Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Τίτλος: Μελέτη των απειλών και υλοποίηση μηχανισμών για την ασφάλεια των υπηρεσιών SMS & MMS

Φαϊτάκη Κωνσταντίνα (ΑΜ: 1237)
Φεσάκης Ηλίας (ΑΜ: 1342)

Ηράκλειο – Ημερομηνία

Επόπτης Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 – ΔΙΚΤΥΟ GSM

ΚΕΦΑΛΑΙΟ 2 – ΑΣΦΑΛΕΙΑ GSM

ΚΕΦΑΛΑΙΟ 3 – ΕΠΙΘΕΣΕΙΣ GSM

ΚΕΦΑΛΑΙΟ 4 – ΕΠΙΘΕΣΕΙΣ SMS

ΚΕΦΑΛΑΙΟ 5 – MIDP ΚΑΙ CLDC

ΚΕΦΑΛΑΙΟ 6 – ΚΩΔΙΚΑΣ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΛΗΨΗΣ ΜΗΝΥΜΑΤΩΝ

ΚΕΦΑΛΑΙΟ 7 – ΥΛΟΠΟΙΗΣΗ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ SMS

ΕΙΣΑΓΩΓΗ

Σ’ αυτήν την πτυχιακή περιγράφεται η λειτουργία του δικτύου GSM, που χρησιμοποιείται στην κινητή τηλεφωνία, τα διάφορα στοιχεία που το αποτελούν, οι μηχανισμοί ασφαλείας αυτού του δικτύου καθώς και οι πιθανές επιθέσεις που μπορούν να γίνουν από κάποιον επιτιθέμενο ώστε να παρακάμψει αυτούς τους μηχανισμούς και να έχει πρόσβαση στα δεδομένα που μεταδίδονται. Επίσης περιγράφεται η λειτουργία αποστολής και λήψης σύντομων γραπτών μηνυμάτων (sms) και μηνυμάτων πολυμέσων (mms), καθώς και επιθέσεις που μπορούν να γίνουν με την χρήση των sms. Επιπλέον αναφέρεται πως βελτιώνονται τα πράγματα στης τρίτης γενιάς κινητής τηλεφωνίας (3G). Τέλος γίνεται αναφορά στο MIDP και στο CLDC τα οποία σχετίζονται με τον κώδικα. Αυτά όσο αναφορά το θεωρητικό κομμάτι. Στο πρακτικό υπάρχει υλοποίηση προγράμματος σε κώδικα Java, το οποίο τρέχει σε κινητά και αυτό που κάνει είναι να κρυπτογραφεί σε επίπεδο εφαρμογής τα μηνύματα SMS και MMS ώστε να μην μπορούν να διαβαστούν από κάποιον τρίτο ο οποίος επιτίθεται στο δίκτυο με σκοπό την ανάκτηση των δεδομένων.

Σκοπός

Σκοπός της πτυχιακής αυτής είναι να υλοποιηθεί πρόγραμμα το οποίο να τρέχει σε κινητά και να προσφέρει υπηρεσίες κρυπτογράφησης στα μηνύματα SMS και MMS ώστε να παραμένουν ασφαλή κατά την μετάδοσή τους.

Περιγραφή κεφαλαίων

Στο πρώτο κεφάλαιο περιγράφεται το δίκτυο GSM με όλα τα στοιχεία που το αποτελούν (κινητό, σταθμός βάσης, μητρώο, κέντρο αυθεντικοποίησης κ.τ.λ) και την λειτουργία που συντελεί το καθένα. Στην συνέχεια του κεφαλαίου αναφέρεται ο τρόπος που αποστέλλονται και λαμβάνονται τα μηνύματα SMS και έπειτα γίνεται αναφορά σε άλλους δύο τύπους μηνυμάτων, τα EMS και τα MMS. Τα MMS μάλιστα περιγράφονται πιο αναλυτικά μαζί με τα στοιχεία του δικτύου που είναι απαραίτητα στο GSM ώστε να πραγματοποιείται η μετάδοσή τους. Στο τέλος του κεφαλαίου αναφέρεται η τεχνολογία της τρίτης γενιάς κινητής τηλεφωνίας καθώς και τα πλεονεκτήματα που προσφέρει απέναντι στο GSM.

Στο δεύτερο κεφάλαιο περιγράφονται οι μηχανισμοί ασφαλείας του δικτύου GSM όπως για παράδειγμα αυθεντικοποίηση του κινητού στον σταθμό βάσης, κρυπτογράφηση των δεδομένων σε επίπεδο δικτύου κ.α, καθώς και οι αδυναμίες που παρουσιάζουν αυτοί οι μηχανισμοί οι οποίες επιτρέπουν σε κάποιον να προσβάλει το δίκτυο και να ανακτήσει δεδομένα που κανονικά δεν θα έπρεπε.

Στο τρίτο κεφάλαιο αναφέρονται οι επιθέσεις που μπορούν να γίνουν στο δίκτυο GMS με κύριο σκοπό την ανάκτηση των δεδομένων που μεταδίδονται στον αέρα. Μερικές από τις επιθέσεις αυτές το πετυχαίνουν ανακτώντας πρώτα το κλειδί κρυπτογράφησης, είτε από το ασύρματο κανάλι, είτε από την κάρτα SIM, είτε από το κέντρο αυθεντικοποίησης. Άλλες πάλι στοχεύουν στο σπάσιμο των αλγορίθμων που

χρησιμοποιούνται για την κρυπτογράφηση και την παραγωγή των κλειδιών. Δύο από αυτές έχουν σαν στόχο να μην έχει ο νόμιμος χρήστης πρόσβαση στις υπηρεσίες του δικτύου.

Στο τέταρτο κεφάλαιο περιγράφονται διαφόρων ειδών επιθέσεις που γίνονται στην υπηρεσία των μηνυμάτων SMS. Στις επιθέσεις αυτές ο στόχος και το τελικό αποτέλεσμα δεν είναι πάντα το ίδιο. Για παράδειγμα κάποιες επιθέσεις στοχεύουν στην διακοπή κάποιας υπηρεσίας για κάποιο χρονικό διάστημα ή στην απόκτηση του ελέγχου μίας μηχανής του δικτύου, ή την ανάκτηση ενός κωδικού πρόσβασης που μεταδίδεται μέσω μηνυμάτων, ή ακόμα και την υπερχρέωση ενός χρήστη κινητού τηλεφώνου.

Στο πέμπτο κεφάλαιο παρουσιάζεται το MIDP και το CLDC, τα οποία είναι πρότυπα στοιχεία του περιβάλλοντος ανάπτυξης, διασύνδεσης και εκτέλεσης προγραμμάτων της γλώσσας Java σε μικροσυσκευές που έχουν περιορισμένους πόρους όπως για παράδειγμα τα κινητά τηλέφωνα. Το MIDP παρέχει στις μικροσυσκευές δυνατότητες όπως να αναπαράγουν ήχους, να εκτελούν εφαρμογές παιχνιδιών, να υποστηρίζουν διάφορα πρότυπα συνδεσιμότητας και να παρέχουν πρότυπα ασφαλείας απ' άκρον εις άκρον. Στην ουσία καθορίζει τις δυνατότητες που θα παρέχουν οι εφαρμογές στις συσκευές αυτές. Το CLDC καθορίζει το σύνολο των βιβλιοθηκών που θα έχει η εικονική μηχανή της Java ώστε να δίνει την δυνατότητα σε μία συσκευή περιορισμένης επεξεργαστικής ισχύς και μνήμης να εκτελεί εφαρμογές Java που προορίζονται γι' αυτές.

Στο έκτο κεφάλαιο αρχίζει το πρακτικό μέρος στο οποίο περιγράφεται ο αρχικός κώδικας αποστολής και λήψης μηνυμάτων που υπάρχει μέσα στην εφαρμογή επεξεργασίας του κώδικα. Επίσης αναφέρονται όλα τα απαραίτητα προγράμματα που χρειάζονται για την επεξεργασία και εκτέλεση του κώδικα. Χρησιμοποιείται ένας προσομοιωτής κινητού τηλεφώνου ώστε ο κώδικας να εκτελείται σε περιβάλλον μικροσυσκευής και όχι υπολογιστή. Περιγράφονται αναλυτικά οι κλάσεις που αποτελείται η εφαρμογή καθώς και στιγμιότυπα του προσομοιωτή κατά την εκτέλεση του κώδικα. Στο τέλος παρέχονται κάποια συμπεράσματα.

Στο έβδομο και τελευταίο κεφάλαιο παρουσιάζεται ο προηγούμενος κώδικας τροποποιημένος κατά τέτοιο τρόπο ώστε να καλύπτει τις απαιτούμενες ανάγκες όπως για παράδειγμα κρυπτογράφηση των μηνυμάτων. Περιγράφονται οι αλλαγές που έγιναν στον κώδικα καθώς και οι λόγοι που οδήγησαν σε αυτές τις αλλαγές. Παρουσιάζονται επίσης στιγμιότυπα του προσομοιωτή καθώς εκτελείται ο τροποποιημένος κώδικας και τέλος συμπεράσματα για το κατά πόσο η υλοποίηση πέτυχε τον στόχο της.

Τι είναι τα SMS;

Είναι η τεχνολογία που επιτρέπει την αποστολή και την λήψη σύντομων γραπτών μηνυμάτων μεταξύ χρηστών κινητών τηλεφώνων. Τα αρχικά SMS σημαίνουν Short Message Service, δηλαδή Υπηρεσία Σύντομων Μυνημάτων. Στην Ευρώπη, τα SMS εμφανίστηκαν για πρώτη φορά το 1992. Περιλαμβάνονταν από την αρχή μέσα στις προδιαγραφές του δικτύου GSM και αργότερα προστέθηκε και στην ασύρματη τεχνολογία CDMA και TDMA. Τα πρότυπα GSM και SMS αναπτύχθηκαν αρχικά

από τον οργανισμό ETSI (European Telecommunications Standards Institute). Τώρα ο 3GPP(Third Generation Partnership Project) είναι υπεύθυνος για την εξέλιξη και την συντήρηση των προτύπων GSM και SMS.

Τεχνικά χαρακτηριστικά

Τα δεδομένα που μπορούν να μεταφερθούν μέσω των SMS είναι πολύ περιορισμένα. Ένα μήνυμα μπορεί να περιέχει το πολύ μέχρι 140 Bytes (1120 bits) δεδομένων. Έτσι ένα μήνυμα χαρακτήρων μπορεί να περιέχει μέχρι:

- 160 χαρακτήρες, εάν η κωδικοποίηση των χαρακτήρων είναι 7-bit, η οποία είναι κατάλληλη για αλφάβητα που περιέχουν λατινικούς χαρακτήρες όπως το αγγλικό.
- 70 χαρακτήρες, εάν χρησιμοποιείται η κωδικοποίηση 16-bit Unicode UCS2, η οποία χρησιμεύει όταν το μήνυμα περιέχει μη λατινικούς χαρακτήρες.

Τα μηνύματα SMS μπορούν να περιέχουν χαρακτήρες από όλες τις γλώσσες που υποστηρίζονται από την κωδικοποίηση Unicode, όπως για παράδειγμα τα ελληνικά, τα αραβικά, τα κινέζικα και διάφορες άλλες γλώσσες.

Ένα σημαντικό πλεονέκτημα των SMS είναι ότι υποστηρίζεται απ' όλα τα κινητά τηλέφωνα GSM. Σχεδόν όλα τα συνδρομητικά προγράμματα που προσφέρουν οι πάροχοι περιλαμβάνουν την ανέξοδη υπηρεσία των μηνυμάτων SMS. Αντίθετα με τα SMS, άλλες τεχνολογίες της κινητής τηλεφωνίας, όπως το WAP και η Java, δεν υποστηρίζονται από πολλές παλιές κινητές τηλεφωνικές συσκευές.

Ένα μειονέκτημα της τεχνολογίας SMS είναι το γεγονός ότι ένα μήνυμα μεταφέρει μόνο περιορισμένη ποσότητα δεδομένων. Για να ξεπεραστεί αυτό το μειονέκτημα αναπτύχθηκε το λεγόμενο συνδεδεμένο SMS ή μεγάλο SMS. Ένα τέτοιο μήνυμα μπορεί να περιέχει πάνω από 160 λατινικούς χαρακτήρες. Η λειτουργία του έχει ως εξής: Το τηλέφωνο του αποστολέα χωρίζει το μεγάλο μήνυμα σε μικρότερα κομμάτια και τα στέλνει ως ενιαία μεμονωμένα μηνύματα. Ο παραλήπτης όταν λάβει αυτά τα μηνύματα, τα ενώνει ξανά σε ένα μεγάλο ενιαίο μήνυμα SMS. Το μειονέκτημα αυτού του μηνύματος είναι ότι δεν υποστηρίζεται απ' όλες τις συσκευές GSM.

Ιστορική Αναδρομή

Η υπηρεσία SMS θεωρείται μία απρόβλεπτη επιτυχία στην κυψελοειδή βιομηχανία. Στις 3 Δεκεμβρίου του 1992, ένας μηχανικός με το όνομα Neil Papworth έστειλε το πρώτο SMS με το μήνυμα "ΚΑΛΑ ΧΡΙΣΤΟΎΓΕΝΝΑ" στους συναδέλφους του στη Vodafone στη Μεγάλη Βρετανία. Στην αρχή η υπηρεσία είχε μικρή προώθηση από τις εταιρείες κινητής, αλλά η γρήγορη αποδοχή της ξεκίνησε νωρίς στις ευρωπαϊκές αγορές, όπου η τάση αυτή υιοθετήθηκε κυρίως από την νεολαία.

Κάθε γενιά υιοθέτει μία τεχνολογία επικοινωνίας ως δική της. Η σημερινή νεολαία υιοθέτησε την υπηρεσία SMS. Αν και έχει μία δυσκολία στην χρήση λόγω της διεπαφής με τις συσκευές, οι νέοι χρήστες αποφάσισαν να ξεπεράσουν αυτό το

εμπόδιο και να χρησιμοποιήσουν την υπηρεσία. Η δυσκολία αυτή έκανε την μεγαλύτερη γενιά απρόθυμη και ανίκανη να χρησιμοποιήσει τα SMS.

Η υπηρεσία SMS είναι μία από τις λίγες στην ιστορία του εμπορίου όπου ανέβηκε τόσο γρήγορα χωρίς να υπάρχει αντίστοιχη μείωση στην τιμολόγησή της. Συνήθως, όπως και στην περίπτωση των κινητών τηλεφώνων, η μείωση των τιμών στις συσκευές και στις υπηρεσίες οδήγησαν σε αύξηση της χρήσης. Αυτό έφερνε τους νέους πιο κοντά στην αγορά της κινητής τηλεφωνίας, αλλά η τιμή του SMS έμενε σταθερή επειδή τα δίκτυα είχαν πρόβλημα χειρισμού του ολοένα και αυξανόμενου όγκου των μηνυμάτων.

Εξαιτίας της εκτεταμένης χρήσης των SMS από του νέους, πρόεκυψε ένα νέο αλφάβητο με διάφορες συντμήσεις όπως για παράδειγμα "C U L8er" αντί για το "See you later". Αυτό γινόταν για εξοικονόμηση χρόνου αλλά και χώρου στο μήνυμα γιατί οι χρήστες προσπαθούσαν να πουν όσο το δυνατόν περισσότερα με όσο το δυνατόν λιγότερα γράμματα. Επίσης υπήρχαν τα Smiles, μικρά προσωπάκια που δείχνανε την διάθεση αυτού που τα έγραφε. Αυτό βοήθησε να μειωθεί η ψυχρότητα του μέσου και έκανε την υπηρεσία ακόμα πιο δημοφιλή.

Στην αρχή οι χρήστες προπλήρωναν για τις τηλεφωνικές τους συνδιαλέξεις. Με αυτόν τον τρόπο μπορούσαν να ελέγχουν τις τηλεφωνικές τους δαπάνες. Αυτό έπαιξε καταλυτικό ρόλο στην επιτάχυνση της χρήσης των SMS διότι οι φορείς αδυνατούσαν τεχνικά εκείνη την εποχή να χρεώσουν τα μηνύματα. Αυτό τους έκανε να αποσιωπήσουν την διάδοση της υπηρεσίας γιατί δεν μπορούσαν να την τιμολογήσουν και οι χρήστες θα μπορούσαν να επικοινωνούν χωρίς καμία χρέωση. SMS μπορούσαν να στείλουν μόνο όσοι χρήστες είχαν προπληρωμένο λογαριασμό για τηλεφωνικές συνδιαλέξεις. Με την διάδοση όμως του Διαδικτύου και την ελεύθερη ανταλλαγή πληροφοριών, η νεαροί κυρίως χρήστες έμαθαν γι' αυτό και το χρησιμοποιούσαν υπέρ τους στέλνοντας δωρεάν εκατομμύρια μηνύματα από τηλέφωνα τα οποία οι χρήστες τους προπλήρωναν για τις κλήσεις φωνής. Τελικά μετά από μερικούς μήνες, οι φορείς συναντήθηκαν μεταξύ τους και αποφάσισαν να εφαρμόσουν χρέωση και στα SMS σε χρήστες με προπληρωμένο λογαριασμό, αφαιρώντας κάθε φορά το κόστος του μηνύματος απ' αυτόν.

Μετά απ' αυτό ξεκίνησε μία εκστρατεία διανομής μαζικών SMS τα οποία εστάλλησαν στους χρήστες που είχαν χρησιμοποιήσει την υπηρεσία, ότι από μία ορισμένη ημερομηνία και μετά τα μηνύματα θα χρεώνονταν. Αυτό οδήγησε σε μία άμεση και παρατεταμένη πτώση στην χρήση των SMS μεταξύ 25-40%, αφού έκανε τους χρήστες να μειώσουν ή και να σταματήσουν τη χρήση της υπηρεσίας. Έπειτα συνέβη κάτι ενδιαφέρον, η χρήση άρχισε και πάλι να ανεβαίνει και σύντομα έφτασε στα επίπεδα που βρισκόταν πριν να αρχίσουν οι φορείς να χρεώνουν την υπηρεσία. Ο όγκος των μηνυμάτων συνέχισε την ανοδική του πορεία τροφοδοτούμενος από την διαπροσωπική επικοινωνία μεταξύ μεμονωμένων ανθρώπων που αντάλλαζαν καθημερινά μηνύματα λέγοντας πως νιώθουν και τι κάνουν. Τα SMS είχαν γίνει ένα σημαντικό μέρος του τρόπου που επικοινωνούσαν οι νέοι μεταξύ τους στην καθημερινή τους ζωή.

Η ανοδική πορεία της υπηρεσίας συνεχίστηκε με γοργούς ρυθμούς κατά την διάρκεια του έτους 2000 στην Ευρώπη. Εκείνη την περίοδο η βιομηχανία κινητής τηλεφωνίας προωθούσε την υπηρεσία WAP. Το WAP δεν υιοθετήθηκε ιδιαίτερα από

το ευρύ κοινό. Το μόνο που κατάφερε ήταν να κινήσει το ενδιαφέρον εκείνων των χρηστών που ήθελαν να έχουν πρόσβαση στο Διαδίκτυο μέσω του κινητού τους τηλεφώνου. Τελικά αποτέλεσε καταστροφή για τις επιχειρήσεις που ασχολήθηκαν μ' αυτό. Αργότερα αυτές οι επιχείρησης συνειδητοποίησαν ότι τα SMS και όχι το WAP είχε την μεγαλύτερη απήχηση στο κοινό.

Αρχικά, υπήρχαν δυσκολίες που δεν επέτρεπαν σε συνδρομητές που ανήκαν σε ένα δίκτυο να στείλουν μηνύματα σε συνδρομητές άλλων δικτύων. Αυτός ήταν ένας αποτρεπτικός παράγοντας στην προώθηση της χρήσης SMS.
Οι δυσκολίες αυτές είναι οι εξής:

1) Διαφορετικά επιτρεπόμενα μεγέθη μηνυμάτων
2) Διαφορετικές τεχνολογίες αποστολής και λήψης μηνυμάτων (κάποια δίκτυα κάνουν χρήση του πρωτοκόλλου ασύρματων εφαρμογών (WAP) και όχι SMS.
3) Πολλαπλοί κανόνες μετατροπής χαρακτήρων.
4) Ασυνέπειες στους κανόνες προτεραιότητας των μηνυμάτων.

Τον Απρίλιο του 2002, κύριοι φορείς της κυψελοειδής τηλεφωνίας υπογράφουν διάφορες συμφωνίες λειτουργικότητας ώστε να παρέχουν υπηρεσίες αποστολής και λήψης σύντομων γραπτών μηνυμάτων μεταξύ των δικτύων τους. Φορείς όπως η Inphomatch, η Telecommunications Systems (TCS) και η Mobilespring, παρέχουν αυτήν την λειτουργία. Τώρα ουσιαστικά όλοι οι συνδρομητές των δικτύων αυτών είναι σε θέση να στέλνουν γραπτά μηνύματα ο ένας στον άλλο.

Εμπορικές εφαρμογές

Τα SMS χρησιμοποιούνται κυρίως για διαπροσωπική επικοινωνία, δηλαδή επικοινωνία μεταξύ δυο ατόμων. Τα μηνύματα αυτά όμως μπορούν να χρησιμοποιηθούν και για εμπορικούς σκοπούς. Τέτοιες εμπορικές εφαρμογές υπάρχουν πολλές. Μία απ' αυτές είναι η τηλεψηφοφορία όπως γίνεται για παράδειγμα στον διαγωνισμό της Eurovision, όπου οι τηλεθεατές βλέποντας τους διαγωνιζόμενους στην συνέχεια καλούνται να ψηφίσουν την προτίμηση τους στέλνοντας SMS από το κινητό τους. Τα μηνύματα αυτά αποτελούνται συνήθως από λίγα συγκεκριμένα γράμματα συνοδευόμενα από κάποιον αριθμό ο οποίος δείχνει την προτίμηση του τηλεθεατή.

Άλλη μια τέτοια υπηρεσία είναι η παροχή κυκλοφοριακών δεδομένων σε οδηγούς. Με την υπηρεσία αυτή, ο χρήστης έχει τη δυνατότητα να λαμβάνει κυκλοφοριακή ενημέρωση (κυκλοφοριακά συμβάντα, κατάσταση κυκλοφορίας) και πληροφορίες για σημεία ενδιαφέροντος (πρατήρια βενζίνης, ξενοδοχεία, εστιατόρια, φαρμακεία, κλπ) με SMS, αποστέλλοντας σχετικό μήνυμα σε προκαθορισμένο αριθμό για μεμονωμένη χρήση της υπηρεσίας. Η παροχή της υπηρεσίας γίνεται είτε με αναζήτηση της πληροφορίας από το χρήστη με αποστολή SMS (Υπηρεσίες τύπου SMS PULL), είτε σε συνδρομητική βάση (Υπηρεσίες τύπου SMS PUSH).

Άλλες εμπορικές χρήσεις είναι η συμμετοχή των χρηστών σε παιχνίδια, διαγωνισμούς και κληρώσεις ή ακόμα και για διαφημιστικούς σκοπους. Επίσης υπάρχουν και κάποιες υπηρεσίες πληροφόρησης οι οποίες είναι συνδρομητικές. Ο χρήστης στέλνει

ένα μήνυμα όπου ζητάει να εγγραφεί στην υπηρεσία και από ‘κει και πέρα η υπηρεσία στέλνει στον χρήστη πληροφορίες μέσω SMS και μέχρι έναν μέγιστο αριθμό μηνυμάτων την εβδομάδα (SMS Push).

Όλες αυτές οι υπηρεσίες επιβαρύνουν οικονομικά τον χρήστη ανεξάρτητα αν στέλνει ή δέχεται μήνυμα από την υπηρεσία. Οι χρεώσεις είναι συνήθως υπερπολλαπλάσιες σε σχέση με ένα απλό μήνυμα προς έναν άλλον χρήστη.

Στατιστικά

Η χρήση των SMS ολοένα και αυξάνεται. Έχει υπολογιστεί πως το 74% των χρηστών κινητού τηλεφώνου παγκοσμίως χρησιμοποιούν αυτή τη διαδικασία. Στις περισσότερες περιπτώσεις χρησιμοποιείται αυτή η μέθοδος επικοινωνίας διότι το κόστος αποστολής ενός γραπτού μηνύματος είναι κατά πολύ μικρότερο από την απ' ευθείας συνομιλία. Το 50% όσων στέλνουν μηνύματα είναι πάνω από 35 ετών ενώ το 75% είναι πάνω από 25. Έχει καταγραφεί επίσης ότι το 6% των μηνυμάτων δεν διαβάζονται από τους παραλήπτες, πιθανόν λόγω του ότι δεν γνωρίζουν πώς να το κάνουν.

Ο αριθμός των SMS που αποστέλλονται σε όλον τον κόσμο είναι τεράστιος. Η AT&T κατέγραψε όλο το 2007, 78 εκατομμύρια μηνύματα σπάζοντας το ρεκόρ του προηγούμενου χρόνου που ήταν 64,5 εκατομμύρια. Επίσης υπάρχουν κάποιες ειδικές μέρες του χρόνου όπου ο αριθμός των μηνυμάτων εκτινάσσονται στα ύψη. Τέτοιες μέρες είναι κυρίως η πρωτοχρονιά και τα Χριστούγεννα. Την πρωτοχρονιά του 2008 προς 2009 καταγράφηκαν 43 δις SMS σε παγκόσμια κλίμακα, καταγράφοντας άνοδο 30% σε σχέση με την προηγούμενη χρονιά. Η αύξηση αυτή δεν αντιπροσωπεύει ξεχωριστά κάθε χώρα καθώς υπάρχουν χώρες όπου καταγράφεται διπλάσιος ή και τριπλάσιος αριθμός μηνυμάτων σε σχέση με την προηγούμενη πρωτοχρονιά. Ένα παράδειγμα είναι η Πορτογαλία και η Ολλανδία όπου τα SMS τριπλασιάστηκαν και διπλασιάστηκαν αντίστοιχα. Στην Ινδία, 220 εκατομμύρια χρήστες έστειλαν πάνω από 1 δις μηνύματα εκείνη την μέρα, ενώ στις Φιλιππίνες στάλθηκαν 1,4 δις SMS από μόλις 50 εκατομμύρια χρήστες.

Η διείσδυση της κινητής τηλεφωνίας στη χώρα μας έχει φτάσει σε επίπεδα άνω του 75%. Αυτό πρακτικά σημαίνει ότι 3 στους 4 κατοίκους της χώρας έχει στα χέρια τους μία προσωπική επικοινωνιακή συσκευή, ικανή να χρησιμοποιηθεί για επικοινωνία με ήχο, κείμενο αλλά και εικόνα.

ΚΕΦΑΛΑΙΟ 1 – ΔΙΚΤΥΟ GSM

1.1 1η γενιά - Ιστορική αναδρομή

Τα κινητά τηλέφωνα έχουν περάσει από τρεις διακριτές γενιές με διαφορετικές τεχνολογίες:

1. Αναλογική φωνή
2. Ψηφιακή φωνή
3. Ψηφιακή φωνή και δεδομένα (Internet, ηλεκτρονικό ταχυδρομείο, κτλ.)

Το πρώτο σύστημα κινητής τηλεφωνίας επινοήθηκε στις Η.Π.Α. από την AT&T και επιβλήθηκε σε όλη τη χώρα από την FCC. Έτσι ολόκληρες οι Η.Π.Α. είχαν ένα μοναδικό (αναλογικό) σύστημα, οπότε ένα κινητό τηλέφωνο που είχε αγοραστεί στην Καλιφόρνια δούλευε και στην Νέα Υόρκη. Αντίθετα, όταν η κινητή τηλεφωνία έφτασε στην Ευρώπη, κάθε χώρα επινόησε το δικό της σύστημα, γεγονός που οδήγησε σε αποτυχία.

Τα κινητά ραδιοτηλέφωνα ήταν σε σποραδική χρήση στις θαλάσσιες και στρατιωτικές επικοινωνίες κατά τις πρώτες δεκαετίες του 20ου αιώνα. Το 1946 έγινε η εγκατάσταση του πρώτου συστήματος τηλεφωνίας στο Σεντ Λιούις. Το σύστημα αυτό χρησιμοποιούσε ένα μόνο μεγάλο πομπό στην κορυφή ενός ψηλού κτιρίου και είχε ένα μόνο κανάλι, το οποίο χρησιμοποιείτο τόσο για αποστολή όσο και για λήψη. Οι συσκευές είχαν το μέγεθος βαλίτσας και έμπαιναν στον αποθηκευτικό χώρο των αυτοκινήτων και συνοδεύονταν από ένα ακουστικό που ήταν στο χώρο των επιβατών.

Για να τηλεφωνήσει κάποιος θα έπρεπε να καλέσει μέσω τηλεφωνικού κέντρου και με την προϋπόθεση ότι τη στιγμή που καλούσε υπήρχε ελεύθερη γραμμή. Αν όλες οι γραμμές ήταν κατειλημμένες, τότε δε μπορούσε να πραγματοποιηθεί κλήση. Επιπλέον, ο καλών θα έπρεπε να γνωρίζει σε ποια περιοχή κινούνταν το άτομο το οποίο ήθελε να καλέσει στο τηλέφωνο. Για να μιλήσει ο χρήστης, έπρεπε να πατήσει ένα πλήκτρο το οποίο ενεργοποιούσε τον πομπό και απενεργοποιούσε τον δέκτη. Τέτοια συστήματα, που είναι γνωστά ως **συστήματα πίεσε για να μιλήσεις** (push-to-talk systems), εγκαταστάθηκαν σε πολλές πόλεις από τα τέλη της δεκαετίας του 1950. Η τεχνολογία αυτή χρησιμοποιείται συχνά στους ασύρματους CB, στα ραδιοταξί, καθώς και στα περιπολικά της αστυνομίας.

Πάνω απ' όλα οι πρώτοι χρήστες κινητών τηλεφώνων ήταν οπλισμένοι με πολλή υπομονή. Καθώς τα δίκτυα ήταν κακοφτιαγμένα και περιορισμένα, οι νέοι πελάτες έπρεπε να περιμένουν αρκετούς μήνες, ακόμη και έναν ολόκληρο χρόνο, μέχρι να εγκριθεί η σύνδεσή τους. Εν τω μεταξύ μετά την έλευση του τρανζίστορ, οι συσκευές περιορίστηκαν στο μέγεθος, φτάνοντας το μέγεθος ενός κουτιού.

Την δεκαετία του 1960 εγκαταστάθηκε το **Βελτιωμένο Σύστημα Κινητής Τηλεφωνίας** ή **IMTS** (Improved Mobile Telephone System). Και αυτό χρησιμοποιούσε έναν πομπό υψηλής ισχύος (200 Watt) στην κορυφή ενός λόφου, είχε όμως δύο συχνότητες, μία για αποστολή και μία για λήψη. Έτσι δεν χρειαζόταν

πια το πλήκτρο “πίεσε για να μιλήσεις”. Επειδή όλες οι εισερχόμενες επικοινωνίες από τα κινητά χρησιμοποιούσαν διαφορετικό κανάλι από τα εξερχόμενα σήματα, οι κινητοί χρήστες δεν μπορούσαν να ακούν ο ένας τον άλλον σε αντίθεση με τα συστήματα “πίεσε για να μιλήσεις” που χρησιμοποιούνται στα ραδιοταξί.

Το IMTS υποστήριζε 23 κανάλια, τα οποία εκτείνονταν από τα 150 MHz έως τα 450 MHz. Λόγω του μικρού πλήθους καναλιών, οι χρήστες έπρεπε συχνά να περιμένουν πολλή ώρα πριν ακούσουν τον τόνο επιλογής. Επιπλέον, λόγω της υψηλής ισχύος του πομπού στην κορυφή του λόφου, τυχόν γειτονικά συστήματα θα έπρεπε να βρίσκονται αρκετές εκατοντάδες χιλιόμετρα μακριά το ένα από το άλλο, ώστε να αποφεύγονται οι παρεμβολές. Γενικά, η περιορισμένη χωρητικότητα έκανε το σύστημα μη πρακτικό.

Όλα αυτά άλλαξαν με το **Προηγμένο Σύστημα Κινητής Τηλεφωνίας** ή **AMPS** (Advance Mobile Phone System), το οποίο επινοήθηκε στα Bell Labs και εγκαταστάθηκε για πρώτη φορά το 1982 στις Η.Π.Α. Χρησιμοποιήθηκε επίσης και στην Αγγλία όπου ονομαζόταν TACS, καθώς και στην Ιαπωνία όπου ονομαζόταν MCS-L1.

Το σύστημα αυτό, αν και αναλογικό ήταν πιο εκτεταμένο, ενώ προσέφερε περισσότερες δυνατότητες, κυρίως ότι πλέον ήταν εφικτό να καλέσει κάποιος άμεσα το άτομο που επιθυμούσε χωρίς να παρεμβαίνει κάποιο τηλεφωνικό κέντρο. Ωστόσο, και πάλι οι συσκευές αν και δεν είχαν το μέγεθος που είχαν οι συσκευές τη δεκαετία του '50, το βάρος τους έφτανε τα 2,5 κιλά και ο κόσμος που τα χρησιμοποιούσε κυκλοφορούσε με μία μικρή συσκευή.

Παράλληλα, στις ΗΠΑ ξεκίνησαν προσπάθειες από τις αρχές της δεκαετίας του '70 με πρωτοπόρο τη Motorola και την Bell, να στήσουν τα πρώτα πραγματικά δίκτυα κινητής τηλεφωνίας και να κατασκευάσουν τα πρώτα κινητά τηλέφωνα. Η Motorola παρουσίασε το πρώτο κανονικό και λειτουργικό κινητό τηλέφωνο το 1973, ενώ το πρώτο δίκτυο στήθηκε το 1978. Ωστόσο, οι ΗΠΑ έμειναν τεχνολογικά πίσω από την Ευρώπη, η οποία καθιέρωσε το GSM, προσφέροντας περισσότερες υπηρεσίες και καλύτερη απόδοση στους συνδρομητές.

Σε όλα τα συστήματα κινητής τηλεφωνίας η γεωγραφική περιοχή υποδιαιρείται σε κυψέλες (cells), γι’ αυτόν τον λόγο οι συσκευές ονομάζονται μερικές φορές κυψελωτά τηλέφωνα (cell phones). Στο AMPS έχουν συνήθως διάμετρο 10 έως 20 χιλιόμετρα. Στα ψηφιακά συστήματα οι κυψέλες είναι μικρότερες. Κάθε κυψέλη χρησιμοποιεί κάποιο σύνολο συχνοτήτων το οποίο δεν χρησιμοποιείται από κανέναν από τους γείτονές της. Η βασική ιδέα που δίνει στα κυψελωτά συστήματα πολύ μεγαλύτερη χωρητικότητα από τα προηγούμενα συστήματα, είναι η χρήση σχετικά μικρών κυψελών και η επαναχρησιμοποίηση των συχνοτήτων μετάδοσης σε κοντινές, αλλά όχι γειτονικές, κυψέλες. Ενώ ένα σύστημα IMTS με διάμετρο 100 km, μπορεί να έχει μία κλήση σε κάθε συχνότητα, ένα σύστημα AMPS μπορεί να έχει 1000 κυψέλες των 10 km στην ίδια περιοχή, έτσι ώστε να μπορεί να έχει 10 με 15 κλήσεις σε κάθε συχνότητα, αλλά σε κυψέλες αρκετά απομακρυσμένες μεταξύ τους. Έτσι η κυψελωτή σχεδίαση αυξάνει την χωρητικότητα του συστήματος κατά μία τουλάχιστον τάξη μεγέθους και ακόμα περισσότερο όσο μικραίνουν οι κυψέλες. Επιπλέον, οι μικρότερες κυψέλες σημαίνει ότι χρειάζεται λιγότερη ισχύς, γεγονός που

οδηγεί σε μικρότερους και φτηνότερους πομπούς και συσκευές χειρός. Τα τηλέφωνα χειρός εκπέμπουν 0,6 Watt.

Ένα σημαντικό πρόβλημα είναι η ανεύρεση θέσεων σε ψηλά σημεία για την τοποθέτηση των κεραιών για τους σταθμούς βάσης. Αυτό το πρόβλημα έχει οδηγήσει μερικούς τηλεπικοινωνιακούς φορείς να συνάψουν συμμαχίες με την ρωμαιοκαθολική Εκκλησία, αφού αυτή έχει στην ιδιοκτησία της μεγάλο πλήθος υπερυψωμένων θέσεων για κεραίες σε όλο τον κόσμο, οι οποίες βρίσκονται όλες κάτω από την ίδια διαχείριση.

Σε περιοχές όπου το πλήθος τον χρηστών έχει αυξηθεί σε σημείο που το σύστημα να είναι υπερφορτωμένο, μειώνεται η ισχύς και οι υπερφορτωμένες κυψέλες διασπώνται σε μικρότερες κυψέλες, επιτρέποντας έτσι μεγαλύτερη επαναχρησιμοποίηση συχνοτήτων.

1.2 Δίκτυο GSM

Προκειμένου να γίνουν κατανοητές οι αδυναμίες και οι απειλές της υπηρεσίας SMS πρέπει πρώτα να περιγραφεί εν συντομία η λειτουργία του δικτύου GSM.

Στην κινητή τηλεφωνία χρησιμοποιείται κυρίως το πρότυπο σύστημα GSM που είναι τα αρχικά των λέξεων Global Systems for Mobile communication που σημαίνει παγκόσμιο σύστημα κινητών επικοινωνιών. Μέσα από τα δίκτυα GSM παρέχεται και η λειτουργία SMS. Το GSM είναι ένα ψηφιακό σύστημα, το οποίο επιτρέπει την ύπαρξη της υπηρεσίας SMS και των χαρακτηριστικών γνωρισμάτων τους.

Με το GSM όλα τα προβλήματα που αντιμετώπιζε η κινητή τηλεφωνία στα πρώτα της βήματα, εξαφανίστηκαν. Η συμφωνία των 13 ευρωπαϊκών τηλεπικοινωνιακών οργανισμών, προέβλεπε τα δίκτυα να κατασκευαστούν με τέτοια λεπτομέρεια και η ισχύς του σήματος να είναι τόσο ισχυρή, ώστε οι συσκευές να καταναλώνουν παρά πολύ λίγη ενέργεια. Με αυτό τον τρόπο, αυξήθηκε η διάρκεια λειτουργίας των συσκευών και ταυτόχρονα έγινε και μικρότερο και πιο πρακτικό το μέγεθός τους. Ταυτόχρονα, επέτρεπε και τη δυνατότητα πραγματοποίησης αυτοματοποιημένης κλήσης και χωρίς περιορισμούς από το δίκτυο.

Από τις αρχές της δεκαετίας του '90, ο χώρος των τηλεπικοινωνιών και της κινητής τηλεφωνίας γνώρισε ραγδαία ανάπτυξη. Τα δίκτυα 2G άρχισαν να επεκτείνονται με ταχύτατους ρυθμούς, ενώ στην αγορά κυκλοφορούσαν όλο και μικρότερα στο μέγεθος μοντέλα συσκευών κινητής τηλεφωνίας. Από τα μέσα της δεκαετίας του '90 ξεκίνησε και η υπηρεσία αποστολής και λήψης γραπτών μηνυμάτων (sms), ενώ στις αρχές του 21ου αιώνα παρουσιάζονται τα πρώρα κινητά με έγχρωμη οθόνη. Τα δίκτυα αναβαθμίστηκαν στο 2,5 G, με το οποίο έγινε δυνατή η αποστολή και λήψη φωτογραφιών και βίντεο, ενώ από το 2002 ξεκίνησε και η λειτουργία των πρώτων δικτύων 3G που επιτρέπει την βιντεοκλήση.

1.2.1 Το GSM στην Ελλάδα

Στην Ελλάδα, η ιστορία του GSM ξεκινάει το 1992 όταν η κυβέρνηση Μητσοτάκη προκήρυξε διαγωνισμό για την αδειοδότηση δύο γραμμών, τις οποίες κέρδισαν η Telestet και η Panafon. Πρώτη ξεκίνησε τη λειτουργία της στις 29 Ιουνίου 1993 η Telestet και δύο μέρες μετά, 1 Ιουλίου 1993, ξεκίνησε να προσφέρει τις υπηρεσίες της και η Panafon.

Αρχικά το κόστος των συσκευών και των υπηρεσιών ήταν απαγορευτικό. Επιπλέον, οι προβλέψεις δεν ήταν ιδιαίτερα αισιόδοξες για την ελληνική αγορά, καθώς οι ειδικοί υποστήριζαν πως ο αριθμός των χρηστών θα φτάσει μέχρι τα τέλη της δεκαετίας τους 200.000. Ωστόσο, γρήγορα επήλθε η διάψευση όλων των προβλέψεων. 13 χρόνια μετά, λειτουργούσαν στη χώρα μας 13.551.000 συσκευές (Δεκέμβριος 2006), που καλύπτουν το 120,5% του ελληνικού πληθυσμού, γεγονός που κατατάσσει την Ελλάδα στις πρώτες θέσεις παγκοσμίως σε αναλογία πληθυσμού και κινητών τηλεφώνων. Επιπλέον, το 1998 μπήκε στην αγορά και ο ΟΤΕ με τη θυγατρική του Cosmote, κερδίζοντας μέσα σε μικρό χρονικό διάστημα σημαντικό μερίδιο από την πελατεία. Ο ανταγωνισμός οδήγησε και σε μείωση των τιμών των υπηρεσιών.

Το 2003, εν' όψει και των Ολυμπιακών Αγώνων ξεκίνησε και η υπηρεσία 3G. Στο διάστημα των 15 χρόνων λειτουργίας της κινητής τηλεφωνίας με δίκτυο GSM στην Ελλάδα, οι δύο πρώτες εταιρείες εξαγοράστηκαν από αντίστοιχες του εξωτερικού (Panafon από τη Vodafone) και η Telestet από μεγάλο αιγυπτιακό όμιλο και φέρει την επωνυμία TIM, για να περάσει στην συνέχεια στα χέρια της Wind.

Τους πρώτους μήνες του 1993, τα κινητά τηλέφωνα λειτουργούσαν μόνο στην Αττική και τα νησιά του Σαρωνικού. Το κόστος ήταν απαγορευτικό για τους πολλούς. Οι συσκευές στοίχιζαν από 700-1400€, το τέλος ενεργοποίησης 85€, το μηνιαίο πάγιο 40€ και το λεπτό ομιλίας 0,25€. Έτσι, μόνο 1000 ήταν οι συνδρομητές τις πρώτες μέρες του Ιουλίου.

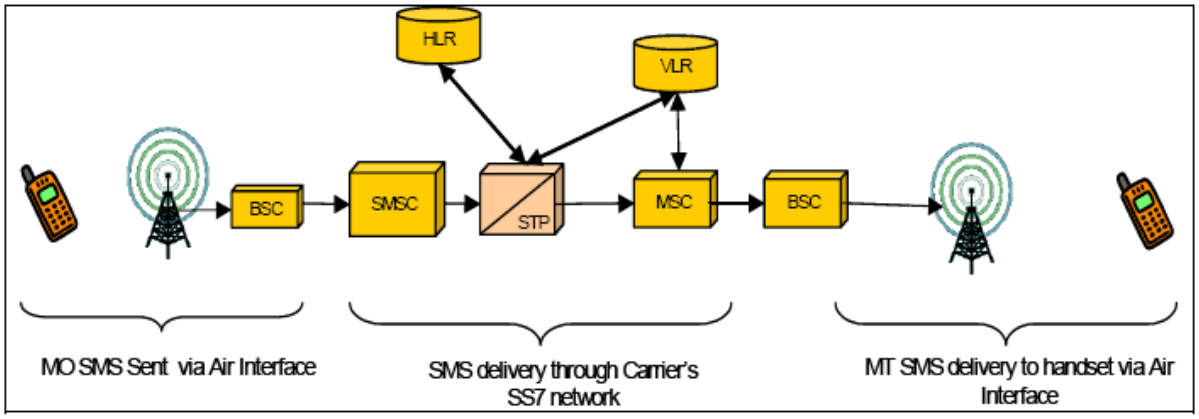
1.2.2 Η κινητή τηλεφωνία σε αριθμούς

- Το 1999 ο αριθμός των συσκευών κινητής τηλεφωνίας παγκοσμίως έφτασε το 1 δισεκατομμύριο, ενώ 30 μήνες μετά ξεπέρασε τα 2 δισεκατομμύρια συσκευές.
- Το 2015 ο αριθμός των κινητών τηλεφώνων παγκοσμίως θα φτάσει τα 4 δισεκατομμύρια.
- Κάθε μέρα γίνονται 1,2 εκατομμύρια νέες συνδέσεις κινητών τηλεφώνων σε όλο τον κόσμο.
- 445 εκατομμύρια φτάνει ο αριθμός των συνδέσεων και κατόχων κινητών τηλεφώνων στην Κίνα.
- Το 2007 πουλήθηκαν 1 δισεκατομμύριο συσκευές.
- Καθημερινά αποστέλλονται 7 δισεκατομμύρια γραπτά μηνύματα (sms).
- Το 64% των συνδρομητών κινητής τηλεφωνίας βρίσκονται στις αναπτυσσόμενες χώρες.

1.3 Περιγραφή του GSM

Οι περισσότερες χώρες χρησιμοποιούν το GSM. Οι Ηνωμένες Πολιτείες είναι μία από τις λίγες χώρες η οποία ευνοεί τη χρήση των προτύπων CDMA και TDMA πάνω από το GSM. Τα πρότυπα CDMA και TDMA επιτρέπουν εξαιρετικά περιορισμένες δυνατότητες στην υπηρεσία SMS. Υπάρχει και ένα άλλο πρότυπο το οποίο παρέχει περισσότερες δυνατότητες στην υπηρεσία, το GPRS (Γενική Υπηρεσία Ραδιομεταγωγής Πακέτων) το οποίο είναι γνωστό ως μία από τις καλύτερες νέες τεχνολογίες στον τομέα των ασύρματων επικοινωνιών. Το GPRS επιτρέπει την αποστολή πληροφοριών Διαδικτύου μέσω των κινητών συσκευών σε υψηλή ταχύτητα. Οι εμπειρογνώμονες θεωρούν ότι το GPRS μπορεί να μεταφέρει πληροφορίες με μία ταχύτητα της τάξεως των 100kbps, την στιγμή που το GSM μεταφέρει δεδομένα με ταχύτητα μόλις 9,6kbps.

Παρακάτω παρέχεται μία συνοπτική εικόνα για τα διάφορα στοιχεία που αποτελούν την υποδομή της κινητής τηλεφωνίας η οποία παρέχει και υπηρεσίες SMS.



1.1 Στοιχεία που αποτελούν το δίκτυο GSM.

1.4 Στοιχεία δικτύων GSM

1.4.1 Κινητός σταθμός – Mobile station

Αυτός είναι ο εξοπλισμός από όπου ο χρήστης πραγματοποιεί τις ασύρματες υπηρεσίες. (π.χ. κινητό τηλέφωνο, GSM modem).

1.4.2 Σύστημα πομποδεκτών βάσης /ελεγκτής σταθμών βάσης - Base Transceiver System/Base Station Controller (BTS/BSC)

Το σύστημα πομποδεκτών βάσης (BTS), είναι η σύνδεση μεταξύ του κέντρου κινητών υπηρεσιών (MSC) και του κινητού σταθμού μέσω ασύρματης διασύνδεσης. Το BTS αποτελείται από συσκευές εκπομπής και λήψης ράδιο-σημάτων τα οποία χρησιμοποιούνται για την μετάδοση φωνής, μηνυμάτων και πληροφοριών ελέγχου στο κινητό τηλέφωνο.

1.4.3 Κέντρο μεταγωγής κινητών υπηρεσιών - Mobile Switching Center (MSC)

Το MSC είναι το κύριο στοιχείο του δικτύου που συνδέει το δίκτυο PSTN με το κυψελοειδές τηλεφωνικό σύστημα. Διαχειρίζεται και δρομολογεί όλη την κυκλοφορία από και προς το κυψελοειδές σύστημα και επίσης είναι υπεύθυνο για να παρέχει διασύνδεση με άλλα δίκτυα. Εκτός από την διαχείριση και μεταγωγή των κλήσεων, το MSC ελέγχει την αυθεντικοποίηση και την επεξεργασία των στοιχείων του κάθε συνδρομητή. Το SMSC λειτουργεί ως συσκευή αποθήκευσης και προώθησης, που δέχεται τα μηνύματα και τα αποθηκεύει έως ότου αυτά να μπορέσουν να σταλούν στον προορισμό τους.

1.4.4 Οι κατάλογοι - Registers

Κατάλογος εγχώριων συνδρομητών - Home Location Register (HLR)

Ο κατάλογος εγχώριων συνδρομητών (HLR) είναι μία βάση δεδομένων που αποθηκεύει όλες τις πληροφορίες για τους χρήστες που ανήκουν στην περιοχή εξυπηρέτησης του MSC, όπως για παράδειγμα το σύνολο των υπηρεσιών που έχουν αγοράσει με το συμβόλαιο τους. Επίσης περιέχει την πιο πρόσφατη θέση του συνδρομητή.

Κατάλογος επισκεπτών συνδρομητών - Visitor Location Register (VLR)

Ο κατάλογος επισκεπτών συνδρομητών(VLR) είναι μία βάση δεδομένων που αποθηκεύει τις πληροφορίες των συνδρομητών που εξυπηρετούνται προσωρινά από το MSC της περιοχής αυτής. Τις πληροφορίες αυτές τις παίρνει από το HLR που ανήκει ο συγκεκριμένος συνδρομητής, ώστε να μπορεί να εξυπηρετείται για όσο διάστημα βρίσκεται στην περιοχή κάλυψης του συγκεκριμένου MSC.

1.4.5 Κέντρο Αυθεντικοποίησης - Authentication Center (AuC)

Το Κέντρο Αυθεντικοποίησης (AuC) αποθηκεύει παραμέτρους που χρησιμοποιούνται για την πιστοποίηση και την κρυπτογράφηση της ταυτότητας του χρήστη. Είναι μία προστατευμένη βάση δεδομένων που περιέχει το αντίγραφο του μυστικού κλειδιού που αποθηκεύεται στην κάρτα SIM του συνδρομητή και χρησιμοποιείται για την αυθεντικοποίηση και την κρυπτογράφηση των δεδομένων που μεταδίδονται ασύρματα.

1.4.6 Κατάλογος ταυτότητας συσκευών - Equipment Identity Register (EIR)

Κάθε συσκευή κινητού τηλεφώνου έχει έναν μοναδικό διεθνή αριθμό που λέγεται IMEI. Ο κατάλογος ταυτότητας συσκευών (EIR) είναι μία βάση δεδομένων που περιέχει αριθμούς μόνον έγκυρων συσκευών. Μέσω του EIR μπορούν να απαγορευτούν κλήσεις που γίνονται π.χ. από κλεμμένα κινητά.

1.4.7 Κέντρο υπηρεσίας σύντομων μηνυμάτων - Short Message Service Centre (SMSC)

Το κέντρο υπηρεσίας σύντομων μηνυμάτων (SMSC) δέχεται και αποθηκεύει ένα SMS έως ότου χρειαστεί, μέχρι αυτό να σταλεί στον προορισμό του.

1.4.8 Σύστημα σηματοδοσίας 7 - Signalling System 7 (SS7)

Το σύστημα σηματοδοσίας 7 (SS7) είναι το στοιχείο που είναι υπεύθυνο για την κίνηση κατά την μεταφορά των SMS. Είναι το πρωτόκολλο ελέγχου δικτύου που χρησιμοποιούν οι παγκόσμιοι φορείς παροχής τηλεφωνικών υπηρεσιών. Το SS7 αποτελείται από τέσσερα επίπεδα που ταιριάζουν, σε γενικές γραμμές, με τα επίπεδα

του προτύπου OSI. Η αρχική χρήση του SS7 ήταν ο έλεγχος του δικτύου. Επίσης χρησιμοποιείται για μεταφορά πληροφοριών και για τα ενσύρματα αλλά και για τα ασύρματα τηλεφωνικά συστήματα και έχει γίνει το πρότυπο σηματοδοσίας παγκοσμίως.

1.4.9 Σημείο μεταφοράς σηματοδοσίας - Signalling Transfer Point (STP))
Τα STP είναι δρομολογητές ενός SS7 δικτύου και παρέχουν δρομολόγηση των μηνυμάτων, μετάφραση των διευθύνσεων και οργάνωση των κλήσεων και των μεταπομπών.

1.5 Επισκόπηση παράδοσης μηνυμάτων

Τα μηνύματα μπορούν να εισαχθούν στο σύστημα της κυψελοειδής τηλεφωνίας είτε από τα κινητά τηλέφωνα που είναι μέρος αυτού του συστήματος, είτε από εξωτερικές πηγές όπως είναι το ηλεκτρονικό ταχυδρομείο. Όταν ένα μήνυμα εισέρχεται στο δίκτυο, παραδίδεται στο *κέντρο υπηρεσίας σύντομων μηνυμάτων* (SMSC). Το περιεχόμενο και οι πληροφορίες προορισμού του μηνύματος εξετάζονται από το SMSC και στην συνέχεια αντιγράφονται σε ένα κατάλληλα σχηματοποιημένο πακέτο, το οποίο τοποθετείται σε μία σειρά αναμονής εξόδου στο SMSC έως ότου εξυπηρετηθούν.

Προτού ένα SMSC διαβιβάσει ένα μήνυμα κειμένου στη κινητή συσκευή που προορίζεται το μήνυμα, πρέπει πρώτα να καθορίσει τη θέση αυτής της συσκευής. Για να γίνει αυτό, το SMSC ρωτά μία βάση δεδομένων γνωστή ως *κατάλογος εγχώριας θέσης* (HLR). Το SMSC στην συνέχεια λαμβάνει, εάν διατίθεται την συγκεκριμένη στιγμή, τη διεύθυνση του *κέντρου μεταγωγής κινητών υπηρεσιών* (MSC), το οποίο την συγκεκριμένη περίοδο παρέχει υπηρεσίες στη συσκευή προορισμού. Τέλος το MSC παραδίδει το μήνυμα ασύρματα μέσω των σταθμών βάσης (BS).

Προκειμένου να ειδοποιηθεί μία κινητή συσκευή ότι ένα μήνυμα κειμένου προορίζεται γι' αυτήν, στέλνεται ένα μήνυμα μέσω του *καναλιού ειδοποίησης (Paging Channel* - PCH). Το μήνυμα αυτό το λαμβάνουν όλες οι συσκευές που βρίσκονται στην περιοχή. Όταν η συσκευή προορισμού λάβει το μήνυμα από το κανάλι ειδοποίησης, αυτή με την σειρά της ενημερώνει το δίκτυο μέσω του ανερχόμενου καναλιού τυχαίας πρόσβασης *Random Access Channel (RACH)* uplink, για το εάν είναι έτοιμη να δεχτεί κάποια εισερχόμενη σύνδεση. Στην συσκευή ορίζεται έπειτα ένα *αυτόνομο αφιερωμένο κανάλι ελέγχου* (*Standalone Dedicated Control Channel* - SDCCH) λαμβάνοντας το από το *κανάλι ευρείας πρόσβασης* (*Access Grant Channel* - AGCH). Τέλος, ο σταθμός βάσης επιβεβαιώνει την ταυτότητα της συσκευής, ενεργοποιεί την κρυπτογράφηση, και παραδίδει έπειτα το περιεχόμενο του μηνύματος μέσω του αφιερωμένου καναλιού (SDCCH).

1.6 Υπηρεσίας ενισχυμένου μηνύματος - Enhanced Message Service (EMS)

EMS (Υπηρεσία ενισχυμένων μηνυμάτων) είναι μία αναπτυγμένη έκδοση του SMS. Το EMS εμπλουτίζει τις εφαρμογές που μεταδίδουν μηνύματα και επιτρέπει την αποστολή ενός συνδυασμού απλών μελωδιών, εικόνων, ήχων, κινούμενων εικόνων

(animation), τροποποιημένου και τυποποιημένου κειμένου. Επιπλέον, το EMS επιτρέπει την διασύνδεση των ασύρματων δικτύων κινητής τηλεφωνίας με το Διαδίκτυο, επιτρέποντας στους χρήστες να κατεβάζουν εικόνες και ήχους (ringtone) στο τηλέφωνό τους.

Το μειονέκτημα του EMS είναι ότι υποστηρίζεται λιγότερο από τις κινητές συσκευές απ' ότι η υπηρεσία SMS. Επίσης, πολλές συσκευές που διαθέτουν την λειτουργία αυτή, υποστηρίζουν μόνο ένα υποσύνολο των χαρακτηριστικών γνωρισμάτων που καθορίζονται στις προδιαγραφές του EMS. Μπορεί παράδειγμα ένα ορισμένο χαρακτηριστικό γνώρισμα του EMS να υποστηριχθεί από μία συσκευή αλλά όχι από μία άλλη. Έτσι δημιουργούνται ασυμβατότητες.

Η υπηρεσία ενισχυμένων μηνυμάτων έχει τυποποιηθεί μέσω της επέκτασης της χρήσης της κεφαλίδας δεδομένων χρήστη (User Data Header-UDH) που υπάρχει στα πρότυπα GSM και SMS. Το UDH δίνει την δυνατότητα εισαγωγής δυαδικών δεδομένων σε ένα κανονικό σύντομο μήνυμα, τοποθετημένο πριν από το κανονικό κείμενο.

Επειδή το EMS είναι βασισμένο στο πρότυπο SMS, μπορεί το SMSC να διαχειριστεί τα EMS με τον ίδιο τρόπο όπως και τα SMS. Το EMS λειτουργεί σε όλο το GSM σύστημα το οποίο χρησιμοποιείται ευρέως στις χώρες της Ευρώπης. Εάν ένα μήνυμα EMS σταλεί σε ένα τηλέφωνο που δεν υποστηρίζει την λειτουργία αυτή, ακόμα και τότε ο παραλήπτης θα λάβει μόνο το μέρος του μηνύματος που περιέχει απλό κείμενο.

1.7 Υπηρεσία μηνυμάτων πολυμέσων - Multimedia Message Service (MMS)

Η υπηρεσία μηνυμάτων πολυμέσων (Multimedia Message Service) είναι το επόμενο βήμα στην εξέλιξη των μηνυμάτων που μεταδίδονται ασύρματα, πέρα από το κείμενο, τις εικόνες και τα λογότυπα.

Η υπηρεσία MMS εξαρτάται από το σχηματισμό ενός νέου τύπου υποδομής δικτύων που είναι γνωστό ως 3G (τρίτη γενιά) και επιτρέπει στους χρήστες να στέλνουν μηνύματα που αποτελούνται από έναν συνδυασμό κειμένου, ήχων, εικόνων και βίντεο MMS, προς στις συσκευές που διαθέτουν την δυνατότητα να δεχτούν αυτού του είδους τα μηνύματα. Η υπηρεσία παρέχει αυτόματη και άμεση παράδοση των προσωπικών MMS από τηλέφωνο σε τηλέφωνο ή από τηλέφωνο στο ηλεκτρονικό ταχυδρομείο.

Το WAP (Wireless Application Protocol) μπορεί να χρησιμοποιηθεί ως μηχανισμός μεταφοράς των MMS. Με τη χρησιμοποίηση του προτύπου WAP ως μεταφορέας μηνυμάτων MMS, οποιοσδήποτε φορέας διαθέτει δυνατότητα χρήσης του WAP, μπορεί να μεταδώσει MMS. Αυτό κάνει την υπηρεσία να λειτουργεί και εκτός δικτύου GSM, αρκεί να υποστηρίζεται η λειτουργία WAP.

1.8 Ιστορικά

Η υπηρεσία MMS προωθήθηκε από τους διάφορους φορείς το 2003. Οι εταιρείες κινητής τηλεφωνίας υπέθεσαν ότι η μεγάλη δημοτικότητα των SMS μεταφραζόταν σε απαίτηση για MMS.

Στην αρχή τα MMS δεν ήταν επιτυχή γιατί περιορίστηκαν από ζητήματα διαλειτουργικότητας μεταξύ των δικτύων κινητής και την έλλειψη συσκευών που θα μπορούσαν να λάβουν αυτά τα μηνύματα. Επιπλέον η αποστολή των μηνυμάτων αποδείχθηκε μία πρόκληση για τους χρήστες λόγω των περίπλοκων και μεγάλων διαδικασιών με συνέπεια τα περισσότερα μηνύματα που εστάλησαν τον πρώτο καιρό να μην φτάσουν ποτέ στον προορισμό τους. Εντούτοις, οι αναλυτές προέβλεψαν ότι το έτος 2007 η κυκλοφορία των SMS θα μειωθεί σημαντικά λόγω του ότι τα MMS θα γίνουν το κύριο μέσο ανταλλαγής για την κινητή επικοινωνία.

Ο ακόλουθος πίνακας παρουσιάζει και συγκρίνει όλες τις τεχνολογίες μηνύματος που έχουν αναφερθεί ήδη.

Type	Characteristics	Content Formatting	Applications	Support	Timeframe
Text Messaging	100-200 characters	Yes	Simple person to person messaging	All phones	1990s onwards
Smart Messaging	Simple rudimentary images and ringtones	Yes	Simple person to person messaging with a visual feel	EMS standards expected to be widely adopted	2001 onwards
Enhanced Messaging	Text messages plus simple media formats e.g. sound, animation, picture, text formating enhancements	Yes	Simple person to person messaging with a visual feel	EMS standards expected to be widely adopted	2001 onwards
Multimedia Messaging	Messages in multiple rich media formats e.g. video, audio plus text	Sometimes	Person or server to person messaging with rich image and video content	MMS standards becoming widely adopted	2002 onwards

1.2 Πίνακας όλων των τύπων μηνυμάτων με τα στοιχεία τους.

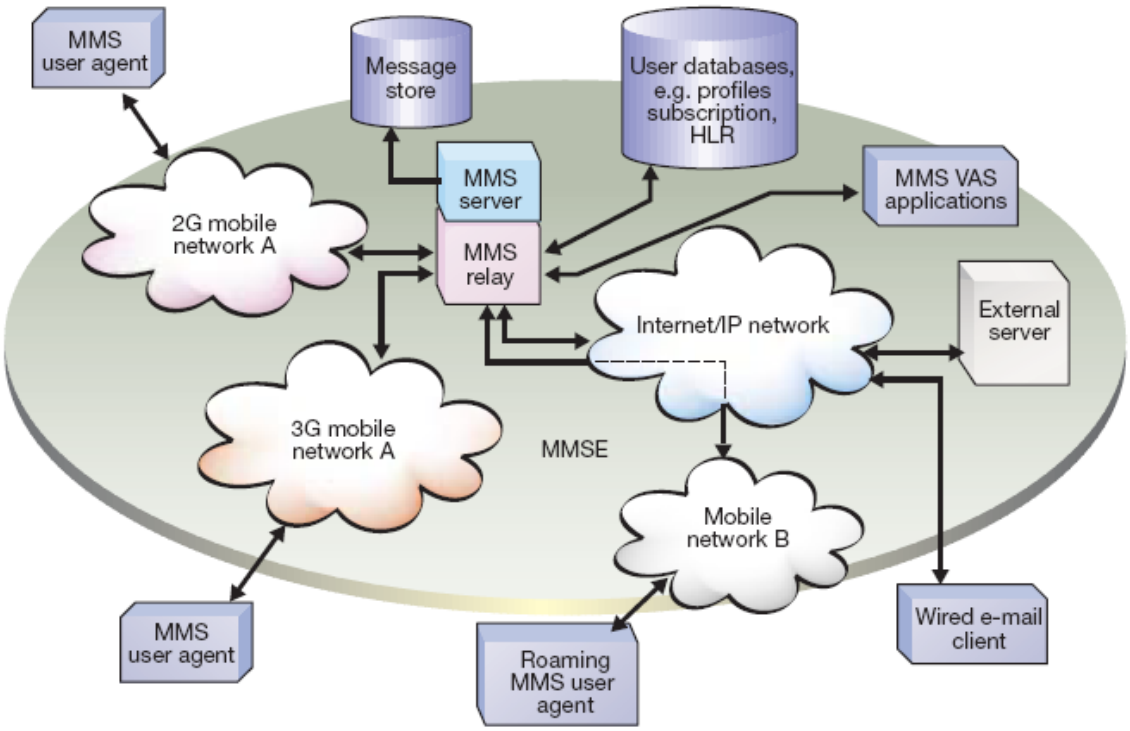
1.9 MMS δίκτυο

Το περιβάλλον υπηρεσίας μηνυμάτων πολυμέσων (MMSE) συνδυάζει τους διαφορετικούς τύπους δικτύων και ενσωματώνει τα ήδη υπάρχοντα συστήματα μετάδοσης μηνυμάτων σε αυτά τα δίκτυα. Το περιβάλλον MMS καλύπτει όλα τα απαραίτητα στοιχεία υπηρεσιών για την αποθήκευση και την παράδοση των μηνυμάτων. Αυτά μπορούν να βρεθούν μέσα σε ένα δίκτυο ή να διανεμηθούν σε διάφορων τύπων δίκτυα.

Το MMSE μπορεί να περιλάβει:

- Δίκτυα δεύτερης και τρίτης γενιάς.
- Δεύτερης γενεάς δίκτυα με κάλυψη τρίτης γενιάς.
- Δίκτυα περϊαγωγής.

Η συνδεσιμότητα μεταξύ διαφορετικών τύπων δικτύων παρέχεται μέσω του πρωτόκολλου του Διαδικτύου (IP).



1.3 - Δίκτυο MMS

1.10 Στοιχεία δικτύων τύπου MMS

1.10.1 Διακομιστής MMS – MMS Server

Ο διακομιστής MMS είναι υπεύθυνος για την αποθήκευση και την διαχείριση των εισερχόμενων και εξερχόμενων μηνυμάτων.

1.10.2 Proxy Delay MMS

Ο Proxy Delay MMS (PDMMS) είναι συνδεδεμένος με τον διακομιστή MMS και είναι υπεύθυνος για την μεταφορά των MMS μεταξύ διαφορετικών συστημάτων μετάδοσης μηνυμάτων. Ο διακομιστής και ο PDMMS μπορούν να είναι είτε χωριστά είτε να έχουν συνδυαστεί είτε να έχουν διανεμηθεί σε διαφορετικές περιοχές.

Ο PD μπορεί να παράγει τα στοιχεία χρέωσης (αρχείο λεπτομερειών κλήσεων - Call Detail Record CDR) κάθε φορά που λαμβάνει ή στέλνει μηνύματα πολυμέσων. Είναι επίσης αρμόδιος για τη μετατροπή των μηνυμάτων, δηλαδή προσαρμόζει τα μηνύματα στις δυνατότητες της συσκευής που πρόκειται να το λάβει. Παραδείγματος χάριν, εάν μία νέα συσκευή MMS στείλει μία έγχρωμη εικόνα υψηλής ανάλυσης σε μία παλαιότερη συσκευή που υποστηρίζει μόνο ασπρόμαυρες εικόνες χαμηλής ανάλυσης, τότε ο PD θα μετατρέψει την εικόνα σε κατάλληλη μορφή ώστε αυτή να διαβαστεί από την παλιά συσκευή. Αυτή η λειτουργία εφαρμόζεται σε όλο το περιεχόμενο του μηνύματος, όπως στα βίντεο κλιπ, στις εικόνες και στα αρχεία MP3. Το χαρακτηριστικό αυτό είναι νέο στην τεχνολογία MMS και δεν υπάρχει στα SMS.

1.10.3 Βάση Δεδομένων χρηστών MMS – MMS User Database

Αυτή η βάση δεδομένων περιέχει πληροφορίες σχετικές με τους χρήστες, όπως η συνδρομή ή διάφορα στοιχεία του χρήστη όπως ο κατάλογος εγχώριων συνδρομητών (HLR) που είναι εγγεγραμμένος ο συγκεκριμένος συνδρομητής.

1.10.4 MMS User Agent

Είναι μία λειτουργία στο επίπεδο εφαρμογής του δικτύου και βρίσκεται μέσα στις κινητές συσκευές και επιτρέπει στους χρήστες να συνθέσουν, να στείλουν, να λάβουν και γενικά να διαχειριστούν μηνύματα πολυμέσων.

1.11 Εξέλιξη από το GSM στο 3G

Το 3G είναι τα αρχικά των λέξεων **3**rd **G**eneration και αποτελεί ένα γενικό όρο ο οποίος αναφέρεται στην τρίτη γενιά τεχνολογίας κινητής τηλεφωνίας. Ως γενιά χαρακτηρίζεται το σύνολο των ασύρματων τεχνολογιών που επιτρέπουν τη μετάδοση φωνής ή και δεδομένων στα δίκτυα κινητής τηλεφωνίας. Μεταξύ των τεχνολογιών αυτών είναι οι W-CDMA, CDMA2000, UMTS και EDGE. Το W-CDMA ή Wideband Code Multiple Division Access, έχει ήδη επιλεχθεί ως το "σύστημα" τρίτης γενιάς, που θα χρησιμοποιηθεί στην Ευρώπη, Ιαπωνία και στις ΗΠΑ.

Σε αντίθεση με προηγούμενες τεχνολογίες που βασίζονται σε μετάδοση δεδομένων πάνω από απευθείας σύνδεση των δύο μερών (circuit-switched), οι τεχνολογίες που ανήκουν στην ομάδα τρίτης γενιάς βασίζονται σε υψηλής ταχύτητας μετάδοση δεδομένων μοιρασμένων σε πακέτα (packet-switched). Η τεχνολογία στηρίζεται στα γνωστά δίκτυα GSM με μία παραλλαγή του CDMA με το όνομα W-CDMA (Wideband-CDMA) η οποία είναι ικανή να επιτύχει ταχύτητες μετάδοσης έως και 2Mbps. Στα άμεσα επερχόμενα δίκτυα τρίτης γενιάς όμως, ο συνδυασμός του W-

CDMA με τις υπό διάθεση συσκευές θα είναι ικανός να προσφέρει στον τελικό χρήστη ταχύτητες έως και 384Kbps, οι οποίες όμως είναι αρκετές για να μετατρέψουν το κινητό σε μία ασύρματη συσκευή πολυμέσων.

1.12 Ιστορικά

Το 1992 η ITU εξέδωσε ένα πρόγραμμα δράσης το οποίο ονομαζόταν IMT-2000 όπου τα αρχικά σημαίνουν **Διεθνής Κινητές Τηλεπικοινωνίες** (International Mobile Telecommunications). Ο αριθμός 2000 σήμαινε τρία πράγματα: (1) το έτος στο οποίο υποτίθεται ότι το σύστημα θα έμπαινε σε λειτουργία, (2) τη συχνότητα στην οποία υποτίθεται ότι θα λειτουργούσε (σε MHz), και (3) το εύρος ζώνης που θα έπρεπε να έχει η υπηρεσία (σε kHz).

Το σύστημα δεν τα κατάφερε σε κανέναν από τους τρεις τομείς. Μέχρι το 2000 δεν είχε υλοποιηθεί τίποτα. Η ITU σύστησε σε όλες τις κυβερνήσεις να δεσμεύσουν φάσμα στα 2 GHz έτσι ώστε οι συσκευές να μπορούν να εκτελούν με διαφάνεια περϊαγωγή από χώρα σε χώρα. Η Κίνα δέσμευσε το απαιτούμενο εύρος ζώνης, αλλά κανείς άλλος δεν το έκανε. Τελικά αναγνωρίστηκε ότι τα 2 Mbps δεν είναι προς το παρόν εφικτά για τους χρήστες που είναι πολύ κινητοί (λόγω της δυσκολίας στο να εκτελούνται οι μεταβιβάσεις ανάμεσα στις κυψέλες). Μία πιο ρεαλιστική εκτίμηση είναι τα 2 Mbps για ακίνητους χρήστες εντός κτιρίων, 384 kbps για ανθρώπους που περπατούν και 144 kbps για συνδέσεις από αυτοκίνητα.

Οι βασικές υπηρεσίες τις οποίες υποτίθεται ότι θα παρέχει το δίκτυο IMT-2000 στους χρήστες του είναι:

1) Μετάδοση φωνής υψηλής ποιότητας.
2) Ανταλλαγή μηνυμάτων (αντικαθιστώντας το ηλεκτρονικό ταχυδρομείο, το φαξ, το SMS, την ηλεκτρονική συνομιλία, κτλ)
3) Πολυμέσα (αναπαραγωγή μουσικής, προβολή βίντεο, ταινιών, τηλεόρασης, κτλ.)
4) Πρόσβαση στο Internet (Περιήγηση στον Ιστό, ακόμα και σε σελίδες με ήχο και βίντεο)

Άλλες υπηρεσίες μπορεί να είναι η εικονοδιάσκεψη, η τηλεπαρουσία (telepresence), τα ομαδικά παιχνίδια κ.α. Επιπλέον, όλες αυτές οι υπηρεσίες υποτίθεται ότι θα είναι διαθέσιμες σε παγκόσμια κλίμακα (με αυτόματη σύνδεση μέσω δορυφόρου όταν δεν μπορεί να εντοπιστεί κάποιο επίγειο δίκτυο), θα είναι συνεχώς ενεργές και με εγγυήσεις ως προς την ποιότητα υπηρεσιών.

Η εταιρεία Ericsson πρότεινε για την τεχνολογία 3G το πρότυπο W-CDMA το οποίο προωθήθηκε έντονα από την Ευρωπαϊκή Ένωση η οποία το ονόμασε Παγκόσμιο Σύστημα Κινητών Τηλεπικοινωνιών ή UMTS(Universal Mobile Telecommunication System). Το σύστημα αυτό χρησιμοποιεί εξάπλωση φάσματος άμεσης ακολουθίας και δουλεύει σε εύρος ζώνης 5 MHz. Έχει σχεδιαστεί για να συνεργάζεται με δίκτυα GSM, αν και δεν είναι συμβατό προς τα πίσω με αυτό. Παρ' όλα αυτά όμως έχει την ιδιότητα ότι ένας χρήστης μπορεί να βγει από μία κυψέλη W-CDMA και να μπει σε μία GSM χωρίς να χαθεί η κλήση του.

Διάφοροι φορείς κινητής τηλεφωνίας προσθέτουν κάποιες τεχνολογίες στο υπάρχον δίκτυο ως μεταβατικό στάδιο προς το 3G το οποίο ονομάζεται 2.5G. Μία τέτοια τεχνολογία είναι το GPRS ή Γενική Υπηρεσία Ραδιομεταγωγής Πακέτων. Το GPRS είναι ένα δίκτυο μεταγωγής πακέτων πάνω από το GSM το οποίο επιτρέπει στους κινητούς σταθμούς να στέλνουν και να λαμβάνουν πακέτα IP σε μία κυψέλη η οποία χρησιμοποιεί κάποιο σύστημα φωνής. Όταν υποστηρίζεται το GPRS μερικές χρονικές υποδοχές σε κάποια κανάλια δεσμεύονται για κίνηση πακέτων δεδομένων.

1.13 Τεχνικά χαρακτηριστικά και λειτουργία

Η τρίτη γενιά κινητών τηλέφωνων (3G) χαρακτηρίζονται από τον υψηλότερο ρυθμό μετάδοσης δεδομένων και μίας μεγαλύτερης σειράς υπηρεσιών. Το παγκόσμιο σύστημα κινητών τηλεπικοινωνιών (UMTS) είναι ένα από τα νέα συστήματα 3G. Το UMTS εισάγει ένα νέο δίκτυο ασύρματης πρόσβασης, το οποίο συνδέεται με μία εξέλιξη του κεντρικού δικτύου GSM. Αυτή η νέα τεχνολογία ονομάζεται πολλαπλή πρόσβαση ευρείας ζώνης με διαίρεση κώδικα (W-CDMA). Το 3G έχει στηριχτεί στην ασφάλεια του GSM. Αυτό σημαίνει ότι έχουν προστεθεί όλα τα χαρακτηριστικά ασφάλειας του GSM, τα οποία είναι χρήσιμα και ισχυρά και επίσης έχουν προστεθεί και επιπλέον καινούργια.

Όσον αφορά την αυθεντικοποίηση στο UMTS υπάρχει αμοιβαία επικύρωση και συμφωνία για τα κλειδιά μεταξύ του συνδρομητή και του δικτύου. Ο συνδρομητής καθορίζεται από μία έξυπνη κάρτα, γνωστή ως USIM, η οποία είναι η αντίστοιχη SIM στο GSM. Ο μηχανισμός επικύρωσης χρηστών στο 3G παρέχει προστασία ενάντια στις επιθέσεις ψεύτικων σταθμών βάσης επιτρέποντας στο κινητό να επικυρώνει το δίκτυο, πράγμα που δεν γίνεται στο GSM. Επιπλέον υπάρχει ενισχυμένη προστασία ακεραιότητας των κρίσιμων μεταδόσεων μεταξύ του κινητού και του ελεγκτή του δικτύου.

Στα δίκτυα 3G, τα δεδομένα κρυπτογραφούνται μεταξύ του κινητού εξοπλισμού και του ελεγκτή του ασύρματου δικτύου (RNC). Με αυτόν τον τρόπο, η ασύρματη κυκλοφορία και τα ευαίσθητα δεδομένα των χρηστών που μεταδίδονται προστατεύονται από τις υποκλοπές. Επιπλέον εφαρμόζεται επικύρωση καθ' όλη την διάρκεια κάποιας κλήσης.

Για την κρυπτογράφηση χρησιμοποιείται κλειδί μήκους 128 bit που παράγεται κατά τη διάρκεια της αυθεντικοποίησης. Το 3G χρησιμοποιεί επίσης και ένα κλειδί ακεραιότητας δεδομένων μήκους επίσης 128 bit που παράγεται και αυτό κατά τη διάρκεια της επικύρωσης, προκειμένου να εξασφαλιστεί η ακεραιότητα των δεδομένων. Ο αλγόριθμος προστασίας της ακεραιότητας είναι ο UIA1, ο οποίος βρίσκεται στο τηλέφωνο του χρήστη και σε κάθε ελεγκτή του ασύρματου δικτύου και έχει τυποποιηθεί έτσι ώστε όλα τα κινητά και οι ελεγκτές των σταθμών βάσης να μπορούν να επικοινωνούν μεταξύ τους. Αυτός ο αλγόριθμος είναι βασισμένος σε έναν τρόπο λειτουργίας που ονομάζεται KASUMI.

Στο UMTS, η μόνιμη ταυτότητα χρηστών (IMSI) ενός χρήστη, στον οποίο παρέχονται υπηρεσίες, δεν μπορεί να υποκλαπεί κρυφακούγοντας το ασύρματο κανάλι. Επίσης δεν μπορεί να καθοριστεί η παρουσία ή η άφιξη ενός χρήστη σε μία ορισμένη περιοχή και ένας εισβολέας δεν μπορεί να διαπιστώσει εάν διαφορετικές

υπηρεσίες παραδίδονται στον ίδιο χρήστη. Για να επιτύχει αυτούς τους στόχους, ο χρήστης προσδιορίζεται στο δίκτυο από μία προσωρινή ταυτότητα. Επίσης, για να αποφύγει αυτός ο χρήστης την ανιχνευσιμότητα, που μπορεί να οδηγήσει στην επιβεβαίωση της ταυτότητάς του, δεν θα πρέπει να προσδιορίζεται για μεγάλο χρονικό διάστημα με την ίδια προσωρινή ταυτότητα.

Επιπλέον υπάρχει η απαίτηση ότι οποιαδήποτε δεδομένα χρηστών μεταδίδονται ασύρματα και που θα μπορούσαν να αποκαλύψουν την ταυτότητα του χρήστη, θα πρέπει να κρυπτογραφούνται. Η επιβεβαίωση της ταυτότητας του χρήστη έχει ενισχυθεί με την χρήση κλειδιών ομάδας τα οποία είναι κλειδιά που χρησιμοποιούνται από μία ομάδα χρηστών. Πολύ σημαντικό είναι το γεγονός ότι οι χρήστες ενημερώνονται εάν η ασφάλεια είναι ενεργοποιημένη και ποιο επίπεδο ασφάλειας είναι διαθέσιμο και είναι σε θέση να διαμορφώσει τα χαρακτηριστικά γνωρίσματα ασφάλειας για τις μεμονωμένες υπηρεσίες.

Τα πλεονεκτήματα της τεχνολογίας τρίτης γενιάς είναι πολλά. Παρακάτω απαριθμούνται τα σημαντικότερα οφέλη της τεχνολογίας 3G:

- Οι βίντεο-κλήσεις είναι χωρίς αμφιβολία μία από τις πιο πολυσυζητημένες υπηρεσίες των δικτύων 3G. Πλέον, εκτός από το να ακούγεται ο συνομιλητής θα εμφανίζεται επίσης και ζωντανά στην οθόνη του κινητού. Φυσικά, θα πρέπει να έχουν και οι δύο συνομιλητές κάποια συμβατή συσκευή 3G.
- Οι υψηλές ταχύτητες ασύρματης μεταφοράς δεδομένων είναι ένα ακόμη από τα πλεονεκτήματα των δικτύων 3G. Η σύνδεση στο Internet εκτός από άμεση και απρόσκοπτη, θα δώσει πλέον ταχύτητες που φθάνουν τα 384kbps.
- Οι υψηλές ταχύτητες μεταφοράς δεδομένων βοηθούν αρκετά στην πιο γρήγορη και άμεση χρήση διαφόρων multimedia εφαρμογών. Έτσι, η αποστολή ενός MMS δεν θα παίρνει περισσότερο από 10 δευτερόλεπτα από την στιγμή που στα δίκτυα 2G ο χρόνος αυτός μπορεί να ξεπεράσει και το 1 λεπτό.
- Το video-streaming είναι μία ακόμη από τις υπηρεσίες που παρέχουν τα δίκτυα 3G. Το αυξημένο bandwidth επιτρέπει τη μετάδοση σε πραγματικό χρόνο, κινούμενης εικόνας και ήχου υψηλής ανάλυσης. Έτσι, παρέχεται η δυνατότητα παρακολούθησης τηλεοπτικών προγραμμάτων, ζωντανά ή μαγνητοσκοπημένα, ανεξαρτήτως τόπου και χρόνου.
- Υψηλής ποιότητας παιχνίδια, τα οποία θα μπορούν να παίζονται online σε πραγματικό χρόνο και ταυτόχρονα με άλλους παίκτες.
- Υπηρεσίες εύρεσης θέσεως, σε συνδυασμό με την τεχνολογία GPS, οι οποίες θα μπορούν να παρέχουν χάρτες τη περιοχής που βρίσκεται ο χρήστης, εύρεση βέλτιστης διαδρομής προς κάποιον προορισμό, γειτονικά σημεία ενδιαφέροντος κλπ.

Μετά την ευρεία διείσδυση της τεχνολογίας 3G αναμένεται να διατεθούν ακόμη περισσότερες υπηρεσίες, όπως μετάδοση τηλεοπτικών εκπομπών και υπηρεσίες παγκόσμιας περιαγωγής.

Για την παροχή των υπηρεσιών 3G απαιτείται κάποιο κινητό τηλέφωνο που είναι συμβατό με την τεχνολογία τρίτης γενιάς και φυσικά το αντίστοιχο πρόγραμμα υπηρεσιών του δικτύου κινητής τηλεφωνίας. Φυσικά για την εκμετάλλευση

υπηρεσιών όπως η τηλεδιάσκεψη, θα πρέπει και ο συνομιλητής να διαθέτει ανάλογη συσκευή και πρόγραμμα χρήσης.

1.14 Συσκευές 3G

Το πρώτο εξάμηνο του 2004, όπου και ξεκίνησε η εμπορική διάθεση των υπηρεσιών τρίτης γενιάς στην Ελλάδα, οι συσκευές 3G που διαθέτονταν στην ελληνική αγορά ήταν ελάχιστες. Η πρώτη από αυτές ήταν το Nokia 6650 που έκανε την εμφάνισή του στο δίκτυο της TIM (πλέον Wind), χωρίς ωστόσο να υποστηρίζει υπηρεσίες όπως οι βίντεο-κλήσεις. Λίγο αργότερα ακολούθησαν τα Motorola A835 και Siemens U15, που ουσιαστικά δεν έχουν καμία απολύτως διαφορά. Και οι δύο συσκευές ήταν συμβατές με όλες τις υπηρεσίες που παρέχουν τα δίκτυα τρίτης γενιάς, αν και το κόστος τους είναι ακόμη και σήμερα απαγορευτικό.

Το δεύτερο 3G κινητό της Nokia που έφθασε στη Ελλάδα είναι το Nokia 7600, αν και δυστυχώς ούτε αυτό επιτρέπει την πραγματοποίηση βίντεο-κλήσεων. Προτέρημά του το χαμηλό κόστος και το ιδιόμορφο design του.

Αργότερα, κυκλοφόρησε το Sony Ericsson Z1010, ενώ σταδιακά παρουσιάζονται πολλές ακόμη συσκευές τρίτης γενιάς από όλες τις μεγάλες εταιρείες όπου οι χρήστες μπορούν να χρησιμοποιήσουν, στις περισσότερες των περιπτώσεων, όλες τις καινοτομίες που προσφέρει η 3η γενιά. Ενδεικτικά, οι πρώτες πλήρης προτάσεις συσκευών παρείχε η Nokia με το 6680, η Sony Ericsson με το Z1010 και τα μεταγενέστερα V800 και Z800i, η Motorola με το E1000 και άλλα.

Τα ελάχιστα χαρακτηριστικά που θα πρέπει να είναι σε θέση να υποστηρίζει το δίκτυο είναι τα ακόλουθα:

1. Δυνατότητες για πολυμέσα και εφαρμογές πλήρους και χαμηλής κινητικότητας σε διαφορετικά γεωγραφικά περιβάλλοντα, πέραν των δυνατοτήτων των συστημάτων της 2ης γενιάς, όπως το GSM1800 /GSM900.

2. Αποτελεσματική πρόσβαση στο Διαδίκτυο (Internet), στα Εσωτερικά Διαδίκτυα (Intranets) και λοιπές υπηρεσίες που βασίζονται στο Πρωτόκολλο Διαδικτύου (IP).

3. Μετάδοση ομιλίας σε υψηλή ποιότητα, αντίστοιχη με την ποιότητα των σταθερών δικτύων.

4. Φορητότητα υπηρεσιών ανεξάρτητα από το εκάστοτε περιβάλλον UMTS/IMT-2000, όπου απαιτείται (π.χ. δημόσιο/ιδιωτικό/επαγγελματικό και σταθερό/κινητό).

5. Συνεχές και αδιάκοπο περιβάλλον λειτουργίας, συμπεριλαμβανομένης πλήρους περιαγωγής με GSM1800/GSM900, καθώς επίσης μεταξύ των επίγειων και δορυφορικών συνιστωσών των δικτύων UMTS/IMT-2000.

6. Νέα διεπαφή για πρόσβαση σε όλες τις υπηρεσίες, συμπεριλαμβανομένων των υπηρεσιών δεδομένων μεταγωγής πακέτων, η οποία θα υποστηρίζει την ασύμμετρη κίνηση και θα επιτρέπει τη διάθεση εύρους ζώνης και ρυθμού μετάδοσης ανάλογα με τη ζήτηση σε εναρμονισμένες ζώνες ραδιοσυχνοτήτων.

7. Καλή γενική φασματική απόδοση, συμπεριλαμβανομένης και της χρήσης ζευγών ραδιοσυχνοτήτων και απλών ραδιοσυχνοτήτων.

8. Διαχείριση των κλήσεων, έλεγχος των υπηρεσιών και διαχείριση της θέσης και της κινητικότητας, συμπεριλαμβανομένης δυνατότητας περιαγωγής σε πλήρη κλίμακα, βάσει της εξέλιξης των υφισταμένων συστημάτων δικτύων πυρήνα, παραδείγματος χάριν βάσει ενός εξελιγμένου δικτύου πυρήνα GSM, λαμβάνοντας υπόψη τη σύγκλιση μεταξύ κινητών/σταθερών δικτύων.

ΚΕΦΑΛΑΙΟ 2 - Ασφάλεια GSM

2.1 Δίκτυο GSM – Ασφάλεια και ευπάθεια

Όπως με οποιοδήποτε σύστημα που συνδέεται με το Διαδίκτυο, έτσι και το δίκτυο GSM και συγκεκριμένα το σύστημα SMS, θα γίνει κάποια στιγμή στόχος κακόβουλης επίθεσης.

Ακόμα κι αν τα κυψελοειδή δίκτυα λειτουργούν χωριστά από το Διαδίκτυο και θεωρούνται μερικές φορές ασφαλέστερα και λιγότερο ανοιχτά στην κακόβουλη χρήση (όπως το spam), εντούτοις όλα τα υπάρχοντα συστήματα έχουν διάφορες πιθανές αδυναμίες. Αν και αυτές οι αδυναμίες έχουν εξεταστεί στις απαιτήσεις ασφάλειας του GSM, αυτό εξακολουθεί να είναι ακόμα ανασφαλές. Το γεγονός ότι μόνο η επικοινωνία μεταξύ των κινητών συσκευών και των BTS κρυπτογραφείται και υπάρχει έλλειψη αμοιβαίας επικύρωσης (το δίκτυο δεν αυθεντικοποιείται στους συνδρομητές), συμβάλλει στην ανασφάλεια του GSM. Επιπλέον, η έλλειψη ακεραιότητας στοιχείων η οποία μεταφράζεται ως απώλεια στοιχείων και η δυσκολία αναβάθμισης στους τομείς ασφάλειας, καθιστά το GSM ακόμα πιο ανασφαλές.

Τα SMS μεταδίδονται μέσω της ασύρματης διασύνδεσης με την χρήση ηλεκτρομαγνητικών κυμάτων που περιέχουν τις πληροφορίες του μηνύματος. Η ασύρματη διεπαφή όμως έχει κάποιες αδυναμίες ασφαλείας λόγω του ότι ο οποιοσδήποτε που βρίσκεται στην περιοχή κάλυψης του σήματος, μπορεί να ακούσει τις πληροφορίες που εκπέμπονται μέσα σε αυτήν την περιοχή.

Αυτές οι αδυναμίες είναι:

- Μετάδοση μέσω του αέρα: οποιοσδήποτε μπορεί να ακούσει (μυστικότητα)
- Τα ραδιοκύματα δεν σταματούν στους εταιρικούς τοίχους ή σε άλλα τεχνητά σύνορα (μυστικότητα)
- Το φάσμα των ραδιοκυμάτων είναι περιορισμένο (διαθεσιμότητα)
- Χωρίς κάλυψη δικτύων δεν θα υπάρχει διαθέσιμη υπηρεσία (διαθεσιμότητα)

Οι κύριοι στόχοι ασφάλειας που το δίκτυο GSM πρέπει να εξετάσει καθορίζονται ως εξής:

Διαθεσιμότητα: Η διαθεσιμότητα αποτελείται από τη διαθεσιμότητα πληροφοριών στους χρήστες που έχουν το δικαίωμα πρόσβασης σε αυτό το κομμάτι των πληροφοριών. Για το χρήστη που του έχει χορηγηθεί πρόσβαση στις πληροφορίες, οι πληροφορίες πρέπει πάντα να είναι προσιτές χωρίς πρόσθετες προσπάθειες που καταβάλλονται για να τις ανακτήσουν

Ακεραιότητα: Η ακεραιότητα είναι η διαβεβαίωση ότι οι πληροφορίες που λαμβάνονται από τον παραλήπτη είναι ακριβώς αυτές που έστειλε ο αποστολέας και οποιαδήποτε τροποποίησή τους από μη εξουσιοδοτημένα άτομα γίνεται άμεσα αντιληπτή. Τα μέτρα που λαμβάνονται για να εξασφαλίσουν ακεραιότητα περιλαμβάνουν τον έλεγχο του φυσικού περιβάλλοντος των δικτυωμένων τερματικών και των κεντρικών υπολογιστών, που περιορίζει την πρόσβαση στα δεδομένα και που διατηρεί τις αυστηρές πρακτικές επικύρωσης. Η ακεραιότητα δεδομένων μπορεί

επίσης να απειληθεί από τους περιβαλλοντικούς παράγοντες, όπως η θερμότητα, η σκόνη, και τα ηλεκτρικά κύματα.

Εμπιστευτικότητα: Η εμπιστευτικότητα είναι ένα μεγάλο μέρος της ιδιωτικότητας και της εμπιστοσύνης. Οι πληροφορίες πρέπει να μένουν μυστικές και να προσπελάζονται μόνο από άτομα που έχουν εξουσιοδότηση. Για να μένουν μυστικές οι πληροφορίες χρησιμοποιείται ένα προσωρινό κλειδί κρυπτογράφησης που παράγεται από το αλγόριθμο A8: A8 (Ki, RAND) = > kc, όπου Ki είναι ένα μυστικό κλειδί, μοναδικό για κάθε κινητό και RAND ένας τυχαίος αριθμός. Για την κρυπτογράφηση χρησιμοποιείται ο αλγόριθμος A5. Η κρυπτογράφηση των δεδομένων μπορεί να παρασταθεί ως εξής: A5 (kc, δεδομένα) = > κρυπτογραφημένα δεδομένα.

***Ανωνυμία*:** Η ανωνυμία συνίσταται στην παρεμπόδιση της ανίχνευσης της θέσης του χρήστη ή του προσδιορισμού των κλήσεων που δέχεται ή κάνει, κρυφακούγοντας το κανάλι των ραδιοκυμάτων. Η ανωνυμία στο GSM παρέχεται με τη χρησιμοποίηση προσωρινής ταυτότητας. Όταν ένας χρήστης ενεργοποιεί αρχικά την συσκευή, χρησιμοποιείται η πραγματική ταυτότητα και στην συνέχεια εκδίδεται μία προσωρινή. Από εκεί και έπειτα χρησιμοποιείται η προσωρινή, έως ότου το δίκτυο ζητήσει εκ νέου την πραγματική ταυτότητα. Έτσι όταν ο χρήστης αυτός ανιχνευθεί θα εμφανιστεί η προσωρινή ταυτότητα που χρησιμοποιείται εκείνη την στιγμή.

Προστασία αυθεντικοποίησης και σηματοδοσίας: Η αυθεντικοποίηση χρησιμοποιείται για να επιβεβαιωθεί η ταυτότητα του χρήστη στο δίκτυο και είναι βασισμένη στην λειτουργία κρυπτογράφησης.

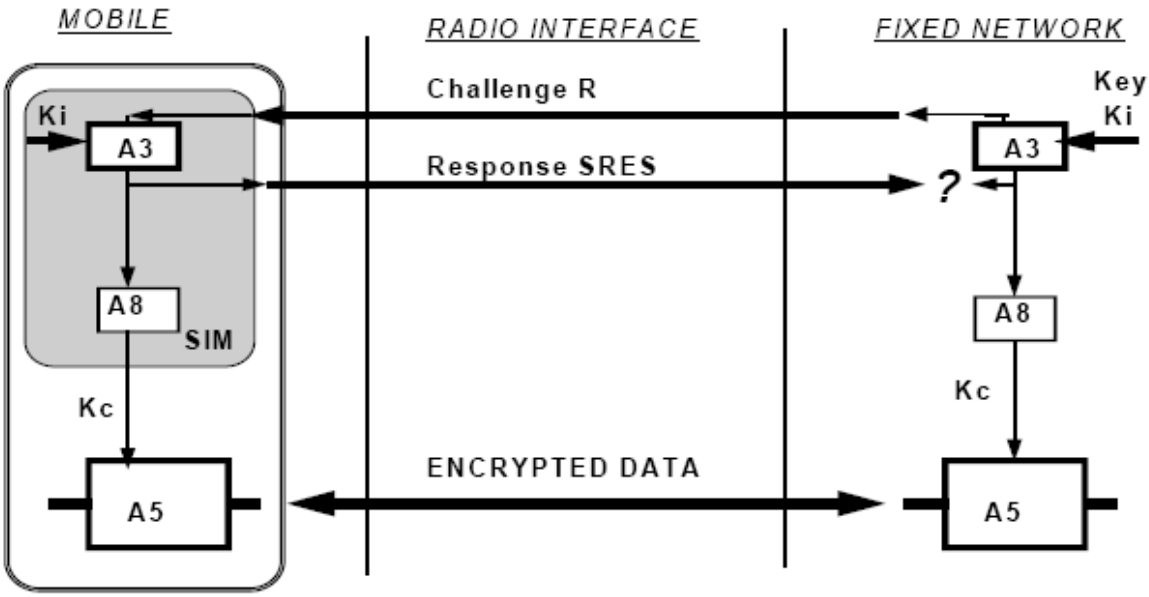
Ο οργανισμός ETSI έχει αναπτύξει τρεις αλγορίθμους ασφάλειας για το GSM: A3, A5 και A8. Ο A3 και ο A8 βρίσκεται μέσα στην κάρτα SIM και στο κέντρο αυθεντικοποίησης (AuC). Ο A5 υπάρχει στην κινητή συσκευή και επιτρέπει την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων που μεταδίδονται στον αέρα μέσω της ραδιοζεύξης.

Η αυθεντικοποίηση πραγματοποιείται με την μέθοδο πρόκλησης και απάντησης. Μία τυχαία πρόκληση π.χ. ένας αριθμός, στέλνεται στο κινητό από το δίκτυο. Το κινητό κρυπτογραφεί την απάντηση χρησιμοποιώντας τον αλγόριθμο A3 και το κλειδί που ορίζεται στο κινητό και την στέλνει πίσω στο δίκτυο. Ο αλγόριθμος μπορεί να παρασταθεί ως εξής: A3(Ki, RAND) = > SRES. Το δίκτυο μπορεί να ελέγξει αν η απάντηση είναι σωστή, λαμβάνοντας υπόψη το κλειδί του κινητού.

Ένας τυχαίος αριθμός R παράγεται από το δίκτυο και στέλνεται στο κινητό. Το κινητό χρησιμοποίει τον αριθμό R ως είσοδο στην συνάρτηση κρυπτογράφησης και χρησιμοποιώντας το μυστικό κλειδί Ki, που είναι αποθηκευμένο στην κάρτα SIM και το οποίο είναι μοναδικό για κάθε κινητό, μετασχηματίζει τον R σε μία απάντηση SRES η οποία στέλνεται πίσω στο δίκτυο. Το δίκτυο μπορεί να ελέγξει αν το κινητό έχει πραγματικά το μυστικό κλειδί εκτελώντας την ίδια διαδικασία και συγκρίνοντας το αποτέλεσμα με την απάντηση που έλαβε από το κινητό. Στην συνέχεια και το κινητό αλλά και το δίκτυο χρησιμοποιούν τον αλγόριθμο A8 για να μετασχηματίσουν την απάντηση σε κάποιο μυστικό κλειδί με το οποίο θα χρησιμοποιηθεί για την κρυπτογράφηση των δεδομένων και της σηματοδοσίας.

Το να διαβαστούν τα δεδομένα που εκπέμπονται στον αέρα από κάποιον τρίτο, δεν αποκαλύπτεται καμία χρήσιμη πληροφορία, δεδομένου ότι την επόμενη φορά θα χρησιμοποιηθεί μία νέα τυχαία πρόκληση. Έτσι παρέχεται μυστικότητα στην επικοινωνία μεταξύ των συσκευών και του δικτύου.

Η διαδικασία μπορεί να παρασταθεί γραφικά ως εξής:



2.1 Διαδικασία αυθεντικοποίησης του κινητού σταθμού από το δίκτυο.

2.2 Άλλοι μηχανισμοί ασφάλειας GSM

2.2.1 Κάρτα SIM

Μία προσωπική κάρτα SIM, στην οποία υπάρχει τσιπ, μπορεί να είναι ένα σταθερό εγκατεστημένο τσιπ (plug-in SIM) ή μία ανταλλάξιμη κάρτα SIM. Η κάρτα SIM είναι ένα ασφαλές περιβάλλον, βασισμένο σε μικροεπεξεργαστή, που εφαρμόζεται σε μία μικρή κάρτα σε μέγεθος πιστωτικής, στην οποία υπάρχει και σταθερή μνήμη.

Δύο τύποι καρτών SIM χρησιμοποιούνται στο GSM, η ID-1 και η plug-in κάρτα. Υπάρχουν επίσης και τρεις τύποι μνήμης, η ROM, η RAM, και η EEPROM. Η ROM περιέχει το λειτουργικό σύστημα, τις εφαρμογές, και τους αλγορίθμους ασφάλειας A3 και A8, τα οποία παρέχουν σημαντικές λειτουργίες για την αυθεντικοποίηση και την κρυπτογράφηση των δεδομένων του χρήστη, βασισμένες στην ταυτότητα IMSI των συνδρομητών και τα μυστικά κλειδιά τους. Η RAM χρησιμοποιείται στην αποθήκευση των μεταδιδόμενων δεδομένων και στην εκτέλεση των διαφόρων λειτουργιών. Και τέλος, η EEPROM περιέχει τα στοιχεία προσδιορισμού των συνδρομητών π.χ. IMSI, PIN κ.τ.λ, τον αριθμό κλήσης IMSI και MSISDN, τα κλειδιά Ki, τις πληροφορίες που είναι σχετικές με το δίκτυο (TMSI, LAI), και ο αριθμός της συσκευής IMEI.

Τα χαρακτηριστικά γνωρίσματα ασφάλειας που υποστηρίζονται από την κάρτα SIM είναι η επιβεβαίωση της ταυτότητας των συνδρομητών στο δίκτυο, η ακεραιότητα των δεδομένων που εκπέμπονται στον αέρα και των όρων πρόσβασης αρχείων. Η SIM μπορεί να υποστηρίξει πέντε όρους πρόσβασης. Ένας από τους όρους πρόσβασης είναι ο αριθμός PIN που χρησιμοποιείται για να ελέγξει την πρόσβαση των χρηστών στην SIM. Εάν ο συνδρομητής πληκτρολογήσει τρεις φορές λάθος κωδικό, τότε η SIM μπλοκάρει.

2.2.2 Προσδιοριστικά του GSM

Σε αυτό το κομμάτι αναφέρονται τα στοιχεία που εξασφαλίζουν ασφάλεια στο δίκτυο GSM.

- **K_i** - Κλειδί επικύρωσης συνδρομητών - Subscriber Authentication Key
- **IMSI** - Διεθνής ταυτότητα συνδρομητών κινητής - International Mobile Subscriber Identity
- **TMSI** - Προσωρινή ταυτότητα συνδρομητών κινητής - Temporary Mobile Subscriber Identity
- **IMEI** - Διεθνής αριθμός κινητού εξοπλισμού - International Mobile Station Equipment
- **MSRN** - Αριθμός περιαγωγής κινητού εξοπλισμού - Mobile Station Roaming Number
- **MSISDN** – Διεθνής υπηρεσία ψηφιακής δικτύωσης κινητού εξοπλισμού - Mobile Station International Service Digital Network
- **PIN** – Προσωπικός αριθμός ταυτότητας προστασίας της SIM - Personal Identity Number
- **LAI** - Ταυτότητα περιοχής θέσης - location area identity
- **Κλειδί συνόδου kc** = A8 (Ki, RAND) - A8: Αλγόριθμος παραγωγής κλειδιού, RAND: Τυχαία πρόκληση μήκους 128 bit, που παράγεται από τον κατάλογο εγχώριας θέσης (HLR).

Το κλειδί αυθεντικοποίησης συνδρομητών Ki, είναι ένα κλειδί μήκους 128 bit, που χρησιμοποιείται για την επιβεβαίωση της ταυτότητας του συνδρομητή στο δίκτυο. Το κλειδί αποθηκεύεται στην κάρτα SIM του συνδρομητή και στον κατάλογο εγχώριων συνδρομητών (HLR) του δικτύου που ανήκει ο συνδρομητής. Ο συνδρομητής κατά την καταχώρηση του στο δίκτυο, λαμβάνει απ' αυτό ένα μοναδικό προσδιοριστικό ταυτότητας IMSI (διεθνής ταυτότητα συνδρομητών κινητής). Αυτό το IMSI αποθηκεύεται στην SIM. Ένα κινητό τηλέφωνο μπορεί μόνο να χρησιμοποιηθεί, εάν υπάρχει μία κάρτα SIM μαζί με έναν έγκυρο αριθμό IMSI, μέσα στην συσκευή. Αυτός είναι ο μόνος τρόπος να τιμολογηθεί σωστά ο κατάλληλος συνδρομητής.

Το IMSI αποτελείται από διάφορα μέρη:

- Κωδικός χωράς κινητής - mobile country code (MCC) που αποτελείται από 3 δεκαδικά ψηφία, διεθνώς τυποποιημένα.

- Κωδικός δικτύων κινητής - mobile Network code (MKM) που αποτελείται από δύο δεκαδικά ψηφία και προσδιορίζουν μοναδικά τα διάφορα δίκτυα κινητής μέσα σε μία χώρα.

- Αριθμός αναγνώρισης συνδρομητών κινητής - Mobile Subscriber Identification Number (MSIN) που αποτελείται από δέκα δεκαδικά ψηφία μέγιστο και είναι ο αριθμός αναγνώρισης του συνδρομητή στο οικείο δίκτυό του.

- Ο πραγματικός αριθμός τηλεφώνου ενός κινητού είναι το MSISDN (Mobile Subscriber ISDN Number - Αριθμός ISDN συνδρομητών κινητής).

- Το VLR είναι αρμόδιο για την τρέχουσα θέση ενός συνδρομητή και ορίζει αριθμούς TMSI (προσωρινή ταυτότητα συνδρομητών κινητής) σε αυτούς. Ο αριθμός TMSI έχει τοπική σημασία και χρησιμοποιείται μόνο μέσα στην περιοχή δικαιοδοσίας του VLR. Επίσης, αντικαθιστά τον αριθμό IMSI, ο οποίος καθορίζει την ταυτότητα του κινητού. Με αυτόν τον τρόπο κανένας δεν μπορεί να προσδιορίσει την πραγματική ταυτότητα του χρήστη κρυφακούγοντας το κανάλι επικοινωνίας, δεδομένου ότι αυτός ο αριθμός TMSI ορίζεται μόνο κατά τη διάρκεια της παρουσίας του κινητού στην περιοχή ενός VLR και μπορεί ακόμη και να αλλάξει κατά τη διάρκεια αυτής της περιόδου (ID hopping).

  Το κινητό αποθηκεύει τον TMSI στην κάρτα SIM. Επίσης, ο TMSI αποθηκεύεται, από την πλευρά του δικτύου, μόνο στο VLR της περιοχής και όχι στο οικείο HLR του συνδρομητή και μπορεί να έχει μήκος μέχρι και 32 bit. Η ένωση μεταξύ του IMSI και του TMSI αποθηκεύεται στο VLR.

- Το MSRN (Mobile station Roaming Number - αριθμός περιαγωγής κινητού) είναι ένας προσωρινός αριθμός ISDN, που εξαρτάται από την τοποθεσία και ορίζεται από το τοπικό VLR της περιοχής που βρίσκεται το κινητό.

- Το IMEI (International Mobile Station Equipment Identity - διεθνής ταυτότητα εξοπλισμού κινητής) προσδιορίζει μοναδικά τις συσκευές κινητής σε διεθνές επίπεδο. Είναι ένα είδος αύξοντος αριθμού. Ο αριθμός IMEI διατίθεται από τον κατασκευαστή των συσκευών και αποθηκεύεται στο EIR του δικτύου. Με τη βοήθεια του IMEI το δίκτυο αναγνωρίζει τις ξεπερασμένες, κλεμμένες, ή μη λειτουργικές συσκευές κινητής τηλεφωνίας.

- Κάθε περιοχή θέσης (Location Area - LA) έχει δικό της αναγνωριστικό, που λέγεται LAI (Location Area ID - ταυτότητα περιοχής θέσης). Το LAI έχει ιεραρχημένη δομή και είναι μοναδικό σε παγκόσμιο επίπεδο.

ΚΕΦΑΛΑΙΟ 3 - Επιθέσεις GSM

3.1 Επιθέσεις δικτύων GSM

Οι ακόλουθες επιθέσεις λαμβάνουν χώρα στο δίκτυο GSM και αναφέρονται και στις κλήσεις φωνής και στα μηνύματα κειμένου.

3.1.1 Υποκρισία διαγραφής - Deregistration Spoofing

Μία επίθεση που απαιτεί από την μεριά του επιτιθέμενου να έχει στην κατοχή του μία τροποποιημένη κινητή συσκευή, η οποία εκμεταλλεύεται την αδυναμία που έχει το δίκτυο να μην μπορεί να αυθεντικοποιήσει τα μηνύματα που λαμβάνει μέσω των ηλεκτρομαγνητικών κυμάτων. Ο εισβολέας στέλνει μία αίτηση διαγραφής (αποσύνδεση IMSI) προς στο δίκτυο, υποκρινόμενος έναν νόμιμο χρήστη ο οποίος εκείνη την στιγμή βρίσκεται συνδεδεμένος. Το δίκτυο διαγράφει τον πραγματικο χρήστη από την επισκεπτόμενη περιοχή θέσης και καθοδηγεί και το HLR για να κάνει το ίδιο. Το αποτέλεσμα είναι να μην μπορεί ο νόμιμος χρήστης να έχει πρόσβαση στις υπηρεσίες του δικτύου.

3.1.2 Υποκρισία αναπροσαρμογής θέσης - Location Update Spoofing

Αυτή η επίθεση μοιάζει με την προηγούμενη και βασίζεται στην ίδια αδυναμία του δικτύου, μόνο που εδώ ο επιτιθέμενος στέλνει αίτηση αναπροσαρμογής θέσης από διαφορετική περιοχή από αυτήν στην οποία βρίσκεται ο πραγματικός χρήστης. Στην συνέχεια το δίκτυο καταχωρεί τα δεδομένα του εισβολέα στον κατάλογο της νέας θέσης και ο πραγματικός χρήστης στην παλιά θέση δεν μπορεί πλέον να δεχθεί τις υπηρεσίες του δικτύου.

3.1.3 Εγγραφή σε ένα ψεύτικο BTS - Camping on a false BTS

Στο δίκτυο GSM γίνεται αυθεντικοποίηση της τηλεφωνικής συσκευής στο BTS του δικτύου, αλλά όχι το ανάποδο. Αυτό σημαίνει ότι ένας επιτιθέμενος μπορεί να προσποιηθεί σε ένα κινητό τηλέφωνο ότι είναι το BTS του δικτύου της περιοχής. Έτσι ο χρήστης, όταν καταχωρηθεί στον ψεύτικο σταθμό βάσης, μένει εκτός υπηρεσίας από το δίκτυο του παρόχου. Υπάρχει και μία άλλη παραλλαγή ως προς το αποτέλεσμα αυτής της επίθεσης: Ο επιτιθέμενος μπορεί να παίξει το ρόλο του ενδιάμεσου και να ξεγελάσει και το BTS υποκρινόμενος ότι είναι ο χρήστης, δηλαδή μπορεί να αναπαράγει την ασύρματη κίνηση από και προς το επιτιθέμενο κινητό, υποκλέπτοντας τα δεδομένα.

3.1.4 Πρόσβαση στο δίκτυο - Accessing the Signalling Network

Οι μεταδόσεις δεδομένων κρυπτογραφούνται μόνο μεταξύ κινητού τηλεφώνου και BTS. Μετά από το BTS η κυκλοφορία μεταδίδεται σε μορφή απλού κειμένου μέσα στο δίκτυο του παρόχου. Αυτό δίνει νέες δυνατότητες.

Εάν ο επιτιθέμενος έχει πρόσβαση στο δίκτυο του παρόχου, τότε είναι σε θέση να ακούσει όλα όσα μεταδίδονται, συμπεριλαμβανομένου και των κλήσεων που γίνονται εκείνη την ώρα. Το δίκτυο SS7, που χρησιμοποιείται στο δίκτυο GSM, δεν προστατεύει εάν ο επιτιθέμενος έχει άμεση πρόσβαση σ' αυτό. Σε μία άλλη περίπτωση, ο επιτιθέμενος θα μπορούσε να επιτεθεί στο HLR ενός συγκεκριμένου τοπικά δικτύου. Εάν ο επιτιθέμενος έχει πρόσβαση σε ένα συγκεκριμένο HLR του δικτύου, θα μπορούσε να ανακτήσει τα μυστικά κλειδιά (Ki) όλων των συνδρομητών που είναι στην δικαιοδοσία του.

Η πρόσβαση στο δίκτυο δεν είναι πολύ δύσκολη. Αν και τα BTS συνδέονται συνήθως με το BSC μέσω ενός καλωδίου, μερικά από αυτά μπορεί να συνδέονται με το BSC μέσω μικροκυματικής ζεύξης ή ακόμα και μέσω δορυφορικής σύνδεσης. Αυτή η σύνδεση μπορεί να είναι σχετικά εύκολη στην πρόσβαση, έχοντας το σωστό είδος εξοπλισμού. Το μεγαλύτερο μέρος αυτού του εξοπλισμού, που κυκλοφορεί στο εμπόριο, φαίνεται ότι εκμεταλλεύεται αυτήν την ιδιαίτερη ευπάθεια. Αυτό μπορεί να είναι μία πραγματικά μεγάλη απειλή και μία τέτοια επίθεση θα μπορούσε να μην ανιχνευθεί για μεγάλο χρονικό διάστημα εάν εφαρμοστεί προσεκτικά.

Η δυνατότητα να διαρρέουν τα δεδομένα που διαβιβάζονται μεταξύ του BTS και του BSC, θα επέτρεπε στον επιτιθέμενο είτε να παρακολουθήσει την κλήση με το να κρυφακούσει το κανάλι σε όλη την διάρκεια της, είτε θα μπορούσε να ανακτήσει το κλειδί συνόδου και παρακολουθώντας στην συνέχεια το ασύρματο κανάλι, να ήταν σε θέση να αποκρυπτογραφήσει την κλήση σε πραγματικό χρόνο. Εφόσον τώρα πια γνωρίζει το κλειδί, η αποκρυπτογράφηση σε πραγματικό χρόνο δεν είναι πια πρόβλημα.

3.1.5 Ανάκτηση του κλειδιού από την SIM - Retrieving the Key from the SIM

Η ασφάλεια ολόκληρου του προτύπου ασφάλειας GSM είναι βασισμένη στο μυστικό κλειδί Ki. Εάν αυτό το κλειδί λειτουργεί σωστά, τότε και ολόκληρος ο μηχανισμός ασφαλείας λειτουργεί και αυτός σωστά.

Μόλις ο επιτιθέμενος είναι σε θέση να ανακτήσει το μυστικό κλειδί Ki, μπορεί όχι μόνο να ακούσει τις κλήσεις των συνδρομητών, αλλά και να πραγματοποιήσει κλήσεις που τιμολογούνται στον αρχικό συνδρομητή, επειδή τώρα είναι σε θέση να τον αντιπροσωπεύσει πλήρως. Το δίκτυο GSM όμως έχει τον τρόπο να το αποτρέπει αυτό: Εάν δύο τηλέφωνα με την ίδια ταυτότητα εξυπηρετούνται συγχρόνως, το δίκτυο GSM παρατηρεί ότι το "ίδιο" τηλέφωνο είναι σε δύο διαφορετικές θέσεις συγχρόνως και γι' αυτό κλείνει τον λογαριασμό, αποτρέποντας κατά συνέπεια τον επιτιθέμενο και το νόμιμο συνδρομητή από την πραγματοποίηση κλήσεων.

Η Smartcard Developer Association και η ομάδα έρευνας σε θέματα ασφάλειας του ISAAC, ανακάλυψαν ένα αδύναμο κρίκο στον αλγόριθμο COMP128, ο οποίος

χρησιμοποιείται για την αυθεντικοποίηση στο GSM και αυτή η αδυναμία μπορεί να επιτρέψει σε κάποιον να ανακτήσει το μυστικό κλειδί Ki από μία κάρτα SIM. Η επίθεση έγινε σε μία κάρτα SIM που υπήρχε φυσική πρόσβαση, αλλά η ίδια επίθεση ισχύει και όταν γίνεται επίθεση με ασύρματο τρόπο. Η επίθεση είναι βασισμένη στο μοντέλο πρόκλησης και επιλογής, επειδή ο αλγόριθμος COMP128 σπάει κατά τέτοιο τρόπο, ώστε αποκαλύπτει τις πληροφορίες για το Ki όταν δοθούν κατάλληλες τιμές RAND ως παράμετροι στον αλγόριθμο A8.

Στην κάρτα SIM παρέχεται πρόσβαση μέσω ενός αναγνώστη έξυπνων καρτών, ο οποίος είναι συνδεδεμένος με έναν ηλεκτρονικό υπολογιστή. Ο υπολογιστής αυτός κάνει περίπου 150.000 προκλήσεις στη SIM και αυτή με την σειρά της παράγει το SRES και το κλειδί συνόδου kc, βασισμένο στην πρόκληση και το μυστικό κλειδί της κάρτας. Το μυστικό κλειδί θα μπορούσε να συναχθεί από τις απαντήσεις SRES (η 32μπιτη υπογεγραμμένη απάντηση που παράγεται από τον κινητό τηλέφωνο και από το MSSC), μέσω της χρήσης της διαφορικής μικροανάλυσης. Ο αναγνώστης έξυπνων καρτών που χρησιμοποιείται στην εφαρμογή της επίθεσης θα μπορούσε να κάνει 6,25 ερωτήσεις ανά δευτερόλεπτο στην κάρτα SIM. Έτσι η επίθεση απαιτεί περίπου οκτώ ώρες για να γίνει. Επίσης, πρέπει να αναλυθούν τα αποτελέσματα, αλλά αυτό είναι πολύ σύντομη δουλειά, έναντι της πραγματικής επίθεσης. Κατά συνέπεια ο επιτιθέμενος πρέπει να έχει φυσική πρόσβαση στην κάρτα SIM για τουλάχιστον οκτώ ώρες.

3.1.6 Ασύρματη ανάκτηση του κλειδιού από την SIM - Retrieving the Key from the SIM over the Air

Οι ερευνητές της SDA και της ISAAC είναι βέβαιοι ότι η ίδια επίθεση κλωνοποίησης της κάρτας SIM μπορεί επίσης να γίνει με ασύρματο τρόπο. Δυστυχώς οι ερευνητές δεν μπορούν να επιβεβαιώσουν τις υποψίες τους επειδή ο εξοπλισμός που απαιτείται θεωρείται παράνομος στις Ηνωμένες Πολιτείες.

Η ασύρματη επίθεση είναι βασισμένη στο γεγονός ότι το κινητό τηλέφωνο πρέπει να ανταποκριθεί σε κάθε πρόκληση που γίνεται από το δίκτυο GSM. Εάν το σήμα του νόμιμου BTS είναι ανίσχυρο μπροστά στο σήμα του ψεύτικου, ο επιτιθέμενος μπορεί να βομβαρδίσει το κινητό με προκλήσεις και να αναδημιουργήσει το μυστικό κλειδί από τις απαντήσεις που θα λάβει. Και πάλι όμως η συσκευή πρέπει να είναι ασύρματα διαθέσιμη στον επιτιθέμενο για ολόκληρο το διάστημα που λαμβάνει χώρα η επίθεση. Δεν είναι γνωστό πόσο χρόνο χρειάζεται η επίθεση αυτή για να βγει εις πέρας. Εκτιμάται ότι ο χρόνος αυτός κυμαίνεται από οκτώ έως δεκατρείς ώρες.

3.1.7 Ανάκτηση του κλειδιού από το AuC (κέντρο αυθεντικοποίησης) - Retrieving the Key from the AuC

Η ίδια επίθεση που χρησιμοποιείται για την ανάκτηση του κλειδιού Ki από μία κάρτα SIM, μπορεί να χρησιμοποιηθεί και για την ανάκτηση του Ki από το AuC. Το AuC πρέπει να απαντήσει στις αιτήσεις που υποβάλλονται από το δίκτυο GSM και να επιστρέψει έγκυρες τριπλέτες που χρησιμοποιούνται για την αυθεντικοποίηση των

κινητών τηλεφώνων. Η διαδικασία είναι βασικά η ίδια με αυτήν που χρησιμοποιείται στα κινητά για να υπάρχει πρόσβαση στην κάρτα SIM. Η διαφορά είναι ότι το AuC είναι πολύ γρηγορότερο στην επεξεργασία των αιτημάτων από μία κάρτα SIM, επειδή πρέπει να επεξεργαστεί πολύ περισσότερα αιτήματα απ' ότι μία κάρτα SIM. Η ασφάλεια του AuC διαδραματίζει ένα σημαντικό ρόλο στο εάν αυτή η επίθεση γίνεται ή όχι.

3.1.8 Σπάσιμο του αλγορίθμου A8 - Cracking the A8 Algorithm

Μία άλλη δυνατότητα είναι κάποιος να είναι σε θέση να σπάσει τον αλγόριθμο παραγωγής κλειδιών A8 και να ανακτήσει το μυστικό κλειδί Ki βασισμένος στην τυχαία πρόκληση, την τιμή RAND, το κλειδί συνόδου kc και την απάντηση με το ελάχιστο κόστος εργασίας. Παραδείγματος χάριν, ο επιτιθέμενος μπορεί να βρει μία τιμή RAND από τον οποίο παράγεται το κλειδί Ki, ως αποτέλεσμα. Και οι τρεις μεταβλητές μπορούν να ληφθούν σχετικά εύκολα. Η μεταβλητή RAND και το SRES στέλνονται ασύρματα σε μορφή απλού κείμενου, ενώ το κλειδί συνόδου kc, μπορεί να συναχθεί σχετικά εύκολα από τα κρυπτογραφημένα πλαίσια και το γνωστό απλό κείμενο, δεδομένου ότι υπάρχει αρκετός χρόνος στην διάθεση του επιτιθέμενου.
Μία αδυναμία όπως αυτή, που αφορά τον αλγόριθμο παραγωγής κλειδιών, θα κατάστρεφε φυσικά ολόκληρο το πρότυπο ασφάλειας του GSM και θα έδινε στην κοινοπραξία της GSM κάτι για να σκεφτεί κατά τον σχεδιασμό των επόμενων αλγορίθμων ασφάλειάς που θα χρησιμοποιήσουν.

3.1.9 Επιθέσεις στον αλγόριθμο A5 - A5 Algorithm Attacks

Ο αλγόριθμος A5 χρησιμοποιείται για την κρυπτογράφηση ροής δεδομένων που μεταδίδονται ασύρματα. Κάθε πλαίσιο που μεταδίδεται μέσω του ρεύματος δεδομένων κρυπτογραφείται χρησιμοποιώντας το κλειδί συνόδου kc και τον αριθμό του πλαισίου που από/κρυπτογραφείται. Το ίδιο kc χρησιμοποιείται και κατά την διάρκεια μίας κλήσης, αλλά ο αριθμός των 22bit πλαισίων αλλάζει κατά τη διάρκεια αυτής της κλήσης, παράγοντας κατά συνέπεια ένα μοναδικό ρεύμα κλειδιών για κάθε πλαίσιο.

Ο αλγόριθμος A5 που χρησιμοποιείται στις ευρωπαϊκές χώρες αποτελείται από τρεις καταχωρητές ανατροφοδότησης γραμμικής μετατόπισης (LSFR), οι οποίοι καταχωρούν και παράγουν ένα bit εξόδου βασισμένοι στην προηγούμενη κατάσταση και σ' ένα πολυώνυμο ανατροφοδότησης. Τα αποτελέσματα των τριών καταχωρητών περνούν από αποκλειστική διάζευξη μεταξύ τους και το αποτέλεσμα αυτής αντιπροσωπεύει ένα ρεύμα δεδομένων κλειδιού. Τα τρία LSFR αρχικοποιούνται με το κλειδί συνόδου Kc και τον αριθμό του πλαισίου.

Παρακάτω περιγράφονται δύο τύποι επιθέσεων ενάντια στον αλγόριθμο A5.

1. Επίθεση brute-force ενάντια στον αλγόριθμο A5 - Brute-Force Attack against A5

Μία επίθεση αυτού του τύπου δεν μπορεί να γίνει σε πραγματικό χρόνο λόγω της πολυπλοκότητας της επίθεσης η οποία είναι της τάξης του 2^{54}. Αυτό απαιτεί πάρα πολύ χρόνο προκειμένου να γίνει εφικτή η υποκλοπή των κλήσεων που γίνονται στο GSM σε πραγματικό χρόνο. Γι' αυτόν λοιπόν πρέπει να καταγραφούν όλα τα πλαίσια που μεταδίδονται ασύρματα μεταξύ του κινητού τηλεφώνου και του σταθμού βάσης και κατόπιν αυτού να πραγματοποιηθεί η εν λόγω επίθεση.

Εάν έχουμε έναν επεξεργαστή της κατηγορίας Pentium III, ο οποίος διαθέτει περίπου 20 εκατομμύρια τρανζίστορ και η εφαρμογή ενός συνόλου LSFR (A5/1) απαιτεί περίπου 2000 τρανζίστορ, θα είχαμε ένα σύνολο 10.000 παράλληλων διεργασιών A5/1 σε έναν επεξεργαστή. Εάν το τσιπ λειτουργεί χρονισμένο στην συχνότητα των 600 MHz και κάθε διεργασία A5 παρήγαγε ένα bit εξόδου σε κάθε κύκλο ρολογιού και θα πρέπει να παραχθούν 100+114+114 = 328 bit, θα ήταν εφικτό να γίνεται δοκιμή 2 εκατομμυρίων κλειδιών ανά δευτερόλεπτο για κάθε μία διεργασία A5/1. Με εύρος κλειδιών της τάξης 2^{54} θα απαιτούσε περίπου 900.000 δευτερόλεπτα ή 250 ώρες. Η επίθεση μπορεί να βελτιστοποιηθεί με το να προχωράει στο επόμενο κλειδί μετά από το πρώτο άκυρο ρεύμα δεδομένων του προηγούμενου. Αυτό θα μείωνε τον απαιτούμενο χρόνο στο ένα τρίτο. Η επίθεση μπορεί επίσης να κατανεμηθεί μεταξύ των πολλών επεξεργαστών, μειώνοντας δραστικά το χρόνο που απαιτείται για να ολοκληρωθεί η επίθεση.

2. ***Επίθεση "διαίρε-και-βασίλευε" ενάντια στον A5 - Divide-and-Conquer Attack against A5***

Η επίθεση "διαίρε-και-βασίλευε" κατορθώνει να μειώσει την πολυπλοκότητα της προηγούμενης επίθεσης από 2^{54} σε 2^{45}, το οποίο είναι μία σχετικά δραματική αλλαγή (2^9=512 φορές γρηγορότερα). Η επίθεση αυτή είναι βασισμένη στην επίθεση γνωστού αρχικού κειμένου.

Ο επιτιθέμενος προσπαθεί να καθορίσει τις αρχικές καταστάσεις των καταχωρητών LSFR από μία γνωστή ακολουθία δεδομένων κλειδιού. Επίσης πρέπει να ξέρει τα 64 συνεχόμενα bit του ρεύματος του κλειδιού, τα οποία μπορούν να ανακτηθούν εάν έχει στην διάθεση του κάποιο κρυπτογραφημένο κείμενο και το αντίστοιχο αρχικό. Τα απαραίτητα 64 bit μπορεί να μην ανακτηθούν πάντα, αλλά συνήθως είναι γνωστά τα 32 έως 48 bit. Ο επιτιθέμενος χρειάζεται μόνο ένα κομμάτι αρχικού κειμένου μήκους 64 bit. Εν ολίγοις η επίθεση αυτή εφαρμόζεται μαντεύοντας το περιεχόμενο των δύο πρώτων καταχωρητών LSFR και έπειτα υπολογίζεται ο τρίτος LSFR από το γνωστό ρεύμα κλειδιού. Αυτή θα ήταν μία επίθεση με πολυπλοκότητα 2^{40}, εάν ο χρονισμός των δύο πρώτων καταχωρητών δεν εξαρτιόταν από τον τρίτο καταχωρητή.

ΚΕΦΑΛΑΙΟ 4 - Επιθέσεις SMS

4.1 Τύποι επιθέσεων στην υπηρεσία SMS

4.1.1 Επίθεση άρνησης εξυπηρέτησης - Denial of Service Attack (DOS)

Όλες οι συσκευές που συνδέονται στο Διαδίκτυο έχουν δημόσιες διευθύνσεις IP. Αυτές οι συσκευές είναι ευαίσθητες σε επιθέσεις άρνησης εξυπηρέτησης (DOS). Μία επίθεση DOS έχει σαν αποτέλεσμα την ελάχιστη εξυπηρέτηση ή και την απενεργοποίηση μίας υπηρεσίας, η οποία έχει γονατίσει κάτω από υπερβολικό φορτίο δικτύου.

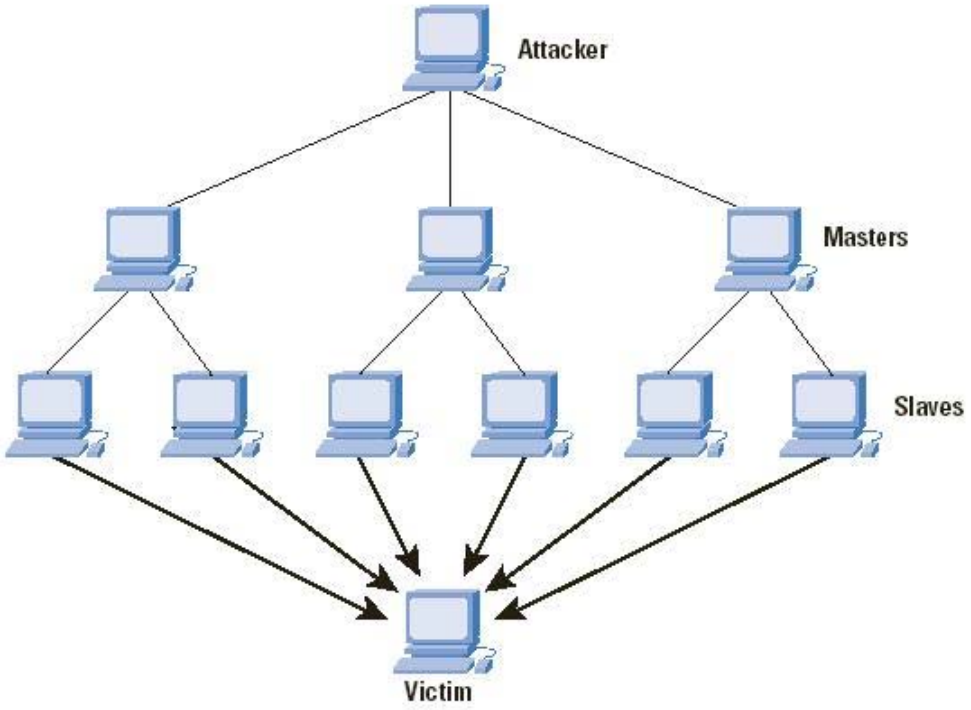
Οι επιθέσεις DOS κέρδισαν την κακή φήμη στις αρχές του 2000 όταν μία ομάδα χάκερ κατόρθωσε να διακόψει την υπηρεσία της Yahoo για περίπου δυόμιση ώρες. Σε εκείνη την περίπτωση, οι χάκερ χρησιμοποίησαν μία κατανεμημένη επίθεση (DDOS), μία παραλλαγή της απλής DOS, στην οποία ο επιτιθέμενος μολύνει εκατομμύρια μεμονωμένους υπολογιστές με έναν ιό και έπειτα, βάσει συντονισμού, κάθε μολυσμένη μηχανή αρχίζει να ανοίγει γρήγορα συνδέσεις σε έναν συγκεκριμένο κεντρικό υπολογιστή ταυτόχρονα. Στην περίπτωση της Yahoo, οι εκατομμύρια μολυσμένοι υπολογιστές σε όλο τον κόσμο άρχισαν να επιτίθενται ταυτόχρονα στους κεντρικούς υπολογιστές της Yahoo, κατακλύζοντας το δίκτυο της με δεδομένα που έφταναν σε ταχύτητα το 1Gbps. Η επίθεση DOS υπάρχει σε ποικίλες μορφές, έχοντας ως στόχος διαφόρων τύπων υπηρεσίες.

Υπάρχουν τρεις βασικοί τύποι επιθέσεων:

1.Περιορίσμενη ζήτηση, ή μη ανανεώσιμοι πόροι
2. Καταστροφή ή αλλαγή των πληροφοριών
3. Φυσική καταστροφή ή αλλαγή των τμημάτων του δικτύου

Τα περισσότερα SMSC βρίσκονται πίσω από κάποιο τείχος προστασίας, αλλά αυτό αυξάνει πολύ τη καθυστέρηση του δικτύου και την μετάδοση των μηνυμάτων. Γι' αυτόν τον λόγο οι πάροχοι βάζουν SMSC που εκτίθεται ευκολότερα στις επιθέσεις DOS για χάριν της υψηλής απόδοσης. Εντούτοις, ένα ανάρμοστα διαμορφωμένο τείχος προστασίας, δεν μπορεί να προστατεύσει ένα SMSC από τις επιθέσεις DOS και ειδικά όταν πρόκειται για τη κατανεμημένη επίθεση DOS (DDOS), η οποία είναι και δυσκολότερη στην αντιμετώπιση της.

Όταν ένας διακομιστής παρέχει μία λειτουργία εξαιτίας της οποίας χρειάζεται να συνδεθούν πολλές μηχανές πάνω σ' αυτόν (όπως γίνεται με τους περισσότερους διακομιστές δικτύου και ηλεκτρονικού ταχυδρομείου), ο διαχειριστής δεν μπορεί να χρησιμοποιήσει ένα τείχος προστασίας για να αποκλείσει υπολογιστές που είχαν επιτεθεί στο παρελθόν, χωρίς να πειράξει την βάση δεδομένων των χρηστών του. Επιπλέον, είναι σχεδόν αδύνατο να βρεθεί ο επιτιθέμενος που σχεδίασε αρχικά την επίθεση και αυτόν που διένειμε τον ιό. Το παρακάτω σχήμα δείχνει πως γίνεται μία κατανεμημένη επίθεση DOS.



4.1 Κατανεμημένη επίθεση DOS - DDOS

Επειδή τα μηνύματα κειμένου και οι κλήσεις των κινητών τηλεφώνων στηρίζονται στον ίδιο περιορισμένο πόρο, δηλαδή τα κανάλια ελέγχου, είναι εφικτή κάποια επίθεση σε αυτό το σύστημα. Εάν αρχίσουν να στέλνονται πολλά μηνύματα κειμένου, έτσι ώστε να μην υπάρχουν άλλα διαθέσιμα κανάλια ελέγχου, οι κλήσεις θα αρχίσουν να απορρίπτονται. Για να έχει μεγαλύτερη επιτυχία, η επίθεση θα πρέπει να στοχεύσει στα τηλέφωνα που βρίσκονται μέσα σε μία ορισμένη γεωγραφική περιοχή και αυτό μπορεί να γίνει με χρήση των δημόσιων βάσεων δεδομένων και των αναζητήσεων της Google. Στην πραγματικότητα, θα έπαιρνε λίγο περισσότερο χρόνο από ένα ενσύρματο μόντεμ που αρνείται να εξυπηρετήσει στις μεγάλες μητροπολιτικές περιοχές των ΗΠΑ. Παραδείγματος χάριν, μία πόλη στο μέγεθος της Ουάσιγκτον, θα μπορούσε να μείνει χωρίς υπηρεσίες μετά από μία επίθεση DoS με ένα εύρος ζώνης περίπου 2,8 Mbps.

Οι επιθέσεις DOS ενάντια στις κινητές συσκευές, θα μπορούσαν να εκμεταλλευτούν το γεγονός ότι αυτές οι συσκευές τροφοδοτούνται από μπαταρίες. Σε αυτήν την περίπτωση, ο στόχος της επίθεσης είναι να αδειάσουν γρήγορα οι μπαταρίες της συσκευής που βρίσκεται υπό επίθεση. Οι επιτυχείς επιθέσεις θα διακόψουν ή θα περιορίσουν εντυπωσιακά το χρόνο λειτουργίας του στόχου.

Ένας τρόπος να πραγματοποιηθεί μία επίθεση DoS μέσω των μηνυμάτων SMS είναι με τη βοήθεια των σιωπηλών μηνυμάτων SMS. Αυτό είναι ένα SMS το οποίο κανένα ηχητικό ή οπτικό σημάδι δεν φανερώνει την παρουσία του. Αυτό εμφανίζεται όταν το κινητό πρέπει να αναγνωρίσει την λήψη του μηνύματος αλλά μπορεί να απορρίψει το περιεχόμενό του.

Μία μαζική και συνεχής αποστολή από σιωπηλά μηνύματα και με τη βοήθεια των μαζικών παρόχων SMS, θα αποτελέσει μία αόρατη επίθεση DoS σε ένα κινητό τηλέφωνο. Με μία τέτοια επίθεση, το κινητό τηλέφωνο μπορεί να παρουσιάσει δυσλειτουργία ή το SMSC να τεθεί εκτός λειτουργίας. Ο πρώτος τρόπος για να γίνει μία τέτοια επίθεση είναι με τον χειρισμό του σχεδίου κωδικοποίησης των δεδομένων. Εάν η κωδικοποίηση των δεδομένων τεθεί στο 192, ενεργοποιείται το αναγνωριστικό ένδειξης αναμονής μηνυμάτων, το οποίο μεταφράζει τα δεδομένα σε απορριπτέο μήνυμα. Με τα bits 4 έως 7, να τεθούν σε 1100, το κινητό μπορεί να απορρίψει το περιεχόμενο του μηνύματος.

Μία δεύτερη προσέγγιση είναι να τροποποιηθεί ο προγραμματισμένος χρόνος παράδοσης ή η έγκυρη περίοδος όπου θα σταλεί ένα SMS μέσω WAP. Αλλάζοντας την ημερομηνία του μηνύματος, θέτοντας την σε μία που έχει παρέλθει, ένα μήνυμα μπορεί να ληφθεί από ένα τηλέφωνο χωρίς ο παραλήπτης να πληροφορηθεί την παραλαβή.

4.1.2 Επίθεση διακοπής υπηρεσιών - Service Interruption Attack

Μία Service Interruption επίθεση είναι παρόμοια μίας Denial of Service επίθεσης, αλλά διαφέρει στο ότι χρησιμοποιεί ένα μικρότερο αριθμό λανθασμένων μηνυμάτων στην προσπάθεια να πάρει τον έλεγχο και να αποδιοργανώσει μία υπηρεσία. Οι επιθέσεις αυτές είναι πιο ήπιες από μία DOS επίθεση και συνήθως εξαρτώνται από την εκ βαθέων έρευνα και τις εκ των έσω πληροφορίες για τον καθορισμό του πώς να διαμορφώσουν ένα μήνυμα ώστε να προκαλέσουν ένα χαώδες αποτέλεσμα στην εφαρμογή-στόχο. Η πλειοψηφία αυτών των επιθέσεων εξαπολύονται από άτομα που είχαν ανάμιξη στο σχεδιασμό και την παραγωγή του συστήματος. Σχεδιάζουν την επίθεση βασιζόμενοι στα τρωτά σημεία που βρίσκουν στον κώδικα. Σε κάποιες περιπτώσεις τα άτομα αυτά σκοπίμως τοποθετούν αυτές τις ατέλειες.

Αυτά τα είδη των επιθέσεων πραγματοποιούνται επίσης κατά τη διάρκεια των σταδίων δοκιμής ενός ‘ανοιχτού’ λογισμικού, μολονότι η πλειοψηφία των συνεισφερόντων και των ερευνητών της ανοιχτής κοινότητας, βοηθούν στην εξάλειψη της συντριπτικής πλειοψηφίας αυτών των ελαττωμάτων, πριν το λογισμικό πάρει την τελική του μορφή. Τα εμπορικά λογισμικά δοκιμάζονται συνήθως βιαστικά σε μικρό βαθμό και ο αριθμός των ατόμων που παίρνουν μέρος στην επανεξέταση του κώδικα είναι συνήθως πολύ μικρότερος από τον αναγκαίο, κάτι το οποίο στατιστικά οδηγεί σε περισσότερες ενδεχόμενες ‘τρύπες’.

Το γεγονός ότι ο κώδικας αποκρύπτεται από το κοινό υποδηλώνει ότι τέτοια τρωτά σημεία μπορούν να παραμείνουν μη ανιχνεύσιμα κατά τη διάρκεια του κύκλου ζωής του προϊόντος. Αυτά τα τρωτά σημεία ανακαλύπτονται μερικές φορές κατά λάθος, όπως στην περίπτωση του περιβόητου “WinNuke” το 1997. Το τρωτό σημείο WinNuke ήταν ένα σημείο στο TCP/IP πρωτόκολλο της αρχικής έκδοσης των Windows 95, το οποίο όταν λάμβανε μία ειδικά διαμορφωμένη TCP επικεφαλίδα, θα μπορούσε να προκαλέσει τη διάλυση όλου του συστήματος. Και ενώ το ελαττωματικό σημείο θα διορθωνόταν σχετικά γρήγορα, οποιοσδήποτε υπολογιστής συνδεδεμένος στο Internet και μη προστατευμένος από ένα firewall, θα κατέρρεε μέσα σε κάποια λεπτά σύνδεσής του, εξαιτίας του μεγάλου αριθμού των hackers που θα πλημμύριζαν το Internet με αυτά τα καταστρεπτικά πακέτα.

Δεν είναι όλες οι Service Interruption επιθέσεις σχεδιασμένες να διαλύουν τις μηχανές που στοχεύουν. Συνήθως είναι απλά δεδομένα που εισάγονται στους υπολογιστές και για τα οποία ο επιτιθέμενος γνωρίζει ότι θα έχουν διάφορες μορφές χαοτικών και απρόβλεπτων αποτελεσμάτων στο στόχο. Υπάρχουν αμέτρητες αναφορές σφαλμάτων σε συστήματα τα οποία, μετά την σκόπιμη ή την χωρίς πρόθεση παραλαβή λανθασμένων εισαγόμενων δεδομένων, χρειάστηκε να επανεκκινήσουν ή να αναδιαταχθούν.

Οι Service Interruption επιθέσεις είναι δύσκολο να προβλεφθούν σε επίπεδο μακροεντολών. Εξαρτώνται από την συγκεκριμένη εφαρμογή του συστήματος. Ένα καλά σχεδιασμένο και εφαρμοζόμενο SMSC θα μπορούσε να αναπτυχθεί ώστε να είναι ουσιαστικά απρόσβλητο από αυτό το είδος επίθεσης. Είναι σημαντικό να σημειωθεί ότι το SMPP και άλλα πρωτόκολλα με τα οποία το SMSC επικοινωνεί είναι αρκετά απλά και ‘κλειστά’ ώστε να μην έχουν ‘τρύπες’ οι οποίες θα μπορούσαν να προκαλέσουν τη διακοπή του συστήματος.

4.1.3 Επίθεση πειρατείας της υπηρεσίας - Service Hijacking Attack

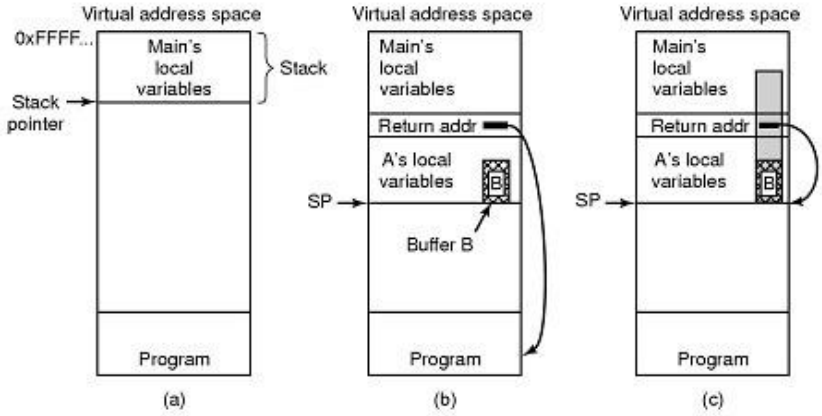
Η πειρατεία υπηρεσιών εμφανίζεται όταν ένα άτομο που δεν έχει εξουσιοδότηση, παίρνει τον έλεγχο μίας διαδικασίας σε μία μηχανή, ή στη χειρότερη περίπτωση, όταν παίρνει τον έλεγχο ολόκληρης της μηχανής. Η πιο επικίνδυνη πτυχή αυτής της μορφής επίθεσης είναι ότι μόλις ένας επιτιθέμενος πάρει τον έλεγχο, τότε μπορεί σκόπιμα να τροποποιήσει τα δεδομένα για να επιτύχει έναν δικό του στόχο, ο οποίος συχνά έρχεται σε αντίθεση με τους στόχους του χειριστή της υπηρεσίας.

Η πειρατεία υπηρεσιών μπορεί να προκαλέσει την αλλαγή ή την απώλεια δεδομένων του συστήματος. Αυτό μπορεί να περιλαμβάνει τη απόκτηση μηνυμάτων, την αλλαγή του κειμένου και την φραγή των εξερχόμενων πακέτων από το σύστημα. Κατά την επίθεση αυτή, ο επιτιθέμενος προσπαθεί να διαβάσει ή να τροποποιήσει τα δεδομένα του συνδεδεμένου δικτύου SMS ή να τροποποιήσει το λογισμικό στην επιτιθέμενη μηχανή για να επιτρέψει την εύκολη πρόσβαση σε πιθανή επίθεση στο μέλλον ή ακόμα και να διαβιβάσει ορισμένα δεδομένα σε έναν εξωτερικό ακροατή. Μπορεί επίσης να υπονομεύσει το ίδιο το σύστημα, κάνοντας το να αποδίδει λιγότερο, ή να καταστρέψει τις δυνατότητες του ώστε να το θέσει εκτός λειτουργίας.

4.1.4 Επίθεση υπερχείλισης Buffer - Buffer Overflow Attack

Μπορεί να είναι μέρος της προηγούμενης επίθεσης, που έχει ως τελικό στόχο τον έλεγχο της μηχανής. Η υπερχείλιση του buffer εμφανίζεται λόγω των ανεξέλεγκτων ορίων στη διαθέσιμη μνήμη και αυτό μπορεί να επιτρέψει στις κακόβουλες εντολές να εκτελεστούν στη επιτιθέμενη μηχανή. Εάν δεν υπάρχει κανένας έλεγχος πάνω σ’ αυτό, ένας επιτιθέμενος μπορεί να στείλει ένα μήνυμα πολύ μεγαλύτερο από το μέγεθος του buffer και να γεμίσει την μνήμη με υπερβολικά πολλά δεδομένα του κακόβουλου κώδικα. Αυτά τα δεδομένα μπορεί συχνά να επικαλύψουν ένα μέρος του υπάρχοντος προγράμματος, επιτρέποντας στον επιτιθέμενο να το επαναπρογραμματίσει. Η υπερχείλιση του buffer μπορεί να αφορά τον πυρήνα του SMSC, το λειτουργικό σύστημα που υπάρχει κάτω από αυτό, τη βάση δεδομένων που χρησιμοποιεί ή τον τύπο του διακομιστή που χρησιμοποιείται από το SMSC.

Buffer Overflow



- (a) Situation when main program is running
- (b) After program *A* called
- (c) Buffer overflow shown in gray

4.2 Buffer Overflow Attack

Η υπερχείλιση του buffer στα κινητά της σειράς 45 της εταιρείας Siemens, επιτρέπει σε μακρινούς επιτιθεμένους να προκαλέσουν μία επίθεση άρνησης της υπηρεσίας μέσω ενός μηνύματος SMS το οποίο φέρει ένα μεγάλο, σε μήκος, όνομα εικόνας. Οι τεχνικές για να χρησιμοποιηθεί η επίθεση υπερχείλισης του buffer ποικίλλουν ανάλογα με την αρχιτεκτονική, το λειτουργικό σύστημα και την περιοχή της μνήμης. Ένας τεχνικά κακόβουλος χρήστης μπορεί να εκμεταλλευτεί την υπερχείλιση του buffer, που είναι βασισμένοι σε στοίβα, για να χειριστεί το πρόγραμμα με έναν από τους παρακάτω τρόπους:

- Με την επικάλυψη μίας τοπικής μεταβλητής στην μνήμη που βρίσκεται κοντά στο buffer για να αλλάξει τη συμπεριφορά του προγράμματος ωφελώντας τον επιτιθέμενο.
- Με την επικάλυψη της διεύθυνσης επιστροφής της συνάρτησης, που βρίσκεται στην στοίβα. Μόλις επιστρέψει η συνάρτηση, η εκτέλεση θα συνεχιστεί από την διεύθυνση που έχει καθοριστεί από τον επιτιθέμενο..
- Από την επικάλυψη ενός δείκτη συνάρτησης ή το χειριστή εξαιρέσεων, ο οποίος θα εκτελεστεί στη συνέχεια.

Με μία μέθοδο που λέγεται "Trampolining", εάν η διεύθυνση των δεδομένων των χρηστών είναι άγνωστη αλλά η θέση είναι αποθηκευμένη σε έναν κατάλογο, τότε η διεύθυνση επιστροφής μπορεί να επικαλυφθεί με τη διεύθυνση ενός κώδικα που θα προκαλέσει ένα άλμα στα παρεχόμενα δεδομένα του χρήστη.

4.1.5 Επίθεση ανάκτησης κωδικού πρόσβασης - Password Compromise Attack

Η κοινωνική τεχνική πειρατείας και η χρήση ωμής βίας μπορεί να κάνει πολλά σε αυτόν τον τύπο επίθεσης, ο οποίος είναι ο πιο κοινός και ο δυσκολότερος στον εντοπισμό.

Εάν υπάρχει αρκετός χρόνος, ο κωδικός πρόσβασης του SMSC μπορεί να ανιχνευθεί χρησιμοποιώντας τη μέθοδο δοκιμής και λάθους. Μόλις ένας κωδικός πρόσβασης μαντευθεί σωστά, μπορεί να προκαλέσει ισχυρό χτύπημα στο σύστημα μετάδοσης μηνυμάτων. Στο σχετικό περιβάλλον του SMSC, όπως και σε οποιοδήποτε άλλο σύστημα, αυτή η μορφή επίθεσης είναι γενικά η πιο επικίνδυνη. Δεδομένου ότι το SMSC προστατεύεται κανονικά από το δημόσιο Διαδίκτυο και επιτρέπει μόνο τις συνδέσεις από έναν προκαθορισμένο κατάλογο μηχανών, ο αριθμός των πιθανόν κινδύνων στο SMSC μειώνεται, δεδομένου ότι ένας επιτιθέμενος πρέπει να συνδεθεί άμεσα με το SMSC προκειμένου να το προσβάλει.

Εντούτοις, ένας επιτιθέμενος μπορεί να χρησιμοποιήσει μία παραλλαγή της επίθεσης που ακούει στο όνομα piggyback attack, δηλαδή στην ουσία έμμεση επίθεση. Σ' αυτήν την επίθεση ο επιτιθέμενος παίρνει τον έλεγχο μίας μηχανής στο δημόσιο Διαδίκτυο η οποία συνδέεται επίσης και με το SMSC, όπως ένας διακομιστής ηλεκτρονικού ταχυδρομείου και χρησιμοποιεί έπειτα εκείνη την μηχανή για να προσπαθήσει να εισβάλει στο SMSC.

4.1.6 Κατασκοπεία - Snooping

Μία άλλη μέθοδος που χρησιμοποιείται για την απόκτηση κωδικών πρόσβασης και άλλων πολύτιμων πληροφοριών είναι η κατασκοπεία. Η κατασκοπεία εφαρμόζεται με οποιοδήποτε τρόπο σε ένα ρεύμα δεδομένων ώστε να αποκτηθούν επιτυχώς οι πληροφορίες. Δεδομένου ότι οι μεμονωμένοι χρήστες ανταλλάζουν τα πιο προσωπικά τους δεδομένα μέσω SMS, η ανάγκη για την ιδιωτικότητα των δεδομένων αυτών γίνεται όλο και πιο σημαντική.

Η κατασκοπεία πραγματοποιείται με έναν από τους παρακάτω τρεις τρόπους:

- Η πρώτη μέθοδος γίνεται μέσω ενός συστήματος στο δίκτυο το οποίο έχει ήδη προσβληθεί από τον επιτιθέμενο με μία επίθεση πειρατείας της υπηρεσίας. Σε αυτήν την περίπτωση ένας επιτιθέμενος έχει πάρει τον έλεγχο του συστήματος και μαζί με αυτό αποκτά πρόσβαση σε ένα εμπιστευτικό ρεύμα δεδομένων. Σε αυτήν την μέθοδο περιλαμβάνονται επίσης ως επιτιθέμενοι, διαχειριστές ή λειτουργοί του συστήματος οι οποίοι κάνουν κακόβουλη χρήση της δύναμης που τους δίνεται.

  Περισσότερο ανατρεπτικό είναι το γεγονός ότι οι τράπεζες και άλλοι χρηματοδοτικοί οργανισμοί αρχίζουν να εφαρμόζουν τα συστήματα ασφάλειας, χρησιμοποιώντας τα συστήματα SMS. Επιπλέον, οι εταιρίες αρχίζουν να εφαρμόζουν λύσεις όπως η χρήση κωδικών πρόσβασης μίας χρήσης ως μέθοδο για να είναι ασφαλής οι πόροι των δικτύων. Ένας κωδικός πρόσβασης μίας χρήσης μέσω των SMS είναι ένα σύστημα στο οποίο ένας χρήστης προσπαθεί να μπει νόμιμα σε ένα συγκεκριμένο σύστημα εισάγοντας

προς το παρόν μόνο το όνομα χρήστη και στην συνέχεια το σύστημα στέλνει έναν μοναδικό κωδικό πρόσβασης στο κινητό τους τηλέφωνο, ο οποίος ισχύει μόνο για ένα σύντομο χρονικό διάστημα.

Οι κωδικοί πρόσβασης μίας χρήσης είναι ένα αποδεδειγμένο χαρακτηριστικό γνώρισμα ασφάλειας που απαλλάσσει τους χρήστες από την απομνημόνευση κωδικών πρόσβασης, αλλά οι εφαρμογές SMS του συστήματος εκθέτουν πλήρως τον χρήστη εάν το τηλέφωνο του συνδρομητή κλαπεί, ή εάν μία ξένη μηχανή κατασκοπεύει για δεδομένα του δικτύου SMS.

Όπως αναφέρεται παραπάνω, ένα δίκτυο μπορεί να επιτρέψει σε κάποιον να καταγράψει τις πληροφορίες σύνδεσης άλλων έγκυρων χρηστών και να δώσουν την ευκαιρία σε έναν επιτιθέμενο να συνεχίσει την επίθεση σε έναν μελλοντικό χρόνο.

- Δεύτερον, η κατασκοπεία συχνά ολοκληρώνεται από εξωτερικά συστήματα τα οποία έχουν συνδεθεί παράνομα και παρακολουθούν το κανάλι απ' όπου περνούν τα δεδομένα.

  Σε ένα περιβάλλον δικτύων TCP/IP τα δεδομένα δρομολογούνται από την προέλευση στον προορισμό τους περνώντας μέσα από έναν μεγάλο αριθμό υποδικτύων. Δεδομένου ότι οι πληροφορίες διαβιβάζονται πέρα από μία σύνδεση, παραλαμβάνεται επίσης και από κάθε άλλη συσκευή που συνδέεται με αυτήν την σύνδεση συμπεριλαμβανομένης και της επόμενης μηχανής στην ακολουθία δρομολόγησης για την οποία προορίζονται τα δεδομένα. Όταν μία μηχανή λάβει τα δεδομένα, προσδιορίζει από την διεύθυνση IP που υπάρχει στην επικεφαλίδα, αν τα δεδομένα προορίζονται γι' αυτήν. Έπειτα προωθεί τα δεδομένα σε ένα χωριστό υποδίκτυο που προορίζεται για μία άλλη οντότητα δρομολόγησης. Αυτή η διαδικασία συνεχίζεται έως ότου τα δεδομένα φτάσουν στην συσκευή προορισμού.

  Ενώ τα δεδομένα λαμβάνονται από κάθε άλλη συσκευή που βρίσκεται κατά μήκος της πορείας δρομολόγησης, αυτές οι μηχανές αγνοούν χαρακτηριστικά τα δεδομένα μετά από την ανάγνωση της διεύθυνσης IP, η οποία είναι γραμμένη στην επικεφαλίδα και με την οποία καθορίζεται εάν τα δεδομένα προορίζονται για τις μηχανές αυτές. Αυτές οι μηχανές όμως μπορούν να προγραμματιστούν για να καταγράψουν αυτές τις πληροφορίες. Είναι μία κοινή τακτική για τους χάκερ στην προσπάθεια τους να πάρουν τον έλεγχο κάποιας υπηρεσίας από μία συσκευή η οποία βρίσκεται κατά μήκος μίας κοινής διαδρομής του δικτύου κάποιας υπηρεσίας ή σε ένα κοντινό υποδίκτυο σ' αυτήν την υπηρεσία.

  Χρησιμοποιώντας αυτές τις μηχανές, οι επιτιθέμενοι είναι σε θέση να βλέπουν και να καταγράφουν τα δεδομένα καθώς αυτά περνούν από το SMSC. Σε αυτήν την περίπτωση, αν και το SMSC είναι ο πραγματικός στόχος της επίθεσης, οι διεισδύσεις στο δίκτυο γίνονται μέσω συνδέσεων με άλλες μηχανές, εκτός από το SMSC, όπως οι μηχανές υπηρεσιών WAP ή ηλεκτρονικού ταχυδρομείου. Οι κρυπτογραφημένες επικοινωνίες παρέχουν μία σχετική ασφάλεια όσον αφορά τις επιθέσεις στις οποίες κάποιος επιτιθέμενος υποκρίνεται έναν νόμιμο χρήστη, αλλά μπορούν επίσης να

προσπαθήσουν να σπάσουν το σχέδιο κρυπτογράφησης κλέβοντας τα κλειδιά, εάν αυτά δεν προστατεύονται καλά.

- Και τρίτον, η πειρατεία κάποιας σύνδεσης είναι μία διαδικασία στην οποία μία κατασκοπευτική μηχανή αρχίζει να μεταδίδει δεδομένα σε μία άλλη μηχανή – στόχο η οποία ακούει αυτά τα δεδομένα, πείθοντας αυτήν την μηχανή ότι τα δεδομένα προέρχονται από μία έγκυρη πηγή ενός τρίτου υπολογιστή. Η πειρατεία σύνδεσης είναι ιδιαίτερα επικίνδυνη επειδή επιτρέπει σε έναν επιτιθέμενο να περιμένει έως ότου εγκαθιδρυθεί μία νόμιμη σύνδεση και ανταλλαχτούν οι πληροφορίες του λογαριασμού και στην συνέχεια, αφού πραγματοποιήσει την επίθεση, να αποκτήσει πρόσβαση σε όλες τις δυνατότητες της συγκεκριμένης σύνδεσης κλέβοντας στην ουσία την σύνδεση από τον νόμιμο χρήστη.

  Ευτυχώς η πειρατεία σύνδεσης είναι μία σύνθετη και συχνά υπολογιστικά βαριά λειτουργία και μπορεί να είναι δύσκολη να γίνει σε πραγματικό χρόνο. Κατ' αρχάς η επιτιθέμενη μηχανή πρέπει να είναι σε θέση να κατασκοπεύσει μία σύνδεση, δηλαδή θα πρέπει να συνδεθεί με το δίκτυο κατά μήκος της διαδρομής που περνάει το ρεύμα δεδομένων από μία μηχανή πηγής προς τον προορισμό. Αφετέρου, εάν τα δεδομένα που περνούν απ' την σύνδεση κρυπτογραφούνται, θα πρέπει ο επιτιθέμενος να κατέχει τα κλειδιά κρυπτογράφησης, αλλιώς δεν θα μπορεί να κωδικοποιήσει και να αποκωδικοποιήσει τα δεδομένα με τον σωστό τρόπο.

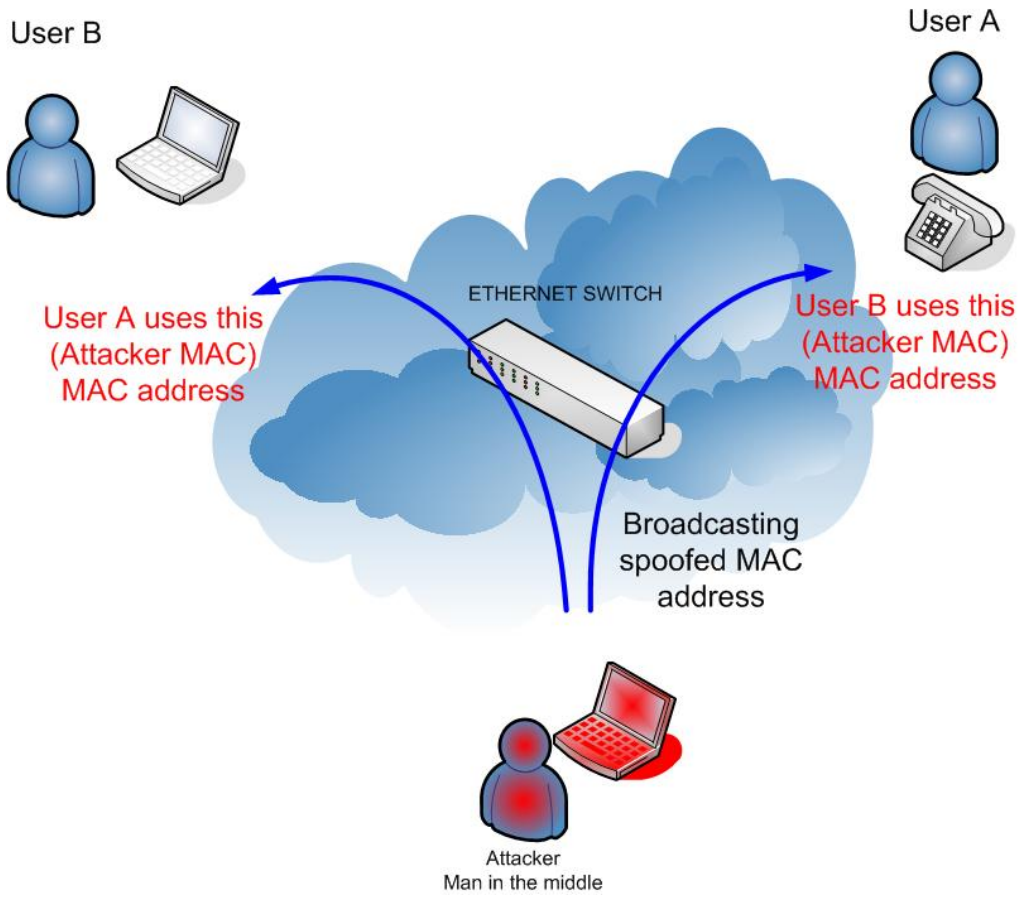
  Το τρίτο κριτήριο είναι το δυσκολότερο: Η πειρατεία στο TCP εκτελείται από την επιτιθέμενη μηχανή η οποία κρυφακούει το ρεύμα δεδομένων και στην συνέχεια βρίσκει ένα μοτίβο στο πεδίο των προσδιοριστικών του TCP σε κάθε πακέτο που περνάει από το ρεύμα. Εάν βρεθεί ένα μοτίβο, ο επιτιθέμενος πρέπει να προβλέψει μία μελλοντική τιμή αυτού του πεδίου και να περιμένει μέχρι τη στιγμή που ο αποστολέας θα στείλει ένα πακέτο με την ίδια τιμή. Εκείνη την στιγμή, η μηχανή πρέπει να αρχίσει να μεταδίδει τα δεδομένα χρησιμοποιώντας την αρχική διεύθυνση IP και MAC στην επικεφαλίδα TCP/IP και αρχίζοντας από την τιμή που προβλέφθηκε νωρίτερα. Εάν αυτό το πακέτο φθάσει στη μηχανή προορισμού την χρονική στιγμή μεταξύ της άφιξης του προηγούμενου πακέτου και της άφιξης του επόμενου, τότε ο κεντρικός υπολογιστής θα ανταποκριθεί στο πακέτο από την επιτιθέμενη μηχανή και θα αγνοήσει τα δεδομένα που προέρχονται από την αρχική πηγή.
  Ο επιτιθέμενος πρέπει έπειτα να εμποδίσει την αρχική μηχανή από το να στείλει περισσότερα δεδομένα, εφόσον αυτά φέρουν τις ίδιες πληροφορίες επικεφαλίδων TCP/IP και μπορούν να κάνουν τις στοίβες πρωτοκόλλων στη μηχανή προορισμού, να αναγνωρίσουν ότι έχει εμφανιστεί ένα πρόβλημα.

  Εάν ένας επιτιθέμενος είναι σε θέση να επιτύχει όλους αυτούς τους στόχους και γνωρίζει το πρωτόκολλο που χρησιμοποιείται για την επικοινωνία με τη μηχανή προορισμού, τότε η επίθεση πειρατείας στην σύνδεση μπορεί να επιτευχθεί.

4.1.7 Υποκρισία - Spoofing

Τα πακέτα SMS μπορούν να δημιουργηθούν και να εισαχθούν στο δίκτυο με τη χρησιμοποίηση οποιασδήποτε μηχανής δημιουργίας πακέτων όπως το Nessus, που δημιουργεί τα πακέτα για το δίκτυο TCP/IP.

Η υποκρισία αλλάζει την ταυτότητα του αποστολέα και προσποιείται ότι το μήνυμα προήλθε από κάποια άλλη πηγή. Τα χαρακτηριστικά γνωρίσματα των SMS του Διαδικτύου, όπως τα μηνύματα που βασίζονται στις ιστοσελίδες ή στο ηλεκτρονικό ταχυδρομείο, δεν έχουν κανένα σύστημα αυθεντικοποίησης και αυτό επιτρέπει σε οποιοδήποτε άτομο να στείλει μηνύματα που μπορούν να εμφανιστούν να προέρχονται από οποιαδήποτε άλλη πηγή η οποία φέρει εφαρμογή δημιουργίας μηνυμάτων. Χωρίς κάποια μορφή αυθεντικοποίησης, ένας τελικός χρήστης αναγκάζεται να αποφασίσει εάν πρέπει να εμπιστευθεί την πηγή ενός μηνύματος που λαμβάνει βασισμένος απλώς στη μεμονωμένη κρίση του και χωρίς να έχει τεχνική υποστήριξη.



4.3 Spoofing

Στην περίπτωση όπου σε ορισμένους χρήστες εμπιστοσύνης χορηγείται η άμεση πρόσβαση σε έναν πάροχο δικτύου SS7, ένας χάκερ θα μπορούσε να στείλει ένα κακόβουλο μήνυμα ελέγχου στο δίκτυο υποκρινόμενος κάποιον απ' αυτούς τους χρήστες. Ακριβώς όπως δεν υπάρχει κανένας έλεγχος στα μηνύματα SMTP, ένας χάκερ μπορεί να δημιουργήσει ένα κακόβουλο μήνυμα ελέγχου κάτω από την αμφίεση ενός χρήστη εμπιστοσύνης για να μπορέσει να προσβάλει το δίκτυο.

Επιπλέον, κάποιος μπορεί να χρησιμοποιήσει τα νόμιμα εργαλεία SMS τα οποία είναι διαθέσιμα στην αγορά ώστε να πετύχει την υποκρισία. Παραδείγματος χάριν, το Clickatell που είναι ένας πάροχος μαζικών λύσεων και εφαρμογών μηνυμάτων SMS, που επιτρέπει στους χρήστες να στείλουν μαζικά ή εξατομικευμένα μηνύματα SMS.

Ενώ το κεφάλαιο για την κακή χρήση της υποκρισίας στα SMS είναι τεράστιο, μπορεί επίσης να χρησιμοποιηθεί και με θετικό τρόπο. Παραδείγματος χάριν, η υποκρισία μπορεί να βοηθήσει τις ανακριτικές αρχές που επιβάλλουν τον νόμο να παγιδέψουν τους τρομοκράτες και τους έμπορους ναρκωτικών. Επίσης μπορούν να παγιδέψουν εγκληματίες και γκάγκστερ του υπόκοσμου στέλνοντας τους ένα μήνυμα SMS στα κινητά τους τηλέφωνα.

4.1.8 SMS phishing

Σε αυτήν την τεχνική ένας χρήστης κινητού τηλεφώνου λαμβάνει ένα SMS που τον προτρέπει να επισκεφτεί μία ιστοσελίδα. Το SMS θα μπορούσε να γράφει κάτι σαν το ακόλουθο: *επιβεβαιώνουμε ότι έχετε υπογράψει την υπηρεσία μας. Η χρέωση είναι 2$ την ημέρα, εκτός και αν ακυρώσετε την συμφωνία στην διεύθυνση [διεύθυνση ιστοσελίδας]*. Κατά την επίσκεψη στην ιστοσελίδα, ο χρηστής καλείται να κατεβάσει ένα πρόγραμμα που ισχυρίζεται ότι είναι ένα ελεύθερης χρήσης αντιβιοτικό, αλλά στην πραγματικότητα είναι ένας δούρειος ίππος (Trojan Horse), που κάνει το κινητό του χρήστη να ελέγχεται απομακρυσμένα από έναν χάκερ.

Στην Ισπανία κυκλοφόρησε ένας ιός με το όνομα VBS/Eliles.A ο οποίος περιλάμβανε μία ρουτίνα η οποία έστελνε μηνύματα phising στους χρήστες δύο παρόχων κινητής τηλεφωνίας. Παρά τον υπολογισμό τυχαίων διευθύνσεων IP για να στείλει τα μηνύματα, αυτό το σκουλήκι παρήγαγε τους τηλεφωνικούς αριθμούς μέσα από τις σειρές που χρησιμοποιούνται από κινητά τηλέφωνα. Ο Eliles.A έστελνε το μήνυμα phising του δωρεάν μέσω των πυλών ηλεκτρονικού ταχυδρομείου – SMS των παρόχων κινητής τηλεφωνίας. Επίσης έδειχνε ότι είναι χρήσιμο στο τηλέφωνο με την προσφορά ενός αντιβιοτικού ελεύθερης χρήσης που υποθετικά προερχόταν από τον δίκτυο του προμηθευτή τους. Το smishing μήνυμα στοχεύει συγκεκριμένα στα κινητά της σειράς 60 της εταιρείας Nokia. Οι χρήστες που κατεβάζουν και εγκαθιστούν το λογισμικό από τη σελίδα που τους υποδεικνύει το SMS, βρίσκονται να είναι μολυσμένοι με το κακόβουλο λογισμικό. Ευτυχώς, η σύνδεση που κατεβάζει αυτό το λογισμικό δεν υφίσταται πλέον.

4.1.9 Φραγή ραδιοσυχνότητας - Radio Frequency Jamming

Είναι μία επίθεση η οποία δεν βασίζεται στο Διαδίκτυο και σχετίζεται με την ασύρματη διεπαφή του συστήματος επικοινωνιών. Ο μόνος σκοπός του είναι να κάνει

τους συνδρομητές σε μία ιδιαίτερη περιοχή να μην μπορούν να χρησιμοποιήσουν μία υπηρεσία. Είναι ευρέως γνωστό ότι τα συστήματα GSM, IS-136 και iDEN, είναι εύκολο να μπλοκαριστούν. Οι τεχνολογίες που χρησιμοποιούν CDMA είναι βασισμένες στις αρχές διάχυσης φάσματος και επομένως είναι πιο δύσκολο να μπλοκαριστούν. Ο εξοπλισμός που προκαλεί την φραγή μπορεί να είναι μία πηγή θορύβου που εκπέμπει τα αυθαίρετα σήματα στην συχνότητα που λειτουργεί το GSM(κανονικά 900 και 1800 MHz). Αυτή η επίθεση δεν επηρεάζει μόνο τα μηνύματα SMS αλλά και τις κλήσεις φωνής.

Ένα πλεονέκτημα αυτής της τεχνικής είναι ότι έχει επιπτώσεις σε όλη την ασύρματη κυκλοφορία (SMS ή φωνή) μέσα σε ένα σύστημα. Ένα άλλο πλεονέκτημα είναι ότι είναι απλό να εφαρμοστεί. Όλα τα κομμάτια που απαιτούνται κυκλοφορούν στο εμπόριο. Μία φορητή λύση γι' αυτήν την επίθεση θα μπορούσε να είναι πολύ αποτελεσματική. Ένα άλλο πλεονέκτημα είναι ότι ο εξοπλισμός που προκαλεί την φραγή δεν ανιχνεύεται εύκολα. Το κύριο μειονέκτημα αυτής της προσέγγισης είναι ότι είναι πολύ ευαίσθητη σε θέμα τοποθεσίας και τεχνολογίας. Ενώ θα ήταν πολύ απλό να μπλοκάρει ένα ή περισσότερα δίκτυα σε μία συγκεκριμένη περιοχή, θα ήταν πολύ δυσκολότερο να μπλοκαριστούν όλα τα δίκτυα σε μία μεγάλη γεωγραφική περιοχή.

Οι συσκευές φραγής εξουδετερώνουν το κινητό τηλέφωνο μεταδίδοντας ένα σήμα στην ίδια συχνότητα και σε μία αρκετά υψηλή ισχύ με αποτέλεσμα τα δύο σήματα να συγκρούονται μεταξύ τους και να ακυρώνουν το ένα το άλλο. Τα κινητά τηλέφωνα σχεδιάζονται με τέτοιο τρόπο έτσι ώστε όταν δέχονται παρεμβολές χαμηλού επιπέδου να ανεβάζουν την ισχύ που εκπέμπουν. Γι' αυτόν τον λόγο η συσκευή που προκαλεί την φραγή θα πρέπει να το αναγνωρίζει αυτό και να προσαρμόζει και αυτή ανάλογα την ισχύ της.

Τα κινητά τηλέφωνα είναι πλήρως αμφίδρομες συσκευές, που σημαίνει ότι χρησιμοποιούν δύο διαφορετικές συχνότητες ταυτόχρονα, μία για την αποστολή δεδομένων και μία για την λήψη. Μερικές συσκευές φραγής εμποδίζουν μόνο μία από τις συχνότητες που χρησιμοποιούν τα κινητά τηλέφωνα η οποία έχει επίδραση και στην φραγή της άλλης συχνότητας. Το τηλέφωνο εξαπατείται ως προς το ότι δεν υπάρχει καμία διαθέσιμη υπηρεσία, επειδή λαμβάνει μόνο μία από τις δύο συχνότητες. Οι λιγότερο σύνθετες συσκευές, εμποδίζουν μόνο μία ομάδα συχνοτήτων, ενώ οι πιο περίπλοκες μπορούν να εμποδίσουν διάφορους τύπους δικτύων, λόγω της ύπαρξης κινητών τηλεφώνων που αλλάζουν αυτόματα συχνότητα λειτουργίας για να βρουν ένα ανοικτό κανάλι επικοινωνίας. Μερικές από αυτές τις περίπλοκες συσκευές εμποδίζουν όλες τις συχνότητες ταυτόχρονα και άλλες συντονίζονται ανάλογα στις συγκεκριμένες συχνότητες.

4.1.10 SMS Spam

Τα ανεπιθύμητα e-mail είναι όλεθρος για οποιοδήποτε οργανισμό που βρίσκεται σε κατάσταση online. Ομοίως τα εκούσια ηλεκτρονικά μηνύματα γίνονται ενοχλητικά για τους χρήστες των ασύρματων συσκευών. Εάν ένας προμηθευτής ασύρματης επικοινωνίας έχει μία λίστα με αρκετές χιλιάδες συνδρομητών, μπορεί να στείλει μαζικά μηνύματα σε όλους αυτούς, προκαλώντας περιττό φορτίο κυκλοφορίας στο δίκτυο και διαταραχή στους παραλήπτες. Τα νομικά μέτρα έχουν ληφθεί από

ορισμένες κυβερνήσεις για την προστασία ενάντια στο spam που μεταδίδεται ασύρματα και έχουν γίνει διάφορες επενδύσεις για να αναπτυχθούν συστήματα αντί-spam.

Συχνά αυτά τα μηνύματα αποτελούνται από ένα απλό αίτημα να κληθεί ένας αριθμός. Η κανονική κινητή τηλεφωνική εθιμοτυπία οδηγεί συχνά το χρήστη στο να καλέσει αυτόν τον αριθμό. Όταν κάποιος χρηστής κάνει αυτήν την κλήση δεν ξέρει ότι μιλάει σε γραμμή υψηλής χρέωσης. Συχνά γίνεται προσπάθεια να μείνει ο εξαπατημένος χρηστής στην γραμμή για όσο περισσότερο χρόνο γίνεται ώστε να μεγιστοποιηθούν τα κέρδη απ' αυτήν την άπατη.

4.1.11 Εισβολή στο OSS

Το σημαντικότερο μέρος ενός δικτύου GSM δεν είναι ο ίδιος ο εξοπλισμός του δικτύου αλλά το σύστημα λειτουργίας και υποστήριξης (OSS). Είναι ένα δίκτυο συσκευών που διαχειρίζονται σημαντικές λειτουργίες όπως το σύστημα τιμολόγησης και οι οποίες είναι πολύ κρίσιμες για την ασφάλεια του GSM. Το πιο ενδιαφέρον μέρος του OSS είναι ότι αυτή η υποδομή είναι προσιτή μέσω των δικτύων IP, έτσι όλες οι ευπάθειες στο δίκτυο IP κληρονομούνται άμεσα και στο OSS.

4.1.12 Ιοί κινητών

Είναι πραγματικά μία απειλή στο λογισμικό του κινητού που μπορεί να επιτρέψει την μη εξουσιοδοτημένη πρόσβαση στα προγράμματα ή στην εκτέλεση των εντολών. Οι περισσότεροι ιοί SMS μπορούν πάντα να προκαλέσουν το κλείσιμο και την καταστροφή του λογισμικού του συστήματος.

Παραδείγματα ιών είναι:

Palm.Phage.A:

Ένα Malware που έχει επιπτώσεις σε υπολογιστές τσέπης (PalmOS) και το οποίο μπορεί να είναι ένας απλός ιός που μολύνει εκτελέσιμα αρχεία ή μπορεί να είναι ένα κομμάτι κακόβουλου κώδικα που ενσωματώνεται στα αρχεία αυτά, τα οποία είναι επίσης και κοινόχρηστα. Ανακαλύφθηκε στις 1 Οκτωβρίου του 1998.

Είναι ο πρώτος γνωστός ιός που μολύνει τις εφαρμογές PalmPilot. Ο ιός έχει έναν μηχανισμό μόλυνσης που ονομάζεται "overwriting" και κάνει τις επηρεαζόμενες εφαρμογές να παύουν να λειτουργούν. Οι εφαρμογές PalmPilot (εκτελέσιμα αρχεία με κατάληξη .PRC) είναι τυποποιημένες βάσεις δεδομένων των Pilot μαζί με επιπλέον πόρους "εφαρμογής" που βρίσκονται εσωτερικά. Όταν μία εφαρμογή τρέχει, ο πόρος αυτός ενεργοποιείται και εκτελεί τις λειτουργίες της εφαρμογής. Υπάρχει ένα σύνολο από βιβλιοθήκες ρουτινών του συστήματος, το οποίο χρησιμοποιούν οι εφαρμογές αυτές για να μπορούν να προσπελάζουν βάσεις δεδομένων και να χρησιμοποιούν τους πόρους του συστήματος. Όταν ο ιός ξεκινάει να εκτελείται, ανοίγει το αρχείο του και διαβάζει τους πόρους του κώδικα και των δεδομένων απο 'κεί. Έπειτα αναζητά για άλλες βάσεις δεδομένων εφαρμογών και όταν τις βρει, αντικαθιστά τα δεδομένα και τον κώδικά τους με τον ιό. Οι

επηρεαζόμενες εφαρμογές φέρουν πλέον τον κακόβουλο κώδικα και τα δεδομένα του ιού.

Cabir:

Ο Cabir είναι ο πρώτος ιός για κινητά τηλέφωνα που μεταδίδεται μέσω της ασύρματης διασύνδεσης Bluetooth και μολύνει συσκευές που έχουν λειτουργικό Symbian. Αυτή η τεχνολογία χρησιμοποιείται από έναν μεγάλο αριθμό τηλεφώνων από διάφορους κατασκευαστές.

Ο Cabir είναι δημιούργημα μίας ομάδας προγραμματιστών, που είναι γνωστή ως 29a(ο αριθμός 666 στο δεκαεξαδικό σύστημα) και η οποία πρόσφατα κατασκεύασε το πρώτο ιό που στοχεύει την 64bit έκδοση των Windows. Οι δημιουργοί του Cabir δεν σχεδίασαν αυτόν τον ιό για μαζική εξάπλωση, αλλά τον χρησιμοποίησαν δοκιμαστικά για να αποδείξουν ότι και αυτά τα είδη συσκευών μπορούν να μολύνονται από κακόβουλο κώδικα.

Ο Cabir περιέχεται σ' ένα αρχείο με όνομα Caribe.sis, ο οποίος εγκαθιστά αυτόματα τον εαυτό του στο σύστημα όταν ο χρήστης αποδέχεται τη μετάδοση. Εμφανίζει ένα μήνυμα στην οθόνη με το κείμενο Caribe και κατόπιν ξεκινά μία διαρκή αναζήτηση για άλλες συσκευές σε ακτίνα 10 μέτρων στις οποίες μπορεί να στείλει τον εαυτό του, αν και για να το καταφέρει οι συσκευές αυτές θα πρέπει να έχουν ενεργή την τεχνολογία Bluetooth. Ο ιός επηρεάζει μόνο την αυτονομία της κινητής συσκευής μειώνοντας δραστικά το επίπεδο φόρτισης της μπαταρίας.

Επίσης, το αρχείο Caribe.sis είναι δυνατό να αντιγραφεί και σε άλλες συσκευές που χρησιμοποιούν τεχνολογία Bluetooth π.χ. ορισμένοι εκτυπωτές, παρά το ότι δεν χρησιμοποιούν το προαναφερθέν λειτουργικό σύστημα. Ωστόσο, σ'αυτές τις περιπτώσεις, ο ιός παύει να έχει δυνατότητα περαιτέρω εξάπλωσης.

Υποψήφια θύματα του ιού είναι τα κινητά τηλέφωνα που βασίζονται στην πλατφόρμα Series 60, όπως τα Nokia N-Gage, 3650, 3660, 6600, 7610 κ.α.

Dust:

Είναι δημιούργημα της ίδιας ομάδας που δημιούργησε τον Cabir, της 29a και φτιάχτηκε για τον ίδιο λόγο, για να αποδειχτεί ότι και τα κινητά μολύνονται από ιούς. Απελευθερώθηκε στις 16 Ιουλίου του 2004.

Είναι ένας ιός ο οποίος προσβάλει κινητά τηλέφωνα που φέρουν το λειτουργικό σύστημα Windows CE. Αυτό που κάνει είναι να μολύνει όλα τα εκτελέσιμα αρχεία που βρίσκονται μόνο στον αρχικό κατάλογο της συσκευής. Μόλις ο ιός λάβει δράση εμφανίζεται στην οθόνη του κινητού ένα μήνυμα που πληροφορεί τον χρήστη για την παρουσία του ιού και τον ρωτάει αν θέλει να τον μολύνει. Αν ο χρήστης απαντήσει θετικά τότε ο ιός προσθέτει κακόβουλο κώδικα στα εκτελέσιμα αρχεία του αρχικού καταλόγου της συσκευής. Παρόλα αυτά ο ιός δεν αποτελεί σοβαρό κίνδυνο αφού φτιάχτηκε μόνο και μόνο για να αποδείξουν οι δημιουργοί του ότι και οι συσκευές κινητής τηλεφωνίας προσβάλλονται από κακόβουλο λογισμικό. Αυτό αποδεικνύεται

και απ' το γεγονός ότι τα αρχεία τα οποία μολύνονται συνεχίζουν να λειτουργούν κανονικά χωρίς πρόβλημα.

Brador Trojan:

Στις 5 Αυγούστου του έτους 2004, βρέθηκε ο πρώτος γνωστός ιός για υπολογιστές τσέπης (Pocket PC) που τρέχουν το λειτουργικό Windows CE. Αυτός ο ιός είχε μέγεθος 5.632 Bytes, ανήκε στην κατηγορία των Δούρειων Ίππων και είχε κωδικοποιηθεί χρησιμοποιώντας τον γενικό assembler των επεξεργαστών ARM.

Ο ιός αυτός περιείχε εντολές για να στείλει μία σύντομη σημείωση με την μορφή:
From: br@mail.ru
To: brokensword@ukr.net

Η σημείωση στέλνεται ως ανακοίνωση ότι η συσκευή είναι έτοιμη να δεχτεί επίθεση και μαζί στέλνεται και η διεύθυνση IP της συσκευής. Επίσης ο ιός δεσμεύει την πόρτα με αριθμό 2989 για να λάβει οδηγίες από τον απομακρυσμένο εισβολέα, ο οποίος μετά απ' αυτό μπορεί να ελέγξει πλήρως την απομακρυσμένη συσκευή. Ο ιός καθώς εκτελείται μπορεί να αντιγράψει τον εαυτό του στον φάκελο "C:\windows\startup\" και οποιοδήποτε αρχειο υπάρχει μέσα σ'αυτόν τον φάκελο μπορεί να εκτελέσει το αρχείο του ιού.

4.2 Τεχνικές ασφάλειας SMS

Τα SMS κληρονομούν και χρησιμοποιούν όλους τους μηχανισμούς ασφάλειας του δικτύου που μεταδίδονται παρά και την στήριξη από ειδικά χαρακτηριστικά γνωρίσματα ασφάλειας. Η μεταβαλλόμενη ιδιόκτητη προστασία των μηνυμάτων SMS γενικά δεν είναι διαθέσιμη. Είναι απίθανο να εμφανιστούν νέα πρότυπα αυθεντικοποίησης ή κρυπτογράφησης για SMS στο μέλλον, εφόσον η τρέχουσα ασφάλεια θεωρείται αρκετά καλή για περιστασιακή χρήση, δεδομένου ότι οι νέοι αλγόριθμοι θα απαιτούσαν αλλαγές στα δίκτυα και στα κινητά τηλέφωνα (μία δαπανηρή πρόταση) και θα εμφανίζονταν προβλήματα διαλειτουργικότητας τα οποία θα έπρεπε να εξεταστούν.

Από την άποψη της ιδιωτικότητας των μηνυμάτων, ένα μήνυμα SMS περνάει μέσα από τρία σημαντικά κομμάτια του δικτύου που θα πρέπει να εξεταστούν: μέσα από το δίκτυο SS7, ασύρματα και πιθανόν μέσα από το Διαδίκτυο. Όλα τα μηνύματα περνούν μέσα από τα δύο πρώτα δίκτυα, ενώ τα μηνύματα που προέρχονται από εφαρμογές ή από χρήστες του Διαδικτύου, περνούν μέσα από το δημόσιο Διαδίκτυο. Επομένως, αναλύονται οι ακόλουθοι δύο τύποι μηνυμάτων SMS:

1.Μηνύματα που προέρχονται από τους χρήστες ή τις εφαρμογές εκτός του δικτύου κινητής όπως το Διαδίκτυο. (Μηνύματα προερχόμενα από εφαρμογές)
2.Μηνύματα που προέρχονται από ένα άλλο κινητό τηλέφωνο. (Μηνύματα προερχόμενα από κινητά)

4.2.1 Μηνύματα προερχόμενα από εφαρμογές

Τα μηνύματα αυτά προέρχονται από συνδέσεις εκτός του δικτύου κινητής τηλεφωνίας και τα οποία δρομολογούνται χρησιμοποιώντας μία δημόσια διεύθυνση IP. Αυτού του τύπου η επικοινωνία δεν θεωρείται ασφαλής καθ' ότι τα δεδομένα δεν κρυπτογραφούνται.

Τα περισσότερα SMSC προστατεύονται από έναν κατάλογο που λέγεται λεύκη λίστα. Η λεύκη λίστα περιέχει διάφορες διευθύνσεις IP και MAC διαφόρων μηχανών απ'όπου ο διακομιστής μπορεί να δέχεται συνδέσεις. Ο διακομιστής αρνείται τις συνδέσεις από IP που δεν βρίσκονται στην λίστα. Μερικά SMSC απαιτούν επίσης και έναν συνδυασμό ονόματος χρήστη και κωδικού πρόσβασης, πριν επιτρέψουν στα μηνύματα να σταλούν μέσα από την συγκεκριμένη σύνδεση. Εντούτοις, μερικά SMSC δεν έχουν κανένα μηχανισμό ασφάλειας και δέχονται όλες τις συνδέσεις από οποιαδήποτε μηχανή στο διαδίκτυο.

Ενώ αυτά τα χαρακτηριστικά γνωρίσματα ασφάλειας αποτρέπει τα μηνύματα Spam και τους μεγάλους όγκους ανώνυμων μηνυμάτων από την εισαγωγή τους στο ασύρματο δίκτυο, εντούτοις, δεν προστατεύουν τη μυστικότητα των μηνυμάτων που περνάνε. Γι' αυτόν τον λόγο, μερικά SMSC επιτρέπουν μόνο τις εισερχόμενες συνδέσεις οι οποίες είναι Secure Shell (SSH) tunnelled, να συνδεθούν στο SMSC. Η μέθοδος SSH είναι μία μέθοδος μετάδοσης κρυπτογραφημένων δεδομένων, τα οποία δρομολογούνται μέσω μίας εφαρμογής SSH προς τον αποστολέα. Αντί να συνδεθεί άμεσα με ένα SMSC στην κανονική SMPP, UCP, ή CIMD πόρτα, η εφαρμογή αποστολής συνδέεται σε μία ξεχωριστή πόρτα η οποία είναι αποκλειστικά για συνδέσεις SSH στο SMSC. Ο αποστολέας παρέχει έπειτα υπηρεσίες SSH στις SMPP, UCP, ή CIMD πόρτες οι οποίες λαμβάνουν τα δεδομένα, τα αποκρυπτογραφούν και τα δίνει έπειτα για αποστολή στην πόρτα προορισμού χρησιμοποιώντας κοινόχρηστα buffer μνήμης.

4.2.2 Μηνύματα προερχόμενα από κινητά

Τα μηνύματα που στέλνονται από ένα κινητό τηλέφωνο σε ένα άλλο δεν αφήνουν ποτέ το ιδιωτικό δίκτυο του παρόχου και έχουν περισσότερα πλεονεκτήματα ασφάλειας έναντι των μηνυμάτων που προέρχονται από εφαρμογές. Εντούτοις σε κάποια σημεία όλα τα μηνύματα που μεταδίδονται ασύρματα εμφανίζουν πρόσθετα ζητήματα ασφάλειας.

Όταν φτάσει ένα μήνυμα SMS στο SMSC πρέπει να καθοδηγηθεί κατάλληλα ώστε να φτάσει στον κατάλληλο σταθμό βάσης και από κει στο κινητό τηλέφωνο. Τα SMS μεταδίδονται από το SMSC στο MSC χρησιμοποιώντας μηνύματα τύπου SS7. Το δίκτυο SS7 δεν κρυπτογραφεί τα δεδομένα πριν από τη μεταφορά τους, αλλά ολόκληρο το δίκτυο είναι χωρισμένο από το Διαδίκτυο και προστατεύεται από κάποια πιθανή πρόσβαση. Όλοι οι φορείς ελέγχου παρέχουν πολύ δύσκολη πρόσβαση στα δίκτυα SS7 και περιορίζουν ακόμη και το ποιος μπορεί να δει τα δεδομένα που αφορούν το δίκτυο. Η παρεμβολή στα δίκτυα αυτά απαιτεί την άμεση πρόσβαση στον εξοπλισμό του παρόχου υπηρεσίας ή την εγκατάσταση μίας συσκευής ακρόασης σε κάποιο σημείο κατά μήκος της καλωδιωτής σύνδεσης του δικτύου. Αυτό καθιστά το SS7 αρκετά ασφαλές για τους περισσότερους χρήστες, αν και δεν καλύπτει την

απαίτηση FIPS(Federal Information Processing Standard - πρότυπο επεξεργασίας ομοσπονδιακών πληροφοριών), που αφορά τις ευαίσθητες πληροφορίες.

Αφού έχει δρομολογηθεί επιτυχώς ένα μήνυμα προς τον σταθμό βάσης για τη μετάδοση του στο κινητό τηλέφωνο, στην συνέχεια μεταδίδεται ασύρματα προς αυτό. Σε αυτό το σημείο το μήνυμα αφήνει το σταθερό δίκτυο και πρέπει να προστατευθεί με άλλα μέσα. Οι σημαντικότεροι τύποι ψηφιακών ασύρματων δικτύων (CDMA, iDEN, GSM, και TDMA) κατέχουν μηχανισμούς ισχυρής αυθεντικοποίησης και δρομολόγησης για να εξασφαλίσουν ότι τα μηνύματα δεν θα διαβαστούν από μη έγκυρα κινητά. Από όλες τις τεχνολογίες, το δίκτυο GSM προσφέρει την πλουσιότερη προστασία για τα δεδομένα των μηνυμάτων SMS.

Κάθε συσκευή GSM περιέχει μία κάρτα ταυτότητας συνδρομητών (SIM), η οποία είναι μία αφαιρούμενη κάρτα μνήμης που φέρει το κλειδί αυθεντικοποίησης συνδρομητών, τον προσωπικό αριθμό ταυτότητας (PIN) του συνδρομητή και πληροφορίες για την αυθεντικοποίηση και την παραγωγή του κλειδιού κρυπτογράφησης. Η αυθεντικοποίηση παρέχεται από ένα ειδικό κέντρο αυθεντικοποίησης (AUC), το οποίο χρησιμοποιεί έναν αλγόριθμο πρόκλησης-απάντησης και ένα κοινό μυστικό στοιχείο για να επιβεβαίωση την ταυτότητα του συνδρομητή.

Ο αλγόριθμος A5 του GSM παρέχει τη μυστικότητα μέσω της κρυπτογράφηση της φωνής, των δεδομένων και των μηνυμάτων SMS. Επειδή ο αλγόριθμος A5 είναι ένας αλγόριθμος συμμετρικής κρυπτογράφησης, που σημαίνει ότι βασίζεται στην χρήση ενός μυστικού κλειδιού, ο αποστολέας και ο παραλήπτης χρησιμοποιούν το ίδιο κλειδί για να κρυπτογραφήσουν και να αποκρυπτογραφήσουν τα δεδομένα. Για να αποφύγουν την μετάδοση αυτών των κλειδιών ασύρματα, το κινητό τηλέφωνο και το δίκτυο χρησιμοποιούν τις πληροφορίες από την αυθεντικοποίηση με σκοπό να παράγουν και οι δύο το κλειδί ανεξάρτητα. Το κλειδί κρυπτογράφησης A5 θεωρείται ότι πρέπει να έχει περίπου μήκος 40 με 64bit, που σημαίνει ότι μία μηχανή που δοκιμάζει ένα εκατομμύριο κλειδιά ανά δευτερόλεπτο, θα χρειαζόταν 13 ημέρες (40bit) έως 584.542 έτη (64bit) για να το ανακτήσει. Μία συντηρητική εκτίμηση θα ήταν ότι ένα μήνυμα προστατεύεται για αρκετές εβδομάδες.

Τελικά, το δίκτυο GSM παρέχει την ανωνυμία μέσω της μη διαβίβασης του τηλεφωνικού αριθμού του συνδρομητή μέσω της ασύρματης διασύνδεσης. Αυτό εφαρμόζεται μέσω της χρήσης της διεθνούς ταυτότητας συνδρομητών κινητής (IMSI) και της προσωρινής ταυτότητας συνδρομητών κινητής(TMSI). Το IMSI μεταδίδεται όταν ενεργοποιείται ένα κινητό τηλέφωνο, εντούτοις από εκείνο το σημείο και έπειτα, το δίκτυο χρησιμοποιεί μία σειρά από κρυπτογραφημένες προσωρινές ταυτότητες συνδρομητών κινητής (TMSI) για να προσδιορίσει τον συνδρομητή του δικτύου. Ακόμα κι αν ένας επιτιθέμενος αποκρυπτογραφήσει τα δεδομένα που εκπέμπονται στον αέρα και αποκωδικοποιεί κάποιες πληροφορίες, η ταυτότητα του χρήστη δεν θα αποκαλυπτόταν.

4.3 Ασφάλεια αποστολέα - παραλήπτη

Είναι σημαντικό να γίνει κατανοητό ότι τα χαρακτηριστικά γνωρίσματα ασφάλειας στην ασύρματη διεπαφή και στο ίδιο το δίκτυο είναι ανεξάρτητα. Τα χαρακτηριστικά γνωρίσματα ασφάλειας του δικτύου δεν συνεχίζονται πέρα από τον σταθμό βάσης. Αυτό έρχεται σε αντίθεση για την ασύρματη διασύνδεση όπου ο καθένας μπορεί να έχει πρόσβαση.

Τα ασύρματα δίκτυα δεν σχεδιάστηκαν για να παρέχουν υψηλή ασφάλεια κατά την αποστολή και λήψη διαφόρων δεδομένων και γι' αυτό δεν είναι κατάλληλα για την μετάδοση πολύ ευαίσθητων δεδομένων. Η μόνη εναλλακτική λύση για να προστατευτούν τα δεδομένα στα μηνύματα SMS, είναι να παρέχεται κρυπτογράφηση των μηνυμάτων στην αρχική συσκευή αποστολής (συγγραφέας) και έπειτα αποκρυπτογράφηση στη συσκευή λήψης (αναγνώστης). Μία τέτοια μέθοδος θα προστάτευε τα δεδομένα καθ'όλη την διάρκεια της μετάδοσης μέσω του δικτύου, αλλά η εφαρμογή ενός τέτοιου συστήματος είναι ιδιαίτερα δύσκολη.

Προκειμένου να είναι υποχωρητικός με όλους τους τύπους SMS, ο αλγόριθμος κρυπτογράφησης πρέπει να πληροί τρεις ιδιότητες: Κατ'αρχάς, ο αλγόριθμος δεν πρέπει να δίνει σαν αποτέλεσμα τα καθαρά αρχικά δυαδικά δεδομένα. Το κρυπτογραφημένο μήνυμα πρέπει να είναι υπό μορφή κρυπτογραφημένου κειμένου προκειμένου να ανταποκρίνεται στα πρότυπα των μηνυμάτων SMS. Δεύτερον, ο αλγόριθμος κρυπτογράφησης δεν μπορεί να αλλάξει δραστικά το μέγεθος του μηνύματος, δεδομένου ότι αυτό θα ανάγκαζε τα αρχικά μεγάλα μηνύματα να υπερβούν το μέγιστο μέγεθος μετά την κρυπτογράφηση. Τρίτο, και δυσκολότερο, είναι ότι δεν μπορεί να είναι ένας κοινός αλγόριθμος δημόσιου κλειδιού, δεδομένου ότι τα κλειδιά δεν μπορούν εύκολα να ανταλλαχθούν μέσω του δικτύου. Οι αλγόριθμοι μυστικού κλειδιού είναι χρήσιμοι δεδομένου ότι η κινητή συσκευή μπορεί να αποθηκεύσει ένα μυστικό κλειδί, αλλά αυτός ο τύπος αλγορίθμου απαιτεί ένα δύσκολο σύστημα διαχείρισης κλειδιών που απαγορεύει απόλυτα τη χρήση του.

Αυτήν την περίοδο η μεγάλη πλειοψηφία των τηλεφωνικών συσκευών είναι τηλέφωνα χωρίς τη δυνατότητα εγκατάστασης νέου λογισμικού σε αυτά. Οι μεμονωμένοι προγραμματιστές έχουν κατορθώσει να αποσυναρμολογήσουν δύο συσκευές τηλεφώνων GSM της Nokia και να ξαναγράψουν το firmware τους ώστε να επιτρέψουν την κρυπτογράφηση των μηνυμάτων κειμένου. Χωρίς όμως την υποστήριξη από τους κατασκευαστές, αυτό το λογισμικό είναι εξαιρετικά αναξιόπιστο και δεν μπορεί να προχωρήσει εμπορικά. Επιπλέον, υποφέρει και από το πρόβλημα διαχείρισης των κλειδιών. Οι νέες συσκευές όμως, όπως οι υπολογιστές τσέπης και τα έξυπνα τηλέφωνα, περιέχουν αρκετή μνήμη και υπολογιστική δύναμη ώστε να τρέχουν προηγμένους αλγόριθμους κρυπτογράφησης που θα μπορούσαν να ενισχύσουν την ασφάλεια στα SMS.

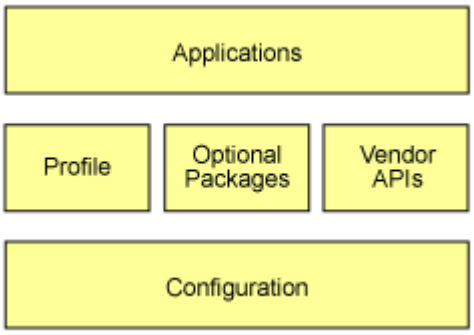
ΚΕΦΑΛΑΙΟ 5 - MIDP και CLDC

5.1 Εισαγωγή

Η γλώσσα Java όσο αναφορά την λειτουργία της χωρίζεται σε τρία επίπεδα. Το χαμηλότερο επίπεδο λέγεται Configuration και παρέχει τις βασικές κλάσεις και διασυνδέσεις που χρειάζονται για την εκτέλεση της εικονικής μηχανής της Java (JVM). Καθορίζει επίσης και την JVM που θα εκτελεστεί καθώς και λίγες χαμηλού επιπέδου λειτουργίες του περιβάλλοντος της γλώσσας. Ένα παράδειγμα είναι το CLDC και το CDC. Εδώ θα περιγράφει το CLDC γιατί είναι αυτό που χρησιμοποιείται στις μικροσυσκευές.

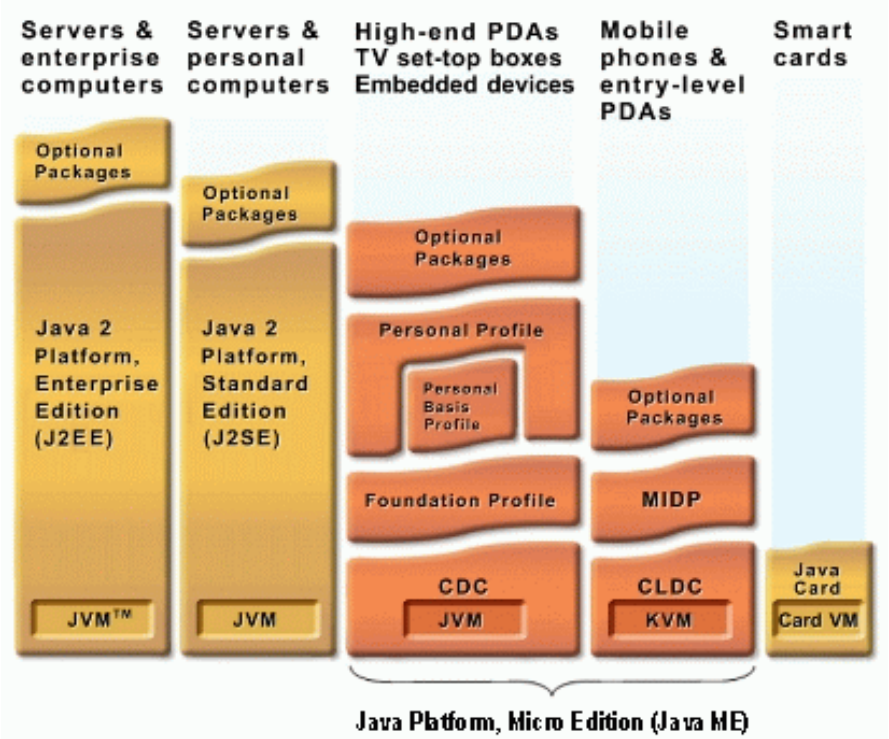
Στο δεύτερο επίπεδο υπάρχουν τα Profiles τα οποία επεκτείνουν τις λειτουργίες του Configuration συμπληρώνοντάς το με πρόσθετες βιβλιοθήκες που χρειάζονται για την πραγματική εφαρμογή, παρέχοντας έτσι στην συσκευή λειτουργίες υψηλού επιπέδου όπως είναι για παράδειγμα αποθήκευση δεδομένων σε τοπικό αποθηκευτικό χώρο, πρωτόκολλα δικτύωσης, εξελιγμένη διασύνδεση με τον χρήστη και λειτουργία παιχνιδιών και πολυμέσων. Ένα παράδειγμα Profile είναι το MIDP το οποίο χρησιμοποιείται στις μικροσυσκευές.

Στο υψηλότερο επίπεδο βρίσκονται οι πραγματικές εφαρμογές οι οποίες χωρίζονται σε τρεις κατηγορίες: J2EE, J2SE και J2ME. Η τελευταία είναι αυτή που χρησιμοποιείται στις μικροσυσκευές. Το παρακάτω σχήμα δείχνει τα τρία επίπεδα. Στο δεύτερο επίπεδο μπορούν να προστεθούν επιπλέον βιβλιοθήκες που χρειάζονται αλλά δεν παρέχονται από το Profile.



Εικόνα 5.1 – Τα τρία επίπεδα λειτουργίας της Java.

Το παρακάτω σχήμα δείχνει ένα πιο γενικό διαχωρισμό που περιλαμβάνει και τις τρεις κατηγορίες εφαρμογών Java.



Εικόνα 5.2 – Οι τρεις κατηγορίες εφαρμογών Java

5.2 CLDC

Το CLDC προέρχεται από τα αρχικά των λέξεων **Connected Limited Device Configuration,** που σημαίνει διαμόρφωση συσκευών με περιορισμένη συνδεσιμότητα και καθορίζει το βασικό σύνολο των κλάσεων του πυρήνα της Java και των διασυνδέσεων καθώς και την εικονική μηχανή που χρησιμοποιείται για συσκευές με περιορισμένους πόρους όπως είναι τα κινητά τηλέφωνα, τα μπίπερ, τα PDA κ.α. Το CLDC περιέχει 69 κλάσεις και 15 διασυνδέσεις (Interfaces), σύνολο το οποίο είναι απαραίτητο για την λειτουργία της εικονικής μηχανής. Όταν το CLDC συνδυαστεί με το MIDP παρέχει μία σταθερή πλατφόρμα της Java για την ανάπτυξη εφαρμογών που τρέχουν σε συσκευές με περιορισμένη μνήμη, επεξεργαστική ισχύ και δυνατότητες γραφικών.

Το CLDC αναπτύχθηκε από την κοινότητα της Java σε συνεργασία με πάνω από 500 συνεργάτες που αντιπροσώπευαν εταιρίες ασύρματων μικροσυσκευών, φορείς παροχής υπηρεσιών και τελικά σημεία πώλησης. Περιγράφει το βασικότερο σύνολο βιβλιοθηκών και χαρακτηριστικών γνωρισμάτων της εικονικής μηχανής της Java που πρέπει να υπάρχει σε κάθε μικροεφαρμογή J2ME στις κινητές συσκευές.

Το CLDC σχεδιάστηκε με σκοπό να παρέχει πολλά από τα πλεονεκτήματα της Java σε συσκευές όπου οι διαθέσιμοι πόροι είναι περιορισμένοι. Στις συσκευές αυτές περιλαμβάνονται τα κινητά τηλέφωνα, τα PDA, τα μπίπερ και διάφορα άλλα όπου υπάρχει περιορισμός στην επεξεργαστική δύναμη, στην μνήμη και στην απόδοση γραφικών.

Το CLDC σχεδιάστηκε για να λειτουργεί σε συσκευές με τα παρακάτω χαρακτηριστικά:

- Επεξεργαστής: Εύρος 16bit και άνω, συχνότητα λειτουργίας 16 MHz και άνω.
- Μνήμη: 160-512 KB διαθέσιμη επί της συνολικής.
- Ενέργεια: Περιορισμένη ενέργεια, συνήθως συσκευές που λειτουργούν με μπαταρία.
- Συνδεσιμότητα σε δίκτυο: Σύνδεση με μερικά μόνο από τα είδη των δικτύων τα οποία έχουν συνήθως χαμηλό εύρος μετάδοσης δεδομένων.

Το CLDC καθορίζει τις βιβλιοθήκες και τα χαρακτηριστικά της εικονικής μηχανής της Java για εφαρμογές J2ME. Οι εφαρμογές CLDC περιλαμβάνουν κάποια εικονική μηχανή της Java όπως για παράδειγμα την KVM (K Virtual Machine – Εικονική Μηχανή K). Οι εφαρμογές J2ME είναι εφαρμογές οι οποίες εκτελούνται σε κινητές μικροσυσκευές.

Μία εικονική μηχανή της Java περιέχει ένα σύνολο από βιβλιοθήκες που χρησιμεύουν για να τρέχουν οι εφαρμογές. Οι βιβλιοθήκες αυτές ενσωματώνονται ως μέρος στο CLDC μαζί με την εικονική μηχανή. Επίσης οι δυνατότητες των εφαρμογών J2ME μπορούν να επεκταθούν με την προσθήκη διαφόρων Profiles όπως είναι για παράδειγμα το MIDP, το οποίο περιγράφεται παρακάτω.

Η εικονική μηχανή KVM (K Virtual Machine) είναι μία εικονική μηχανή της Java που παρέχει την βάση για να εκτελούνται εφαρμογές J2ME. Το γράμμα 'K' προέρχεται από την λέξη Kilobyte που δηλώνει το πολύ μικρό μέγεθος την μηχανής. Οι απαιτήσεις της σε μνήμη ξεκινάει μόλις από τα 70KB. Εφαρμογές που γράφονται για να τρέχουν από την KVM έχουν την δυνατότητα να εκτελούνται και από άλλες εικονικές μηχανές της Java.

Η γλώσσα Java δεν εκτελείται από τον επεξεργαστή με τον ίδιο τρόπο που εκτελούνται εφαρμογές που είναι γραμμένες σε άλλες γλώσσες. Μία εφαρμογή σε γλώσσα C για παράδειγμα, μεταγλωττίζεται και οι εντολές της εκτελούνται σχεδόν άμεσα από τον επεξεργαστή. Η Java όμως εκτελείται διαφορετικά. Ο επεξεργαστής εκτελεί την εικονική μηχανή ως πρόγραμμα και στην συνέχεια η εικονική μηχανή είναι αυτή που εκτελεί την εφαρμογή της Java, δηλαδή είναι ένας εικονικός επεξεργαστής αποκλειστικά για αυτές τις εφαρμογές. Με άλλα λόγια η εικονική μηχανή μεσολαβεί ανάμεσα στην εφαρμογή και τον επεξεργαστή. Αυτό κάνει τις εφαρμογές να εκτελούνται πιο αργά απ' ότι αν ήταν γραμμένες σε άλλη γλώσσα, αλλά δίνει το πλεονέκτημα ότι μία τέτοια εφαρμογή μπορεί να τρέξει σε οποιοδήποτε σύστημα ανεξάρτητα από το λειτουργικό του.

Το CLDC είναι ένα θεμελιώδες κομμάτι της αρχιτεκτονικής J2ME (Java 2 Platform, Micro Edition). Η τεχνολογία J2ME είναι ένα περιβάλλον εφαρμογής το οποίο περιλαμβάνει διαμορφώσεις όπως το CLDC και προφίλ όπως το MIDP. Επιπλέον μπορεί να περιλαμβάνει προαιρετικά πακέτα τα οποία παρέχουν συγκεκριμένες λειτουργίες όπως ασύρματη μετάδοση μηνυμάτων, καταγραφή πολυμέσων και αναπαραγωγή ήχου. Η δυνατότητα επιλογής διαφόρων συνδυασμών μεταξύ πακέτων, διαμορφώσεων και προφίλ, επιτρέπει στους προγραμματιστές να προσαρμόζουν τις δυνατότητες των εφαρμογών, ανάλογα με το υλικό.

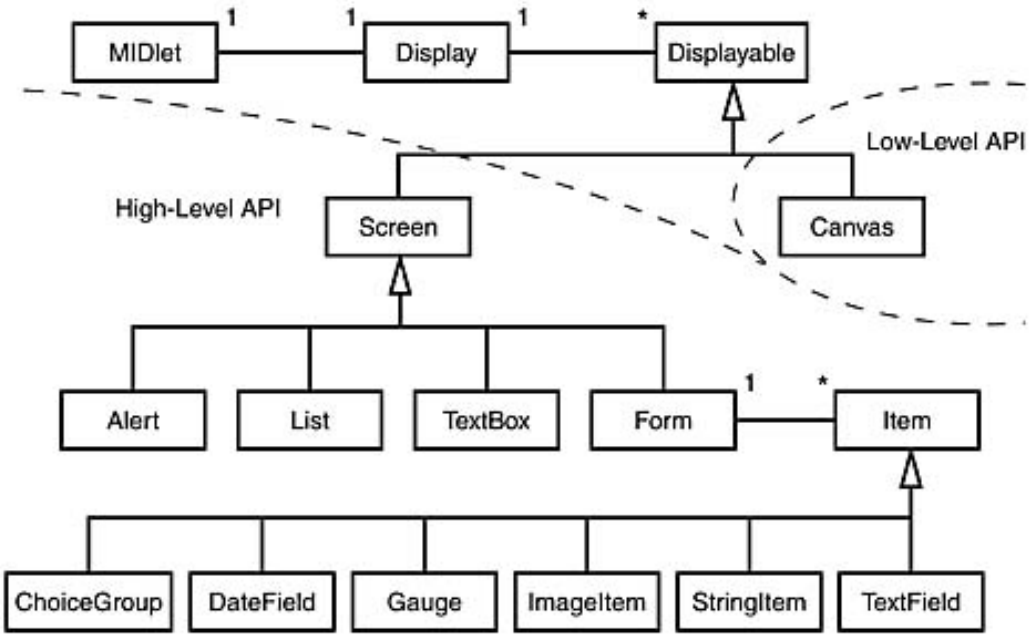
5.3 MIDP

Το **MIDP** προέρχεται από τα αρχικά των λέξεων **Mobile Information Device Profile** που σημαίνει προφίλ πληροφοριών για κινητές συσκευές. Είναι ένα στοιχείο του περιβάλλοντος ανάπτυξης μικροεφαρμογών της Java (J2ME), το οποίο όταν συνδυαστεί με το CLDC παρέχει ένα τυποποιημένο περιβάλλον εκτέλεσης κώδικα σε κινητές μικροσυσκευές όπως είναι τα κινητά τηλέφωνα, τα PDA κ.α. Παρέχει διάφορες βιβλιοθήκες οι οποίες επεκτείνουν τις δυνατότητες του CLDC και δίνουν την δυνατότητα να εκτελούνται στις συσκευές διάφορες λειτουργίες υψηλού επιπέδου όπως αποθήκευση και δικτύωση. Αυτές οι βιβλιοθήκες είναι χρήσιμες για τις εφαρμογές οι οποίες θέλουν να χρησιμοποιήσουν αυτές τις λειτουργίες.

Οι προδιαγραφές του MIDP καθορίστηκαν από μία ομάδα εμπειρογνωμόνων από περισσότερες από 50 εταιρίες, συμπεριλαμβανομένων κορυφαίων κατασκευαστών κινητών συσκευών. Καθορίστηκε να είναι μία πλατφόρμα που θα παρέχει δυναμική και ασφαλή αποδοχή και βελτιστοποίηση γραφικών και δικτυακών εφαρμογών. Έχει υιοθετηθεί ευρέως ως επιλεγμένη πλατφόρμα για την εκτέλεση εφαρμογών σε εκατομμύρια συνολικά κινητές συσκευές. Οι επιχειρήσεις σε όλον τον κόσμο έχουν ήδη εκμεταλλευτεί το MIDP, γράφοντας μία μεγάλη σειρά από μικροεφαρμογές που προορίζονται για τους πελάτες τους, αλλά και για τις ίδιες τις εταιρίες.

Το MIDP διαιρείται σε δύο μέρη: Χαμηλού επιπέδου API και υψηλού επιπέδου API.
Το χαμηλού επιπέδου API χρησιμοποιείται στην ανάπτυξη παιχνιδιών και γενικά σε εφαρμογές όπου απαιτούνται μη τυποποιημένα σχήματα στην οθόνη. Παρέχεται μεγαλύτερος έλεγχος της οθόνης μέχρι και επιπέδου pixel. Τα υψηλού επιπέδου API περιλαμβάνουν τυποποιημένα σχήματα όπως λίστες, Textbox, forms κτλ. Οι φόρμες έχουν μία σειρά από αντικείμενα όπως Textfield, Stringitem κτλ. τα οποία μπορούν να ενσωματωθούν σ' αυτήν ώστε να υπάρχει ενιαία διαχείριση των αντικειμένων. Στα τυποποιημένα σχήματα δεν παρέχεται έλεγχος για το πώς θα εμφανιστούν στην οθόνη. Απλά γίνεται αυτόματη προσαρμογή ανάλογα με την οθόνη που εμφανίζονται.

Το παρακάτω σχήμα δείχνει τα αντικείμενα του MIDP καθώς και τον διαχωρισμό τους σε υψηλού και χαμηλού επιπέδου API.



Εικόνα 5.3 – Τα αντικείμενα που περιλαμβάνονται στο MIDP

Όλες οι συσκευές έχουν ένα αντικείμενο που λέγεται Display το οποίο χρησιμοποιείται για την διαχείριση της οθόνης. Οι περισσότερες εφαρμογές έχουν μόνο ένα στιγμιότυπο του Display και κάθε νέα οθόνη που δημιουργείται από μία κλάση, παρουσιάζεται στον χρήστη καλώντας την μέθοδο setCurrent του αντικειμένου Display.

Οι προγραμματιστές που χρησιμοποιούν MIDP μπορούν να γράψουν εφαρμογές μία φορά και να τις προσαρμόσουν κατάλληλα ώστε να λειτουργούν σε μία μεγάλη ποικιλία κινητών μικροσυσκευών διαφόρων ειδών. Δεν χρειάζεται δηλαδή να γράψουν μία ίδια εφαρμογή, η οποία προορίζεται π.χ. για κινητά και PDA, δύο φορές. Τις εφαρμογές αυτές μπορεί κάποιος να τις κατεβάσει ασύρματα στην συσκευή του και να τις εκτελέσει απευθείας.

Το MIDP έχει περάσει στην δεύτερη έκδοσή του που ονομάζεται απλά MIDP 2.0. Η δεύτερη έκδοση παρέχει μία σειρά από νέα χαρακτηριστικά όπως μία πιο ενισχυμένη διασύνδεση με τον χρήστη, λειτουργικότητα παιχνιδιών και πολυμέσων, λειτουργία ασύρματης διασύνδεσης και εφαρμογή χαρακτηριστικών γνωρισμάτων ασφάλεια απ' άκρο εις άκρον στην επικοινωνία μεταξύ των συσκευών.

Το MIDP 2.0 παρέχει την δυνατότητα στους προγραμματιστές να προσαρμόσουν καλύτερα την διάταξη που θα εμφανίζεται η εφαρμογή, ανάλογα με τα χαρακτηριστικά της συσκευής, όπως για παράδειγμα το μέγεθος της οθόνης. Αυτό δίνει μεγαλύτερη ευελιξία στις εφαρμογές ώστε να εμφανίζονται σωστά ανεξάρτητα από την κινητή συσκευή που προορίζονται. Το MIDP 2.0 περιλαμβάνει επίσης το Audio Building Block (ABB) το οποίο δίνει την δυνατότητα στις μικροεφαρμογές να διαχειρίζονται κομμάτια ήχου σε μορφή WAV ή σε μορφή ακολουθίας τόνων. Επίσης έχει προστεθεί ένα Game API το οποίο δίνει την δυνατότητα ανάπτυξης παιχνιδιών

για τις μικροσυσκευές. Το MIDP 2.0 παρέχει επίσης υποστήριξη για τα κύρια πρότυπα συνδεσιμότητας πέρα από το HTTP, όπως είναι για παράδειγμα το HTTPS, τα datagrams, τα sockets κ.α. Μία ακόμα δυνατότητα είναι η αναβάθμιση των εφαρμογών μέσω ασύρματης διασύνδεσης, δίνοντας στον πάροχο υπηρεσιών την δυνατότητα να προσδιορίσει ποιες αναβαθμίσεις θα λειτουργήσουν στην συγκεκριμένη συσκευή. Τέλος έχει προστεθεί ένα πρότυπο ασφαλείας απ' άκρο εις άκρον το οποίο προστατεύει το δίκτυο, τις εφαρμογές και τις πληροφορίες που μεταδίδονται στις κινητές συσκευές. Υποστηρίζει το HTTPS και τα πρότυπα SSL και WTLS ώστε να είναι δυνατή η μετάδοση κρυπτογραφημένων δεδομένων. Το MIDP 2.0 προστατεύει ενάντια σε μη εξουσιοδοτημένη πρόσβαση στα δεδομένα, στις εφαρμογές και σε άλλους πόρους του δικτύου και των συσκευών.

ΚΕΦΑΛΑΙΟ 6 - ΚΩΔΙΚΑΣ ΑΠΟΣΤΟΛΗΣ ΚΑΙ ΛΗΨΗΣ ΜΗΝΥΜΑΤΩΝ

6.1 Υλοποίηση κώδικα - Εισαγωγή

Σε αυτήν την πτυχιακή υλοποιείται κώδικας σε java που κάνει κρυπτογράφηση στα SMS και στα MMS που στέλνονται μεταξύ των κινητών τηλεφώνων. Για την δημιουργία του κώδικα χρησιμοποιείται η εφαρμογή Netbeans 5.5 και επιπλέον ένα πρόσθετο πακέτο με όνομα Netbeans Mobility Pack 5.5 που επιτρέπει την εκτέλεση εντολών J2ME. Για την προσομοίωση χρησιμοποιείται το Wireless Toolkit 2.5.2.

6.2 Προγράμματα που χρησιμοποιούνται για την υλοποίηση

Αρχικά χρειάζονται τα παρακάτω προγράμματα τα οποία βρίσκονται στο Διαδίκτυο.

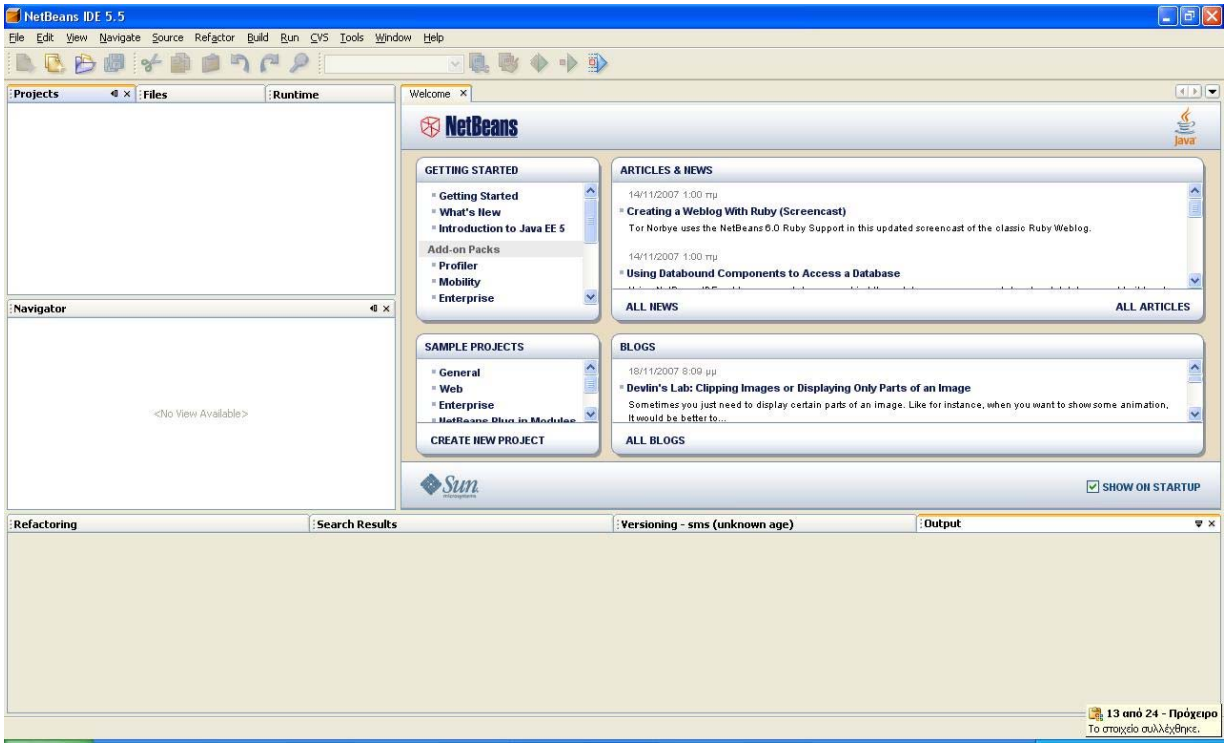
1. http://java.sun.com/javase/downloads/index.jsp
Σε αυτήν την διεύθυνση βρίσκεται το JDK 6 το οποίο επιτρέπει την εκτέλεση κώδικα java στον υπολογιστή. Είναι ένα εκτελέσιμο αρχείο μεγέθους 179,34 MB το οποίο εγκαθίσταται στον υπολογιστή και επιτρέπει να εκτελούνται τα εκτελέσιμα αρχεία jar που παράγονται από τον πηγαίο κώδικα της java.

2. http://www.netbeans.info/downloads/all.php?b_id=2323
Σ' αυτό το site υπάρχει η εφαρμογή Netbeans 5.5 που επιτρέπει την επεξεργασία του κώδικα. Το αρχείο αυτό έχει μέγεθος 54,7 MB και επιτρέπει την εγκατάσταση της πλατφόρμας ανάπτυξης εφαρμογών σε γλώσσα προγραμματισμού Java. Επιτρέπει την επεξεργασία και την ανάπτυξη του κώδικα καθώς και την εκτέλεσή του.

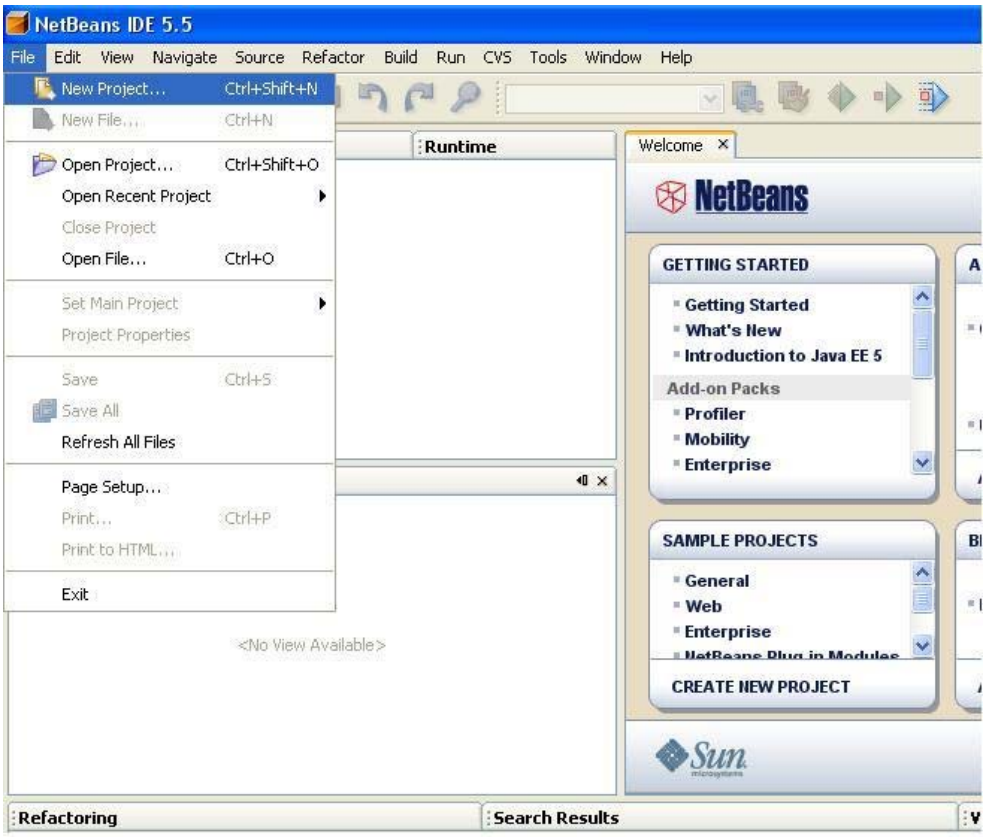
3. http://www.netbeans.info/downloads/start.php?f_id=13708&lang_id=1
Σε αυτό το link βρίσκεται το πακέτο Netbeans Mobility Pack 5.5, το οποίο εγκαθίσταται πάνω στο πρόγραμμα Netbeans 5.5 και χάρη στις βιβλιοθήκες που έχει, δίνει την δυνατότητα της εκτέλεσης του κώδικα στα κινητά. Το μέγεθος του ανέρχεται στα 23,5 MB.

4. http://java.sun.com/products/sjwtoolkit/download.html
Στο site αυτό βρίσκεται ο προσομοιωτής Wireless Toolkit 2.5.2. Έχει μέγεθος 37,08 MB και είναι ένας προσομοιωτής κινητού τηλεφώνου, ο οποίος εμφανίζεται ως κινητό τηλέφωνο στην οθόνη του υπολογιστή και δίνει την δυνατότητα να εκτελείται κώδικας java που προορίζεται για τα πραγματικά κινητά.

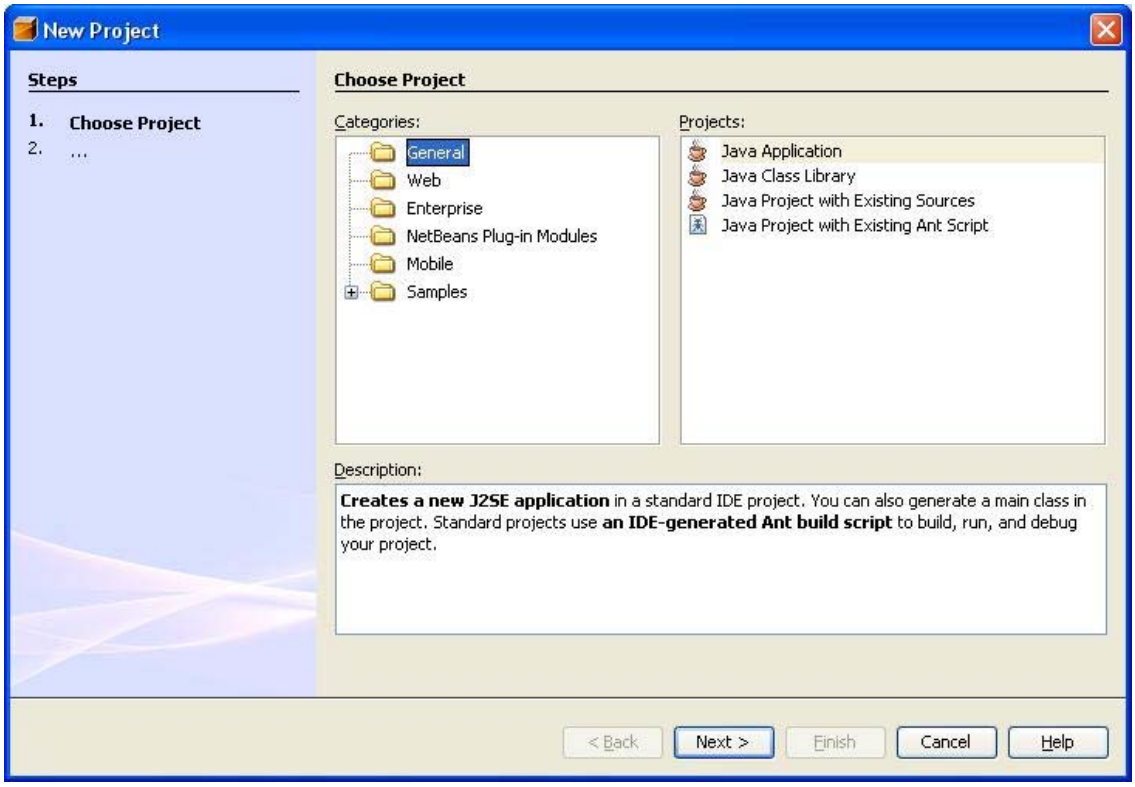
Τα τέσσερα αυτά προγράμματα πρέπει να εγκατασταθούν στο σύστημα με την σειρά που παρατίθενται για να λειτουργήσουν σωστά. Στην αρχή πρέπει να βρεθεί κώδικας ο οποίος να εκτελείται στον προσομοιωτή Wireless Toolkit και να πραγματοποιεί αποστολή και λήψη SMS και MMS. Τέτοιος κώδικας υπάρχει στα Netbeans και βρίσκεται στα έτοιμα project της εφαρμογής. Παρακάτω υπάρχουν στιγμιότυπα τα οποία δείχνουν ακριβώς πως φορτώνεται το συγκεκριμένο project.



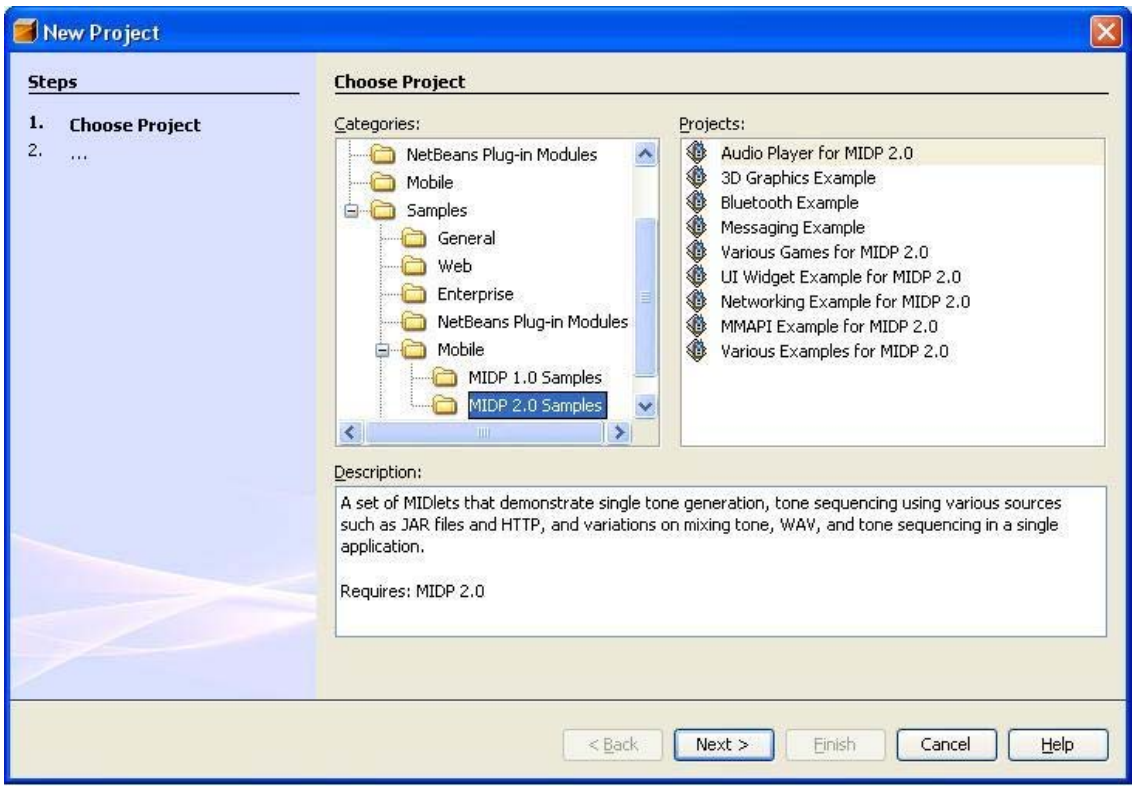
Εικόνα 6.1 - Εισαγωγή στο πρόγραμμα των Netbeans (το περιβάλλον που παρουσιάζεται κατά την έναρξη του προγράμματος).



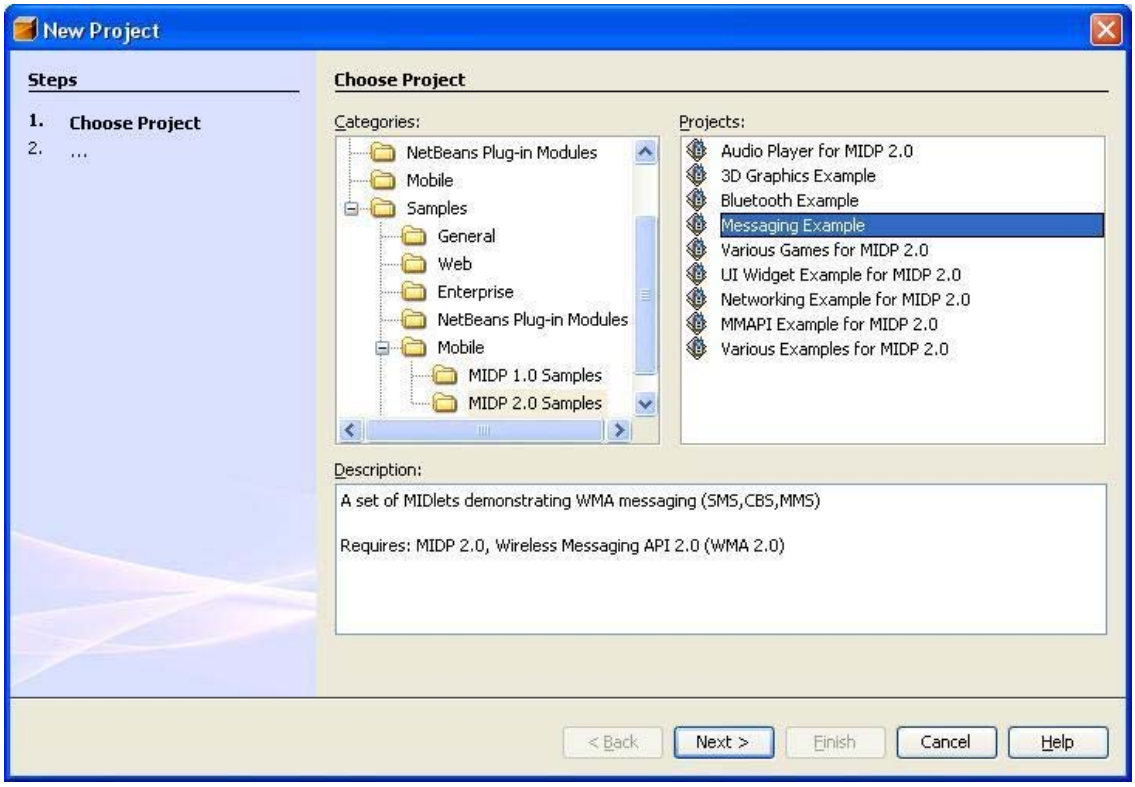
Εικόνα 6.2 - Επιλογή File→New Project για το άνοιγμα νέας εργασία.



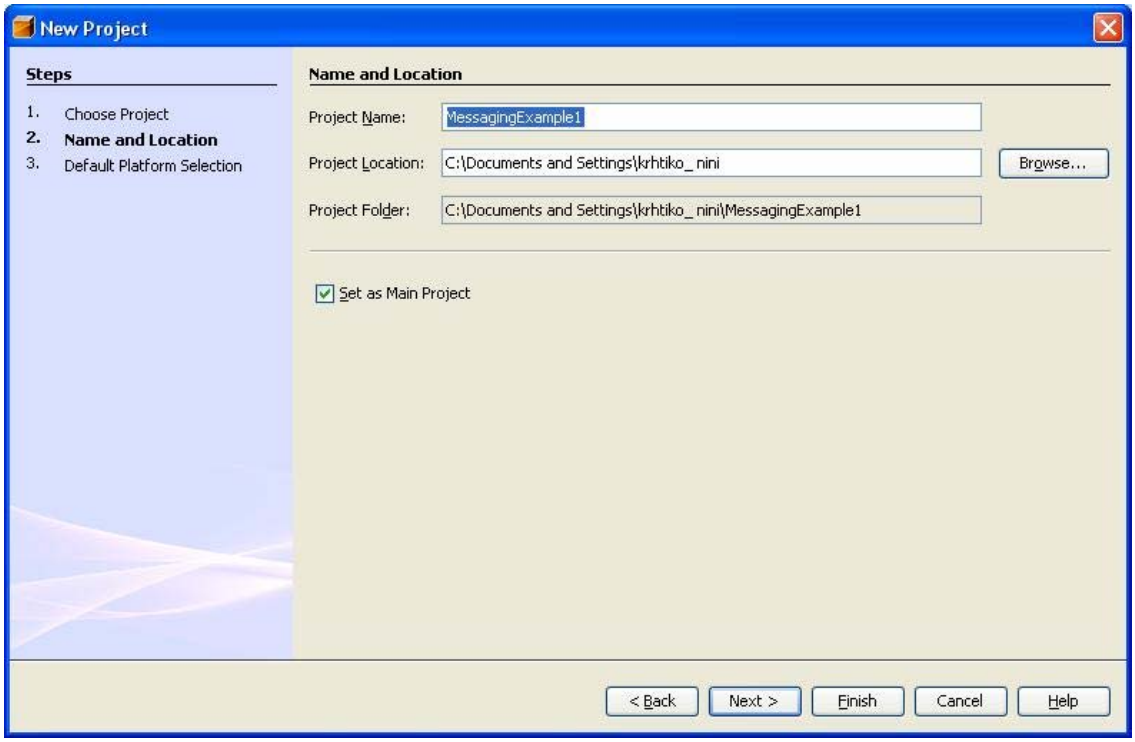
Εικόνα 6.3 - Παράθυρο που εμφανίζεται μετά την επιλογή File→New Project.



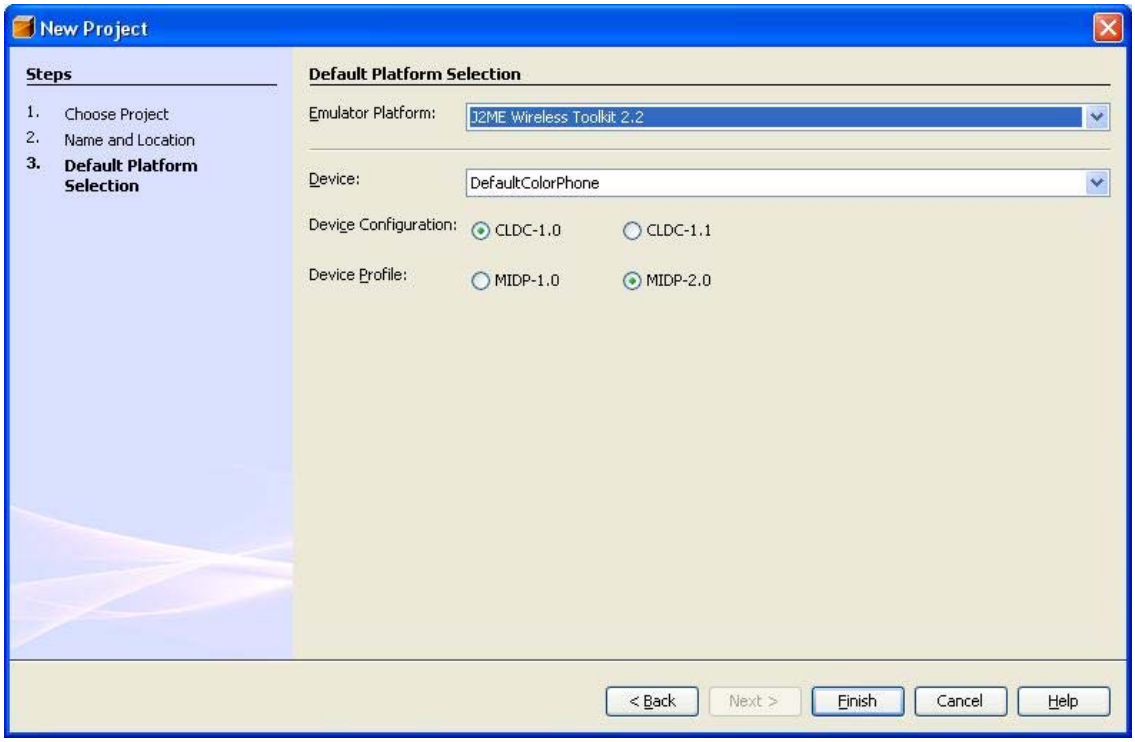
Εικόνα 6.4 - Ανάπτυξη των υποφακέλων και εμφάνιση των Project που περιέχει ο φάκελος MIDP 2.0 Samples.



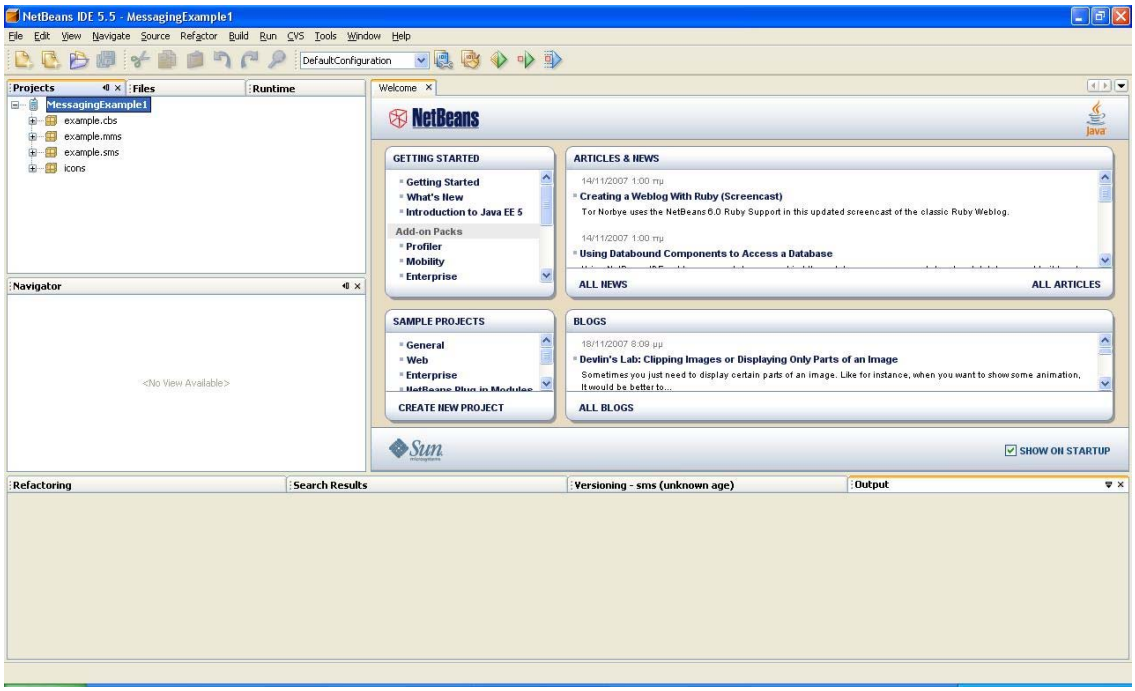
Εικόνα 6.5 - Επιλογή του Project με όνομα Messaging Example από την δεξιά στήλη και στην συνέχεια πάτημα του κουμπιού "Next".



Εικόνα 6.6 - Στο επόμενο παράθυρο ορίζεται το όνομα και η διαδρομή του Project και στην συνέχεια πάλι "Next".



Εικόνα 6.7 - Το παράθυρο επιλογών του προσομοιωτή. Οι ρυθμίσεις μένουν ως έχουν και τέλος, πάτημα του κουμπιού "Finish".



Εικόνα 6.8 - Άνοιγμα του Project στο κεντρικό παράθυρο των Netbeans.

Εικόνα 6.9 – Κουμπί εκτέλεσης

Αυτό είναι το κουμπί εκτέλεσης του κώδικα. Η εκτέλεση μπορεί να γίνει και με το πάτημα του κουμπιού F6 από το πληκτρολόγιο. Κατά την διάρκεια της εκτέλεσης εμφανίζεται ο προσομοιωτής με την μορφή κινητού τηλεφώνου. Για να γίνει όμως η αποστολή και η λήψη χρειάζονται δύο κινητά, γι' αυτόν τον λόγο η εκτέλεση γίνεται δύο φορές. Τα κινητά εξαφανίζονται μόλις τερματιστεί η εκτέλεση.

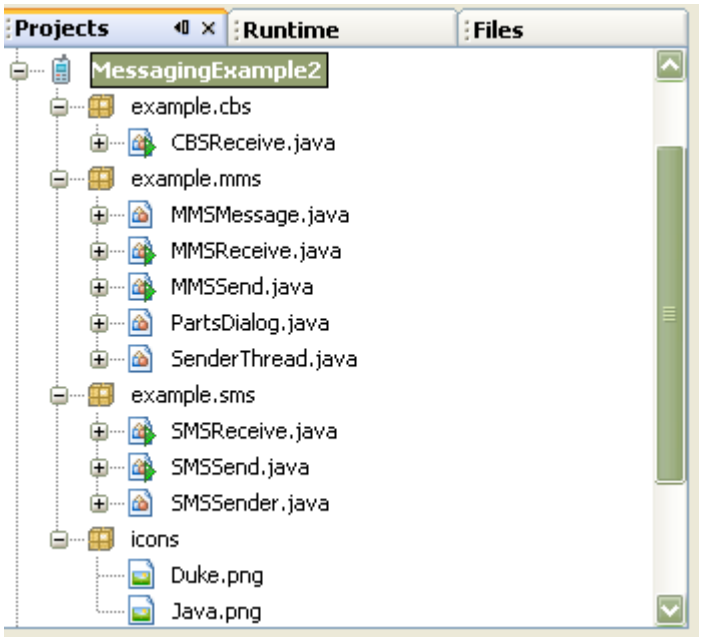
Στην ουσία, η εμφάνιση κινητών δεν έχει κάποιο συγκεκριμένο περιορισμό, δηλαδή μπορούν να εμφανιστούν τόσα κινητά όσες φορές πατηθεί το κουμπί εκτέλεσης των Netbeans. Επίσης, κάθε κινητό που εμφανίζεται έχει έναν μοναδικό αριθμό όπως γίνεται και με τα κανονικά κινητά. Ο αριθμός αυτός τα κάνει να ξεχωρίζουν μεταξύ τους και να λειτουργούν χωρίς πρόβλημα όσα κινητά-προσομοιωτές και να εκτελούνται την ίδια στιγμή. Την πρώτη φορά που εκτελείται ο κώδικας το κινητό έχει τον αριθμό +5550000 και κάθε φορά αυξάνεται κατά ένα, δηλαδή το δεύτερο θα έχει αριθμό +5550001, το τρίτο +5550002 κ.τ.λ.

Η εκτέλεση μιας μικροεφαρμογής παρουσιάζει πρόβλημα όταν υπάρχουν ελληνικοί χαρακτήρες στην διαδρομή του δίσκου όπου έχει αποθηκευτεί το project και στο όνομα του χρήστη του υπολογιστή.

6.3 Περιγραφή κώδικα αποστολής SMS και MMS.

Τα SMS είναι μικρά τμήματα κειμένου τα οποία ανταλλάσσονται μεταξύ χρηστών κινητών τηλεφώνων. Τα κείμενα αυτά μπορεί να περιέχουν γράμματα, αριθμούς και σύμβολα. Σ'αυτό το project πραγματοποιείται αποστολή και λήψη SMS και MMS. Το μήκος των μηνυμάτων SMS στο συγκεκριμένο project μπορούν να φτάσουν μέγιστο τους 65535 χαρακτήρες. Στην πραγματική αγορά όμως δεν ξεπερνούν τους 160 χαρακτήρες. Τα MMS μπορεί να περιέχουν τμήματα κειμένου ή εικόνες και έναν τίτλο σε μορφή κειμένου (θέμα) μέγιστου μήκους 40 χαρακτήρων.

6.4 Ανάλυση κλάσεων



Εικόνα 6.10 – Διάγραμμα πακέτων και κλάσεων

Αυτή είναι η δενδρική δομή του Project με όνομα MessagingExample που υπάρχει μέσα στα Netbeans Mobility Pack.

Το Project αποτελείται από 4 πακέτα με τα εξής ονόματα:

1. “example.cbs”
2. “example.mms”
3. “example.sms”
4. “icons”

Το πρώτο πακέτο περιέχει μία κλάση με το όνομα “CBSReceive”. Είναι δευτερεύουσας σημασίας και δεν θα χρησιμοποιηθεί καθόλου και γι’ αυτό δεν θα αναφερθεί παρακάτω. Το δεύτερο πακέτο περιέχει πέντε κλάσεις οι οποίες χρησιμοποιούνται για την αποστολή και λήψη MMS. Οι δύο από αυτές είναι τύπου MIDLET, δηλαδή μπορούν να εκτελεστούν από τον προσομοιωτή ως αρχικές εκτελέσιμες εφαρμογές και επίσης εμφανίζονται και στην αρχική οθόνη του κινητού. Στην παραπάνω εικόνα ξεχωρίζουν από το μικρό πράσινο βελάκι που υπάρχει πάνω στο εικονίδιο του αρχείου.

Τα ονόματα των κλάσεων έχουν ως εξής:

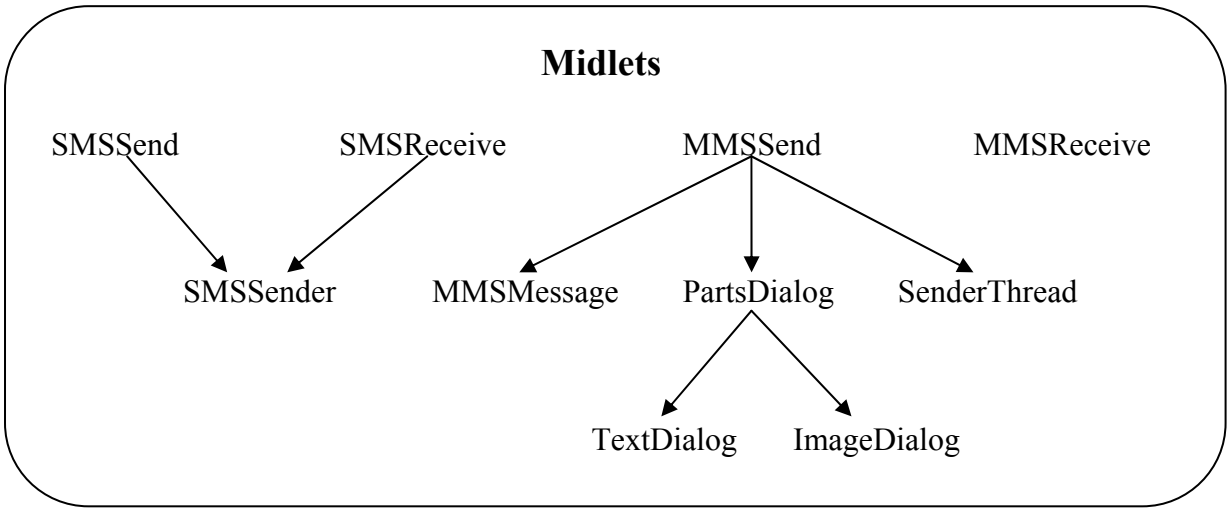
1. “MMSMessage”
2. “MMSReceive” (Midlet)
3. “MMSSend” (Midlet)
4. “PartsDialog”
5. “SenderThread”

Το τρίτο πακέτο περιέχει τρεις κλάσεις η οποίες χρησιμοποιούνται για την αποστολή και λήψη SMS. Όπως και στην περίπτωση των MMS, οι δύο από αυτές τις κλάσεις είναι τύπου MIDLET και ως εκ' τούτου εμφανίζονται στην αρχική οθόνη του προσομοιωτή ως εκτελέσιμες εφαρμογές. Τα ονόματα τους είναι τα παρακάτω:

1. "SMSReceive" (Midlet)
2. "SMSSend" (Midlet)
3. "SMSSender"

Το τέταρτο και τελευταίο πακέτο περιέχει δύο αρχεία τύπου "png" τα οποία χρησιμοποιούνται ως εικόνες στην εφαρμογή των MMS. Υπάρχει η δυνατότητα να προστεθούν και άλλα αρχεία του ίδιου τύπου ώστε να χρησιμοποιηθούν για αποστολή στα MMS.

Στο παρακάτω σχήμα παρουσιάζεται ο τρόπος με τον οποίο γίνονται οι κλήσεις των κλάσεων.



Εικόνα 6.11 – Κλήσεις των κλάσεων στο αρχικό project

6.5 Κανόνες περιγραφής κώδικα

Η εναλλαγή των χρωμάτων έχουν σκοπό στο να ξεχωρίζουν κάποιου είδους "ομάδες" που υπάρχουν μέσα στην κλάση. Ομάδες εννοούνται τα imports, τα σχόλια και οι μέθοδοι.

Οποιαδήποτε γραμμή ξεκινάει με "//" είναι σχόλιο. Σχόλιο είναι επίσης και οποιοδήποτε κομμάτι κώδικα το οποίο στην αρχή του έχει "/*" και στο τέλος του "*/".

Τα μαύρα γράμματα προστέθηκαν έπειτα και σκοπός τους είναι να περιγράφουν τις εντολές που βρίσκονται στο πάνω μέρος τους. Σε αυτές τις περιγραφές τα ονόματα των κλάσεων θα ξεκινούν με κεφαλαία ενώ τα ονόματα των μεθόδων με μικρά και θα

ακολουθούνται από άνοιγμα και κλείσιμο παρένθεσης. π.χ. Κλάση, μέθοδος(). Σε διαφορετικές κλάσεις μπορούν να υπάρχουν μέθοδοι με το ίδιο όνομα.

Τα παραπάνω ισχύουν για όλο τον κώδικα που περιγράφεται εδώ μέσα.

6.6 Πακέτο example.sms

Οι κλάσεις αυτού του πακέτου πραγματοποιούν αποστολή και λήψη SMS. Θα γίνει η ανάλυση αυτού του πακέτου πρώτα για λόγους απλότητας.

6.6.1 Κλάση SMSSend

Είναι το ένα από τα δύο MIDLET που υπάρχει σε αυτό το πακέτο και κατά την εκτέλεση του ξεκινάει την αποστολή ενός μηνύματος SMS προς κάποιον υποψήφιο παραλήπτη.

```
package example.sms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.midlet.*;
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
import java.io.IOException;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
/**
An example MIDlet to send text via an SMS MessageConnection
*/
```

```
public class SMSSend extends MIDlet
   implements CommandListener {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “extends” σημαίνει ότι η κλάση είναι υποκλάση της MIDlet και ως εκ’ τούτου κληρονομεί όλες τις μεθόδους της. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή.

```
    /** user interface command for indicating Exit request. */
    Command exitCommand  = new Command("Exit", Command.EXIT, 2);
    /** user interface command for proceeding to the next screen */
    Command okCommand = new Command("OK", Command.OK, 1);
    /** current display. */
    Display display;
    /** The port on which we send SMS messages */
    String smsPort;
    /** Area where the user enters the phone number to send the message to */
    TextBox destinationAddressBox;
    /** Error message displayed when an invalid phone number is entered */
    Alert errorMessageAlert;
    /** Alert that is displayed when a message is being sent */
    Alert sendingMessageAlert;
    /** Prompts for and sends the text message */
    SMSSender sender;
    /** The last visible screen when we paused */
    Displayable resumeScreen = null;
```

Δηλώσεις διαφόρων μεταβλητών της κλάσης που φτιάχνονται κατά την εκκίνηση του MIDLET και σχόλια που περιγράφουν την κάθε μεταβλητή.

```
    /**
     * Initialize the MIDlet with the current display object and
     * graphical components.
     */


    public SMSSend() {
        smsPort = getAppProperty("SMS-Port");

        display = Display.getDisplay(this);

        destinationAddressBox = new TextBox("Destination Address?",
            null, 256, TextField.PHONENUMBER);
        destinationAddressBox.addCommand(exitCommand);
        destinationAddressBox.addCommand(okCommand);
        destinationAddressBox.setCommandListener(this);

        errorMessageAlert = new Alert("SMS", null, null, AlertType.ERROR);
        errorMessageAlert.setTimeout(5000);

        sendingMessageAlert = new Alert("SMS", null, null, AlertType.INFO);
        sendingMessageAlert.setTimeout(5000);
        sendingMessageAlert.setCommandListener(this);
        sender = new SMSSender(smsPort, display, destinationAddressBox,
            sendingMessageAlert);
        resumeScreen = destinationAddressBox;
    }
```

Constructor(δημιουργός) της κλάσης. Εκτελείται με την έναρξη του MIDLET. Μετά την εκτέλεση εμφανίζεται στην οθόνη του προσομοιωτή ένα πεδίο κειμένου όπου δίνεται ο αριθμός του παραλήπτη. Στην προ-τελευταία εντολή καλείται ο Constructor της κλάσης "SMSSender".

```
/**
 * startApp should return immediately to keep the dispatcher
 * from hanging.
 */


public void startApp() {
   display.setCurrent(resumeScreen);
}

/**
 * Remember what screen is showing
 */


public void pauseApp() {
   resumeScreen = display.getCurrent();
}


/**
 * Destroy must cleanup everything.
 * @param unconditional true if a forced shutdown was requested
 */


public void destroyApp(boolean unconditional) {
}
```

Οι τρεις παραπάνω μέθοδοι βρίσκονται πάντα μέσα σε μία κλάση MIDLET. Ορίζονται εντολές που εκτελούνται κατά την έναρξη, παύση και τερματισμό του MIDLET.

```
/**
 * Respond to commands, including exit
 * @param c user interface command requested
 * @param s screen object initiating the request
 */
```

```
public void commandAction(Command c, Displayable s) {
      try {
         if (c == exitCommand || c == Alert.DISMISS_COMMAND) {
            destroyApp(false);
            notifyDestroyed();
         } else if (c == okCommand) {
            promptAndSend();
         }
      } catch (Exception ex) {
         ex.printStackTrace();
      }
   }
```

Μέθοδος που υπάρχει μέσα στην διασύνδεση CommandListener και ορίζεται εδώ. Ορίζονται οι εντολές που θα εκτελεστούν κατά το πάτημα των κουμπιών “Exit” και “Ok”.

```
   /**
    * Prompt for and send the message
    */


   private void promptAndSend() {
      String address = destinationAddressBox.getString();
      if (!SMSSend.isValidPhoneNumber(address)) {
         errorMessageAlert.setString("Invalid phone number");
         display.setCurrent(errorMessageAlert, destinationAddressBox);
         return;
      }
      String statusMessage = "Sending message to " + address + "...";
      sendingMessageAlert.setString(statusMessage);
      sender.promptAndSend("sms://" + address);
   }
```

Εδώ ελέγχεται ο αριθμός του αποστολέα και αν είναι σωστός τότε καλείται η μέθοδος promptAndSend() που βρίσκεται στην κλάση SMSSender για να αποσταλεί το μήνυμα.

```
   /**
    * Check the phone number for validity
    * Valid phone numbers contain only the digits 0 thru 9, and may contain
    * a leading '+'.
    */
```

```
    private static boolean isValidPhoneNumber(String number) {
        char[] chars = number.toCharArray();
        if (chars.length == 0) {
            return false;
        }
        int startPos = 0;
        // initial '+' is OK
        if (chars[0] == '+') {
            startPos = 1;
        }
        for (int i = startPos; i < chars.length; ++i) {
            if (!Character.isDigit(chars[i])) {
                return false;
            }
        }
        return true;
    }
}
```

Αυτή είναι η μέθοδος που ελέγχει τον αριθμό του παραλήπτη. Ο αριθμός είναι σωστός μόνο όταν αποτελείται από αριθμητικά ψηφία με εξαίρεση το πρώτο στη σειρά όπου θα μπορούσε να είναι και το σύμβολο “+”.

6.6.2 Κλάση SMSSender

Είναι η μία από τις τρεις κλάσεις μέσα στο πακέτο που δεν είναι MIDLET. Χρησιμοποιείται για την ολοκλήρωση της αποστολής και της λήψης των SMS και καλείται από τις δύο άλλες μεθόδους του ίδιου πακέτου.

Ακολουθεί ο κώδικας.

```
package example.sms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
import java.io.IOException;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
/**
 * Prompts for text and sends it via an SMS MessageConnection
 */
```

```
public class SMSSender
    implements CommandListener, Runnable {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή.

Η διασύνδεση Runnable επιτρέπει να εκτελεστεί η μέθοδος run() της κλάσης αυτής ως ξεχωριστό νήμα εκτέλεσης.

Γενικά, κατά την έναρξη κάποιου νήματος, εκτελείται ο κώδικας που βρίσκεται μέσα στην μέθοδο run() παράλληλα και ανεξάρτητα από την υπόλοιπη εφαρμογή. Μέσα σε μία εφαρμογή μπορούν να ξεκινήσουν πολλά νήματα ταυτόχρονα και ένα συγκεκριμένο νήμα μπορεί να ξεκινήσει και να εκτελείται πολλές φορές την ίδια στιγμή. Αυτό επιτρέπει την εκτέλεση εντολών στο παρασκήνιο χωρίς να εμποδίζεται η ομαλή εκτέλεση κάποιας άλλης εργασίας που δουλεύει στο προσκήνιο. Σε αυτό το project χρησιμοποιούνται νήματα στην αποστολή και λήψη των SMS και MMS.

```
    /** user interface command for indicating Send request */
    Command sendCommand = new Command("Send", Command.OK, 1);
    /** user interface command for going back to the previous screen */
    Command backCommand = new Command("Back", Command.BACK, 2);
    /** Display to use. */
    Display display;
    /** The port on which we send SMS messages */
    String smsPort;
    /** The URL to send the message to */
    String destinationAddress;
    /** Area where the user enters a message to send */
    TextBox messageBox;
    /** Where to return if the user hits "Back" */
    Displayable backScreen;
    /** Displayed when a message is being sent */
    Displayable sendingScreen;
```

Διάφορες μεταβλητές.

```
    /**
     * Initialize the MIDlet with the current display object and
     * graphical components.
     */
```

```
public SMSSender(String smsPort, Display display,
      Displayable backScreen, Displayable sendingScreen) {
      this.smsPort = smsPort;
      this.display = display;
      this.destinationAddress = null;
      this.backScreen = backScreen;
      this.sendingScreen = sendingScreen;

      messageBox = new TextBox("Enter Message", null, 65535, TextField.ANY);
      messageBox.addCommand(backCommand);
      messageBox.addCommand(sendCommand);
      messageBox.setCommandListener(this);
   }
```

Constructor της κλάσης. Καλείται από τον Constructor της SMSSend και δημιουργεί (δεν εμφανίζει ακόμα) ένα πεδίο κειμένου όπου γράφεται το μήνυμα που θα σταλεί. Αυτό γίνεται μετά την εισαγωγή του αριθμού του παραλήπτη και εφόσον ο αριθμός αυτός είναι έγκυρος.

```
   /**
    * Prompt for message and send it
    */


  public void promptAndSend(String destinationAddress)
      {
         this.destinationAddress = destinationAddress;
         display.setCurrent(messageBox);
      }
```

Καλείται από την συνόνομη μέθοδο της κλάσης SMSSend μετά την εισαγωγή του αριθμού του παραλήπτη και τον έλεγχο του για εγκυρότητα. Αυτό που κάνει είναι να εμφανίζει το πεδίο κειμένου για την εισαγωγή του μηνύματος που θα σταλεί στον παραλήπτη.

```
/**
    * Respond to commands, including exit
    * @param c user interface command requested
    * @param s screen object initiating the request
    */


   public void commandAction(Command c, Displayable s) {
      try {
         if (c == backCommand) {
            display.setCurrent(backScreen);
         } else if (c == sendCommand) {
            display.setCurrent(sendingScreen);
```

```
            new Thread(this).start();
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

Κατά την εμφάνιση του πεδίου εισαγωγής του κειμένου προστίθενται επίσης δύο εντολές κουμπιών, το “Back” και το “Send”. Εδώ ορίζονται οι εντολές τους. Όταν πατηθεί το κουμπί “Back” εμφανίζεται η προηγούμενη οθόνη, δηλαδή το πεδίο εισαγωγής του αριθμού αποστολής. Το κουμπί “Send” ξεκινάει την εκτέλεση της μεθόδου run ως ξεχωριστό νήμα εκτέλεσης.

```
/**
* Send the message. Called on a separate thread so we don't have
* contention for the display
*/


public void run() {
    String address = destinationAddress + ":" + smsPort;
    MessageConnection smsconn = null;
    try {
        /** Open the message connection. */
        smsconn = (MessageConnection)Connector.open(address);
        TextMessage txtmessage = (TextMessage)smsconn.newMessage(
            MessageConnection.TEXT_MESSAGE);
        txtmessage.setAddress(address);
        txtmessage.setPayloadText(messageBox.getString());
        smsconn.send(txtmessage);
    } catch (Throwable t) {
        System.out.println("Send caught: ");
        t.printStackTrace();
    }
    if (smsconn != null) {
        try {
            smsconn.close();
        } catch (IOException ioe) {
            System.out.println("Closing connection caught: ");
            ioe.printStackTrace();
        }
    }
}
}
```

Η λειτουργία της αποστολής που γίνεται σε αυτήν την μέθοδο έχει ως εξής: Για να γίνει η αποστολή δημιουργείται πρώτα ένα αντικείμενο τύπου MessageConnection, το οποίο ανοίγει μία σύνδεση με τον παραλήπτη και του στέλνει τα δεδομένα. Για να ανοίξει η σύνδεση δίνεται ο αριθμός του παραλήπτη και η πόρτα επικοινωνίας για τα

SMS. Εδώ η πόρτα έχει αριθμό 50000. Η συνολική διεύθυνση για το άνοιγμα της σύνδεσης είναι: “sms://αριθμός παραλήπτη//50000”, όπου αριθμός παραλήπτη αυτός που δίνεται νωρίτερα στο πεδίο κειμένου εισαγωγής αριθμού. Στη συνέχεια δημιουργείται ένα αντικείμενο τύπου TextMessage. Αυτό το αντικείμενο θα σταλεί στη συνέχεια από το MessageConnection αφού πρώτα ενσωματωθεί το μήνυμα που θα σταλεί και η διεύθυνση του παραλήπτη. Τέλος, στέλνεται το TextMessage με την μέθοδο send και στην συνέχεια κλείνει η σύνδεση με την μέθοδο close(). Οι δύο αυτές μέθοδοι βρίσκονται στην κλάση MessageConnection.

6.6.3 Κλάση SMSReceive

```
package example.sms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.midlet.*;
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
import java.io.IOException;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
/**
 * An example MIDlet displays text from an SMS MessageConnection
 */
```

```
public class SMSReceive extends MIDlet
    implements CommandListener, Runnable, MessageListener {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “extends” σημαίνει ότι η κλάση είναι υποκλάση της MIDlet και ως εκ’ τούτου κληρονομεί όλες τις μεθόδους της. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή. Η διασύνδεση Runnable επιτρέπει να εκτελεστεί η μέθοδος run() της κλάσης αυτής ως ξεχωριστό νήμα εκτέλεσης.

Γενικά, κατά την έναρξη κάποιου νήματος, εκτελείται ο κώδικας που βρίσκεται μέσα στην μέθοδο run() παράλληλα και ανεξάρτητα από την υπόλοιπη εφαρμογή. Μέσα σε μία εφαρμογή μπορούν να ξεκινήσουν πολλά νήματα ταυτόχρονα και ένα συγκεκριμένο νήμα μπορεί να ξεκινήσει και να εκτελείται πολλές φορές την ίδια στιγμή. Αυτό επιτρέπει την εκτέλεση εντολών στο παρασκήνιο χωρίς να εμποδίζεται

η ομαλή εκτέλεση κάποιας άλλης εργασίας που δουλεύει στο προσκήνιο. Σε αυτό το project χρησιμοποιούνται νήματα στην αποστολή και λήψη των SMS και MMS.

Η διασύνδεση MessageListener επιτρέπει την δημιουργία ενός αντικειμένου τύπου Listener το οποίο τοποθετείται σε μία συγκεκριμένη πόρτα και δίνει την δυνατότητα λήψης και επεξεργασίας εισερχόμενων μηνυμάτων που στέλνονται σε αυτήν την πόρτα.

```
/** user interface command for indicating Exit request. */
Command exitCommand  = new Command("Exit", Command.EXIT, 2);
/** user interface command for indicating Reply request */
Command replyCommand = new Command("Reply", Command.OK, 1);
/** user interface text box for the contents of the fetched URL. */
Alert content;
/** current display. */
Display display;
/** instance of a thread for asynchronous networking and user interface. */
Thread thread;
/** Connections detected at start up. */
String[] connections;
/** Flag to signal end of processing. */
boolean done;
/** The port on which we listen for SMS messages */
String smsPort;
/** SMS message connection for inbound text messages. */
MessageConnection smsconn;
/** Current message read from the network. */
Message msg;
/** Address of the message's sender */
String senderAddress;
/** Alert that is displayed when replying */
Alert sendingMessageAlert;
/** Prompts for and sends the text reply */
SMSSender sender;
/** The screen to display when we return from being paused */
Displayable resumeScreen;
```

Διάφορες μεταβλητές.

```
/**
 * Initialize the MIDlet with the current display object and
 * graphical components.
 */
```

```
public SMSReceive() {
    smsPort = getAppProperty("SMS-Port");

    display = Display.getDisplay(this);

    content = new Alert("SMS Receive");
    content.setTimeout(Alert.FOREVER);
    content.addCommand(exitCommand);
    content.setCommandListener(this);
    content.setString("Receiving...");

    sendingMessageAlert = new Alert("SMS", null, null, AlertType.INFO);
    sendingMessageAlert.setTimeout(5000);
    sendingMessageAlert.setCommandListener(this);

    sender = new SMSSender(smsPort, display, content, sendingMessageAlert);

    resumeScreen = content;
}
```

Constructor της κλάσης. Δημιουργείται η οθόνη που θα εμφανίζεται κατά την αναμονή λήψης ενός μηνύματος SMS. Επίσης κατασκευάζεται και ένα αντικείμενο τύπου SMSSender.

```
/**
* Start creates the thread to do the MessageConnection receive
* text.
* It should return immediately to keep the dispatcher
* from hanging.
*/


public void startApp() {
    /** SMS connection to be read. */
    String smsConnection = "sms://:" + smsPort;
    /** Open the message connection. */
    if (smsconn == null) {
        try {
            smsconn = (MessageConnection) Connector.open(smsConnection);
            smsconn.setMessageListener(this);
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
    /** Initialize the text if we were started manually. */
    connections = PushRegistry.listConnections(true);
    if (connections == null || connections.length == 0) {
        content.setString("Waiting for SMS on port " + smsPort + "...");
    }
```

```
        done = false;
        thread = new Thread(this);
        thread.start();

        display.setCurrent(resumeScreen);
    }
```

Εντολές που εκτελούνται κατά την έναρξη του MIDLET. Εδώ τοποθετείται ένας MessageListener ο οποίος εισακούει για εισερχόμενα μηνύματα στην πόρτα 50000 και εμφανίζεται η οθόνη αναμονής στον προσομοιωτή. Στην συνέχεια ξεκινάει την μέθοδο run() ως νήμα για να ελέγξει αν υπάρχουν εισερχόμενα μηνύματα.

```
    /**
    * Notification that a message arrived.
    * @param conn the connection with messages available
    */


    Public void notifyIncomingMessage(MessageConnection conn) {
        if (thread == null) {
            done = false;
            thread = new Thread(this);
            thread.start();
        }
    }
```

Αυτή η μέθοδος εκτελείται όταν φτάσει ένα εισερχόμενο μήνυμα SMS. Εδώ ξεκινάει την μέθοδο run() ως νήμα, ώστε να γίνει η λήψη και η περαιτέρω επεξεργασία του μηνύματος.

```
    public void run() {
        /** Check for sms connection. */
        try {
            msg = smsconn.receive();
            if (msg != null) {
                senderAddress = msg.getAddress();
                content.setTitle("From: " + senderAddress);
                if (msg instanceof TextMessage) {
                    content.setString(((TextMessage)msg).getPayloadText());
                } else {
                    StringBuffer buf = new StringBuffer();
                    byte[] data = ((BinaryMessage)msg).getPayloadData();
                    for (int i = 0; i < data.length; i++) {
                        int intData = (int)data[i] & 0xFF;
                        if (intData < 0x10) {
                            buf.append("0");
                        }
                        buf.append(Integer.toHexString(intData));
```

```
                buf.append(' ');
            }
            content.setString(buf.toString());
        }
        content.addCommand(replyCommand);
        display.setCurrent(content);
    }
  } catch (IOException e) {
      e.printStackTrace();
  }
}
```

Είναι η μέθοδος που εκτελείται ως νήμα. Λαμβάνει το εισερχόμενο μήνυμα με την κλήση της μεθόδου receive() και στην συνέχεια το εμφανίζει στην οθόνη του προσομοιωτή προσθέτοντας επίσης και δύο εντολές κουμπιών, την “Reply” που χρησιμοποιείται για απάντηση στον αποστολέα και την “Exit” για έξοδο.

```
/**
 * Pause signals the thread to stop by clearing the thread field.
 * If stopped before done with the iterations it will
 * be restarted from scratch later.
 */


public void pauseApp() {
  done = true;
  thread = null;
  resumeScreen = display.getCurrent();
}


/**
 * Destroy must cleanup everything.  The thread is signaled
 * to stop and no result is produced.
 * @param unconditional true if a forced shutdown was requested
 */


public void destroyApp(boolean unconditional) {
  done = true;
  thread = null;
  if (smsconn != null) {
    try {
      smsconn.close();
    } catch (IOException e) {
      // Ignore any errors on shutdown
    }
  }
}
```

Μέθοδος που εκτελείται κατά τον τερματισμό του MIDLET. Μηδενίζει το νήμα και κλείνει την σύνδεση εάν αυτή έχει μείνει ανοικτή.

```
    /**
     * Respond to commands, including exit
     * @param c user interface command requested
     * @param s screen object initiating the request
     */


    public void commandAction(Command c, Displayable s) {
        try {
            if (c == exitCommand || c == Alert.DISMISS_COMMAND) {
                destroyApp(false);
                notifyDestroyed();
            } else if (c == replyCommand) {
                reply();
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
```

Ορισμός εντολών για τα κουμπιά “Exit” και “Reply”. Στην τελευταία εκτελείται η μέθοδος reply()(απάντηση) της παρούσας κλάσης.

```
    /**
     * Allow the user to reply to the received message
     */


    private void reply() {
        // remove the leading "sms://" for diplaying the destination address
        String address = senderAddress.substring(6);
        String statusMessage = "Sending message to " + address + "...";
        sendingMessageAlert.setString(statusMessage);
        sender.promptAndSend(senderAddress);
    }
}
```

Καλείται από το πάτημα του κουμπιού “Reply” μετά από λήψη κάποιου μηνύματος. Η μέθοδος αυτή επιτρέπει να σταλλεί απάντηση σε αυτόν που έστειλε το μήνυμα. Με την εκτέλεση της μεθόδου αυτής εμφανίζεται η οθόνη εισαγωγής μηνύματος όπου γράφεται το κείμενο προς αποστολή και στη συνέχεια με το πάτημα του κουμπιού “Send”, αποστέλλεται αυτό το μήνυμα στον αποστολέα του μηνύματος που λήφθηκε προηγουμένως.

Εδώ τελειώνει η περιγραφή του κώδικα που εκτελείται για την αποστολή και λήψη SMS.

6.7 Πακέτο “example.mms”

Το πακέτο αυτό έχει πέντε κλάσεις οι οποίες πραγματοποιούν αποστολή και λήψη MMS. Λόγω της μεγαλύτερης πολυπλοκότητας των MMS, οι κλάσεις και ο κώδικας είναι περισσότεροι απ ‘ότι στο προηγούμενο πακέτο που περιγράφεται παραπάνω.

6.7.1 Κλάση MMSSend

Η κλάση αυτή είναι τύπου MIDLET και επιτρέπει την αποστολή μηνυμάτων MMS.

```
package example.mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.midlet.*;
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
/**
 * An example MIDlet to send text via an MMS MessageConnection
 */
```

```
public class MMSSend extends MIDlet
    implements CommandListener {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “extends” σημαίνει ότι η κλάση είναι υποκλάση της MIDlet και ως εκ’ τούτου κληρονομεί όλες τις μεθόδους της. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή.

```
    /** user interface command for indicating Exit request. */
    private static Command CMD_EXIT  = new Command("Exit", Command.EXIT,
2);
    /** user interface command for sending the message */
    private static Command CMD_SEND = new Command("Send", Command.ITEM,
1);
    /** user interface command for adding message's part */
    private static Command CMD_ADD_PART =
            new Command("Add Part", Command.ITEM, 1);

    /** current display. */
    private Display display;
    /** The application-ID on which we send MMS messages */
    private String appID;
    /** Area where the user enters the subject of the message */
    private TextField subjectField;
    /** Area where the user enters the phone number to send the message to */
    private TextField destinationField;
    /** Area where the user enters the phone number to send the message to */
    private StringItem partsLabel;
    /** Error message displayed when an invalid phone number is entered */
    private Alert errorMessageAlert;
    /** Alert that is displayed when a message is being sent */
    private Alert sendingMessageAlert;
    /** The last visible screen when we paused */
    private Displayable resumeScreen = null;
    private MMSMessage message;
    private PartsDialog partsDialog;
```

Διάφορες μεταβλητές.

```
    public MMSSend() {
        appID = getAppProperty("MMS-ApplicationID");

        display = Display.getDisplay(this);

        Form mainForm = new Form("New MMS");

        subjectField = new TextField("Subject:",
            null, 256, TextField.ANY);
        mainForm.append(subjectField);

        destinationField = new TextField("Destination Address: ",
            "mms://", 256, TextField.ANY);
        mainForm.append(destinationField);

        partsLabel = new StringItem("Parts:", "0");
        mainForm.append(partsLabel);
```

```
        mainForm.addCommand(CMD_EXIT);
        mainForm.addCommand(CMD_SEND);
        mainForm.addCommand(CMD_ADD_PART);
        mainForm.setCommandListener(this);

        errorMessageAlert = new Alert("MMS", null, null, AlertType.ERROR);
        errorMessageAlert.setTimeout(5000);

        sendingMessageAlert = new Alert("MMS", null, null, AlertType.INFO);
        sendingMessageAlert.setTimeout(5000);
        sendingMessageAlert.setCommandListener(this);
        resumeScreen = mainForm;
        message = new MMSMessage();
    }
```

Constructor της κλάσης. Δημιουργείται η κεντρική οθόνη όπου μπορούν να προστεθούν μέρη(εικόνες και τμήματα κειμένου) στο μήνυμα, να οριστεί τίτλος θέματος(subject) και να δοθεί ο αριθμός του παραλήπτη. Επίσης, φτιάχνεται και ένα αντικείμενο της κλάσης MMSMessage όπου θα ενσωματωθούν όλα τα παραπάνω χαρακτηριστικά του μηνύματος και στην συνέχεια θα σταλούν στον παραλήπτη.

```
    /**
     * startApp should return immediately to keep the dispatcher
     * from hanging.
     */


    public void startApp() {
        display.setCurrent(resumeScreen);
    }
```

Εμφάνιση της κεντρικής οθόνης κατά την έναρξη του MIDLET.

```
    public void pauseApp() {
        resumeScreen = display.getCurrent();
    }


    public void destroyApp(boolean unconditional) {
    }
```

Οι τρεις παραπάνω μέθοδοι βρίσκονται πάντα μέσα σε μία κλάση MIDLET. Ορίζονται εντολές που εκτελούνται κατά την έναρξη, παύση και τερματισμό του MIDLET.

```
    /** Respond to commands, including exit
     * @param c user interface command requested
     * @param s screen object initiating the request*/
```

```
public void commandAction(Command c, Displayable s) {
   try {
      if ((c == CMD_EXIT) || (c == Alert.DISMISS_COMMAND)) {
         destroyApp(false);
         notifyDestroyed();
      } else if (c == CMD_ADD_PART) {
               if (partsDialog == null) {
                  partsDialog = new PartsDialog(this);
               }
               partsDialog.show();
      } else if (c == CMD_SEND) {
         promptAndSend();
      }
   } catch (Exception ex) {
      ex.printStackTrace();
   }
}
```

Ορισμός των εντολών κατά το πάτημα κουμπιών. Εδώ οι εντολές εμφανίζονται στην κεντρική οθόνη της αποστολής του MMS. Οι εντολές “Add Part” και “Send” εμφανίζονται μαζί σε ένα μενού. Η εντολή “Add Part” φτιάχνει ένα καινούργιο αντικείμενο της κλάσης PartDialog, ενώ η εντολή “send” καλεί την μέθοδο promptAndSend() που βρίσκεται παρακάτω στην παρούσα κλάση.

```
void show() {
        partsLabel.setText(Integer.toString(partsDialog.counter));
        display.setCurrent(resumeScreen);
}
```

Η κλήση της παραπάνω μεθόδου επαναφέρει την αρχική κεντρική οθόνη αποστολής μετά από εισαγωγή εικόνας ή τμήματος κειμένου. Επίσης, εμφανίζεται και ο αριθμός των συνολικών μερών που έχουν προστεθεί στο μήνυμα μέχρι εκείνη την στιγμή.

```
Display getDisplay() {
        return display;
}


MMSMessage getMessage() {
        return message;
}
```

Μέθοδοι οι οποίες επιστρέφουν τις μεταβλητές display και message αντίστοιχα.

```
/**
 * Prompt for and send the message
 */
```

```
private void promptAndSend() {
      try {
         String address = destinationField.getString();
         message.setSubject(subjectField.getString());
         message.setDestination(address);
         String statusMessage = "Sending message to " + address + "...";
         sendingMessageAlert.setString(statusMessage);
         new SenderThread(message, appID).start();
      } catch (IllegalArgumentException iae) {
         errorMessageAlert.setString(iae.getMessage());
         display.setCurrent(errorMessageAlert);
      }
   }
}
```

Αυτή η μέθοδος καλείται όταν πατηθεί το κουμπί με την εντολή “Send”. Στην μεταβλητή message ενσωματώνεται το subject και ο αριθμός του παραλήπτη και στην συνέχεια ξεκινάει ένα νέο νήμα της κλάσης SenderThread, το οποίο αποστέλλει το μήνυμα.

6.7.2 Κλάση MMSMessage

Η κλάση αυτή είναι στην ουσία το μήνυμα MMS. Στις μεταβλητές της κλάσης αυτής αποθηκεύονται τα μέρη του μηνύματος, ο αριθμός του παραλήπτη και ο τίτλος του θέματος(subject). Στη συνέχεια το αντικείμενο αυτό στέλνεται στον παραλήπτη με την βοήθεια της κλάσης SenderThread.

```
package example.mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import java.util.Vector;
import javax.wireless.messaging.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class MMSMessage {
```

Έναρξη της κλάσης.

```
   private String destination;
   private Vector parts = new Vector();
   private String subject;
```

Διάφορες μεταβλητές.

```
/**
     * Check the phone number for validity
     * Valid phone numbers contain only the digits 0 thru 9, and may contain
     * a leading '+'.*/


    private static boolean isValidPhoneNumber(String address) {
        String protocol = "mms://";
        if (!address.startsWith(protocol)) {
            return false;
        }
        String number = address.substring(protocol.length());
        char[] chars = number.toCharArray();
        if (chars.length == 0) {
            return false;
        }
        int startPos = 0;
        // initial '+' is OK
        if (chars[0] == '+') {
            startPos = 1;
        }
        for (int i = startPos; i < chars.length; ++i) {
            if (!Character.isDigit(chars[i])) {
                return false;
            }
        }
        return true;
    }
```

Μέθοδος που ελέγχει τον αριθμό του παραλήπτη για εγκυρότητα. Έγκυρος είναι ένας αριθμός ο οποίος περιέχει μόνο αριθμητικά ψηφία εκτός από το πρώτο το οποίο μπορεί να είναι και το σύμβολο “+”.

```
    public String getSubject() {
        return subject;
    }
```

Μέθοδος που επιστρέφει την τιμή της μεταβλητής subject.

```
    public void setSubject(String subject) {
        this.subject = subject;
    }
```

Μέθοδος που αποθηκεύει στην μεταβλητή subject τον τίτλο του θέματος.

```
    public String getDestination() {
        return destination;
    }
```

Μέθοδος που επιστρέφει τον αριθμό του παραλήπτη.

```
    public void setDestination(String destination) {
        if (!isValidPhoneNumber(destination)) {
            throw new IllegalArgumentException("Invalid phone number");
        }
        this.destination = destination;
    }
```

Μέθοδος που ελέγχει τον αριθμό του παραλήπτη για εγκυρότητα μέσω της κλήσης της μεθόδου isValidPhoneNumber() και αν είναι σωστός τότε τον αποθηκεύει στην μεταβλητή destination.

```
    public MessagePart[] getParts() {
        MessagePart[] partsArray = new MessagePart[parts.size()];
        parts.copyInto(partsArray);
        return partsArray;
    }
```

Μέθοδος που επιστρέφει όλα τα μέρη που υπάρχουν στο μήνυμα MMS.

```
    public void addPart(MessagePart part) {
        parts.addElement(part);
    }
}
```

Μέθοδος που προσθέτει ένα καινούργιο μέρος στο μήνυμα.

6.7.3 Κλάση PartsDialog

Η κλάση αυτή χρησιμοποιείται για την προσθήκη μερών στο μήνυμα MMS. Καλείται με την επιλογή της εντολής “Add part” στην κεντρική οθόνη αποστολής του MMS.

```
package example.mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
import java.io.InputStream;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class PartsDialog implements CommandListener {
   /** current display. */
   private MMSSend mmsSend;
   private List typeList;

   public int counter = 0;

   private final static Command CMD_BACK =
            new Command("Back", Command.BACK, 1);
   private final static Command CMD_NEXT = new Command("Next",
Command.OK, 1);
   private final static Command CMD_OK = new Command("OK", Command.OK,
1);
   private final static Command CMD_CANCEL =
            new Command("Cancel", Command.CANCEL, 1);
```

Διάφορες μεταβλητές.

```
  /** Creates a new instance of PartsDialog */

   public PartsDialog(MMSSend mmsSend) {
                this.mmsSend = mmsSend;

      String[] stringArray = {"Text", "Image"};

      typeList = new List("Add Part: Type", Choice.EXCLUSIVE,
                           stringArray, null);
      typeList.addCommand(CMD_BACK);
      typeList.addCommand(CMD_NEXT);
      typeList.setCommandListener(this);
   }
```

Constructor της κλάσης.

```
        public void show() {
                mmsSend.getDisplay().setCurrent(typeList);
        }
```

Μέθοδος που εμφανίζει την οθόνη της εικόνας που αναφέρεται πιο πάνω.

```
   /**Respond to commands, including exit
    * @param c user interface command requested
    * @param s screen object initiating the request*/
```

```
public void commandAction(Command c, Displayable s) {
   try {
      if (c == CMD_BACK) {
                              mmsSend.show();
      } else if (c == CMD_NEXT) {
         if (typeList.getSelectedIndex() == 0) {
                  mmsSend.getDisplay().setCurrent(new TextDialog());
         } else {
                  mmsSend.getDisplay().setCurrent(new ImageDialog());
         }
      }
   } catch (Exception ex) {
      ex.printStackTrace();
   }
}
```

Χειρισμός των εντολών κατά το πάτημα των κουμπιών “Back” και “Next”.

```
private class TextDialog extends Form implements CommandListener {
```

Εσωτερική κλάση. Επιτρέπει την εισαγωγή μερών κειμένου μέσα στο MMS.

```
   private Displayable mainForm;
   private TextField text;
   private String mimeType = "text/plain";
```

Διάφορες μεταβλητές.

```
   public TextDialog() {
      super("Add Text");

      text = new TextField("Text: ", null, 256, TextField.ANY);
      append(text);
      append("MIME-Type: " + mimeType);

      addCommand(CMD_OK);
      addCommand(CMD_CANCEL);
      setCommandListener(this);
   }
```

Constructor της κλάσης. Επιτρέπει την εισαγωγή κειμένου για να εισαχθεί ως μέρος στο MMS.

```
public void commandAction(Command c, Displayable s) {
        try {
           if (c == CMD_OK) {
              String encoding = "UTF-8";
              byte[] contents = text.getString().getBytes(encoding);
              mmsSend.getMessage().addPart(
                        new MessagePart(contents, 0, contents.length,
                                          mimeType, "id" + counter,
                                           "contentLocation", encoding));
              counter++;
                                       mmsSend.show();
           } else if (c == CMD_CANCEL) {
                                       mmsSend.show();
           }
        } catch (Exception ex) {
           ex.printStackTrace();
        }
     }
  }
```

Ορισμός των εντολών “Ok” και “Cancel”. Με το “Ok” ολοκληρώνεται η εισαγωγή του κειμένου στο μήνυμα.

```
   private class ImageDialog extends Form implements CommandListener {
```

Εσωτερική κλάση όπως και η προηγούμενη μόνο που εδώ εισάγεται εικόνα στο μήνυμα.

```
      private Displayable mainForm;
      private ChoiceGroup cg;
      private String mimeType = "image/png";
      private String[] resouces = {"/icons/Java.png", "/icons/Duke.png"};
      private String[] imagesNames = {"Java", "Duke"};
```

Διάφορες μεταβλητές. Οι εικόνες διατίθενται στο τέταρτο πακέτο.

```
      public ImageDialog() {
         super("Add Image");
         cg = new ChoiceGroup("Select Image", Choice.EXCLUSIVE,
                                 imagesNames, null);
         append(cg);
         append("MIME-Type: " + mimeType);
         addCommand(CMD_OK);
         addCommand(CMD_CANCEL);
         setCommandListener(this);
      }
```

Constructor της κλάσης. Επιτρέπει την επιλογή μίας εικόνας ώστε να εισαχθεί ως μέρος στο μήνυμα.

```
    public void commandAction(Command c, Displayable s) {
       try {
          if (c == CMD_OK) {
             int index = cg.getSelectedIndex();
             String resouce = resouces[index];
             InputStream is = getClass().getResourceAsStream(resouce);
             byte[] contents = new byte[is.available()];
             is.read(contents);
             String contentLocation = imagesNames[index];
             mmsSend.getMessage().addPart(
                      new MessagePart(contents, 0, contents.length,
                                      mimeType, "id" + counter,
                                      contentLocation, null));
             counter++;
                                    mmsSend.show();
          } else if (c == CMD_CANCEL) {
                                    mmsSend.show();
          }
       } catch (Exception ex) {
          ex.printStackTrace();
       }
    }
  }
}
```

Ορισμός των εντολών “Ok” και “Cancel” που εμφανίζονται στην οθόνη επιλογής εικόνας. Με το “Ok” εισάγεται η εικόνα στο μήνυμα. Η εντολή “Cancel” και στην περίπτωση της εισαγωγής κειμένου στο MMS, πηγαίνει την οθόνη πίσω στην κεντρική εικόνα αποστολής χωρίς να γίνει προσθήκη κειμένου ή εικόνας.

6.7.4 Κλάση SenderThread

Η κλάση αυτή εκτελείται ως ξεχωριστό νήμα εκτέλεσης και πραγματοποιεί την αποστολή του MMS.

```
package example.mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import java.io.IOException;
import javax.microedition.io.*;
import javax.wireless.messaging.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class SenderThread extends Thread {
```

Το extend Thread δίνει την δυνατότητα στην κλάση να χρησιμοποιείται ως νήμα εκτέλεσης.

```
    private MMSMessage message;
    private String appID;
```

Διάφορες μεταβλητές.

```
    public SenderThread(MMSMessage message, String appID) {
        this.message = message;
        this.appID = appID;
    }
```

Constructor της κλάσης. Καλείται από την μέθοδο promptAndSend() της κλάσης MMSSend, η οποία περιγράφεται παραπάνω. Οι δύο παράμετροι είναι το μήνυμα MMS που θα σταλεί, και το αλφαριθμητικό “appID”. Το τελευταίο είναι στην ουσία η πόρτα της σύνδεσης που χρησιμοποιείται για την αποστολή του μηνύματος.

```
    /**
    * Send the message. Called on a separate thread so we don't have
    * contention for the display
    */


    public void run() {
        String address = message.getDestination() + ":" + appID;

        MessageConnection mmsconn = null;

        try {
            /** Open the message connection. */
            mmsconn = (MessageConnection) Connector.open(address);

            MultipartMessage mmmessage =
                    (MultipartMessage) mmsconn.newMessage(
                    MessageConnection.MULTIPART_MESSAGE);
            mmmessage.setAddress(address);

            MessagePart[] parts = message.getParts();

            for (int i = 0; i < parts.length; i++) {
                mmmessage.addMessagePart(parts[i]);
            }

            mmmessage.setSubject(message.getSubject());
```

```
            mmsconn.send(mmmessage);
        } catch (Exception e) {
            e.printStackTrace();
        }

        if (mmsconn != null) {
            try {
                mmsconn.close();
            } catch (IOException ioe) {
                ioe.printStackTrace();
            }
        }
    }
}
```

Η μέθοδος που εκτελείται κατά την έναρξη της συγκεκριμένης κλάσης ως νήμα. Πραγματοποιεί την αποστολή του MMS ακολουθώντας παρόμοια βήματα όπως στην αποστολή SMS που γίνεται με την μέθοδο run() της κλάσης SMSSender. Εδώ δημιουργείται ένα αντικείμενο της κλάσης MultipartMessage, στο οποίο ενσωματώνονται τα μέρη και το subject του μηνύματος, τα οποία υπάρχουν μέσα στην μεταβλητή message. Αυτό το αντικείμενο θα σταλεί στην συνέχεια ως το μήνυμα MMS. Για να σταλεί όμως θα πρέπει πρώτα να φτιαχτεί ένα αντικείμενο τύπου MessageConnection, το οποίο θα ανοίξει μία σύνδεση με τον παραλήπτη και θα στείλει στη συνέχεια το μήνυμα. Τέλος, κλείνει η σύνδεση.

6.7.5 Κλάση MMSReceive

Η κλάση αυτή περιέχει όλες τις μεθόδους που χρειάζονται για να γίνει σωστά η λήψη ενός μηνύματος MMS. Εκτελείται ως MIDLET από τον χρήστη-παραλήπτη και θέτει το κινητό σε κατάσταση αναμονής για να δεχτεί MMS.

```
package example.mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.midlet.*;
import javax.microedition.io.*;
import javax.microedition.lcdui.*;
import javax.wireless.messaging.*;
import java.io.IOException;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
/**
 * An example MIDlet displays text from an MMS MessageConnection
 */
```

```
public class MMSReceive extends MIDlet
   implements CommandListener, Runnable, MessageListener {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “extends” σημαίνει ότι η κλάση είναι υποκλάση της MIDlet και ως εκ’ τούτου κληρονομεί όλες τις μεθόδους της. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή. Η διασύνδεση Runnable επιτρέπει να εκτελεστεί η μέθοδος run() της κλάσης αυτής ως ξεχωριστό νήμα εκτέλεσης.

Η διασύνδεση MessageListener επιτρέπει την δημιουργία ενός αντικειμένου τύπου Listener το οποίο τοποθετείται σε μία συγκεκριμένη πόρτα και δίνει την δυνατότητα λήψης και επεξεργασίας εισερχόμενων μηνυμάτων που στέλνονται σε αυτήν την πόρτα.

```
   /** user interface command for indicating Exit request. */
   private static final Command CMD_EXIT  =
         new Command("Exit", Command.EXIT, 2);
   /** user interface text box for the contents of the fetched URL. */
   private Form content;
   /** current display. */
   private Display display;
   /** instance of a thread for asynchronous networking and user interface. */
   private Thread thread;
   /** Connections detected at start up. */
   private String[] connections;
   /** Flag to signal end of processing. */
   private boolean done;
   /** The applicationID on which we listen for MMS messages */
   private String appID;
   /** MMS message connection for inbound text messages. */
   private MessageConnection mmsconn;
   /** Current message read from the network. */
   private Message msg;
   /** Address of the message's sender */
   private String senderAddress;
   /** Alert that is displayed when replying */
   private Alert sendingMessageAlert;
   /** The screen to display when we return from being paused */
   private Displayable resumeScreen;
   /** The subject of the message received */
   private String subject;
   /** The text of the received message */
   private String contents;
```

Διάφορες μεταβλητές.

```
/**
 * Initialize the MIDlet with the current display object and
 * graphical components.
 */


public MMSReceive() {
    appID = getAppProperty("MMS-ApplicationID");

    display = Display.getDisplay(this);

    content = new Form("MMS Receive");
    content.addCommand(CMD_EXIT);
    content.setCommandListener(this);
    content.append("Receiving...");

    sendingMessageAlert = new Alert("MMS", null, null, AlertType.INFO);
    sendingMessageAlert.setTimeout(5000);
    sendingMessageAlert.setCommandListener(this);

    resumeScreen = content;
}
```

Constructor της κλάσης. Δημιουργεί την οθόνη που θα εμφανιστεί κατά την αναμονή λήψης MMS, καθώς και τα αντικείμενα που θα περιέχονται σε αυτήν, όπως εντολές κουμπιών.

```
/**
 * Start creates the thread to do the MessageConnection receive
 * text.
 * It should return immediately to keep the dispatcher
 * from hanging. */


public void startApp() {
    /** MMS connection to be read. */
    String mmsConnection = "mms://:" + appID;
    /** Open the message connection. */
    if (mmsconn == null) {
        try {
            mmsconn = (MessageConnection) Connector.open(mmsConnection);
            mmsconn.setMessageListener(this);
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
    /** Initialize the text if we were started manually. */
    connections = PushRegistry.listConnections(true);
    if (connections == null || connections.length == 0) {
```

```
            content.deleteAll();
            content.append("Waiting for MMS on applicationID " + appID + "...");
        }
        done = false;
        thread = new Thread(this);
        thread.start();

        display.setCurrent(resumeScreen);
    }
```

Μέθοδος που εκτελείται κατά την εκκίνηση του συγκεκριμένου MIDLET. Ανοίγει μία σύνδεση για λήψη MMS και θέτει MessageListener σε αυτήν ώστε να μπορεί να γίνει λήψη του μηνύματος. Επίσης, ξεκινάει την εκτέλεση του αντικειμένου ως νήμα και τέλος, εμφανίζει στον προσομοιωτή την οθόνη αναμονής.

```
    /**
     * Notification that a message arrived.
     * @param conn the connection with messages available
     */


    public void notifyIncomingMessage(MessageConnection conn) {
        if (thread == null && !done) {
            thread = new Thread(this);
            thread.start();
        }
    }
```

Μέθοδος που εκτελείται όταν υπάρχει εισερχόμενο μήνυμα MMS. Ξεκινάει την μέθοδο run() ως νήμα για να γίνει η λήψη και η επεξεργασία του μηνύματος.

```
    /** Message reading thread. */
    public void run() {
        /** Check for mms connection. */
        try {
            msg = mmsconn.receive();
            if (msg != null) {
                senderAddress = msg.getAddress();
                content.deleteAll();
                String titleStr = senderAddress.substring(6);
                int colonPos = titleStr.indexOf(":");
                if (colonPos != -1) {
                    titleStr = titleStr.substring(0, colonPos);
                }
                content.setTitle("From: " + titleStr);
                if (msg instanceof MultipartMessage) {
                    MultipartMessage mpm = (MultipartMessage)msg;
                    StringBuffer buff = new StringBuffer("Subject: ");
```

```
                buff.append((subject = mpm.getSubject()));
                buff.append("\nDate: ");
                buff.append(mpm.getTimestamp().toString());
                buff.append("\nContent:");
                StringItem messageItem = new StringItem("Message",
                                                        buff.toString());
                messageItem.setLayout(Item.LAYOUT_NEWLINE_AFTER);
                content.append(messageItem);
                MessagePart[] parts = mpm.getMessageParts();
                if (parts != null) {
                   for (int i = 0; i < parts.length; i++) {
                      buff = new StringBuffer();
                      MessagePart mp = parts[i];
                      buff.append("Content-Type: ");
                      String mimeType = mp.getMIMEType();
                      buff.append(mimeType);
                      String contentLocation = mp.getContentLocation();
                      buff.append("\nContent:\n");
                      byte[] ba = mp.getContent();
                      if (mimeType.equals("image/png")) {
                         content.append(buff.toString());
                         Image img = Image.createImage(ba, 0, ba.length);
                         ImageItem ii = new ImageItem(contentLocation,
                               img, Item.LAYOUT_NEWLINE_AFTER,
                               contentLocation);
                         content.append(ii);
                      } else {
                         buff.append(new String(ba));
                         StringItem si = new StringItem(
                                        "Part", buff.toString());
                         si.setLayout(Item.LAYOUT_NEWLINE_AFTER);
                         content.append(si);
                      }
                   }
                }
                display.setCurrent(content);
             }
          } catch (IOException e) {
             e.printStackTrace();
          }
       }
```

Κατά την εκτέλεση της μεθόδου αυτής, γίνεται στην αρχή η λήψη του μηνύματος MMS και στην συνέχεια η επεξεργασία του και η εμφάνιση του στην οθόνη του κινητού. Στην αρχή της οθόνης αυτής εμφανίζονται με την μορφή κειμένου ο τίτλος θέματος (subject) του μηνύματος και η ημερομηνία. Στην συνέχεια ακολουθούν τα μέρη του μηνύματος τα οποία διατάσσονται το ένα κάτω από το άλλο με την ίδια σειρά που προστέθηκαν από τον αποστολέα. Επίσης εμφανίζεται και ο αριθμός του αποστολέα στον τίτλο της οθόνης. Όλα αυτά γίνονται από την παραπάνω μέθοδο.

```
    /**
     * Pause signals the thread to stop by clearing the thread field.
     * If stopped before done with the iterations it will
     * be restarted from scratch later.
     */


    public void pauseApp() {
        done = true;
        thread = null;
        resumeScreen = display.getCurrent();
    }


     /**
     * Destroy must cleanup everything.  The thread is signaled
     * to stop and no result is produced.
     * @param unconditional true if a forced shutdown was requested
     */


    public void destroyApp(boolean unconditional) {
        done = true;
        thread = null;
        if (mmsconn != null) {
            try {
                mmsconn.close();
            } catch (IOException e) {
                // Ignore any errors on shutdown
            }
        }
    }
```

Μέθοδος που εκτελείται κατά τον τερματισμό του MIDLET η οποία κλείνει την σύνδεση.

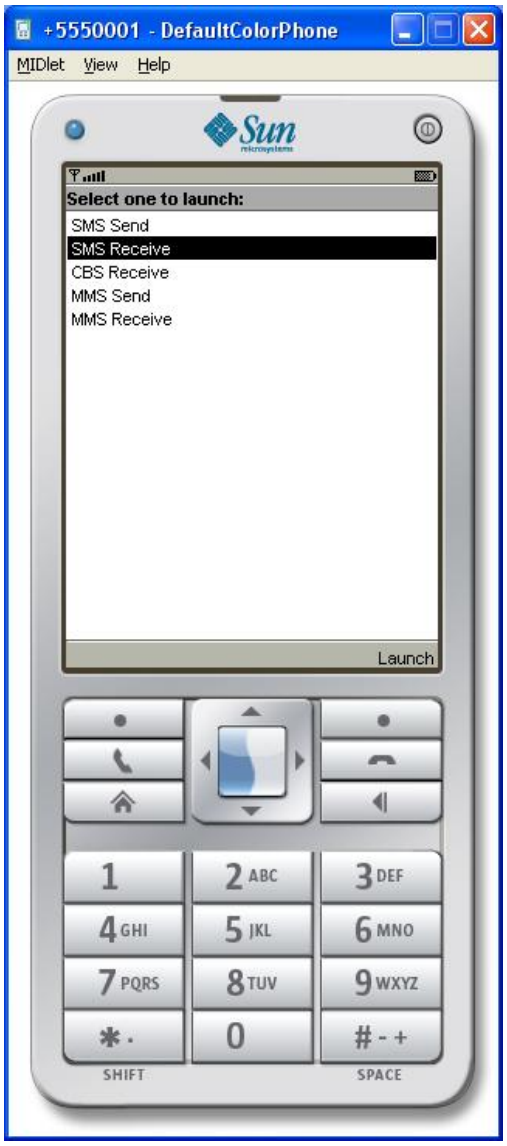
```
public void commandAction(Command c, Displayable s) {
        try {
            if (c == CMD_EXIT || c == Alert.DISMISS_COMMAND) {
                destroyApp(false);
                notifyDestroyed();
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Ορισμός των εντολών που εκτελούνται κατά το πάτημα του κουμπιού “Exit”.

6.8 Screenshots εκτέλεσης του κώδικα

Ακολουθούν τα screenshots του προσομοιωτή κατά την διάρκεια της εκτέλεσης του κώδικα που περιέχει το project με όνομα Messaging Example: εκτελείται δύο φορές ο κώδικας ώστε να εμφανιστούν δύο συσκευές όπου η μία θα λειτουργεί ως αποστολέας και η άλλη ως παραλήπτης SMS.

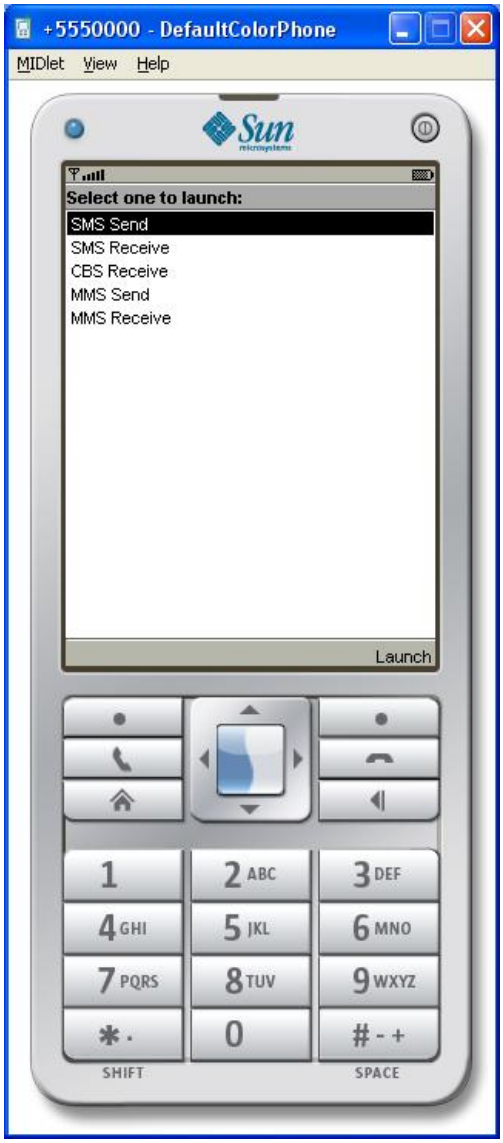
6.8.1 Αποστολή και λήψη SMS



Εικόνα 6.12

Εικόνα 6.13

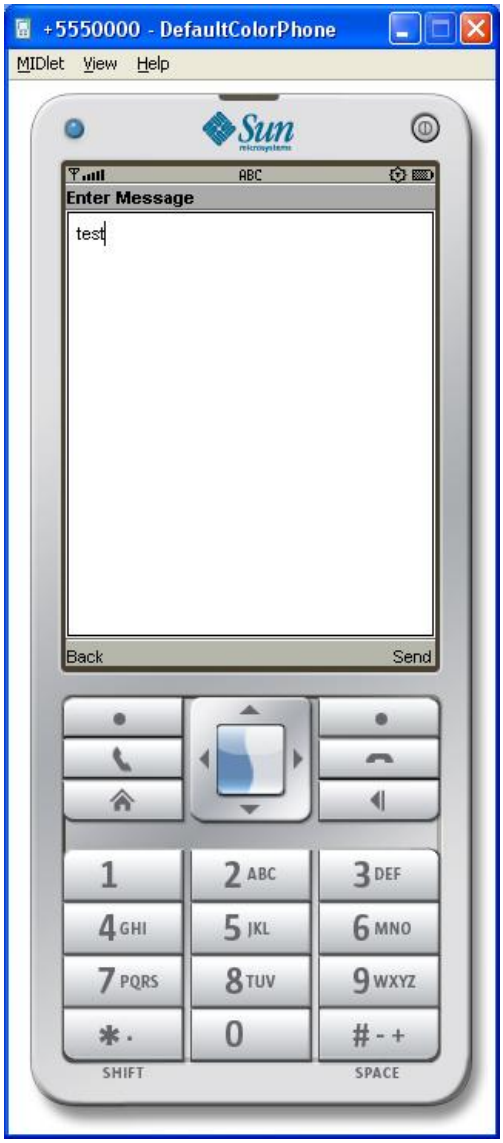
Επιλογή από το αρχικό μενού της εφαρμογής το δεύτερο MIDLET με τίτλο "SMS Receive" ώστε το κινητό με αριθμό 5550001 να καταστεί ικανό να δεχτεί SMS.

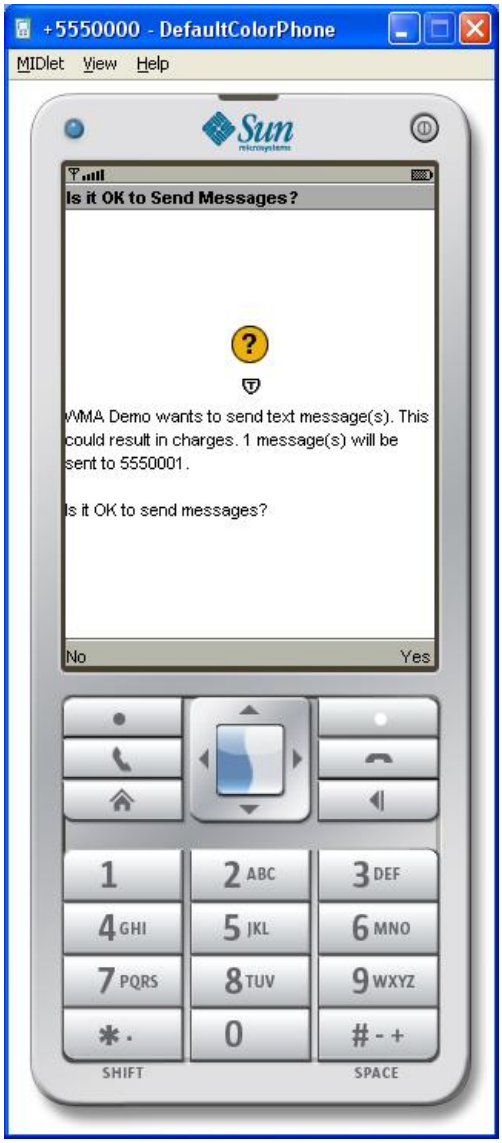


Εικόνα 6.14

Εικόνα 6.15

Για το κινητό με αριθμό +5550000, επιλογή από το αρχικό μενού την εφαρμογή SMS Send και στη συνέχεια εισαγωγή του αριθμού προορισμού που στη περίπτωσή αυτή είναι το +5550001.

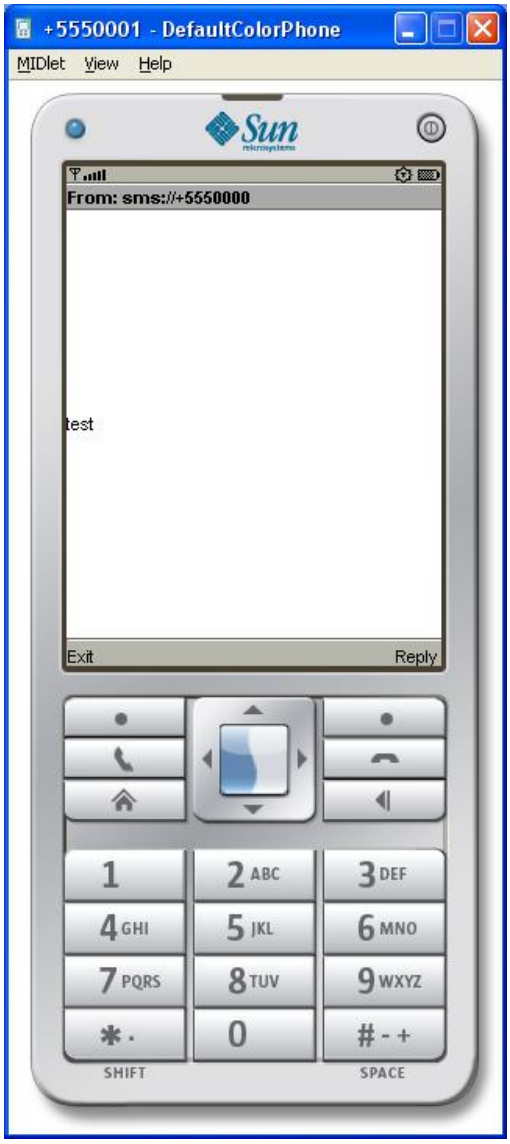




Εικόνα 6.16 ***Εικόνα 6.17***

Εισαγωγή του κειμένου που θα σταλεί και πάτημα του κουμπιού Send. Στην ερώτηση που ακολουθεί, πάτημα του "Yes".

Εικόνα 6.18



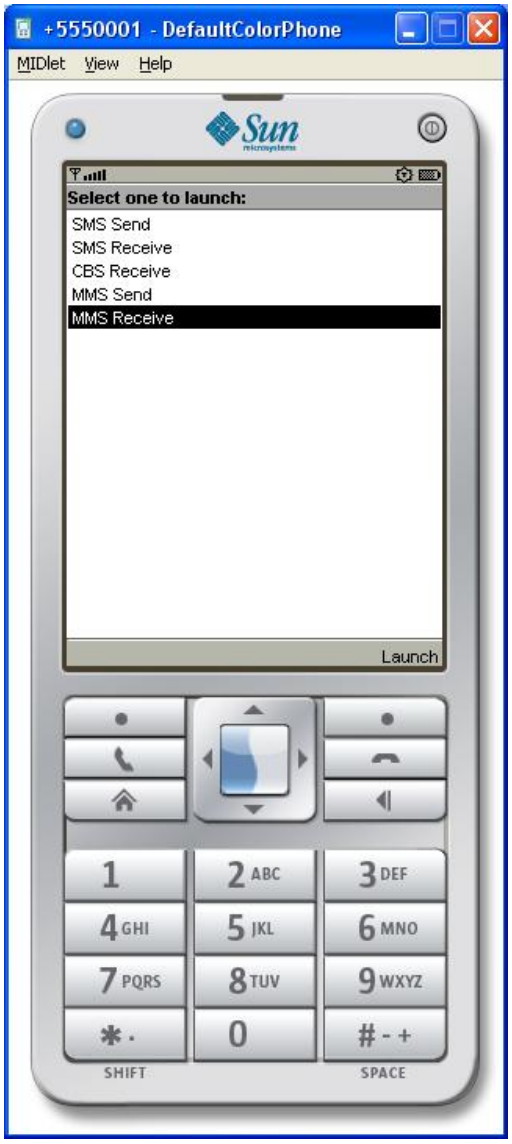
Εικόνα 6.19

Πραγματοποίηση της αποστολής. Παραλαβή και εμφάνιση του μηνύματος.

6.8.2 Αποστολή και λήψη MMS

Όπως και τα SMS, έτσι και τα MMS αποστέλλονται μεταξύ χρηστών κινητών, μόνο που τα MMS περιέχουν περισσότερα είδη πληροφοριών. Συγκεκριμένα αποτελούνται από ένα subject τύπου text, και από ένα ή περισσότερα κομμάτια (ή και καθόλου) κειμένου ή εικονας.

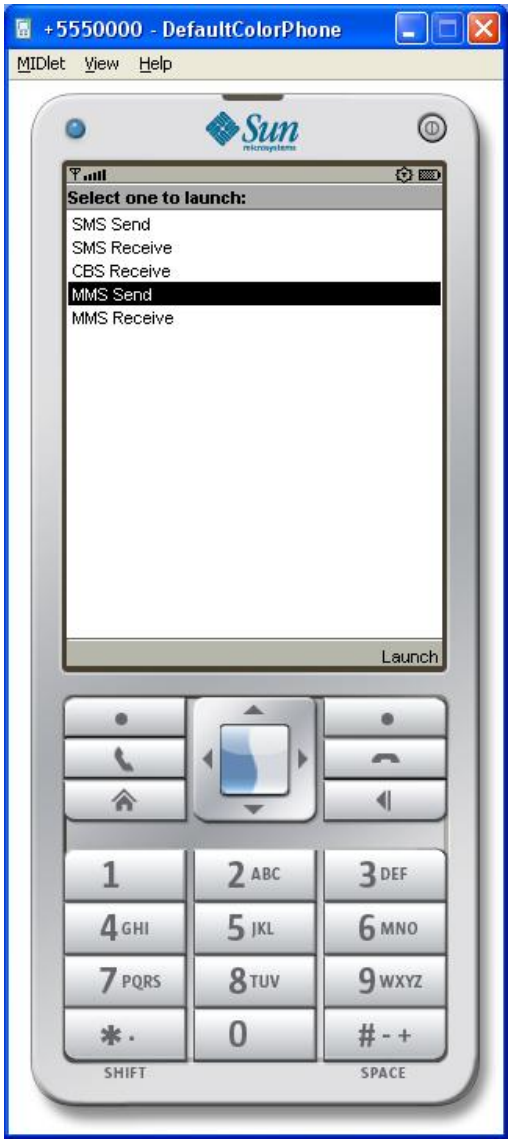
Παρακάτω ακολουθούν τα screenshots που δείχνουν τη διαδικασία αποστολής και λήψης MMS.



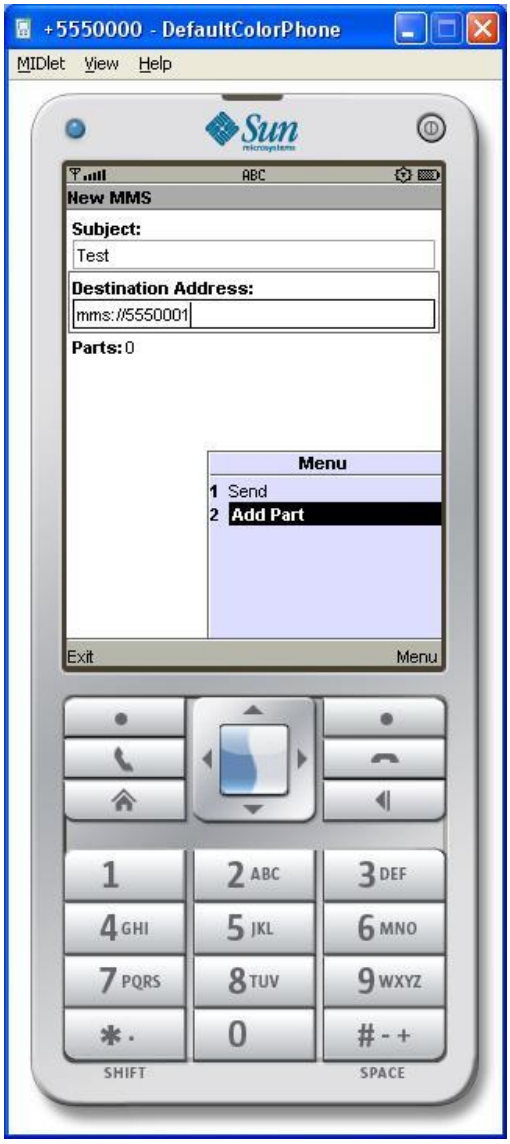
Εικόνα 6.20

Εικόνα 6.21

Επιλογή από το αρχικό μενού του δέκτη, την πέμπτη εφαρμογή με τίτλο “MMS Receive” ώστε να είναι ικανός να δεχτεί MMS.

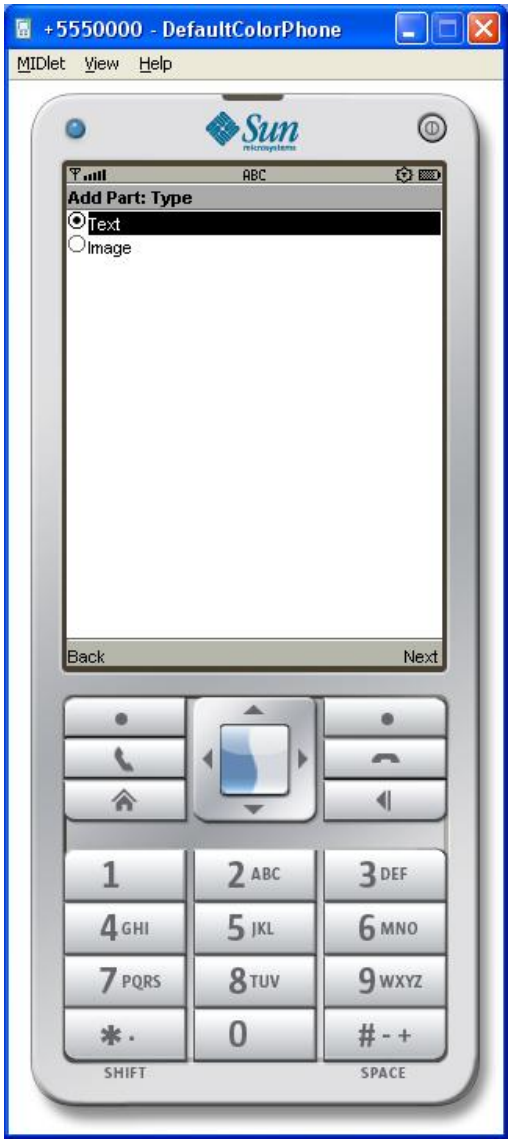


Εικονα.6.22



Εικόνα 6.23

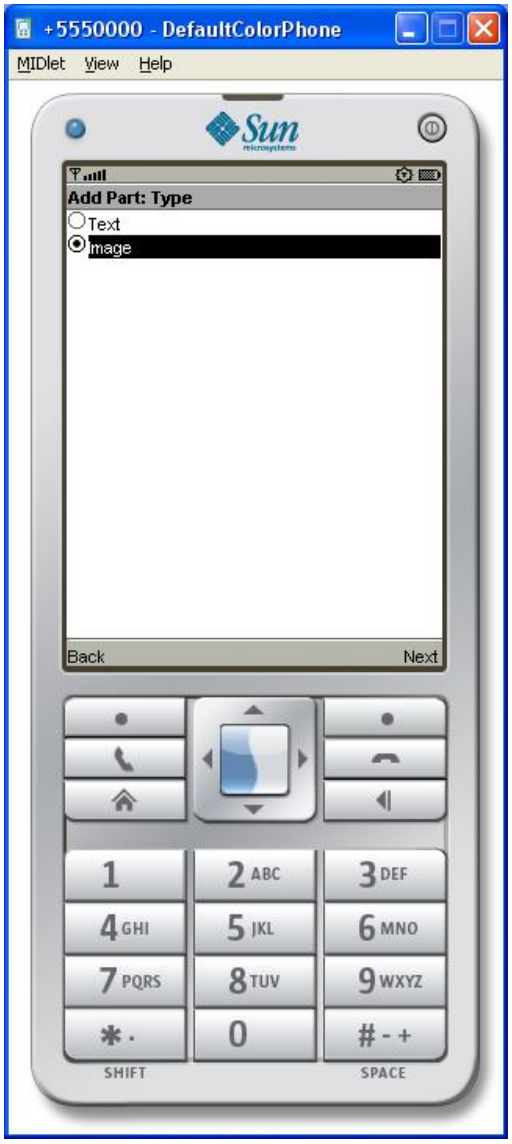
Επιλογή από το μενού του αποστολέα το τέταρτο MIDLET με τίτλο “MMS Send”. Στο πεδίο subject γράφεται το subject του MMS και στο πεδίο Destination Address τον αριθμό του παραλήπτη. Με το πάτημα του δεξιού πλήκτρου με τίτλο “Menu”, εμφανίζονται δύο επιλογές. Η πρώτη επιλογή επιτρέπει την αποστολή του μηνύματος, ενώ με την δεύτερη γίνεται η προσθήκη μερών(κείμενο και εικόνα) στο μήνυμα.



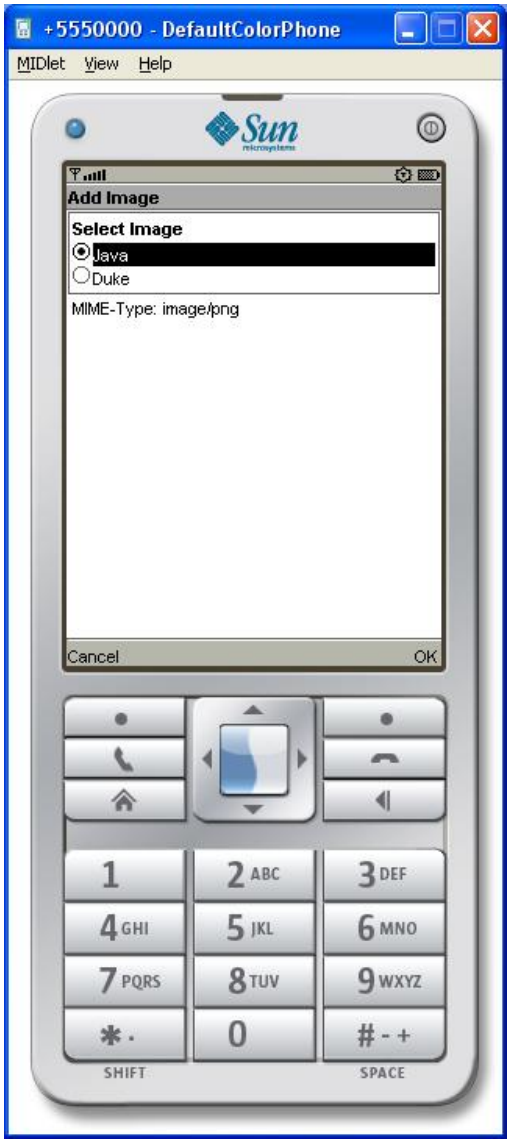
Εικόνα 6.24

Εικόνα 6.25

Με την επιλογή προσθήκης μερών από το μενού παρουσιάζεται η συγκεκριμένη οθόνη. Βάζοντας την πρώτη επιλογή από την εικόνα 6.24, εμφανίζεται ένα πεδίο για την προσθήκη κειμένου. Στη συνέχεια με το πάτημα του "OK" γίνεται η εισαγωγή του κείμενου αυτού στο μήνυμα MMS.

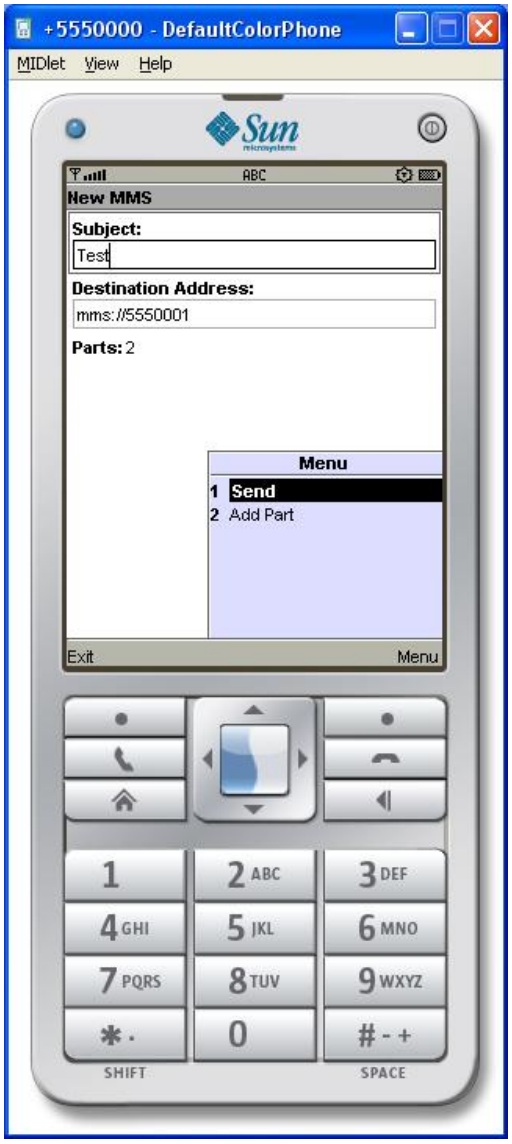


Εικόνα 6.26

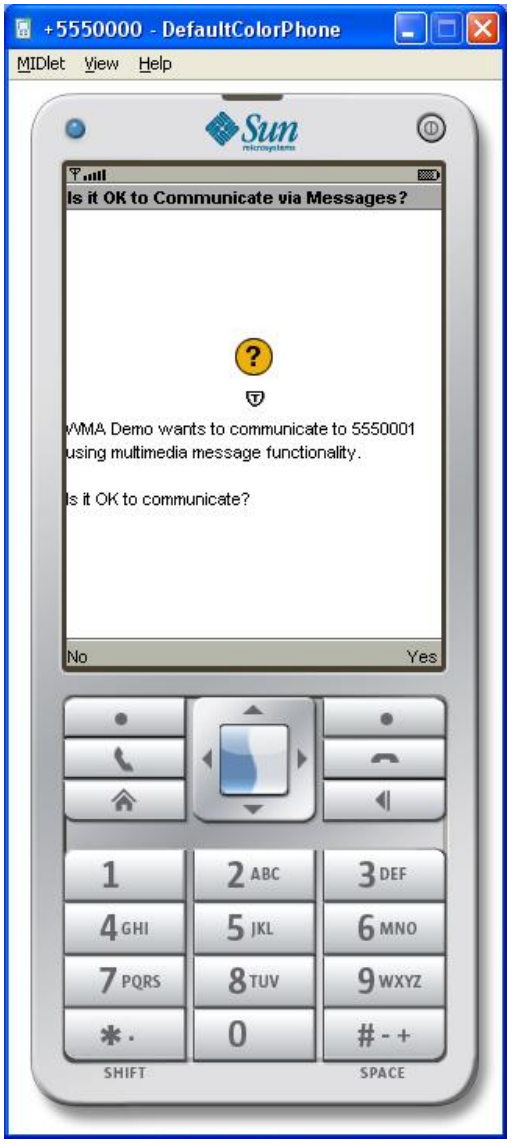


Εικόνα 6.27

Παρόμοιες ενέργειες γίνονται και για την προσθήκη εικόνας στο μήνυμα.

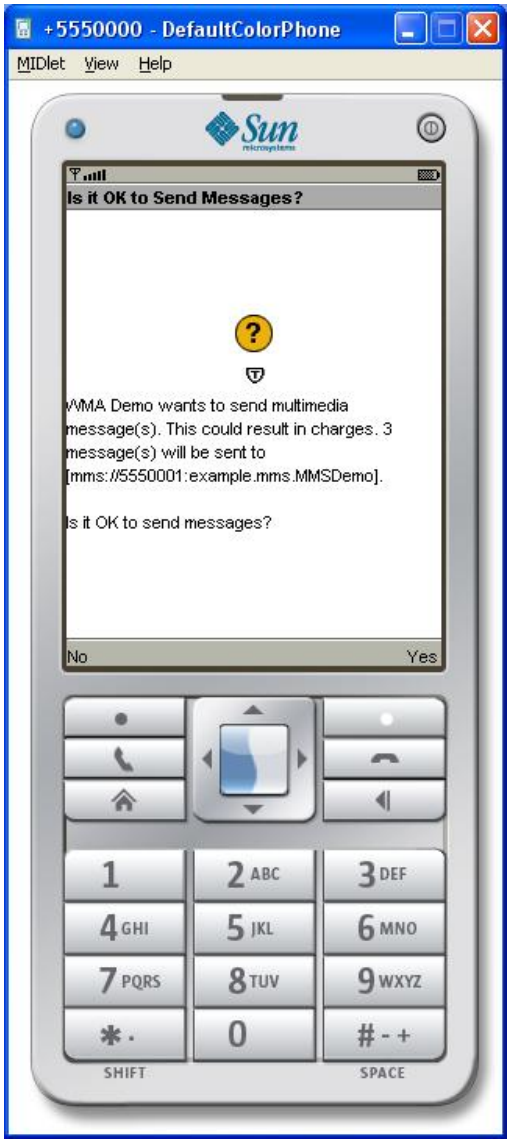


Εικόνα 6.28

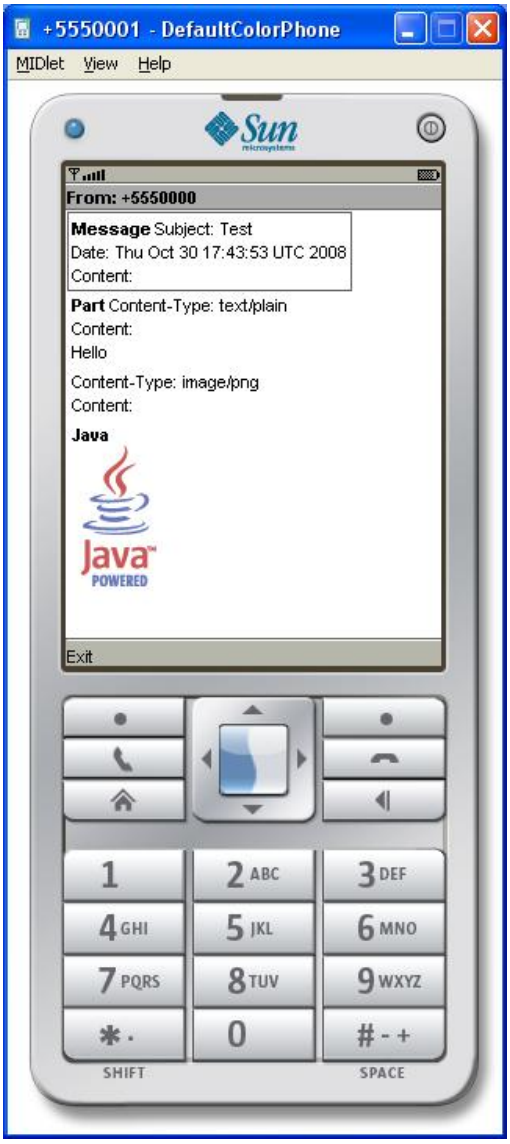


Εικόνα 6.29

Με το πάτημα του Send από το menu πραγματοποιείται η αποστολή του MMS. Στο μήνυμα που εμφανίζεται, πάτημα του κουμπιού “Yes”.



Εικόνα 6.30



Εικόνα 6.31

Και στο επόμενο μήνυμα πάτημα του κουμπιού “Yes”. Εικόνα που εμφανίζεται στον παραλήπτη αφού λάβει το μήνυμα που έχει σταλεί.

6.9 Συμπεράσματα

Ο κώδικας αυτός πραγματοποιεί αποστολή και λήψη SMS και MMS, τα οποία όμως δεν είναι κρυπτογραφημένα, με συνέπεια κάποιος επιτιθέμενος να μπορεί να διαβάσει όλα τα δεδομένα που αποστέλλονται στον αέρα.

Γι' αυτόν τον λόγο θα πρέπει να προστεθεί κρυπτογράφηση στην μεριά του αποστολέα και αποκρυπτογράφηση στην μεριά του παραλήπτη. Έτσι κάποιος παρείσακτος που θα θελήσει να ακούσει τα δεδομένα δεν θα καταφέρει να καταλάβει τίποτα αφού δεν θα ξέρει το κλειδί για να αποκρυπτογραφήσει το μήνυμα.

Επίσης, ο κώδικας αυτός δεν εξυπηρετεί πρακτικά, γιατί για να γίνει λήψη ενός μηνύματος θα πρέπει ο παραλήπτης να τρέξει το κατάλληλο MIDLET ώστε να είναι σε θέση να λάβει τέτοια μηνύματα. Και τέλος, τα MMS θα έπρεπε να περιέχουν και ήχο, εκτός από εικόνα και κείμενο.

Στο επόμενο κεφάλαιο παρουσιάζεται ο κώδικας με προσθήκες και τροποποιήσεις ώστε να γίνουν δυνατοί οι παραπάνω στόχοι.

ΚΕΦΑΛΑΙΟ 7 – ΥΛΟΠΟΙΗΣΗ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ SMS

7.1 Τροποποίηση του κώδικα

Τροποποιήθηκε ο κώδικας ώστε πρώτον, να γίνεται κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων, δεύτερον, η λήψη των SMS και MMS να γίνεται αυτόματα και τρίτον στα MMS να μεταδίδεται και ήχος.

Καταρχήν άλλαξε η δομή των κλάσεων ως εξής:

Υπάρχει μόνο μία αρχική κλάση και όχι πέντε, στην οποία ενσωματώνονται όλες οι λειτουργίες αποστολής και λήψης μηνυμάτων. Όταν ξεκινά ο προσομοιωτής εμφανίζεται μόνο αυτή η συγκεκριμένη κλάση. Κατά την έναρξη της εμφανίζεται ένα μενού με δύο επιλογές: αποστολή SMS και αποστολή MMS. Κατά την έναρξη όμως του αρχικού MIDLET και την εμφάνιση του μενού, παράλληλα τίθεται και ένας MessageListener στην πόρτα των SMS ώστε όταν έρθει κάποιο μήνυμα SMS να μπορεί να ληφθεί από το κινητό χωρίς να χρειάζεται η εκτέλεση ενός ξεχωριστού MIDLET από τον χρήστη. Έτσι, η λήψη των SMS γίνεται αυτόματα με την έναρξη και μόνο του αρχικού MIDLET.

Για τα MMS όμως η κατάσταση είναι λίγο πιο περίπλοκη. Καταρχήν δεν γίνεται να ληφθεί SMS και MMS από την ίδια πόρτα και δεύτερον δεν γίνεται να ανοιχτούν δύο συνδέσεις ταυτόχρονα, μία για τα SMS και μία για τα MMS, γιατί μετά από δοκιμές διαπιστώθηκε ότι για κάποιο λόγο δεν λειτουργεί στο περιβάλλον του προσομοιωτή.

Η αυτόματη λήψη MMS γίνεται ως εξής:

Μέσω της ανοιχτής σύνδεσης SMS, γίνεται κάποιου είδους ειδοποίηση από τον αποστολέα προς τον παραλήπτη, ώστε ο τελευταίος να προετοιμαστεί κατάλληλα για να λάβει MMS. Ο τρόπος αυτός περιγράφεται αναλυτικά λίγο παρακάτω.

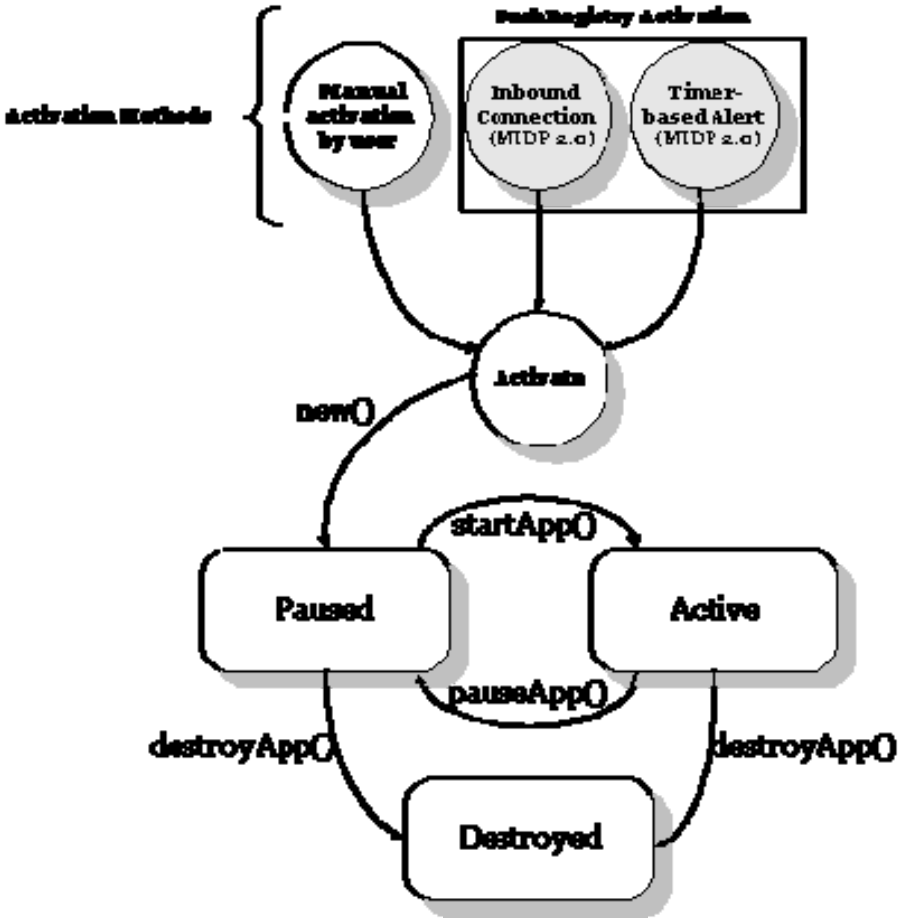
7.1.1 Push Registry

Υπάρχει μία λειτουργία η οποία ονομάζεται Push Registry και μπορεί να κάνει ένα MIDLET να εκτελεστεί αυτόματα όταν η συσκευή λάβει εισερχόμενα δεδομένα από μία εγκατεστημένη σύνδεση, π.χ. ένα SMS.

Το Push Registry υπάρχει μόνο στο MIDP 2.0 και δίνει την δυνατότητα στα MIDLET να αρχίσουν να εκτελούνται χωρίς την παρέμβαση του χρήστη. Υπάρχουν δύο τύποι αυτόματης ενεργοποίησης: ο ένας ενεργοποιεί το MIDLET όταν εμφανιστούν εισερχόμενα δεδομένα μέσω κάποιας υπάρχουσας σύνδεσης και ο άλλος βασίζεται σε ρολόι ώστε η κλάση να αρχίσει να εκτελείται κάποια συγκεκριμένη χρονική στιγμή. Για παράδειγμα μπορεί να υπάρξει ενεργοποίηση όταν ληφθεί ένα νέο μήνυμα ηλεκτρονικού ταχυδρομείου ή όταν έρθει κάποια χρονική στιγμή με βάση το ρολόι, όπου μία εφαρμογή πρέπει να κάνει κάτι ενώ απουσιάζει ο χρήστης.

Κάθε κλάση MIDLET έχει κύκλο ζωής όπου μέσα σ’αυτόν περνάει διάφορες καταστάσεις. Ενεργοποιείται, εκτελείται, παύει προσωρινά την εκτέλεση και φτάνει στο τέλος της όπου και απενεργοποιείται. Στο MIDP 1.0 ο μόνος τρόπος για να

ενεργοποιηθεί ένα MIDLET ήταν να το ξεκινήσει ο ίδιος ο χρήστης. Στο MIDP 2.0 η λειτουργία Push Registry προσθέτει δύο επιπλέον τρόπους ενεργοποίησης των MIDLET. Ενεργοποίηση με βάση εισερχόμενα δεδομένα από κάποια σύνδεση και με βάση κάποια χρονική σήμανση. Το παρακάτω σχήμα δείχνει τον κύκλο ζωής των εφαρμογών αυτών.



Εικόνα 7.1 – Κύκλος ζωής των Midlet.

Για να υποστηρίζεται η ενεργοποίηση μέσω εισερχόμενης σύνδεσης θα πρέπει να υποστηρίζονται επίσης και διάφοροι τύποι συνδέσεων. Στο MIDP 1.0 υπάρχει μόνο η http σύνδεση. Στο MIDP 2.0 παρέχεται επίσης δυνατότητα χρήσης TCP Sockets και UDP Datagrams. Επιπλέον ένα πρόσθετο πακέτο με όνομα Wireless Messaging API, δίνει την δυνατότητα αυτόματης ενεργοποίησης MIDLET μέσω εισερχόμενου μηνύματος SMS.

Το Push Registry θα μπορούσε να χρησιμοποιηθεί για την αυτόματη λήψη SMS και MMS. Αυτή η λειτουργία όμως έχει δύο βασικά μειονεκτήματα:

1. Για να γίνει αυτόματη εκκίνηση ενός MIDLET θα πρέπει η κλάση να δηλωθεί στις ιδιότητες του project ως MIDLET. Αυτό έχει σαν αποτέλεσμα την εμφάνιση της κλάσης ως αρχική εκτελέσιμη εφαρμογή. Η κλάση αυτή όμως θα πρέπει να τρέχει μόνο στο παρασκήνιο χωρίς να φαίνεται στον χρήστη ότι υπάρχει τέτοια εφαρμογή και εφόσον εκτελείται αυτόματα, δεν υπάρχει κανένας λόγος να εμφανίζεται στο αρχικό μενού.

2. Στον κώδικα που περιγράφεται παραπάνω, η κλάση λήψης μηνυμάτων εμφανίζεται στην αρχική οθόνη οπότε και παρακάμπτει το παραπάνω μειονέκτημα. Παρ' όλ' αυτά όμως δεν μπορεί να εφαρμοστεί Push Registry, διότι για να αρχίσει να εκτελείται κάποιο MIDLET αυτόματα με την λήψη ενός εισερχόμενου μηνύματος, απαιτείται να εκτελείται εκείνη την ώρα κάποιο άλλο MIDLET. Στην αρχική οθόνη όμως δεν εκτελείται καμία τέτοια κλάση και άρα δεν μπορεί να εκτελεστεί αυτόματα το MIDLET λήψης και έτσι δεν γίνεται να εφαρμοστεί Push Registry.

Γι' αυτόν τον λόγο αλλάχθηκε η δομή των κλάσεων από πέντε ανεξάρτητα MIDLET σε ένα κεντρικό και μέσα από αυτό θα εκτελούνται οι κλάσεις αυτές ως μέθοδοι στην κλάση MIDLET. Έτσι ο χρήστης, με το που ανοίξει ο προσομοιωτής, θα ξεκινήσει την εκτέλεση της αρχικής κλάσης, αφού ο τίτλος της θα δείχνει ότι είναι απλά το όνομα της εφαρμογής και όχι μενού επιλογών.

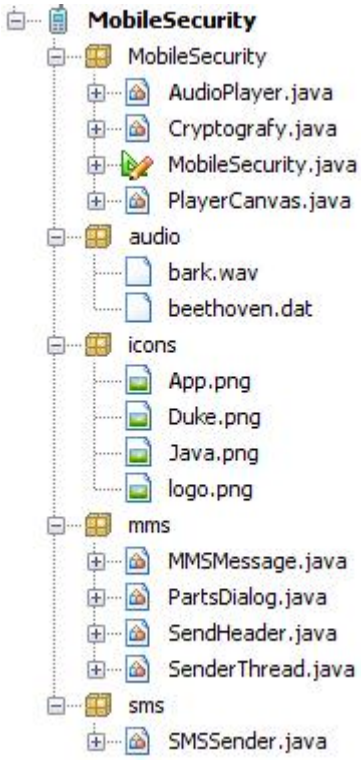
Για την προθήκη ήχου στα μηνύματα πολυμέσων χρησιμοποιήθηκε ένα άλλο project που υπάρχει στα Netbeans με το όνομα Audio Player for MIDP 2.0 το οποίο βρίσκεται στον ίδιο φάκελο με το Messaging Example. Το project αυτό τροποποιήθηκε και ενσωματώθηκε στο παρόν project.

7.2 Ανάλυση του κώδικα

Ακολουθεί η ανάλυση του κώδικα μετά τις αλλαγές αυτές.

Παρακάτω φαίνεται η δομή των κλάσεων του νέου project το οποίο ονομάστηκε "Mobile Security". Το project αυτό δημιουργήθηκε με βάση τα βήματα των εικόνων 6.2 και 6.3. Στην εικόνα 6.3 επιλέχθηκε ο πέμπτος φάκελος με το όνομα "Mobile" και από την δεξιά στήλη επιλέχθηκε η πρώτη επιλογή με το όνομα "Mobile Application". Στην συνέχεια δόθηκε το όνομα και διάφορα άλλα στοιχεία του project όπως έγινε και στον προηγούμενο κώδικα.

Στην αρχή το project αποτελείται μόνο από το πρώτο πακέτο με όνομα "MobileSecurity" και από την συνόνομη κλάση που βρίσκεται μέσα του. Τα τρία πακέτα έχουν προστεθεί από το προηγούμενο project και το τελευταίο, που περιέχει δύο αρχεία ήχου, προστέθηκε από το project "Audio Player for MIDP 2.0". Έτσι το project τώρα αποτελείται από πέντε πακέτα τα οποία φαίνονται στην παρακάτω εικόνα.



Εικόνα 7.2 – Δομή πακέτων και κλάσεων

Το πρώτο πακέτο έχει όνομα "MobileSecurity" και περιέχει το κεντρικό MIDLET με όνομα το ίδιο με αυτό του πακέτου. Επίσης περιέχει την κλάση Cryptografy η οποία ενσωματώνει τις μεθόδους κρυπτογράφησης και αποκρυπτογράφησης. Τέλος, προστέθηκαν και οι κλάσεις AudioPlayer και PlayerCanvas που χρησιμοποιούνται στην αναπαραγωγή του ήχου. Το δεύτερο πακέτο περιέχει δύο κομμάτια ήχου για χρήση στα MMS. Το τρίτο ονομάζεται "icons" και περιλαμβάνει τις δύο εικόνες που χρησιμοποιούνται στα μηνύματα MMS και άλλες δύο που χρησιμοποιούνται στον Player του ήχου. Το τέταρτο λέγεται "mms" και έχει τις κλάσεις για την αποστολή και λήψη MMS και το πέμπτο και τελευταίο, το οποίο ονομάζεται "sms", περιέχει αντίστοιχα μία κλάση που χρησιμοποιείται για αποστολή και λήψη SMS.

Το τέταρτο και πέμπτο πακέτο έχει τις παρακάτω κλάσεις:

4ο πακέτο "mms"

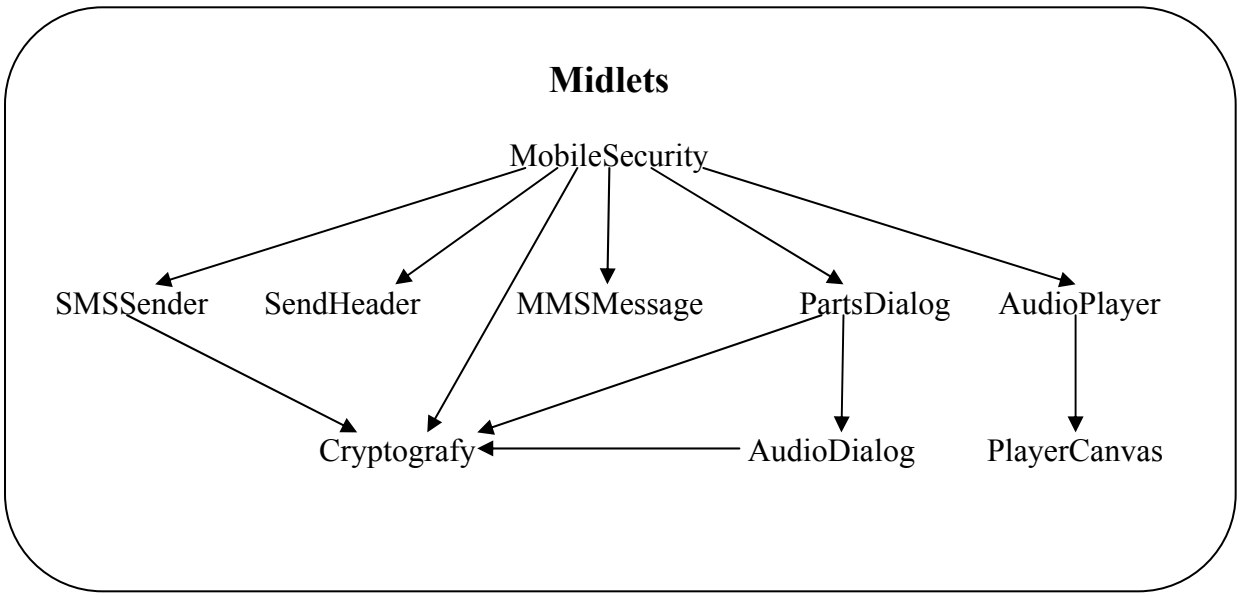
MMSMessage
PartsDialog
SendHeader
SenderThread

5ο πακέτο "sms"

SMSSender

Οι κλάσεις “MMSMessage” και “SenderThread” του τέταρτου πακέτου, είναι ακριβώς οι ίδιες με αυτές του προηγούμενου κώδικα και γι’ αυτό δεν χρειάζεται ξανά να περιγραφεί εδώ. Οι κλάσεις “PartDialog” και “SMSSender” είναι και αυτές από τον προηγούμενο κώδικα με μικρές μόνο τροποποιήσεις που θα περιγραφούν παρακάτω. Η κλάση “SendHeader” δημιουργήθηκε έπειτα και θα περιγραφεί παρακάτω η χρησιμότητά της.

Το παρακάτω σχήμα δείχνει τις κλάσεις και τον τρόπο που καλούνται μεταξύ τους.



Εικόνα 7.3 – Κλήσεις των κλάσεων στο τελικό project.

Ακολουθεί αναλυτική περιγραφή του κώδικα.

7.3 Κλάση Cryptografy

Αυτή η κλάση περιέχει μεθόδους που κάνουν την διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης στα SMS και στα MMS. Μετά από δοκιμές διαπιστώθηκε ότι το J2ME υποστηρίζει μόνο δύο αλγόριθμους κρυπτογράφησης, τον AES και τον DES. Θα χρησιμοποιηθεί ο AES καθώς είναι πιο μεταγενέστερος και θεωρείται η καλύτερη επιλογή. Υποστηρίζει μήκη κλειδιών και τμημάτων 128 και 256 bit. Το μήκος του τμήματος μπορεί να επιλεγεί ανεξάρτητα από το μήκος του κλειδιού. Η πιο συνηθισμένη χρήση του AES γίνεται με μήκος κλειδιού και τμήματος 128 bit. Στο project θα χρησιμοποιηθεί αυτό το μήκος το οποίο θεωρείται αρκετά ασφαλές.

Η διαδικασία κρυπτογράφησης έχει ως εξής: Υπάρχει πάντα ένα μόνιμο κλειδί που δεν αλλάζει και είναι γνωστό σε όλους. Αυτό το κλειδί χρησιμοποιείται για την κρυπτογράφηση των παραμέτρων οι οποίοι χρησιμοποιούνται για την παραγωγή των κλειδιών συνόδου. Τα κλειδιά συνόδου παράγονται από τον αποστολέα πριν την αποστολή κάποιου μηνύματος και με αυτά κρυπτογραφούνται τα δεδομένα του μηνύματος. Τα κλειδιά αυτά ισχύουν μόνο για μία αποστολή. Με κάθε αποστολή

παράγεται καινούργιο κλειδί και εφόσον η κρυπτογράφηση είναι συμμετρική θα πρέπει το κλειδί αυτό να γίνει γνωστό και στον παραλήπτη. Αυτό γίνεται στέλνοντας στον παραλήπτη την παράμετρο για την παραγωγή του προσωρινού κλειδιού, κρυπτογραφημένη με το μόνιμο κλειδί. Ο παραλήπτης όταν λάβει το μήνυμα, ξεχωρίζει την παράμετρο, την αποκρυπτογραφεί και στην συνέχεια παράγει το κλειδί συνόδου με το οποίο αποκρυπτογραφεί τα δεδομένα του μηνύματος.

Τα δεδομένα για να παραχθεί το κλειδί συνόδου λαμβάνονται από μία γεννήτρια ψευδοτυχαίων αριθμών. Η γεννήτρια όμως αυτή παράγει κάθε φορά την ίδια αλληλουχία τυχαίων αριθμών και γι' αυτό της δίνεται ως παράμετρος τα milliseconds του συστήματος ώστε τα δεδομένα που δίνει να είναι πραγματικά τυχαία. Τα δεδομένα αυτά είναι 16 αριθμοί τύπου long, οι οποίοι με κατάλληλη διαδικασία συνεισφέρουν στην παραγωγή του κλειδιού.

Αυτό που κρυπτογραφείται με το μόνιμο κλειδί και αποστέλλεται στον παραλήπτη δεν είναι το κλειδί συνόδου, αλλά τα millisecond όπου παίρνει η γεννήτρια ως παράμετρο. Αυτή η παράμετρος αναφέρεται στην συνέχεια και ως "Salt" ή "σπόρος". Στην συνέχεια η γεννήτρια παράγει ακριβώς τους ίδιους τυχαίους αριθμούς και από αυτούς, ο παραλήπτης μπορεί να παράγει το ίδιο κλειδί με τον αποστολέα.

Η παράμετρος για να παραχθεί το προσωρινό κλειδί, κρυπτογραφείται και συνενώνεται μαζί με τα επίσης κρυπτογραφημένα δεδομένα και όλα μαζί αποστέλλονται σε μορφή string. Η παράμετρος τοποθετείται πρώτη στη σειρά μέσα στο αλφαριθμητικό και ακολουθούν τα δεδομένα του μηνύματος. Παρακάτω περιγράφεται με ποιο τρόπο ο παραλήπτης ξεχωρίζει τον σπόρο από το μήνυμα.

Για να γίνει η διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης δημιουργείται ένα αντικείμενο της κλάσης Cipher. Για την δημιουργία του αντικειμένου αυτού καλείται η μέθοδος getInstance(), όπου δέχεται ως παράμετρο τον αλγόριθμο κρυπτογράφησης. Στην αρχή το αντικείμενο αυτό αρχικοποιείται μέσω της μεθόδου init(), παίρνοντας ως παράμετρο την σταθερά λειτουργίας (κρυπτο/ση-αποκρυ/ση) και το μυστικό κλειδί.

Η Cipher επεξεργάζεται τα δεδομένα σε μορφή byte, τα οποία παίρνει από κάποιον πίνακα 16 ή 32 θέσεων και εξάγει το αποτέλεσμα επίσης σε πίνακα byte. Στο παρόν project τα δεδομένα εισάγονται από πίνακα 16 θέσεων. Στην περίπτωση της κρυπτογράφησης, όταν τα αρχικά δεδομένα είναι λιγότερα από 16 byte, καλείται μόνο η μέθοδος dofinal(), ενώ σε αντίθετη περίπτωση καλείται πρώτα η update(), η οποία κρυπτογραφεί με την σειρά ομάδες των 16 byte, μέχρις ότου μείνουν λιγότερα από 16 byte τα οποία και κρυπτογραφούνται στο τέλος με την μέθοδο dofinal(). Η update() μπορεί να κληθεί πάνω από μία φορές συνεχόμενα. Το αποτέλεσμα της dofinal() έχει μήκος 32 byte και όχι 16 όπως συμβαίνει στην update(). Και οι δύο αυτές μέθοδοι παίρνουν ως είσοδο έναν πίνακα 16 θέσεων ο όποιος περιέχει τα αρχικά δεδομένα. Τα παραπάνω ισχύουν και για την αποκρυπτογράφηση με την διαφορά ότι στην dofinal() τα εισαγόμενα δεδομένα είναι 32 byte και όχι 16.

Στην περίπτωση των SMS γίνεται κρυπτογράφηση κάποιου αλφαριθμητικού. Επειδή όμως η διαδικασία γίνεται μόνο με δεδομένα τύπου byte, θα πρέπει το μήνυμα να μετατραπεί σε σειρά από byte. Αυτό γίνεται με την εξής εντολή:

byte[ ] = text.getBytes( );

Μετά από αυτήν την εντολή, γίνεται η κρυπτογράφηση και το αποτέλεσμα βρίσκεται σε κάποιον πίνακα byte. Η αποστολή όμως του μηνύματος θα πρέπει να γίνει μόνο σε μορφή string. Για την μετατροπή ενός πίνακα byte σε αλφαριθμητικό θα αρκούσε η παρακάτω απλή εντολή:

String str = new String(byte[ ]);

Υπάρχει όμως ένα πρόβλημα σε αυτήν την μετατροπή. Η java δουλεύει τους χαρακτήρες και επομένως τα string, σε κωδικοποίηση Unicode. Σε αυτήν την κωδικοποίηση όμως ορισμένα byte αντιστοιχούν σε έναν αόριστο συγκεκριμένο χαρακτήρα και κατά την μετατροπή, αυτά τα συγκεκριμένα byte αλλάζουν σε μία άλλη τιμή, η οποία απεικονίζει τον χαρακτήρα '?', ο όποιος σημαίνει ένας οποιοσδήποτε χαρακτήρας. Εξαιτίας αυτού του προβλήματος χρειάστηκε να γίνει κάποιου είδους κωδικοποίηση των byte.

Η κωδικοποίηση έχει ως εξής:

Ο αποστολέας παίρνει τα byte από τον πίνακα σε μορφή ακέραιου αριθμού με πρόσημο και τα μετατρέπει ένα-ένα σε μικρά αλφαριθμητικά μέγιστου μήκους 4 χαρακτήρες, όπου απεικονίζουν αυτούς τους ακέραιους. Στην συνέχεια τα ενώνει σε ένα μεγάλο ενιαίο string και τα διαχωρίζει μεταξύ τους με τελείες. Τέλος, στέλνει το μεγάλο αλφαριθμητικό και ο παραλήπτης ακολουθεί την αντίστροφη διαδικασία ώστε να πάρει τα κρυπτογραφημένα bytes και να προχωρήσει στην αποκρυπτογράφηση τους. Η ίδια κωδικοποίηση γίνεται και στον σπόρο που χρησιμοποιείται για να παραχθεί το κλειδί συνόδου αλλά και στο subject των MMS.

Ακολουθεί ένα παράδειγμα της κωδικοποίησης:

".142.-16.-125.235.198.2.112.-8.-84.102"

Αυτό είναι το τελικό αλφαριθμητικό που θα σταλεί στον παραλήπτη και απεικονίζει τις τιμές των κρυπτογραφημένων byte. Ο αριθμός που βρίσκεται ανάμεσα σε δύο τελείες δείχνει την τιμή που έχει κάποιο byte. Η τιμή αυτή κυμαίνεται από -127 μέχρι 128, συμπεριλαμβανομένου και το μηδέν.

Όταν ο παραληπτης λαβει το μηνυμα πρεπει να ξεχωρισει τον κρυπτογραφημενο σπορο από τα υπολοιπα δεδομενα. Για να το κανει αυτό βασιζεται στο γεγονος ότι το μηκος του σπορου είναι σταθερο και ισο με 16 Byte. Μετρωντας τις τελειες, ξεχωριζει τα 16 πρωτα Byte και τα επεξεργαζεται στην συνεχεια ως σπορο και τα υπολοιπα ως δεδομενα του μηνυματος.

Το μειονέκτημα αυτής της κωδικοποίησης είναι ότι στέλνεται περισσότερη πληροφορία από αυτήν που θα στέλνονταν χωρίς κωδικοποίηση, αφού για έναν χαρακτήρα στο αρχικό μήνυμα απαιτούνται από 2 έως 5 συνολικά χαρακτήρες. Συνήθως για έναν χαρακτήρα αντιστοιχούν 4 ή 5 και πιο σπάνια 2 ή 3 χαρακτήρες. Για παράδειγμα, ο χαρακτήρας 'R' κωδικοποιείται ως ".114", που είναι 4 χαρακτήρες και αν τύχει να υπάρχει και πρόσημο, οι χαρακτήρες αυτοί γίνονται 5, δηλαδή ".-114". Στα MMS δεν χρειάζεται αυτή η κωδικοποίηση στα μέρη του μηνύματος

διότι αυτά στέλνονται σε μορφή byte κατευθείαν. Το subject όμως στέλνεται ως αλφαριθμητικό μαζί με τον σπόρο και ισχύουν τα ίδια πράγματα με τα μηνύματα SMS.

Στα MMS, το προσωρινό κλειδί παράγεται κατά την είσοδο του χρήστη στην αρχική οθόνη αποστολής των MMS. Αυτό γίνεται διότι κατά την προσθήκη μερών στο μήνυμα θα πρέπει ήδη να υπάρχει το κλειδί ώστε να γίνει κρυπτογράφηση του μέρους αυτού.

Θα μπορούσε το κλειδί να παράγεται κατά την εισαγωγή του πρώτου μέρους και να αποθηκευόταν για τυχόν επόμενη προσθήκη κάποιου άλλου μέρους του μηνύματος. Ο χρήστης όμως θα μπορούσε κάλλιστα να μην προσθέσει μέρη στο μήνυμα και να στείλει απλά το subject. Σε αυτήν την περίπτωση δεν θα παραγόταν κλειδί και το μήνυμα δεν θα κρυπτογραφόταν. Μία άλλη λύση θα ήταν να παράγεται το κλειδί κατά την διαδικασία της αποστολής. Σ' αυτήν την περίπτωση όμως τα μέρη δεν θα κρυπτογραφόνταν, καθώς η διαδικασία αυτή πρέπει να γίνεται κατά την προσθήκη των μερών στο μήνυμα και όχι κατά την αποστολή. Έτσι, αφού η προσθήκη προηγείται της αποστολής, τα μέρη θα μένανε όπως ήταν.

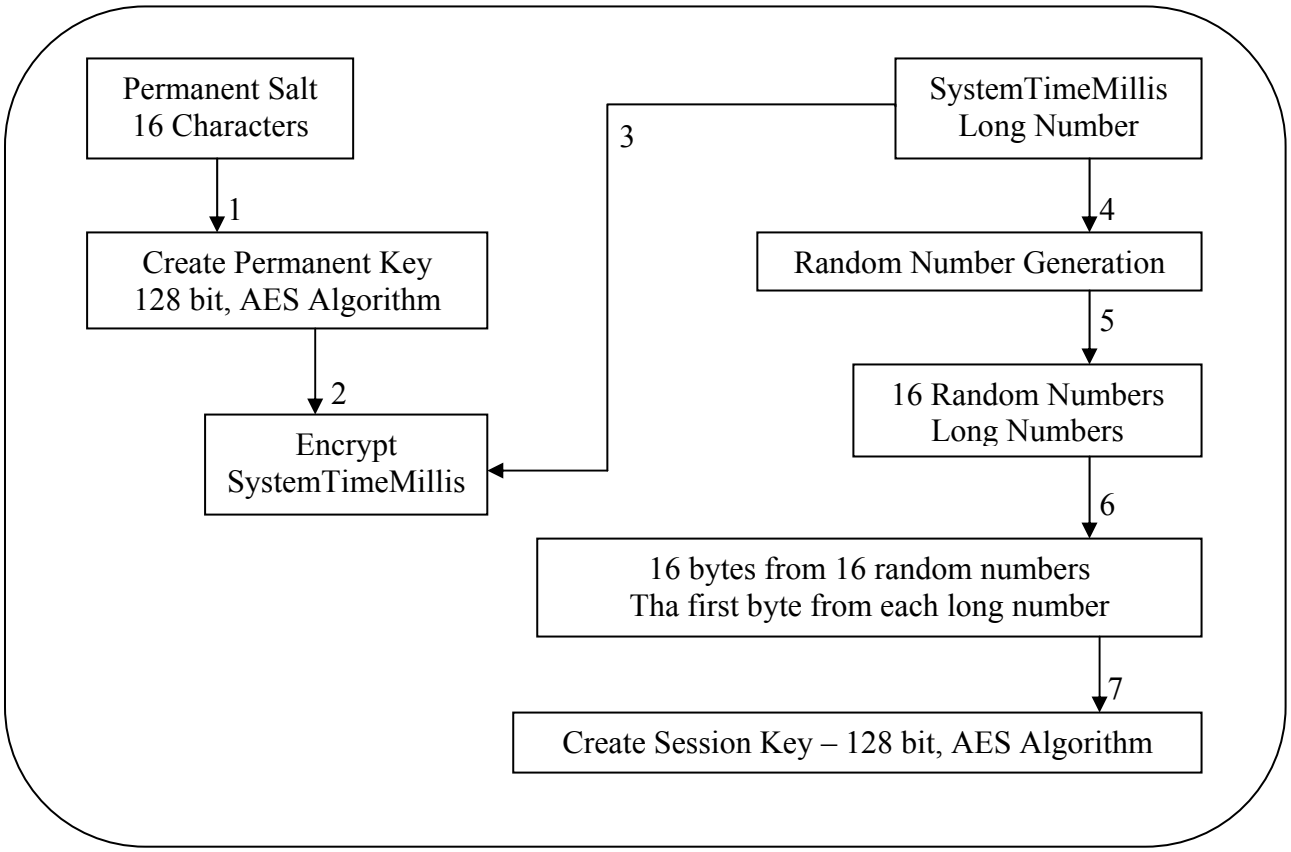
Έτσι, κατά την είσοδο του χρηστή στην αρχική οθόνη αποστολής MMS δημιουργείται ένα αντικείμενο της κλάσης Cryptografy και μέσω της κλήσης της μεθόδου createKey() παράγεται το προσωρινό κλειδί το οποίο αποθηκεύεται σε μία μεταβλητή τύπου secretKeySpec. Η μεταβλητή αυτή και το αντικείμενο της κλάσης Cryptografy είναι δημοσίως γνωστά μέσα στην κλάση MobileSecurity. Αυτό γίνεται για να μπορεί κάποια μέθοδος να χρησιμοποιήσει το κλειδί αυτό όποτε θέλει εφόσον απαιτείται η χρησιμοποίηση του.

Με το κλειδί αυτό κρυπτογραφούνται τα μέρη αλλά και το subject του MMS. Το subject όμως αποστέλλεται ως αλφαριθμητικό και λόγω της κωδικοποίησης που περιγράφεται παραπάνω, οι χαρακτήρες που στέλνονται είναι πολύ περισσότεροι από αυτούς που γράφει ο χρήστης. Στο subject όμως επιτρέπεται να σταλούν μέχρι 40 χαρακτήρες, έτσι με την κωδικοποίηση που γίνεται, το θέμα δεν μπορεί να σταλεί λόγω σχετικά μεγάλου μήκους. Η λύση είναι πολύ απλή και βρίσκεται στην χρήση της σύνδεση SMS με τον παρακάτω τρόπο.

Πριν την αποστολή του MMS, στέλνεται μία ειδοποίηση στον παραλήπτη μέσω της ανοιχτής πόρτας των SMS, ώστε αυτός να ανοίξει την πόρτα των MMS για να δεχτεί το μήνυμα. Μαζί με αυτήν την ειδοποίηση στέλνεται μαζί και το θέμα του MMS, το οποίο αποθηκεύεται ώστε να εμφανιστεί λίγο αργότερα στην οθόνη του παραλήπτη όταν θα έχει έρθει και το υπόλοιπο μήνυμα MMS.

Τα παρακάτω σχήματα δείχνουν πως γίνεται η αποστολή και η λήψη των κρυπτογραφημένων μηνυμάτων SMS και MMS καθώς και η δημιουργία των κλειδιών.

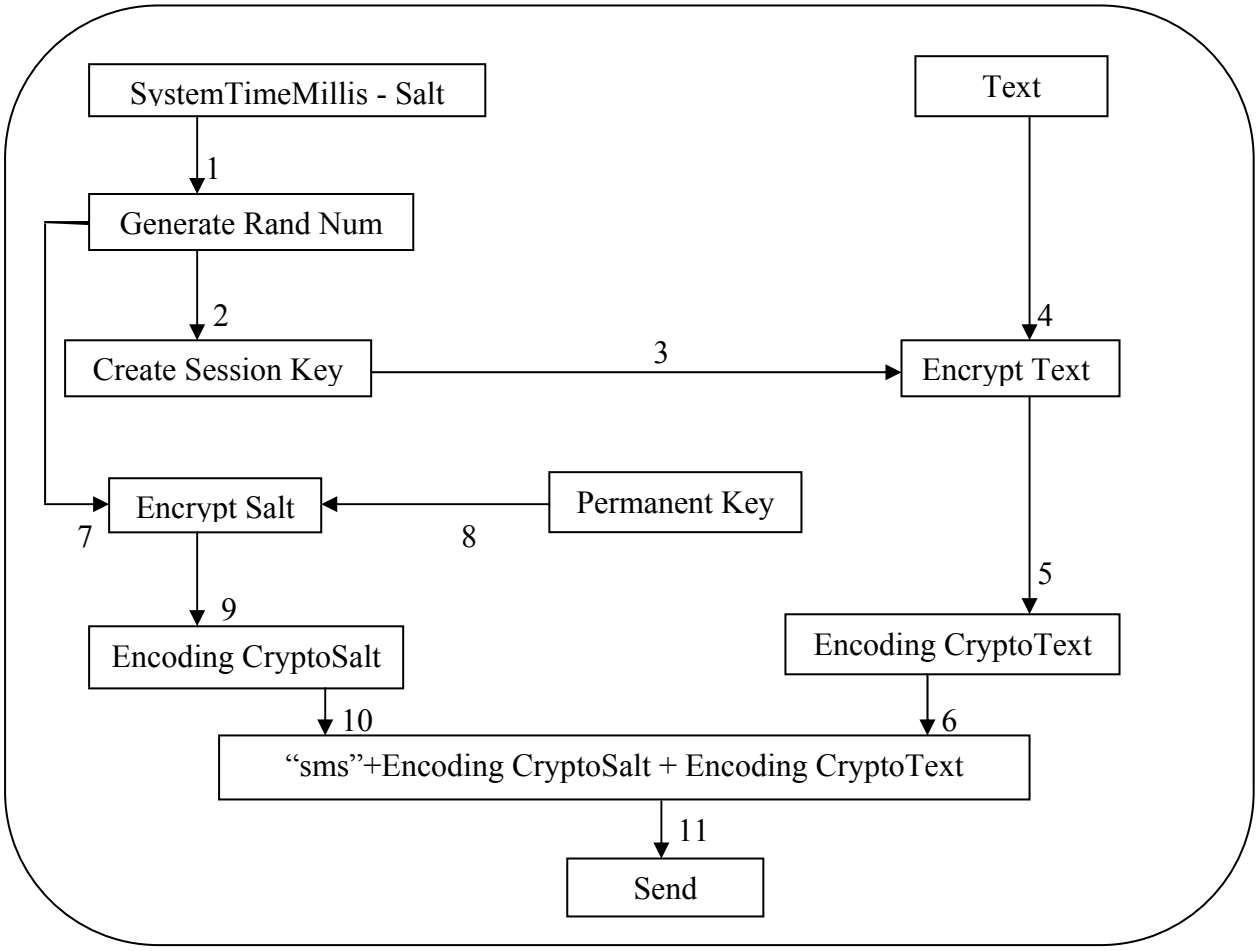
ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ ΚΛΕΙΔΙΩΝ



Εικόνα 7.4 – Δημιουργία μόνιμου και προσωρινού κλειδιού

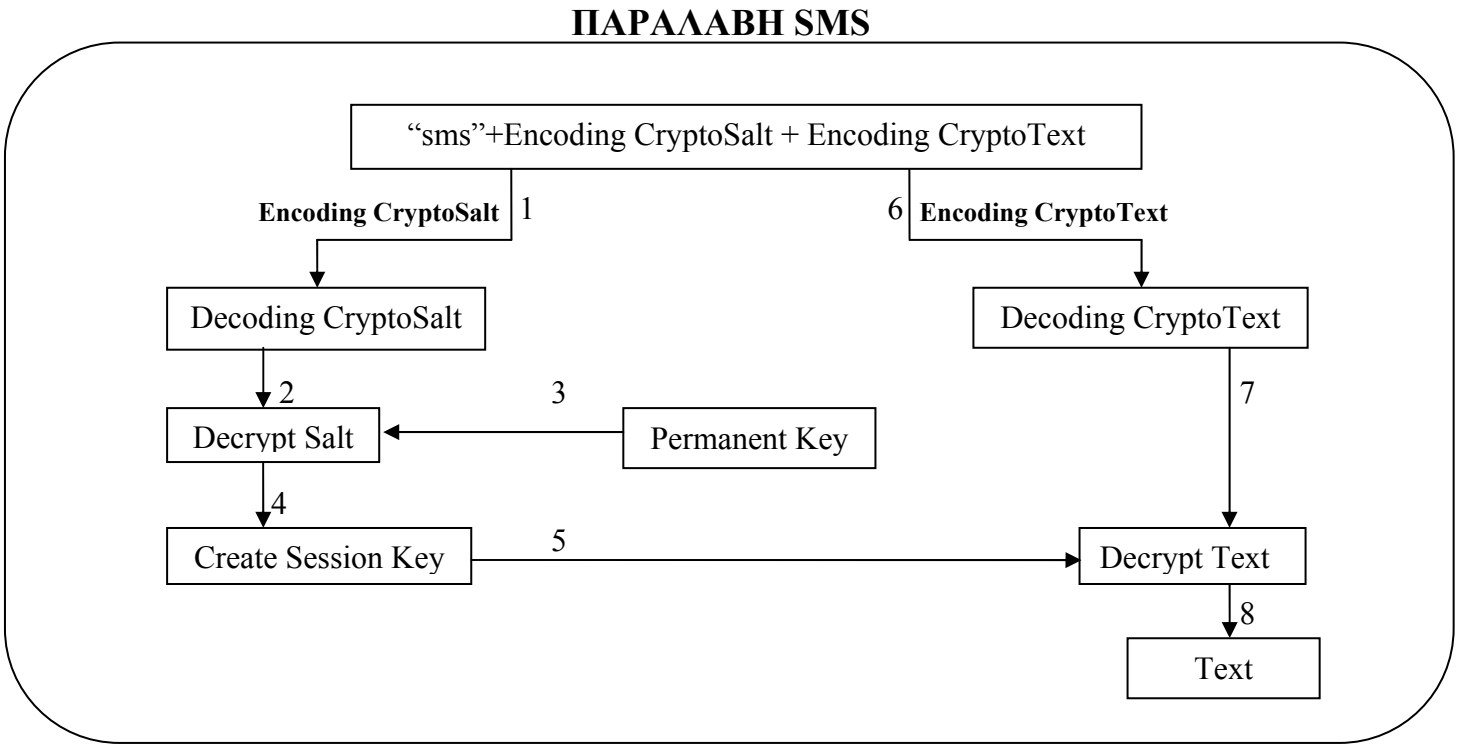
Permanent Salt – Μόνιμο String που χρησιμοποιείται για την παραγωγή του μόνιμου κλειδιού.
Create Permanent Key – Δημιουργία μόνιμου κλειδιού.
Encrypt SystemTimeMillis – Κρυπτογράφηση των Milliseconds του συστήματος.
Random Number Generation – Παραγωγή τυχαίων αριθμών τύπου long.
Create Session Key – Παραγωγή προσωρινού κλειδιού.

Οι αριθμοί δείχνουν την σειρά των βημάτων.



Εικόνα 7.5 – Αποστολή κρυπτογραφημένου μηνύματος sms

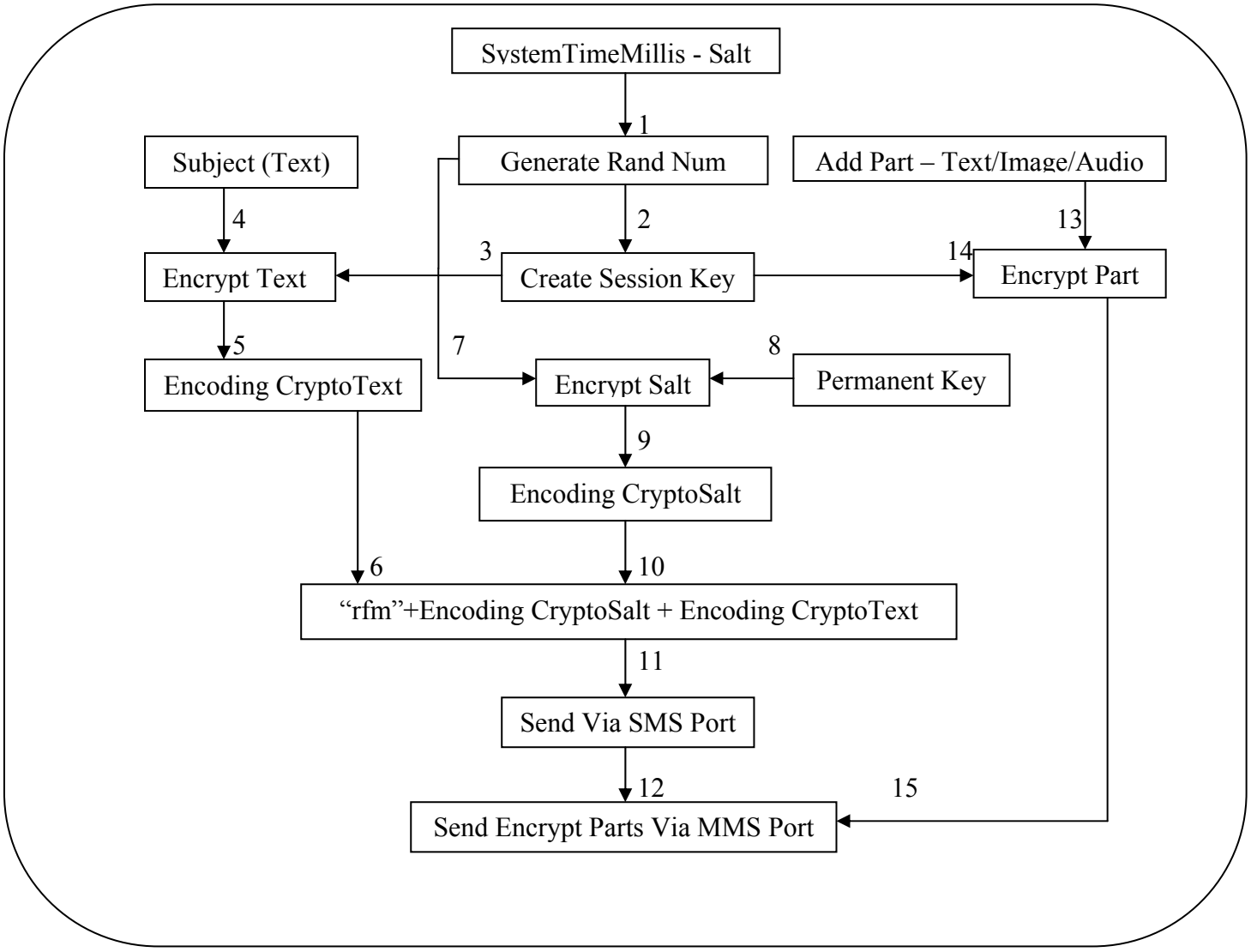
SystemTimeMillis (Salt) – Τα millisecond του συστήματος. Αριθμός τύπου long.
Generate Rand Num – Παραγωγή 16 τυχαίων αριθμών τύπου long.
Create Session Key – Δημιουργία κλειδιού συνόδου με την συνεισφορά των Rand Num.
Encrypt Salt – Κρυπτογράφηση του σπόρου (Salt).
Permanent Key – Μόνιμο κλειδί.
Text – Αρχικό μήνυμα.
Encrypt Text – Κρυπτογραφημένο κείμενο.
Encoding CryptoSalt – Κωδικοποίηση του κρυπτογραφημένου σπόρου.
Encoding CryptoText – Κωδικοποίηση του κρυπτογραφημένου κειμένου.
Send – Αποστολή.



Εικόνα 7.6 – Παραλαβή κρυπτογραφημένου μηνύματος sms

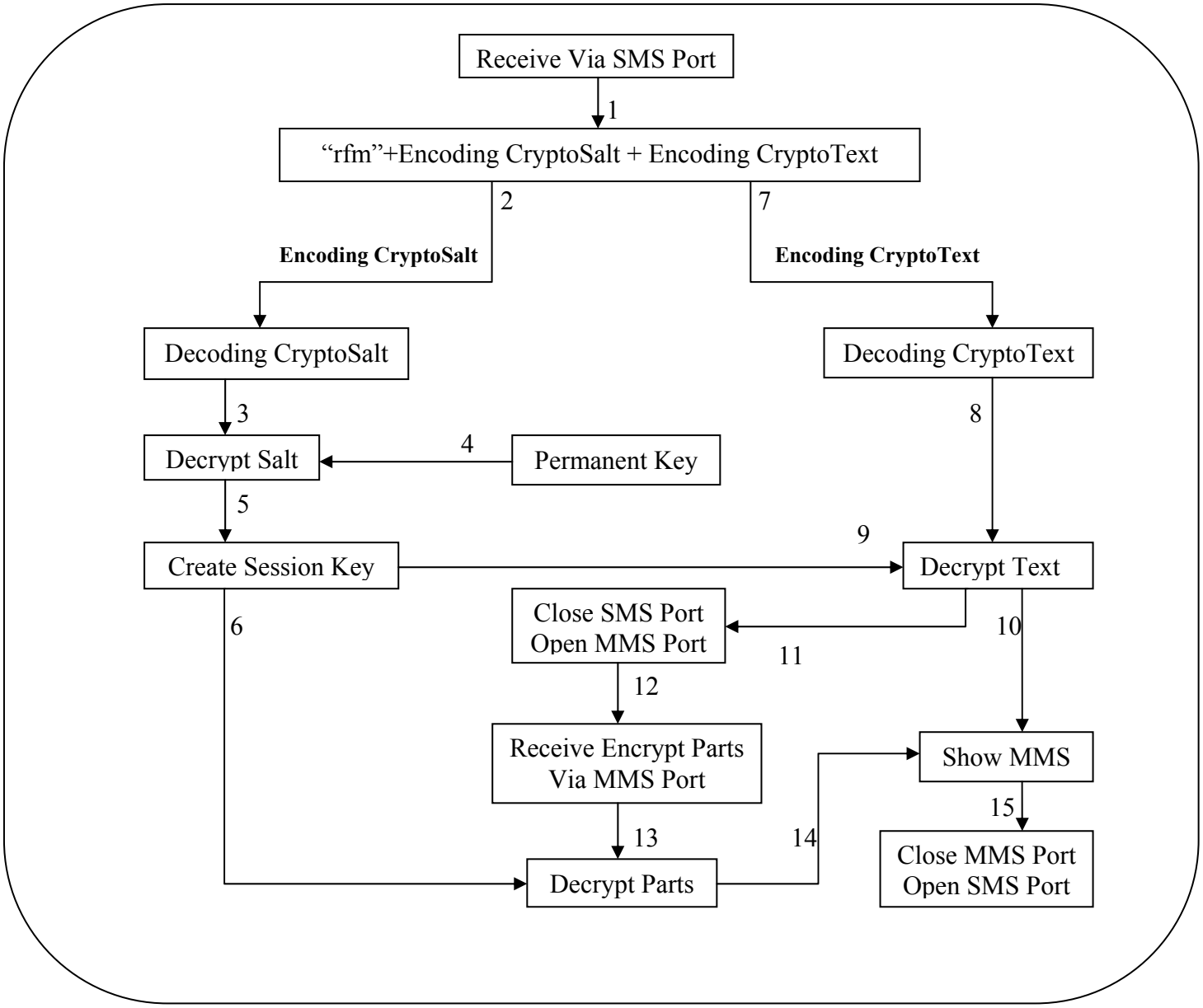
Decoding CryptoSalt – Αποκωδικοποίηση κρυπτογραφημένου σπόρου.
Decrypt Salt – Αποκρυπτογράφηση σπόρου με το μόνιμο κλειδί.
Decoding CryptoText – Αποκωδικοποίηση κρυπτοκειμένου.
Decrypt Text – Αποκρυπτογράφηση αρχικού κειμένου.

ΑΠΟΣΤΟΛΗ MMS



Εικόνα 7.7 - Αποστολή κρυπτογραφημένου μηνύματος mms

Subject (Text) – Το θέμα του MMS
Add Part (Text/Image/Audio) – Προσθήκη μερών στο μήνυμα (Κείμενο/Εικόνα/Ήχος)
Encrypt Part – Κρυπτογράφηση των μερών.
Send Via SMS Port – Αποστολή μέσω της θύρας των SMS.
Send Encrypt Parts Via MMS Port – Αποστολή των κρυπτογραφημένων μερών μέσω της θύρας των MMS.



Εικόνα 7.8 – Παραλαβή κρυπτογραφημένου μηνύματος MMS

Receive Via SMS Port – Λήψη μέσω της θύρας SMS.
Receive Encrypt Parts Via MMS Port – Λήψη των κρυπτογραφημένων μερών μέσω της θύρας MMS.
Decrypt Parts – Αποκρυπτογράφηση των μερών του MMS.
Show MMS – Εμφάνιση μηνύματος.
Close/Open SMS/MMS Port – Άνοιγμα / κλείσιμο θύρας SMS/MMS.

Ακολουθεί ο κώδικας.

```
package MobileSecurity;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.lcdui.*;
import java.security.*;
import javax.crypto.*;
import javax.crypto.spec.*;
import java.util.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class Cryptografy {

   SecretKeySpec key, sessionKey;
   Cipher cipher;
   final static String salt = new String("j#s2j6&jr$@6yfl)");
   byte[] keyBits = new byte[16];
   long sessionKeyString;
```

Στην μεταβλητή salt αποθηκεύεται το αλφαριθμητικό που χρησιμοποιείται για την παραγωγή του μόνιμου κλειδιού.

```
   public Cryptografy() {
      this.keyBits = salt.getBytes();


 public void createKey(){
      sessionKeyString = System.currentTimeMillis();
      Random random = new Random(sessionKeyString);
      byte[] sessionKeyByte = new byte[16];
      key = new SecretKeySpec(keyBits, 0, keyBits.length,"AES");
      long f;
      int i=0;
      do{
         f = random.nextLong();
         sessionKeyByte[i] = (byte)f;
         i++;
      }while(i < 16 );
      random = null;
      sessionKey = new SecretKeySpec(sessionKeyByte, 0, sessionKeyByte.length,
"AES");
   }
```

Η μέθοδος αυτή δημιουργεί ένα μυστικό κλειδί συνόδου παίρνοντας έναν τυχαίο αρχικό αριθμό ως σπόρο από τα millisecond του συστήματος.

```
public String encryption(String plainText){
    String encryptKey = encrypter(String.valueOf(sessionKeyString), key);
    String encryptSMS = encrypter(plainText, sessionKey);
    int length = encryptKey.length();
    return encryptKey + encryptSMS;
}
```

Εδώ καλείται η μέθοδος encrypter() η οποία κρυπτογραφεί τον σπόρο με το μόνιμο κλειδί και στην συνέχεια το μήνυμα με το κλειδί συνόδου. Τα δύο κρυπτογραφήματα ενώνονται σε ένα αλφαριθμητικό όπου προηγείται ο σπόρος.

```
public String decryption(String cryptoText){
    key = new SecretKeySpec(keyBits, 0, keyBits.length,"AES");
    byte[] sessionKeyByte = new byte[16];
    int i = 0;
    int dots = 0;
    do{
        i++;
        if(cryptoText.charAt(i)=='.')
            dots++;
    }while(dots < 16);
    String encryptKey = decrypter(cryptoText.substring(0, i), key);
    Random random = new Random(Long.parseLong(encryptKey));
    long f;
    int k=0;
    do{
        f = random.nextLong();
        sessionKeyByte[k] = (byte)f;
        k++;
    }while(k < 16 );
    random = null;
    sessionKey = new SecretKeySpec(sessionKeyByte, 0, 16, "AES");
    String decryptSMS = decrypter(cryptoText.substring(i), sessionKey);
    return decryptSMS;
}
```

Η μέθοδος αυτή διαχωρίζει και αποκρυπτογραφεί τον σπόρο, παράγει το προσωρινό κλειδί και τέλος αποκρυπτογραφεί με αυτό τα δεδομένα του μηνύματος.

```
private String encrypter(String text, SecretKeySpec key){
    int size = 16;
    byte[] plain = new byte[size];
    byte[] encrypt = new byte[size];
    String encryptStr = new String("");
```

```
    try{
        String temp = new String();

        int index = 0;
        int cycles;

        cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(cipher.ENCRYPT_MODE, key);
        temp = text.substring(index*size);
        plain = temp.getBytes();
        cycles = plain.length / size;
        while(index < cycles)
        {
            temp = text.substring(index*size, (index+1)*size);
            plain = temp.getBytes();
            cipher.update(plain, 0, plain.length, encrypt, 0);
            temp = byteToString(encrypt);
            encryptStr = encryptStr + temp;
            temp = null;
            index++;
        }
        temp = text.substring(index*size);
        plain = temp.getBytes();
        cipher.doFinal(plain, 0, plain.length, encrypt, 0);
        cipher = null;
        encryptStr = encryptStr + byteToString(encrypt);
    }catch(Exception ex){
    }

    return encryptStr;
}
```

Αυτή είναι η μέθοδος που δημιουργεί το αντικείμενο cipher που πραγματοποιεί την κρυπτογράφηση του δοθέντος κειμένου. Μετά την διαδικασία, το αποτέλεσμα μετατρέπεται σε String με την κωδικοποίηση που περιγράφεται παραπάνω και αυτό το αλφαριθμητικό επιστρέφεται από την μέθοδο.

```
private String decrypter(String text, SecretKeySpec key){
    int count = 0;
    for(int i=0; i<text.length(); i++){
        if(text.charAt(i)=='.')
            count++;
    }
    byte[] cryptotext = new byte[count + 32];
    byte[] decryptotext = new byte[count + 16];
    byte[] temp = new byte[16];
    byte[] temp2 = new byte[16];
    String str = new String();
```

```
try{
   text = text + ".";
   int b, e=0, i=0, num, index=0;
   String tmp = new String();
   do{
      b = i+1;
      e = b;
      do{
         e++;
      }while(text.charAt(e)!='.');
      tmp = text.substring(b,e);
      num = Integer.parseInt(tmp);
      cryptotext[index] = (byte)num;
      index++;
      i = e;
   }while(i<text.length()-1);
   int cycles;
   cycles = index / 16;
   if(index % 16 == 0)
      cycles--;
   index = 0;
   cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
   cipher.init(cipher.DECRYPT_MODE, key);

   boolean finalDec = true;
   while(index < cycles)
   {
      finalDec = false;
      for(i=0; i<16; i++){
          temp[i] = cryptotext[i+(index*16)];
      }
      cipher.update(temp, 0, temp.length, temp2, 0);
      if(index!=0){
         for(i=0; i<16; i++){
            decryptotext[i+((index-1)*16)] = temp2[i];
         }
      }
      index++;
   }

   for(i=0; i<16; i++){
          temp[i] = cryptotext[i + index*16];
      }

   byte[] temp32 = new byte[32];
   cipher.doFinal(temp, 0, temp.length, temp32, 0);
   cipher = null;
   if(finalDec)
      index++;
   for(i=0; i<32; i++){
```

```
            decryptotext[(index-1)*16 + i] = temp32[i];
        }
        i = 0;

        String finalText = new String(decryptotext);
        do{
            i++;
        }while(finalText.charAt(i)!='\u0000');
        str = finalText.substring(0,i);
    }catch(Exception ex){

    }

    return str;
}
```

Ισχύουν τα ίδια με την από πάνω μέθοδο μόνο που εδώ γίνεται η αποκρυπτογράφηση. Επίσης γίνεται και η αποκωδικοποίηση του αλφαριθμητικού που λήφθηκε. Οι δύο παραπάνω μέθοδοι χρησιμοποιούνται στα SMS και στο subject των MMS, όπου η κρυπτογράφηση και η αποκρυπτογράφηση γίνεται με δεδομένα τύπου string.

```
private String byteToString(byte[] pinByte){
    StringBuffer strBuf = new StringBuffer();
    for(int i=0; i<pinByte.length; i++){
        strBuf.append('.');
        strBuf.append((int)pinByte[i]);
    }return strBuf.toString();
}
```

Εδώ γίνεται η μετατροπή και η κωδικοποίηση των κρυπτογραφημένων byte σε string.

```
public SecretKeySpec getKey(){
    return sessionKey;
}
```

Μέθοδος που επιστρέφει το κλειδί συνόδου.

```
public byte[] encryptionData(byte[] data, SecretKeySpec keyMMS){
    int size = 16;
    int index = 0;
    int cycles;

    cycles = data.length / size;
    byte[] encryptData = new byte[cycles*16 + 32];
    try{
        byte[] plain = new byte[size];
        byte[] encrypt = new byte[size];
```

```
        cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(cipher.ENCRYPT_MODE, keyMMS);

        int j;
        while(index < cycles)
        {
           j = 0;
           for(int i=index*size; i<(index+1)*size; i++){
              plain[j] = data[i];
              j++;
           }
           cipher.update(plain, 0, plain.length, encrypt, 0);
           j = 0;
           for(int i=index*size; i<(index+1)*size; i++){
              encryptData[i] = encrypt[j];
              j++;
           }
           index++;
        }
        plain = null;
        plain = new byte[size];
        j = 0;
        for(int i=index*size; i<data.length; i++){
           plain[j] = data[i];
           j++;
        }
        encrypt = new byte[size*2];
        cipher.doFinal(plain, 0, plain.length, encrypt, 0);
        cipher = null;
        j = 0;
        for(int i=index*size; i<(index+1)*size + 16; i++){
           encryptData[i] = encrypt[j];
           j++;
        }
     }catch(Exception ex){
     }
     return encryptData;
  }
```

Εδώ γίνεται η κρυπτογράφηση των μερών των μηνυμάτων MMS. Τα δεδομένα εδώ είναι τύπου byte, όποτε δεν χρειάζεται κωδικοποίηση για να σταλούν.

```
  public byte[] decryptionMMS(byte[] cipherData, SecretKeySpec keyMMS){
     int size = 16;
     byte[] decryptData = new byte[cipherData.length];

     try{
        int index = 0;
        int cycles;
```

```
    byte[] encrypt = new byte[size];
    byte[] decrypt = new byte[size];

    cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
    cipher.init(cipher.DECRYPT_MODE, keyMMS);
    cycles = (cipherData.length - 16) / size;
    cycles--;
    int j;
    while(index < cycles)
    {
       j = 0;
       for(int i=index*size; i<(index+1)*size; i++){
          encrypt[j] = cipherData[i];
          j++;
       }
       cipher.update(encrypt, 0, encrypt.length, decrypt, 0);
       j = 0;
       for(int i=index*size; i<(index+1)*size; i++){
          decryptData[i] = decrypt[j];
          j++;
       }
       index++;
    }
    encrypt = new byte[size*2];
    decrypt = new byte[size*2];

    j = 0;
    for(int i=index*size; i<(index+1)*size + 16; i++){
       encrypt[j] = cipherData[i];
       j++;
    }
    cipher.doFinal(encrypt, 0, encrypt.length, decrypt, 0);
    cipher = null;
    j = 0;
    for(int i=index*size; i<(index+1)*size + 16; i++){
       decryptData[i] = decrypt[j];
       j++;
    }
    if(cycles>0){
       int i;
       for(i=16; i<decryptData.length; i++){
          decryptData[i-16] = decryptData[i];
       }
       for(int k=i-16; k<decryptData.length; k++){
          decryptData[k]=0;
       }
    }
}catch(Exception ex){
}
return decryptData;
```

```
    }
}
```

Εδώ γίνεται η αποκρυπτογράφηση των μερών των MMS. Όπως και από πάνω τα δεδομένα είναι τύπου byte και άρα δεν χρειάζονται αποκωδικοποίηση.

7.4 Κλάση MobileSecurity

Η κλάση αυτή είναι η κεντρική εφαρμογή. Είναι η μόνη μέσα στο project η οποία είναι τύπου MIDLET, δηλαδή εκτελείται κατευθείαν από τον προσομοιωτή. Μέσα στην κλάση αυτή προστέθηκαν μερικές από τις κλάσεις αποστολής και λήψης SMS και MMS ως εσωτερικές μέθοδοι. Οι κλάσεις που βρίσκονται στα δύο τελευταία πακέτα καλούνται από τις μεθόδους της αρχικής κλάσης. Με κόκκινα γράμματα σημειώνονται οι εντολές που σχετίζονται με την κρυπτογράφηση και την αποκρυπτογράφηση των μηνυμάτων.

```
package MobileSecurity;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.wireless.messaging.*;
import java.io.*;
import sms.*;
import mms.*;
import javax.crypto.spec.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class MobileSecurity extends MIDlet
        implements CommandListener, MessageListener {
```

Από εδώ ξεκινάει ο κώδικας της κλάσης. Το “extends” σημαίνει ότι η κλάση είναι υποκλάση της MIDlet και ως εκ’ τούτου κληρονομεί όλες τις μεθόδους της. Το “implements”(διασύνδεση) επιτρέπει την υλοποίηση μεθόδων μέσα στην παρούσα κλάση, που υπάρχουν όμως και στην διασύνδεση CommandListener. Με τις μεθόδους της συγκεκριμένης διασύνδεσης ορίζονται οι εντολές που θα εκτελούνται κατά το πάτημα κάποιου κουμπιού στον προσομοιωτή.

Η διασύνδεση MessageListener επιτρέπει την δημιουργία ενός αντικειμένου τύπου Listener το οποίο τοποθετείται σε μία συγκεκριμένη πόρτα και δίνει την δυνατότητα λήψης και επεξεργασίας εισερχόμενων μηνυμάτων που στέλνονται σε αυτήν την πόρτα.

```
MessageConnection conn;
boolean sms;
```

Διάφορες μεταβλητές

```
//SMS send variables
TextBox destinationAddressBox;
Alert errorMessageAlert;
Alert sendingMessageAlert;
SMSSender sender;
String smsPort;
String subject2 = new String("");
```

Μεταβλητές που χρησιμοποιούνται κατά την αποστολή των SMS.

```
//SMS Receive variables
String senderAddress;
Command replyCommand = new Command("Reply", Command.OK, 1);
```

Μεταβλητές που χρησιμοποιούνται κατά την λήψη των SMS.

```
//MMS send variables
Cryptografy generalCrypto;
Command CMD_SEND = new Command("Send", Command.ITEM, 1);
Command CMD_ADD_PART = new Command("Add Part", Command.ITEM, 1);
PartsDialog partsDialog;
StringItem partsLabel;
Displayable resumeScreen = null;
MMSMessage message;
TextField subjectField, destinationField;
String appID;
byte[][] audioByte;
String[] audioName, audioExt;
int lenAudByte;
SecretKeySpec sessionKeyMMS;
```

Μεταβλητές που χρησιμοποιούνται κατά την αποστολή των MMS.

```
public MobileSecurity() {
    initialize();
}
```

Constructor της κλάσης. Καλεί την μέθοδο initialize().

```
public void connReadySMS(){
   String smsConnection = "sms://:50000";
   try {
      if (conn!=null){
         conn.close();
         conn = null;
      }
      conn = (MessageConnection) Connector.open(smsConnection);
      conn.setMessageListener(this);
   } catch (IOException ioe) {
      errorMessageAlert.setString(ioe.toString());
      getDisplay().setCurrent(errorMessageAlert);
   }
}
```

Η μέθοδος αυτή καλείται αρχικά από την initialize(). Αυτό που κάνει είναι να προετοιμάζει το κινητό κατά την έναρξη της εφαρμογής ώστε να μπορεί να δεχτεί αυτοματα μηνύματα SMS.

```
public void connReadyMMS() {
   String mmsConnection = "mms://:example.mms.MMSDemo";
   try {
      if (conn!=null){
         conn.close();
         conn = null;
      }
      conn = (MessageConnection) Connector.open(mmsConnection);
      conn.setMessageListener(this);
   } catch (IOException ioe) {
      errorMessageAlert.setString(ioe.toString());
      getDisplay().setCurrent(errorMessageAlert);
   }
}
```

Κάνει κάτι αντίστοιχο με την προηγούμενη μέθοδο μόνο που εδώ η σύνδεση γίνεται για την λήψη MMS. Και στις δύο μεθόδους πριν από το άνοιγμα της σύνδεσης, γίνεται έλεγχος για το αν υπάρχει ήδη ανοιχτή σύνδεση και αν ναι, τότε την κλείνει και την θέτει σε τιμή null, δηλαδή κενή τιμή.

```
public void commandAction(Command command, Displayable displayable) {
   if (displayable == list1) {
      if (command == list1.SELECT_COMMAND) {
         switch (get_list1().getSelectedIndex()) {
            case 0:
               SMSSend();
               break;
            case 1:
               if(partsDialog != null)
```

```
                    partsDialog = null;
                MMSSend();
                break;
            }
        } else if (command == exitCommand2) {
            exitMIDlet();
        } else if (command == okCommand1) {
            if (get_list1().getSelectedIndex() == 0) {
                SMSSend();
            }else {
                if(partsDialog != null)
                    partsDialog = null;
                MMSSend();
            }
        }
    }
    else if(command == exitCommand2){
        initialize();
    }
    else if(sms) {
        try {
            if (command == exitCommand2 || command ==
Alert.DISMISS_COMMAND) {
                initialize();
            } else if (command == okCommand1) {
                promptAndSend();
            } else if (command == replyCommand) {
                reply();
            }
        } catch (Exception ex) {
            errorMessageAlert.setString(ex.toString());
            getDisplay().setCurrent(errorMessageAlert);
        }
    }else if(!sms) {
        try {
            if ((command == exitCommand2) || (command ==
Alert.DISMISS_COMMAND)) {
                initialize();
            } else if (command == CMD_ADD_PART) {
                if (partsDialog == null) {
                    partsDialog = new PartsDialog(this);
                }
                partsDialog.show();
            } else if (command == CMD_SEND) {
                connReadySMS();
                promptAndSendMMS();
            }else  if (command == PLAY){
                for(int a=0; a<audioName.length; a++){
                    audioName[a] = audioName[a] + audioExt[a];
                }
```

```
                AudioPlayer ap = new AudioPlayer(audioByte, audioName,
getDisplay());
                ap.startApp();
            } else{
                initialize();
            }
            } catch (Exception ex) {
                errorMessageAlert.setString(ex.toString());
                getDisplay().setCurrent(errorMessageAlert);
            }
        }
}
```

Η μέθοδος αυτή χειρίζεται όλες τις εντολές κουμπιών σε πολλές από τις οθόνες που εμφανίζονται κατά την εκτέλεση της εφαρμογής.

```
    private void initialize() {
        getDisplay().setCurrent(get_list1());
        connReadySMS();
    }
```

Είναι η πρώτη μέθοδος που εκτελείται κατά την εκκίνηση του MIDLET. Εμφανίζει το αρχικό μενού και με την κλήση της μεθόδου connReadySMS() προετοιμάζει το κινητό για λήψη SMS.

```
    public Display getDisplay() {
        return Display.getDisplay(this);
    }
```

Επιστρέφει την οθόνη που εμφανίζεται την συγκεκριμένη στιγμή.

```
    public void exitMIDlet() {
        getDisplay().setCurrent(null);
        destroyApp(true);
        notifyDestroyed();
    }
```

Εκτελείται κατά την έξοδο από την εφαρμογή.

```
public List get_list1() {
        if (list1 == null) {
            list1 = new List(null, Choice.IMPLICIT, new String[] {
                "SMS Send",
                "MMS Send"
            }, new Image[] {
                null,
```

```
            null
        });
        list1.addCommand(get_okCommand1());
        list1.addCommand(get_exitCommand2());
        list1.setCommandListener(this);
        list1.setSelectedFlags(new boolean[] {
            true,
            false
        });
    }
    return list1;
}
```

Καλείται από την initialize() και εμφανίζει το αρχικό μενού των δύο επιλογών "SMSSend" και "MMSSend".

```
public Command get_okCommand1() {
    if (okCommand1 == null) {
        okCommand1 = new Command("OK", Command.OK, 1);
    }
    return okCommand1;
}
```

Επιστρέφει ένα αντικείμενο της εντολής "Ok".

```
public Command get_exitCommand2() {
    if (exitCommand2 == null) {
        exitCommand2 = new Command("Exit", Command.EXIT, 1);
    }
    return exitCommand2;
}
```

Επιστρέφει ένα αντικείμενο της εντολής "Exit".

```
public void notifyIncomingMessage(MessageConnection conn) {

    Message msg;
    String header = new String();
    this.conn = conn;
    try {
        msg = conn.receive();
        if (msg instanceof TextMessage) {
            sms = true;
            header = ((TextMessage)msg).getPayloadText();
            String tmp = header.substring(0,3);
            if ((tmp.compareTo("sms")) == 0){
                SMSReceive(msg);
```

```
            }else if ((tmp.compareTo("rfm")) == 0){  //rfm = ready for mms
               generalCrypto = new Cryptografy();
               subject2 = generalCrypto.decryption(header.substring(3));
               sessionKeyMMS = generalCrypto.getKey();
               connReadyMMS();
            }
         }else if (msg instanceof MultipartMessage) {
            sms = false;
            MMSReceive(msg);
            connReadySMS();
         }
      }catch(IOException ioe) {
         errorMessageAlert.setString(ioe.toString());
         getDisplay().setCurrent(errorMessageAlert);
      }
   }
```

Μέθοδος που εκτελείται κατά την λήψη κάποιου μηνύματος. Για να μπορεί να γίνει λήψη MMS, πρέπει ο παραλήπτης να ειδοποιηθεί μέσω της ανοιχτής σύνδεσης για τα SMS. Για τον λόγο αυτό, κατά την λήψη ενός μηνύματος κειμένου ελέγχονται τα τρία πρώτα γράμματα, ώστε ανάλογα με το τι γράμματα είναι, να γίνει και η κατάλληλη ενέργεια. Στην περίπτωση αυτή, στα μηνύματα SMS προστίθεται από τον αποστολέα το μοτίβο “sms”, ενώ αν το μοτίβο είναι το “rfm”, τότε ο παραλήπτης κλείνει την σύνδεση SMS και ανοίγει μία άλλη για την λήψη MMS. Έτσι επιτυγχάνεται η λήψη των MMS.

```
   public void SMSSend() {

      Display display = getDisplay();

      sms = true;
      smsPort = "50000";
      destinationAddressBox = new TextBox("Destination Address?",
         null, 16, TextField.PHONENUMBER);
      destinationAddressBox.addCommand(exitCommand2);
      destinationAddressBox.addCommand(okCommand1);
      destinationAddressBox.setCommandListener(this);

      errorMessageAlert = new Alert("SMS", null, null, AlertType.ERROR);
      errorMessageAlert.setTimeout(5000);

      sendingMessageAlert = new Alert("SMS", null, null, AlertType.INFO);
      sendingMessageAlert.setTimeout(5000);
      sendingMessageAlert.setCommandListener(this);

      sender = new SMSSender(smsPort, display, destinationAddressBox,
         sendingMessageAlert);
      resumeScreen = destinationAddressBox;
      display.setCurrent(resumeScreen);
   }
```

Στο προηγούμενο project ήταν ο Constructor της κλάσης SMSSend. Εδώ έχει τοποθετηθεί ως μέθοδος στο αρχικό MIDLET.

```
public void promptAndSend() {
    String address = destinationAddressBox.getString();
    if (!isValidPhoneNumber(address)) {
      errorMessageAlert.setString("Invalid phone number");
      getDisplay().setCurrent(errorMessageAlert, destinationAddressBox);
      return;
    }
    String statusMessage = "Sending message to " + address + "...";
    sendingMessageAlert.setString(statusMessage);
    sender.promptAndSend("sms://" + address);
}
```

Εδώ ελέγχεται ο αριθμός του αποστολέα και αν είναι σωστός τότε καλείται η μέθοδος promptAndSend() που βρίσκεται στην κλάση SMSSender για να αποσταλεί το μήνυμα.

```
private static boolean isValidPhoneNumber(String number) {
      char[] chars = number.toCharArray();
    if (chars.length == 0) {
      return false;
    }
    int startPos = 0;
    // initial '+' is OK
    if (chars[0] == '+') {
      startPos = 1;
    }
    for (int i = startPos; i < chars.length; ++i) {
      if (!Character.isDigit(chars[i])) {
        return false;
      }
    }
    return true;
}
```

Αυτή είναι η μέθοδος που ελέγχει τον αριθμό του παραλήπτη. Ο αριθμός είναι σωστός μόνο όταν αποτελείται από αριθμητικά ψηφία με εξαίρεση το πρώτο στη σειρά όπου θα μπορούσε να είναι και το σύμβολο “+”.

```
public void MMSSend() {
    sms = false;
    appID = "example.mms.MMSDemo";

    Form mainForm = new Form("New MMS");
```

```
        subjectField = new TextField("Subject:",
            null, 40, TextField.ANY);
        mainForm.append(subjectField);

        destinationField = new TextField("Destination Address: ",
            "mms://", 256, TextField.ANY);
        mainForm.append(destinationField);

        partsLabel = new StringItem("Parts:", "0");
        mainForm.append(partsLabel);

        mainForm.addCommand(exitCommand2);
        mainForm.addCommand(CMD_SEND);
        mainForm.addCommand(CMD_ADD_PART);
        mainForm.setCommandListener(this);

        errorMessageAlert = new Alert("MMS", null, null, AlertType.ERROR);
        errorMessageAlert.setTimeout(5000);

        sendingMessageAlert = new Alert("MMS", null, null, AlertType.INFO);
        sendingMessageAlert.setTimeout(5000);
        sendingMessageAlert.setCommandListener(this);

        resumeScreen = mainForm;
        generalCrypto = new Cryptografy();
        generalCrypto.createKey();
        message = new MMSMessage();
        getDisplay().setCurrent(resumeScreen);
    }
```

Στο αρχικό project ήταν ο Constructor της κλάσης MMSSend. Εδώ υπάρχει ως μέθοδος. Δημιουργείται η κεντρική οθόνη όπου μπορούν να προστεθούν μέρη στο μήνυμα, να οριστεί τίτλος θέματος(subject) και να δοθεί ο αριθμός του παραλήπτη. Επίσης, φτιάχνεται και ένα αντικείμενο της κλάσης MMSMessage όπου θα ενσωματωθούν όλα τα παραπάνω χαρακτηριστικά του μηνύματος και στην συνέχεια θα σταλούν στον παραλήπτη. Τέλος έχουν προστεθεί εντολές όπου δημιουργείται ένα αντικείμενο της κλάσης Cryptografy και παράγεται ένα νέο κλειδί συνόδου.

```
    public void show() {
        partsLabel.setText(Integer.toString(partsDialog.counter));
        getDisplay().setCurrent(resumeScreen);
    }
```

Η κλήση της παραπάνω μεθόδου επαναφέρει την αρχική κεντρική οθόνη αποστολής μετά από εισαγωγή εικόνας ή τμήματος κειμένου. Επίσης, εμφανίζεται και ο αριθμός των συνολικών μερών που έχουν προστεθεί στο μήνυμα μέχρι εκείνη την στιγμή.

```
public MMSMessage getMessage() {
    return message;
}
```

Μέθοδος η οποία επιστρέφει την μεταβλητή message.

Και οι έξι παραπάνω μέθοδοι έχουν μεταφερθεί σχεδόν αυτούσιες από το προηγούμενο project και έχουν ενσωματωθεί ως εσωτερικές μέθοδοι στο MIDLET.

```
private void promptAndSendMMS() {
    try {
        subject2 = generalCrypto.encryption(subjectField.getString());
        String address = destinationField.getString();
        conn.close();
        conn = null;
        message.setSubject("");
        message.setDestination(address);
        String statusMessage = "Sending message to " + address + "...";
        sendingMessageAlert.setString(statusMessage);
        new SendHeader("rfm" + subject2, address + ":50000", getDisplay(), message,
        appID).start();
    }catch (IllegalArgumentException iae) {
        errorMessageAlert.setString(iae.getMessage());
        getDisplay().setCurrent(errorMessageAlert);
    }catch (Exception e) {
        errorMessageAlert.setString(e.getMessage());
        getDisplay().setCurrent(errorMessageAlert);
    }
}
```

Είναι η μέθοδος promptAndSend του προηγούμενου project, αλλά έχει προστεθεί στον τίτλο του το “MMS” για να ξεχωρίζει από την αντίστοιχη μέθοδο των SMS που έχει το ίδιο όνομα.

Έχει προστεθεί στο τέλος μία εντολή όπου καλείται η μέθοδος sendHeader() για να σταλεί το μοτίβο “rfm” μέσα από την ανοιχτή σύνδεση για τα SMS, ώστε ο παραλήπτης να ανοίξει μία σύνδεση για να δεχτεί το MMS.

```
public void SMSReceive(Message msg) {
    Alert content;
    Display display;

    smsPort = "50000";
    display = getDisplay();

    content = new Alert("SMS Receive");
    content.setTimeout(Alert.FOREVER);
    content.addCommand(exitCommand2);
```

```
        content.addCommand(replyCommand);
        content.setCommandListener(this);
        content.setString("Receiving...");

        sendingMessageAlert = new Alert("SMS", null, null, AlertType.INFO);
        sendingMessageAlert.setTimeout(5000);
        sendingMessageAlert.setCommandListener(this);

        sender = new SMSSender(smsPort, display, content, sendingMessageAlert);

        resumeScreen = content;
        if (msg != null) {
            senderAddress = msg.getAddress();
            content.setTitle("From: " + senderAddress);
            if (msg instanceof TextMessage) {

                String temp = new String();
                temp =((TextMessage)msg).getPayloadText();
                temp = temp.substring(3,temp.length());
                Cryptografy crypto = new Cryptografy();
                content.setString(crypto.decryption(temp));
            } else {
                StringBuffer buf = new StringBuffer();
                byte[] data = ((BinaryMessage)msg).getPayloadData();
                for (int i = 0; i < data.length; i++) {
                    int intData = (int)data[i] & 0xFF;
                    if (intData < 0x10) {
                        buf.append("0");
                    }
                    buf.append(Integer.toHexString(intData));
                    buf.append(' ');
                }

            }
            display.setCurrent(content);
        }
    }
```

Μέθοδος παραλαβής των μηνυμάτων SMS. Στον κώδικα έχουν προστεθεί δύο εντολές όπου πραγματοποιούν την αποκρυπτογράφηση του μηνύματος.

```
    private void reply() {
        // remove the leading "sms://" for diplaying the destination address
        String address = senderAddress.substring(6);
        String statusMessage = "Sending message to " + address + "...";
        sendingMessageAlert.setString(statusMessage);
        sender.promptAndSend(senderAddress);
    }
```

Μέθοδος που υπάρχει στο αρχικό project. Καλείται από το πάτημα του κουμπιού "Reply" μετά από λήψη κάποιου μηνύματος. Η μέθοδος αυτή επιτρέπει να σταλλεί απάντηση σε αυτόν που έστειλε το μήνυμα. Με την εκτέλεση της μεθόδου αυτής εμφανίζεται η οθόνη εισαγωγής μηνύματος όπου γράφεται το κείμενο προς αποστολή και στη συνέχεια με το πάτημα του κουμπιού "Send", αποστέλλεται αυτό το μήνυμα στον αποστολέα του μηνύματος που λήφθηκε προηγουμένως.

```
public void MMSReceive(Message msg) {

    final Command CMD_EXIT  = new Command("Exit", Command.EXIT, 2);
    Form content;
    Display display;

    appID = "example.mms.MMSDemo";

    content = new Form("MMS Receive");
    content.addCommand(CMD_EXIT);
    content.setCommandListener(this);
    content.append("Receiving...");

    sendingMessageAlert = new Alert("MMS", null, null, AlertType.INFO);
    sendingMessageAlert.setTimeout(5000);
    sendingMessageAlert.setCommandListener(this);

    resumeScreen = content;

    display = getDisplay();
    display.setCurrent(content)
    try {
       if (msg != null) {
          senderAddress = msg.getAddress();
          content.deleteAll();
          String titleStr = senderAddress.substring(6);
          int colonPos = titleStr.indexOf(":");
          if (colonPos != -1) {
             titleStr = titleStr.substring(0, colonPos);
          }
          content.setTitle("From: " + titleStr);
          if (msg instanceof MultipartMessage) {
             MultipartMessage mpm = (MultipartMessage)msg;
             StringBuffer buff = new StringBuffer("Subject: ");
             buff.append(subject2);
             buff.append("\nDate: ");
             buff.append(mpm.getTimestamp().toString());
             buff.append("\nContent:");
             StringItem messageItem = new StringItem("Message",
                                                     buff.toString());
             messageItem.setLayout(Item.LAYOUT_NEWLINE_AFTER);
             content.append(messageItem);
```

```
MessagePart[] parts = mpm.getMessageParts();
if (parts != null) {
   lenAudByte = 0;
   int maxLenAudio = 0;
   for(int i=0; i<parts.length; i++){
      MessagePart mp = parts[i];
      if((mp.getMIMEType().compareTo("audio"))==0){
         lenAudByte++;
         if(maxLenAudio < mp.getContent().length)
            maxLenAudio = mp.getContent().length;
      }
   }
   audioByte = new byte[lenAudByte][maxLenAudio];
   audioName = new String[lenAudByte];
   audioExt = new String[lenAudByte];
   lenAudByte = 0;
   for (int i = 0; i < parts.length; i++) {
      buff = new StringBuffer();
      MessagePart mp = parts[i];
      buff.append("Content-Type: ");
      String mimeType = mp.getMIMEType();
      buff.append(mimeType);
      String contentLocation = mp.getContentLocation();
      buff.append("\nContent:\n");

      byte[] ba2 = mp.getContent();
      byte[] ba = generalCrypto.decryptionMMS(ba2, sessionKeyMMS);

      if (mimeType.equals("image/png")) {
         content.append(buff.toString());
         Image img = Image.createImage(ba, 0, ba.length);
         ImageItem ii = new ImageItem(contentLocation,
               img, Item.LAYOUT_NEWLINE_AFTER,
               contentLocation);
         content.append(ii);
      } else if (mimeType.equals("audio")){
         int j = 0;
         do{
            audioByte[lenAudByte][j] = ba[j];
            j++;
         }while(j<ba.length);
         int k=-1;
         do{
            k++;
         }while(contentLocation.charAt(k)!='.');
         audioName[lenAudByte] = contentLocation.substring(0,k);
         audioExt[lenAudByte] = contentLocation.substring(k);
         lenAudByte++;
      }
      else {
```

```
                    buff.append(new String(ba));
                    StringItem si = new StringItem(
                                "Part", buff.toString());
                    si.setLayout(Item.LAYOUT_NEWLINE_AFTER);
                    content.append(si);
                }
            }
          if (lenAudByte != 0){
            buff = new StringBuffer();
            buff.append("Content-Type: audio");
            buff.append("\nContent:\n");
            for(int i = 0; i < lenAudByte; i++){
               buff.append(audioName[i] + "\n");
            }

            StringItem si = new StringItem(
                  "Part", buff.toString());
            si.setLayout(Item.LAYOUT_NEWLINE_AFTER);
            content.addCommand(PLAY);
            content.setCommandListener(this);
            content.append(si);
          }
        }
      }
      display.setCurrent(content);
      }
    } catch (Exception ioe) {
      errorMessageAlert.setString(ioe.toString());
      getDisplay().setCurrent(errorMessageAlert);
    }
  }
```

Είναι η μέθοδος εμφάνισης των ληφθέντων μηνυμάτων MMS. Έχουν προστεθεί εντολές που πραγματοποιούν αποκρυπτογράφηση στο subject και στα μέρη του μηνύματος. Η μέθοδος παίρνει ως παράμετρο το μήνυμα msg και στην συνέχεια γίνεται η περαιτέρω επεξεργασία του και η εμφάνισή του.

```
  public void destroyApp(boolean unconditional) {
    try{
      if(conn != null){
        conn.close();
        conn = null;
      }
    }catch (IOException ioe) {
      errorMessageAlert.setString(ioe.toString());
      getDisplay().setCurrent(errorMessageAlert);
    }
  }
}
```

Κλείσιμο της σύνδεσης κατά τον τερματισμό της εφαρμογής.

7.5 Κλάση SMSSender

Η περιγραφή της υπάρχει στο αρχικό project. Εδώ παρατίθεται μόνο η μέθοδος run() λόγω προσθήκης εντολών. Οι υπόλοιπες μέθοδοι παραμένουν ως έχουν και γι' αυτό παραλείπονται. Οι επιπλέον εντολές σχετίζονται με την κρυπτογράφηση του SMS.

```
public void run() {

    String address = destinationAddress + ":" + smsPort;
    MessageConnection smsconn = null;

    try {
        /** Open the message connection. */
        smsconn = (MessageConnection)Connector.open(address);
        String str = new String();
        TextMessage txtmessage = (TextMessage)smsconn.newMessage(
            MessageConnection.TEXT_MESSAGE);
        txtmessage.setAddress(address);
        txtmessage.setPayloadText(messageBox.getString());
        str = (messageBox.getString());
        Cryptografy crypto = new Cryptografy();
        crypto.createKey();
        txtmessage.setPayloadText("sms" + crypto.encryption(str));
        smsconn.send(txtmessage);
        display.setCurrent(sendingScreen);
    } catch (Throwable t) {
        messageAlert.setString("Send error: " + t.getMessage());
        messageAlert.setTimeout(5000);
        display.setCurrent(messageAlert);
    }
    if (smsconn != null) {
        try {
            smsconn.close();
        } catch (IOException ioe) {
            messageAlert.setString("Connection close error: "
                    + ioe.getMessage());
            messageAlert.setTimeout(5000);
            display.setCurrent(messageAlert);
        }
    }
}
```

Προστίθεται στην αρχή το μοτίβο "sms", μη κρυπτογραφημένο, ώστε ο παραλήπτης να αναγνωρίσει ότι το μήνυμα είναι καθαρά μήνυμα SMS και όχι ειδοποίηση για λήψη MMS.

7.6 Κλάση PartDialog

Αυτή η κλαση ανηκει στο πακετο “mms” από το οποίο θα περιγραφούν μόνο οι κλάσεις PartDialog και SendHeader. Η περιγραφή της κλάσης υπάρχει ήδη από το αρχικό project. Εδώ θα παρουσιαστούν και θα περιγραφούν μόνο οι μέθοδοι που έχουν τροποποιηθεί. Οι πρόσθετες εντολές έχουν κόκκινο χρώμα και σχετίζονται με την κρυπτογράφηση.

```
public PartsDialog(MobileSecurity mmsSend) {
                this.mmsSend = mmsSend;

    String[] stringArray = {"Text", "Image", "Audio"};

    typeList = new List("Add Part: Type", Choice.EXCLUSIVE,
                        stringArray, null);
    typeList.addCommand(CMD_BACK);
    typeList.addCommand(CMD_NEXT);
    typeList.setCommandListener(this);

  }


    public void commandAction(Command c, Displayable s) {
      try {
        if (c == CMD_OK) {
          String encoding = "UTF-8";
          byte[] data = text.getString().getBytes(encoding);
          int cycles = data.length / 16;
          byte[] contents = new byte[cycles * 16 + 32];
          Cryptografy crypto = new Cryptografy();
          contents = crypto.encryptionData(data, mmsSend.getSessionKey());
          mmsSend.getMessage().addPart(
                  new MessagePart(contents, 0, contents.length,
                                  mimeType, "id" + counter,
                                  "contentLocation", encoding));
          counter++;
                                  mmsSend.show();
        } else if (c == CMD_CANCEL) {
                                  mmsSend.show();
        }
      } catch (Exception ex) {
        ex.printStackTrace();
      }
    }
  }
```

Ο κώδικας αυτός εκτελείται κατά το πάτημα του κουμπιού “Ok” και προσθέτει στο MMS ένα μέρος κειμένου.

```
    public void commandAction(Command c, Displayable s) {
      try {

        if (c == CMD_OK) {

          int index = cg.getSelectedIndex();
          String resouce = resouces[index];
          InputStream is = getClass().getResourceAsStream(resouce);

          byte[] data = new byte[is.available()];
          is.read(data);

          int cycles = data.length / 16;
          byte[] contents = new byte[cycles * 16 + 32];
          Cryptografy crypto = new Cryptografy();
          contents = crypto.encryptionData(data, mmsSend.getSessionKey());

          String contentLocation = imagesNames[index];
          mmsSend.getMessage().addPart(
                  new MessagePart(contents, 0, contents.length,
                                  mimeType, "id" + counter,
                                  contentLocation, null));
          counter++;
                                  mmsSend.show();
        } else if (c == CMD_CANCEL) {
                                  mmsSend.show();
        }
      } catch (Exception ex) {
        ex.printStackTrace();
      }
    }
  }
}
```

Εδώ ισχύουν ακριβώς τα ίδια με την παραπάνω περιγραφή με την διαφορά ότι εδώ γίνεται κρυπτογράφηση εικόνας και όχι κειμένου.

```
private class AudioDialog extends Form implements CommandListener {

    private ChoiceGroup cg;
    private String mimeType = "audio";
    private String[] resouces = {"/audio/beethoven.dat", "/audio/bark.wav"};
    private String[] audioNames = {"beethoven.bat", "bark.wav"};
    public AudioDialog() {
      super("Add audio");

      cg = new ChoiceGroup("Select audio", Choice.EXCLUSIVE,
                           audioNames, null);
      append(cg);
```

```
        append("MIME-Type: " + mimeType);

        addCommand(CMD_OK);
        addCommand(CMD_CANCEL);
        setCommandListener(this);
    }


    public void commandAction(Command c, Displayable s) {
        try {

            if (c == CMD_OK) {

                int index = cg.getSelectedIndex();
                String resouce = resouces[index];
                InputStream is = getClass().getResourceAsStream(resouce);
                byte[] data = new byte[is.available()];
                is.read(data);

                int cycles = data.length / 16;
                byte[] contents = new byte[cycles * 16 + 32];

                Cryptografy crypto = new Cryptografy();
                contents = crypto.encryptionData(data, mmsSend.getSessionKey());

                String contentLocation = audioNames[index];
                mmsSend.getMessage().addPart(
                        new MessagePart(contents, 0, contents.length,
                                        mimeType, "id" + counter,
                                        contentLocation, null));
                counter++;
                                        mmsSend.show();
            } else if (c == CMD_CANCEL) {
                                        mmsSend.show();
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Εδώ γίνεται κρυπτογράφηση των ηχητικών κομματιών και ισχύουν τα ίδια με την εικόνα και το κείμενο.

7.7 Κλάση SendHeader

Η κλάση αυτή εκτελείται κατά την αποστολή MMS και σκοπός της είναι να στέλνει ένα προειδοποιητικό μήνυμα SMS στον παραλήπτη, ώστε ο τελευταίος να κλείσει την σύνδεση που έχει ανοιχτεί για τα SMS και να ανοίξει μία άλλη για να δεχτεί το MMS που θα του στείλει στην συνέχεια ο αποστολέας.

```
package mms;
```

Δήλωση που δείχνει σε ποιο πακέτο ανήκει η κλάση αυτή.

```
import com.sun.midp.lcdui.DisplayAccess;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
import javax.microedition.io.*;
import javax.wireless.messaging.*;
import java.io.*;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class SendHeader extends Thread{

    MessageConnection conn;
     String header;
     String address;
     Alert errorMessageAlert;
     Display disp;
     boolean ok;
     MMSMessage message;
     String appID;
```

Διάφορες μεταβλητές.

```
  public SendHeader(String head, String addressa, Display display,
        MMSMessage message, String appID) {

      this.address = addressa;
      this.header = head;
      this.disp = display;
      this.message = message;
      this.appID = appID;
  }
```

Κατά την δημιουργία του αντικειμένου της κλάσης περνάνε οι εξής παράμετροι:

- Το κείμενο που θα σταλεί
- Η διεύθυνση αποστολής
- Η οθόνη του προσομοιωτή, ώστε να πάρει τον έλεγχο η κλάση αυτή
- Το αντικείμενο MMSMessage
- Και την μεταβλητή AppID

```
public void run(){
   if (address.charAt(0) == 'm') {
      char[] pin = address.toCharArray();
      pin[0] = 's';
      address = address.valueOf(pin);
   }
   conn = null;
   try {
      conn = (MessageConnection) Connector.open(address);
   }catch(IOException ioe){
      errorMessageAlert.setString(ioe.toString());
      disp.setCurrent(errorMessageAlert);
   }
   TextMessage msg = (TextMessage)conn.newMessage(
      MessageConnection.TEXT_MESSAGE);

   msg.setAddress(address);
   msg.setPayloadText(header);

   try {
      conn.send(msg);
      conn.close();
      conn = null;
      new SenderThread(message, appID).start();
   }catch(IOException ioe) {
      errorMessageAlert.setString(ioe.toString());
      disp.setCurrent(errorMessageAlert);
   }
}
```

Η κλάση εκτελείται ως νήμα και γι' αυτό υπάρχει η μέθοδος run(). Σε αυτήν την μέθοδο απλά ανοίγει μία σύνδεση για SMS και στην συνέχεια, αφού δημιουργηθεί ένα αντικείμενο της κλάσης TextMessage, ενσωματώνεται σε αυτό η διεύθυνση του παραλήπτη και η ειδοποίηση που πρέπει να σταλεί. Επειδή την διεύθυνση την παίρνει έτοιμη από το όρισμα, θα πρέπει να αντικατασταθεί το "m" που βρίσκεται στην πρώτη θέση με το γράμμα "s". Αυτό γίνεται επειδή η διεύθυνση προορίζεται για μήνυμα MMS, ενώ θα πρέπει να σταλεί SMS. Μετά την αποστολή της ειδοποίησης κλείνει η σύνδεση και ξεκινά η εκτέλεση του νήματος για την αποστολή του MMS.

7.8 Κλάση AudioPlayer

Η κλάση αυτή προέρχεται από το project AudioPlayer που υπάρχει έτοιμο στα Netbeans. Κατά την παραλαβή ενός MMS που περιέχει και ήχο, εμφανίζεται κάτω δεξιά μία επιπλέον εντολή κουμπιού με το όνομα "PLAY". Κατά το πάτημα του κουμπιού καλείται ο Constructor της παρακάτω κλάσης και δημιουργείται ένα αντικείμενο το οποίο θα αναπαράγει στην συνέχεια τα κομμάτια που παραλήφθηκαν.

```
import java.io.ByteArrayInputStream;
import javax.microedition.lcdui.*;
import java.util.Vector;
import java.io.InputStream;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class AudioPlayer implements CommandListener {

    private static PlayerCanvas   playerGUI = null;
    private static List           theList;
    private static Vector         urls;
    private String[]              audioNm;
    private byte[][]              audioBt;
    InputStream                   bais;
    Alert                         errorMessageAlert;
    private Command               exitCommand = new Command("Exit",
                                      Command.EXIT, 1);
    private Command               playCommand = new Command("Play",
                                      Command.ITEM, 1);
    private Display               display;
    private Displayable           previous;
```

Τοπικές μεταβλητές της κλάσης. Στον πίνακα τύπου string με το όνομα "audioNm" αποθηκεύονται τα ονόματα των κομματιών που έχουν ληφθεί, μαζί με τις επεκτάσεις τους. Στον πίνακα δύο διαστάσεων τύπου byte με όνομα "audioBt" αποθηκεύονται τα αποκρυπτογραφημένα δεδομένα του κάθε κομματιού. Τα παραπάνω δεδομένα λαμβάνονται από τις παραμέτρους που παίρνει η κλάση κατά την δημιουργία του στιγμιότυπού της.

```
    public AudioPlayer(byte[][] audioByte, String[] audioName, Display disp) {
        super();
        this.audioBt = audioByte;
        this.audioNm = audioName;
        this.display = disp;
        this.previous = display.getCurrent();
    }
```

Constructor της κλάσης. Αποθήκευση παραμέτρων στις τοπικές μεταβλητές της κλάσης.

```
public static List getList() {
        return theList;
   }
```

Επιστροφή της μεταβλητής theList, που περιέχει τα ονόματα των διαθέσιμων κομματιών.

```
   public void destroyApp(boolean unconditional) {
        if (playerGUI != null) {
           playerGUI.stopSound();
           playerGUI = null;
        }
        display.setCurrent(this.previous);
   }
```

Μέθοδος που καλείται κατά το πάτημα του κουμπιού “Exit”;

```
   public void commandAction(Command c, Displayable s) {
      if (c == exitCommand) {
         destroyApp(true);
     } else if ((s == theList && c == List.SELECT_COMMAND) ||
                     c == playCommand) {
           int i = theList.getSelectedIndex();
        if (playerGUI == null)
           playerGUI = new PlayerCanvas(display);

        else
           playerGUI.stopSound();
        int w = audioBt[i].length-1;
        do{
           if(audioBt[i][w] != 0)
              break;
           w--;
        }while(w>=0);
        byte[] temp = new byte[w+1];
        for(w=0; w<temp.length; w++){
           temp[w] = audioBt[i][w];
        }


        try{
           bais = new ByteArrayInputStream(temp);
        }catch(Exception ex){
            errorMessageAlert.setString(ex.toString());
```

```
            display.setCurrent(errorMessageAlert);
        }
        playerGUI.setParam(bais, (String)urls.elementAt(i));
        playerGUI.playSound();
        display.setCurrent(playerGUI);

    }
  }
```

Εντολές που εκτελούνται κατά το πάτημα του κουμπιού “playCommand”. Δημιουργείται ένα αντικείμενο της κλάσης PlayerCanvas το οποίο είναι στην ουσία ο Player που θα αναπαράγει τον ήχο. Η μέθοδος της που έχει όνομα setParam(), παίρνει σαν παράμετρο ένα αντικείμενο InputStream το οποίο είναι ένα ρεύμα δεδομένων και περιέχει το κομμάτι που θα αναπαραχθεί. Επίσης παίρνει σαν παράμετρο και το όνομα του κομματιού από την μεταβλητή urls. Τέλος, καλείται η μέθοδος playSound() η οποία αναπαράγει τον ήχο.

```
  private void initPlayList() {
        urls = new Vector();

        theList  = new List("Audio Player", Choice.IMPLICIT);

        for (int n = 0; n < audioNm.length; n++) {
        int e = -1;
        do{
           e++;
        }while(audioNm[n].charAt(e) != '.');
           urls.addElement(audioNm[n]);
        theList.append(audioNm[n].substring(0,e), null);
     }
        theList.addCommand(exitCommand);
        theList.addCommand(playCommand);
        theList.setCommandListener(this);
        display.setCurrent(theList);
  }
}
```

Η μέθοδος αυτή εμφανίζει στην οθόνη την λίστα με τους ήχους που παραλήφθηκαν στο κινητό. Από αυτήν την λίστα στην συνέχεια θα επιλεγεί το κομμάτι που θα αναπαραχθεί.

7.9 Κλάση PlayerCanvas

```
import javax.microedition.lcdui.*;
import javax.microedition.media.*;
import javax.microedition.media.control.*;
import java.io.InputStream;
```

Εισαγωγή των απαραίτητων βιβλιοθηκών.

```
public class PlayerCanvas extends Canvas implements Runnable, CommandListener
{
   private Player player;
   private Thread dThread;
   private InputStream is;
   private Image logo = null;

   Alert errorMessageAlert;
   private Display parentDisplay;
   private Command     backCommand      = new Command("Back",
Command.BACK,
                                 1);
   private Command     playCommand      = new Command("Play",
                                 Command.ITEM, 1);
   private Command     pauseCommand      = new Command("Pause",
                                 Command.ITEM, 10);
   private boolean disMTime, interrupted;
   private String title, url;
   private String mtime;
```

Διάφορες μεταβλητές.

```
   public PlayerCanvas(Display parentDisplay) {
      super();
         this.parentDisplay = parentDisplay;
         initialize();
   }
```

Constructor της κλάσης. Καλείται η μέθοδος initialize().

```
   private void initialize() {

         this.addCommand(backCommand);
         this.addCommand(pauseCommand);

         setCommandListener(this);
         try {
```

```
            logo = Image.createImage("/icons/logo.png");
        } catch (Exception ex) {
            logo = null;
        }
        if ( logo == null)
            System.out.println("can not load logo.png");

    }
```

Δημιουργεί το περιβάλλον του player.

```
    public void commandAction(Command c, Displayable s) {
        if (s == this) {
            if (c == backCommand) {
                stopSound();
                        removeCommand(playCommand);
                        addCommand(pauseCommand);
                parentDisplay.setCurrent(AudioPlayer.getList());
            } else if (c == playCommand) {
                playSound();
                        removeCommand(playCommand);
                        addCommand(pauseCommand);
            } else if (c == pauseCommand) {
                pauseSound();
                        removeCommand(pauseCommand);
                        addCommand(playCommand);
                }
        }
    }
```

Εντολές που εκτελούνται όταν πατηθεί κάποιο κουμπί κατά την διάρκεια της αναπαραγωγής.

```
    public void setParam(InputStream inStr, String url) {
        is = inStr;
            this.url = url;
            title = url;
    }
```

Καλείται από την MobileSecurity.

```
    public void playSound() {
            if (title == null || url == null)
                return;

            if (dThread == null) {
                dThread = new Thread(this);
```

```
            mtime = "";
            disMTime = true;
            interrupted = false;
            dThread.start();
        } else if (player != null) {
            try {
                player.start();
            } catch (Exception ex){
            errorMessageAlert.setString(ex.toString());
            parentDisplay.setCurrent(errorMessageAlert);
        }
    }
```

Η μέθοδος αυτή δημιουργεί ένα νέο νήμα που πραγματοποιεί την αναπαραγωγή του κομματιού.

```
    public void stopSound() {
        try {
            interrupted = true;
            disMTime = false;
            dThread = null;

            Thread.sleep(100);


            if (player != null) {
                player.close();
                player = null;
            }

        } catch (Exception ex) {
        errorMessageAlert.setString(ex.toString());
        parentDisplay.setCurrent(errorMessageAlert);
    }
}
```

Μέθοδος που καλείται κατά την έξοδο από τον player.

```
    void pauseSound() {
        try {
            if (player != null)
                player.stop();
        } catch (MediaException ex){
        errorMessageAlert.setString(ex.toString());
        parentDisplay.setCurrent(errorMessageAlert);
    }
}
```

Μέθοδος που καλείται κατά την προσωρινή παύση της αναπαραγωγής, δηλαδη κατά το πάτημα του κουμπιού “pause”.

```
   public void run() {
        if (player == null) {
           try {
                // method playSound() runs on GUI thread.
                // Manager.createPlayer() will potentially invoke a blocking
                // I/O. This is not the good practice recommonded by MIDP
                // programming style. So here we will create the
                // Player in a separate thread.
                createPlayer();
                player.realize();
                long dur = player.getDuration();
                if (dur != -1)
                   title = title + " [" + timeFM(dur)+ "]";
                player.start();
           } catch (Exception ex){
           errorMessageAlert.setString(ex.toString());
           parentDisplay.setCurrent(errorMessageAlert);
           }
        }
        while (!interrupted) {
           try {
                if (disMTime) {
                   mtime = timeFM(player.getMediaTime());
                   repaint(0,110, 100, 170);
                }
                Thread.sleep(100);
           } catch (Exception ex){
           errorMessageAlert.setString(ex.toString());
           parentDisplay.setCurrent(errorMessageAlert);
        }
    }
}
```

Η μέθοδος run() καλείται κατά την έναρξη νέου νήματος. Καλεί τις απαραίτητες μεθόδους ώστε να εμφανιστεί το περιβάλλον του player και να γίνει η αναπαραγωγή του επιλεγμένου κομματιού.

```
   void createPlayer() {
        try {
           String ctype;
        if (url.endsWith("wav")) {
           ctype = "audio/x-wav";
        } else {
           ctype = "audio/x-tone-seq";
        }
```

```
            player = Manager.createPlayer(is, ctype);
            player.setLoopCount(-1);
            } catch (Exception ex) {
               if (player != null) {
                     player.close();
                     player = null;
               }

               Alert alert = new Alert("Warning", "Cannot create player", null, null);
               alert.setTimeout(1000);

               parentDisplay.setCurrent(alert);
            }
      }
```

Η μέθοδος καλείται από την run() και δημιουργεί τον player παίρνοντας ως παράμετρο το ρεύμα δεδομένων που περιέχει το κομμάτι του ήχου.

```
   public void paint(Graphics g) {
         int w = getWidth();
         int h = getHeight();

         g.setColor(0);
         g.fillRect(0, 0, w, h);

         g.setColor(0xFF7f00);
         g.drawString("Audio Player", w/2, 8, Graphics.TOP | Graphics.HCENTER);

         if ( logo != null ) {
            g.drawImage(logo, w/2, 30, Graphics.TOP | Graphics.HCENTER);
         }
         g.setColor(0xFF7f00);
         g.drawString("Audio Player", w/2, 8, Graphics.TOP | Graphics.HCENTER);

         g.drawString(title, w/2, 84, Graphics.TOP | Graphics.HCENTER);

         g.drawString(mtime, 0, 150, Graphics.TOP | Graphics.LEFT);
   }
```

Μέθοδος που σχετίζεται με την εμφάνιση και την τοποθέτηση των διαφόρων στοιχείων στο περιβάλλον του player.

```
   protected void keyPressed(int keycode) {
         switch (keycode) {
         case KEY_STAR:
            changeVolume(-10);
            break;
         case KEY_POUND:
```

```
            changeVolume(10);
            break;
        }
    }
```

Κατά την διάρκεια της αναπαραγωγής παρέχεται η δυνατότητα αυξομείωσης της έντασης του ήχου με το πάτημα των κουμπιών δίεση (#) και αστερίσκος (*).

```
    private void changeVolume(int diff) {
        VolumeControl vc;

        if ( player != null) {
            vc = (VolumeControl) player.getControl("VolumeControl");
            if (vc != null) {
                int cv = vc.getLevel();
                cv += diff;
                cv = vc.setLevel(cv);
            }
        }
    }
```

Μέθοδος που σχετίζεται με την αυξομείωση της έντασης του ήχου.

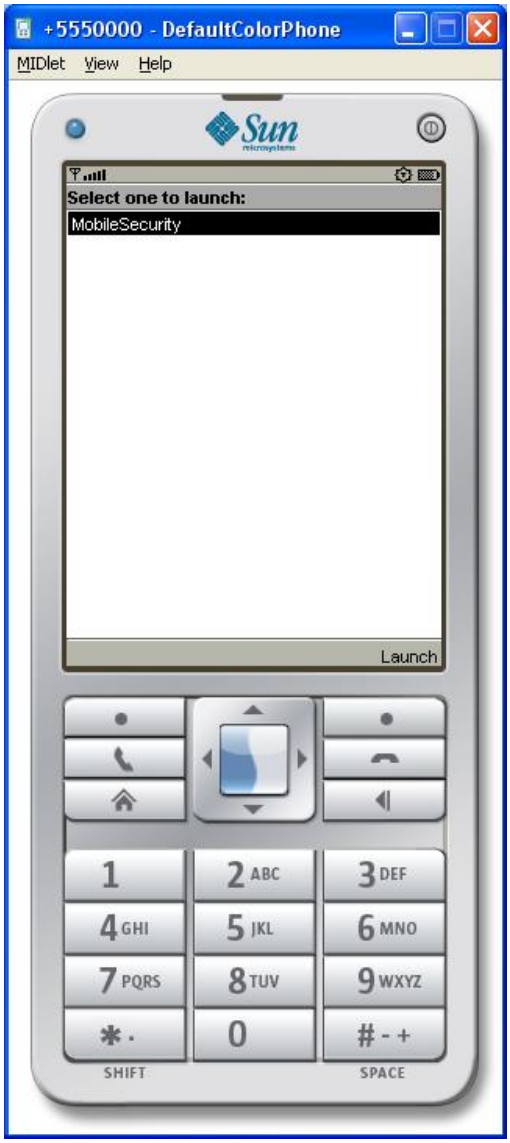
```
    private String timeFM(long val) {
        String ret = "";
        int mval = (int)(val/1000);
        int sec = mval/1000;
        int min = sec/60;
        if (min >= 10)
            ret = ret + min + ":";
        else if (min > 0)
            ret = "0" + min + ":";
        else
            ret = "00:";
        if (sec >= 10)
            ret = ret + sec + ".";
        else if (sec > 0)
            ret = ret + "0" + sec + ".";
        else
            ret = ret + "00.";
        mval = (mval % 1000) / 100;
        ret = ret + mval;
        return (ret);
    }
}
```

Υπολογισμός του χρόνου που διαρκεί το κομμάτι

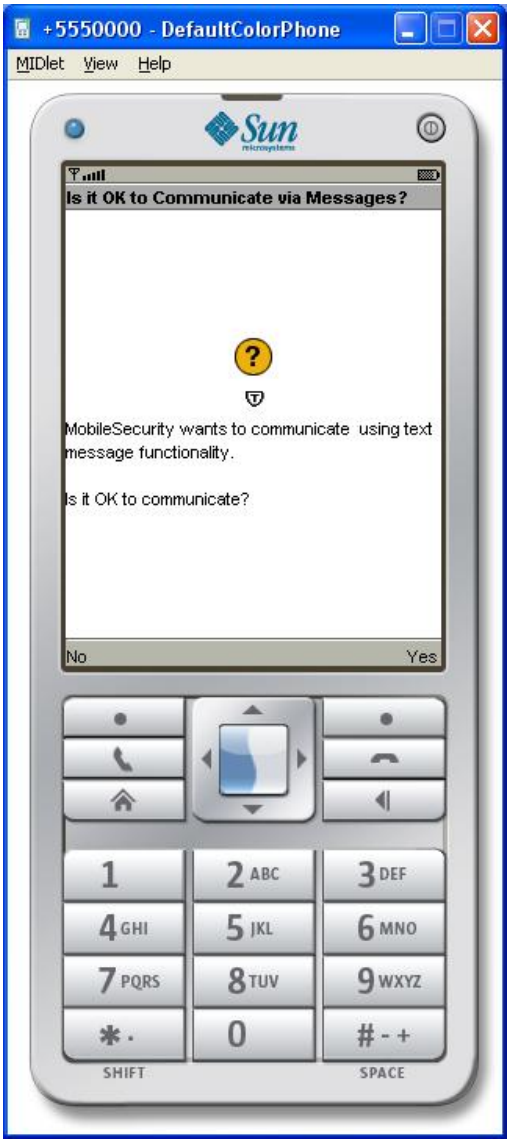
7.10 Screenshots

Ακολουθούν τα στιγμιότυπα της εκτέλεσης του κώδικα.

7.10.1 Αποστολή και λήψη SMS



Εικόνα 7.9



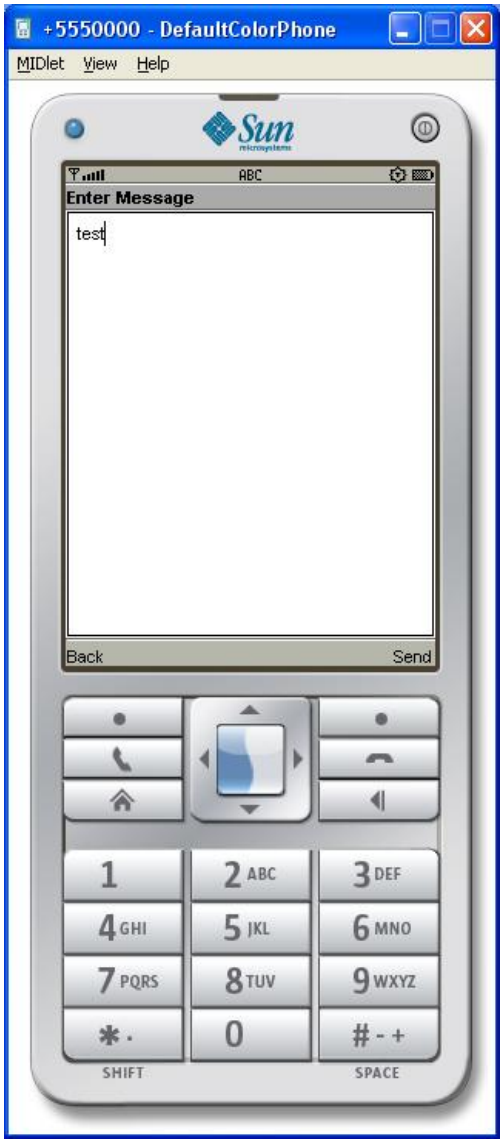
Εικόνα 7.10

Κατά την έναρξη του προσομοιωτή εμφανίζεται η εικόνα 7.9. Παρουσιάζεται μόνο ένα MIDLET και όχι πέντε όπως ήταν αρχικά. Στην συνέχεια πατώντας το κουμπί “Yes” εμφανίζεται η εικόνα 7.11, που βρίσκεται παρακάτω.

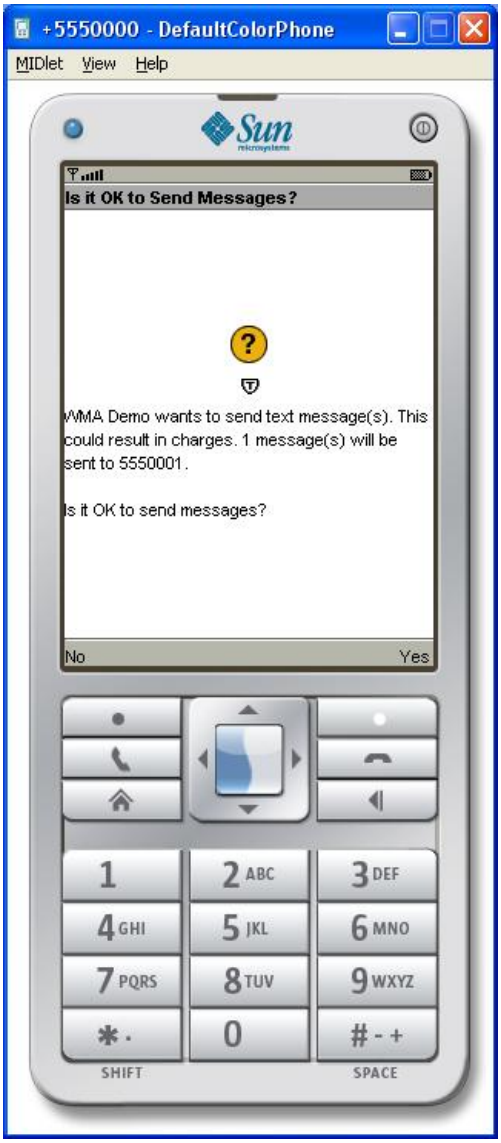
Εικόνα 7.11

Εικόνα 7.12

Το MIDLET ξεκινάει και εμφανίζονται δύο επιλογές, μία για την αποστολή SMS και μία για την αποστολή MMS. Ταυτόχρονα εγκαθίσταται και ένας MessageListener ο οποίος περιμένει για εισερχόμενα μηνύματα SMS. Με την επιλογή του SMS Send εμφανίζεται η εικόνα 7.12, ενώ με την δεύτερη επιλογή εμφανίζεται η 7.17. Από ‘κει και πέρα εμφανίζονται σχεδόν οι ίδιες οθόνες με αυτές του προηγούμενου project, εκτός από μερικές που σχετίζονται με την προσθήκη ήχου στα MMS και οι οποίες είτε έχουν αλλάξει λίγο, είτε έχουν προστεθεί επιπλέον καινούργιες.



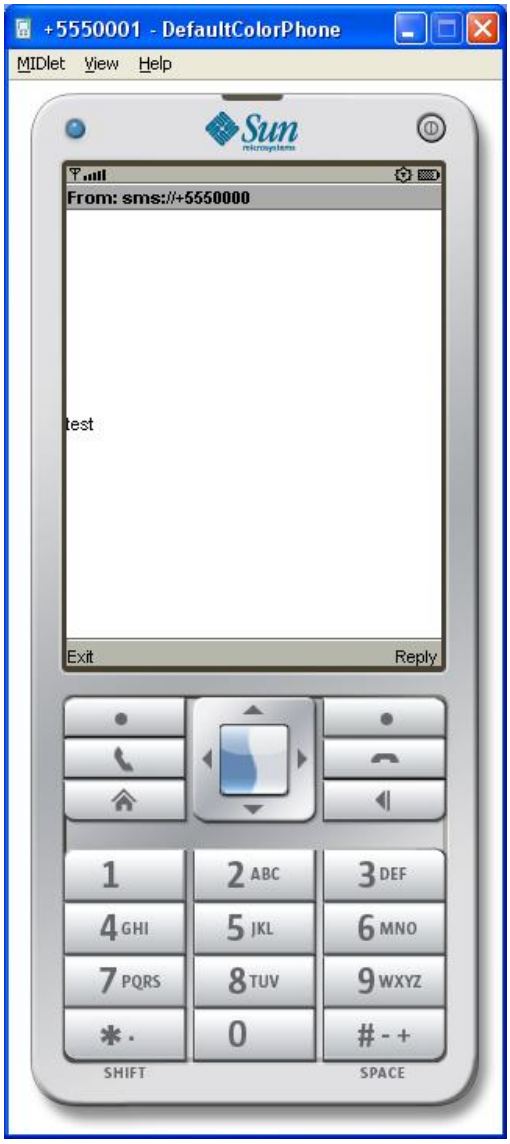
Εικόνα 7.13



Εικόνα 7.14

Εισαγωγή του κειμένου που θα σταλεί και πάτημα του κουμπιού Send. Στην ερώτηση που ακολουθεί, πάτημα του "Yes".

Εικόνα 7.15

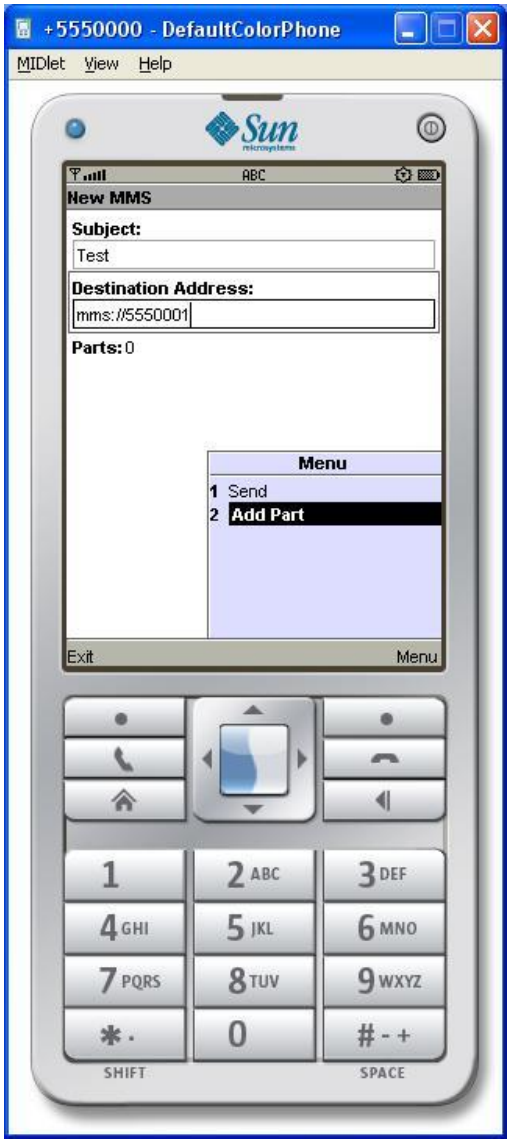


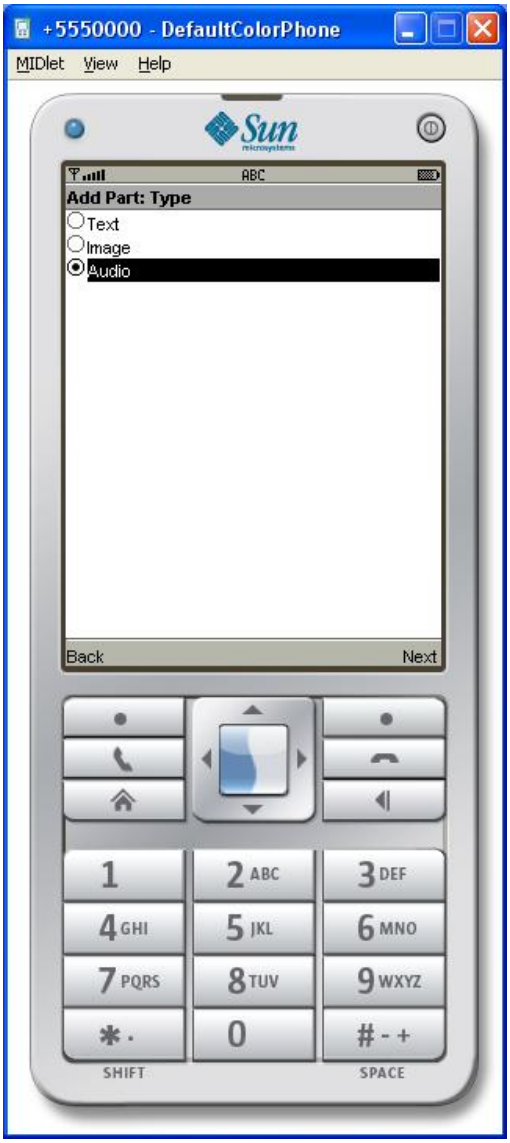
Εικόνα 7.16

Αποστολή και στην συνέχεια παραλαβή και εμφάνιση του μηνύματος.

7.10.2 Αποστολή και λήψη MMS

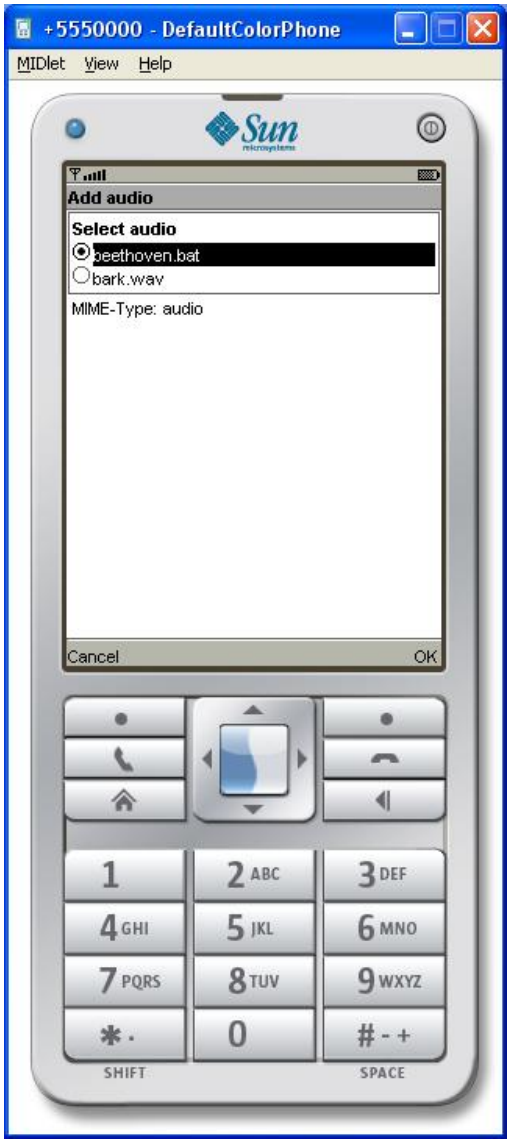
Παρακάτω ακολουθούν τα screenshots που δείχνουν τη διαδικασία αποστολής και λήψης MMS.





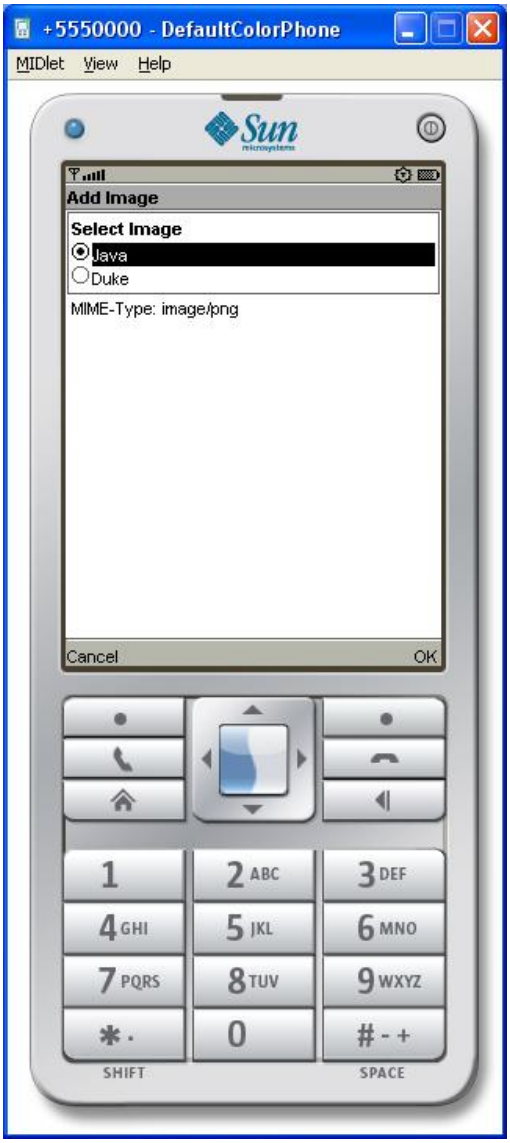
Εικονα.7.17 ***Εικόνα 7.18***

Στο πεδίο subject γράφεται το subject του MMS και στο πεδίο Destination Address ο αριθμός του παραλήπτη. Με το πάτημα του δεξιού πλήκτρου με τίτλο "Menu", εμφανίζονται δύο επιλογές. Η πρώτη επιλογή επιτρέπει την αποστολή του μηνύματος, ενώ με την δεύτερη γίνεται η προσθήκη μερών(κείμενο, εικόνα και ήχος) στο μήνυμα.



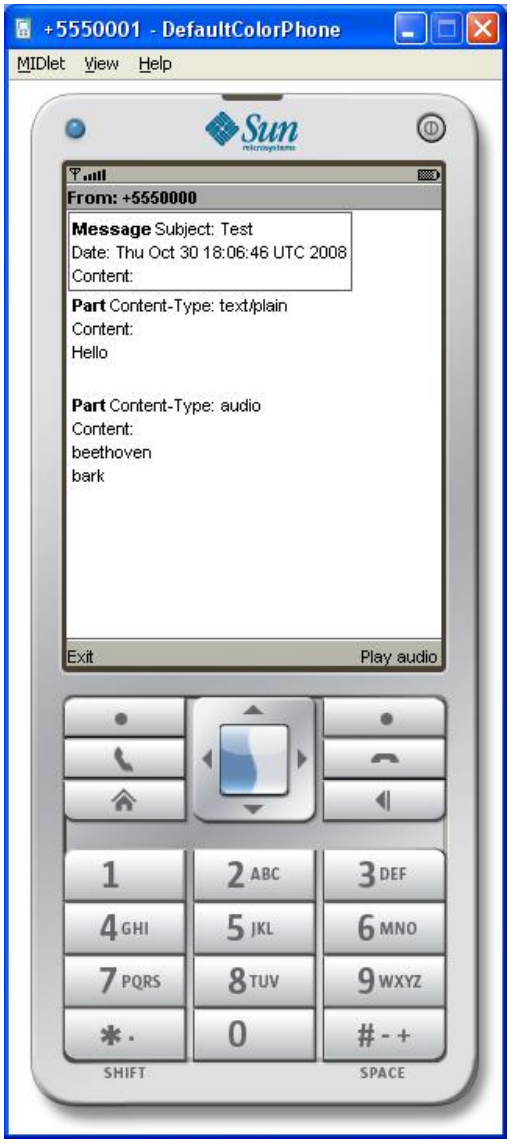
Εικόνα 7.19 ***Εικόνα 7.20***

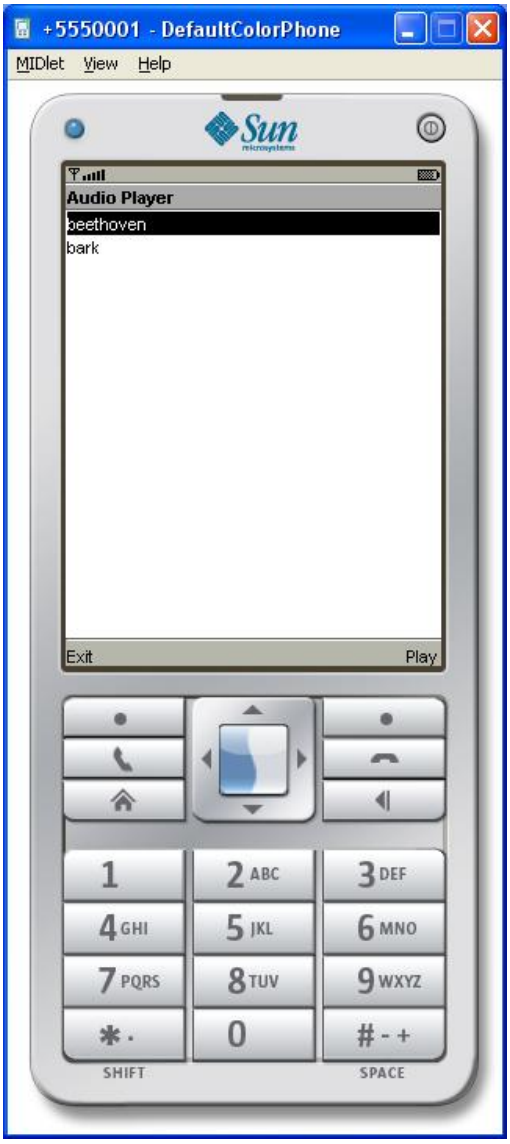
Η εικόνα 7.19 εμφανίζεται όταν επιλέξουμε το “Audio” στην 7.18 και επιτρέπει την εισαγωγή ενός κομματιού ήχου στο μήνυμα.
Η 7.20 παρουσιάζεται με την επιλογή “Text” στην 7.18 και επιτρέπει την εισαγωγή ενός τμήματος κειμένου.



Εικόνα 7.21

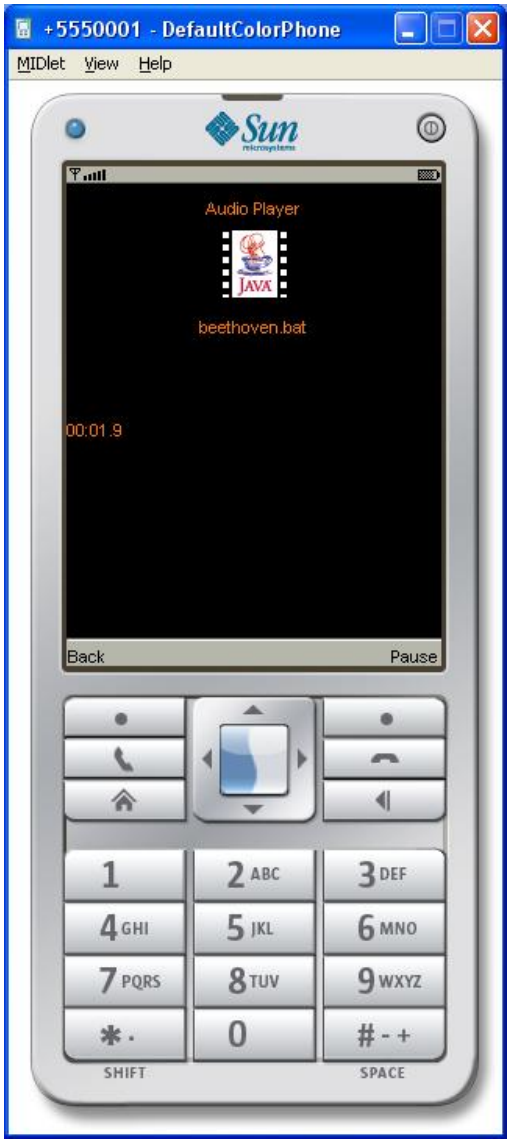
Εμφανίζεται με την δεύτερη επιλογή της εικόνας 7.18 και δίνει την δυνατότητα εισαγωγής μιας εικόνας στο MMS.



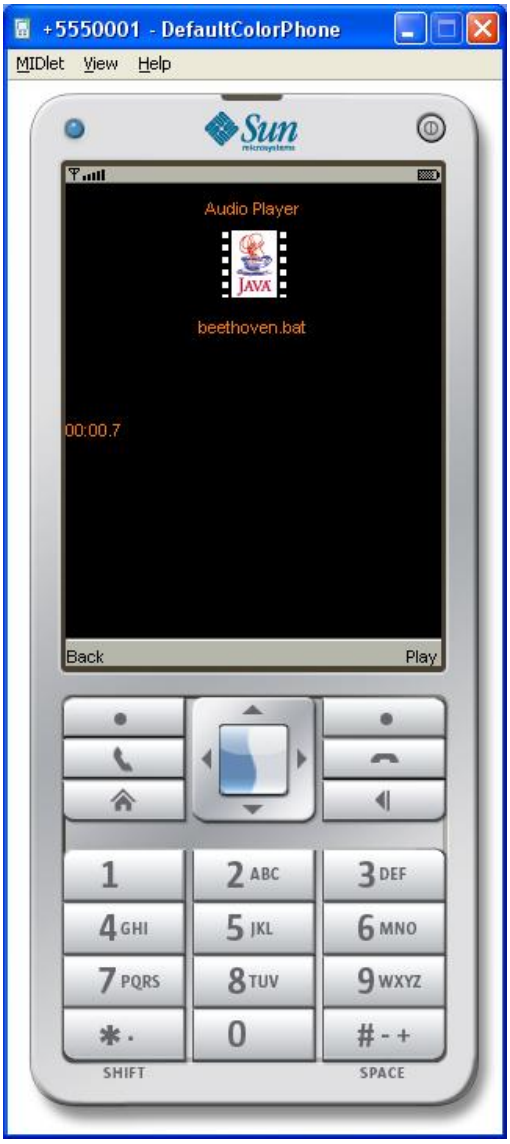


Εικόνα 7.22 ***Εικόνα 7.23***

Η εικόνα 7.22 δείχνει το μήνυμα MMS όπως εμφανίζεται στον παραλήπτη. Τα αρχεία ήχου εμφανίζονται τελευταία και ομαδοποιημένα ανεξάρτητα από την σειρά που τα έβαλε ο αποστολέας. Με το πάτημα του "Play audio" παρουσιάζεται η δεξιά οθόνη όπου εμφανίζονται όλα τα αρχεία ήχου. Με το πάτημα του κουμπιού "Play" εμφανίζεται η παρακάτω εικόνα και αναπαράγεται αυτόματα το επιλεγμένο κομμάτι.



Εικόνα 7.24



Εικόνα 7.25

Player αναπαραγωγής ήχου. Πατώντας το κουμπί “Pause”, γίνεται παύση της αναπαραγωγής και εμφανίζεται στην θέση του το “Play”. Πατώντας το “Play” συνεχίζεται η αναπαραγωγή από το σημείο που σταμάτησε. Επίσης δίνεται η δυνατότητα αυξομείωσης του ήχου χρησιμοποιώντας τα κουμπιά δίεση (#) και αστερίσκος (*).

7.11 Συμπεράσματα

Στο project αυτό, η κρυπτογράφηση των μηνυμάτων γίνεται σε επίπεδο εφαρμογής. Αυτό σημαίνει ότι τα δεδομένα μεταδίδονται από τον αποστολέα μέχρι τον παραλήπτη χωρίς να αποκρυπτογραφηθούν σε κανένα σημείο της διαδρομής του δικτύου. Αυτό έχει ως αποτέλεσμα την ασφάλεια των δεδομένων σε όλη την μετάδοση. Η κρυπτογράφηση αυτή προσθέτει απλά ένα ακόμα επίπεδο ασφαλείας στην ασύρματη επικοινωνία. Στην ουσία ένας επιτιθέμενος θα πρέπει πρώτα να πραγματοποιήσει μία επιτυχημένη επίθεση απ' αυτές που παρουσιάζονται στην αρχή και στην συνέχεια να κάνει άλλη μία επιπλέον επίθεση ώστε να αποκρυπτογραφήσει τα δεδομένα σε επίπεδο εφαρμογής.

Το μόνιμο κλειδί βρίσκεται μέσα στον κώδικα και άρα υπάρχει μέσα στην συσκευή σε μορφή εκτελέσιμου κώδικα γεγονός που κάνει την ανάκτηση του από πολύ δύσκολη έως αδύνατη. Επίσης ο τρόπος που γίνεται η κωδικοποίηση των δεδομένων, η δημιουργία ενός νέου κλειδιού συνόδου και η μετάδοση του μαζί με τα δεδομένα του μηνύματος, βρίσκεται επίσης μέσα στον κώδικα και μόνο εκεί, ο οποίος υποτίθεται ότι παραμένει μυστικός από τις κατασκευάστριες εταιρείες των κινητών συσκευών. Έτσι ένας επιτιθέμενος, αν και πιθανόν να ξέρει τον αλγόριθμο κρυπτογράφησης, δεν θα μπορεί να διαβάσει το μήνυμα που αποστέλλεται γιατί θα πρέπει να βρει το κλειδί συνόδου πιθανόν μέσω εξαντλητικής αναζήτησης κλειδιών και αν ληφθεί υπ' όψιν το μήκος του κλειδιού, αυτή η τεχνική αποτυγχάνει. Ένας επιτιθέμενος ψάχνοντας 1 δις κλειδιά το δευτερόλεπτο, θα χρειαζόταν πάνω από 10 εξάκις εκατομμύρια χρόνια για να ψάξει όλο τον χώρο των κλειδιών. Δεν αξίζει λοιπόν καν η προσπάθεια γιατί το κλειδί θα έχει αξία μόνο για ένα μήνυμα.

Αν τώρα κάποιος επιχειρήσει να μαντέψει το μόνιμο κλειδί πάλι μέσω εξαντλητικής αναζήτησης κλειδιών και ας υποτεθεί ότι το βρίσκει, πράγμα εξαιρετικά απίθανο, δεν θα τον ωφελήσει και τόσο γιατί το μόνο που θα ανακτήσει από ένα μήνυμα είναι ο κωδικοποιημένος σπόρος που χρησιμοποιείται για να παραχθεί το προσωρινό κλειδί. Ο επιτιθέμενος όμως δεν γνωρίζει με ποιον τρόπο παράγεται το κλειδί συνόδου από τον σπόρο και έτσι θα πρέπει να δοκιμάσει διάφορες προγραμματιστικές τεχνικές για να τον μαντέψει. Λόγω όμως της μερικής πολυπλοκότητας που υπάρχει στην παραγωγή του προσωρινού κλειδιού, κάνει τον επιτιθέμενο ανίσχυρο μπροστά στην ανάκτηση του.

Το σύστημα αυτό όμως μπορεί να αποτύχει με έναν σχετικά εύκολο τρόπο. Αν ένας επιτιθέμενος κάνει μία επίθεση εγγραφής ψεύτικου σταθμού βάσης στο επιτιθέμενο κινητό αλλά και στο δικό του νόμιμο κινητό, τότε το ψεύτικο BTS μπορεί να λειτουργήσει ως διάμεσο για τα δύο κινητά και να δρομολογήσει τα κρυπτογραφημένα δεδομένα από την επιτιθέμενη συσκευή προς στην δική του ώστε η τελευταία να κάνει την δύσκολη δουλειά της αποκρυπτογράφησης. Δηλαδή εκμεταλλεύεται την νόμιμη λειτουργία του δικτύου GSM. Αυτή η τεχνική θα μπορούσε να χρησιμοποιηθεί μόνο για να ανακτήσει τα δεδομένα που κρυφακούει εκείνη την ώρα και όχι για να ανακτήσει το προσωρινό ή το μόνιμο κλειδί. Αυτή η αποτυχία οφείλεται μόνο και μόνο στις αδυναμίες του GSM και ο επιτιθέμενος απλά εκμεταλλεύεται την νόμιμη χρήση του κινητού για να διαβάσει τα κρυπτογραφημένα δεδομένα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Κατανεμημένα συστήματα με java – Συστήματα υπολογιστών Τόμος ΙΙΙ
   Ι. Κ. Κάβουρας, Ι. Ζ. Μήλης, Γ. Β. Ξυλωμένος, Α. Α. Ρουκουνάκη
2. Εισαγωγή στην java 2
   Γιώργος Λιακέας
3. Δίκτυα υπολογιστών – Τέταρτη αμερικάνικη έκδοση
   Andrew S. Tanenbaum
4. Sms / Mms Security - Secure Communication Protocol and Threats' description
   Papakonstantinou Artemis - Department of Computer Science University of Crete

ΔΙΑΔΙΚΤΥΟ

1. http://java.sun.com/javase/downloads/index.jsp
2. http://www.netbeans.info/downloads/all.php?b_id=2323
3. http://www.netbeans.info/downloads/start.php?f_id=13708&lang_id=1
4. http://java.sun.com/products/sjwtoolkit/download.html
5. http://www.viruslist.com/en/viruslist.html?id=4094
6. http://www.viruslist.com/en/viruses/encyclopedia?virusid=60663
7. http://antivirus.about.com/od/wirelessthreats/a/brador.htm
8. http://www.servitoros.gr/misc_news/view.php/669/
9. http://www.myphone.gr/forum/showthread.php?t=32856
10. http://www.myphone.gr/forum/showthread.php?t=89234
11. http://www.probertencyclopaedia.com/cgi-bin/res.pl?keyword=Dust+Virus&offset=0
12. http://java.sun.com/products/midp/overview.html
13. http://java.sun.com/products/midp/
14. http://java.sun.com/products/midp/whatsnew.html
15. http://java.sun.com/products/cldc/index.jsp
16. http://java.sun.com/products/cldc/overview.html
17. http://www.myphone.gr/library/article-37.html
18. http://myhowto.org/java/23-j2memidp-programming-for-the-cell-phones-good-bad-and-ugly/
19. http://www.sematopia.com/?cat=27
20. http://www-128.ibm.com/developerworks/wireless/library/wi-prep/
21. http://www.cenriqueortiz.com/blog/cldc-based-profiles/comments-on-cldc-profiles-today.html eikones mono
22. http://developers.sun.com/mobility/midp/articles/pushreg/
23. http://www.funsms.net/sms_history.htm
24. http://www.developershome.com/sms/smsIntro.asp
25. http://www.gsmforum.gr/forum/showthread.php?t=14166
26. http://www.aera.gr/gr/content/view/1896/10255/
27. http://www.myroute.gr/sms.html
28. http://el.wikipedia.org/wiki/SMS
29. http://www.newmedianet.gr/sms_marketing.php
30. http://www.loveforbiz.com/interesting-mobile-and-sms-statistics

ΠΑΡΑΡΤΗΜΑ

Παρουσίαση Powerpoint

Θέμα

Μελέτη των απειλών και υλοποίηση μηχανισμών για την ασφάλεια των υπηρεσιών SMS & MMS

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Φαϊτάκη Κωνσταντίνα - ΑΜ 1237
Φεσάκης Ηλίας - ΑΜ 1342

Υπεύθυνος καθηγητής: Μανιφάβας Χάρης
Ηράκλειο 3-2-2009

Slide 1

Περιεχόμενα

- SMS – MMS
- Δίκτυο GSM
- Μετάδοση SMS
- Άλλοι τύποι μηνυμάτων – EMS MMS
- Επιθέσεις SMS
- MIDP – CLDC
- Υλοποίηση και λειτουργία του κώδικα
- Δημιουργία κλειδιών
- Αποστολή SMS και MMS
- Πακέτα και κλάσεις
- Συμπεράσματα

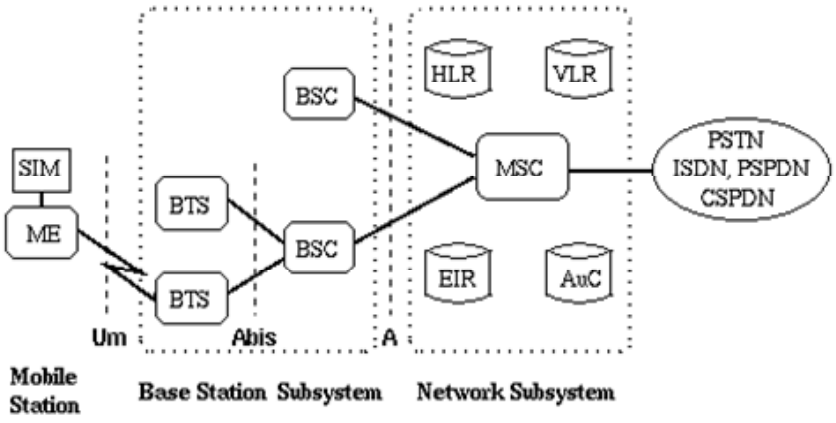
Slide 2

SMS - MMS

- Τι είναι SMS
- Τι είναι MMS
- Ιστορική αναδρομή
- Τεχνικά χαρακτηριστικά (κωδικοποίηση, μήκος)

Slide 3

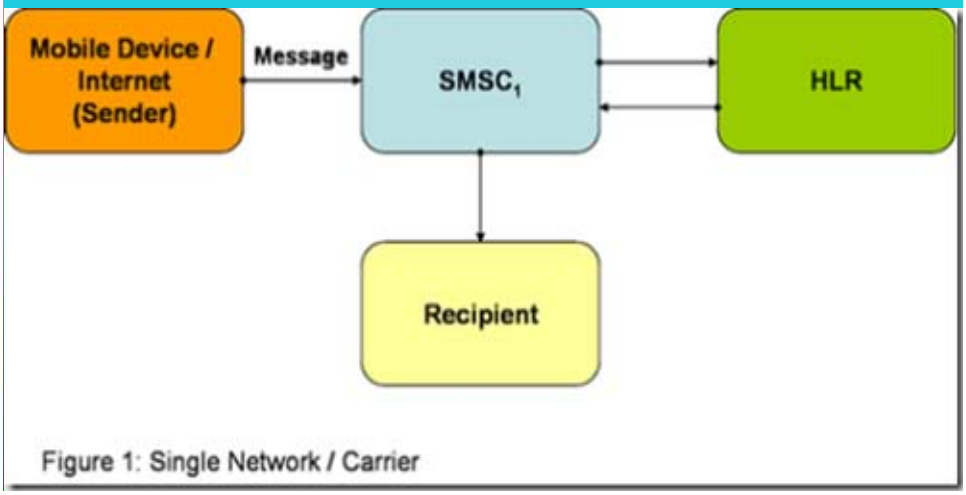
Δίκτυο GSM



SIM Subscriber Identity Module
ME Mobile Equipment
BTS Base Transceiver Station
BSC Base Station Controller
HLR Home Location Register
VLR Visitor Location Register
MSC Mobile services Switching Center
EIR Equipment Identity Register
AuC Authentication Center

Slide 4

Μετάδοση SMS



Slide 5

Άλλοι τύποι μηνυμάτων

- Υπηρεσία ενισχυμένου μηνύματος – EMS
  - Τι περιέχει
  - Μειονεκτήματα
- Υπηρεσία μηνυμάτων με πολυμέσα – MMS
  - Ιστορικά
  - Τι περιέχει (εικόνα, ήχος, βίντεο)
  - Στοιχεία δικτύου MMS
    - MMS Server
    - Proxy Delay MMS
    - MMS User Database
    - MMS User Agent

Slide 6

Επιθέσεις SMS

- Denial of Service Attack (DOS)
  - Συντονισμένη επίθεση
  - Κατάρρευση λόγω μεγάλου φορτίου
  - Στόχος το άδειασμα της μπαταρίας και η κατάρρευση του SMSC
  - Χρήση σιωπηλών μηνυμάτων
- Service Interruption Attack
  - Εκμεταλλεύεται τρωτά σημεία στο σύστημα
  - Χρήση ειδικά διαμορφωμένων μηνυμάτων
  - Στόχος η διακοπή υπηρεσιών
- Service Hijacking Attack
  - Μη εξουσιοδοτημένος έλεγχος του συστήματος
  - Εκμετάλλευση και τροποποίηση δεδομένων

Slide 7

Επιθέσεις SMS (συν)

- Buffer Overflow Attack
  - Ανεξέλεγκτα όρια στη μνήμη
  - Υπερβολικά μεγάλες μεταβλητές
  - Επικάλυψη δεδομένων άλλων προγραμμάτων
- Password Compromise Attack
  - Ανάκτηση κωδικού πρόσβασης του SMSC
  - Μέθοδος δοκιμής και λάθους
  - Επίθεση μέσω ελεγχόμενης μηχανής Διαδικτύου
- Snooping
  - Μη εξουσιοδοτημένη ανάκτηση δεδομένων
  - Γίνεται μετά από επίθεση πειρατείας
  - Κατασκόπευση δικτύου
  - Υποκρισία

Slide 8

Επιθέσεις SMS (συν)

- Spoofing
  - Αλλαγή ταυτότητας του αποστολέα του μηνύματος
- SMS phishing
  - SMS που ξεγελούν τον χρήστη
  - Εγκατάσταση Δούρειου Ίππου
  - Στόχος ο έλεγχος της συσκευής
- Radio Frequency Jamming
  - Παράγωγη θορύβου
  - Στόχος η διακοπή της υπηρεσίας
- SMS Spam
  - Ενοχλητικά μηνύματα
  - Προτροπή του χρήστη ώστε να καλέσει αριθμό υψηλής χρέωσης

Slide 9

Επιθέσεις SMS (συν)

- Ιοί κινητών
  - Palm.Phage.A
    - Malware
    - Στόχος τα PalmOS
  - Cabir
    - Μετάδοση μέσω Bluetooth
    - Απόδειξη ότι και τα κινητά μολύνονται
  - Dust
    - Μολύνει συσκευές με Windows CE
    - Δεν αποτελεί σοβαρή απειλή
  - Brador Trojan
    - Στόχος οι συσκευές με Windows CE
    - Πλήρης απομακρυσμένος έλεγχος

Slide 10

MIDP - CLDC

- Πρότυπα ανάπτυξης διασύνδεσης και εκτέλεσης κώδικα
- Γλώσσα προγραμματισμού Java
- Εκτέλεση προγραμμάτων σε μικροσυσκευές με περιορισμένους πόρους
- CLDC - Περιέχει κλάσεις πυρήνα και την KVM
- MIDP - Περιέχει κλάσεις που επεκτείνουν το CLDC
- Απαραίτητα για την εκτέλεση εφαρμογών στις μικροσυσκευές

Slide 11

Υλοποίηση κώδικα

- Γλώσσα προγραμματισμού Java
- Προγράμματα που χρησιμοποιήθηκαν
  - Jdk 6 – Εκτέλεση Java στον Η/Υ
  - Netbeans 5.5 – Πλατφόρμα ανάπτυξης εφαρμογών
  - Netbeans 5.5 mobility pack – Υποστήριξη J2ME
  - Wireless toolkit 2.5.2 – Προσομοιωτής κινητού
- Τι κάνει ο κώδικας
- Από πού προήλθε
  - Τροποποίηση αρχικού κώδικα

Slide 12

Λειτουργία του κώδικα

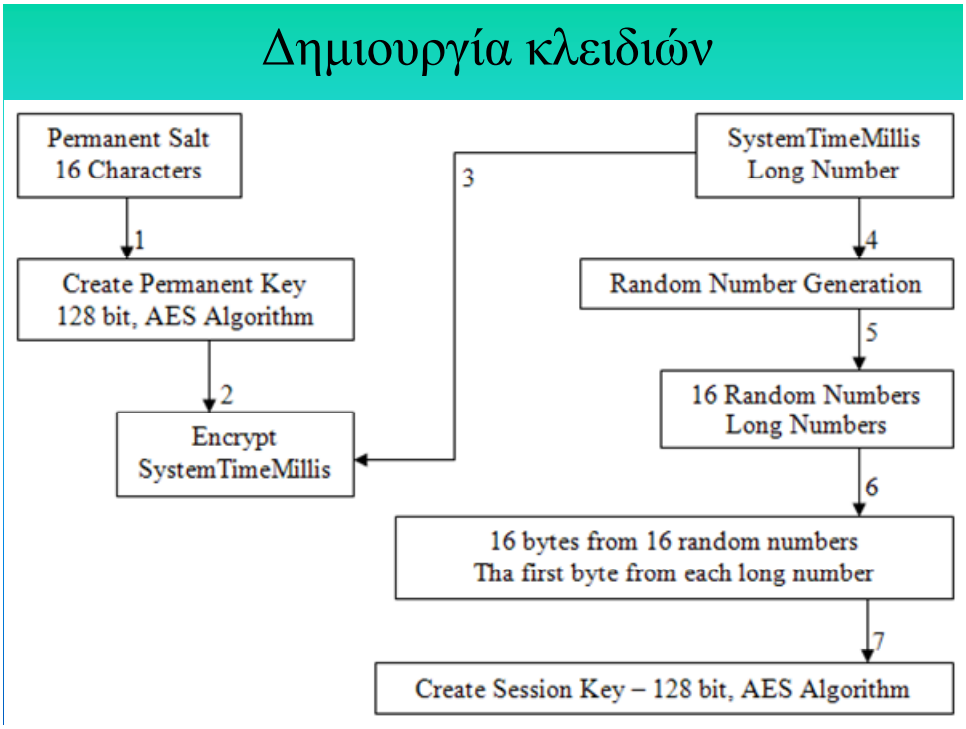
- Αποστολή και λήψη μηνυμάτων sms και mms
  - Sms – Κείμενο
  - Mms – Θέμα, κείμενο, εικόνα και ήχος
  - Πρόβλημα με θύρες – λύση
- Χρήση μόνιμου και προσωρινού κλειδιού
  - Μήκος 128 bit
  - Αλγόριθμος AES
- Παραγωγή προσωρινού κλειδιού
  - Χρήση αρχικού τυχαίου αριθμού (χρόνος συστήματος)
  - Χρήση γεννήτριας τυχαίων αριθμών
  - Κρυπτογράφηση και αποστολή τυχαίου αριθμού

Slide 13

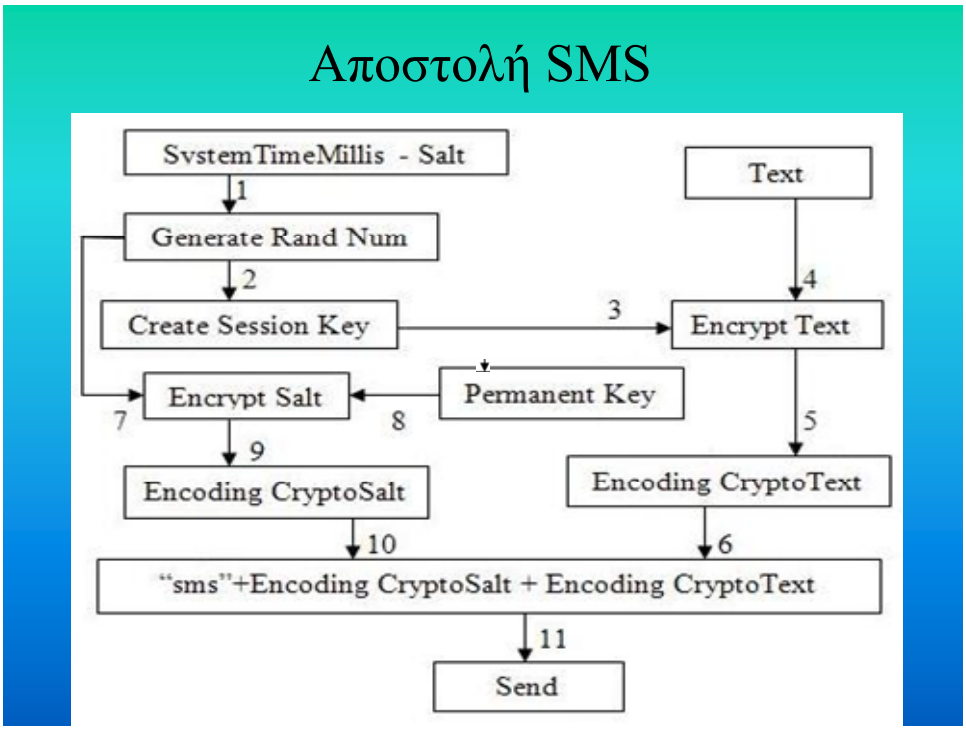
Λειτουργία του κώδικα (συν)

- Κρυπτογράφηση κειμένου και δεδομένων byte
- Κωδικοποίηση κρυπτογραφημένου κειμένου
- Αυτοματοποίηση λήψης
- Διαφορές με τον αρχικό κώδικα

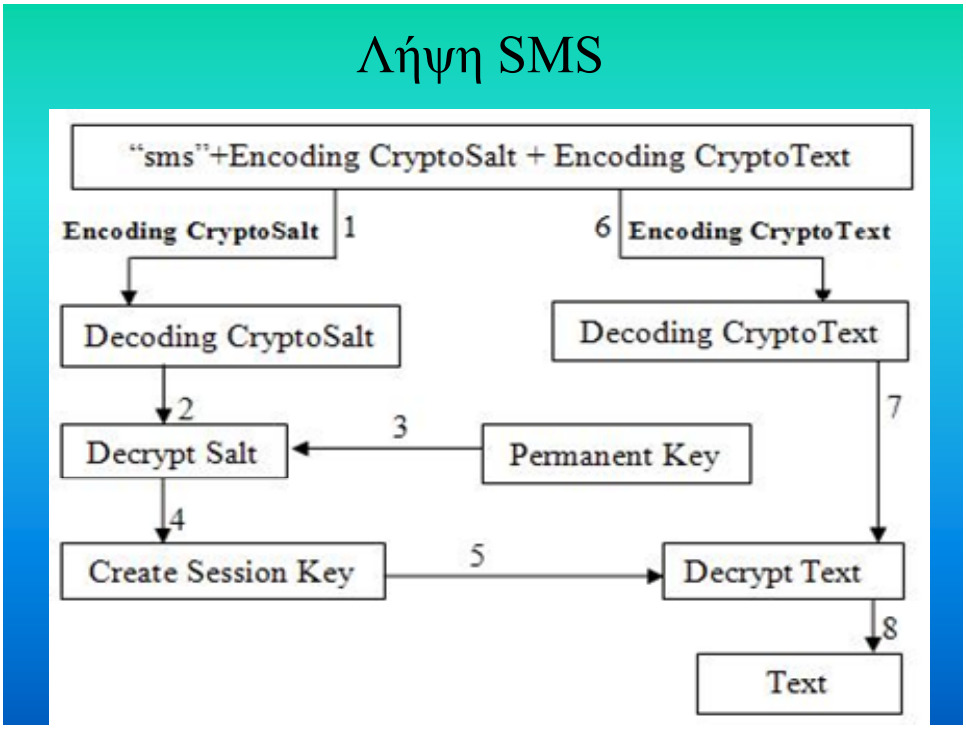
Slide 14



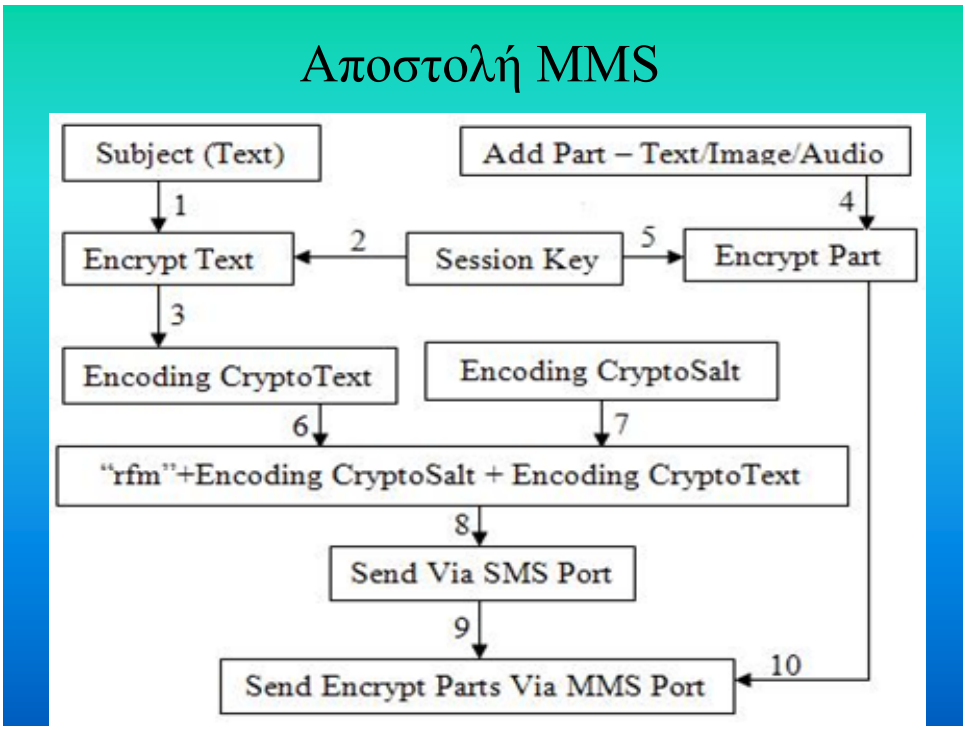
Slide 15



Slide 16



Slide 17



Slide 18

Πακέτα και κλάσεις

- MobileSecurity
  - MobileSecurity
  - Cryptografy
  - AudioPlayer
  - PlayerCanvas
- sms
  - SMSSender
- audio
  - Bark.wav
  - beethoven

Slide 19

Πακέτα και κλάσεις (συν)

- icons
  - App.png
  - Duke.png
  - Java.png
  - Logo.png
- mms
  - MMSMessage
  - PartsDialog
  - SendHeader
  - SenderThread

Slide 20

Συμπεράσματα

- Κρυπτογράφηση σε επίπεδο εφαρμογής – απ' άκρο εις άκρον
- Τα δεδομένα είναι ασφαλή σε όλη την διαδρομή τους
- Προσθήκη ενός ακόμα επιπέδου ασφαλείας
- Αρκετά ασφαλές λόγω του μήκους των κλειδιών
- Πολυπλοκότητα στην παραγωγή του προσωρινού κλειδιού – αυξάνει την ασφάλεια

Slide 21

ΤΕΛΟΣ

Slide 22