Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων

Πτυχιακή εργασία

Τίτλος:

Υλοποίηση ασφαλούς κατανεμημένης εφαρμογής που προσομοιώνει ηλεκτρονικές συναλλαγές

Αλέξανδρος Κονιδάρης (ΑΜ: 871)

Ηράκλειο – 08/10/2007

Επόπτης Καθηγητής: Δρ. Μανιφάβας Χαράλαμπος

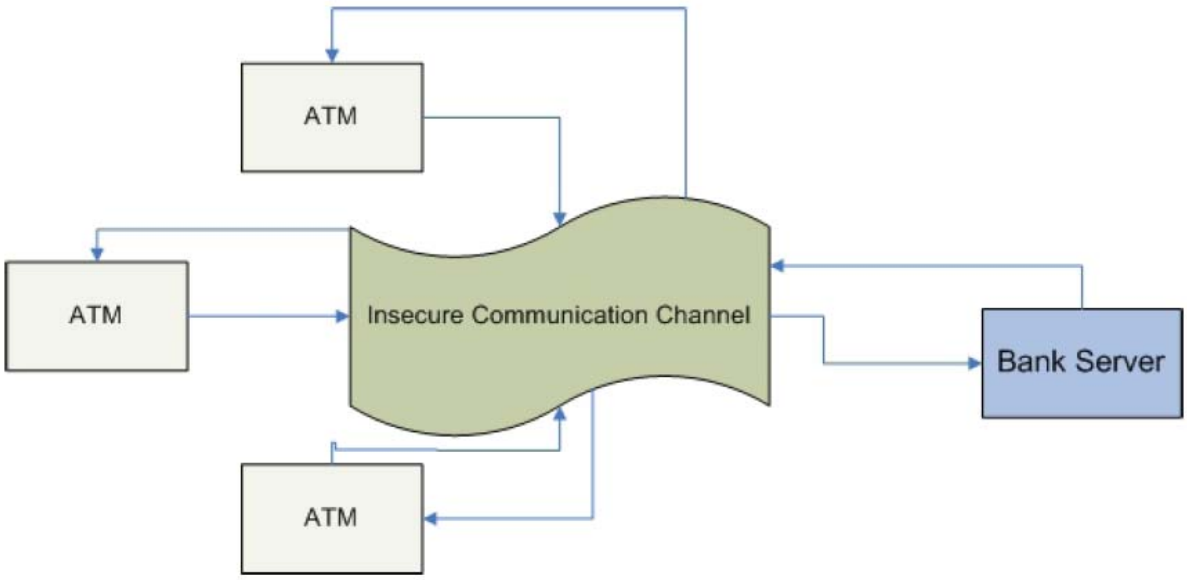
Περιεχόμενα

Πρόλογος

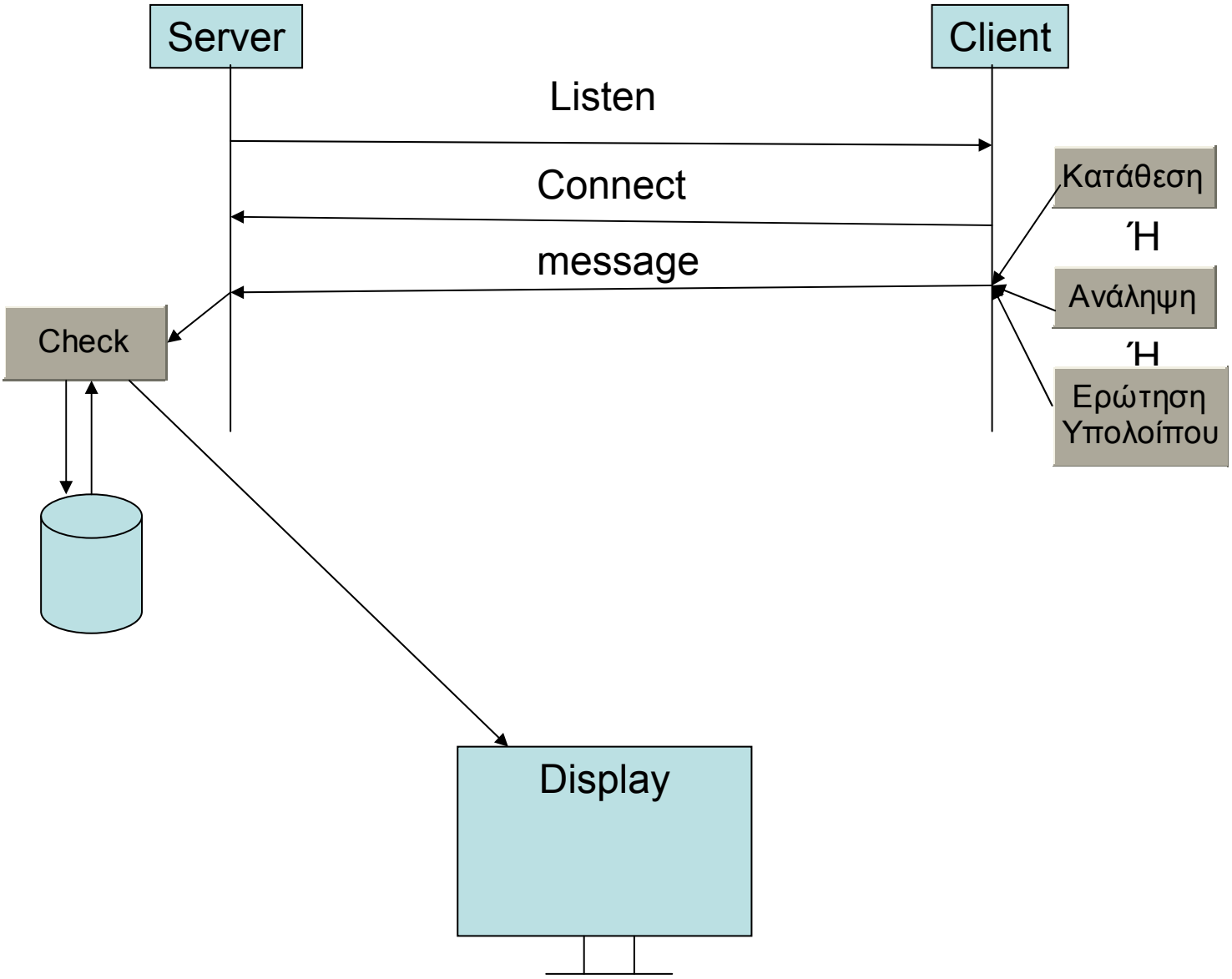
Η εργασία αυτή θα εξετάσει την τρέχουσα τεχνολογία συναλλαγής μεταξύ ενός κεντρικού υπολογιστή (Server) μιας τράπεζας και ενός ATM (Client). Στην πράξη θα υλοποιηθεί ένα τέτοιο σύστημα δίνοντας έμφαση στα θέματα ασφάλειας. Η ασφάλεια αφορά την πιστοποίηση της ταυτότητας του συναλλασσόμενου, την ασφαλή μεταφορά δεδομένων καθώς και την ασφαλή καταχώρηση στοιχείων τα οποία θα ήταν δυνατό να χρησιμοποιηθούν σε μελλοντικές διενέξεις.

Οι αυτόματες ταμειακές μηχανές (ATM) αποτελούν στοιχεία ενός κατανεμημένου συστήματος που σκοπός του είναι να παρέχει ένα σύνολο τραπεζικών υπηρεσιών στους πελάτες μίας τράπεζας, για παράδειγμα ερωτήσεις υπόλοιπου λογαριασμού, καταθέσεις, αναλήψεις μετρητών κλπ.

Ο κεντρικός υπολογιστής της τράπεζας (Bank Server) ανήκει σε ένα κατανεμημένο σύστημα στο οποίο πολλαπλά ATM μπορούν να συνδεθούν μαζί του με σκοπό την εκτέλεση εντολών που επιθυμεί ο εκάστοτε πελάτης.



Σχεδιάγραμμα



Listen = Ο Server «ακούει» για τυχόν αίτημα σύνδεσης
Connect = Ο Client κάνει αίτημα σύνδεσης στον Server. Μόλις γίνει η σύνδεση ο Server στέλνει το Public Key του στον Client.
message = Το μήνυμα αυτό περιλαμβάνει το PIN, το Account Number, το ποσό και το είδος της συναλλαγής. Όλα αυτά γίνονται ένα αλφαριθμητικό (message) και κωδικοποιούνται με το Public Key του Server. Ο Server μόλις το λάβει το αποκωδικοποιεί με το Private Key του και στη συνέχεια ξεχωρίζονται τα στοιχεία αυτά. Το μήνυμα (message) αυτό στη συνέχεια υπογράφεται ψηφιακά από τον αποστολέα με σκοπό να ελεγχθεί η ακεραιότητα του μηνύματος από τον παραλήπτη.

Check = Αφού ο Server ξεχωρίσει τα στοιχεία του πελάτη, ελέγχει στην βάση δεδομένων για τον πελάτη σύμφωνα με το PIN και το Account Number και την ενημερώνει ανάλογα με το είδος της συναλλαγής που επιθυμεί ο πελάτης. Τα αποτελέσματα που παίρνει μας τα εμφανίζει στην οθόνη.

Κρυπτογραφία

Λίγα λόγια για την κρυπτογραφία

Η λέξη **κρυπτογραφία** προέρχεται από τα συνθετικά "κρυπτός" + "γράφω" και είναι ένας επιστημονικός κλάδος που ασχολείται με την μελέτη, την ανάπτυξη και την χρήση τεχνικών κρυπτογράφησης και αποκρυπτογράφησης με σκοπό την απόκρυψη του περιεχομένου των μηνυμάτων.

Η κρυπτογραφία είναι ένας κλάδος της επιστήμης της κρυπτολογίας, η οποία ασχολείται με την μελέτη της ασφαλούς επικοινωνίας. Ο κύριος στόχος της είναι να παρέχει μηχανισμούς για 2 ή περισσότερα μέλη να επικοινωνήσουν χωρίς κάποιος άλλος να είναι ικανός να διαβάζει την πληροφορία εκτός από τα μέλη. Η λέξη κρυπτολογία αποτελείται από την ελληνική λέξη "κρύπτος" και την λέξη "λόγος" και χωρίζεται σε δύο κλάδους: την **Κρυπτογραφία** και την **Κρυπτανάλυση**.

Ιστορικά η κρυπτογραφία χρησιμοποιήθηκε για την κρυπτογράφηση μηνυμάτων δηλαδή μετατροπή της πληροφορίας από μια κανονική κατανοητή μορφή σε έναν γρίφο, που χωρίς την γνώση του κρυφού μετασχηματισμού θα παρέμενε ακατανόητος. Κύριο χαρακτηριστικό των παλαιότερων μορφών κρυπτογράφησης ήταν ότι η επεξεργασία γινόταν πάνω στην γλωσσική δομή. Στις νεότερες μορφές η κρυπτογραφία κάνει χρήση του αριθμητικού ισοδύναμου, η έμφαση έχει μεταφερθεί σε διάφορα πεδία των μαθηματικών, όπως διακριτά μαθηματικά, θεωρία αριθμών, θεωρία πληροφορίας, υπολογιστική πολυπλοκότητα, στατιστική και συνδυαστική ανάλυση.

Η κρυπτογραφία παρέχει 4 βασικές λειτουργίες (αντικειμενικοί σκοποί):

- **Εμπιστευτικότητα:** Η πληροφορία προς μετάδοση είναι προσβάσιμη μόνο στα εξουσιοδοτημένα μέλη. Η πληροφορία είναι ακατανόητη σε κάποιον τρίτο.
- **Ακεραιότητα:** Η πληροφορία μπορεί να αλλοιωθεί μόνο από τα εξουσιοδοτημένα μέλη και δεν μπορεί να αλλοιώνεται χωρίς την ανίχνευση της αλλοίωσης.
- **Μη απάρνηση:** Ο αποστολέας ή ο παραλήπτης της πληροφορίας δεν μπορεί να αρνηθεί την αυθεντικότητα της μετάδοσης ή της δημιουργίας της.
- **Πιστοποίηση:** Οι αποστολέας και παραλήπτης μπορούν να εξακριβώνουν τις ταυτότητές τους καθώς και την πηγή και τον προορισμό της πληροφορίας με διαβεβαίωση ότι οι ταυτότητές τους δεν είναι πλαστές.

Ορολογία

Κρυπτογράφηση (encryption) ονομάζεται η διαδικασία μετασχηματισμού ενός μηνύματος σε μία ακατανόητη μορφή με την χρήση κάποιου κρυπτογραφικού αλγορίθμου ούτως ώστε να μην μπορεί να διαβαστεί από κανέναν εκτός του νόμιμου παραλήπτη.

Η αντίστροφη διαδικασία όπου από το κρυπτογραφημένο κείμενο παράγεται το αρχικό μήνυμα ονομάζεται αποκρυπτογράφηση (decryption).

- **Αλγόριθμος κρυπτογράφησης (encryption algorithm):** ο αλγόριθμος με τον οποίον πραγματοποιούνται οι διάφοροι μετασχηματισμοί στο αρχικό μήνυμα.
- **Αρχικό κείμενο (plaintext):** είναι το μη κρυπτογραφημένο μήνυμα που αποτελεί στοιχείο εισόδου στον αλγόριθμο κρυπτογράφησης.
- **Ζεύγος δημόσιου (public) και ιδιωτικού (private) κλειδιού:** Ζεύγος κλειδιών, που έχει επιλεγεί με τρόπο ώστε, το δημόσιο κλειδί του παραλήπτη

να χρησιμοποιηθεί για κρυπτογράφηση και το ιδιωτικό κλειδί του παραλήπτη για αποκρυπτογράφηση. Οι ακριβείς μετασχηματισμοί πραγματοποιούνται από τον αλγόριθμο κρυπτογράφησης / αποκρυπτογράφησης, εξαρτώμενοι από τις τιμές του δημόσιου και του ιδιωτικού κλειδιού που παρέχονται ως είσοδοι.

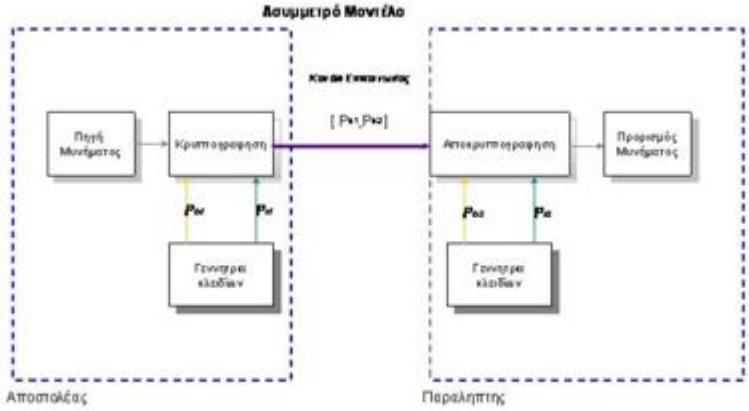
- **Κρυπτογράφημα ή κρυπτογραφημένο μήνυμα (ciphertext):** Είναι το μήνυμα που παράγεται από τον αλγόριθμο κρυπτογράφησης ως έξοδος. Εξαρτάται από το αρχικό μήνυμα και το δημόσιο κλειδί του παραλήπτη. Για ένα συγκεκριμένο μήνυμα από δύο διαφορετικά κλειδιά παράγονται από τη συνάρτηση κρυπτογράφησης δύο διαφορετικά κρυπτογραφημένα κείμενα.
- **Αλγόριθμος αποκρυπτογράφησης (decryption algorithm):** Είναι ο αλγόριθμος που δέχεται ως είσοδο το κρυπτογραφημένο μήνυμα και το ιδιωτικό κλειδί και παράγει το πρωτότυπο αρχικό μήνυμα.

Τυπικό σύστημα κρυπτογράφησης - αποκρυπτογράφησης.

Η κρυπτογράφηση και αποκρυπτογράφηση ενός μηνύματος γίνεται με τη βοήθεια ενός αλγόριθμου κρυπτογράφησης (cipher) και ενός κλειδιού κρυπτογράφησης (key). Συνήθως ο αλγόριθμος κρυπτογράφησης είναι γνωστός, οπότε η εμπιστευτικότητα του κρυπτογραφημένου μηνύματος που μεταδίδεται βασίζεται ως επί το πλείστον στην μυστικότητα του κλειδιού κρυπτογράφησης. Το μέγεθος του κλειδιού κρυπτογράφησης μετριέται σε αριθμό bits. Γενικά ισχύει ο εξής κανόνας: όσο μεγαλύτερο είναι το κλειδί κρυπτογράφησης, τόσο δυσκολότερα μπορεί να αποκρυπτογραφηθεί το κρυπτογραφημένο μήνυμα από επίδοξους εισβολείς. Διαφορετικοί αλγόριθμοι κρυπτογράφησης απαιτούν διαφορετικά μήκη κλειδιών για να πετύχουν το ίδιο επίπεδο ανθεκτικότητας κρυπτογράφησης.

Ασύμμετρα κρυπτοσυστήματα

Το ασύμμετρο κρυπτοσύστημα ή κρυπτοσύστημα δημοσίου κλειδιού δημιουργήθηκε για να καλύψει την αδυναμία μεταφοράς κλειδιών που παρουσίαζαν τα συμμετρικά συστήματα. Χαρακτηριστικό του είναι ότι έχει δυο είδη κλειδιών ένα ιδιωτικό και ένα δημόσιο. Το δημόσιο είναι διαθέσιμο σε όλους ενώ το ιδιωτικό είναι μυστικό. Η βασική σχέση μεταξύ τους είναι : ό,τι κρυπτογραφεί το ένα, μπορεί να το αποκρυπτογραφήσει μόνο το άλλο (Σχήμα 1.4).



Σχήμα 1.4 Μοντέλο Ασύμμετρου Κρυπτοσυστήματος

Εφαρμογές κρυπτογραφίας

Η εξέλιξη της χρησιμοποίησης της κρυπτογραφίας ολοένα αυξάνεται καθιστώντας πλέον αξιόπιστη την μεταφορά της πληροφορίας για διάφορους λειτουργικούς σκοπούς

1. Ασφάλεια συναλλαγών σε τράπεζες δίκτυα - ATM
2. Κινητή τηλεφωνία (ΤΕΤΡΑ-ΤΕΤΡΑΠΟΛ-GSM)
3. Σταθερή τηλεφωνία (cryptophones)
4. Διασφάλιση Εταιρικών πληροφοριών
5. Στρατιωτικά δίκτυα (Τακτικά συστήματα επικοινωνιών μάχης)
6. Διπλωματικά δίκτυα (Τηλεγραφήματα)
7. Ηλεκτρονικές επιχειρήσεις (πιστωτικές κάρτες, πληρωμές)
8. Ηλεκτρονική ψηφοφορία
9. Ηλεκτρονική δημοπρασία
10. Ηλεκτρονικό γραμματοκιβώτιο
11. Συστήματα συναγερμών
12. Συστήματα βιομετρικής αναγνώρισης
13. Έξυπνες κάρτες
14. Ιδιωτικά δίκτυα (VPN)
15. Word Wide Web
16. Δορυφορικές εφαρμογές (δορυφορική τηλεόραση)
17. Ασύρματα δίκτυα (Hipperlan, bluetooth, 802.11x)
18. Συστήματα ιατρικών δεδομένων και άλλων βάσεων δεδομένων
19. Τηλεσυνδιάσκεψη - Τηλεφωνία μέσω διαδικτύου (VOIP)

Αλγόριθμος RSA

Ένα από τα πρώτα ασύμμετρα κρυπτοσυστήματα αναπτύχθηκε το 1977 από τους R. Rivest, A. Shamir και L. Adleman στο MIT και δημοσιεύτηκε για πρώτη φορά το 1978. Από εκείνη τη στιγμή το RSA κυριάρχησε ως η πλέον αποδεκτή και εύκολα υλοποιήσιμη προσέγγιση για ασύμμετρα κρυπτοσυστήματα. Ο RSA είναι αλγόριθμος κρυπτογράφησης στον οποίο το αρχικό και το κρυπτογραφημένο κείμενο είναι ακέραιοι αριθμοί με τιμές μεταξύ 0 και n-1, για κάποιο n.

Η κρυπτογράφηση και η αποκρυπτογράφηση για ένα κείμενο M και για το αντίστοιχο κρυπτογραφημένο C συμβολίζονται ως ακολούθως:

$$C = Me \bmod n$$
$$M = Cd \bmod n = (Me)d \bmod n = Med \bmod n$$

Τόσο ο αποστολέας όσο και ο παραλήπτης θα πρέπει να γνωρίζουν τις τιμές των n και e. Αντιθέτως, την τιμή του d πρέπει να γνωρίζει μόνον ο παραλήπτης. Ουσιαστικά ο RSA είναι ένας αλγόριθμος για ασύμμετρο κρυπτοσύστημα με δημόσιο κλειδί KU={e,n} και ιδιωτικό κλειδί KR={d,n}. Για να είναι ικανοποιητικός ο αλγόριθμος αυτός θα πρέπει να ικανοποιούνται οι ακόλουθες απαιτήσεις:

- Είναι δυνατόν να βρεθούν τιμές για τα e,d,n, τέτοιες ώστε να ισχύει:
  $Med = M \bmod n$, για κάθε $M<n$.
- Είναι σχετικά εύκολο να υπολογιστούν τα Me και C, για κάθε $M<n$.
- Είναι αδύνατο να προσδιοριστεί το d, δοθέντων των e και n.

Οι δύο πρώτες απαιτήσεις ικανοποιούνται εύκολα. Η τρίτη απαίτηση μπορεί να ικανοποιηθεί μόνο για μεγάλες τιμές των e και n.

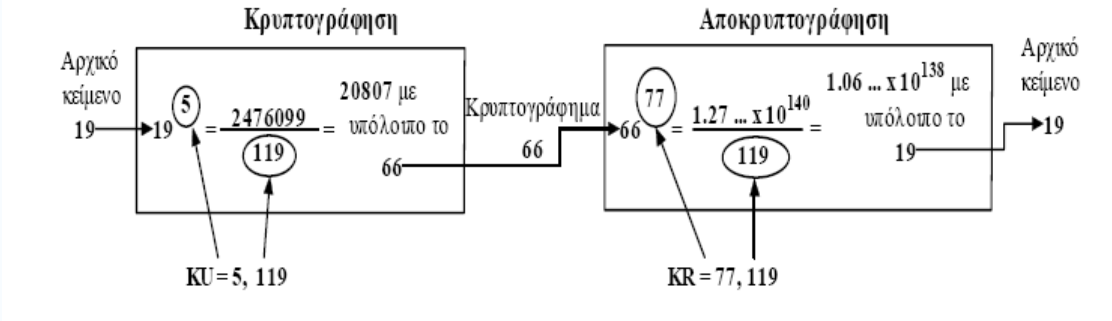
Στο Σχήμα 18 (α) ,Σχήμα 18 (β) και Σχήμα 18 (γ) παρουσιάζεται συνοπτικά ο αλγόριθμος RSA. Αρχικά επιλέγονται δύο πρώτοι αριθμοί p, q και υπολογίζεται το γινόμενό τους n, το οποίο είναι βασικός παράγοντας στη διαδικασία κρυπτογράφησης και αποκρυπτογράφησης.

Ακολούθως χρησιμοποιείται η τιμή της συνάρτησης φ(n), γνωστή ως συνάρτηση του Euler για το n (φ(n) =(p-1)(q-1)), η οποία δείχνει το πλήθος των θετικών ακεραίων που είναι μικρότεροι από n και πρώτοι με αυτόν. Επιλέγεται ένας ακέραιος αριθμός e, ο οποίος είναι πρώτος αριθμός ως προς το φ(n), δηλαδή ο μέγιστος κοινός διαιρέτης του e και του φ(n) να είναι το 1. Τελικά υπολογίζεται ο d, ως ο φυσικός αντίστροφος αριθμός του e modulo φ(n), από τη σχέση $d=e-1 \bmod \varphi(n)$. Μπορεί να αποδειχθεί ότι το d και το e πληρούν τις απαιτούμενες ιδιότητες.

Υποθέτουμε ότι ο χρήστης A έχει δημοσιεύσει το δημόσιο κλειδί του και ο χρήστης B επιθυμεί να αποστείλει ένα μήνυμα M στον A. Τότε ο B υπολογίζει την παράσταση $C=Me \bmod n$ και μεταδίδει το C. Για την αποκρυπτογράφηση του μηνύματος, ο χρήστης A υπολογίζει την παράσταση $M=Cd \bmod n$.

Σχετικό παράδειγμα περιλαμβάνεται στο Σχήμα 19. Για τις ανάγκες του παραδείγματος, τα κλειδιά δημιουργήθηκαν ως εξής:

1. Επιλέχθηκαν δύο πρώτοι αριθμοί, p=7 και q=17
2. Υπολογίσθηκε η τιμή του $n=pq=7*17=119$
3. Υπολογίσθηκε η τιμή του $\varphi(n)=(p-1)(q-1)=96$
4. Επιλέχθηκε το e, το οποίο είναι πρώτος αριθμός ως προς το $\varphi(n)=96$ και μικρότερο του φ(n). Στην περίπτωση αυτή e=5.
5. Προσδιορίστηκε το d έτσι, ώστε $de=1 \bmod 96$ και $d<96$. Η σωστή τιμή του d είναι 77, γιατί $77*5=385=4*96+1$.



Σχήμα 19 : Παράδειγμα αλγορίθμου RSA

Με τη διαδικασία αυτή υπολογίσθηκε το δημόσιο κλειδί KU={5,119} και το ιδιωτικό κλειδί KR={77,119}. Το παράδειγμα παρουσιάζει τη χρήση αυτών των κλειδιών για ένα αρχικό κείμενο με M=19. Για την κρυπτογράφηση, το 19 υψώνεται στην 5η δύναμη δίνοντας αποτέλεσμα 2476099. Διαιρούμενο με το 119 δίνει υπόλοιπο 66. Ακολούθως, 195 66 mod 199 και το κρυπτογραφημένο κείμενο είναι C=66. Για την αποκρυπτογράφηση προκύπτει ότι 6677 19 mod 119.

Υπάρχουν δύο πιθανές προσεγγίσεις με τις οποίες είναι δυνατόν να προκληθεί επιτυχημένη επίθεση στον RSA αλγόριθμο. Η πρώτη είναι η προσέγγιση της εξαντλητικής αναζήτησης: δοκιμάζονται όλα τα πιθανά ιδιωτικά κλειδιά. Έτσι, όσο μεγαλύτερο πλήθος bits χρησιμοποιείται για τα e,d τόσο πιο ασφαλής είναι ο αλγόριθμος. Παρόλα αυτά, επειδή απαιτούνται πολύπλοκοι υπολογισμοί τόσο κατά τη δημιουργία των κλειδιών όσο και κατά την κρυπτογράφηση και αποκρυπτογράφηση, όσο μεγαλύτερο είναι το μέγεθος των κλειδιών τόσο βραδύτερος θα είναι ο ρυθμός λειτουργίας του συστήματος.

Οι περισσότερες, όμως, συζητήσεις για την κρυπτανάλυση του RSA έχουν επικεντρωθεί στη διαδικασία ανεύρεσης δύο πρώτων αριθμών που να είναι παράγοντες του n. Για ένα μεγάλο αριθμό n, η διαδικασία αυτή αποτελεί δύσκολο πρόβλημα αλλά όχι σε τόσο μεγάλο βαθμό όσο ήταν τα προηγούμενα χρόνια. Για παράδειγμα, τον Ιανουάριο του 1977 οι σχεδιαστές του RSA ζήτησαν από τους αναγνώστες του επιστημονικού περιοδικού Scientific American να αποκρυπτογραφήσουν ένα κρυπτογραφημένο μήνυμα που είχαν δημοσιεύσει σε στήλη του περιοδικού. Μάλιστα, προσέφεραν αμοιβή 100 δολαρίων για μία μόνο πρόταση του αποκρυπτογραφημένου κειμένου, γεγονός που εκτιμούσαν πως δεν είναι δυνατό να συμβεί στα επόμενα 40 τετράκις εκατομμύρια χρόνια. Όμως τον Απρίλιο του 1994, μια ερευνητική ομάδα που εργαζόταν αξιοποιώντας την υπολογιστική ισχύ περισσότερων από 1600 υπολογιστές στο Internet κέρδισε το βραβείο μετά από 8 μήνες προσπάθεια. Στην περίπτωση αυτή, χρησιμοποιήθηκε δημόσιο κλειδί μεγέθους 129 δεκαδικών ψηφίων (μήκος του n), δηλαδή περίπου 428 bits. Επιπλέον, το 1996 αναλύθηκε σε γινόμενο πρώτων παραγόντων ένας αριθμός 130 ψηφίων με 10 φορές λιγότερες πράξεις από όσες είχαν απαιτηθεί κατά την ανάλυση του αριθμού με 129 ψηφία. Τα αποτελέσματα αυτά, βεβαίως, με κανένα τρόπο δε μειώνουν τις δυνατότητες του RSA. Απλώς σημαίνουν ότι πρέπει να χρησιμοποιούνται μεγαλύτερα μεγέθη κλειδιών. Ένα κλειδί μεγέθους 2048 bits θεωρείται ισχυρό για όλες τις σημερινές τυπικές εφαρμογές.

Μέγεθος κλειδιών

Στον Αλγόριθμο RSA το μήκος των κλειδιών μπορεί να είναι 512,1024 ή 2048 bits. Στην συγκεκριμένη πτυχιακή εργασία το μέγεθος των κλειδιών είναι 1024 bits.Τα κλειδιά αυτά τα διατηρεί ο Server και δημιουργούνται κάθε φορά που σηκώνουμε έναν Server. Η διάδοση των Public Key στους Client γίνεται κατά την διαδικασία αίτησης για σύνδεση με τον Server.

Η αποτελεσματικότητα των κρυπτοσυστημάτων δημόσιου κλειδιού εξαρτάται από τον δογματισμό (πρακτικά και θεωρητικά) από μαθηματικά προβλήματα παραγοντισμού ακεραίων. Τα προβλήματα έχουν σχεδόν λυθεί. Για το λόγο αυτό τα κλειδιά ασύμμετρων αλγορίθμων πρέπει να είναι μεγαλύτερα για ισοδύναμη αντίσταση σε επιθέσεις σε σχέση με τα κλειδιά των συμμετρικών αλγορίθμων. Το 2002 το μέγεθος του κλειδιού ήταν 1024 bits και θεωρούνταν το μικρότερο αναγκαίο για την κρυπτογράφηση με τον αλγόριθμο RSA.

Το 2003 η ασφάλεια RSA ισχυρίζεται ότι κλειδί 1024 bits RSA είναι ισοδύναμα με 80 bits συμμετρικού κλειδιού, τα 2048 bits είναι ισοδύναμα με 112 bits συμμετρικού κλειδιού και τα 3072 bits RSA είναι ισοδύναμα με 128 bits συμμετρικού κλειδιού. Ο RSA ισχυρίζεται ότι τα 1024 bits κλειδιού θα μπορούσαν να είναι τρωτά κάποια στιγμή μεταξύ 2006 και 2010 και ότι τα 2048 bits κλειδιού είναι αρκετά μέχρι το 2030. Επίσης το τμήμα του NIST που δίνει τις κατευθυντήριες γραμμές αναφέρει ότι τα 15360 bits RSA είναι ισοδύναμα με 256 bits συμμετρικού κλειδιού.

Το Public key είναι της μορφής:

Sun RSA public key, 1024 bits
modulus:
95989152446271183553236654785141605508610134937667716699394888254487491030194725701477390238595279595773536787844242327564566485362842302164842856113829894144013556978344400349914604871796504548596353257216442597082057812344793007636086965397206533138271788247065730561050815893176530019815374457869404666251
public exponent: 65537

Το Private key είναι της μορφής:

Sun RSA private CRT key, 1024 bits
modulus:
95989152446271183553236654785141605508610134937667716699394888254487491030194725701477390238595279595773536787844242327564566485362842302164842856113829894144013556978344400349914604871796504548596353257216442597082057812344793007636086965397206533138271788247065730561050815893176530019815374457869404666251
public exponent: 65537
private exponent:
52855036762875752641800069597805897706474418107261774150953863198493829269063842812292515396496450169716032501685081295677594500164622887196891590225060327478729801577795564620283996497292252337761963684882820946818355000939179000752256758388471859058128840299352168129201624456168448121973167641527912926577
prime p:
10299578764516067864698788917601545291559331018563204708879310464095

13444112817440352748989184457903410751076166603410272619064582501554009919830184869449

prime q:

93197163341253688426170815085232497348542054567873925789974669263381924408622224027803954592854694450586637606991951641151425434717632871220761328593492 99

prime exponent p:

94985510555465026129117109751718479244067620849590321895214233860250970987713610036303179716237947370515250170668402599581472490862626291157598690262524 9

prime exponent q:

11016653560655695625944814447797368768163866163195131041929501850609880958749963677669060778351851899059991783898662577841526051585478460310164304326011 11

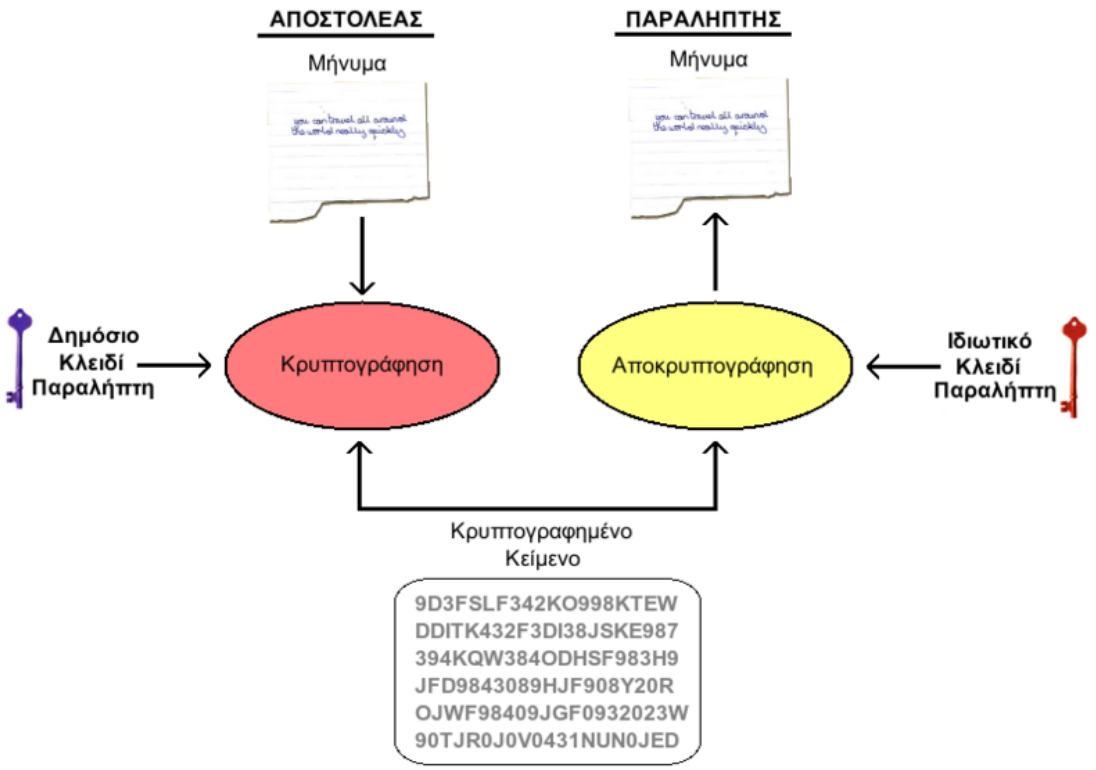
crt coefficient:

45070847651411091680118664148544616747190436966903838177458025778495440100878661849374975705761024623804284298390505796390575836492064169803685702415475 86

Εμπιστευτικότητα

Οι κρυπτογραφικοί αλγόριθμοι δημοσίου κλειδιού μπορούν να εγγυηθούν εμπιστευτικότητα (confidentiality), δηλαδή ότι το κρυπτογραφημένο μήνυμα που θα στείλει ο αποστολέας μέσω του διαδικτύου στον παραλήπτη θα είναι αναγνώσιμο από αυτόν και μόνο. Για να επιτευχθεί η εμπιστευτικότητα, ο αποστολέας θα πρέπει να χρησιμοποιήσει το δημόσιο κλειδί του παραλήπτη για να κρυπτογραφήσει το μήνυμα. Στην συνέχεια στέλνει το κρυπτογραφημένο μήνυμα στον παραλήπτη και ο τελευταίος μπορεί να το αποκρυπτογραφήσει με το ιδιωτικό κλειδί του. Δεδομένου ότι το ιδιωτικό κλειδί του παραλήπτη είναι γνωστό μονάχα στον ίδιο και σε κανέναν άλλον, μονάχα ο παραλήπτης μπορεί να αποκρυπτογραφήσει το μήνυμα και να το διαβάσει.

Άρα λοιπόν με αυτόν τον τρόπο ο αποστολέας γνωρίζει ότι το κρυπτογραφημένο μήνυμα μπορεί να αποκρυπτογραφηθεί μονάχα από τον παραλήπτη και έτσι διασφαλίζεται η εμπιστευτικότητα του μηνύματος.



Αλγόριθμος	Κρυπτογράφηση/ Αποκρυπτογράφηση	Ψηφιακή Υπογραφή	Ανταλλαγή Κλειδιών
RSA	x	x	x
Diffie-Hellman	-	-	x
DSS	-	x	-
Elliptic Curve	x	x	x

Πίνακας 5 : Αλγόριθμοι δημοσίου κλειδιού και υποστηριζόμενες εφαρμογές

Λειτουργία

Ο αλγόριθμος RSA βασίζεται στην δυσκολία παραγοντοποίησης μεγάλων αριθμών (σήμερα, συνήθως της τάξης των 1024 με 2048 bits). Χρησιμοποιούνται δυο κλειδιά, ένα δημόσιο κατά την διάρκεια της κρυπτογράφησης και ένα κρυφό για την αποκρυπτογράφηση.

Δημιουργία των κλειδιών

1. Επιλογή δυο τυχαίων (μεγάλων) πρώτων αριθμών pκαι q έτσι ώστε $p \neq q$
2. Υπολογίζουμε $n = p.q$
3. Υπολογίζουμε την συνάρτηση του Όιλερ, $\phi(n) = (p-1)(q-1)$.
4. Επιλογή ενός αριθμού $e > 1$έτσι ώστε $e \wedge \phi(n) = 1$.
5. Υπολογίζουμε τον αριθμό d έτσι ώστε $d.e \equiv 1 \mod \phi(n)$.

- Για την εύρεση πρώτων αριθμών χρησιμοποιούνται πιθανολογικοί αλγόριθμοι.
- Συνηθισμένες επιλογές για το eείναι το 3, 7 και 216 + 1. Μικροί αριθμοί οδηγούν σε ταχύτερους υπολογισμούς αλλά και σε πιο αδύνατη ασφάλεια.

Τα κλειδιά είναι τα εξής:

- δημόσιο: (n, e)
- κρυφό: (n, d)

Μπορούμε τώρα να δημοσιεύσουμε το πρώτο κλειδί, δίνοντας έτσι την δυνατότητα σε οποιονδήποτε να μας στείλει κρυπτογραφημένα μηνύματα που μόνο εμείς (χάρη στο κρυφό κλειδί) μπορούμε να αποκρυπτογραφήσουμε.

Παραγωγή κλειδίου

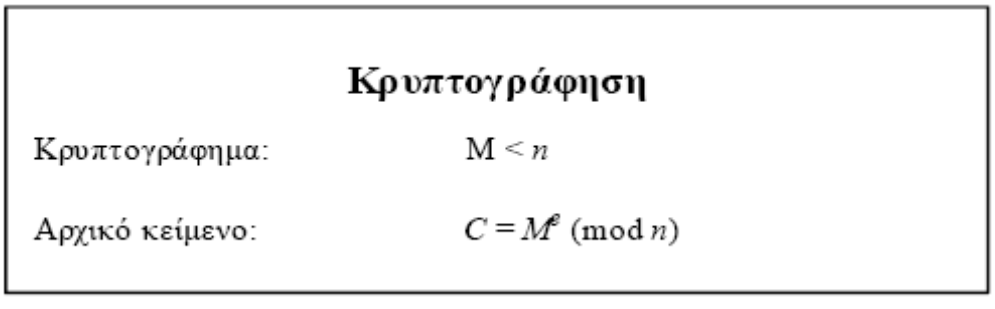
Επέλεξε p, q	p και q κι οι δύο πρώτοι
Υπολόγισε $n = p$ x q	
Υπολόγισε $\phi(n) = (p - 1)(q - 1)$	
Επέλεξε ακέραιο e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Υπολόγισε d	$d = e^{-1} \bmod \phi(n)$
Δημόσιο κλειδί	$KU = \{e, n\}$
Ιδιωτικό κλειδί	$KR = \{d, n\}$

Σχήμα 18 (α): Αλγόριθμος RSA

Κρυπτογράφηση

Το μήνυμα μπορεί να αντιπροσωπευθεί από έναν αριθμό m(π.χ. "RSA" → 0x525341, όπου *0x52* είναι ο δεκαεξαδικός κωδικός ASCII του χαρακτήρα *R*, *0x53* του *S* και τέλος *0x41* του *A*). Το κρυπτογραφημένο μήνυμα c υπολογίζεται με τον εξής τρόπο:

$$c = m^e \mod n$$



Σχήμα 18 (β) Αλγόριθμος RSA

Αποκρυπτογράφηση

Αφού ληφθεί ένα κρυπτογραφημένο μήνυμα c, για να διαβάσουμε το αρχικό μήνυμα προβαίνουμε στον ακόλουθο υπολογισμό:

$$m = c^d \mod n \equiv (m^e)^d \mod n \equiv m^{e.d} \mod n$$

Ξέρουμε πως e.d ≡ 1 (mod p-1) και e.d ≡ 1 (mod q-1), όποτε με το μικρό θεώρημα του Φερμά, έχουμε:

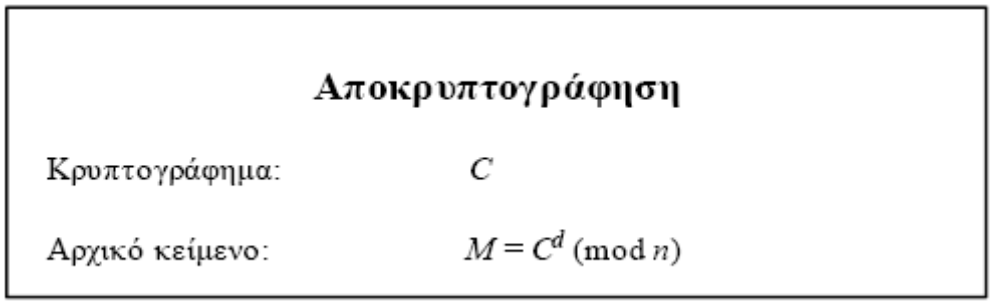
$$m^{e.d} \equiv m^1 \equiv m \mod p-1$$

και

$$m^{e.d} \equiv m^1 \equiv m \mod q-1$$

Οι αριθμοί p και q είναι πρώτοι μεταξύ τους, χρησιμοποιώντας λοιπόν το Κινέζικο Θεώρημα Υπολοίπων, έχουμε:

$$m^{e.d} \equiv m \mod n$$



Σχήμα 18 (γ) Αλγόριθμος RSA

Ασφάλεια

Αν και ο αλγόριθμος θεωρείται σχετικά ασφαλής, η κακή του χρήση μπορεί να οδηγήσει σε μεγάλες αδυναμίες ασφάλειας

Θεωρητικά, ένας μόνος κρυπταλγόριθμος παρέχει πλήρη ασφάλεια, το one-time-pad. Ο αλγόριθμος RSA είναι ασφαλής διότι η σημερινή υπολογιστική ισχύ καθιστά το πρόβλημα της παραγοντοποίησης του n, για μεγάλα n, δύσκολο, δηλαδή πάρα πολύ χρονοβόρο (μετριέται σε χρόνια)

Αλγόριθμος SHA-1

Ο Αλγόριθμος Secure Hash Algorithm – SHA αναπτύχθηκε από το US National Institute of Standards and Technology και εκδόθηκε ως Federal Information Processing Standard FIPS PUB 180 το 1993. Μια αναθεωρημένη έκδοση του ανακοινώθηκε ως FIPS PUB 180-1 το 1995 και έχει γίνει ευρύτερα γνωστή και αποδεκτή με το όνομα SHA-1.

Ο αλγόριθμος λαμβάνει ως είσοδο ένα μήνυμα με μέγιστο μήκος 2^{64} bits και παράγει ως έξοδο μια σύνοψη του μηνύματος μήκους 160 bits. Για την επεξεργασία της εισόδου απαιτείται χωρισμός του μηνύματος σε τμήματα των 512 bits. Στο Σχήμα 14 περιγράφεται η διαδικασία παραγωγής σύνοψης ενός μηνύματος, η οποία αποτελείται από τα ακόλουθα βήματα:

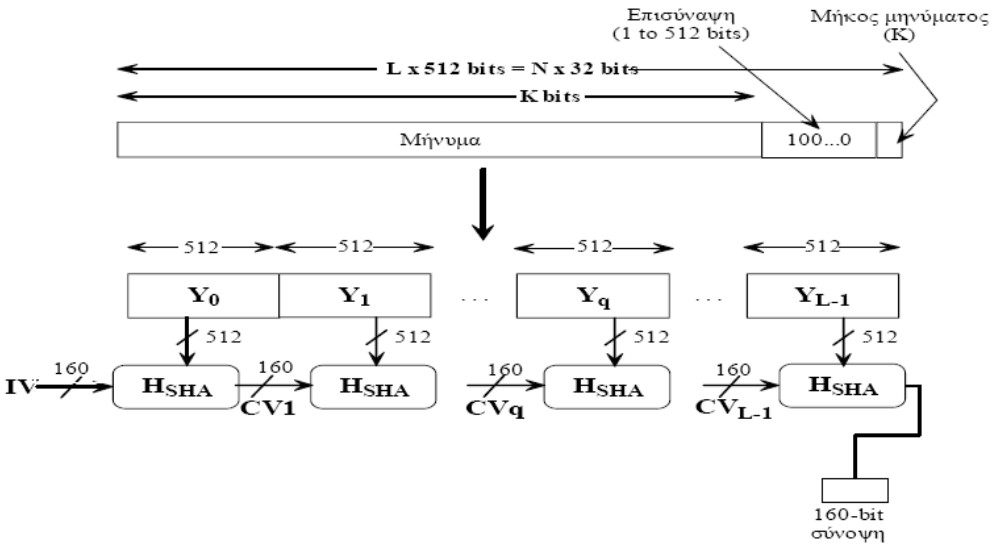
Βήμα 1: Επισύναψη Επιπρόσθετων Ψηφίων

Στο μήνυμα προσαρτώνται κάποια bits για να διατηρείται το μήκος του μηνύματος ίσο με 448 mod 512. Αυτό συμβαίνει, ώστε το μήκος του μηνύματος να είναι πάντοτε 64-bits μικρότερο από κάποιο πολλαπλάσιο του 512. Η προσάρτηση λαμβάνει χώρα πάντοτε, ακόμη και στις περιπτώσεις που το μήνυμα έχει το επιθυμητό μήκος. Το πλήθος των bits που προστίθενται κυμαίνεται από 1 έως 512 bits. Η προσάρτηση ψηφίων αποτελείται από ένα 1-bit και ακολουθείται από τον απαιτούμενο αριθμό των 0-bits.

Βήμα 2: Επισύναψη του Μήκους του Μηνύματος

Ένα τμήμα των 64-bits προστίθεται στο μήνυμα. Το τμήμα αυτό συμπεριφέρεται ως ένας μη προσημασμένος ακέραιος 64-bit, με το πιο σημαντικό byte πρώτο, ο οποίος δηλώνει το μήκος του αρχικού μηνύματος, προφανώς πριν την επισύναψη των πρόσθετων ψηφίων. Η επισύναψη της τιμής του μήκους καθιστά λιγότερο αποτελεσματική την επίτευξη ενός είδους επίθεσης, γνωστής στη βιβλιογραφία ως “επίθεση τύπου επισύναψης” (padding attack) [2].

Το αποτέλεσμα των δύο πρώτων βημάτων είναι ένα μήνυμα με μέγεθος έναν ακέραιο αριθμό πολλαπλάσιο των 512 bits. Στο Σχήμα 14 το επαυξημένο μήνυμα αναπαριστάται με μια ακολουθία τμημάτων Y0Y1…YL-1 των 512 bits το καθένα, ώστε το συνολικό μήκος του μηνύματος να είναι L*512 bits. Ισοδύναμα, το αποτέλεσμα είναι πολλαπλάσιο των 16 λέξεων των 32 bits η καθεμία. Αν οι λέξεις του παραγόμενου μηνύματος είναι M[0…N-1] με N ακέραιο πολλαπλάσιο του 16, τότε N=L*16.

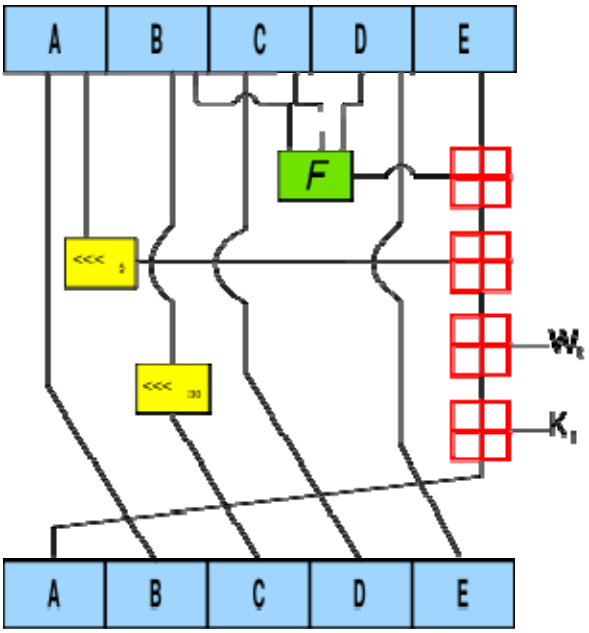


Σχήμα 14 : Διαδικασία παραγωγής σύνοψης μηνύματος

Βήμα 3: Αρχικοποίηση του Καταχωρητή MD

Για την αποθήκευση ενδιάμεσων και τελικών αποτελεσμάτων της συνάρτησης σύνοψης χρησιμοποιείται ένας καταχωρητής (buffer) των 160 bits. Ο καταχωρητής μπορεί να αναπαρασταθεί με 5 μικρότερους καταχωρητές (registers) των 32 bits (A, B, C, D, E). Αυτοί οι καταχωρητές αρχικοποιούνται με τις εξής τιμές, εκφρασμένες στο δεκαεξαδικό σύστημα:

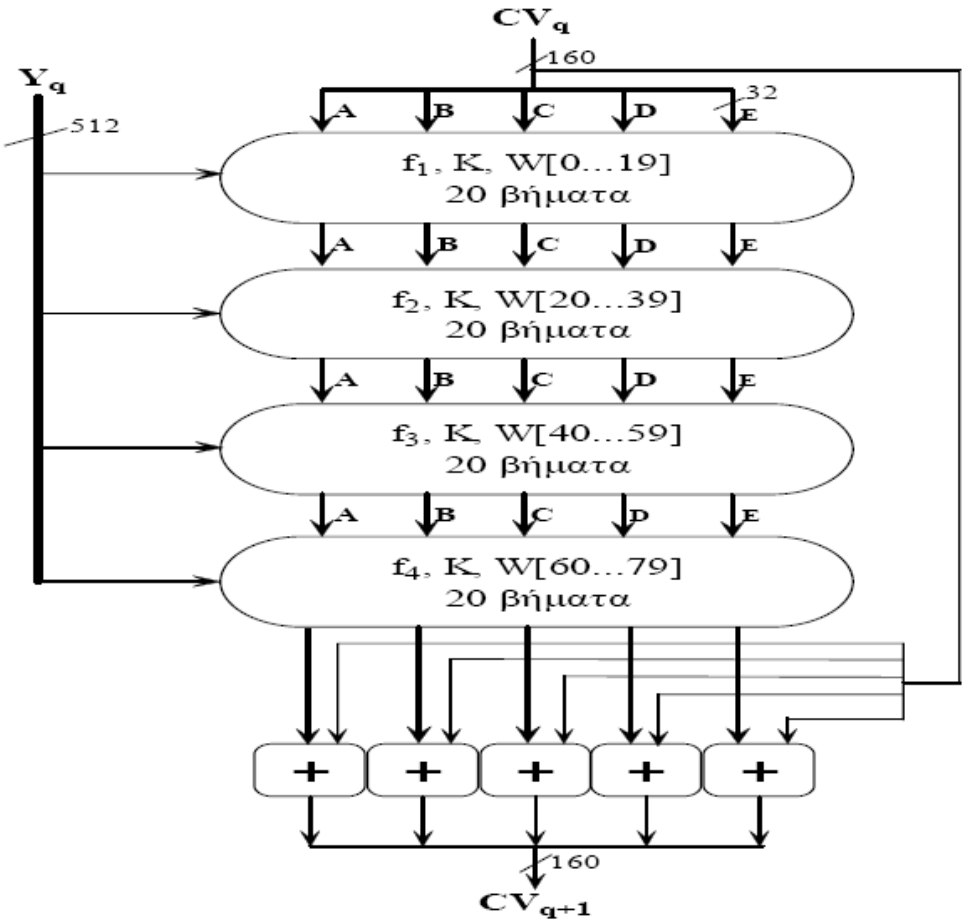
A = 67452301
B = EFCDAB89
C = 98BADCFE
D = 10325476
E = C3D2E1F0



Βήμα 4: Επεξεργασία Μηνύματος σε Τμήματα των 512 bits ή ισοδύναμα των 16 λέξεων

Το βασικότερο στοιχείο του αλγορίθμου είναι ένα τμήμα κώδικα γνωστό ως συνάρτηση συμπίεσης (compression function), η οποία αποτελείται από 4 κύκλους επεξεργασίας των 20 βημάτων ο καθένας. Η λογική της συνάρτησης απεικονίζεται στο Σχήμα 15. Οι τέσσερις κύκλοι έχουν παρόμοια δομή, αλλά ο καθένας χρησιμοποιεί διαφορετικές αρχικές λογικές συναρτήσεις τις f1, f2, f3, f4. Ο κάθε κύκλος λαμβάνει ως είσοδο το τρέχον τμήμα των 512 bits που επεξεργάζεται (έστω Yq το τρέχον τμήμα) και την τιμή ABCDE που υπάρχει στον καταχωρητή των 160-bit και ενημερώνει τα περιεχόμενα του. Επίσης, κάθε κύκλος χρησιμοποιεί μια πρόσθετη σταθερά Kt, όπου 0 ≤ t ≤ 79, η οποία υποδεικνύει κάποιο από τα 80 βήματα των 5 κύκλων. Στην πραγματικότητα υπάρχουν μόνον 4 διακριτές σταθερές που χρησιμοποιούνται. Οι τιμές αυτές στο δεκαδικό και στο δεκαεξαδικό σύστημα είναι:

Αριθμός βημάτων	**Δεκαεξαδικό**	**Λήψη τμήματος ακεραίου:**
$0 \le t \le 19$	K_t = 5A827999	$[2^{30} \text{ x} \sqrt{2}\ ]$
$20 \le t \le 39$	K_t = 6ED9EBA1	$[2^{30} \text{ x } \sqrt{3}\ ]$
$40 \le t \le 59$	K_t = 8F1BBCDC	$[2^{30} \text{ x } \sqrt{5}\ ]$
$60 \le t \le 79$	K_t =CA62C1D6	$[2^{30} \text{ x } \sqrt{10}\ ]$



Σχήμα 15: Λειτουργία συνάρτησης συμπίεσης

Η έξοδος του τέταρτου κύκλου, ή ισοδύναμα του 80ου βήματος, προστίθεται στην είσοδο του πρώτου κύκλου (CVq) ώστε να παραχθεί το CVq+1. Η πρόσθεση γίνεται ανεξάρτητα για καθεμία από τις 5 λέξεις του καταχωρητή για την καθεμία από τις λέξεις στο CVq, χρησιμοποιώντας πρόσθεση υπολοίπου 2^{32}.

Βήμα 5: Έξοδος
Μετά την επεξεργασία όλων των τμημάτων L, η έξοδος από το τελευταίο στάδιο είναι η σύνοψη των 160 bits του αρχικού μηνύματος.

Ο αλγόριθμος SHA-1 έχει την ιδιότητα όλα τα bits του κωδικού σύνοψης να είναι αποτέλεσμα εφαρμογής συνάρτησης σε όλα τα bits της εισόδου. Η περίπλοκη επανάληψη της βασικής συνάρτησης f παράγει «καλά ανομοιογενή» (well–mixed) αποτελέσματα. Για παράδειγμα, είναι αδύνατο δύο τυχαία επιλεγμένα μηνύματα, ακόμη και αν παρουσιάζουν όμοια κανονικότητα, να παράγουν την ίδια σύνοψη. Στο βαθμό που δεν υπάρχει κάποια εγγενής σχεδιαστική αδυναμία του αλγορίθμου SHA-1, που δεν έχει ακόμη αποκαλυφθεί και δημοσιευτεί, η δυσκολία να υπάρξουν δύο μηνύματα με την ίδια σύνοψη είναι της τάξεως των 2^{80} λειτουργιών, ενώ η δυσκολία

ανεύρεσης του μηνύματος δοθείσης μιας σύνοψης είναι της τάξεως των 2^{160} λειτουργιών.

Επιθέσεις στον SHA-1

Η ερευνητική ομάδα των Xiaoyun Wang, Yiqun Lisa Yin και Hongbo Yu (κυρίως από το Πανεπιστήμιο Shandong της Κίνας) ειδικοί της κρυπτογραφίας απέδειξαν ότι ο SHA-1 δεν είναι απρόσβλητος από συγκρούσεις. Δηλαδή, ανέπτυξαν έναν αλγόριθμο που ανακαλύπτει τις συγκρούσεις πιο γρήγορα από τη μέθοδο εξάντλησης όλων των πιθανών συνδυασμών (brute force).

Ο SHA-1 παράγει κρυπτογράφημα μήκους 160-bit. Δηλαδή, κάθε μήνυμα κατατεμαχίζεται σε έναν αριθμό 160-bit. Δεδομένου ότι υπάρχει ένας άπειρος αριθμός μηνυμάτων που κατατεμαχίζονται σε κάθε πιθανή τιμή, υπάρχει και ένας άπειρος αριθμός πιθανών συγκρούσεων. Όμως, επειδή, ο αριθμός των πιθανών κατατεμαχισμών είναι τόσο μεγάλος, οι πιθανότητες ανεύρεσης ενός κατά τύχη είναι υπερβολικά μικρές (μία στις 2^80 για την ακρίβεια). Αν κατατεμαχίσετε 2^80 τυχαία μηνύματα, θα βρείτε ένα ζεύγος που κατατεμαχίστηκε έχοντας το ίδιο αποτέλεσμα. Πρόκειται για τη μέθοδο ανεύρεσης συγκρούσεων εξαντλώντας όλους τους πιθανούς συνδυασμούς (brute force) και εξαρτάται αποκλειστικά από το μήκος της τιμής κατακερματισμού (hash value). "Σπάζοντας" τη συνάρτηση hash σημαίνει ότι υπάρχει η δυνατότητα να βρεθούν συγκρούσεις πιο γρήγορα από αυτό τον τρόπο. Αυτό ακριβώς έκαναν οι Κινέζοι.

Μπορούν να βρουν συγκρούσεις στον SHA-1 στους 2^69 υπολογισμούς, περίπου 2,000 φορές πιο γρήγορα από τη μέθοδο brute force. Στις μέρες μας, κάτι τέτοιο είναι ακραίο να πραγματοποιηθεί με την τρέχουσα τεχνολογία. Δύο συγκρίσιμοι μαζικοί υπολογισμοί αποδεικνύουν ακριβώς αυτό το σκεπτικό.

Η σπουδαιότητα αυτών των αποτελεσμάτων εξαρτάται από τον αποδέκτη. Αν είναι ειδικός της κρυπτογραφίας, τότε πρόκειται για μεγάλη υπόθεση. Αν και δεν είναι ριζοσπαστικά, τα αποτελέσματα αυτά αποτελούν σημαντική πρόοδο του τομέα. Οι τεχνικές που περιγράφουν οι ερευνητές είναι πολύ πιθανό να έχουν και άλλες εφαρμογές και κατ' επέκταση, να βελτιωθεί ο σχεδιασμός των συστημάτων ασφαλείας. Έτσι προοδεύει η επιστήμη της κρυπτογραφίας: μαθαίνουμε πως να σχεδιάζουμε νέους αλγόριθμους σπάζοντας άλλους αλγόριθμους. Επιπλέον, οι αλγόριθμοι του NSA θεωρούνται ως είδος άγνωστης τεχνολογίας: προέρχονται από μια ανώτερη φυλή χωρίς επεξηγήσεις. Μία επιτυχημένη κρυπτανάλυση σχετικά με αλγόριθμους του NSA αποτελεί ένα ενδιαφέρον σημείο απέναντι στο αιώνιο ερώτημα: πόσο καλοί είναι πραγματικά;

Για το μέσο χρήστη του Διαδικτύου τα νέα αυτά δεν πρέπει να αποτελούν λόγο πανικού. Κανείς στο άμεσο μέλλον δεν θα αρχίσει να σπάει ψηφιακές υπογραφές ή να διαβάζει κρυπτογραφήματα. Ο ηλεκτρονικός κόσμος δεν είναι λιγότερο ασφαλής μετά από τις παρούσες ανακοινώσεις απ' ότι ήταν προηγουμένως.

Ωστόσο, υπάρχει ένα ρητό που χρησιμοποιεί το NSA: «Οι επιθέσεις πάντοτε βελτιώνονται, ποτέ δεν χειροτερεύουν.» Όπως η επίθεση αυτή βασίζεται σε άλλες ανακοινώσεις που περιγράφουν επιθέσεις κατά απλοποιημένων εκδόσεων των SHA-1, SHA-0, MD4 και MD5, έτσι κι άλλοι ερευνητές θα βασιστούν στο παρόν αποτέλεσμα. Η επίθεση κατά του SHA-1 θα συνεχίσει να βελτιώνεται, καθώς όλο και περισσότερος κόσμος θα τη διαβάσει και θα αναπτύξει γρηγορότερα τεχνάσματα, βελτιστοποιήσεις, κλπ. Ο Νόμος του Moore θα συνεχίσει να είναι πρωτοπόρος

καθιστώντας ακόμη και την υπάρχουσα επίθεση ακόμη πιο γρήγορη και πιο οικονομική.

Ο Jon Callas, CTO της εταιρείας PGP το έθεσε καλύτερα:

- «Είναι καιρός να προωθηθούμε, χωρίς να τρέξουμε, προς την έξοδο κινδύνου. Μπορεί να μην βλέπουμε τον καπνό, αλλά ο συναγερμός έχει σημάνει».
- «Είναι καιρός να απομακρυνθούμε από τον αλγόριθμο SHA-1.
- «Ευτυχώς, υπάρχουν εναλλακτικές λύσεις».

Το Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας των ΗΠΑ (NIST) ήδη διαθέτει πρότυπα για μεγαλύτερες – και πιο δύσκολες στην παραβίαση – συναρτήσεις hash: SHA-224, SHA-256, SHA-384 και SHA-512. Αυτές αποτελούν ήδη κυβερνητικά πρότυπα και μπορεί να χρησιμοποιούνται ήδη. Πρόκειται για μία καλή προσωρινή λύση.

Hash Functions

- Για οποιουδήποτε μεγέθους μήνυμα παράγει μία συγκεκριμένου μεγέθους σύνοψη h = H(M)
- Ο αλγόριθμος είναι γνωστός και δεν χρησιμοποιεί κλειδί (vs MAC)
- Χρησιμοποιείται κυρίως για να αποκαλύψει τυχόν αλλαγές στο μήνυμα
- Είναι δυνατό να χρησιμοποιηθεί και για άλλους λόγους, π.χ. Ψηφιακές υπογραφές

Ιδιότητες:

1. Εφαρμόζεται σε οποιουδήποτε μεγέθους μήνυμα M
2. Η έξοδος είναι συγκεκριμένου μεγέθους h
3. Είναι εύκολο να υπολογιστεί το h=H(M) για κάθε M
4. Δοθέντος του h είναι αδύνατο να βρεθεί x έτσι ώστε H(x)=h
   - Ιδιότητα one-way
5. Δοθέντος του x είναι αδύνατο να βρεθεί y έτσι ώστε H(y)=H(x)
   - Ιδιότητα weak collision resistance
6. Είναι αδύνατο να βρεθεί οποιοδήποτε x,y έτσι ώστε H(y)=H(x)
   - Ιδιότητα strong collision resistance

	bit 1	bit 2	• • •	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 11.7 Simple Hash Function Using Bitwise XOR

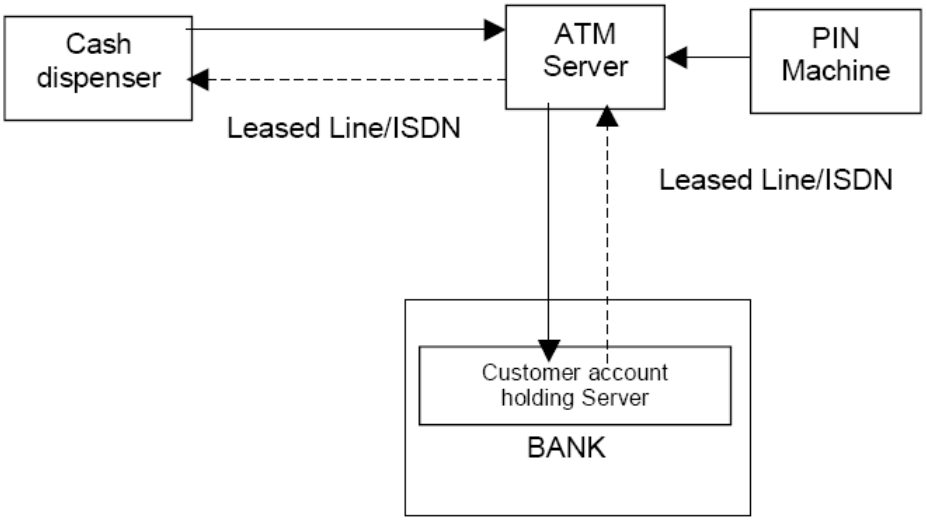
Λειτουργία και ασφάλεια των ATM στην πραγματικότητα

Πρόλογος

Οι αυτόματες μηχανές ανάληψης (ATM) εμφανίζονται παντού και μας επιτρέπουν να κάνουμε ανάληψη από την τράπεζα 24 ώρες το 24ωρο , 7 μέρες την εβδομάδα ,365 μέρες το χρόνο με την χρήση της κάρτας ανάληψης. Η κάρτα ανάληψης αποτελείται από δυο συστατικά, τον αριθμό της κάρτας και τον προσωπικό αριθμό αναγνώρισης (PIN). Κάθε τράπεζα εκδίδει έναν αριθμό κάρτας ο οποίος είναι μοναδικός για κάθε πελάτη. Εάν είναι μια χρεωστική κάρτα ο αριθμός αυτός θα είναι επίσης μοναδικός παγκοσμίως. Το PIN είναι ένας κωδικός που ελέγχει την αυθεντικότητα του πελάτη. Τα ATM ελέγχουν και τον αριθμό της κάρτας και το PIN. Στο σημείο αυτό θα εξετάσουμε τον σχεδιασμό ασφάλειας πίσω από το PIN καθώς θα παρουσιάσουμε ακόμα και το πώς αποθηκεύονται και διαχειρίζονται τα στοιχεία αυτά με ασφάλεια.

Αρχή λειτουργίας ATM

Τα συστήματα ATM έχουν 3 κύρια συστατικά: τον Cash dispenser, τον ATM Server και το PIN machine



Ο Cash dispenser διαβάζει τον αριθμό της κάρτας και το PIN το οποίο εισάγεται από τον πελάτη με αποτέλεσμα στη συνέχεια αυτά τα δυο να στέλνονται στον κεντρικό ATM Server. Ο ATM Server έχει μια βάση δεδομένων η οποία αποθηκεύει τον αριθμό της κάρτας ATM καθώς και λεπτομέρειες του PIN. Το τρίτο συστατικό το PIN machine χρησιμοποιείται για να επικυρώσει τον αριθμό ATM PIN του πελάτη. Συνδέεται άμεσα με τον ATM Server και είναι μια συσκευή απόδειξης πλαστογραφήσεων που αποθηκεύει ένα μοναδικό μυστικό κλειδί. Ο πελάτης αφού πρώτα εισάγει την κάρτα του στο ATM και πληκτρολογήσει το PIN του ο Cash dispenser θα διαβάσει τον αριθμό της κάρτας από την μαγνητική ταινία καθώς και το PIN που έχει πληκτρολογηθεί και θα τα στείλει στον ATM Server. Ο ATM Server ελέγχει το PIN σε σχέση με τον αριθμό της κάρτας με την βοήθεια του PIN machine και στέλνει μια θετική ή αρνητική επιβεβαίωση στον Cash dispenser. Σε αυτό το σημείο ο πελάτης έχει επικυρωθεί και μπορεί να χρησιμοποιήσει τον λογαριασμό του.

Ασφάλεια ATM PIN

Η ασφάλεια του ATM PIN είναι ένα κριτικό σημείο στην όλη διαδικασία. Είναι η πιο μυστική πληροφορία του πελάτη. Για αυτό το λόγο η ασφάλεια ATM PIN είναι πολύ αυστηρή σε όλα τα δίκτυα ATM. Υπάρχουν δυο τρόποι όπου ένας επιτιθέμενος θα προσπαθούσε να πάρει τον αριθμό ATM PIN. Θα μπορούσε να εισχωρήσει στο δίκτυο όταν ένας Cash dispenser μεταδίδει το PIN στον ATM Server ή θα μπορούσε να αναγκάσει έναν ATM Server και ένα Pin machine να «παραχωρήσουν» το PIN του χρήστη.

Ευτυχώς αυτές οι απειλές έχουν εξαλειφθεί στα σημερινά συστήματα ATM και θα πρέπει να δούμε το πώς. Για να αποτρέψουμε την εισχώρηση και την κλοπή του PIN κατά τη διαδικασία της μετάδοσης, το PIN κρυπτογραφείται με την χρήση του DES και του Triple DES κρυπτογραφικού αλγορίθμου και στη συνέχεια αυτό μεταδίδεται από τον Cash dispenser στον ATM Server. Το κοινό μυστικό κλειδί αποθηκεύεται στον Cash dispenser καθώς και στον ATM Server. Αυτή η εφαρμογή αποθηκεύει το διαμοιραζόμενο κλειδί DES σε μια κρυπτογραφημένη μορφή χρησιμοποιώντας τον ιδιόκτητο αλγόριθμο του προμηθευτή (π.χ. ACI ATM software) και κατά συνέπεια έχει αποτραπεί η κλοπή του κλειδιού από την μηχανή.

Η επίλυση του δεύτερου προβλήματος είναι ενδιαφέρουσα. Το σύστημα διαχωρίζει το PIN του πελάτη σε δυο μέρη και τα αποθηκεύει σε δυο διαφορετικά σημεία. Έτσι ακόμα και όταν ένα μηχάνημα μείνει έκθετο το PIN συνεχίζει να παραμένει ασφαλές. Τώρα το πρόβλημα είναι το πώς θα διαχωρίσουμε το PIN με ασφάλεια σε δυο μέρη. Επίσης πρέπει να έχουμε στο νου μας ότι ο πελάτης μπορεί πάντα οποιαδήποτε στιγμή να αλλάξει το PIN του. Για αυτό το λόγο έχει σχεδιαστεί ένας αλγόριθμος ο οποίος επιτρέπει στο PIN του πελάτη να διαχωριστεί καθώς επίσης δίνεται η δυνατότητα στον πελάτη να αλλάξει το PIN του.

Ας θεωρήσουμε ότι το PIN του πελάτη είναι το α και ότι χωρίζεται σε δυο μέρη στο β και το γ.

$$\alpha = \beta + \gamma$$

Το β είναι ένα μεταβλητό κομμάτι του PIN και καλείται PIN Offset. Το γ είναι ένα σταθερό κομμάτι του PIN και ονομάζεται Natural PIN. Το PIN Offset αποθηκεύεται στον ATM Server και το Natural PIN δημιουργείται στο PIN machine κάθε φορά. Πως το PIN machine δημιουργεί το σταθερό κομμάτι γ για κάθε πελάτη και το διατηρεί ακόμα μυστικό? Θυμηθείτε ότι ο αριθμός της κάρτας κάθε πελάτη είναι μοναδικός. Για αυτό το σταθερό κομμάτι γ μπορεί να χρησιμοποιηθεί σαν κρυπτογραφική συνάρτηση του αριθμού κάρτας.

$$\gamma = f(\text{card\#})$$

Υπάρχουν διαφορετικές μέθοδοι για να παραχθεί ένας σταθερός αριθμός από έναν αριθμό κάρτας και μια κοινή μέθοδος είναι να το δημιουργήσουμε χρησιμοποιώντας τον αλγόριθμο DES. Το PIN machine αποθηκεύει ένα κλειδί DES σε μια μνήμη EEPROM (Electrically Erasable Programmable Read Only Memory). Αυτό το κλειδί χρησιμοποιείται για την κρυπτογράφηση του αριθμού της κάρτας με αποτέλεσμα την παραγωγή της κρυπτογραφημένης μεταβλητής με χρήση DES

$$\text{Card\#} + \text{DES key} = \text{DES encrypted value}$$

Αυτή η κρυπτογραφημένη μεταβλητή στη συνέχεια μετατρέπεται σε δεκαδικό σύστημα και τα πρώτα τέσσερα ψηφία δεσμεύονται. Αυτό είναι το Natural PIN γ. Ανακεφαλαιώνοντας η διαδικασία είναι:

DES encrypted value → Decimalized value → First 4 digits of the value = γ

Το Natural PIN, το σταθερό κομμάτι, γ δεν αποθηκεύεται πουθενά σε ολόκληρη την διαδικασία. Κανένας δεν μπορεί να κλέψει το PIN με την συγκατάθεση του PIN machine. Το PIN Offset ή β είναι το μεταβλητό κομμάτι. Όταν ο πελάτης αλλάζει το PIN του μόνο αυτό το κομμάτι αλλάζει. Γι' αυτό ακόμα και όταν ένας ATM Server παραβιάζεται μόνο το β θα αποκαλυφθεί με αποτέλεσμα να είναι άχρηστο χωρίς το γ ώστε να μπορεί να πάρει το πραγματικό PIN α του πελάτη.

Μέγεθος κλειδιών συμμετρικών αλγορίθμων

Η πολιτική των Ηνωμένων πολιτειών έχει περιορίσει την δύναμη της κρυπτογραφίας που μπορεί να σταλεί και εκτός χώρας. Για αρκετά χρόνια το όριο ήταν 40 bits. Στις μέρες μας μήκος κλειδιού 40 bits προσφέρει ελάχιστη προστασία ακόμα και σε έναν περιστασιακό επιτιθέμενο με έναν μόνο υπολογιστή. Ο περιορισμός δεν έχει ακόμα εξαλειφθεί (είναι ακόμα παράνομο να εξάγει προϊόντα κρυπτογράφησης) αλλά το όριο έφτασες δραστικά στα 128 bits μέγεθος κλειδιού το 1999 / 2000.

Όταν ο αλγόριθμος DES εμφανίστηκε το 1977, ένα μέγεθος κλειδιού της τάξης των 56 bits φαινόταν να είναι επαρκές (ωστόσο υπήρχε προβληματισμός τι στιγμή που το NSA εσκεμμένα ελάττωσε το μέγεθος του κλειδιού από την αρχική τιμή των 112 bits στον IBM Lucifer cipher, ή 64 bits, σε μια από τις εκδόσεις η οποία υιοθετήθηκε ως DES) ώστε να περιορίσουν τη δύναμη της κρυπτογράφησης σε χρήστες εκτός Ηνωμένων Πολιτειών. Κάποιοι πίστευαν ότι τα 56 bits ήταν εύθραυστα στα τέλη της δεκαετίας του 70'. Στα τέλη της δεκαετίας του 90' έγινε ξεκάθαρο ότι ο DES θα μπορούσε να σπάσει μέσα σε λίγες μέρες. Στο βιβλίο Cracking DES (O' Reilly and Associates) αναφέρεται μια επιτυχημένη προσπάθεια να σπάσει ο 56 bits DES με χρήση επίθεσης brute force από ομάδες με ελάχιστες πηγές. Τα 56 bits πλέον θεωρούνται ανεπαρκές μέγεθος για κλειδιά συμμετρικών αλγορίθμων, και ίσως να έχουν υπάρξει ανεπαρκές κάποιες στιγμές στο παρελθόν. Πιο τεχνικά και οικονομικά επιτήδειοι οργανισμοί ήταν έτοιμοι να προβούν σε αυτή την ενέργεια πριν αυτή περιγραφεί στο βιβλίο. Το Distributed net και οι εθελοντές του έσπασαν ένα 64 bits RC5 κλειδί μέσα σε μερικά χρόνια, χρησιμοποιώντας περίπου 70000, κυρίως οικιακούς υπολογιστές.

Ο DES έχει αντικατασταθεί σε πολλές εφαρμογές από τον TripleDES, ο οποίος έχει 112 bits ασφάλεια με 168 bits κλειδιού.

Διαδικασία Επικύρωσης ATM PIN

Ο μηχανισμός επικύρωσης του ATM PIN είναι πολύ απλός. Όταν ο πελάτης εισάγει την κάρτα του και πληκτρολογήσει το PIN, ο αριθμός της κάρτας και το PIN στέλνονται στον ATM Server κωδικοποιημένα. Ο ATM Server αποκωδικοποιεί τον αριθμό της κάρτας και το PIN. Πρώτα επικυρώνει τον αριθμό της κάρτας σύμφωνα με τη βάση δεδομένων. Ο έγκυρος αριθμός κάρτας, το PIN Offset β της κάρτας αυτής και το PIN που πληκτρολογήθηκε από τον πελάτη στέλνονται στο PIN machine.

Τώρα το PIN machine δημιουργεί το Natural PIN γ από τον αριθμό της κάρτας, τοποθετείται με το PIN Offset β και δημιουργείται το πραγματικό PIN α του πελάτη.

Στη συνέχεια συγκρίνει το ακριβές PIN α του πελάτη με το PIN που προμηθεύεται ο πελάτης. Αν αυτά τα δυο ταιριάζουν τότε στέλνεται θετική επιβεβαίωση στον ATM Server με αποτέλεσμα την επικύρωση του πελάτη.

Σημειώστε ότι σε αυτή την διαδικασία, το Natural PIN δεν αφήνει ποτέ το tamper proof του PIN machine, και το PIN machine δεν είναι αναγκαίο να αποθηκεύσει ατομικά PIN όλων των πελατών. Αποθηκεύει με ασφάλεια το κλειδί DES για δημιουργία του Natural PIN από τον αριθμό κάρτας του κάθε πελάτη.

Παραγωγή και διανομή του ATM PIN

Το σύστημα ATM διαπραγματεύεται με κρίσιμες πληροφορίες του πελάτη και είναι πιο ασφαλές εξ' αρχής, αλλά υπάρχουν ακόμα ρίσκα ασφαλείας κατά την διαδικασία παραγωγής και διανομής μιας νέας κάρτας και ενός PIN. Ο αριθμός κάρτας παράγεται από τον ATM Server και το PIN παράγεται από το PIN machine και από τον αριθμό της κάρτας, αλλά για πρώτη φορά το PIN Offset του καινούργιου PIN παράγεται τυχαία από το PIN machine.

Υπάρχουν δυο τρόποι να εμφανίσεις το PIN mailer. Στην πρώτη μέθοδο, ο διαχειριστής θα παράγει ένα νέο PIN χρησιμοποιώντας το PIN machine, παίρνουμε το PIN και παράγουμε την τυπωμένη αναφορά του PIN mailer. Στη δεύτερη μέθοδο ο διαχειριστής ζητάει από το PIN machine να παράγει το PIN και κατευθείαν να το τυπώσει σε έναν συνδεδεμένο εκτυπωτή και να σφραγίσει τον print mailer πριν περάσει στον διαχειριστή. Η δεύτερη μέθοδος είναι ξεκάθαρα πιο ασφαλής από την πρώτη καθώς ο διαχειριστής ποτέ δεν είναι σε θέση να γνωρίζει το μυστικό PIN.

Συμβουλές για την προστασία του PIN

- Να μην γράφουμε πουθενά το PIN. Αν χρειαστεί να το γράψουμε, να μην το φυλάμε μέσα στο πορτοφόλι. Ποτέ μην το γράφουμε πάνω στην κάρτα.
- Να αλλάζουμε το PIN κάθε 6 μήνες.
- Επιλέξτε ένα PIN που μπορείτε να θυμάστε εύκολα αλλά να μην μπορεί να συσχετιστεί με εσάς προσωπικά. Αποφύγετε στο να χρησιμοποιείται ημερομηνίες γέννησης, αριθμούς σπιτιού ή αριθμούς τηλεφώνου.
- Να στέκεστε μπροστά από το ATM και το πληκτρολόγιο όταν πληκτρολογείται το PIN. Αυτό αποτρέπει από το να δουν τον αριθμό αυτοί που βρίσκονται πίσω σας.
- Ποτέ να μην γνωστοποιείται το PIN σε κανέναν συμπεριλαμβανομένων και των υπαλλήλων της τράπεζας.
- Το PIN έχει σχεδιαστεί να είναι ένας τετραψήφιος αριθμός για να είναι εύκολο να το θυμάστε. Οι τετραψήφιοι αριθμοί είναι ευπαθής στις άμεσες επιθέσεις και για να αποτραπούν επιθέσεις τέτοιου είδους τα ATM σχεδιάστηκαν να κλειδώνουν έναν λογαριασμό για 24 ώρες μετά από 3 αποτυχημένες προσπάθειες.
- Αν κάποιος ή κάτι σας κάνουν να μην νιώθετε άνετα ακυρώστε την συναλλαγή σας και αφήστε το μηχάνημα αμέσως. Επικοινωνήστε με την τράπεζα σας για να σιγουρέψτε ότι ακυρώθηκε η συναλλαγή σας και προειδοποιήστε την τράπεζα για κάθε ύποπτο πρόσωπο.

Λειτουργία Εφαρμογής

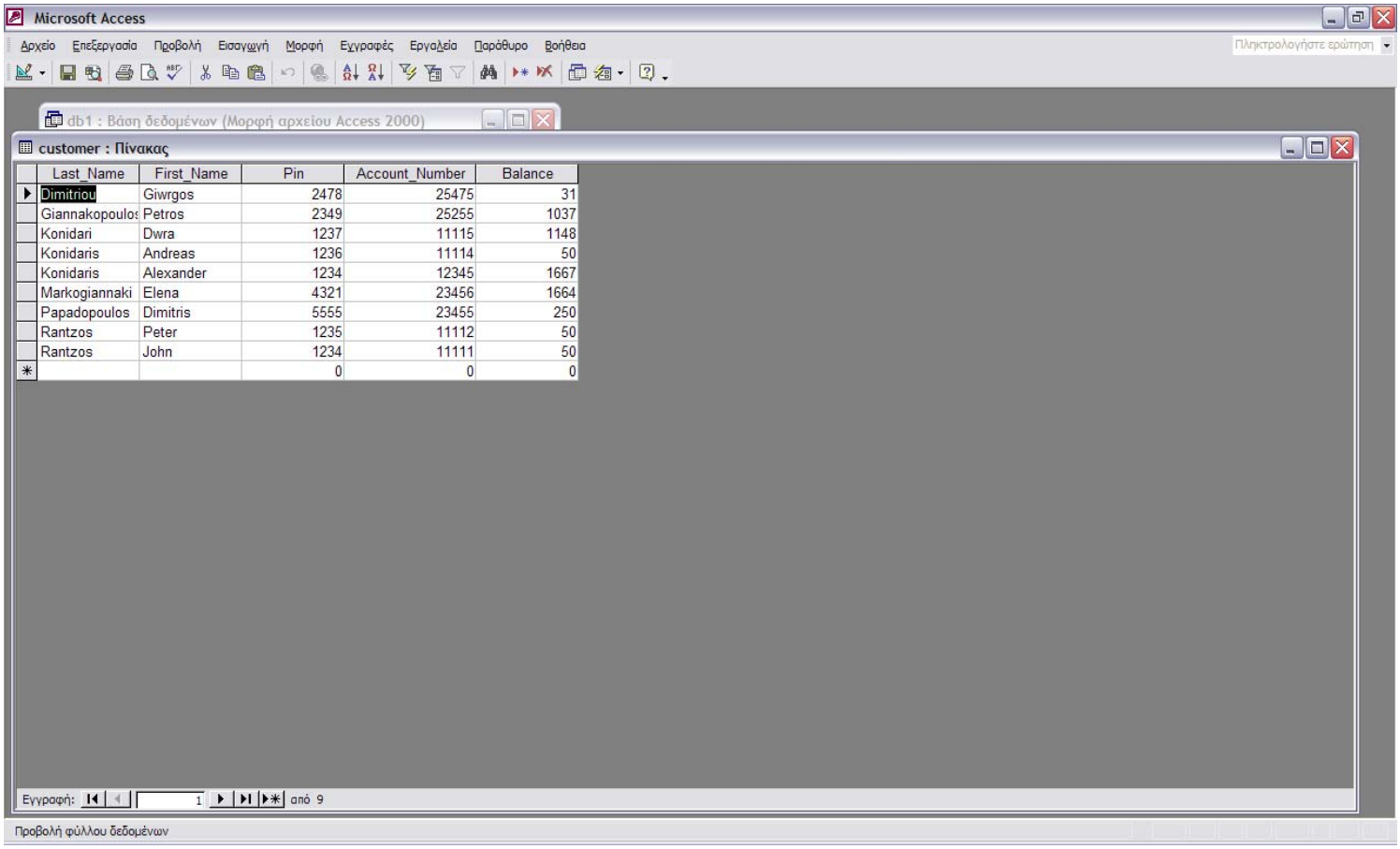
Αρχικά τρέχουμε το πρόγραμμα του Bank Server καθορίζοντάς του το port στο οποίο περιμένει τις συνδέσεις των ATM. Τώρα μπορούμε να τρέξουμε το πρόγραμμα του ATM δίνοντάς του την IP του Server και το port στο οποίο θα συνδεθεί. Υπάρχει η δυνατότητα να απευθύνονται πολλά ATM σε έναν μόνο Bank Server χωρίς αυτό να δημιουργεί πρόβλημα στη λειτουργία του συστήματος. Ο Bank Server είναι multithreaded έτσι ώστε κάθε δοσοληψία να εξυπηρετείται από ένα διαφορετικό thread.

Στο σημείο αυτό εμφανίζεται το ATM. Λόγο του ότι δεν ήταν δυνατό να να γίνει χρήση μηχανήματος ανάγνωσης καρτών ,η ανάγνωση του Account Number γίνεται εφόσον καταχωρηθεί από τον χρήστη στο αντίστοιχο πεδίο που θα του ζητηθεί. Δίνοντας το σωστό Account Number και το σωστό PIN και με την προϋπόθεση ότι υπάρχει το Public Key του Server μπορούν να εκτελεστούν οποιεσδήποτε εντολές.

Υπάρχουν 3 είδη transaction
1) Ανάληψη,
2) Κατάθεση,
3) Εμφάνιση υπολοίπου,

Για να γίνει ανάληψη / κατάθεση πρέπει πρώτα να δοθεί το Pin,το Account Number και το ποσό ,το οποίο πρέπει να είναι ακέραιος αριθμός, και στη συνέχεια να επιλεχθεί η ενέργεια που θα εκτελεστεί. Σε περίπτωση που το διαθέσιμο υπόλοιπο σε μια ανάληψη είναι μικρότερο από το ποσό που ζητά ο πελάτης εμφανίζεται μήνυμα που τον ενημερώνει για το διαθέσιμο υπόλοιπο και για το ότι δεν μπορεί να εκτελεστεί η συγκεκριμένη δοσοληψία.

Οι λογαριασμοί των χρηστών δημιουργούνται και διατηρούνται από τον Bank Server. Τα στοιχεία (Όνομα, Account Number, Balance, PIN) υπάρχουν σε ένα αρχείο (db1.mdb) με συγκεκριμένο όνομα το οποίο χρησιμοποιεί ο Server για να αρχικοποιήσει τους λογαριασμούς των χρηστών αλλά και να τα τροποποιήσει κατά τη διαδικασία εκτέλεσης εντολών.

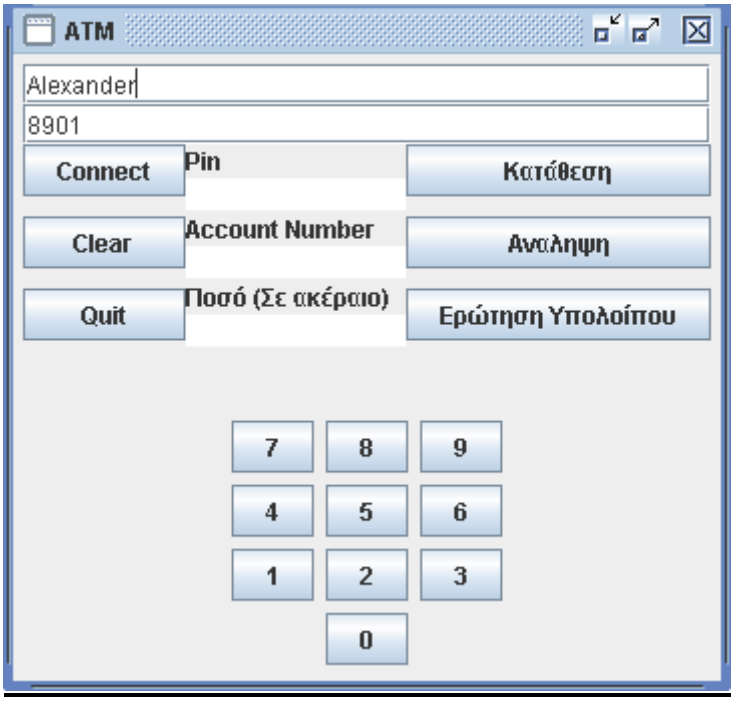


Last_Name	First_Name	Pin	Account_Number	Balance
Dimitriou	Giwrgos	2478	25475	31
Giannakopoulos	Petros	2349	25255	1037
Konidari	Dwra	1237	11115	1148
Konidaris	Andreas	1236	11114	50
Konidaris	Alexander	1234	12345	1667
Markogiannaki	Elena	4321	23456	1664
Papadopoulos	Dimitris	5555	23455	250
Rantzos	Peter	1235	11112	50
Rantzos	John	1234	11111	50
		0	0	0

Ο Bank Server μπορεί διατηρήσει ένα αρχείο με τις δοσοληψίες τις οποίες έχει εκτελέσει. Για κάθε δοσοληψία αποθηκεύει τα εξής στοιχεία: είδος δοσοληψίας, Ονοματεπώνυμο, Pin, Account Number, χρηματικό πόσο Ανάληψης / Κατάθεσης , Υπόλοιπο , Ημερομηνία και ώρα. Τα παραπάνω στοιχεία είναι αρκετά για να αποδείξουν ότι ο συγκεκριμένος χρήστης εκτέλεσε τις αποθηκευμένες δοσοληψίες.

Επεξήγηση G.U.I

Στο Tab **Setup** ο χρήστης θα πρέπει να σηκώσει τον Server με σκοπό να κάνει Listen και να δεχτεί αιτήματα σύνδεσης από τους Client (ATM).Στη συνέχεια θα πρέπει να δημιουργήσει τα κλειδιά (Public και Private Key) του Server πατώντας το κουμπί Create Keys.



Ο Client θα πρέπει να κάνει αίτημα σύνδεσης στον Server με σκοπό να συνδεθεί αλλά και να μάθει το Public Key του Server. Αυτό θα γίνει με το Connect

Ο χρήστης πλέον μπορεί να δηλώσει το Pin , το Account Number και το ποσό που επιθυμεί για να κάνει την συναλλαγή (Το ποσό δηλώνεται μόνο για κατάθεση / ανάληψη). Στη συνέχεια θα πρέπει να επιλέξει το είδος της συναλλαγής (Ανάληψη , Κατάθεση, Ερώτηση Υπολοίπου) που προτιμάει.

Αν για παράδειγμα επιλέξει να κάνει κατάθεση και τα στοιχεία του είναι
Pin: 1234
Account Number: 12345
Ποσό: 100

Το μήνυμα που θα κρυπτογραφηθεί θα έχει την μορφή 123412345100+

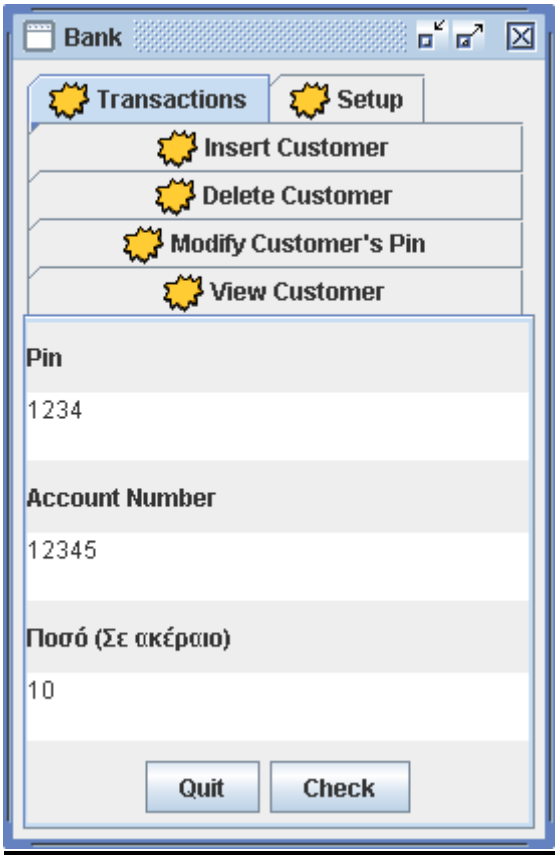
Αν επιλέξει να κάνει ανάληψη και τα στοιχεία του είναι
Pin: 1234
Account Number: 12345
Ποσό: 100

Το μήνυμα που θα κρυπτογραφηθεί θα έχει την μορφή 123412345100-

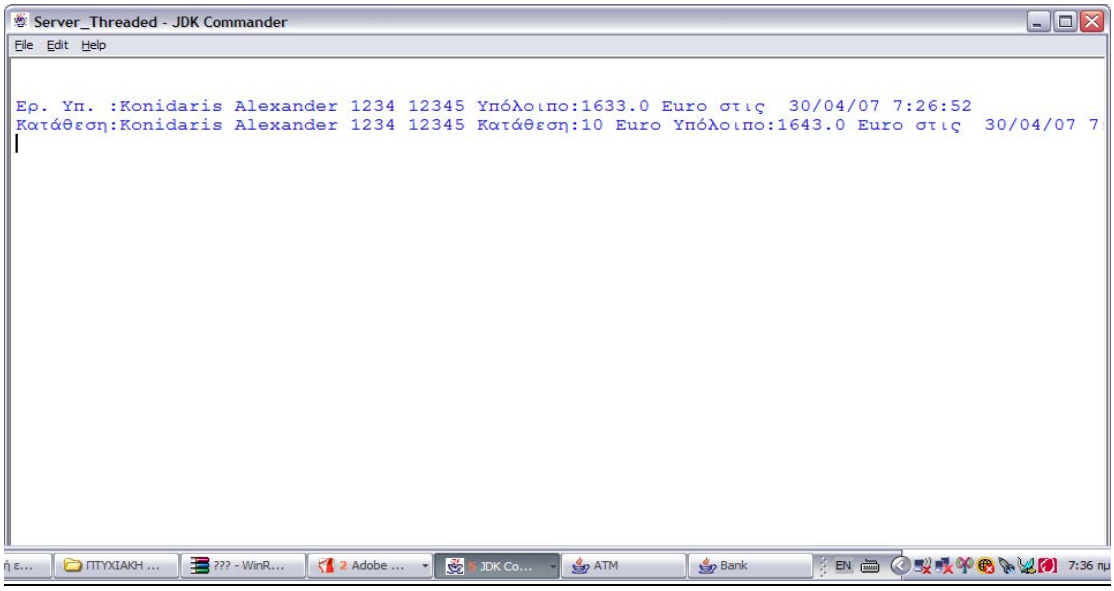
Αν επιλέξει να κάνει ερώτηση Υπολοίπου και τα στοιχεία του είναι
Pin: 1234
Account Number: 12345

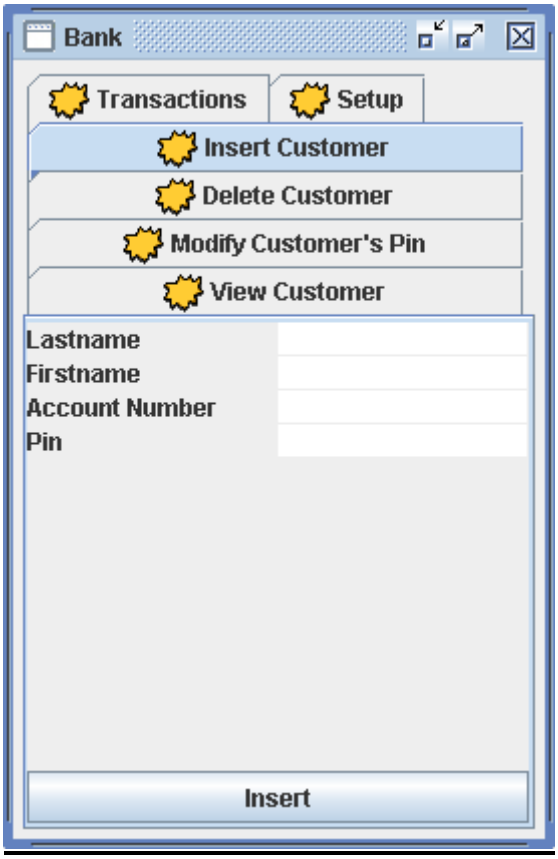
Το μήνυμα που θα κρυπτογραφηθεί θα έχει την μορφή 123412345*

Το μήνυμα αυτό θα κρυπτογραφηθεί με το Public Key του παραλήπτη (Server) και θα αποσταλεί πλέον το κρυπτογραφημένο (ciphertext) μήνυμα με σκοπό να αποκρυπτογραφηθεί και να διαχωριστεί.

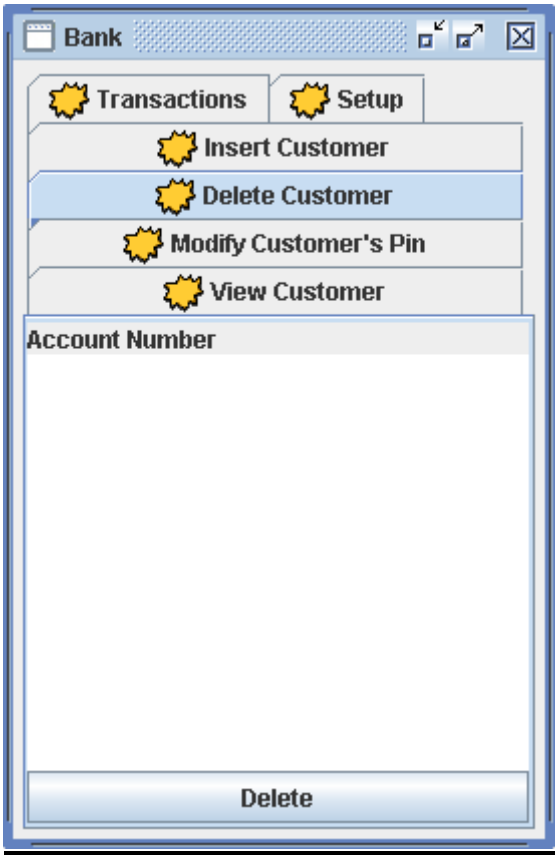


Στο Tab **Transactions** ο Server εκτελεί τις εντολές (εφόσον έχει πατηθεί το Check) που έχει λάβει από τον Client και μας επιστρέφει το αποτέλεσμα της εντολής αυτής στην οθόνη. Το αρχείο αυτό μπορεί να αποθηκευτεί σαν αρχείο .txt με σκοπό να μπορεί η τράπεζα να κρατάει αρχείο , για παράδειγμα καθημερινό, με τις συναλλαγές που έχουν γίνει κάθε μέρα. Επίσης στο tab αυτό βλέπουμε το αποτέλεσμα της αποκρυπτογράφησης του μηνύματος με την χρήση του Private Key του Server και το διαχωρισμό αυτού στα αντίστοιχα πεδία στα οποία ανήκουν.

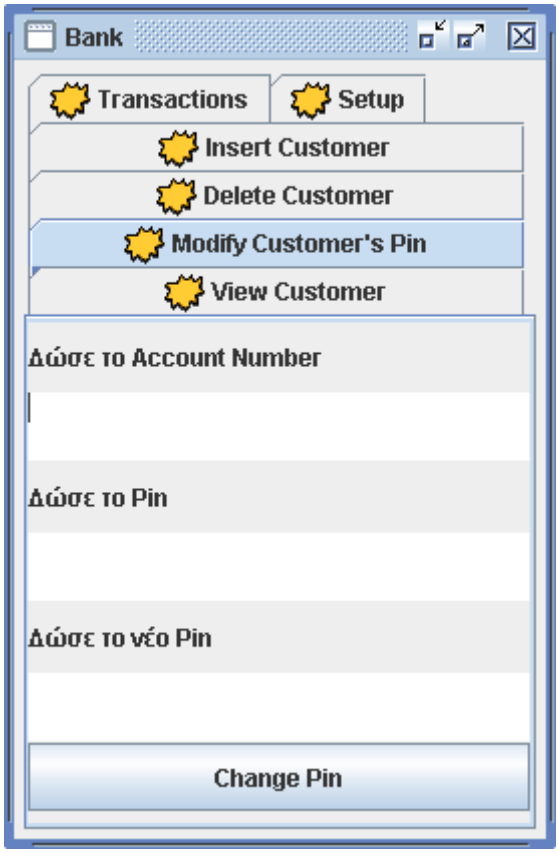




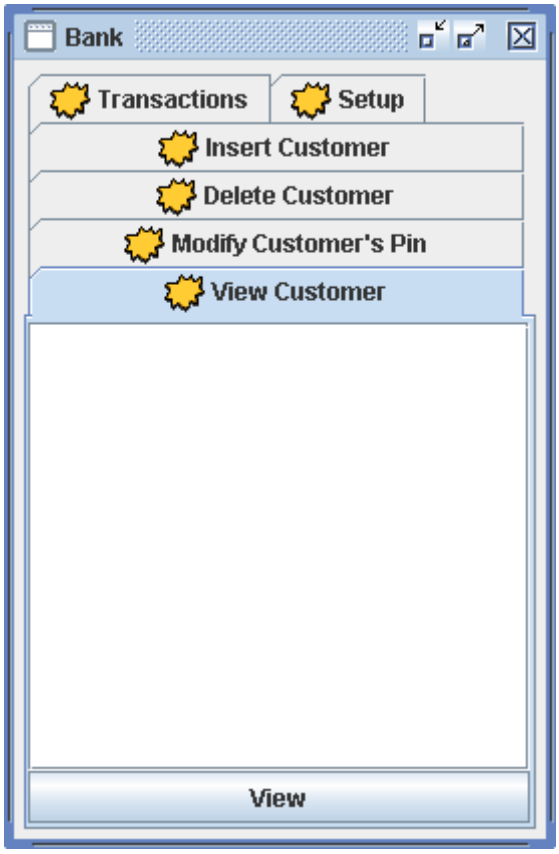
Στο Tab **Insert Customer** ο χρήστης έχει τη δυνατότητα να κάνει εγγραφή νέου πελάτη στην βάση δεδομένων της τράπεζας αρκεί να γράψει ένα επώνυμο ,ένα όνομα, ένα **μοναδικό** Account Number (5ψηφιο) και ένα PIN (4ψηφιο).Στη συνέχεια πατώντας Insert η εγγραφή καταχωρείται στην βάση δεδομένων και είναι έτοιμη για εκτέλεση δοσοληψιών.



Στο Tab **Delete Customer** ο χρήστης απλά γράφοντας το Account Number ενός υπαρκτού πελάτη της βάσης δεδομένων και πατώντας το Delete διαγράφει την εγγραφή του πελάτη αυτού από την βάση δεδομένων.

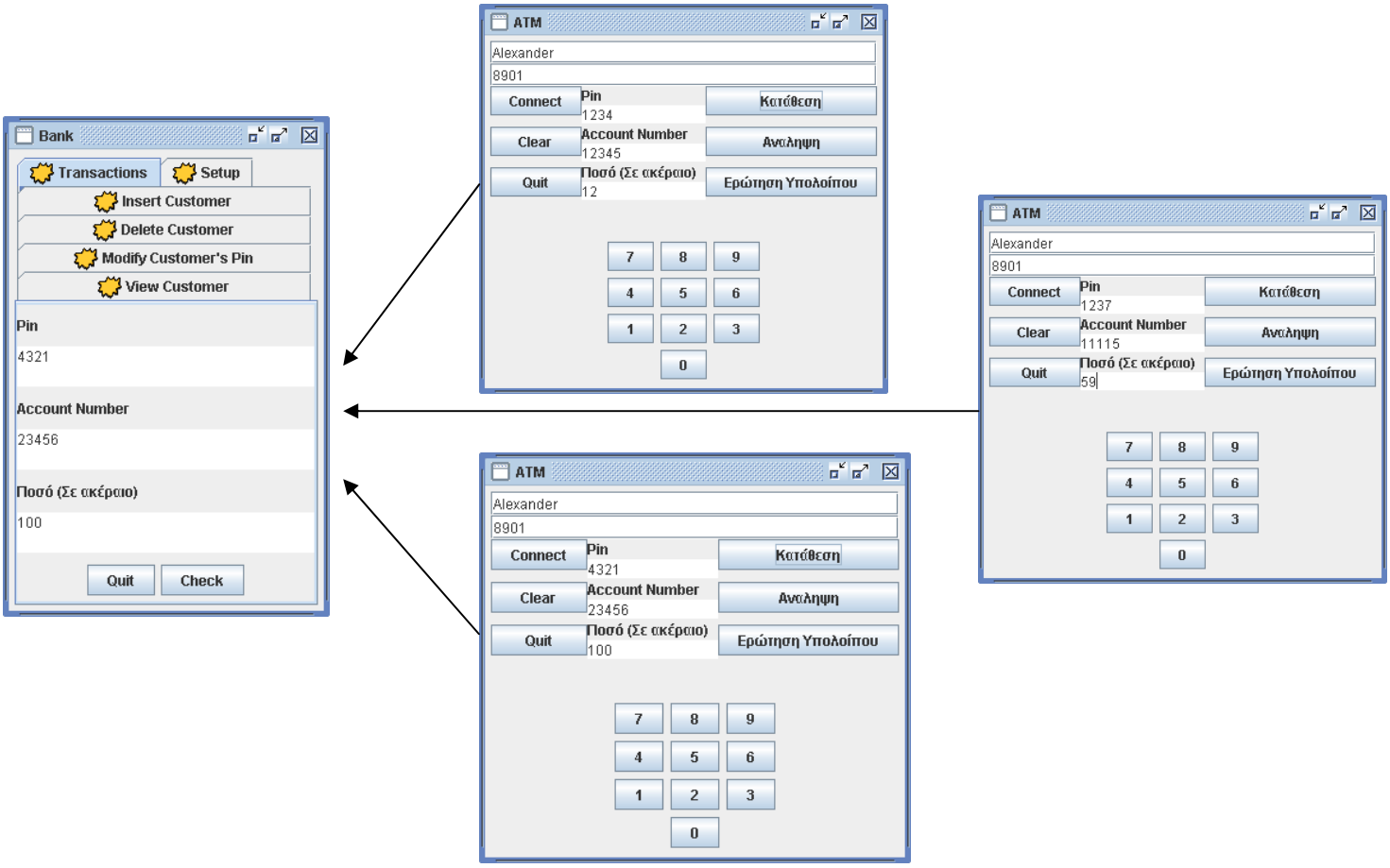


Στο Tab **Modify Customer's Pin** μπορούμε να κάνουμε τροποποίηση του Pin που χρησιμοποιεί ο πελάτης με κάποιο νέο, αρκεί να δώσουμε τα στοιχεία που μας ζητούνται και να πατήσουμε το κουμπί Change Pin.

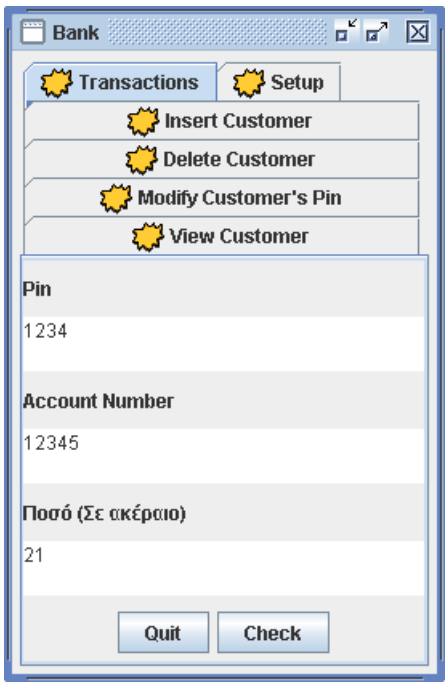


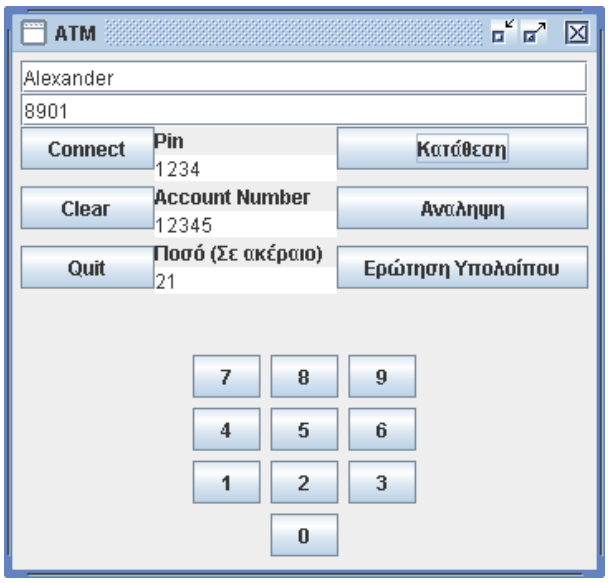
Τέλος στο Tab **View Customer** πατώντας το κουμπί View μας δίνεται η δυνατότητα να δούμε τα περιεχόμενα της βάσης δεδομένων με τα απαραίτητα στοιχεία κάθε πελάτη έτσι ώστε να εκτελέσουμε κάποια εντολή.

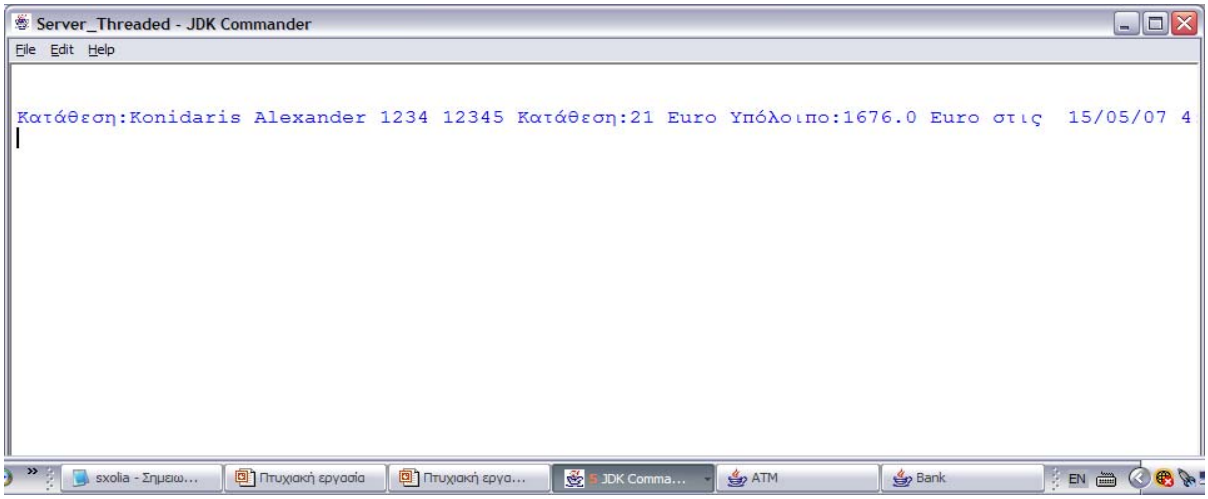
Multi Threaded



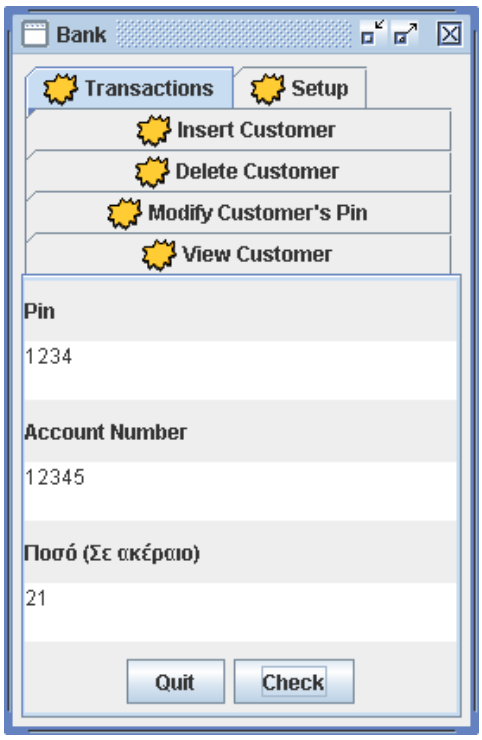
Κατάθεση

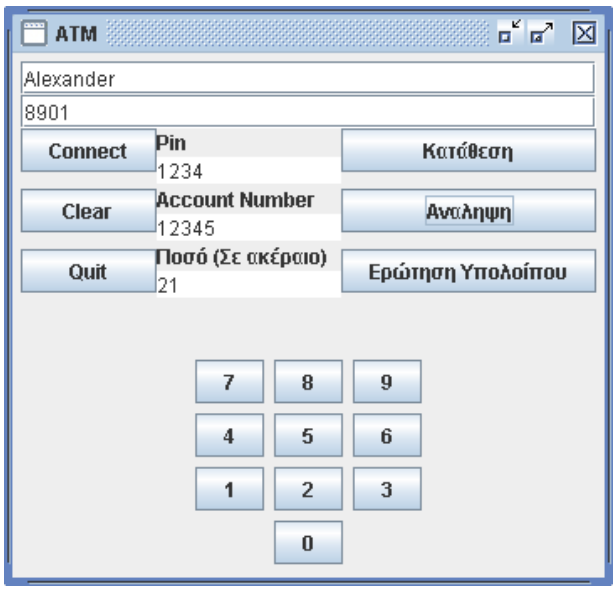

Bank
Transactions
Setup
Insert Customer
Delete Customer
Modify Customer's Pin
View Customer
Pin
1234
Account Number
12345
Ποσό (Σε ακέραιο)
21
Quit
Check

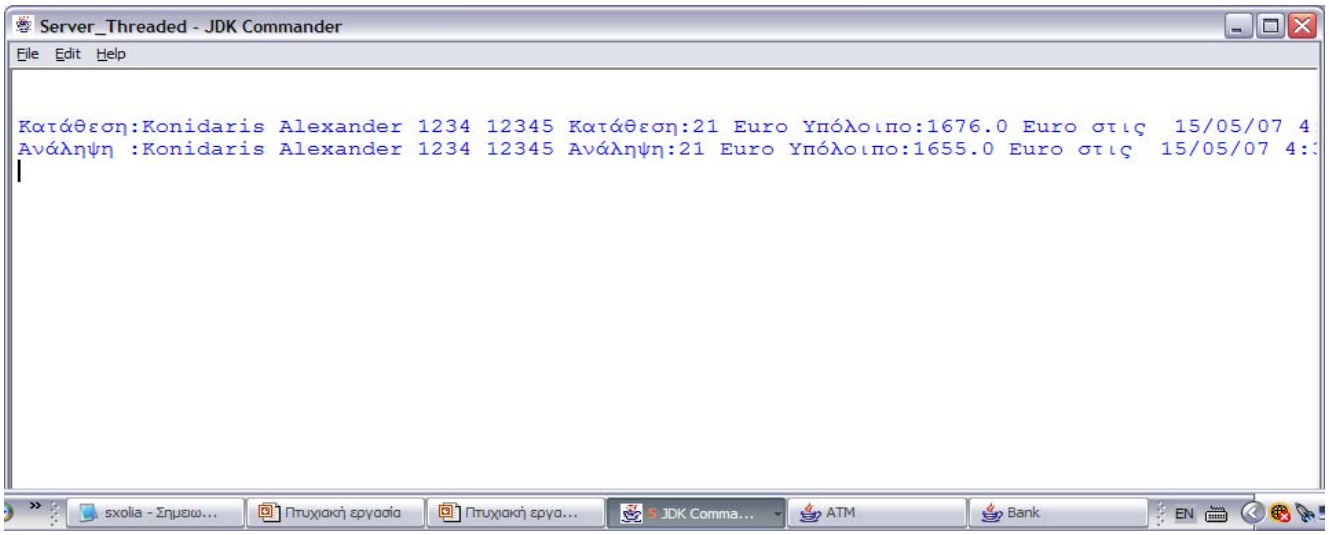

ATM
Alexander
8901
Connect
Pin
1234
Κατάθεση
Clear
Account Number
12345
Ανάληψη
Quit
Ποσό (Σε ακέραιο)
21
Ερώτηση Υπολοίπου
7
8
9
4
5
6
1
2
3
0


Server_Threaded - JDK Commander
File Edit Help
Κατάθεση:Konidaris Alexander 1234 12345 Κατάθεση:21 Euro Υπόλοιπο:1676.0 Euro στις 15/05/07 4
sxolia - Σημεω...
Πτυχιακή εργασία
Πτυχιακή εργα...
JDK Comma...
ATM
Bank
EN

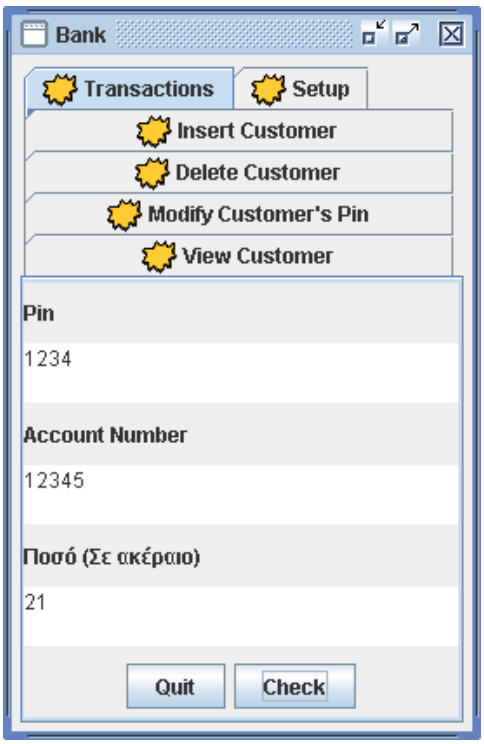
Ανάληψη

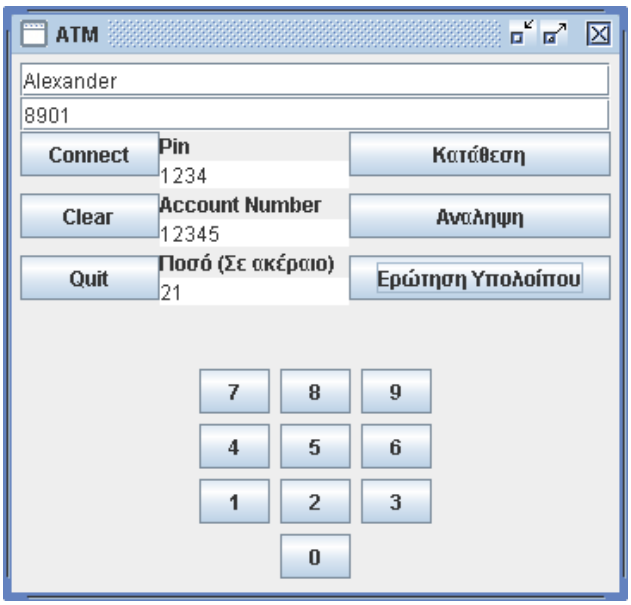

Bank
Transactions
Setup
Insert Customer
Delete Customer
Modify Customer's Pin
View Customer
Pin
1234
Account Number
12345
Ποσό (Σε ακέραιο)
21
Quit
Check

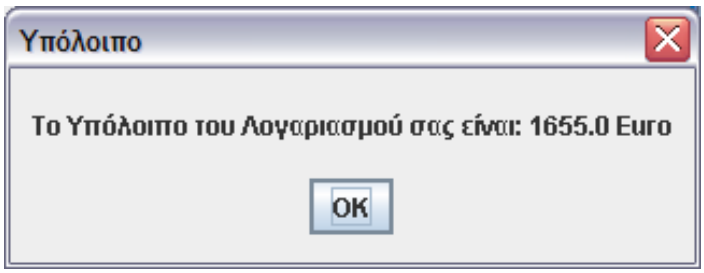

ATM
Alexander
8901
Connect
Pin
1234
Κατάθεση
Clear
Account Number
12345
Ανάληψη
Quit
Ποσό (Σε ακέραιο)
21
Ερώτηση Υπολοίπου
7
8
9
4
5
6
1
2
3
0


Server_Threaded - JDK Commander
File Edit Help
Κατάθεση:Konidaris Alexander 1234 12345 Κατάθεση:21 Euro Υπόλοιπο:1676.0 Euro στις 15/05/07 4
Ανάληψη :Konidaris Alexander 1234 12345 Ανάληψη:21 Euro Υπόλοιπο:1655.0 Euro στις 15/05/07 4:
sxolia - Σημειω...
Πτυχιακή εργασία
Πτυχιακή εργα...
JDK Comma...
ATM
Bank
EN

Ερώτηση Υπολοίπου







Αντοχή

Παρακάτω παρουσιάζονται οι επιθέσεις από τις οποίες προστατεύεται το σύστημα και οι συγκεκριμένοι μηχανισμοί με τους οποίους το επιτυγχάνει.

Read content of messages

Τα μηνύματα κρυπτογραφούνται από τον αποστολέα με το Public Key του παραλήπτη. Αν ο παραλήπτης είναι αυτός που πράγματι πρέπει να διαβάσει το μήνυμα τότε θα έχει το αντίστοιχο Private Key και θα μπορέσει να το αποκρυπτογραφήσει και να το διαβάσει. Το Private Key είναι γνωστό μόνο στον παραλήπτη. Επομένως κανείς άλλος δεν θα είναι σε θέση να διαβάσει το μήνυμα.

Read the contents of stored data

Τα μηνύματα κρυπτογραφούνται με το Public Key του Server επομένως μόνο αυτός είναι σε θέση να τα αποκρυπτογραφήσει και να τα διαβάσει. Με αυτό τον τρόπο, ικανοποιείται η απαίτηση της εκφώνησης για προστασία της ανάγνωσης από μη-εξουσιοδοτημένους χρήστες.

Modify content of messages

Ο αποστολέας υπογράφει ηλεκτρονικά τα μηνύματά του. Έτσι ο παραλήπτης έχει την δυνατότητα να ελέγξει την ακεραιότητα τους. Αν κάποιος παρεμβληθεί και αλλάξει τα περιεχόμενα του μηνύματος τότε η υπογραφή στον παραλήπτη δεν θα ταιριάζει με το τροποποιημένο μήνυμα οπότε το μήνυμα θα απορριφθεί.

Πως θα εκτελέσουμε ένα πρόγραμμα που είναι γραμμένο σε Java

Πρώτα απ' όλα θα πρέπει να έχουμε εγκατεστημένη κάποια έκδοση από κάποιο jdk (π.χ. jdk1.5.0_06). Σε περίπτωση που δεν έχουμε κάποιο θα το κάνουμε download από το site της sun (www.java.sun.com/j2se/1.5.0/download.html).

Στην συνέχεια κάνουμε εγκατάσταση του αρχείου που έχουμε κατεβάσει. Αφού τελειώσει η εγκατάσταση πλέον ο υπολογιστής έχει Java machine για να τρέξει και να εκτελέσει τα προγράμματα .java.

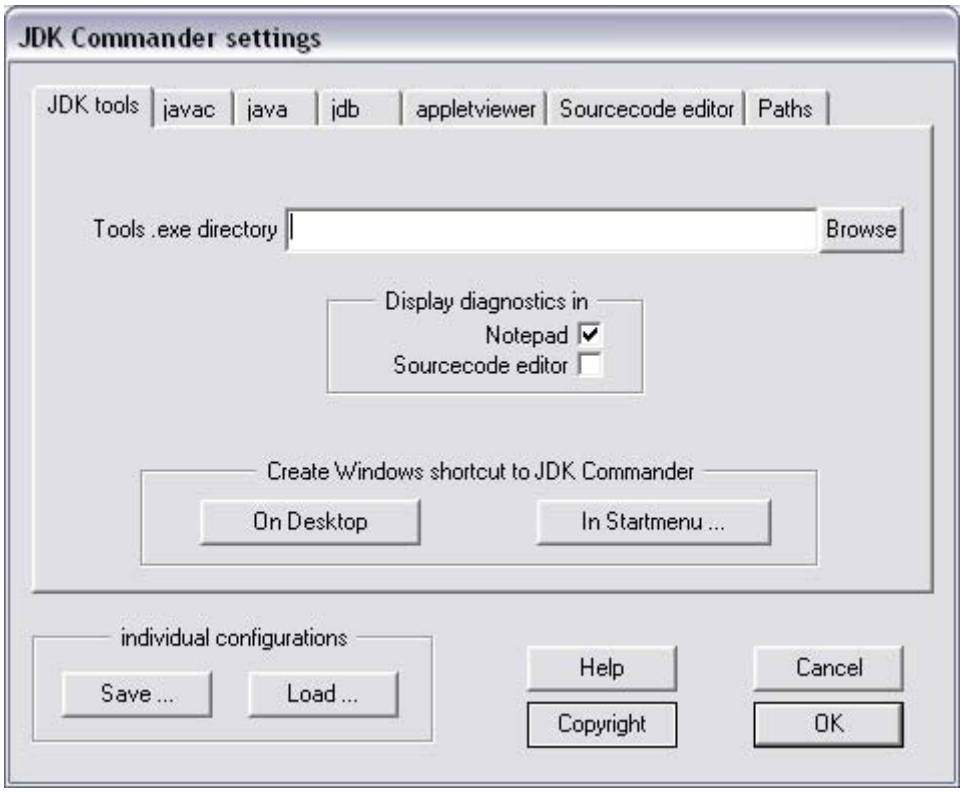
JDKcommander

Το JDK Commander είναι ένα πρόγραμμα το οποίο μας βοηθάει πάρα πολύ στο compile ενός προγράμματος Java καθώς και στο τρέξιμο του προγράμματος αυτού. Γλιτώνουμε έτσι από το ξόδεμα περιττού χρόνου για το compile και το τρέξιμο της εφαρμογής με τη χρήση του command prompt.

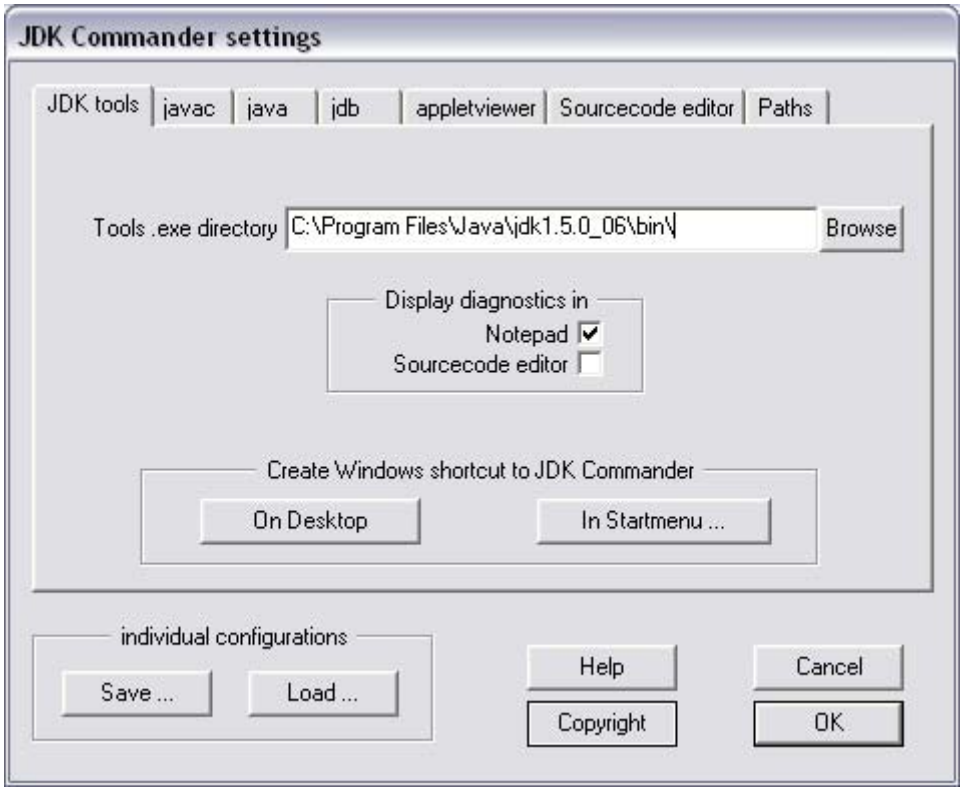
Το πρόγραμμα αυτό μπορούμε να το κάνουμε download από το
www.geocities.com/jdkcommander/
www.homepage.sunrise.ch/mysunrise/jdkcommander

Δεν χρειάζεται κάποιου είδους εγκατάσταση, απλά πατάμε το εκτελέσιμο αρχείο JDKcommander.exe

Την πρώτη φορά που θα το τρέξουμε θα μας εμφανιστεί η εξής οθόνη.



Πατώντας το κουμπί Browse ψάχνουμε την διαδρομή που έχουμε εγκαταστήσει την Java (στην ουσία ψάχνουμε για το αρχείο javac.exe).Αν αφήσουμε την default διαδρομή εγκατάστασης του jdk το αρχείο αυτό θα βρίσκεται στην διαδρομή
C:\Program Files\Java\jdk1.5.0_06\bin\



Εφόσον κάνουμε αυτή την διαδικασία πατάμε OK

(Αυτή η διαδικασία για τον εντοπισμό του javac.exe γίνεται μια μόνο φορά για τον προσδιορισμό του compiler της Java)

Μετά το OK μας εμφανίζεται η οθόνη του JDK Commander η οποία είναι η εξής.

Στο πρώτο πεδίο με το κουμπί Browse βρίσκουμε την διαδρομή στην οποία έχουμε αποθηκεύσει το project μας. Στο πεδίο αυτό κάνουμε compile τα αρχεία που τελειώνουν σε .java (για την συγκεκριμένη εργασία είναι το Client_Threaded.java και Server_Threaded.java, τα υπόλοιπα γίνονται compile έμμεσα από τα άλλα δυο). Επιλέγουμε πρώτα το Server_Threaded.java και μετά πατάμε το κουμπί javac. Στη συνέχεια πατάμε ξανά το Browse και επιλέγουμε το Client_Threaded.java και πατάμε το κουμπί javac

Εφόσον έχουμε κάνει compile τα απαραίτητα προγράμματα και μας βεβαιώσει ότι δεν υπάρχουν σφάλματα στο πρόγραμμα μας πατάμε το Browse στο δεύτερο πεδίο και επιλέγουμε ποιο θέλουμε να εκτελεστεί (στην συγκεκριμένη περίπτωση εκτελούμε πρώτο το Server_Threaded.class και στη συνέχεια το Client_Threaded.class).Σε περίπτωση που θέλουμε να σηκώσουμε παραπάνω από έναν Client πατάμε java ξανά στο Client_Threaded.class.

<u>Πως μπορούμε να δούμε τον κώδικα</u>

Μπορούμε να χρησιμοποιήσουμε οποιονδήποτε κειμενογράφο θέλουμε όπως είναι το (Notepad, Ultra Edit, Notepad++ κ.λ.π.).

Σύνδεση Βάσεων Δεδομένων με την Java

Πριν από μερικά χρόνια για να αντιμετωπίσει η Microsoft το πρόβλημα της επικοινωνίας μεταξύ των διαφορετικών τύπων βάσεων δεδομένων, ανέπτυξε το πρότυπο ODBC (Open Data Base Connectivity), ένα πρότυπο για την προσπέλαση βάσεων δεδομένων. Το ODBC καθορίζει ένα συγκεκριμένο τύπο συμπεριφοράς των εφαρμογών στην περίπτωση που αυτές καλούν βάσεις δεδομένων, και έναν άλλον όταν μια συγκεκριμένη βάση δεδομένων ανταποκρίνεται σε κλήσεις τύπου ODBC από μια εφαρμογή.

Η εταιρεία Sun συμπεριέλαβε στην Java τη διασύνδεση JDBC (Java Data Base Connectivity) η οποία καθορίζει τον τρόπο επικοινωνίας μιας εφαρμογής με σχεσιακές βάσεις δεδομένων. Η γλώσσα με αυτό τον τρόπο κατορθώνει να υποστηρίζει τις σχεσιακές βάσεις δεδομένων που δημιουργούνται από διαφορετικά DBMS, όπως κατορθώνει να υποστηρίζει και τα διαφορετικά λειτουργικά συστήματα.

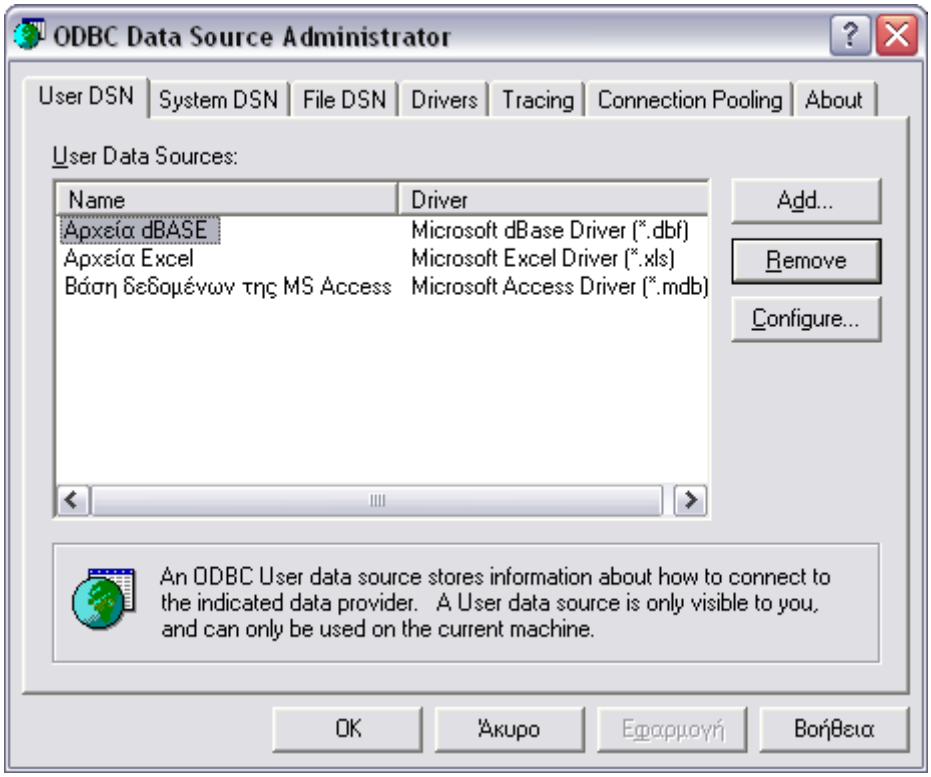
Ο βασικός στόχος της βιβλιοθήκης JDBC είναι να επιτρέπει στα προγράμματα της γλώσσας Java να εκτελούν εντολές SQL και να λαμβάνουν τα αποτελέσματα των εντολών αυτών από τη βάση δεδομένων. Όλες οι βάσεις δεδομένων που είναι συμβατές με το ODBC της Microsoft είναι συμβατές και με το JDBC. Κατ' αυτόν τον τρόπο μια εφαρμογή Java αποδεσμεύεται από τη βάση δεδομένων κάθε συγκεκριμένου κατασκευαστή.

Γέφυρα JDBC – ODBC: Η γέφυρα επιτρέπει σε μια εφαρμογή συμβατή με JDBC να επικοινωνεί με μια βάση δεδομένων συμβατή με το ODBC. Η γέφυρα μεταφέρει εντολές και αποτελέσματα μέσω του οδηγού ODBC.

«Γεφύρωση» στην πράξη

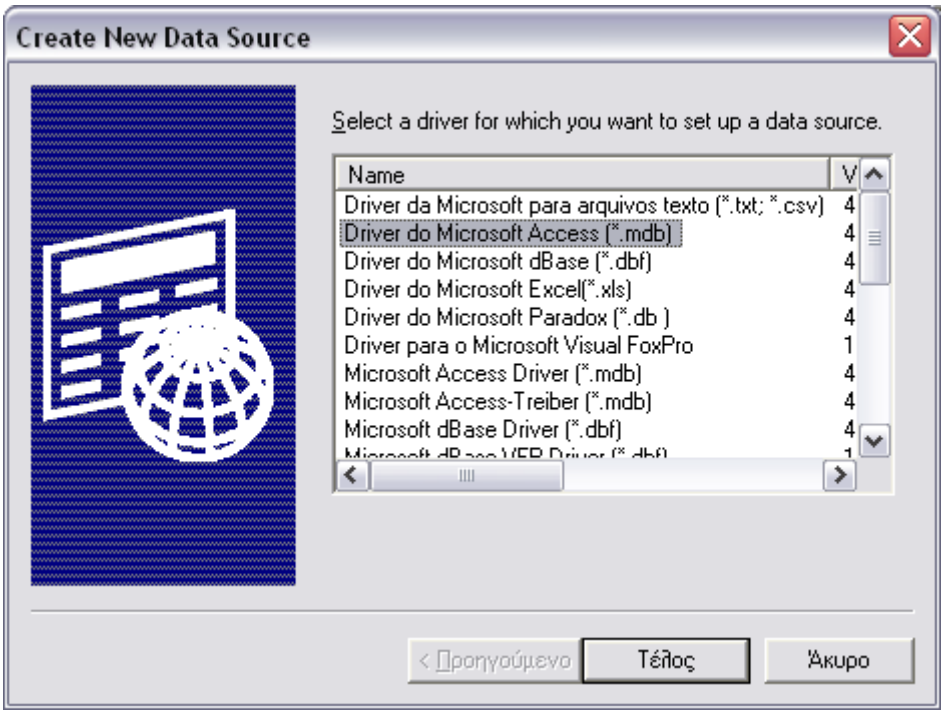
Πρώτα απ' όλα θα πρέπει να δημιουργήσουμε μια βάση δεδομένων (για την συγκεκριμένη πτυχιακή εργασία έχει δημιουργηθεί η db1.mdb η οποία έχει έναν πίνακα customer με τους πελάτες της τράπεζας και τα απαραίτητα στοιχεία τους).

Στην συνέχεια πηγαίνουμε Έναρξη - Πίνακας Ελέγχου - Εργαλεία διαχείρισης - Πηγές δεδομένων (ODBC) και πατάμε διπλό κλικ και μας εμφανίζεται η παρακάτω οθόνη.



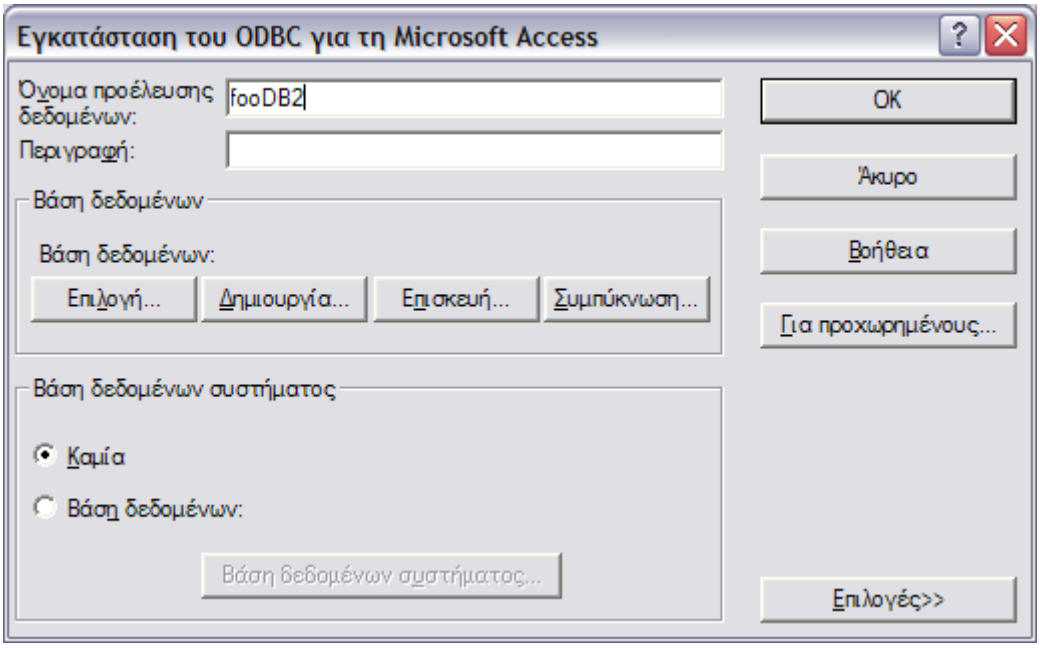
(Για άλλα λειτουργικά μπορεί να είναι: control panel - administrative tools - data base sources)

Πατάμε το κουμπί Add και μας εμφανίζεται η εξής οθόνη



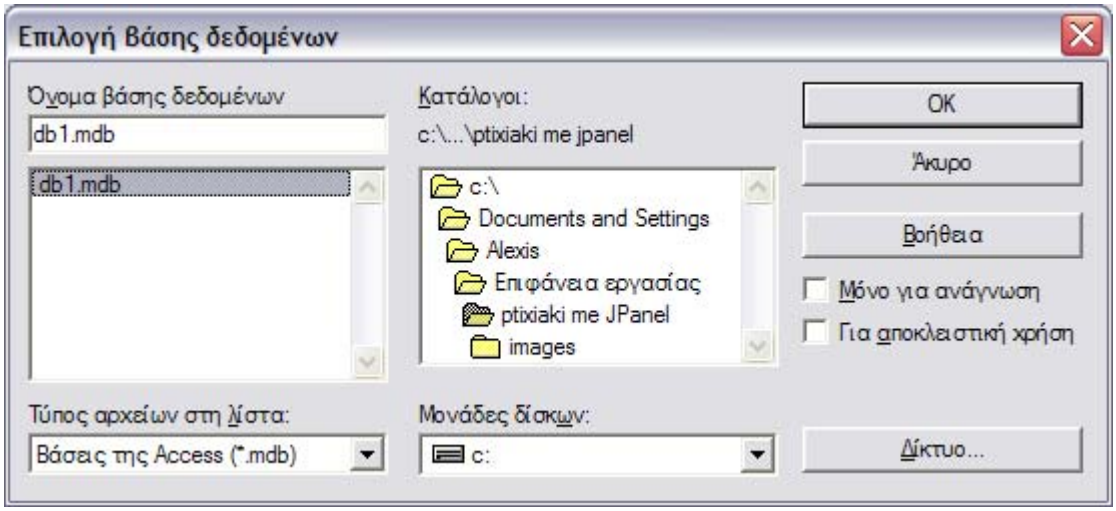
Επιλέγουμε το Driver do Microsoft Access (*.mdb) και κάνουμε διπλό κλικ

Στην επόμενη οθόνη που θα μας εμφανιστεί γράφουμε στο πεδίο όνομα προέλευσης δεδομένων fooDB2 (όνομα το οποίο δηλώνουμε και στο πρόγραμμα της Java στην κλάση DataBase).



και στην συνέχεια πατάμε το κουμπί που λέει «Επιλογή»

Στο παράθυρο που θα μας εμφανιστεί ψάχνουμε για την διαδρομή που είναι αποθηκευμένη η βάση δεδομένων μας .



και πατάμε OK σε όσα παράθυρα μας είναι εμφανή.

Η γεφύρωση έχει γίνει και πλέον το πρόγραμμα είναι έτοιμο να εκτελέσει τις εντολές και να διαβάσει τα δεδομένα από την βάση δεδομένων

Αναφορές

- Bruce Schneier, Applied Cryptography, 2nd edition, Wiley, 758, 1996 δείγματα του βιβλίου.
- Το κρυπτόγραμμα του Bruce Schneier στα ελληνικά.
- J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, 780, 1996.
- Douglas R. Stinson, Cryptography: Theory and Practice (Discrete Mathematics and Its Applications), 1st edition, CRC Press, 434, 1995.
- Wenbo Mao, Modern Cryptography: Theory and Practice, 1st edition, Prentice Hall PTR, 740, 2003.
- William Stallings, Cryptography and Network Security: Principles and Practice, 2nd Edition, Prentice Hall, 569, 1998.
- Henk C.A. van Tilborg, Fundamentals of Cryptology : A Professional Reference and Interactive Tutorial, 1 edition, Springer, 512, 1999.
- David Kahn, The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet, Scribner, 1200, 1996.
- Simon Singh, Κώδικες και Μυστικά, Τραυλός, 606, 2001, ISBN 960-7990-42-0
- Β.Α. Κάτος - Γ.Χ. Στεφανίδης, Τεχνικές Κρυπτογραφίας & Κρυπτανάλυσης, ΖΥΓΟΣ, 396, 2003.

Site τα οποία χρησιμοποιήθηκαν σαν πηγή πληροφόρησης

http://theory.csail.mit.edu/~yiqun/shanote.pdf
http://www.schneier.com/essay-074.html
http://csrc.nist.gov/CryptoToolkit/tkhash.html
http://www.wikipedia.com
http://www.google.com
http://www.yahoo.com
http://java.sun.com

Βιβλία

- Γιώργος Λιακέας - Εισαγωγή στην JAVA 2 (Ένας ολοκληρωμένος και εύχρηστος οδηγός της γλώσσας) ΕΚΔΟΣΕΙΣ ΚΛΕΙΔΑΡΙΘΜΟΣ

Tutorials

http:// www.BruceEckel.com (Thinking in Java, 3rd ed. Revision 4.0)
http://java.sun.com/docs/books/tutorial (The Really Big Index)
http://www.ora.com/catalog/books/javanut2/ (Java Reference Library O'Relly)