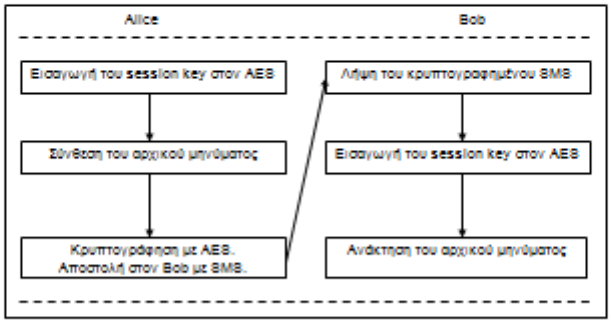
Η λειτουργία της αναπτυχθείσας εφαρμογής ΙΙ



Ανταλλαγή κρυπτογραφημένων μηνυμάτων

Η ανταλλαγή κρυπτογραφημένων σύντομων μηνυμάτων είναι εφικτή χάρις στο συμμετρικό αλγόριθμο κρυπτογράφησης AES.

Οι δύο πλευρές οφείλουν να χρησιμοποιήσουν το ίδιο κλειδί συνεδρίας με μήκος 256 δυαδικά ψηφία.



Ανταλλαγή κρυπτογραφημένων μηνυμάτων II

Για να στείλει η Alice ένα μήνυμα στο Bob ξεκινά:

1. Με την αντίστοιχη επιλογή από το μενού.
2. Εισάγει το κλειδί των 256 bit.
3. Πληκτρολογεί το κείμενο που επιθυμεί.
4. Επιλέγει την κρυπτογράφηση του.
5. Από τον κατάλογο επαφών επιλέγει τον παραλήπτη.

Ο Bob θα ανακτήσει το αρχικό μήνυμα αφού:

1. Ειδοποιηθεί για το νέο μήνυμα.
2. Εισάγει το ίδιο κλειδί με την Alice.

Προκύπτουν τα ακόλουθα ζητήματα για τους χρήστες:

- Πως διατηρούν το κλειδί συνεδρίας εμπιστευτικό;
- Πως προκύπτει το χρησιμοποιούμενο κλειδί;

Ασφαλής μετάδοση σύντομων μηνυμάτων 13

Ανταλλαγή κρυπτογραφημένων μηνυμάτων III

Η αποστολή του μηνύματος από την πλευρά της Alice.

Η ανάκτηση του αρχικού μηνύματος από την πλευρά του Bob.

Ασφαλής μετάδοση σύντομων μηνυμάτων 14

Δημιουργία του κλειδιού συνεδρίας

Το ζήτημα της δημιουργίας του κλειδιού συνεδρίας λύνεται με τη συνάρτηση παραγωγής κλειδιών Password Based Key Derivation Function 2 (PBKDF2), για τη λειτουργία της οποίας απαιτείται η εισαγωγή:

- Μιας κωδικής λέξης (password).
- Μιας αριθμητικής τιμής (salt).

Τα στοιχεία αυτά προωθούνται στη συνάρτηση κατακερματισμού SHA-256, η οποία εκτελείται για μια συγκεκριμένη σειρά επαναλήψεων και παράγει μια σύνοψη 256 bit.

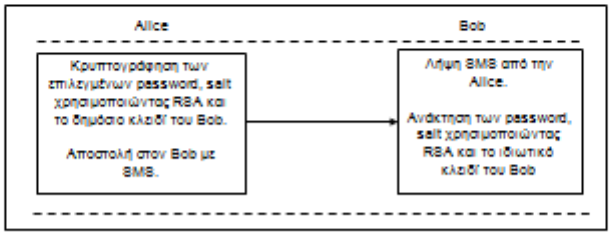
Κάθε φορά που απαιτείται η δημιουργία ενός συμμετρικού κλειδιού εμφανίζεται στους χρήστες της εφαρμογής μια οθόνη στην οποία εισάγουν τα στοιχεία password και salt και ξεκινούν τη διαδικασία.

Ασφαλής ανταλλαγή του κλειδιού συνεδρίας

Η διασφάλιση του κλειδιού συνεδρίας πραγματοποιείται μέσω του αλγορίθμου δημοσίου κλειδιού RSA.

Σύμφωνα με τον ορισμό του RSA:

- Σε κάθε χρήστη αντιστοιχεί ένα ζεύγος συσχετιζόμενων κλειδιών, με το ένα τμήμα να γίνεται δημοσίως γνωστό και το άλλο να διατηρείται μυστικό.
- Μια ενέργεια αναιρείται μόνο από το συσχετιζόμενο της κλειδί.



Ασφαλής ανταλλαγή του κλειδιού συνεδρίας II

Η διαδικασία με την οποία ο Bob ενημερώνει την Alice σχετικά με τα στοιχεία που θα δημιουργήσουν το προσωρινό κλειδί ξεκινά:

1. Με την αντίστοιχη επιλογή από το μενού της εφαρμογής.
2. Με τη συμπλήρωση της φόρμας με τα password, salt.
3. Με την αποστολή μηνύματος στην επαφή που επιλέγεται από τον κατάλογο.

Στην άλλη πλευρά, η Alice λαμβάνει το μήνυμα που έστειλε ο Bob, με τα password, salt να εμφανίζονται στο τηλέφωνο της.

Σε αυτή τη φάση της εφαρμογής αποκρύπτεται από τους χρήστες η χρησιμοποίηση του RSA, αφού είναι ανθρωπίνως αδύνατο να τα διαχειριστεί κανείς τα πολύ μεγάλα κλειδιά που εμπλέκονται.

Για τη διευκόλυνση των χρηστών, αλλά και για την καλύτερη λειτουργία της εφαρμογής, δημιουργούμε στην τηλεφωνική συσκευή τις ακόλουθες συλλογές εγγραφών:

- Ένα ζεύγος κλειδιών του αλγορίθμου RSA.
- Έναν κατάλογο των χρηστών της εφαρμογής.

Ασφαλής μετάδοση σύντομων μηνυμάτων 17

Ασφαλής ανταλλαγή του κλειδιού συνεδρίας III

Η αποστολή του κλειδιού συνεδρίας από τον Bob.

Η λήψη του κλειδιού από την πλευρά της Alice.

Ασφαλής μετάδοση σύντομων μηνυμάτων 18

Δημιουργία ζεύγους κλειδιών για τον RSA

Οι δύο συλλογές εγγραφών δημιουργούνται αυτομάτως κατά την πρώτη εκκίνηση της εφαρμογής, ενώ παράλληλα δημιουργείται και αποθηκεύεται σε μία από αυτές το ζεύγος κλειδιών που θα αντιστοιχεί στο χρήστη.

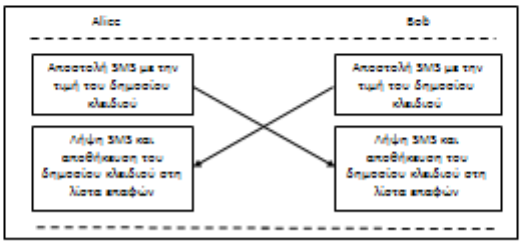
Με το πέρας της διαδικασίας ο χρήστης της εφαρμογής είναι υποχρεωμένος να διαχειριστεί τις επαφές του και να ανταλλάξει δημόσια κλειδιά με όσους χρήστες βρίσκονται σε αυτές, αλλιώς ο αλγόριθμος RSA δεν θα λειτουργεί, καθιστώντας αδύνατη την ασφαλή ανταλλαγή κλειδιών συνεδρίας για τον αλγόριθμο AES.

Ανταλλαγή δημοσίων κλειδιών για τον RSA

Η ανταλλαγή των δημοσίων κλειδιών RSA ανάμεσα σε δύο χρήστες της εφαρμογής πραγματοποιείται μόνο μια φορά, με την αποστολή του κλειδιού μέσω ενός απλού μηνύματος χωρίς κρυπτογράφηση.

Πριν από την αποστολή του μηνύματος πρέπει να έχει δημιουργηθεί μια επαφή με τα στοιχεία του παραλήπτη για τον οποίο προορίζεται το κλειδί.

Με τη δημιουργία μιας νέας επαφής ο αποστολέας του μηνύματος προετοιμάζει το έδαφος για την καταχώρηση του δημοσίου κλειδιού που θα έρθει από την πλευρά του παραλήπτη.



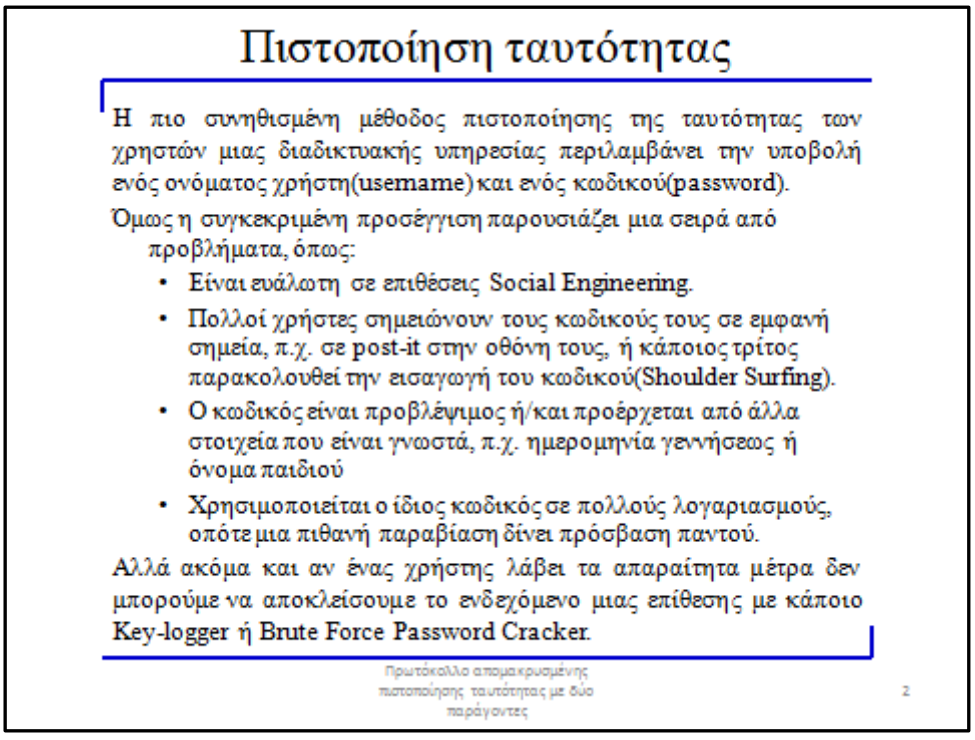
Ανταλλαγή δημοσίων κλειδιών για τον RSA II

Η αποστολή δημοσίου κλειδιού από την Alice στο Bob πραγματοποιείται με:

1. Την κατάλληλη επιλογή από το μενού της εφαρμογής.
2. Την επιλογή της επαφής του Bob από τη λίστα επαφών.

Η εφαρμογή που εκτελείται στη συσκευή του Bob αναγνωρίζει αυτομάτως τον τύπο του μηνύματος και προχωρά στην αποθήκευση του δημοσίου κλειδιού στη λίστα επαφών, ακόμα και αν η αντίστοιχη επαφή δεν προϋπάρχει.

Παρουσίαση πρωτοκόλλου για την απομακρυσμένη πιστοποίηση ταυτότητας ενός χρήστη



Πιστοποίηση με πολλαπλούς παράγοντες

Για να επιλυθούν τα προαναφερθέντα προβλήματα ακολουθούνται διάφορες παραλλαγές της μεθόδου πιστοποίησης ταυτότητας, κατά τις οποίες απαιτείται από το χρήστη η εισαγωγή περισσοτέρων στοιχείων που έχουν διαφορετική μορφή.

Τα στοιχεία αυτά μπορεί να έχουν μια από τις ακόλουθες μορφές:

- Να είναι κάτι που το γνωρίζει μόνο ο χρήστης, όπως είναι ένας κωδικός ή ένας αριθμός pin.
- Να είναι κάτι που έχει στην κατοχή του ο χρήστης, όπως είναι μια μαγνητική κάρτα ή ένα κλειδί.
- Να είναι κάποιο χαρακτηριστικό του χρήστη, όπως είναι τα βιομετρικά δεδομένα.

Ο συνδυασμός δύο εκ των ανωτέρω στοιχείων χαρακτηρίζεται ως πιστοποίηση ταυτότητας με δύο παράγοντες, όπως είναι η πρόσβαση σε ένα λογαριασμό τραπέζης μέσω ATM, όπου ο χρήστης εισάγει την κάρτα στο μηχάνημα(Κάτι που έχει στην κατοχή του) και πληκτρολογεί το pin(Κάτι που γνωρίζει).

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 3

Σκυτάλες ασφάλειας

Σαν σκυτάλη ασφάλειας ορίζεται μια συσκευή που βρίσκεται στην κατοχή ενός χρήστη και χρησιμοποιείται μαζί με έναν ακόμη παράγοντα για να πραγματοποιηθεί η πιστοποίηση ταυτότητας.

Συνήθως η σκυτάλη περιέχει αποθηκευμένο κάποιο κρυπτογραφικό κλειδί ή/και έχει τη δυνατότητα εκτέλεσης μιας συνάρτησης παραγωγής κωδικών.

Ακόμη, ενδέχεται πριν από την ενεργοποίηση της να απαιτεί την εισαγωγή κάποιου pin ή ενός κωδικού, προστατεύοντας με αυτό τον τρόπο τον κάτοχο σε πιθανή απώλεια της.

Υπάρχουν αρκετοί τρόποι με τους οποίους υλοποιείται μια σκυτάλη, με πιο δημοφιλείς μορφές αυτές της:

- Έξυπνης Κάρτας(Smart Card)
- Συσκευής USB Stick
- Θήκης Κλειδιών με δυνατότητα δημιουργίας κωδικών

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 4

Σκυτάλες ασφάλειας II

Σκυτάλες ασφάλειας III

Όμως, σε όλες τις προαναφερθείσες περιπτώσεις ο χρήστης υποχρεώνεται να έχει μαζί του μία παραπάνω συσκευή, το κόστος της οποίας περνά τελικώς σε αυτόν και τον επιβαρύνει οικονομικά.

Γι' αυτό και τον τελευταίο καιρό εμφανίζονται ολοένα και πιο συχνά σκυτάλες ασφάλειας που λαμβάνουν τη μορφή λογισμικού το οποίο είναι συμβατό με τις πλατφόρμες εφαρμογών που συναντάμε σε συσκευές κινητών τηλεφώνων.

Αυτή η προσέγγιση παρουσιάζει μια σειρά από πλεονεκτήματα, όπως:

- Το γεγονός ότι ο χρήστης έχει ήδη στην κατοχή του το κινητό τηλέφωνο, οπότε δεν χρειάζεται να χρεωθεί κάτι παραπάνω.
- Δεν απαιτείται να μεταφέρει μια επιπλέον συσκευή ή μια κάρτα στο πορτοφόλι του.
- Το κινητό είναι ανεξάρτητο από το PC του χρήστη, οπότε δεν επηρεάζεται από κακόβουλο λογισμικό του ιδίου τύπου.

Σκυτάλες ασφάλειας IV

Τα κυριότερα μειονεκτήματα που εμφανίζονται στις σκυτάλες ασφάλειας – κινητά τηλέφωνα είναι:

- Ότι δεν έχουν όλοι οι χρήστες τον ίδιο τύπο κινητού τηλεφώνου, οπότε η εγκατάσταση μιας κοινής εφαρμογής γίνεται δυσκολότερη ή ακόμα και ακατόρθωτη.
- Ενδεχόμενες φθορές στη συσκευή μπορεί να την καταστήσουν ανίκανη να λειτουργεί ως σκυτάλη.
- Πιθανή αμέλεια του χρήστη να φορτίσει την μπαταρία του τηλεφώνου μπορεί σε κάποια περίπτωση ανάγκης να του στερήσει τη δυνατότητα να χρησιμοποιήσει τη σκυτάλη.
- Η κάλυψη του δικτύου να μην είναι ικανοποιητική και το σήμα που λαμβάνει η συσκευή να μην είναι ισχυρό.
- Σε μια συγκεκριμένη στιγμή μπορεί να υπάρχει αρκετή κίνηση στο δίκτυο κινητής τηλεφωνίας με αποτέλεσμα να παρατηρούνται καθυστερήσεις στις παραδόσεις σύντομων μηνυμάτων.

Πρωτόκολλο πιστοποίησης ταυτότητας

Το αναπτυχθέν πρωτόκολλο βασίζεται στο μοντέλο πελάτη-εξυπηρετητή (Client-Server Model), έτσι όλα εκτυλίσσονται σύμφωνα με ένα σενάριο στο οποίο συμμετέχουν δύο οντότητες:

- Η μια οντότητα επιθυμεί αρχικά να πιστοποιήσει την ταυτότητα της και εν συνεχεία να εισέλθει σε ένα περιβάλλον με περιορισμούς πρόσβασης. Ένα τέτοιο παράδειγμα είναι ο πελάτης μιας web-banking υπηρεσίας.
- Η δεύτερη οντότητα είναι εκείνη που φιλοξενεί ένα περιβάλλον στο οποίο επιτρέπει την πρόσβαση μόνο σε όσους έχουν την ανάλογη εξουσιοδότηση. Τυπικό παράδειγμα ο εξυπηρετητής που χρησιμοποιείται για τις ηλεκτρονικές συναλλαγές των πελατών μιας τράπεζας.

Για τις ανάγκες της εργασίας αναπτύχθηκε μια υπηρεσία στην οποία ο χρήστης που πιστοποιεί την ταυτότητα του αποκτά πρόσβαση σε μια ιστοσελίδα με προστατευμένο περιεχόμενο.

Πρωτόκολλο πιστοποίησης ταυτότητας II

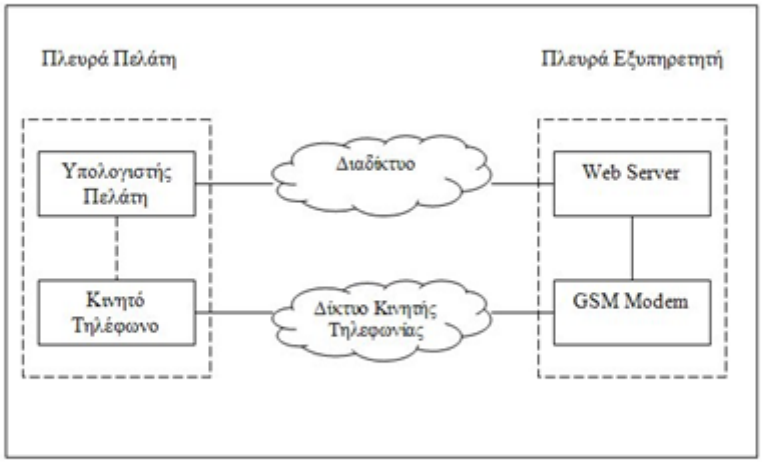
Εφόσον η πιστοποίηση ταυτότητας βασίζεται σε δύο παράγοντες έχουμε την πλευρά του πελάτη να καλείται να αποδείξει ότι γνωρίζει το σωστό μυστικό κωδικό και παράλληλα έχει στην κατοχή της την κατάλληλη σκυτάλη ασφάλειας.

Η επικοινωνία ανάμεσα στις οντότητες που συμμετέχουν στο πρωτόκολλο πραγματοποιείται μέσα από δύο διαφορετικά κανάλια επικοινωνίας, όπου στην προκειμένη περίπτωση είναι το Internet και το Δίκτυο Κινητής Τηλεφωνίας.

Η επαλήθευση της σκυτάλης του πελάτη επιτυγχάνεται μέσω ενός συνδυασμού των τεχνικών One-Time Password και Challenge-Response.

Ένα Challenge δημιουργείται μόλις ο πελάτης αποδείξει ότι γνωρίζει το σωστό μυστικό κωδικό. Η σκυτάλη του χρήστη και ο εξυπηρετητής μοιράζονται κάποια στοιχεία που τους επιτρέπουν να παράγουν το One Time Password, το οποίο χρησιμοποιούν για την κρυπτογράφηση του Challenge και από εκεί προκύπτει το ανάλογο Response.

Πρωτόκολλο πιστοποίησης ταυτότητας III



Η πλευρά του εξυπηρετητή

Στην πλευρά του εξυπηρετητή βρίσκονται:

- Ο Web Server από τον οποίο παρέχεται η διαδικτυακή υπηρεσία, όπου στην προκειμένη περίπτωση έχουμε τον HTTP Server Apache Tomcat.
- Το RDBMS της MySQL, για να διατηρούνται τα δεδομένα των εγγεγραμμένων χρηστών.
- Το GSM-Modem που διαχειρίζεται την κίνηση σύντομων μηνυμάτων, όπου για λόγους οικονομίας έχουμε συνδέσει ένα κινητό τηλέφωνο με δυνατότητα εκτέλεσης εντολών AT.

Εκμεταλλευόμενοι το γεγονός ότι ο Tomcat είναι υλοποιημένος σε Java έχουμε τη δυνατότητα να του προσθέσουμε στοιχεία, όπως:

- Κρυπτογραφικές κλάσεις, οι οποίες τρέχουν και στη σκυτάλη
- Driver MySQL Connector/J
- Java Communications API

Η πλευρά του εξυπηρετητή II

Η διαδικτυακή υπηρεσία έχει αναπτυχθεί με την ακόλουθη λογική:

- Μέσω των σελίδων JSP παρέχεται στους πελάτες μια διασύνδεση για να εγγράφονται και να εισέρχονται στη διαδικτυακή υπηρεσία.
- Μια σειρά από κλάσεις Servlets αναλαμβάνουν τον έλεγχο και την επεξεργασία των δεδομένων που υποβάλλουν οι πελάτες.
- Ακόμη, τα Servlets αναλαμβάνουν την επικοινωνία με τα υπόλοιπα τμήματα, όπως:
  - Την υποβολή ερωτημάτων στη Βάση Δεδομένων και την επιστροφή των αποτελεσμάτων στους πελάτες.
  - Την αναζήτηση και επεξεργασία νέων εισερχομένων μηνυμάτων από το GSM-Modem, καθώς και την προετοιμασία και προώθηση SMS που θα σταλούν από αυτό.

Η πλευρά του πελάτη

Στην πλευρά του πελάτη συναντάμε:

- Τον υπολογιστή με τον οποίο αποκτά πρόσβαση στο Internet.
- Το κινητό τηλέφωνο σκυτάλη που έχει στην κατοχή του.

Για τον υπολογιστή του πελάτη δεν υπάρχουν φοβερές απαιτήσεις από πλευράς hardware και software. Μοναδική απαίτηση είναι η παρουσία μιας όσο το δυνατόν νεότερης έκδοσης ενός browser.

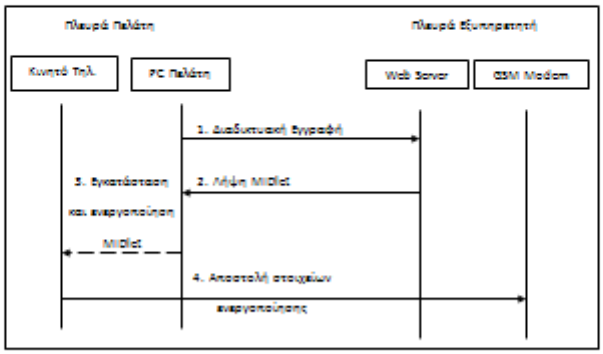
Με μια επίκαιρη έκδοση ενός browser έχουμε:

- Την καλύτερη λειτουργία των ασφαλών συνδέσεων SSL-TLS
- Καλύτερη προστασία από παλαιότερες ευπάθειες του προγράμματος

Όσον αφορά το κινητό τηλέφωνο που θα χρησιμοποιηθεί ως σκυτάλη ασφάλειας ούτε εκεί απαιτείται κάποια εξεζητημένη λύση. Το μοναδικό χαρακτηριστικό που πρέπει να παρέχεται από τη συσκευή είναι μια πλατφόρμα Java Micro Edition που θα είναι συμβατή με την αναπτυχθείσα εφαρμογή.

Πριν από τη συμμετοχή στο πρωτόκολλο

Για τη συμμετοχή του πελάτη στο πρωτόκολλο απαιτείται η εγγραφή του στην αναπτυχθείσα διαδικτυακή υπηρεσία και η εγκατάσταση της εφαρμογής που θα μετατρέψει το κινητό του σε σκυτάλη ασφάλειας. Οι απαιτούμενες ενέργειες παρουσιάζονται συνοπτικά στην εικόνα που ακολουθεί.



Έγγραφη στη διαδικτυακή υπηρεσία

Ο πελάτης επισκέπτεται την αρχική σελίδα και ακολουθεί το σύνδεσμο που παραπέμπει στη εγγραφή στην υπηρεσία. Σε εκείνο το σημείο συμπληρώνει την κατάλληλη φόρμα και αποκτά πρόσβαση στο MIDlet που πρόκειται να εγκαταστήσει στο κινητό του.

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 15

Εγκατάσταση του συνοδευτικού MIDlet

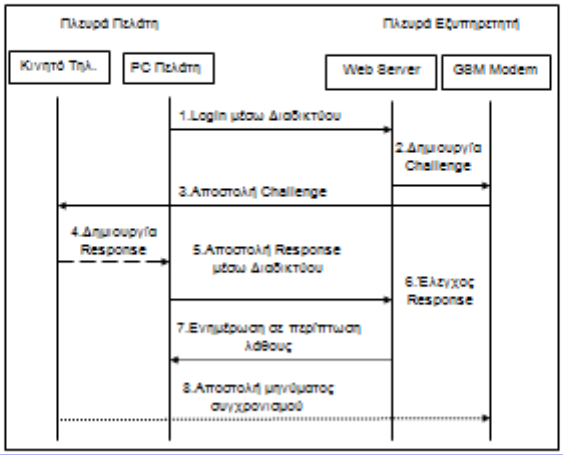
Ο πελάτης, χρησιμοποιώντας μια από τις διαθέσιμες μεθόδους διασύνδεσης, μεταφέρει το MIDlet από τον υπολογιστή του και το εγκαθιστά στο κινητό του τηλέφωνο.

Κατά την πρώτη εκτέλεση της εφαρμογής ο πελάτης καλείται να στείλει στην πλευρά του εξυπηρετητή τα στοιχεία με τα οποία θα ενεργοποιεί τη σκυτάλη ασφάλειας.

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 16

Παρουσίαση του πρωτοκόλλου σε λειτουργία

Το πρωτόκολλο τίθεται σε λειτουργία με τον πελάτη να εισέρχεται στη διαδικτυακή υπηρεσία. Τότε ο εξυπηρετητής προωθεί μια πρόκληση στον πελάτη και εκείνος πρέπει να δώσει την κατάλληλη απάντηση.



Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 17

Παρουσίαση του πρωτοκόλλου σε λειτουργία II

Ο πελάτης εισέρχεται στη διαδικτυακή υπηρεσία και, εφόσον τα διαπιστευτήρια του είναι σωστά, προωθείται στη σελίδα όπου θα εισάγει την απάντηση που δημιουργείται από τη σκυτάλη ασφάλειας. Παράλληλα ενεργοποιεί το MIDlet που είναι εγκατεστημένο στο κινητό του και περιμένει την πρόκληση που θα του αποστείλει ο εξυπηρετητής.

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες 18

Παρουσίαση του πρωτοκόλλου σε λειτουργία III

Το κινητό τηλέφωνο ειδοποιεί τον πελάτη για το μήνυμα που έλαβε και με αυτό το γεγονός ξεκινά η δημιουργία του One-Time Password. Το συγκεκριμένο στοιχείο έχει μήκος 256 δυαδικών ψηφίων και αποτελεί το κλειδί κρυπτογράφησης του Challenge με τον AES-256. Το κρυπτογράφημα που προκύπτει αντιπροσωπεύει την αρχική τιμή του Response, όμως η μορφή του δεν είναι καθόλου πρακτική.

Έτσι, ακολουθώντας την κάτωθι διαδικασία το τελικό Response καταλαμβάνει μήκος 8 δεκαεξαδικών χαρακτήρων:

1. Το αρχικό Response χωρίζεται σε 8 ομάδες των 4 χαρακτήρων.
2. Απομονώνονται τα τελευταία 8 ψηφία του Challenge και αντιστρέφεται η σειρά τους.
3. Σύμφωνα με τη θέση του κάθε ψηφίου του αντιστοιχεί και μια από τις τετράδες χαρακτήρων.
4. Από κάθε μια από τις τετράδες λαμβάνεται ο χαρακτήρας που βρίσκεται στη θέση που δίνεται από την πράξη (Ψηφίο mod 4).
5. Το τελευταίο βήμα εκτελείται 8 φορές, σχηματίζοντας έτσι ένα νέο Response με μήκος 8 χαρακτήρων.

Παρουσίαση του πρωτοκόλλου σε λειτουργία IV

Με το ακόλουθο παράδειγμα θα επιχειρήσουμε να κάνουμε περισσότερο κατανοητή την προαναφερθείσα διαδικασία.

Έστω ότι στο κινητό του πελάτη έχουμε μετρητή 0 και αποθηκευμένο hash:
`1f637163c029ded38ef0a0d2691dee038feb959cdba1b548f3a70d24c381bd32`
Από τον εξυπηρετητή αποστέλλεται το Challenge:`1269209815328`
Οι δύο πλευρές παράγουν το One-Time Password:
`d241683d769ecc76182fabe0578e039b549193c483e2659bfd7d40c4762e0147`
Χρησιμοποιώντας AES-256 με κλειδί το παραγόμενο One-Time Password κρυπτογραφούμε το Challenge και προκύπτει η αρχική τιμή του Response:
`e5e8f7b5df89515146919d802c53e758`

Αρχικό Response	e5e8	f7b5	df89	5151	4691	9d80	2c53	e758
Τελευταία οκτάδα Challenge	0	9	8	1	5	3	2	8
Αντιστροφή οκτάδας	8	2	3	5	1	8	9	0
Ψηφίο mod 4	8mod4	2mod4	3mod4	5mod4	1mod4	8mod4	9mod4	0mod4
Χαρακτήρας που λαμβάνεται	0	2	3	1	1	0	1	0
Τελικό Response	a	b	9	1	6	9	c	a

Παρουσίαση του πρωτοκόλλου σε λειτουργία V

Το συμπιεσμένο Response έχει μήκος 8 χαρακτήρων και εμφανίζεται με αυτή τη μορφή στην οθόνη του κινητού τηλεφώνου του πελάτη. Εκείνος με τη σειρά του υποβάλει την απάντηση στη διαδικτυακή υπηρεσία, και αν αυτή συμφωνεί με αυτή που προβλέπεται, τότε προωθείται στη σελίδα με περιορισμένη πρόσβαση.
Ειδάλλως, ο πελάτης ενημερώνεται για τη λανθασμένη απάντηση και έχει τη δυνατότητα να την διορθώσει ή να αποστείλει ένα μήνυμα συγχρονισμού με τον εξυπηρετητή και να εκκινήσει νέα απόπειρα πιστοποίησης της ταυτότητας του.

Πρωτόκολλο απομακρυσμένης πιστοποίησης ταυτότητας με δύο παράγοντες

21

Παράρτημα 6 - Δημοσιεύσεις

Δημοσίευση της εφαρμογής για την ασφαλή ανταλλαγή μηνυμάτων

Secure Wireless P2P Messaging

Manifavas H., Sarris P.
Department of Applied Informatics and Multimedia
Faculty of Applied Technology
Technological Educational Institute of Crete
Estavromenos 715 00 Heraklion, Crete - Greece
harryman@epp.teicrete.gr, parsarr@gmail.com

ABSTRACT
This paper describes how a software application forms an additional layer of security that allows us to exchange ciphered short messages over a wireless peer-to-peer network, which in this case is a mobile phone network.

Keywords
Wireless Security, Short Messaging Service, SMS, Symmetrical Encryption Algorithm, AES, Public Key Cryptography, RSA, Java Micro Edition, Java ME, Hash function, SHA256.

1. INTRODUCTION
During the last couple of decades we have witnessed the exponential growth of the global mobile phone market, as billions of subscribers own at least one mobile phone device that gives them access to a mobile phone network.

In any form of communication, either by talking or texting, users exchange private information with the risk of having their conversations eavesdropped, as shown in [1].

In addition to that, extensive research on the sector of mobile phone networks security, as in [2] and [3], has proven the existence of a series of vulnerabilities that put users' privacy at stake. This fact makes even greater the need for a solution, like the one described in the following pages, dealing with these problems.

2. SHORT MESSAGING SERVICE
The exchange of text messages has been extremely popular among the mobile phones users, especially for the younger ones, as more than a trillion of messages are sent every year. According to [4] this milestone was reached in the year 2005. This type of communication is achieved through a protocol stack that consists of four layers and is implemented in every mobile phone device and some elements of the respective phone network.

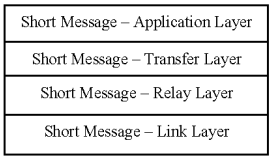


Figure 1: Short Messaging Service Protocol Stack.

Our security solution comes in the form of an application that is written in the Java programming language and is adjusted to its Micro Edition specifications. This means that our application resides on the respective upper layer of the protocol stack and interfaces with the lower layers when a message needs to be relayed.

3. APPLICATION'S REQUIREMENTS
As we mentioned before, the secure messaging application is implemented according to the Java Micro Edition (Java ME) specifications [5], this means that it may be deployed only on devices equipped with the respective application platform.

We should also mention the fact that the Java ME platform includes a set of Application Programming Interfaces (APIs) which are optionally supported by the devices' manufacturers.

Our secure messaging application requires the presence of the optional Wireless Messaging API, which is the one that enables the application to exchange short messages with the other communicating parts.

4. THE APPLICATION'S STRUCTURE
As we can see from the following figure there are five distinct phases in the application's lifecycle that lead to our goal, which is the secure exchange of short messages.

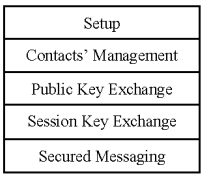


Figure 2: Actions required for secure messaging.

Some of the aforementioned actions occur just once, like in the case of the Setup phase, while others happen more frequently depending on every user's communicational needs.

4.1 Setting up the application
The first phase, labeled as "Setup", is carried out only when the application gets installed on a mobile phone.

In this stage the application has to perform a couple of tasks that will create the proper environment for the remaining phases.

One of these tasks is the creation of the tables that will store essential data, such as the contacts' numbers and their RSA Public Keys.

The other task lays the groundwork for the proper function of the RSA Public Key Algorithm, by generating a series of coefficients that form the RSA Key Pair that is going to be used by the owner of the device.

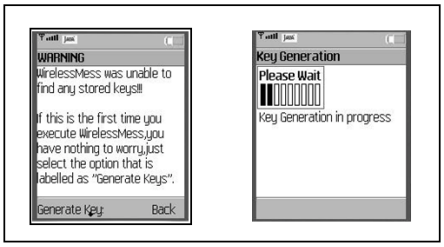


Figure 3: Screen sequence that appears during the setup phase

The key length we selected for our application is 1024 bits long. This size was chosen because it fits best for our purpose; while it provides adequate security it does not drain the limited and valuable resources of the hosting device.

4.2 Managing your contacts

Every user has the ability to add new contacts, edit the existing ones, or delete those that are not needed anymore. These options are accessible from the app's main menu, where we can select the item labeled "Manage your Contacts" and get transferred to a screen where we can choose the desired action from the provided sub-menu.

Before we describe the way that the aforementioned actions interact with the data stored in the user's device it should be better to present the way these data are structured in each record found in the contacts' list.

Every contact consists of four fields and has the same structure as the one appearing at the following figure.

ID	Name	Phone_Number	RSA_Public_Key

Figure 4: The structure applied on every contact.

Now that we have a conceptual view of a record we may start the analysis of the actions that a user can perform on his contacts.

The first and perhaps the simplest case is the deletion of a contact. The process starts when the user picks the "Delete" option from the sub-menu found in the Contacts' Management screen. He is then transferred to a screen with a list of all his contacts, from where he selects which one he wishes to remove.

In this case the Record Management System that operates on the user's device removes from the current record store all the fields that form the selected record.

The next case we examine is the editing of an existing contact. Once again the user is in the Contacts' Management screen, where he selects the "Edit" option from the respective sub-menu. This action leads him to a list of all his contacts so he can choose the one that needs editing.

During this stage the user can only change the values of the "Name" and "Phone_Number" fields, while the rest two fields remain inaccessible to the user. This feature is extremely useful, in a scenario that we will see in the following lines, when a contact is automatically stored by the application.

The last case we analyze is the addition of a new contact. The user selects the "Add" option from the Contact's Management sub-menu and gets transferred to a screen where he inputs the name and the phone number for the new contact. The two remaining fields are not managed by the user.

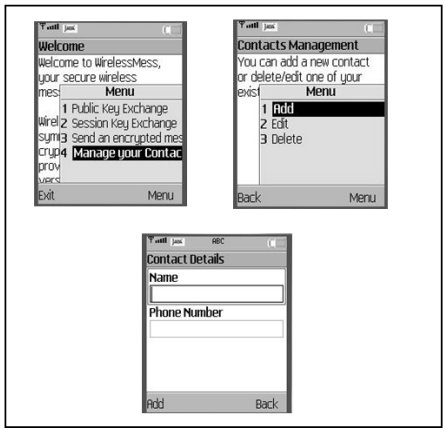


Figure 5: Screen sequence during the addition of a new contact

The value for the "ID" field is automatically assigned by the Record Management System which is operating on the Java ME application environment found on the user's device.

The "RSA_Public_Key" field gets initialized and then it is updated by an SMS sent from the respective contact during the public key exchange phase. This is the point that may trigger the automatic addition of a contact.

If a public key value arrives at the user's device and the respective contact does not exist, then every distinctive and essential value, such as the phone number and the public key, gets automatically stored in a newly created record.

On this occasion the "Name" field gets a copy of the sender's phone number and afterwards, when the user finds out who sent him the message, he has the option of editing the contact's name.

4.3 Exchanging RSA public keys

This is the most essential phase in the lifecycle of the application because it affects the way that the RSA algorithm works, as it is described in [6], which in turn is extremely important for the other parts of the application.

This fact makes the exchange of public keys mandatory for every pair of communicating users. This step is carried out once and after its successful completion every user has the other side's public key stored in his device.

The required process is really simple and may easily take place after the addition of a new contact. In this stage the user selects the "Public Key Exchange" option from the application's main menu. In the next screen he picks the appropriate contact from the list that appears on his device and sends there his RSA public key.

The recipient of the message is obliged to reply, so he follows the same procedure and sends his own public key value to the user that initiated the key exchange.

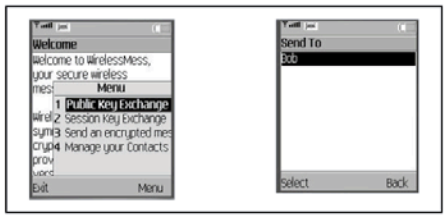


Figure 6: Screen sequence during the RSA public key exchange

Due to the nature of the RSA public key, which does not have to remain confidential, every message sent in this stage is transmitted in clear text.

4.4 Session key exchange

This the point where two sides agree on the key they are going to use in an upcoming encrypted messaging session. Every message in this session will be encrypted by the stream cipher AES-256.

This version of the symmetric algorithm requires a key that is 256 binary digits long, or the equivalent form of 64 hexadecimal digits. The problem with a key of this length is that it is impossible to memorize it and input it manually without making any mistakes.

The solution to this issue comes from the second version of the Password-Based Key Derivation Function, or PBKDF2.

According to this function's specifications [7], we can generate a larger key code from a short password and a numerical salt value. These two elements are inserted in a hash function which executes for a certain number of iterations and results in a hash value that represents the derived session key.

The underlying hash function that is used for the needs of key derivation function is SHA-256, which is a member of the second generation of the Secure Hash Algorithm family [8]. After a thousand of iterations the selected hash function generates a stream of 256 binary digits that represents the key that will be used in the current encrypted messaging session.

The procedure starts when a user selects the "Session Key Exchange" option from the application's main menu. In the next screen the user inserts the necessary elements in their respective text boxes and finally picks the user he wants to inform.

The password and salt values have to remain confidential; otherwise any third party knowing these values has the potential to reconstruct the session key and reveal the contents of the exchanged encrypted messages.

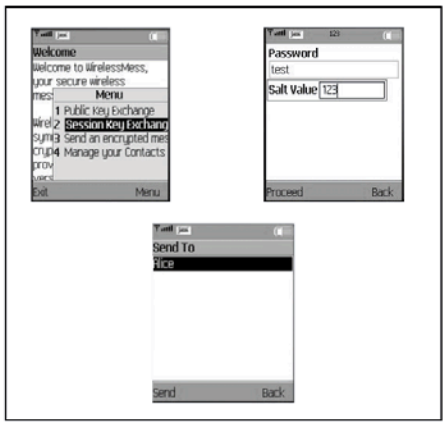


Figure 7: Screen sequence during the exchange of a session key

In order to protect the password and salt values we encrypt them by using the RSA algorithm. The ground for the encryption of this set of elements is prepared when the user selects the destination of his message.

In the background, the Record Management System retrieves the corresponding RSA public key and phone number and makes them available to the application, which afterwards uses the public key to encrypt the message and then forwards it to the provided number.

According to the RSA's specifications, every action performed by using a public key can only be inverted by using its correlated private key.

In our case this means that an encrypted message of this type may only be recovered by the user that originally generated the pair of RSA keys, who we assume that keeps his RSA private key secret and known only to him.

At the recipient's side the session key is decrypted in an automatic manner. With the Record Management System's assistance the application retrieves the RSA private key value and uses it with the RSA algorithm in order to recover the initial message.

If everything goes according to the plan then the user's device presents on its screen the name of the sender and the elements that will generate the key that will encrypt every message in the upcoming messaging session.

Figure 8: Screen that indicates the reception of a new session key

4.5 Exchanging encrypted short messages

The encryption and forwarding of short messages is the reason that led to the development of this application. This certain feature is available to any pair of users who have previously laid the foundation for secure communications, by exchanging their RSA public keys and ensuring that their session keys remain confidential.

A user who wishes to send an encrypted message has to select the option "Send an encrypted message" which is found on the application's main menu. Then he will be transferred to a screen where he generates the session key, by inserting on the password and salt value which were earlier agreed with the other user.

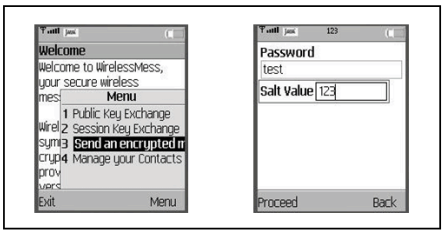


Figure 9: Screen sequence where a user initiates encryption and enters the session key

Now the session key for the symmetrical encryption algorithm AES-256 is generated, so the user gets redirected to the next screen, where he types the text that he wishes to cipher.

The encryption starts by pressing the appropriate button on the user's device. The typed text gets segmented into blocks of 128 bits, with every one of them getting individually encrypted with the current session key.

When the process is completed the user gets forwarded to a list of his contacts, where he selects the destination for his cipher-text message.

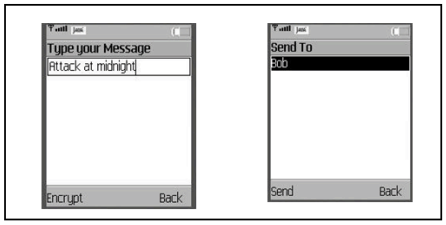


Figure 10: Screen sequence where the user types a clear-text message and then forwards the cipher-text to the selected contact

The recipient of the message has to follow a similar procedure that will invert the encryption. In the beginning, a notification, which contains information about the sender, appears on the device's screen. Then the user is forwarded to a screen where he types the password and salt that will generate the current session key.

According to the cipher's specifications [9], the user who wishes to decrypt the message has to use the same session key; therefore he has to enter the same password and salt values as the ones used during the encryption of the message. Once the user completes the typing then the clear-text is recovered and appears on the screen of his device.

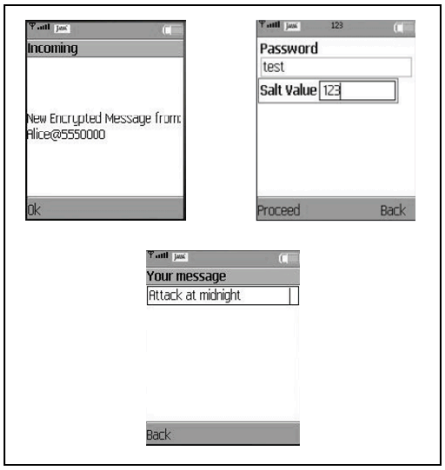


Figure 11: Screen sequence where a user receives an encrypted message, generated the session key by entering the required elements and gains access to the clear-text message

5. CONCLUSIONS

The developed application fulfills its goal and provides secure messaging for everyone who installs it on a mobile phone equipped with the Java Micro Edition Application Platform.

The process required for setting up the application and establishing a secure communications path between two parts might seem complex and perhaps disapproving for some users, but everything is absolutely necessary.

Otherwise, if we skip one of steps we mentioned above we render the application useless and we put the users' privacy at risk.

6. ACKNOWLEDGMENTS

We would like to thank every one of the researchers that we refer to in the next section. Without their pioneering ideas and the breakthroughs they achieved we would not be able to develop the concept that led to the secure messaging application.

7. REFERENCES

[1] K. Nohl, "GSM:SRLSY?," 27 December 2009. [Online]. Available: http://events.ccc.de/congress/2009/Fahrplan/attachments/1519_26C3.Karsten.Nohl.GSM.pdf.

[2] A. Biryukov, A. Shamir and D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC," in *Fast Software Encryption 2000*, Springer, 2001.

[3] O. Dunkelman, N. Keller and A. Shamir, "A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony," *IACR Cryptology ePrint Archive,* no. 13, 2010.

[4] GSM Association, "GSM & GSM Association History," [Online]. Available: http://www.gsma.com/history/.

[5] "Java Community Process," [Online]. Available: http://jcp.org/en/jsr/platform?listBy=1&listByType=platform.

[6] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* vol. 21, no. 1, pp. 120-126, 1978.

[7] RSA Laboratories, "PKCS #5: Password-Based Cryptography Standard," 25 March 1999. [Online]. Available: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf.

[8] National Institute of Standards and Technology, "FIPS 180-2 - Secure Hash Standard," 1 August 2002. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf.

[9] National Institute of Standards and Technology, "FIPS 197 - Advanced Encryption Standard," 26 November 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Δημοσίευση του πρωτοκόλλου για την απομακρυσμένη πιστοποίηση ταυτότητας ενός χρήστη

Two Factor Authentication Protocol

Manifavas C., Sarris P.
Department of Applied Informatics and Multimedia
Faculty of Applied Technology
Technological Educational Institute of Crete
Estavromenos 715 00 Heraklion, Crete - Greece
harryman@epp.teicrete.gr, parsarr@gmail.com

ABSTRACT
In this paper we describe the structure and the inner workings of a two factor authentication protocol, where a user, once he has proven his true identity, gains access to a webpage with secured content.

Keywords
Two Factor Authentication, 2FA, One-Time Password, OTP, Security token, Public Key Cryptography, Symmetrical Cryptography, RSA, AES, Secure Hash Algorithm, SHA-256, Java Micro Edition, J2ME, Java Server Pages, JSP, Java Servlets.

1. INTRODUCTION
The traditional approach to the matter of user authentication, where a user proves the knowledge of the right password, is no longer considered to be secure enough. Many users choose weak passwords which can be easily guessed or cracked and some of them are victims of phishing attacks that trick them into revealing their passwords. Also, we should mention that there is a wide variety of malware that has the ability to capture passwords and send them over to network attackers.

These are the main reasons why some methods that combine two or more factors of authentication gain in popularity and appear in even more applications and services.

The most commonly used authentication factors, apart from "Something the user knows", are the biometric characteristics ("Something the user is") and the security tokens ("Something the user owns").

Our proposed solution requires from the participant to the protocol to prove that he knows the correct password and that he owns the proper security token.

2. FORMS OF TOKEN
There are plenty of options on the market that can fulfill the "Something the user has" authentication factor. These forms of token can be separated into the following categories:

- Paper tokens. These could be a distributed list of "One-Time Passwords" or a grid of codes the user needs to enter in order to respond to a challenge.
- Soft tokens. These rely on a software component found on the user's computer, for example a cookie or a software application.
- Hardware tokens. These are physical devices that the user must possess. Typically, hardware tokens incorporate physical and logical mechanisms that protect their data and prevent them from being copied.

The fact that the hardware tokens have better protection against being copied gives them a great advantage when compared with the other types of tokens and makes them the best choice for a secure two factor authentication mechanism.

The most usual forms of hardware tokens appear on the following picture, where they look like a credit card, a USB stick or a key fob that generates One-Time Passwords.

Figure 1: The most common forms of hardware tokens

There are two points that should be considered before choosing any of the aforementioned tokens. The former is that each token has an additional cost for running and maintenance, which in the end passes to its owner, and the latter is that the user has to carry around an additional device every time he needs to participate in the protocol.

Our proposal tries to work around these two issues and has the form of a software application that gets installed on the user's mobile phone. By this way the cost of deploying the token gets lower and, given the fact that every user has a mobile phone with him, it eliminates the need of carrying an extra device.

The main issue that comes with a token of this kind is whether the user's mobile phone is compatible with the developed application.

3. OVERVIEW OF THE PROTOCOL

The implemented protocol is based on the client-server approach, so this means that we have the respective sides that communicate through the Internet and the mobile phone networks.

The following figure presents an overview of the entities that participate in the two factor authentication protocol.

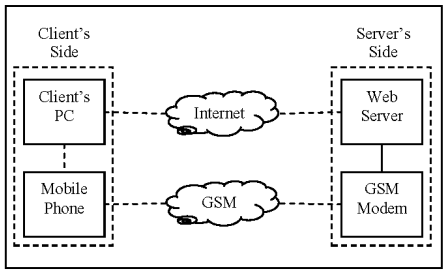


Figure 2: Overview of the participating entities into the two factor authentication protocol

The client proves that he owns the proper security token through a combination of the Challenge-Response technique and the generation of a One-Time Password.

The two sides create the One-Time Password with the assistance of the second version of the Password Based Key Derivation Function. This certain function, according to its specifications [1], needs an underlying hash algorithm in order to work properly. The algorithm we selected is SHA-256 which, as its specifications state [2], produces a 256 bits hash value which we use for the encryption of the Challenge with the AES-256 algorithm.

The generated cipher text represents the initial Response value, which in the end gets shortened, in order to be easier for the user to copy it from his mobile phone to the web browser he is using.

3.1 Server's Side

On the server's side we find the web server that hosts the web service we use in the protocol. On the same side we have a GSM Modem, which is connected to the web server, and is responsible for processing every short message from and to the GSM mobile phone network.

3.1.1 Web Server

In order to provide a demonstrative service, that integrates the two factor authentication protocol, we developed a three-tier web application according to the specifications of the Java Enterprise Edition [3].

This means that on the top two tiers we have a combination of JSP pages and Java Servlets that reside in the appropriate container, which in this case is the Apache Tomcat, while the lower level is dedicated to the database management system, with the implementation of MySQL being our choice.

3.1.2 GSM Modem

The GSM Modem may be a dedicated device or a mobile phone that is connected to the web server and gives the ability to exchange SMS messages by using AT commands, more information about this subject at [4].

Due to limited resources we have chosen the solution of a mobile phone that receives the proper instructions from the web app's side and checks if there are new incoming messages or forwards the outgoing ones.

There is a communication gap between the mobile phone/GSM Modem and the server's side, as the former device operates only with AT Commands while the latter does not. This issue is solved with the assistance of a Java module we developed and integrated into the web application.

This certain module utilizes the Java Communications Application Programming Interface and has the ability to translate the AT Commands from the GSM Modem into the proper instructions. This way the appropriate action listeners found on the web application get triggered and the protocol is set in motion.

3.2 Client's Side

The developed two factor authentication protocol utilizes two different types of communication networks, with one of them being the Internet and the other one being the GSM mobile phone network. These two networks are accessible by a personal computer and a mobile phone, so every user of the protocol must have the aforementioned devices on his side.

3.2.1 Client's Computer

The personal computer found on the client's side does not have to be built with state of the art hardware components neither it requires the installation of an extraordinary software package.

The only requirement that concerns the user's computer is the presence of an up-to-date web browser.

By this way we are more certain that the browser can cope with SSL/TLS protocol connections [5] and at the same time has what it takes to prevent a series of browser related attacks, such as Session High-Jacking and Man-In-The-Browser.

3.2.2 Client's Mobile Phone

The client's mobile phone will host the MIDlet that will transform the device to a security token.

There are certain restrictions that need to be taken under consideration when the client selects the mobile phone he is going to use in the two factor authentication protocol.

The first and most fundamental one is whether the selected mobile phone is equipped with an environment that is compatible with the mobile application we developed, which in this case is the Java Micro Edition application environment.

With this demand met we have to examine if the Java environment provides its support for the Wireless Messaging Application Programming Interface [6]. This API enables a crucial functionality feature for the protocol, which is the ability of exchanging messages with other users and applications compatible to the environment.

4. PREPARING THE GROUND

Prior to the actual users' authentication there is a series of actions that each side has to carry out and get ready to participate in the main stage of the two factor authentication protocol.

4.1 Setting up the web server

We start from the server's side, where the first thing we have to do is to install all the files that compose the web application into a

specific folder, which is located in the installation of Apache Tomcat that runs on the web server.

The next thing we have to do is to connect the GSM modem device with the web server. Once we have checked that the connection is properly established, we start the Apache Tomcat process and by this way we start web application for the very first time.

During this execution cycle a certain script is activated and completes the web app's set up procedure. The aforementioned script creates the database and generates the set of RSA keys that from now on will be used by the web application.

Up next, we have to create an up-to-date version of the assorting MIDlet that will contain the value of the newly generated RSA Public Key. For our convenience, this element is automatically exported when the set up script is executed; therefore it is already available by the time we perform the update.

The MIDlet that derives from this stage has to be available for the clients of the web application, so it is copied to a certain folder in the Apache Tomcat installation that will provide access to it through the World Wide Web.

When all of the above actions are carried out we restart the Apache Tomcat process. By this way we force the web application to start a new lifecycle and now the server's side is ready to receive requests from its clients.

4.2 Signing up to the protocol

Now we take a look at the actions that have to be carried out by a client who wishes to participate in the two factor authentication protocol.

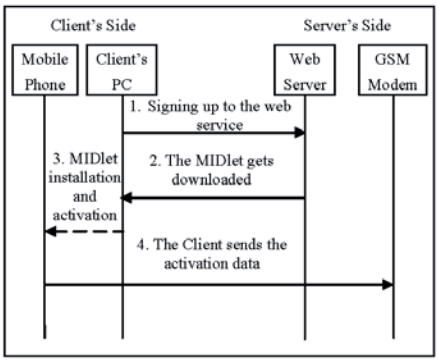


Figure 3: Actions carried out during the user's enrollment into the two factor authentication protocol

The first one of these actions is about the user's enrollment into the web application, where he uses his browser in order to visit the website and fill in the respective registration form, which requires a username, a password and a mobile phone number. From this point, the user has the ability to download the complementary MIDlet to his computer.

Figure 4: Screenshot where the user signs up to the web service

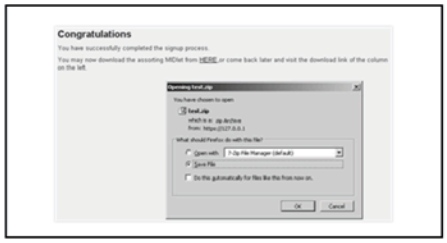


Figure 5: Screenshot where the user downloads the complementary MIDlet

The established connection is secured with the SSL/TLS protocol, thus the data submitted from the user's side are transferred over an encrypted communication channel. At the receiving end, the web server applies the SHA-256 function on the password's value and then stores the generated hash with the remaining data.

The next step is the activation of the user's security token. We assume that the user takes advantage of his mobile phone's connectivity features and manages to transfer and install the assorting MIDlet to this device.

During the mobile app's first execution the user is forwarded to a log-in screen where he is prompted to select the credentials that will allow him to use the MIDlet in the future.

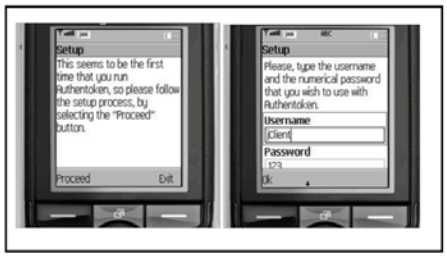


Figure 6: Screenshots from the user's mobile phone during the MIDlet's installation

These elements, a username and a numerical password, are imported into the PBKDF2 function and generate a 256 bit digest that is stored in the phone and keeps the token protected from

anyone who might steal or borrow this device. The stored hash is compared with the one created during the user's log-in and if the two values match, then we are sure that the user knows the proper credentials and he gets access to the token.

The stored hash value represents the activation data and apart from securing the user's token it has an additional role in the way the protocol works, as it participates like an initial seed when the token and the server's side calculate the response value that corresponds to a specific challenge.

The aforementioned fact makes the activation data even more important for the protocol and forces the application of an encryption algorithm at the last step, where the user sends his activation data to the server's side.

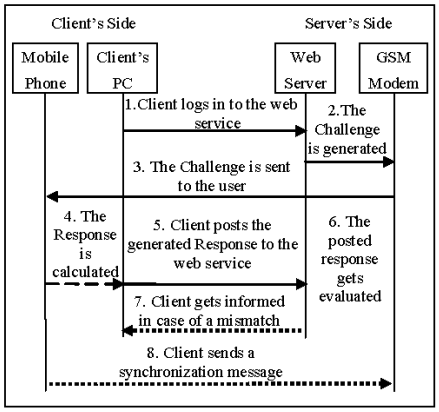
Figure 7: Screenshot from the user's mobile phone where he sends his activation data to the server's side

This message, that contains the activation data, is encrypted by using the RSA algorithm with the RSA Public Key that belongs to the server's side. By this way we are sure that only the server may decrypt the activation data because, as the algorithm's specifications state [7], the encryption is reversed only by the correlated RSA Private Key, which in this case is known only to the server.

The security token is considered active only on the condition that the message with the encrypted activation data arrives at the server's side and its content is successfully decrypted and recovered.

5. THE PROTOCOL IN ACTION

On the following picture we can concisely see the actions that occur when the two factor authentication protocol is set in motion.



Everything starts when the client accesses the web application and logs in to his account.

The client proves the first authentication factor by submitting the proper password over an Internet connection which is secured with the SSL/TLS protocol. The server evaluates the submitted password, by applying the SHA-256 function and comparing the generated hash with the value stored in its database. If the certain password turns to be the right one, then the server generates a challenge and commences the process of verifying the second authentication factor.

Figure 8: Screen sequence where the user logs in to the web service and gets forwarded to a page that requests the second authentication factor

The challenge value, which is nothing more than the number of milliseconds since the January 1st 1970, is forwarded to the GSM modem that is connected to the web server. This device is responsible for converting the challenge into an SMS message and redirecting it to the user's mobile phone.

The MIDlet on the user's device gets activated when it receives the new message and begins the procedure for calculating the response that applies for the current challenge. The response value derives from the following series of steps:

1. The counter maintained by the two sides is increased by one
2. The One-Time Password is derived from the PDBKF2 function, where in this case we use the stored hash value and the increased counter value.
3. The One-Time Password is 256 bits long and is used as the key that encrypts the challenge with the AES-256 algorithm. According to the algorithm's specifications [8], the ciphered challenge that derives from AES equals to a block of 128 bits and represents the initial response value.

In this form the generated response is not convenient for the users of the protocol, because even in a hexadecimal form the response is 32 digits long, thus we apply the following compression scheme:

1. The eight last digits of the challenge's value are isolated
2. The order of this octet gets reversed
3. We assume that each one of the eight digits corresponds to a quartet of the initial response's digits, e.g. the first digit refers to the first four digits of the initial response, and the second digit refers to the next four digits and so on. Also, for every quartet of the initial response we apply a position numbering that starts from 0, for the first digit, and reaches up to 3, for the last digit of the group.
4. We divide every digit from the second step with number 4. From the corresponding quartet we isolate the digit that is found on the position given from the remainder of the aforementioned division.
5. The last step is iterated eight times, so we isolate one digit from each one of the eight quartets of the initial response and we form the compressed response value.

In order to make the compression procedure more understandable we provide the following example, where the Challenge equals to `1269209815328` and the Initial Response is `e5e8f7b5df89515146919d802c53e758`. The Final Response is calculated according to the following table.

Quartets of Initial Response	e5e8	f7b5	df89	5151	4691	9d80	2c53	e758
Challenge's last 8 digits	0	9	8	1	5	3	2	8
Inversion of Challenge's last 8 digits	8	2	3	5	1	8	9	0
Digit mod 4	8 mod 4	2 mod 4	3 mod 4	5 mod 4	1 mod 4	8 mod 4	9 mod 4	0 mod 4
Position of isolated Character	0	2	3	1	1	0	1	0
Final Response	e	b	9	1	6	9	c	e

Table 1: An example of the Initial Response's compression procedure

The outcome of this procedure appears on the screen of the client's mobile phone and is available for submission over the Internet.

Figure 9: Screenshots from the user's mobile phone during the calculation of a Response

The response is posted over a secured Internet connection, because the user initiated an SSL/TLS session when he logged in to the web application.

At the same time, the server's side calculates the value of the expected response and waits for the client's side reaction. When the posted response arrives, the server examines whether this value matches with the one he previously calculated.

If the two values turn out to be the same, then we are sure that the client is the owner of the proper security token, thus he is allowed to access the web pages with protected content.

In any other case, the web app presents the appropriate message which informs the client about the mismatch between the posted and the expected response that leads to his failure to verify the second authentication factor.

This mismatch might be caused by a difference in the value of the counter maintained between the two sides or by a mistyping when the user submitted his response.

The client has a couple of options that will help him get over a situation where the two sides provide a different response. The former one is to resubmit the response generated by his token, by this way the client eliminates any potential mistyping. The latter option is to send a synchronization message to the server that will make sure that the two sides maintain a common counter.

The counter value sent from the client's mobile phone is encrypted with the RSA algorithm using the server's side RSA Public Key. By this way we prevent a third party from intercepting the contents of this message and keeping track of one of the elements that produce the One-Time Password.

6. CONCLUSIONS

The two factor authentication protocol presented in this paper has proven that an alternative solution, where a client's mobile phone is reused as a security token, is feasible and totally realistic.

Our solution manages to offer the best potential security for its users, with the utilization of strong encryption algorithms where it is needed, while at the same time remains easy for them to use it and hides its overall complexity.

The implemented protocol costs less than the other types of security token and it is really easy to deploy, unless the users' mobile phones are not compatible with the assorting MIDlet. Also, the fact that the client's side turns its own mobile phone into a security token eliminates the need of carrying an additional device which could be easily lost or forgotten somewhere.

7. ADDITIONAL RESEARCH

Further research on the subject would include porting of the security token MIDlet to other mobile phone application platforms and extended testing that will help us estimate whether the protocol works in a proper manner.

Also, it would be interesting to implement an alternative scenario by applying a series of modifications on some of the protocol's modules, like the client's security token and the service provided by the server's side.

By this way we will examine whether the solution presented in this paper may function as a protocol for the out-of-band verification of actions carried out online by the client's side.

The modifications on the web service would allow the user to access his bank account and his unpaid bills. From this point he may initiate the online payment of a public utilities bill or transfer a sum of money to another bank account.

Based on the action selected by the user, the web service would create the proper short message and with the assistance of the GSM Modem it would forward it to the user's mobile phone.

On the other end, the user's mobile phone will be equipped with a properly modified MIDlet that will verify the integrity of the messages sent from the server's side and present vital details about the selected online action.

At this stage the user is in total control of the situation, as he may press a button on his phone and send the proper short message that will inform the server about the verification or the repudiation of the selected action.

8. ACKNOWLEDGMENTS

We would like to thank every one of the researchers that we refer to in the next section. Without their pioneering ideas and the breakthroughs they achieved we would not be able to develop the concept that led to the secure messaging application.

9. REFERENCES

[1] RSA Laboratories, "PKCS #5: Password-Based Cryptography Standard," 25 March 1999. [Online]. Available: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf.

[2] National Institute of Standards and Technology, "FIPS 180-2 - Secure Hash Standard," 1 August 2002. [Online]. Available: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf.

[3] "Java Community Process - JSR-000316 Java Platform, Enterprise Edition 6," [Online]. Available: http://jcp.org/aboutJava/communityprocess/final/jsr316/index.html.

[4] "How to Use Microsoft HyperTerminal to Send AT Commands to a Mobile Phone or a GSM/GPRS Modem?," [Online]. Available: http://www.developershome.com/sms/howToUseHyperTerminal.asp.

[5] T. Dierks and C. Allen, "RFC 2246 - The TLS Protocol Version 1.0," January 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2246.txt.

[6] E. C. Ortiz, "The Wireless Messaging API," December 2002. [Online]. Available: http://developers.sun.com/mobility/midp/articles/wma/.

[7] R. L. Rivest, A. Shamir and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 1, pp. 120-126, 1978.

[8] National Institute of Standards and Technology, "FIPS 197 - Advanced Encryption Standard," 26 November 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.