ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

ΟΙ ΜΗ ΣΥΜΜΕΤΡΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΚΡΥΠΤΟΓΡΑΦΗΣΗΣ RSA και DSA/ElGamal

Συγκριτική μελέτη - πλεονεκτήματα και μειονεκτήματα κατά την εφαρμογή τους στην ασφάλεια πληροφορικών συστημάτων

Πτυχιακή εργασία του: Παράσχου-Χαράλαμπου Σ. Μαυρογιάννη

Επιβλέπων: Νικόλαος Κατσάκος-Μαυρομιχάλης

Σπάρτη, Μάιος 2017

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Π.Χ. Μαυρογιάννης
6 Ιουνίου 2017

Αφιερώνεται στους γονείς μου

Περίληψη

Στην ασφάλεια πληροφοριακών συστημάτων, στα θέματα κυρίως αυθεντικοποίησης χρηστών και ψηφιακής υπογραφής δύο είναι οι μη συμμετρικοί αλγόριθμοι που κυριαρχούν σήμερα. Ο αλγόριθμος RSA και ο αλγόριθμος DSA. Ο δεύτερος αναπτύχθηκε από την κυβέρνηση των ΗΠΑ με σκοπό να χρησιμοποιηθεί αποκλειστικά για την ψηφιακή υπογραφή κειμένων και βασίστηκε πάνω στον αλγόριθμο ElGamal. Καίτοι οι δύο αλγόριθμοι χρησιμοποιούνται ευρέως σε διάφορες εφαρμογές ασφάλειας έχουν σημαντικές διαφορές μεταξύ τους έτσι ώστε σε κάθε περίπτωση να αναζητείται η βέλτιστη εφαρμογή των τεχνικών RSA ή/και DSA με στόχο την ελαχιστοποίηση χρήσης των ελεύθερων πόρων του πληροφοριακού συστήματος.

Στη παρούσα εργασία μελετώνται τα υπέρ και τα κατά της υλοποίησης των δυο μη συμμετρικών αλγορίθμων κατά την εφαρμογή τους στην ασφάλεια πληροφοριακών συστημάτων. Συγκεκριμένα μελετώνται οι ταχύτητες απόκρισης τους για την δημιουργία ζεύγους κλειδιών, παραγωγής ψηφιακών υπογραφών και επαλήθευσης ψηφιακών υπογραφών. Μελετάται και η περίπτωση απλής κρυπτογράφησης – αποκρυπτογράφησης μικρών μηνυμάτων με τον αλγόριθμο RSA. Παράγονται συγκεκριμένα παραδείγματα που υλοποιούνται με την γλώσσα προγραμματισμού Java.

Λέξεις Κλειδιά

Μη συμμετρικοί αλγόριθμοι, κρυπτογραφία, ασφάλεια πληροφοριακών συστημάτων, ψηφιακή υπογραφή, RSA, DSA

Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον καθηγητή κ. Νικόλαο Κατσάκο-Μαυρομιχάλη για την επίβλεψη αυτής της πτυχιακής εργασίας. Επίσης ευχαριστώ όλους τους καθηγητές του ΤΕΙ Πελοποννήσου για την πολύτιμη καθοδήγηση τους όλα αυτά τα χρόνια και την εξαιρετική συνεργασία που είχαμε.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου γιατί χωρίς την αμέριστη συμπαράσταση και υπομονή τους αυτή η εργασία –και πολλές άλλες- δεν θα είχε ποτέ ολοκληρωθεί.

Σπάρτη, Μάιος 2017

Παράσχος Μαυρογιάννης

Περιεχόμενα

1 Συμμετρικοί και μη συμμετρικοί αλγόριθμοι

1.1 Βασικές έννοιες της κρυπτογραφίας

1.1.1 Εισαγωγή

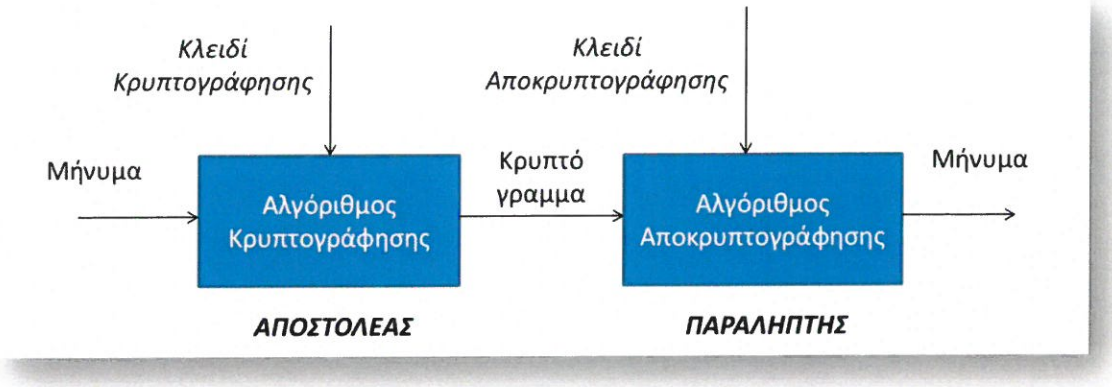
Ο στόχος ενός κρυπτογραφικού συστήματος (cipher system) είναι να μετατρέψει τις εμπιστευτικές πληροφορίες σε μια μορφή η οποία δεν θα παρέχει κανένα νόημα (συμπέρασμα) σε κάποιο μη εξουσιοδοτημένο πρόσωπο. Δύο προφανείς χρήσεις είναι η ασφαλής αποθήκευση δεδομένων σε έναν υπολογιστή και η ασφαλής μετάδοση μέσα από ένα ανασφαλές κανάλι σαν το Internet. Και στα δύο σενάρια το γεγονός πως οι πληροφορίες είναι κρυπτογραφημένες δεν απαγορεύει σε κάποιον να έχει πρόσβαση σε αυτές — διασφαλίζει, όμως, πως δεν μπορεί να καταλάβει το περιεχόμενό τους.

1.1.2 Βασικοί ορισμοί

Η πληροφορία που πρέπει να προστατευτεί ονομάζεται «καθαρή» (ή αρχική - plaintext) και όλες οι λειτουργίες μετατροπής περιγράφονται από την έννοια της κρυπτογράφησης (encryption). Το κρυπτογραφημένο κείμενο ονομάζεται ciphertext ή cryptogram (κρυπτοκείμενο ή κρυπτόγραμμα), ενώ το σύνολο των κανόνων που χρησιμοποιούνται για να κρυπτογραφήσουν το plaintext ονομάζεται αλγόριθμος κρυπτογράφησης (encryption algorithm). Η λειτουργία του αλγορίθμου κρυπτογράφησης βασίζεται σε ένα κλειδί κρυπτογράφησης (encryption key) το οποίο εισάγεται στον αλγόριθμο μαζί με το μήνυμα.

Για να μπορέσει ο παραλήπτης να πάρει το μήνυμα χρειάζεται ένας αντίστοιχος αλγόριθμος αποκρυπτογράφησης (decryption algorithm) που όταν χρησιμοποιηθεί μαζί με το αντίστοιχο κλειδί αποκρυπτογράφησης (decryption key) αναπαράγει το plaintext από το ciphertext. Γενικά, οι κανόνες που αποτελούν έναν τέτοιο αλγόριθμο είναι αρκετά σύνθετοι και χρειάζεται μεγάλη προσοχή στη σχεδίαση και την επιλογή τους.

Εκείνος που υποκλέπτει κάποιο μήνυμα κατά τη διάρκεια της διάδοσής του ονομάζεται υποκλοπέας (interceptor) ή απλούστερα επιτιθέμενος (attacker). Συνήθως χρησιμοποιούνται και άλλοι γενικότεροι όροι όπως «εχθρός», «αντίπαλος» ακόμη και «ο κακός».



Σχήμα 1.1

Το προηγούμενο σύστημα λειτουργεί γιατί ακόμη και στην περίπτωση που κάποιος από τους παραπάνω γνωρίζει τον αλγόριθμο κρυπτογράφησης δεν γνωρίζει το κλειδί αποκρυπτογράφησης. Αυτή ακριβώς η έλλειψη γνώσης (του κλειδιού αποκρυπτογράφησης) τον εμποδίζει να ξέρει την αρχική πληροφορία (plaitext).

Η κρυπτογραφία (cryptography) είναι η επιστήμη που ασχολείται με τη σχεδίαση τέτοιων συστημάτων (cipher systems) ενώ η κρυπτανάλυση (cryptanalysis) είναι η διαδικασία με την οποία κάποιος μπορεί να εξαγάγει την αρχική πληροφορία από το κρυπτόγραμμα χωρίς να του δοθεί το κατάλληλο κλειδί. Ο όρος κρυπτολογία (cryptology) συμπεριλαμβάνει και τις δύο παραπάνω έννοιες.

Σε αυτό το σημείο είναι σημαντικό να κατανοήσουμε πως η κρυπτανάλυση δεν είναι ο μοναδικός τρόπος με τον οποίον κάποιος μπορεί να έχει πρόσβαση στην αρχική πληροφορία. Ας υποθέσουμε, για παράδειγμα, πως κάποιος αποθηκεύει κρυπτογραφημένα δεδομένα στον υπολογιστή του. Ο χρήστης αυτός με κάποιον τρόπο πρέπει να αποθηκεύσει και το αντίστοιχο κλειδί αποκρυπτογράφησης που θα του επιτρέπει να χρησιμοποιεί τα δεδομένα του. Αν π.χ. το γράψει σε ένα κομμάτι χαρτί και το κολλήσει στην οθόνη του (αρκετά συνηθισμένη περίπτωση), τότε όποιος μπορεί να διαβάσει αυτό το χαρτί (πράγμα όχι και τόσο δύσκολο) μπορεί να έχει πρόσβαση στις πληροφορίες του.

Το παραπάνω -αρκετά απλουστευμένο- παράδειγμα δείχνει πως χρειάζονται παραπάνω πράγματα από έναν καλοσχεδιασμένο αλγόριθμο κρυπτογράφησης για να μπορέσουμε να αποθηκεύσουμε

δεδομένα με ασφάλεια. Δίνεται μεγάλη έμφαση στην σημασία της προστασίας των κρυπτογραφικών κλειδιών, η οποία και αποτελεί ένα από τα πιο σημαντικά θέματα ασφάλειας των κρυπτογραφικών συστημάτων.

Στην πράξη οι περισσότερες κρυπταναλυτικές προσπάθειες στοχεύουν στην ανακάλυψη του κλειδιού αποκρυπτογράφησης. Αν μια τέτοια προσπάθεια είναι επιτυχημένη, τότε ο επιτιθέμενος (attacker) έχει την ίδια γνώση με τον υποτιθέμενο παραλήπτη και μπορεί να αποκρυπτογραφεί κάθε επικοινωνία μέχρις ότου αλλαχθούν τα κλειδιά. Υπάρχουν όμως και αρκετές περιπτώσεις που ο μόνος σκοπός του επιτιθέμενου είναι να διαβάσει ένα μόνο συγκεκριμένο μήνυμα. Γενικά, όμως, όταν λέμε ότι ένας αλγόριθμος «σπάει» σημαίνει ότι ο επιτιθέμενος έχει καταφέρει με κάποιον τρόπο να βρει το κλειδί αποκρυπτογράφησης.

Φυσικά ο επιτιθέμενος είναι σε θέση να σπάσει κάποιον αλγόριθμο μόνον εάν έχει αρκετές πληροφορίες που του επιτρέπουν να αναγνωρίζει το σωστό κλειδί ή, ακόμη πιο σωστά, να αναγνωρίζει τα λαθεμένα κλειδιά. Για παράδειγμα, εάν ο επιτιθέμενος υποκλέπτει κάποιο μήνυμα που γνωρίζει ότι είναι γραμμένο στην αγγλική γλώσσα, κάνει διαδοχικές προσπάθειες να αποκρυπτογραφήσει το μήνυμα χρησιμοποιώντας διαφορετικά κλειδιά. Μόνον όταν το μήνυμα που αποκρυπτογραφεί έχει κάποιο νόημα στην αγγλική γλώσσα μπορεί να είναι σίγουρος ότι το κλειδί που χρησιμοποιεί είναι το σωστό (ή, καλύτερα, πως όλα τα προηγούμενα κλειδιά ήταν λάθος).

Επίσης, θα πρέπει να τονιστεί πως η γνώση του κλειδιού κρυπτογράφησης δεν είναι πάντοτε απαραίτητη όταν θέλουμε να εξάγουμε το αρχικό κείμενο από ένα κρυπτόγραμμα. Αυτή η διαπίστωση έχει δραματικές επιπτώσεις στη μοντέρνα κρυπτολογία και έχει οδηγήσει στη φυσική διαίρεση των κρυπτογραφικών συστημάτων σε συμμετρικά (symmetric) και ασύμμετρα (asymmetric).

1.1.3 Συμμετρικά και ασύμμετρα κρυπτοσυστήματα

Ένα κρυπτογραφικό σύστημα ονομάζεται συμβατικό (conventional) ή συμμετρικό (symmetric) όταν είναι εύκολο να βρεθεί το κλειδί αποκρυπτογράφησης από το αντίστοιχο κλειδί κρυπτογράφησης. Πρακτικά, στα συμμετρικά συστήματα τα δύο αυτά

κλειδιά είναι ταυτόσημα. Επιπλέον, υπάρχουν περιπτώσεις που είναι πρακτικά αδύνατο να βρεθεί το κλειδί αποκρυπτογράφησης εάν γνωρίζουμε το αντίστοιχο κλειδί κρυπτογράφησης. Σε αυτές τις περιπτώσεις το σύστημα ονομάζεται ασύμμετρο (asymmetric) ή δημοσίου κλειδιού (public key).

Παραπάνω τονίσαμε την ανάγκη προστασίας του κλειδιού κρυπτογράφησης, ώστε να μην μπορεί κάποιος ο οποίος έχει υποκλέψει κάποιο κρυπτόγραμμα και γνωρίζει τον αλγόριθμο κρυπτογράφησης να από-κρυπτογραφήσει το μήνυμα. Στην πράξη αυτό είναι απαραίτητο μόνο στα συμμετρικά κρυπτοσυστήματα (symmetric cryptosystems). Στα ασύμμετρα συστήματα η γνώση του κλειδιού κρυπτογράφησης δεν έχει καμία πρακτική αξία για τον επιτιθέμενο. Αντίθετα, το κλειδί αυτό μπορεί να είναι (και συνήθως αυτό είναι και το επιθυμητό) δημοσίως γνωστό και ανακοινώσιμο. Η πρώτη συνέπεια είναι πως δεν υπάρχει πλέον η ανάγκη ανάμεσα στον αποστολέα και τον παραλήπτη να μοιράζονται ένα κοινό μυστικό (shared secret), αλλά δεν υπάρχει πλέον και η ανάγκη για αμοιβαία εμπιστοσύνη μεταξύ τους.

Τα παραπάνω συμπεράσματα ίσως φαίνονται εμφανή — οι συνέπειές τους όμως φτάνουν πολύ μακριά. Το Σχήμα 1.1 υποθέτει πως τόσο ο αποστολέας όσο και ο παραλήπτης έχουν ένα κοινό κλειδί. Στην πράξη όμως αυτή η κατάσταση δεν είναι και τόσο εύκολη. Αν, για παράδειγμα, το σύστημα είναι συμμετρικό υπάρχει ανάγκη για την ασφαλή διανομή των κρυπτογραφικών κλειδιών πριν αρχίσει η ανταλλαγή μηνυμάτων. Τονίσαμε, προηγουμένως, την ανάγκη για προστασία των κλειδιών. Γενικότερα, η ασφαλής διαχείριση των κρυπτογραφικών κλειδιών (key management), που περιλαμβάνει τη δημιουργία, διανομή, χρήση, αποθήκευση, αλλαγή και καταστροφή είναι από τα πιο σημαντικά ζητήματα για τη δημιουργία ενός ασφαλούς κρυπτοσυστήματος. Τα προβλήματα που σχετίζονται με τη διαχείριση των κλειδιών είναι διαφορετικά στα συμμετρικά και τα ασύμμετρα συστήματα.

Εάν το σύστημα είναι συμμετρικό, όπως είδαμε και προηγουμένως, υπάρχει η ανάγκη για την ασφαλή διανομή των κλειδιών και στα δύο μέρη, καθώς και η ανάγκη η τιμή τους (value) να παραμένει μυστική. Εάν το σύστημα είναι ασύμμετρο μπορούμε να αποφύγουμε το συγκεκριμένο πρόβλημα με το να διανέμουμε μόνον τα κλειδιά κρυπτογράφησης, τα οποία και δεν χρειάζεται να παραμένουν μυστικά. Βέβαια, υπάρχουν καινούργια προβλήματα με πιο σημαντικό εκείνο της

εγγύησης αυθεντικότητας του κλειδιού κρυπτογράφησης του κάθε μέρους. Θα αναφερθούμε λεπτομερώς σε αυτό το πρόβλημα στο κεφάλαιο των ασύμμετρων κρυπτοσυστημάτων.

Παραπάνω υποθέσαμε, για την επεξήγηση των συμμετρικών και των ασύμμετρων κρυπτοσυστημάτων, πως ο επιτιθέμενος γνωρίζει τον αντίστοιχο αλγόριθμο. Για τον καλύτερο σχεδιασμό ενός κρυπτογραφικού συστήματος όμως, είναι καλύτερα να υποθέτουμε πως ο επίδοξος επιτιθέμενος έχει την περισσότερη δυνατή πληροφόρηση και γνώση γι' αυτό. Η βασική αρχή της κρυπτογραφίας υποθέτει πως η ασφάλεια ενός κρυπτογραφικού συστήματος δεν προϋποθέτει αντίστοιχα την μυστικότητα του κρυπτογραφικού αλγορίθμου. Αντιθέτως, η ασφάλεια ενός τέτοιου συστήματος εξαρτάται μόνον από την ασφάλεια του κλειδιού αποκρυπτογράφησης.

2 Υποδομή δημοσίου κλειδιού, δημιουργία ψηφιακής υπογραφής, επαλήθευση ψηφιακής υπογραφής

2.1 Πιστοποίηση ταυτότητας

Στον «πραγματικό» κόσμο υπάρχουν πολλές περιπτώσεις και συναλλαγές που βασίζονται στο «γνήσιο της υπογραφής» μας. Καθώς οι ψηφιακές υπογραφές τείνουν να χρησιμοποιούνται στον ηλεκτρονικό κόσμο προς αντικατάσταση των πραγματικών υπογραφών πρέπει με κάποιο τρόπο να εξακριβώνεται η γνησιότητά τους.

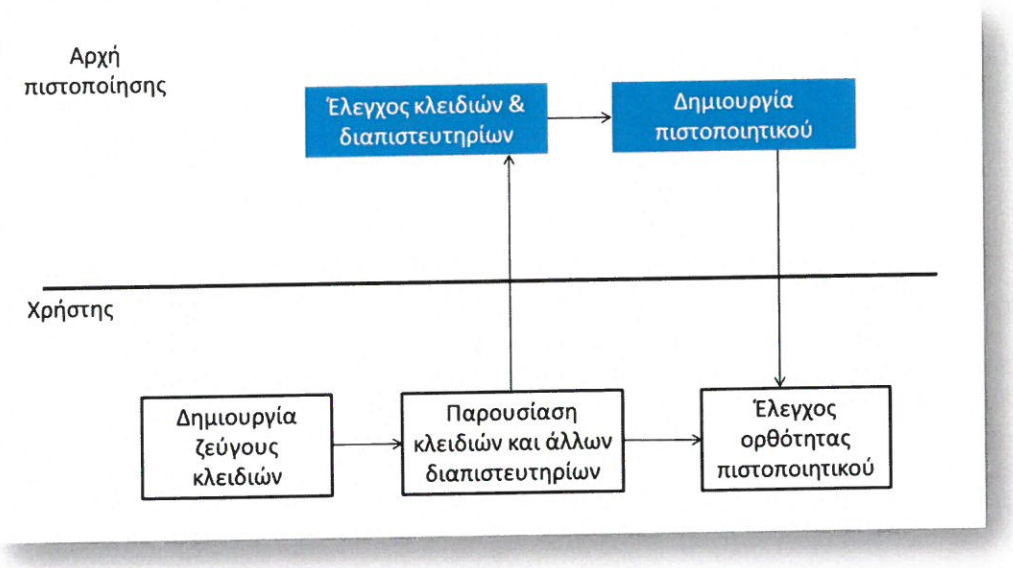
Για να δουλέψει αποτελεσματικά ένα σύστημα δημόσιου κλειδιού πρέπει να βεβαιωθούμε πως κάθε δημόσιο κλειδί είναι αυθεντικό, δηλ. κάθε δημόσιο κλειδί αντιστοιχεί σε ένα ιδιωτικό κλειδί το οποίο με τη σειρά του είναι άρρηκτα συνδεδεμένο με τον κάτοχό του.

Συνεχίζοντας τον παραλληλισμό μας, χρειάζεται μια κατάλληλη Αρχή Πιστοποίησης (Certification Authority) η οποία και θα βεβαιώνει τη γνησιότητα της υπογραφής μας. Τις περισσότερες φορές το ρόλο αυτό παίζουν οι αστυνομικές αρχές.

Στον ηλεκτρονικό κόσμο το ρόλο αυτό παίζει η λεγόμενη CA (Certification Authority — Αρχή Πιστοποίησης). Η CA μπορεί να είναι οποιαδήποτε οντότητα εμπνέει κάποια μορφή εμπιστοσύνης στους χρήστες ενός συστήματος δημόσιου κλειδιού και μπορεί να είναι μια εταιρεία, ένας κυβερνητικός οργανισμός, μια τράπεζα, ένα εκπαιδευτικό ίδρυμα κτλ. Κάθε χρήστης που ανήκει στην περιοχή αρμοδιότητας (domain) μιας CA, αφού περάσει τη διαδικασία εγγραφής, που φαίνεται στο Σχήμα 2.1, εφοδιάζεται με ένα ψηφιακό πιστοποιητικό (digital certificate ή πιο απλά certificate), το οποία περιέχει τα δημόσιο κλειδί ταυ χρήστη υπογεγραμμένο με το ιδιωτικό κλειδί της CA. Το πιστοποιητικό αυτό ουσιαστικά αποτελεί ένα ηλεκτρονικό έγγραφο που πιστοποιεί τη γνησιότητα του δημόσιου κλειδιού του χρήστη.

Καθώς οι CA παίζουν το ρόλο της Αρχής προσδίδουν μια μεγαλύτερη δόση εμπιστοσύνης (Trust) στη γνησιότητα του δημόσιου κλειδιού ενός χρήστη. Τονίζουμε πως η χρήση των CA δεν λύνει το πρόβλημα της αυθεντικότητας (γνησιότητας) τον δημόσιου κλειδιού ενός χρήστη — απλά το μεταθέτει ένα επίπεδο πιο πάνω, προς την πλευρά τους. Διαισθητικά και μόνον, η επίδειξη της αστυνομικής

ταυτότητας σε μια συναλλαγή εμπνέει πολύ μεγαλύτερη εμπιστοσύνη από μια απλή υπογραφή μας, αν και πάντοτε υπάρχει η περίπτωση η αστυνομική ταυτότητα να είναι πλαστή! Βασιζόμαστε όμως στην εμπιστοσύνη που προσδίδει το συγκεκριμένο πιστοποιητικό και τις περισσότερες φορές αποδεχόμαστε τη συναλλαγή.



Σχήμα 2.1

Το παραπάνω πιστοποιητικό συνοδεύει όλα τα μηνύματα του χρήστη (στα οποία απαιτείται πιστοποίηση της ταυτότητάς του) και δουλεύει αποτελεσματικά μόνον εάν όλοι οι υπόλοιποι χρήστες εμπιστεύονται τη CA ή επιβεβαιώνουν με κάποιον τρόπο την ορθότητα (και την εγκυρότητα) τον πιστοποιητικού.

Στο παραπάνω, απλουστευμένο σενάριο, κάναμε ορισμένες παραδοχές σχετικά με τη δημιουργία των κλειδιών, υποθέτοντας πως ο χρήστης έχει την ικανότητα να δημιουργήσει μόνος του ένα κατάλληλο ζεύγος κρυπτογραφικών κλειδιών. Στην πράξη, ωστόσο, δεν είναι αυτή η συνηθισμένη περίπτωση. Το ερώτημα σχετικά με το ποιος δημιουργεί τα κλειδιά σε ένα τέτοιο σύστημα χρειάζεται πολύ μεγάλη ανάλυση που ξεπερνά τα όρια της συγκεκριμένης εργασίας.

Επίσης, στο παραπάνω σχήμα υποθέσαμε πως ο χρήστης παρουσιάζει -μαζί με τα κλειδιά- και κάποια άλλα διαπιστευτήρια. Η διαδικασία αυτή απαιτεί απόλυτη μυστικότητα καθώς εκεί εντοπίζεται όλη η ασφάλεια του συστήματος (ο χρήστης προσκομίζει στην CA πολλά

προσωπικά του δεδομένα που σε μερικές περιπτώσεις περιλαμβάνουν και το ιδιωτικό του κλειδί).

Χρειάζεται να τονίσουμε πως η δημιουργία ενός ψηφιακού πιστοποιητικού δεν σημαίνει την αυτόματη δημιουργία ηλεκτρονικής ταυτότητας. Ένα ψηφιακό πιστοποιητικά επιβεβαιώνει μόνον ότι ένα δημόσιο κλειδί είναι αυθεντικό, με την έννοια ότι μόνον ο ιδιοκτήτης του μπορεί έχει πρόσβαση στο αντίστοιχο ιδιωτικό κλειδί.

2.2 Υποδομή δημόσιου κλειδιού

Για να μπορέσει όμως το παραπάνω σύστημα να δουλέψει σωστά και αποδοτικά χρειάζεται να διευκρινίσουμε κάποια θεμελιώδη αξιώματα της κρυπτογραφίας δημόσιου κλειδιού. Όπως χαρακτηριστικά αναφέρουν οι Piper & Murphy, «κάθε πιστοποιητικό που περιέχει το δημόσιο κλειδί του χρήστη Α και είναι υπογεγραμμένο με το ιδιωτική κλειδί του Β, καθώς και κάθε πιστοποιητικό που περιέχει το δημόσιο κλειδί του χρήστη Β και είναι υπογεγραμμένο με το ιδιωτικό κλειδί του χρήστη Γ, επιτρέπει σε οποιονδήποτε γνωρίζει το δημόσιο κλειδί του χρήστη Γ να γνωρίζει ταυτόχρονα το δημόσιο κλειδί του χρήστη Α».

Μπορεί το παραπάνω αξίωμα να θεωρείται πολύπλοκο αλλά ουσιαστικά δεν είναι τίποτε άλλα από μια απλή διατύπωση της μεταβατικής ιδιότητας των μαθηματικών. Στην πράξη, όμως, ο τρόπος υλοποίησης όλων αυτών των βασικών υποθέσεων γίνεται πολύ σύνθετος και απαιτούνται πρότυπα στους αλγόριθμους (που χρησιμοποιούνται τόσο στη διαδικασία μιας ψηφιακής υπογραφής όσο και στη διαδικασία εξακρίβωσής της) καθώς και στα μορφότυπα (formats) των ψηφιακών πιστοποιητικών ώστε να αποπνέουν την ίδια αίσθηση εμπιστοσύνης. Για παράδειγμα, αν σε ένα πιστοποιητικό χρησιμοποιείται το σύστημα RSA ενώ σε ένα άλλο πιστοποιητικό το σύστημα DSA/El-Gamal, οι διαδικασίες που απαιτούνται για να εξακριβωθεί μια ψηφιακή υπογραφή είναι αρκετά πολύπλοκες και χρονοβόρες.

Για όλα αυτά τα θεμελιώδη ερωτήματα, καθώς και για ένα μεγάλο πλήθος πολιτικών και οικονομικών σκοπιμοτήτων που βρίσκονται πέρα από τα πλαίσια αυτής της εργασίας, είναι απαραίτητη μια υποδομή που θα μπορεί με σιγουριά να υποστηρίξει την τεχνολογική λύση. Χωρίς την υποδομή αυτή είναι αμφίβολο αν οι κάτοχοι ψηφιακών πιστοποιητικών

που χρησιμοποιούν άλλους αλγόριθμούς και πρότυπα θα μπορούν να επικοινωνούν με ασφάλεια (security) και σιγουριά (assurance).

Η υποδομή αυτή ονομάζεται Υποδομή Δημόσιού Κλειδιού (Public Key Infrastructure - PKI) και στις επόμενες ενότητες θα εξεταστούν οι βασικές αρχές λειτουργίας της καθώς και τα σημαντικότερα ερωτήματα που παραμένουν ακόμη ανοικτά. Τέλος, θα παρουσιαστούν εναλλακτικοί τρόποι για την καθολική χρήση της κρυπτογραφίας δημόσιου κλειδιού.

2.3 Η διαδικασία της (ψηφιακής) πιστοποίησης

Για να καταλάβουμε καλύτερα πώς η παραπάνω υποδομή (PKI) λειτουργεί, θα αναλύσουμε όλα τα στάδια (κάτι δηλαδή σαν τον κύκλο ζωής) της διαδικασίας της ψηφιακής πιστοποίησης: από τη στιγμή πού ιδρύεται μια CA μέχρι τη στιγμή που το πιστοποιητικό ενός χρήστη βρίσκεται σε λειτουργία.

Σε μια τυπική λοιπόν εφαρμογή, πρώτα από όλα δημιουργείται το ζεύγος κλειδιών της CA.

Κατά δεύτερο λόγο, λαμβάνει χώρα η διαδικασία δημιουργίας των κλειδιών ενός χρήστη. Όπως αναφέραμε και προηγουμένως, αυτό μπορεί να γίνει είτε από τον ίδιο το χρήστη, είτε από κάποιο άλλο εξουσιοδοτημένο πρόσωπα (ή οντότητα). Για να μπορέσει λοιπόν να χρησιμοποιήσει ο χρήστης το δημόσιο κλειδί του αποδοτικά, ζητά από τη CA ένα ψηφιακό πιστοποιητικό για τη γνησιότητα του δημόσιου κλειδιού του.

Στη συνέχεια, η CA, με κάποιο τρόπο, πιστοποιεί την ταυτότητα του χρήστη. Η διαδικασία αυτή περιέχει και χειρωνακτικές (manual) ενέργειες με σκοπό να πείσει ο χρήστης τη CA πως είναι ακριβώς αυτός που ισχυρίζεται. Μετά από αυτή την ενέργεια, το δημόσιο κλειδί του χρήστη υπογράφεται (ψηφιακά) με το ιδιωτικό κλειδί της CA. Η υπογραφή αυτή είναι το βασικό συστατικό τον πιστοποιητικού.

Τέλος, το πιστοποιητικό αποστέλλεται στο χρήστη, ο οποίος, αφού ελέγξει την ορθότητά του, μπορεί να αρχίσει να το χρησιμοποιεί.

Χρειάζεται να τονίσουμε πως υπάρχουν διαφορετικά επίπεδα στη διαδικασία της πιστοποίησης που κυρίως εξαρτώνται από τη χρησιμοποιούμενη εφαρμογή. Η εφαρμογή είναι αυτή που συνήθως καθορίζει το είδος της πιστοποίησης. Για παράδειγμα, μια πολυεθνική εταιρεία η οποία μεταφέρει —με ηλεκτρονικό τρόπο- κεφάλαια εκατομμυρίων δολαρίων καθημερινά έχει τελείως διαφορετικές απαιτήσεις ως προς τη γνησιότητα του (δημόσιου) κλειδιού της από έναν χρήστη που χρειάζεται ένα ψηφιακό πιστοποιητικό για να αποδεικνύει τη γνησιότητα του (δημόσιου) κλειδιού του όταν στέλνει και λαμβάνει email.

Στη λειτουργία λοιπόν του συστήματος των ψηφιακών πιστοποιητικών και του PKI υπάρχουν διαφορετικά επίπεδα πιστοποίησης της ταυτότητας ενός χρήστη, διαφορετικά επίπεδα ευθύνης της CA καθώς και διαφορετικοί χρησιμοποιούμενοι αλγόριθμοι.

Συνήθως, για περιβάλλοντα υψηλής ασφάλειας, η διαδικασία προσκόμισης των δικαιολογητικών για την πιστοποίηση της ταυτότητας ενός χρήστη στη CA περιλαμβάνει χειρωνακτικές μεθόδους, με τρόπο που να είναι κοινωνικά αποδεκτός. Για παράδειγμα, οι βιομετρικές τεχνικές αποτελούν μια πολύ αποτελεσματική μέθοδο για αυτή την απαίτηση. Μπορεί, λοιπόν, η CA να υποβάλει το χρήστη σε μια διαδικασία «σκαναρίσματος» της κόρης ή της ίριδας του ματιού με σκοπό να πιστοποιήσει την ταυτότητά του. Είναι προφανές ότι για τις περισσότερες εφαρμογές -πλην ελαχίστων εξαιρέσεων- η διαδικασία αυτή δεν θα είναι αποδεκτή από το χρήστη. Συνηθισμένα διαπιστευτήρια είναι η αστυνομική ταυτότητα, το δίπλωμα οδήγησης καθώς και άλλα -συνηθισμένα ή μη- δημόσια έγγραφα (ληξιαρχική πράξη γέννησης, πιστοποιητικό μόνιμης κατοικίας κτλ.). Τονίζουμε, ξανά, πως τα δικαιολογητικά αυτά εξαρτώνται άμεσα από την κρισιμότητα της εφαρμογής.

Στα τέλη της δεκαετίας του '90 πολλές εταιρείες (που έπαιζαν το ρόλο της CA) εξέδιδαν ψηφιακά πιστοποιητικά για χρήση email με μόνη απαίτηση από το χρήστη να στείλει ένα email από την ηλεκτρονική αυτή διεύθυνση. Σταδιακά, καθώς κάθε χρήστης μπορούσε να χρησιμοποιεί δεκάδες ηλεκτρονικές διευθύνσεις, οι εταιρείες απαιτούσαν την αποστολή email μόνον από εταιρικό λογαριασμό ηλεκτρονικού ταχυδρομείου, απορρίπτοντας, έτσι, τα email κατηγορίας gmail,

hotmail, yahoo!, κτλ. (στα οποία ο χρήστης μπορεί να δηλώνει τελείως ψευδή στοιχεία). Από το 2001 και μετά αυτός ο τρόπος σταδιακά καταργείται.

Σε πολλές περιπτώσεις, κυρίως για λόγους ασφάλειας, η διαδικασία εγγραφής (η προσκόμιση δηλαδή των αναλόγων διαπιστευτηρίων) μπορεί να γίνεται σε μια Αρχή Εγγραφής (Registration Authority) η οποία να είναι εντελώς ξεχωριστή από τη CA (Certification Authority). Για παράδειγμα, για την έκδοση (ψηφιακών) πιστοποιητικών στους πολίτες ενός δήμου, είναι λογικό να γίνεται η διαδικασία εγγραφής στη δημοτική αρχή και στη συνέχεια η δημοτική αρχή να εγγυάται για την πιστοποίηση της ταυτότητας των χρηστών. Όποιο και να είναι βέβαια το σενάριο, αν χρησιμοποιείται μια Αρχή Εγγραφής, απαιτούνται ασφαλείς διαδικασίες και πρακτικές για την ανταλλαγή των στοιχείων αυτού του είδους.

2.4 Συναρτήσεις κατακερματισμού (Hash Functions)

Η συνάρτηση κατακερματισμού, είναι μια μαθηματική συνάρτηση που δέχεται ως είσοδο κάποιο δεδομένο τυχαίου μεγέθους και επιστρέφει ένα ακέραιο σταθερού μεγέθους αναπαράστασης. Το μέγεθος αυτό μπορεί να είναι από 32bit μέχρι 256bit ή περισσότερα, ανάλογα με το λόγο χρήσης της συνάρτησης. Οι τιμές που επιστρέφει η συνάρτηση κατακερματισμού ονομάζονται τιμές κατακερματισμού (hash values), κώδικες κατακερματισμού (hash codes), αθροίσματα κατατεμαχισμού (hash sums) ή απλά τιμές κατακερματισμού (hashes).

Οι τιμές αυτές θα πρέπει να είναι διαφορετικές για διαφορετική είσοδο, καθώς η κύρια χρησιμότητα αυτών των συναρτήσεων είναι να ταυτοποιούν τα δεδομένα. Μια εφαρμογή αυτή της ιδιότητας είναι στην υλοποίηση της δομής δεδομένων σύνολο όπου θα πρέπει να αποτρέπεται η προσθήκη στοιχείου που το σύνολο ήδη περιέχει. Σε αυτή την περίπτωση τιμές 32bit αρκούν, εκτός αν το σύνολο μπορεί να φτάσει υπερβολικά μεγάλο μέγεθος. Μια άλλη εφαρμογή είναι στη δημιουργία ψηφιακών υπογραφών όπου χρησιμοποιούνται τιμές κατακερματισμού μεγάλου μεγέθους για να ελαχιστοποιηθεί ο κίνδυνος πλαστογράφησής τους.

Μια συνάρτηση κατακερματισμού μπορεί να αντιστοιχίζει δύο ή περισσότερες εισόδους στην ίδια τιμή κατακερματισμού. Στις περισσότερες εφαρμογές είναι επιθυμητή η ελαχιστοποίηση αυτών των

συγκρούσεων. Αυτό σημαίνει ότι η συνάρτηση κατακερματισμού θα πρέπει να αντιστοιχίζει κάθε είσοδο σε διαφορετική τιμή κατατεμαχισμού. Ανάλογα με την εφαρμογή χρήσης, η συνάρτηση κατακερματισμού σχεδιάζεται με διαφορετικές προδιαγραφές. Η ιδέα αυτών των συναρτήσεων εμφανίστηκε το 1950 αλλά ακόμη και σήμερα ο σχεδιασμός μιας καλής συνάρτησης κατακερματισμού είναι αντικείμενο έρευνας.

Οι συναρτήσεις κατακερματισμού συσχετίζονται (αν και πολλές φορές μπερδεύονται ως έννοιες) με τις συναρτήσεις αθροίσματος ελέγχου (π.χ. ο Κυκλικός Έλεγχος Πλεονασμού), τον υπολογισμό ψηφίου ελέγχου (check digit), δακτυλικά αποτυπώματα (fingerprints) και τους κώδικες ελέγχου λαθών (error correcting codes).

2.5 Ψηφιακές υπογραφές

Σύμφωνα με τον Διεθνή Οργανισμό Προτύπων (International Standards Organization ISO) ο όρος ψηφιακή υπογραφή θεωρείται σαν «μια συγκεκριμένη τεχνική πιστοποίησης ταυτότητας που χρησιμοποιείται για να αποδεικνύει την προέλευση του μηνύματος ώστε να επιλύει τυχόν διαφορές σχετικά με το ποιο μήνυμα έχει σταλεί (αν έχει σταλεί)».

Μια ψηφιακή υπογραφή, λοιπόν -για ένα συγκεκριμένο μήνυμα- αποτελείται από μερικά δεδομένα τα οποία επιβεβαιώνουν την ακεραιότητα των περιεχομένων ενός μηνύματος και τα οποία ο παραλήπτης μπορεί να τα χρησιμοποιήσει σαν απόδειξη. Επίσης, σε περίπτωση διένεξης, ένα τρίτο μέρος (ο διαιτητής) μπορεί να τα χρησιμοποιήσει για να δώσει λύση.

Η ψηφιακή υπογραφή εξαρτάται άμεσα από το μήνυμα το οποίο στέλνεται και βασίζεται σε μια συγκεκριμένη μυστική παράμετρο, η οποία είναι γνωστή μόνο στον αποστολέα αλλά μπορεί να επιβεβαιωθεί από τον καθέναν. Οι προηγούμενες ιδιότητες απαιτούν ένα μαθηματικό σχήμα που πρωτίστως θα έχει χαμηλές υπολογιστικές απαιτήσεις, καθώς η ψηφιακή υπογραφή θα πρέπει να είναι εύκολο να υπολογιστεί και να επιβεβαιωθεί από οποιονδήποτε ενδιαφερόμενο. Τέλος, η ψηφιακή υπογραφή θα πρέπει να είναι αδύνατο να αντιγραφεί.

Υπάρχουν δύο βασικοί τρόποι λειτουργίας των ψηφιακών υπογραφών: οι ψηφιακές υπογραφές με διαιτητή (arbitrated digital signatures) και οι πραγματικές ψηφιακές υπογραφές (true digital signatures).

2.5.1 Ψηφιακές υπογραφές με διαιτητή

Στο σχήμα αυτό χρησιμοποιείται ένα τρίτο - έμπιστο - μέρος (trusted third party - TTP) το οποίο υπογράφει (signs) και επιβεβαιώνει (verifies) μια ψηφιακή υπογραφή. Επίσης, η οντότητα αυτή παίζει το ρόλο του διαιτητή και χρησιμοποιείται για να επιλύει διαφορές που μπορεί να προκύψουν. Η έννοια του έμπιστου τρίτου μέρους προϋποθέτει τον αμοιβαίο σεβασμό και εμπιστοσύνη από τα δύο μέρη που θέλουν να επικοινωνήσουν. Τονίζεται πως η λειτουργία του όλου σχήματος βασίζεται στην εμπιστοσύνη (trust) που τα δύο αυτά μέρη έχουν προς τον διαιτητή.

Ας θεωρήσουμε το ακόλουθο σενάριο για να εξηγήσουμε το συγκεκριμένο τρόπο λειτουργίας των ψηφιακών υπογραφών αλλά και να τονίσουμε κάποιες αδυναμίες του σχήματος των κρυπτογραφικών αθροισμάτων (MACs) που περιγράφηκαν στην προηγούμενη ενότητα.

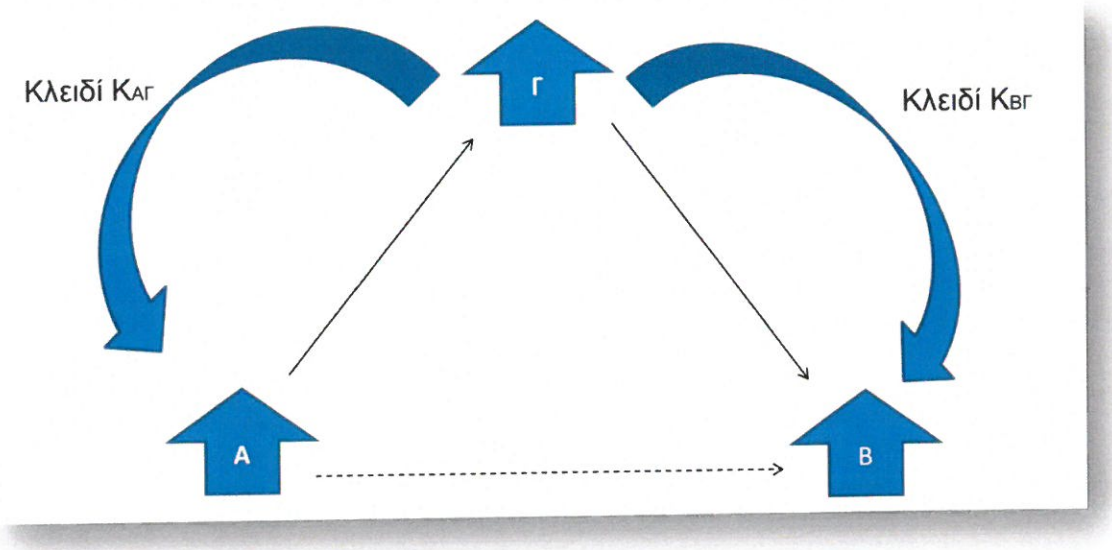
Υποθέτουμε πως η εταιρεία Α θέλει να επικοινωνήσει με την εταιρεία Β. Η εταιρεία Β με τη σειρά της θέλει κάποια μορφή εξασφάλισης (assurance) σχετικά με την αυθεντικότητα της ταυτότητας του Α. Επίσης, χρειάζεται να γνωρίζει, με κάποιο τρόπο, πως τα περιεχόμενα όλων των μηνυμάτων δεν έχουν μεταβληθεί (ακούσια ή εκούσια). Τέλος, η εταιρεία Β θέλει έναν τρόπο εξασφάλισης των μηνυμάτων της εταιρείας Α ώστε να μην μπορεί κάποια στιγμή εκείνη να αρνηθεί το γεγονός ότι έχει στείλει τα συγκεκριμένα μηνύματα.

Οι δύο παραπάνω εταιρείες αποφασίζουν να χρησιμοποιήσουν μιαν άλλη -τρίτη- εταιρεία (την εταιρεία Γ) σαν διαιτητή, της οποίας τις αποφάσεις θα δέχονται σαν οριστικές, αν προκύψει κάποιας μορφής διένεξη.

Για να εξηγήσουμε το σχήμα αυτό καλύτερα θα υποθέσουμε πως για την επικοινωνία μεταξύ των Α και Β χρησιμοποιείται ένας συμμετρικός αλγόριθμος και πως και τα δύο μέρη μοιράζονται ένα συμμετρικό κλειδί με το διαιτητή, αλλά όχι μεταξύ τους.

Έτσι, όταν η εταιρεία Α θέλει να στείλει ένα «υπογεγραμμένο» μήνυμα στην εταιρεία Β λαμβάνει χώρα η παρακάτω ακολουθία, που φαίνεται στο Σχήμα 2.2:

- Η εταιρεία Α στέλνει το μήνυμα πρώτα στην εταιρεία Γ, κρυπτογραφημένο με το κοινό τους κλειδί ($K_{Aγ}$) στο οποίο επισυνάπτει το $MAC_{Aγ}$ (Κρυπτογραφικό Άθροισμα Ελέγχου - Message Authentication Code) του συγκεκριμένου μηνύματος.
- Η εταιρεία Γ χρησιμοποιεί το κοινό αυτό κλειδί ($K_{Aγ}$) για να αποκρυπτογραφήσει το μήνυμα, να υπολογίσει το $MAC_{Aγ}$ και να επιβεβαιώσει την ακεραιότητα του μηνύματος αυτού.
- Στη συνέχεια η εταιρεία Γ στέλνει το ίδιο μήνυμα στην εταιρεία Β (που είναι και ο τελικός παραλήπτης) κρυπτογραφημένο με το κοινό κλειδί Κ. Επισυνάπτει επίσης τη $MAC_{γB}$ τιμή που προκύπτει από το χρησιμοποιούμενο κλειδί.
- Τέλος, η εταιρεία Β με τη σειρά της χρησιμοποιεί το κοινό κλειδί $K_{γB}$ που μοιράζεται με την εταιρεία Γ για να αποκρυπτογραφήσει το μήνυμα και να επιβεβαιώσει την τιμή του $MAC_{γB}$.



Σχήμα 2.2

Με αυτό τον τρόπο οι εταιρείες Α και Β μπορούν να επικοινωνούν μεταξύ τους χωρίς να χρειάζεται να μοιράζονται κάποιο κοινό κρυπτογραφικό κλειδί. Παρ' όλο που ίσως αυτό φαντάζει παράξενο, στις σημερινές εφαρμογές είναι αρκετά συνηθισμένο. Σε περίπτωση,

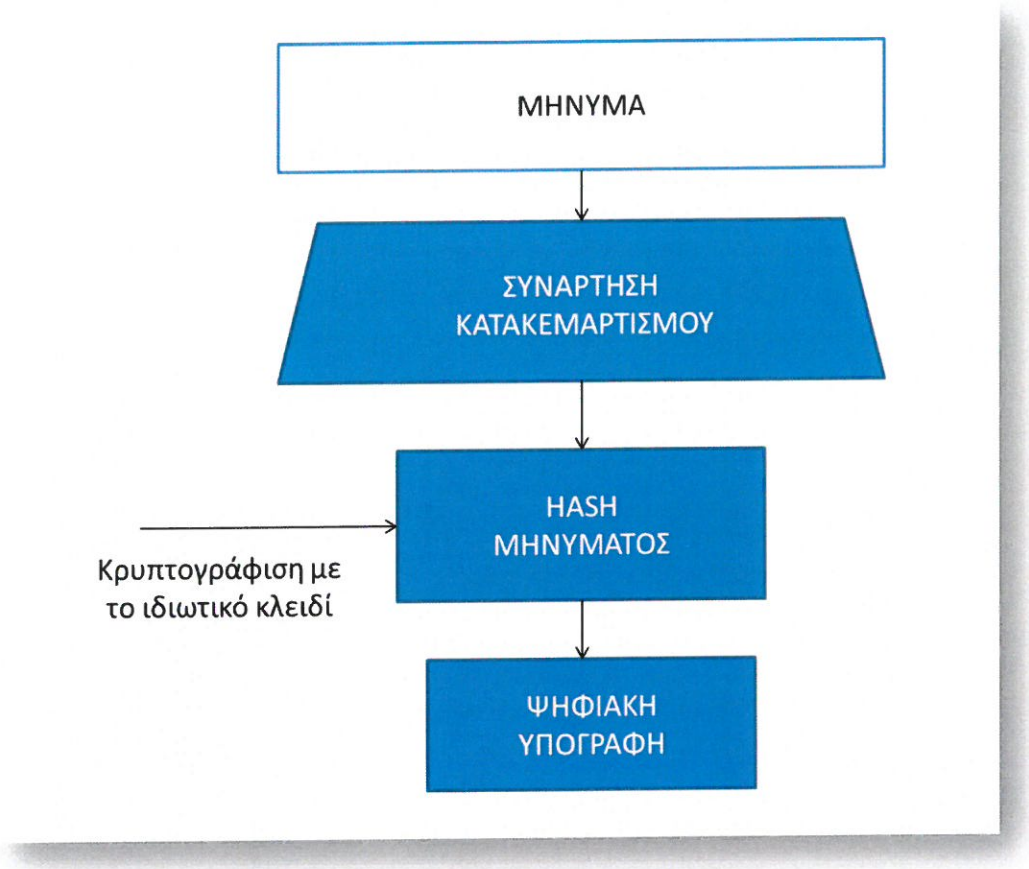
λοιπόν, που δύο μέρη θέλουν να επικοινωνήσουν -αλλά δεν υπάρχει αμοιβαία εμπιστοσύνη μεταξύ τους- το σχήμα αυτό μπορεί να δουλέψει αποτελεσματικά. Αξίζει να σημειώσουμε πως στο παραπάνω παράδειγμα ο Β δεν μπορεί να επιβεβαιώσει την τιμή του $MAC_{AΓ}$ (σε ορισμένες περιπτώσεις αυτό μπορεί να είναι κρίσιμο). Επίσης, σε ορισμένες εφαρμογές κρίνεται σκόπιμο πως πρέπει σε κάθε βήμα να επιβεβαιώνεται η ταυτότητα (identity) του κάθε μέρους. Αυτό δεν μπορεί να γίνει με το παραπάνω σενάριο.

2.5.2 Πραγματικές ψηφιακές υπογραφές

Τα συστήματα πραγματικών ψηφιακών υπογραφών στηρίζονται στις ομοιότητες που παρουσιάζουν με τα κρυπτογραφικά συστήματα δημοσίου κλειδιού, στα οποία οποιοσδήποτε ενδιαφερόμενος μπορεί να κρυπτογραφήσει ένα μήνυμα χρησιμοποιώντας τα δημόσιο κλειδί του επιθυμητού παραλήπτη αλλά μόνον ένας μπορεί να αποκρυπτογραφήσει το μήνυμα χρησιμοποιώντας το αντίστοιχο ιδιωτικό κλειδί. Έτσι, στα συστήματα πραγματικών ψηφιακών υπογραφών μόνον ένας μπορεί να υπογράφει ένα μήνυμα χρησιμοποιώντας το ιδιωτικό του κλειδί αλλά οποιοσδήποτε μπορεί να επαληθεύει τη γνησιότητα της συγκεκριμένης υπογραφές.

Ας εξετάσουμε λοιπόν με ένα παράδειγμα αντίστοιχο με εκείνο της προηγούμενης ενότητας πώς λειτουργεί ένα τέτοιο σύστημα. Η λειτουργία της παραγωγής μιας ψηφιακής υπογραφής φαίνεται στο Σχήμα 2.2

Σύμφωνα με το σχήμα αυτό, στο μήνυμα το οποίο μεταδίδεται εφαρμόζεται μια συνάρτηση κατακερματισμού (βλ. Ενότητα 2.4) η οποία μπορεί να δέχεται ως είσοδο ένα μήνυμα μεταβλητού μήκους και να παράγει ένα αντιπροσωπευτικό δείγμα σταθερού μήκους (το λεγόμενο hash του μηνύματος). Στη συνέχεια το hash αυτό κρυπτογραφείται με το ιδιωτικό κλειδί τον χρήστη ο οποίος θέλει να «υπογράψει» το συγκεκριμένο μήνυμα. Αξίζει να επισημάνουμε ότι η ψηφιακή υπογραφή δεν είναι τίποτε άλλο από το κωδικοποιημένο -με το ιδιωτικό κλειδί- hash ενός μηνύματος. Δεν υπογράφουμε δηλαδή το μήνυμα αλλά το hash του μηνύματος. Από τα παραπάνω προκύπτει τα συμπέρασμα πως η λειτουργία ταυ συστήματος αυτού βασίζεται κατά κύριο λόγο στη συνάρτηση κατακερματισμού (hash function).

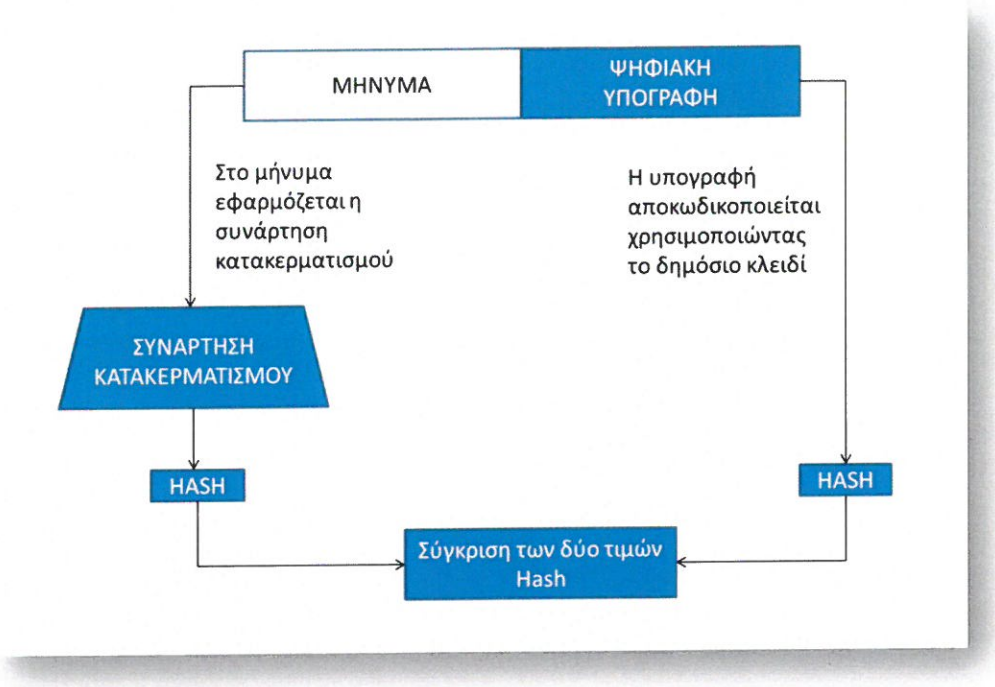


Σχήμα 2.3 Δημιουργία μιας ψηφιακής υπογραφής

Η συνάρτηση αυτή, για να λειτουργεί αποτελεσματικά, πρέπει να πληροί τις παρακάτω απαιτήσεις:

- Μπορεί να συμπιέζει μηνύματα οποιονδήποτε (μεταβλητού) μήκους σε ένα σταθερού — και προκαθορισμένου μήκους — αντιπροσωπευτικό δείγμα (hash).
- Διατηρεί τις ιδιότητες της μονόδρομης συνάρτησης (one way function), είναι δηλαδή υπολογιστικά αδύνατο να επανακατασκευαστεί το μήνυμα από το hash.
- Δεν έχει «συγκρούσεις» (collision free), είναι δηλαδή αδύνατο δύο διαφορετικά μηνύματα να έχουν το ίδιο hash αν εφαρμοστεί η ίδια συνάρτηση κατακερματισμού.

Η αντίστοιχη λειτουργία επαλήθευσης μιας ψηφιακής υπογραφής φαίνεται στο παρακάτω σχήμα.



Σχήμα 2.4— Η διαδικασία επαλήθευσης μιας ψηφιακής υπογραφής

Κατά τη διαδικασία αυτή ο παραλήπτης ενός μηνύματος λαμβάνει εκτός από το μήνυμα και κάποια κωδικοποιημένα δεδομένα (την ψηφιακή υπογραφή). Χρησιμοποιεί το δημόσιο κλειδί του αποστολέα για να αποκρυπτογραφήσει τα δεδομένα αυτά και να παράγει το αρχικό hash του μηνύματος. Παράλληλα, μιας και τα δύο μέρη τα οποία επικοινωνούν χρησιμοποιούν το ίδιο κρυπτοσύστημα, εφαρμόζει στο αρχικό μήνυμα την ίδια -με τον αποστολέα- συνάρτηση κατακερματισμού και υπολογίζει το hash του μηνύματος. Προφανώς, αν οι δύο τιμές του hash είναι όμοιες, η ψηφιακή υπογραφή είναι η σωστή.

2.6 Σύγκριση ψηφιακών και φυσικών υπογραφών

Οι ψηφιακές υπογραφές σχεδιάστηκαν με αρχικό σκοπό να αποτελούν το λογικό ισοδύναμο των παραδοσιακών φυσικών υπογραφών στον «ηλεκτρονικό κόσμο». Σε πολλές περιπτώσεις της καθημερινής μας ζωής, η υπογραφή μας χρησιμοποιείται προκειμένου να επιβεβαιώσουμε κάποια δήλωσή μας, δίνοντας με αυτόν τον τρόπο κάποια συμβολαιογραφική ισχύ. Και στις ψηφιακές υπογραφές το σενάριο είναι το ίδιο.

Η βασική διαφορά μεταξύ πραγματικών και ψηφιακών υπογραφών εντοπίζεται στο ότι η φυσική μας υπογραφή είναι η ίδια -ή τουλάχιστον παραπλήσια- κάθε φορά που τη χρησιμοποιούμε ανεξάρτητα από το είδος του εγγράφου το οποίο υπογράφουμε. Αντίθετα, η ψηφιακή υπογραφή, είναι άμεσα σχετιζόμενη με το κάθε μήνυμα και δεν είναι ποτέ η ίδια. Διαφορετικό μήνυμα σημαίνει άμεσα και διαφορετική ψηφιακή υπογραφή.

Επίσης, στον πραγματικό κόσμο, η χρήση της υπογραφής μας δεν προϋποθέτει κάποιο προηγούμενο μυστικό σε αντίθεση με τις ψηφιακές υπογραφές στις οποίες χρησιμοποιείται το ιδιωτικό κλειδί του χρήστη, του οποίου η κατοχή και χρήση πρέπει να απαγορεύεται σε οποιονδήποτε άλλον πέρα τον ιδιοκτήτη του (αν και σε ορισμένες περιπτώσεις πρέπει να προστατεύεται ακόμη και από αυτόν!).

2.6.1 Επιθέσεις στις ψηφιακές υπογραφές

Η χρήση των ψηφιακών υπογραφών σε εφαρμογές που απαιτούν πιστοποίηση ταυτότητας (authentication) προϋποθέτουν πως ο κάθε χρήστης ενός ιδιωτικού κλειδιού είναι άμεσα συνδεδεμένος με αυτό. Αυτό σημαίνει —χονδρικά —πως, στον ηλεκτρονικό κόσμο, «είμαστε το ιδιωτικό μας κλειδί».

Στην καθημερινή μας ζωή, όμως, υπάρχουν πολλές περιπτώσεις που ο κάτοχος ενός δημοσίου κλειδιού δεν είναι και ο ιδιοκτήτης ή ο δημιουργός του. Αυτό συμβαίνει, κυρίως, γιατί η δημιουργία των κρυπτογραφικών κλειδιών απαιτεί κάποιες (ελάχιστες) μαθηματικές γνώσεις ή/και υπολογιστικούς πόρους (τόσο σε υλικό όσο και σε λογισμικό) που δεν είναι πάντοτε διαθέσιμοι στο χρήστη. Η χρήση κρυπτογραφίας στοχεύει στο να κάνει περισσότερο ασφαλείς τις υπάρχουσες εφαρμογές χωρίς να επιβάλλει πρόσθετα προβλήματα στη λειτουργικότητά τους. Δεν θα πρέπει να ξεχνάμε πως ο χρήστης ενδιαφέρεται περισσότερο για την αποτελεσματική χρήση της εφαρμογής παρά για την ασφάλειά της.

Για το λόγο αυτό, πολλοί χρήστες αναθέτουν σε κάποιο έμπιστο τρίτο μέρος να δημιουργήσει τα κλειδιά για αυτούς ώστε να μπορούν να τα χρησιμοποιούν στις καθημερινές τους συναλλαγές. Το παράδοξο στο παραπάνω σενάριο είναι πως κανείς σχεδόν χρήστης δεν θέλει να υπογράφει αντ' αυτού το τρίτο μέρος, παρ' όλο που αυτό δημιούργησε (και γνωρίζει) τα κλειδιά του! Το ερώτημα κατά πόσον οι οντότητες που

δημιουργούν κλειδιά εκ μέρους κάποιων χρηστών θα πρέπει να γνωρίζουν την τιμή των κλειδιών αυτών ή να κρατούν ένα εφεδρικό τους αντίγραφο (backup key) παραμένει ακόμη ανοικτό. Πριν βιαστούμε να απαντήσουμε, ας σκεφτούμε σοβαρά την πιθανότητα απώλειας των κλειδιών από το χρήστη. Σε πολλές περιπτώσεις η απώλεια αυτή μπορεί να έχει καταστροφικές συνέπειες και ίσως να είναι αναγκαίο να υπάρχει ένα εφεδρικό αντίγραφο. Βέβαια, αν ο χρήστης έχει πέσει θέμα κλοπής των κλειδιών του, τότε το εφεδρικό αντίγραφο δεν παρέχει καμία προστασία καθώς ο επιτιθέμενος θα μπορεί να χρησιμοποιεί εκείνος τα κλειδιά και όχι ο χρήστης.

Για να υποδυθεί (impersonate) έναν χρήστη Α, ο επιτιθέμενος πρέπει είτε να αποκτήσει με κάποιο τρόπο το ιδιωτικό κλειδί του χρήστη, είτε να αντικαταστήσει το δημόσιο κλειδί του χρήστη με το αντίστοιχο δικό του. Τρόποι απόκτησης ενός ιδιωτικού κλειδιού, πέρα από τα φυσικά μέσα (!) είναι και οι μαθηματικές επιθέσεις (π.χ. η εύρεση ενός των παραγόντων p, q σε ένα ιδιωτικό RSA κλειδί) καθώς και η κλοπή μιας κρυπτοσυσκευής.

Από την άλλη πλευρά, η αντικατάσταση του δημόσιου κλειδιού του χρήστη Α με το αντίστοιχο κλειδί του επιτιθέμενου (καθώς μετά από μια τέτοια ενέργεια θα μπορεί να αποκρυπτογραφεί όλα τα μηνύματα που προορίζονται για αυτόν) είναι λιγότερο –φαινομενικά- δύσκολη. Η χρήση των ψηφιακών υπογραφών είναι άρρηκτα συνδεδεμένη με τη χρήση των ψηφιακών πιστοποιητικών (digital certificates) και την Υποδομή Δημόσιου Κλειδιού (Public Key Infrastructure - PKI).

3 Ο Αλγόριθμος RSA

3.1 Ιστορική Αναδρομή

Από το 1976 πολλοί αλγόριθμοι κρυπτογράφησης δημόσιου κλειδιού ήταν προτεινόμενοι, από τους οποίους αρκετοί ήταν ασφαλείς. Από εκείνους που θεωρούνται ακόμα ασφαλή, οι πιο πολλοί δεν είναι ενδεχομένως πρακτικοί. Είτε έχουν ένα πάρα πολύ μεγάλο κλειδί, χαρακτηριστικό που τους κάνει μη λειτουργικούς, είτε το ciphertext είναι κατά πολύ μεγαλύτερο από το αρχικό μήνυμα (plaintext). Μόνο μερικοί αλγόριθμοι δημοσίου κλειδιού μπορούν να χαρακτηριστούν τόσο ασφαλείς όσο και πρακτικοί. Άλλοι είναι κατάλληλοι για την κρυπτογράφηση και κατ' επέκταση για την διαχείριση του κλειδιού και άλλοι είναι μόνο χρήσιμοι για τις ψηφιακές υπογραφές.

Ο πρώτος ολοκληρωμένος αλγόριθμος που κρίθηκε κατάλληλος για τις εργασίες της κρυπτογράφησης αλλά και της ψηφιακής υπογραφής είναι ο RSA, που δημοσιεύτηκε τον Απρίλιο 1997. Από όλους τους δημοσίου κλειδιού αλγόριθμους, οι οποίοι προτάθηκαν κατά την διάρκεια των ετών, ο RSA είναι κατά πολύ ευκολότερος στην κατανόηση του και στην εφαρμογή του. Απολαμβάνει την αμέριστη εμπιστοσύνη της κρυπτογραφικής κοινότητας, κατακτώντας τον τίτλο του δημοφιλέστερου συστήματος δημοσίου κλειδιού. Πήρε το όνομα του από τους τρεις εμπνευστές του: Ron Rivest, Adi Shamir, και Leonard Adlema.

Από τον πρώτο καιρό, ο αλγόριθμος RSA έλαβε δίπλωμα ευρεσιτεχνίας (RSA patent). Σε πολύ μικρό χρονικό διάστημα έγινε πρότυπο εφαρμογής για τα κρυπτογραφικά συστήματα και αναγνωρίστηκε ευρέως η χρησιμότητα του. Υιοθετήθηκε στο πιο πολυχρησιμοποιημένο πρόγραμμα κρυπτογράφησης για τις ηλεκτρονικές επικοινωνίες που κυκλοφορεί σήμερα στο Internet, PGP (Pretty Good Privacy), καθώς είναι ο αλγόριθμος πάνω στον οποίο βασίστηκε η δομή σχεδίασης αλλά και η λειτουργία του.

Αξίζει εδώ να αναφερθεί ότι ο εμπνευστής της διαδικασίας κρυπτογράφησης που χρησιμοποιεί το PGP, Phil Zimmermann «σύρθηκε», ουσιαστικά, στα δικαστήρια με την κατηγορία της παράνομης εξαγωγής όπλων, αφού στις Η.Π.Α. η ισχυρή κρυπτογραφία θεωρείται όπλο. Τελικά, δεν καταδικάστηκε γιατί το δικαστήριο δεν

μπόρεσε να οριοθετήσει σαφώς την έννοια της εξαγωγής στα πλαίσια του Internet.

3.2 Περιγραφή του RSA

Ο αλγόριθμος RSA οφείλει το πραγματικά μεγάλο επίπεδο ασφάλειας του στην επιλογή μεγάλων πρώτων αριθμών. Πρώτος αριθμός κατά την μαθηματική ορολογία είναι ο αριθμός αυτός που έχει πολλαπλάσια του μόνο τον εαυτό του και την μονάδα. Βέβαια, την σημαντικότητα αυτού του συστατικού για τον RSA θα την καταγράψουμε παρακάτω.

Η παραγωγή ή ο υπολογισμός, αν θέλετε, του δημοσίου και του ιδιωτικού κλειδιού είναι λειτουργίες που εξαρτώνται από ένα ζευγάρι μεγάλων πρώτων αριθμών. Για χάρη της τυποποίησης της διαδικασίας κρυπτογράφησης θα τους ορίσουμε ως: p και q. Συνήθως, αυτοί οι αριθμοί έχουν μέγεθος 100 έως 200 ψηφία ή ακόμα και περισσότερα. Είναι προφανές πως επιλέγονται τυχαία, χωρίς να προϋπάρχει κάποια φόρμουλα επιλογής και φυσικά παραμένουν μυστικοί για το ευρύτερο σύνολο των χρηστών.

Αρχικά, λοιπόν, επιλέγονται οι πρώτοι αριθμοί p και q όπου για λόγους μεγιστοποίησης της ασφάλειας, σχεδόν πάντα, έχουν το ίδιο μέγεθος. Στην συνέχεια, υπολογίζεται το γινόμενο τους: n = p*q. Το δημόσιο κλειδί του αλγορίθμου είναι ένας εξίσου πρώτος αριθμός που επιλέγεται και αυτός τυχαία και ορίζουμε ως e. Η επιλογή του δημοσίου κλειδιού θα πρέπει να εξαντλεί δύο σημαντικές προϋποθέσεις. Το e θα πρέπει να είναι μικρότερο από την τιμή του αριθμού n και σχετικά πρώτο με το γινόμενο (p - 1)*(q - 1). Για να γίνει πιο κατανοητή η δεύτερη προϋπόθεση, αρκεί να πούμε πως πρέπει το δημόσιο κλειδί e και το γινόμενο (p - 1)*(q - 1), να μην έχουν κανένα κοινό παράγοντα εκτός φυσικά από την μονάδα.

Για την παραγωγή του ιδιωτικού κλειδιού (d), ο αλγόριθμος χρησιμοποιεί μια μαθηματική διαδικασία. Το ιδιωτικό κλειδί είναι ουσιαστικά το υπόλοιπο μιας διαίρεσης. Εδώ, η «πράξη» του υπολοίπου της διαίρεσης θα οριστεί ως mod. Ο μαθηματικός τύπος βάση του οποίου υπολογίζεται το ιδιωτικό κλειδί του RSA έχει την μορφή: $d = e^{-1 \bmod ((p-1)*(q-1))}$ ή $d = 1/e \bmod ((p - 1)*(q - 1))$. Παρατηρούμε πως η τιμή του d εξαρτάται τόσο από το δημόσιο κλειδί (e), όσο και από

τους δυο πρώτους αριθμούς. Φυσικά, θα πρέπει να σημειώσουμε πως το ιδιωτικό κλειδί και ο αριθμός n είναι και αυτοί σχετικά πρώτοι αριθμοί. Μετά την ολοκλήρωση της διαδικασίας, οι πρώτοι αριθμοί p και q δεν χρειάζονται και καταστρέφονται ή απλά απορρίπτονται, σε καμία όμως περίπτωση, ποτέ, δεν αποκαλύπτεται η τιμή τους.

Πριν ξεκινήσει η διαδικασία κρυπτογράφησης των αρχικών δεδομένων, διαιρούμε το plaintext σε αριθμητικά κομμάτια (blocks) μικρότερα από τον αριθμό n. (Για την δυαδική μορφή των δεδομένων αρκεί να πούμε πως επιλέγουμε την μεγαλύτερη δύναμη του 2, η οποία είναι μικρότερη από το n). Το plaintext ορίζεται ως m. Εάν οι p και q είναι αριθμοί 100 αριθμητικών ψηφίων, τότε προφανώς το n θα έχει λίγο παρακάτω από 200 ψηφία και το κάθε block του αρχικού κειμένου (m_i) θα πρέπει να έχει λίγα λιγότερα από 200 ψηφία. Εάν χρειαστεί γεμίζουμε με μηδενικά στοιχεία κάποιο κομμάτι προσθέτοντας τα στο αριστερό μέρος του. Ο τύπος κρυπτογράφησης είναι απλός: $c_i = m_i^{e \bmod n}$, όπου ως c_i ορίζουμε κάθε κομμάτι του κρυπτογραφημένου πια κειμένου. Το πιο σημαντικό χαρακτηριστικό του παραπάνω τύπου, πέρα από τις όποιες τυποποιημένες μεθόδους που χρησιμοποιούνται, είναι ότι η κρυπτογράφηση ολοκληρώνεται, μόνο με την γνώση και φυσικά την χρήση του δημοσίου κλειδιού e. Σε καμία άλλη περίπτωση αυτό δεν θα ήταν δυνατό.

Το κρυπτογραφημένο κείμενο (c) θα αποτελείται από blocks δεδομένων που θα έχουν περίπου το ίδιο μέγεθος με τα blocks του plaintext. Το σύνολο των κομματιών αυτών αποτελούν το chipertext. Για να αποκρυπτογραφήσει κάποιος τα δεδομένα χρησιμοποιεί κάθε κομμάτι του c και ανακτά το m: $m_i = c_i^{d \bmod n}$. Παρατηρούμε πως η ανάκτηση των αρχικών δεδομένων γίνεται μόνο με την χρήση του ιδιωτικού κλειδιού d. Εδώ τελικά βρίσκεται και η ουσία της όλης προσπάθειας περιγραφής του αλγορίθμου, δείχνοντας την διαφορά του τρόπου που λειτουργεί σε σχέση με έναν συμμετρικό αλγόριθμο.

Θα μπορούσε κάλλιστα στην παραπάνω περιγραφή του αλγορίθμου RSA να κρυπτογραφηθεί ένα μήνυμα με το ιδιωτικό κλειδί d και να αποκρυπτογραφηθεί αργότερα με το δημόσιο κλειδί e, στοχεύοντας στη πιστοποίηση της οντότητας που θα το κρυπτογραφήσει, αφού μόνο αυτή έχει στην κατοχή της το d. Η επιλογή είναι αυθαίρετη και εξαρτάται από τις οντότητες που επικοινωνούν μεταξύ τους.

Καλό είναι να παραθέσουμε ένα απλό αριθμητικό παράδειγμα της όλης λειτουργίας κρυπτογράφησης του RSA, με εξαιρετικά μικρούς πρώτους αριθμούς. Φυσικά, μια τέτοια κρυπτογράφηση δεν είναι λειτουργική, αλλά θα βοηθήσει στην ανάλυση του αλγορίθμου.

Ξεκινώντας με τα βήματα της διαδικασίας επιλέγουμε δυο πρώτους αριθμούς. Όπου p = 47 και q = 71. Υπολογίζουμε το γινόμενο τους: n = p*q - 47*71 = 3337

Επιλέγουμε σαν δημόσιο κλειδί την τιμή e = 79. Ελέγχουμε εάν το e και το γινόμενο (q-1)*(q-1) = 3220 έχουν κοινό παράγοντα εκτός του 1, προϋπόθεση που ισχύει. Σε αυτή την περίπτωση το ιδιωτικό κλειδί θα είναι: $d = 79^{-1 \bmod 3220} = 1019$.

Η τιμή αυτή υπολογίζεται χρησιμοποιώντας τον αλγόριθμο Euclidean. Πρακτικά, ψάχνουμε μια τιμή d τέτοια ώστε η τιμή (q-1)*(q-1) = 3220 να διαιρείται από το e*d-1 = 79*d-1.

Έστω ότι το μήνυμα που θα κρυπτογραφήσουμε είναι το m=6882326879666683. Αρχικά, το σπάμε σε μικρότερα blocks. Στην περίπτωση που εξετάζουμε blocks των τριών ψηφίων είναι η καλύτερη επιλογή. Όποτε το m σπάει σε έξι κομμάτια: m_1=688, m_2=232, m_3=687, m_4=966, m_5=668 και m_6=003 (θυμηθείτε ότι γεμίζουμε με μηδενικά στοιχεία το τελευταίο block για να έχει το ίδιο μέγεθος με τα υπόλοιπα). Το πρώτο block όταν κρυπτογραφηθεί θα μας δώσει:
$d = m_1^{\,e \bmod n} = 688^{\,79 \bmod 3337} = 1570$.

Η εκτέλεση της ίδιας λειτουργίας στα επόμενα κομμάτια παράγει το κρυπτογραφημένο κείμενο: c=1570 2756 2091 2276 2423 158.
Η αποκρυπτογράφηση του c απαιτεί την ίδια διαδικασία χρησιμοποιώντας όμως ως κλειδί αποκρυπτογράφησης το ιδιωτικό κλειδί d = 1019: $m_1 = c_1^{\,d \bmod n} = 1570^{\,1019 \bmod 3337} = 688 = m_1$.
Το υπόλοιπο των αρχικών δεδομένων μπορεί βεβαίως να ανακτηθεί με αυτόν ακριβώς τον τρόπο.

Παρακάτω δύνεται πιο συνοπτικά ο τρόπος λειτουργίας του RSA, πως παράγονται και επιλέγονται τα κλειδιά αλλά και πως αυτά χρησιμοποιούνται στην κωδικοποίηση των δεδομένων.

Δημόσιο κλειδί: e

Επιλέγουμε δυο πρώτους αριθμούς, p και q οι οποίοι πρέπει να παραμείνουν μυστικοί. Υπολογίζουμε το γινόμενο τους:

Όπου n=p*q

Επιλέγουμε τυχαία ένα πρώτο αριθμό e, όπου e < n και e, (p - 1)*(q -1) σχετικά πρώτοι αριθμοί.

Ιδιωτικό κλειδί: d

Όπου : $d=e^{-1 \bmod ((p-1)*(q-1))}$

Δηλαδή d=1/e mod ((p - 1)*(q - 1))

Κρυπτογράφηση:

Το m είναι το αρχικό μήνυμα

και το c το κρυπτογραφημένο μήνυμα που για κάθε κομμάτι τους παράγεται:

$c_i = m_i^{\ e \bmod n}$

Είναι εφικτό μόνο όταν ξέρουμε το δημόσιο κλειδί e.

Αποκρυπτογράφηση:

Το c αποτελεί τα κρυπτογραφημένο δεδομένα που λαμβάνουμε, το d το ιδιωτικό κλειδί και το m_i το αρχικό κομμάτι που θα ανακτήσουμε:

$m_i = c_i^{\ d \bmod n}$

Είναι εφικτό μόνο εάν γνωρίζουμε το ιδιωτικό κλειδί d.

3.3 Ασφάλεια του RSA

Η ασφάλεια του RSA εξαρτάται ολοκληρωτικά από το πρόβλημα της εξεύρεσης μεγάλων πρώτων αριθμών, η οποία από μαθηματικής πλευράς είναι δύσκολη έως και αδύνατη. Εάν αυτό ήταν εφικτό τότε κάποιος κρυπταναλυτής γνωρίζοντας το γινόμενο των πρώτων αριθμών η, θα μπορούσε να υπολογίσει (ανακτήσει) το αρχικό κείμενο (m) από το δημόσιο κλειδί e και από το κρυπτογραφημένο κείμενο (ciphertext c). Ενδεχομένως, θα είχε ανακαλυφθεί ένας εξ ολοκλήρου διαφορετικός τρόπος κρυπτανάλυσης του RSA. Εντούτοις, εάν αυτός ο νέος τρόπος επιτρέπει σε έναν κρυπταναλυτή να υπολογίσει το ιδιωτικό κλειδί d, θα μπορούσε κάλλιστα να χρησιμοποιηθεί ως νέος τρόπος εξεύρεσης των μεγάλων πρώτων αριθμών. Πολλοί ερευνητές είναι αυτοί που αφιερώνουν πολύ χρόνο στην προσπάθεια ανακάλυψης νέων πρώτων αριθμών με δεκάδες ψηφία. Ο υπολογισμός, λοιπόν, του συντελεστή n είναι ο προφανέστερος τρόπος κρυπτανάλυσης. Η τεχνολογία

«επιβάλει» αυτή την περίοδο έναν συντελεστή με περισσότερα από 220 ψηφία.

Μια άλλη συνηθισμένη επίθεση στον αλγόριθμο RSA είναι η εικασία της τιμής της παράστασης (p-1)*(q-1). Αυτή η προσπάθεια κρυπτανάλυσης βέβαια έχει δειχθεί πως δεν είναι ευκολότερη από την εξεύρεση των πρώτων αριθμών. Οι περισσότερες κοινές τεχνικές υπολογισμού των πρώτων αριθμών δεν μπορούν να χαρακτηριστούν παρά πιθανολογικές. Είναι βεβαίως δυνατόν, για κάποιον αναλυτή να δοκιμάσει κάθε πιθανή τιμή του ιδιωτικού κλειδιού, για να καταλήξει στη σωστή που θα του δώσει την δυνατότητα να αναστήσει τα αρχικά δεδομένα. Μια τέτοια προσπάθεια θα ήταν αποδοτικότερη από τον πιθανό υπολογισμό του n, αλλά η εξαντλητική αναζήτηση προφανώς θα χρειαζόταν μια τεραστία χρονική περίοδο για να τεθεί υπό επιτυχία.

Η επιλογή των δύο πρώτων αριθμών, συστατικό ασφάλειας για τον αλγόριθμο, είναι μια ιδιαίτερη διαδικασία. Οι αριθμοί αυτοί, υπήρχε αρχικά η πεποίθηση, πως θα πρέπει να είναι «ισχυροί». «Ισχυροί» πρώτοι αριθμοί καλούνται αυτοί με ορισμένες ιδιότητες οι οποίες καθιστούν το γινόμενο τους (n = p*q), σκληρό ενάντια σε συγκεκριμένες μεθόδους εξεύρεσης του. Εντούτοις, οι πρόοδοι στην διάρκεια των τελευταίων χρόνων έχουν καταστήσει επισφαλές το πλεονέκτημα των «ισχυρών» πρώτων αριθμών. Επομένως, η επιλογή ενός παραδοσιακού «ισχυρού» ζευγαριού των p και q από μόνο του δεν αυξάνει σημαντικά την ασφάλεια στην εφαρμογή του RSA. Ενδεχομένως δεν υπάρχει κάποιο πρόβλημα στο να επιλέγονται όλο και μεγαλύτεροι πρώτοι αριθμοί. Πιο πάνω, εξάλλου, αναφέραμε πως οι αριθμοί αυτοί συνήθως έχουν μέγεθος έως και 200 ψηφία. Είναι σχεδόν σίγουρο πως στο άμεσο μέλλον η κρυπτογραφική τεχνολογία να «απαιτεί» ακόμα μεγαλύτερους πρώτους αριθμούς. Η χρήση τους στην κρυπτογραφική διαδικασία του RSA θα είναι πάντα αναγκαία και σημαντική.

Το κυριότερο πρόβλημα, ίσως, στην λειτουργία του RSA έχει να κάνει επίσης με τους μεγάλους πρώτους αριθμούς. Μια ρεαλιστική αντιμετώπιση της όλης διαδικασίας θα όριζε σημαντικό το ενδεχόμενο οι πρώτοι αριθμοί που χρησιμοποιούνται (p και q), ακριβώς γιατί είναι μεγάλοι, να είναι σύνθετοι αριθμοί. Ποιο μπορεί να ήταν λοιπόν το πρόβλημα σε μια τέτοια περίπτωση; Αρχικά, πρέπει να εξαντληθούν όλες οι πιθανότητες για να μην συμβεί αυτό. Σε μία όμως αντίθετη περίπτωση, το σίγουρο θα ήταν ότι το σύνολο των διαδικασιών της

κρυπτογράφησης και της αποκρυπτογράφησης δεν θα λειτουργούσαν κατάλληλα. Έχει αποδειχθεί πως υπάρχουν κάποιοι πρώτοι αριθμοί που η χρησιμοποίηση τους θα φέρει σε αποτυχία το σύστημα κρυπτογράφησης του RSA. Οι πρώτοι αριθμοί πρέπει να πούμε πως επιλέγονται συνήθως μέσα από κάποιες πιθανολογικές μεθόδους και τεχνικές. Η ανίχνευση αυτών των σύνθετων πρώτων αριθμών δεν είναι δυνατή, μέσα από τέτοιες τεχνικές. Βεβαίως και οι αριθμοί αυτοί είναι επισφαλείς, αλλά σίγουρα είναι και αρκετά σπάνιοι.

Κατά διαστήματα, πολλοί υποστηρίζουν ότι έχουν βρει εύκολους τρόπους να «σπάσουν» τον RSA ανακτώντας το ιδιωτικό του κλειδί, αλλά καμία τέτοια άποψη δεν πλησίασε την πραγματικότητα. Το 1998 ο William Payne πρότεινε μια μέθοδο βασισμένη στο θεώρημα Fermat. Αυτή η μέθοδος ήταν τελικά πιο αργή ακόμα και από την διαδικασία «παραγωγής» του συντελεστή n. Ο αλγόριθμος RSA έχει αντέξει στην πολύπλευρη κρυπτανάλυση που έχει δεχθεί, κερδίζοντας την εμπιστοσύνη όχι μόνο των αναλυτών της κρυπτογραφικής κοινότητας. Είναι εξάλλου αποδεδειγμένο πως ακόμη και η ανάκτηση ορισμένων κομματιών από τις πληροφορίες του κρυπτογραφημένου μηνύματος, είναι τόσο δύσκολη όσο η αποκρυπτογράφηση ολόκληρου του μηνύματος.

Για την λειτουργικότητα και πόσο μάλλον για την διατήρηση του επιπέδου ασφάλειας που προσφέρει ο RSA, κάποια σημαντικά στοιχεία λαμβάνονται πάντα υπόψη. Ένας συντελεστής n δεν θα πρέπει σε καμία περίπτωση να είναι είτε να θεωρείται κοινός. Σε μία κρυπτογραφημένη επικοινωνία μέσω ενός δικτύου, ίσως η χρήση του να επιφέρει αρνητικές επιπτώσεις. Δεν είναι όμως αρκετό να υπάρχει και να εφαρμόζεται ένας ασφαλής κρυπτογραφικός αλγόριθμος, όπως ο RSA. Ολόκληρο το κρυπτογραφικό σύστημα και το κρυπτογραφικό πρωτόκολλο πρέπει να είναι ασφαλή. Μια αποτυχία σε οποιαδήποτε από αυτές τις τρεις περιοχές καθιστά το γενικότερο σύστημα προβληματικό.

Το μήκος των κλειδιών (δημόσιο και ιδιωτικό) που χρησιμοποιεί ο RSA έχει γίνει κατά καιρούς σημείο αναφοράς και απαίτησης. Ένα κλειδί 512 bits δεν θεωρείται πλέον ασφαλές. Εάν χρησιμοποιηθεί ένα 768-bit κλειδί αναμφισβήτητα μειώνεται κατά πολύ η πιθανότητα ανάκτησης του. Η απαίτηση όμως της RSA Security είναι η χρήση κλειδιών μήκους 1024 bits. Θεωρείται πως μια διαδικασία κρυπτογράφησης του RSA με

ένα 1024-bit κλειδί θα είναι για μερικά χρόνια ακόμα ασφαλής. Μια μεγαλύτερη τιμή κλειδιού (2048 bits ή ακόμα και 4096 bits) είναι αυτή που θα καταγραφεί ως εξαιρετικά ασφαλής, αλλά ο χρόνος που θα χρειασθεί για να κρυπτογραφηθεί και να αποκρυπτογραφηθεί το αρχικό μήνυμα σίγουρα δεν χαρακτηρίζει το σύστημα πρακτικό.

Ο μεγαλύτερος όμως κίνδυνος στην χρησιμοποίηση ενός πολύ μεγάλου κλειδιού είναι η ψεύτικη αίσθηση ασφάλειας που παρέχει στους χρήστες. Μια 4096-bit ασφάλεια σε ένα σύστημα, αντηχεί εντυπωσιακά σε μια προσπάθεια μάρκετινγκ, αλλά εάν το ιδιωτικό κλειδί δεν προστατεύεται επαρκώς ή η τυχαία γεννήτρια παραγωγής πρώτων αριθμών δεν είναι και τόσο τυχαία, προφανώς η ύπαρξη ενός μεγάλου κρυπτογραφικού κλειδιού είναι άχρηστη.

3.4 Χρησιμότητα του RSA

Σε μια προσπάθεια να οριοθετήσουμε επαρκώς την χρησιμότητα του αλγορίθμου RSA σήμερα, αρκεί να πούμε πως είναι ο αλγόριθμος εκείνος που θεωρείται ευρέως πρότυπο κρυπτογράφησης και η τεχνολογία ανάπτυξης του είναι εξασφαλίζει την ύπαρξη της πλειοψηφίας των εφαρμογών e-business στον κόσμο του διαδικτύου.
Σε συνέδριο που διοργάνωσε τον Σεπτέμβριο του 2000 η RSA security, η αρχική πεποίθηση ήταν η καθησύχαση της κοινότητας, μετά από την παραπληροφόρηση που είχε υπήρχε σχετικά με την λήξη του διπλώματος ευρεσιτεχνίας του RSA. Το U.S. patent που είχε χορηγηθεί στον RSA το Σεπτέμβριο του 1983, έληγε 17 χρόνια μετά. Εκτός από την επίσημη θέση της RSA security σχετικά με το δίπλωμα ευρεσιτεχνίας, διατυπώθηκαν πολύ σημαντικά συμπεράσματα.

Για σχεδόν δυο δεκαετίες, περισσότερες από 800 επιχειρήσεις που επεκτείνονται στην αναπτυσσόμενη ηλεκτρονική αγορά έχουν αποδείξει την εμπιστοσύνη τους στην ασφάλεια του RSA. Εμπιστοσύνη για έναν αλγόριθμο που μπορεί να παρέχει το επιθυμητό επίπεδο στο κρίσιμο σημείο της ασφάλειας, με χρόνο-λειτουργικές εφαρμογές και πόρους κρυπτογράφησης. Όλο αυτό τον καιρό η RSA security είχε κάνει τροποποιήσεις στον σχεδίασμά και την βάση πάνω στην οποία ο αλγόριθμος εφαρμόζεται, συμπεριλαμβάνοντας διάφορες βελτιώσεις απόδοσης, μη απεικονισμένες στο αρχικό δίπλωμα ευρεσιτεχνίας. Αυτές οι τροποποιήσεις είχαν στόχο ένα ευρύ φάσμα εφαρμογών λογισμικού και λειτουργικών συστημάτων.

Η τεχνολογία κρυπτογράφησης έχει αναδιαμορφώσει ένα εξ ολοκλήρου νέο επίπεδο σπουδαιότητας στον κόσμο των επιχειρήσεων και στην ανάπτυξη του ηλεκτρονικού εμπορίου. Εκεί, η τεχνολογία σχεδίασης του αλγορίθμου RSA συνεχίζει να διατηρεί έναν ηγετικό ρόλο. Όπως είναι φυσικό, οριοθετήθηκαν νέα θεσμικά πλαίσια και κανόνες για την αναπτυσσόμενη ηλεκτρονική αγορά. Μέσα σε αυτό το πλαίσιο, η νομοθεσία ηλεκτρονικών υπογραφών ως κανονισμός και απαίτηση διεξαγωγής διατραπεζικών συναλλαγών και όχι μόνο, βρίσκει εφαρμογή στην χρήση του RSA και άλλων βεβαίως αλγορίθμων-τεχνικών δημοσίου κλειδιού. Όπως γίνεται εύκολα κατανοητό, η RSA κρυπτογράφηση θα διαδραματίζει έναν βασικό ρόλο στην περαιτέρω επέκταση των πρωτοβουλιών ηλεκτρονικού εμπορίου.

Ανεξάρτητα από τα άλλα επίσημα πρότυπα, η ύπαρξη προτύπων όπως το κρυπτοσύστημα RSA είναι εξαιρετικά σημαντική για την ανάπτυξη μιας ψηφιακής οικονομίας. Εάν ένα σύστημα δημοσίου κλειδιού χρησιμοποιείται παντού για επικύρωση των οντοτήτων, τα ψηφιακά υπογεγραμμένα έγγραφα μπορούν κατόπιν να ανταλλαχθούν μεταξύ των χρηστών, σε διαφορετικά μέρη του κόσμου χρησιμοποιώντας διαφορετικό λογισμικό σε διαφορετικές πλατφόρμες. Αυτή η διαλειτουργικότητα είναι αδιαμφισβήτητα απαραίτητη για να αναπτυχθεί μία πραγματικά μεγάλη ψηφιακή οικονομία. Η υιοθέτηση του κρυπτογραφικού συστήματος RSA έχει αυξήσει σημαντικά την πεποίθηση για την επίτευξη ενός τέτοιου στόχου. Φυσικά, η αποδοχή ενός και μόνο παγκόσμιου προτύπου να εξαρτάται και από άλλους παράγοντες, το ευχάριστο όμως είναι ότι ο RSA περικλείει το σύνολο των ενδεχόμενων απαιτήσεων.

Ο αλγόριθμος RSA χρησιμοποιείται σε πολλά λογισμικά εμπορικών προϊόντων και σχεδιάζεται να χρησιμοποιηθεί σε πολύ περισσότερα. Εκτός αυτών, ο RSA εφαρμόζεται στα τρέχοντα λειτουργικά συστήματα των Microsoft, Apple και Sun. Σε επίπεδο hardware, μπορεί να «βρεθεί» στην ασφάλεια των ασύρματων τηλεφώνων, στις κάρτες δικτύου Ethernet, καθώς και στην τεχνολογία ανάπτυξης των έξυπνων καρτών (smart cards).

Επιπλέον, ο αλγόριθμος έχει ενσωματωθεί σε όλα τα σημαντικά πρωτόκολλα ασφάλειας της επικοινωνίας μέσω του διαδικτύου συμπεριλαμβανομένου και του SSL (Secure Sockets Layer). Το SSL

πρωτόκολλο αναπτύχθηκε από την Netscape Communications Corporation για να παρέχει την ασφάλεια και τη μυστικότητα των πληροφοριών που ανταλλάσσονται μέσω του Διαδικτύου. Το πρωτόκολλο υποστηρίζει server και client επικύρωση, ενώ είναι σε θέση να διαπραγματευτεί τα κλειδιά κρυπτογράφησης. Υποστηρίζει το κρυπτοσύστημα RSA, καθώς διατηρεί την ασφάλεια και την ακεραιότητα του καναλιού μετάδοσης με τη χρησιμοποίηση και την εφαρμογή της κρυπτογράφησης.

Όταν οι κύριοι χρηματοδοτικοί οργανισμοί της αμερικανικής οικονομικής βιομηχανίας ανέπτυσσαν τα πρότυπα για τις ψηφιακές υπογραφές, υιοθέτησαν το ANSI X9.31, το οποίο εφαρμόζει την τεχνολογία ψηφιακών υπογραφών του RSA. Μια ψηφιακή υπογραφή είναι το ακριβές εργαλείο που είναι απαραίτητο, για να μετατραπούν τα πιο ουσιαστικά έγγραφα σε ψηφιακής μορφής δεδομένα. Η χρήση της ψηφιακής υπογραφής του RSA προσφέρει πολλά πλεονεκτήματα, κυρίως σε ταχύτητα επαλήθευσης της υπογραφής και βρίσκει εφαρμογή σε μεγάλα εμπορικά συστήματα ακόμα και σήμερα.

Μερικές επιχειρήσεις θεωρούν ότι, με την κρυπτογράφηση των στοιχείων, λύνουν τα περισσότερα προβλήματα ασφάλειας και δεν φροντίζουν να αποτρέψουν στους επιτιθεμένους το «σπάσιμο» των συστημάτων επειδή τα δεδομένα θα είναι άχρηστα σε αυτούς, εφόσον είναι κρυπτογραφημένα. Καλό είναι οι χρήστες να μην βασίζονται σε μια ψεύτικη αίσθηση της ασφάλειας και να στηριχθούν εκεί χωρίς να δώσουν την απαραίτητη προσοχή στην ασφάλεια δικτύων και λειτουργικών συστημάτων. Ένας από τους σχεδιαστές του RSA, ο Ron Rivest, διατύπωσε την άποψη πως η κρυπτογράφηση δεν είναι μια λύση στις ανησυχίες ασφάλειας αλλά ένας τρόπος να μεταφερθεί το πρόβλημα: «Αντί της διαχείρισης 100 MB των κρίσιμων δεδομένων, αρκεί μόνο να διαχειριστείτε ένα κρυπτογραφικό κλειδί, που είναι πολύ μικρότερο στο μέγεθος».

4 Ο Αλγόριθμος DSA

4.1 Ιστορική Αναδρομή

Τον Αύγουστο του 1991, το Εθνικό Ίδρυμα Προτύπων και Τεχνολογίας (NIST) πρότεινε το DSA (Digital Signature Algorithm) ως ένα Πρότυπο Ψηφιακών Υπογραφών (DSS). Το προτεινόμενο πρότυπο είναι ένας αλγόριθμος δημοσίου κλειδιού κατάλληλος μόνο για ψηφιακές εφαρμογές υπογραφών. Ο αλγόριθμος κάνει χρήση ενός δημοσίου κλειδιού, με το οποίο ο παραλήπτης ενός μηνύματος ελέγχει την ακεραιότητα των στοιχείων και φυσικά την ταυτότητα του αποστολέα. Μπορεί επίσης να χρησιμοποιηθεί από μια τρίτη οντότητα, για μια διαδικασία εξακρίβωσης της αυθεντικότητας μιας υπογραφής, αλλά και των δεδομένων που συνδέονται με αυτήν.

Το 1987 το NIST, σε μια προσπάθεια ανάπτυξης προτύπων ασφάλειας, είχε εκδώσει την εξής ανακοίνωση προς τους αναλυτές και ερευνητές της κρυπτογραφικής κοινότητας: «Αναζητάμε την οικονομικώς αποδεκτή ασφάλεια και μυστικότητα των ομοσπονδιακών πληροφοριών σε μια επιλογή με τα πιο επιθυμητά χαρακτηριστικά λειτουργίας και χρήσης». Ως επιλογή χαρακτηρίστηκε ένας αλγόριθμος που θα ανταποκρίνεται στις οικονομικές και λειτουργικές απαιτήσεις που είχαν οριοθετηθεί. Μεταξύ των παραγόντων που· εξετάστηκαν κατά τη διάρκεια αυτής της διαδικασίας ήταν το παρεχόμενο επίπεδο ασφάλειας, η ευκολία της εφαρμογής, ο αντίκτυπος στην εθνική ασφάλεια και τα νομοθετικά πλαίσια και το επίπεδο αποδοτικότητας της ψηφιακής υπογραφής αλλά και των διαδικασιών επαλήθευσης.

Η ανακοίνωση του NIST δημιούργησε μια δίνη κριτικών και κατηγοριών. Η RSA Security ήταν ο πιο σκληρός επικριτής καθώς, απολύτως φυσιολογικά, η επιθυμία της ήταν να χρησιμοποιείται ο αλγόριθμος RSA ως πρότυπο και όχι κάποιος άλλος. Πολλές μεγάλες επιχειρήσεις λογισμικού που είχαν χορηγήσει άδεια ήδη στον αλγόριθμο RSA στράφηκαν εναντίον του DSA. Επιχειρήσεις όπως η IBM, η Apple, η Microsoft, η DEC και η Sun είχαν ήδη ξοδέψει μεγάλα ποσά εφαρμόζοντας την ψηφιακή υπογραφή (digital signature) του RSA και προφανώς δεν ήταν διατεθειμένες να χάσουν αυτή την επένδυση.

Τελικά, τον Μάιο του 1994 εκδόθηκε στην τελική μορφή του το πρότυπο του DSA και καθορίστηκε άμεσα ως το ψηφιακό πρότυπο επικύρωσης της κυβέρνησης των Ηνωμένων Πολιτειών. Η ανακοίνωση

του Εθνικού Ιδρύματος Προτύπων και Τεχνολογίας (NIST) ήταν σαφής. «Ο αλγόριθμος DSA και όλα τα προτεινόμενα πρότυπα ψηφιακών υπογραφών ισχύουν σε όλα τα ομοσπονδιακά τμήματα και τις αντιπροσωπείες για την προστασία των αταξινόμητων πληροφοριών. Επίσης, η υιοθέτηση τους είναι διαθέσιμη σε ιδιωτικές και εμπορικές εταιρίες στο πλαίσιο της ανάγκης ύπαρξης ψηφιακών υπογραφών».

Η κριτική προς τον αλγόριθμο συνεχίστηκε και μετά την έκδοση του. Φυσικά ήταν βασισμένη και είχε εστιάσει σε μερικά όντως κύρια ζητήματα. Αρχικά, ο αλγόριθμος στερείται τη βασική ικανότητα ανταλλαγής των κρυπτογραφικών κλειδιών. Το κρυπτογραφικό σύστημα ήταν πάρα πολύ πρόσφατο και υπόκειται σε λίγη διερεύνηση για τους χρήστες, για να είναι βέβαιοι της δύναμής του. Η επαλήθευση των υπογραφών με εφαρμογή του DSA είναι συγκριτικά πολύ αργή. Ενδεχομένως το πιο σημαντικό ζήτημα που θα συνεχίζει να απασχολεί είναι η ύπαρξη δεύτερων προτύπων επικύρωσης, που είχε προκαλέσει δυσκολία σε προμηθευτές λογισμικού και υλικού, οι οποίοι είχαν ήδη στηριχθεί επάνω στην τεχνολογία του RSA. Τέλος, έγινε λόγος για τις συνθήκες κάτω από τις οποίες το NIST επέλεξε τον DSA, καθώς η διαδικασία χαρακτηρίσθηκε μυστικοπαθής και αυθαίρετη. Φυσικά η επιλογή ενός προτύπου και πόσο μάλλον οι λεπτομέρειες σχεδίασης του δεν μπορεί παρά μόνο να γίνεται κάτω από μυστικότητα και ασφάλεια.

4.2 Περιγραφή του DSA

Ο DSA είναι μια παραλλαγή των αλγορίθμων δημοσίου κλειδιού Schnorr και ElGamal. Ο αλγόριθμος χρησιμοποιεί έναν αριθμό από παραμέτρους. Η λειτουργία παραγωγής των κλειδιών αλλά και η ολοκλήρωση της διαχείρισης της ψηφιακής υπογραφής αναλύεται παρακάτω.

Δημόσιο κλειδί: y

p: τυχαίος πρώτος αριθμός μεγέθους 512-bit έως 1024-bit, όπου μπορεί να γνωστοποιηθεί σε μια ομάδα χρηστών.

q: σχετικά πρώτος αριθμός με την τιμή p-1, μεγέθους 160-bit. Μπορεί να γνωστοποιηθεί σε μια αμάδα χρηστών.

$g = h^{(p-1)/q} \bmod p$, όπου ο h έχει τιμή μικρότερη από το p-1. Μπορεί να γνωστοποιηθεί σε μια ομάδα χρηστών .

$y = g^x \bmod p$

Βλέπουμε πως το δημόσιο κλειδί παράγεται από τυχαίες τιμές πρώτων ή σχετικά πρώτων αριθμών. Το μέγεθος του δημόσιου κλειδιού y είναι ίσο με το μέγεθος του πρώτου αριθμού p (μεγέθους 512- bit έως 1024-bit).

Ιδιωτικό κλειδί: x
x < q : τυχαίος αριθμός μεγέθους 160-bit
Δημιουργία υπογραφής:
k: τυχαίος αριθμός, μικρότερος από την τιμή του q.
r (υπογραφή) = $(g^k \bmod p) \bmod q$.
s (υπογραφή) = (k-1 (H(m) + xr)) mod q.
Για την υπογραφή ενός μηνύματος m ακολουθείται η παραπάνω διαδικασία όπου H(m) είναι αυτό που ονομάζουμε σύνοψη του μηνύματος. Οι παράμετροι r και s που παράγονται είναι η ψηφιακή υπογραφή.

Παρατηρούμε πως η οντότητα που θα παράγει την μοναδική ψηφιακή υπογραφή της, έχει την δυνατότητα να το κάνει μόνο με την χρήση του ιδιωτικού κλειδιού x.

Επαλήθευση της υπογραφής:
$w = s^{-1} \bmod q$
u1 = (H(m) * w) mod q
u2 = (rw) mod q
$v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$
Εάν η τιμή v = r τότε η υπογραφή επαληθεύεται.

Ο παραλήπτης του μηνύματος ελέγχει την ορθότητα της υπογραφής υπολογίζοντας τις παραπάνω τιμές, έτσι ώστε να φτάσει στην ισότητα της επαλήθευσης (v = r). Οι παράμετροι r και s, ως ψηφιακή υπογραφή, χρησιμοποιούνται για την παραγωγή του συντελεστή v.

Στην DSA κρυπτογράφηση, η παραγωγή της υπογραφής είναι γρηγορότερη από την διαδικασία επαλήθευσης της. Με τον αλγόριθμο RSA αντίθετα, η επαλήθευση είναι κατά πολύ γρηγορότερη από την λειτουργία της δημιουργίας και χρήσης μιας ψηφιακής υπογραφής. Σίγουρα είναι θετικό το γεγονός ότι ο DSA κατά την διαδικασία της παραγωγής είναι πολύ γρήγορος, αλλά δεδομένου ότι σε πολλές εφαρμογές ένα κομμάτι των ψηφιακών πληροφοριών υπογράφεται μία

φορά, αλλά ελέγχεται συχνά, μπορούμε να πούμε ότι ο αλγόριθμος μειονεκτεί στα επιθυμητά επίπεδα ταχύτητας. Πολλοί ερευνητές θεωρούν ότι με βάση τις αναπτυσσόμενες τεχνικές, είναι εφικτό να βελτιωθεί η αποδοτικότητα του DSA μέσα στα επόμενα έτη.

4.3 Ασφάλεια του DSA

Η φιλοσοφία σχεδίασης του DSA είναι βασισμένη στην δομή υλοποίησης που είχαν προτείνει οι ερευνητές Schnorr και ElGamal. Γενικά, ο αλγόριθμος θεωρείται ασφαλής όταν το μέγεθος του δημόσιου κρυπτογραφικού κλειδιού είναι αρκετά μεγάλο. Το δημόσιο κλειδί έχει μέγεθος από 512 bits έως 1024 bits σε κάποια τιμή, πολλαπλάσια των 64 bits. Το κλιμακούμενο μέγεθος του εξαρτάται από την μέθοδο επιλογής και το αντίστοιχο μέγεθος του πρώτου αριθμού και συντελεστή p. Η 512-bits εφαρμογή του DSA δεν είναι αρκετά ισχυρή για μακροπρόθεσμη ασφάλεια. Η αρχική πρόταση του NIST ήταν αυτή, αλλά μετά από μια περίοδο έντονης κριτικής και αμφισβήτησης, το Ίδρυμα αναθεώρησε την άποψη του επιτρέποντας την χρήση κλειδιών μέχρι και 1024 bits μέγεθος.

Κάποιοι ερευνητές κατέγραψαν την ύπαρξη μια «καταπακτής» στον σχεδιασμό του DSA, η οποία δίνει την δυνατότητα σε κάποιον κρυπταναλυτή, έχοντας πρόσβαση μέσω αυτής, να ανακτήσει δεδομένα του μηνύματος χωρίς την γνώση του αποστολέα. «Καταπακτή» ονομάζουμε ένα ευαίσθητο σημείο στη δομή ενός αλγορίθμου που μπορεί κάποια τρίτη οντότητα να εκμεταλλευτεί μέσα από την κρυπτολογική ανάλυση. Το δεδομένο είναι ότι ο DSA δεν κρυπτογραφεί κάποια δεδομένα. Η χρήση του είναι οπωσδήποτε διαφορετική. Το πραγματικό, λοιπόν, ζήτημα είναι εάν η σχεδίαση του είναι ευαίσθητη στην δυνατότητα σφυρηλάτησης της ψηφιακής υπογραφής, που σίγουρα θα αποφέρει δυσφήμιση ολόκληρου του συστήματος. Αυτή η πιθανότητα αποφεύγεται εύκολα εάν ακολουθούνται οι κατάλληλες διαδικασίες παραγωγής των κρυπτογραφικών κλειδιών.

5 Σύγκριση των μη συμμετρικών αλγορίθμων RSA και DSA

5.1 Γενικά

Η σύγκριση των δυο μη συμμετρικών αλγορίθμων έγινε με βάση τον χρόνο απόκρισής τους στις εξής υποκατηγορίες:

- Δημιουργία ζεύγους κλειδιών (ιδιωτικού και δημοσίου)
- Δημιουργία ψηφιακής υπογραφής
- Επαλήθευση ψηφιακής υπογραφής

Δεν εξετάστηκε το θέμα του χρόνου απόκρισης στην υποκατηγορία κρυπτογράφησης και αποκρυπτογράφησης, δεδομένου ότι ο DSA δεν χρησιμοποιείται από μόνος του για κρυπτογράφηση και αποκρυπτογράφηση μικρών μηνυμάτων, αλλά σε συνδυασμό με άλλους αλγορίθμους, συνήθως τον Αλγόριθμο Diffie-Hellman. Θεωρήθηκε λοιπόν ότι τα δύο συστήματα κρυπτογράφησης με μη συμμετρικούς αλγορίθμους δεν είναι ισοδύναμα και δεν μπορούν να συγκριθούν με βάση τον χρόνο απόκρισής τους αφού η κρυπτογράφηση (αποκρυπτογράφηση) με τον RSA γίνεται αμέσως μετά την παραγωγή των κλειδιών, ενώ η κρυπτογράφηση (αποκρυπτογράφηση) με τον DSA απαιτεί πολλά μεταβατικά στάδια. Εξάλλου, θυμηθείτε ότι ο αλγόριθμος DSA, σχεδιάστηκε αποκλειστικά για την δημιουργία και επαλήθευση ψηφιακών υπογραφών και προτάθηκε ως γρηγορότερη εναλλακτική λύση ως προς τον προϋπάρχοντα αλγόριθμο RSA.

Για την σύγκριση των δύο αλγορίθμων χρησιμοποιήθηκαν κλάσεις στην γλώσσα προγραμματισμού Java και αναπτύχθηκαν τέσσερα προγράμματα:

- Δύο για την παραγωγή ζεύγους ισοδύναμου μήκους κλειδιών και την μέτρηση του χρόνου απόκρισης για τον RSA και DSA αντίστοιχα
- Δύο για την δημιουργία και επαλήθευση της ψηφιακής υπογραφής του ίδιου μικρού κειμένου και την μέτρηση του χρόνου απόκρισης για τον RSA και DSA αντίστοιχα

Τα προγράμματα υλοποιήθηκαν χρησιμοποιώντας τον editor Eclipse Java Neon και έτρεξαν όλα στο ίδιο υπολογιστικό σύστημα με τα εξής χαρακτηριστικά:

- Επεξεργαστής: Intel(R) Core(TM) i3-2310M CPU @ 2.10 GHz, 2.10 GHz, RAM=4.00 GB
- Λειτουργικό σύστημα: Windows 7, 64-bit

Για να είναι δίκαιη η σύγκριση κάθε πρόγραμμα εκτελέστηκε εκατό φορές, σημειώθηκε ο χρόνος απόκρισης του προγράμματος για κάθε εκτέλεση και υπολογίστηκε ο μέσος όρος της απόκρισης σε χιλιοστά του δευτερολέπτου.

5.2 Δημιουργία ζεύγους κλειδιών (ιδιωτικού και δημοσίου)

5.2.1 RSA

Ο κώδικας σε Java

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;

public class KeysRSA {

    public static void main(String[] args) throws
Exception {

        final long startTime =
System.currentTimeMillis();

        KeyPair pair = generateKeyPair();

        PublicKey publicKey= pair.getPublic();
        PrivateKey privateKey= pair.getPrivate();

        final long endTime =
System.currentTimeMillis();
```

```
        System.out.println("publicKey: " +
publicKey);
        System.out.println("privateKey: " +
privateKey);

        System.out.println("Total execution time: " +
(endTime - startTime) );

    }


    public static KeyPair generateKeyPair() throws
Exception {
        KeyPairGenerator generator =
KeyPairGenerator.getInstance("RSA");
        generator.initialize(2048, new
SecureRandom());
        KeyPair pair = generator.generateKeyPair();

        return pair;
    }

}
```

Το μήκος κλειδιού του παραγόμενου ζεύγους είναι 2048 bits.

Ο χρόνος απόκρισης μετριέται από την στιγμή που τρέχει η εντολή παραγωγής κλειδιών μέχρι που παράγεται (δεν εκτυπώνονται στην οθόνη) το ζεύγος των κλειδιών.

Ο μέσος όρος απόκρισης, μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **1.162 ms** (βλέπε παράρτημα Α, στήλη KeysRSA).

5.2.2 DSA

Ο κώδικας σε Java

```
import java.security.KeyPair;
```

```
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;

public class KeysDSA {

    public static void main(String[] args) throws
Exception {

        final long startTime =
System.currentTimeMillis();

        KeyPair pair = generateKeyPair();

        PublicKey publicKey= pair.getPublic();
        PrivateKey privateKey= pair.getPrivate();

        final long endTime =
System.currentTimeMillis();

        System.out.println("publicKey: " +
publicKey);
        System.out.println("privateKey: " +
privateKey);

        System.out.println("Total execution time: " +
(endTime - startTime) );

    }


    public static KeyPair generateKeyPair() throws
Exception {
        KeyPairGenerator generator =
KeyPairGenerator.getInstance("DSA");
        generator.initialize(2048, new
SecureRandom());
        KeyPair pair = generator.generateKeyPair();

        return pair;
    }
```

}
Το μήκος κλειδιού του παραγόμενου ζεύγους είναι 2048 bits.

Ο χρόνος απόκρισης μετριέται από την στιγμή που τρέχει η εντολή παραγωγής κλειδιών μέχρι που παράγεται (δεν εκτυπώνονται στην οθόνη) το ζεύγος των κλειδιών.

Ο μέσος όρος απόκρισης, μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **236 ms** (βλέπε παράρτημα Α, στήλη KeysDSA).

5.3 Δημιουργία και επαλήθευση ψηφιακής υπογραφής

5.3.1 RSA

Ο κώδικας σε Java

```
import static
java.nio.charset.StandardCharsets.UTF_8;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.util.Base64;


public class SignVerifyRSA {

    public static void main(String[] args) throws
Exception {

        KeyPair pair = generateKeyPair();
```

```
        String message = "ΤΕΙ Πελοποννήσου, Σχολή
Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών
Πληροφορικής";

        final long startTime =
System.currentTimeMillis();

        String signature = sign(message,
pair.getPrivate());

        final long endTimeSign =
System.currentTimeMillis();

        boolean isCorrect = verify(message,
signature, pair.getPublic());

        final long endTimeVerify =
System.currentTimeMillis();

        System.out.println("Υπογραφή: " + signature);

        System.out.println("Χρόνος Υπογραφής (RSA): "
+ (endTimeSign - startTime) );

        System.out.println("Επαλήθευση: " +
isCorrect);

        System.out.println("Χρόνος Επαλήθευσης (RSA):
" + (endTimeVerify - endTimeSign) );
    }

    public static KeyPair generateKeyPair() throws
Exception {
        KeyPairGenerator generator =
KeyPairGenerator.getInstance("RSA");
        generator.initialize(2048, new
SecureRandom());
        KeyPair pair = generator.generateKeyPair();

        return pair;
    }
```

```
    public static String sign(String plainText,
PrivateKey privateKey) throws Exception {
        Signature privateSignature =
Signature.getInstance("SHA256withRSA");
        privateSignature.initSign(privateKey);

privateSignature.update(plainText.getBytes(UTF_8));

        byte[] signature = privateSignature.sign();

        return
Base64.getEncoder().encodeToString(signature);
    }

    public static boolean verify(String plainText,
String signature, PublicKey publicKey) throws
Exception {
        Signature publicSignature =
Signature.getInstance("SHA256withRSA");
        publicSignature.initVerify(publicKey);

publicSignature.update(plainText.getBytes(UTF_8));

        byte[] signatureBytes =
Base64.getDecoder().decode(signature);

        return
publicSignature.verify(signatureBytes);
    }

}
```

Το μήκος κλειδιού του παραγόμενου ζεύγους είναι 2048 bits.

Το κείμενο που υπογράφεται είναι το «ΤΕΙ Πελοποννήσου, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών Πληροφορικής».

Η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η SHA256.

Ο χρόνος απόκρισης μετριέται ακριβώς πριν, και ακριβώς μετά από την εντολή δημιουργίας της υπογραφής. Ο χρόνος απόκρισης μετριέται

ακριβώς πριν, και ακριβώς μετά από την εντολή επαλήθευσης της υπογραφής. Η δημιουργία του ζεύγους κλειδιών έχει προηγηθεί.

Ο μέσος όρος απόκρισης δημιουργίας της υπογραφής , μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **33 ms** (βλέπε παράρτημα Α, στήλη SignRSA).

Ο μέσος όρος απόκρισης επαλήθευσης της υπογραφής , μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε μόλις **5 ms** (βλέπε παράρτημα Α, στήλη VerifyRSA).

5.3.2 DSA

Ο κώδικας σε Java

```
import static
java.nio.charset.StandardCharsets.UTF_8;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.security.Signature;
import java.util.Base64;


public class SignVerifyDSA {

    public static void main(String[] args) throws
Exception {

        KeyPair pair = generateKeyPair();

        String message = "ΤΕΙ Πελοποννήσου, Σχολή
Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών
Πληροφορικής";

        final long startTime =
System.currentTimeMillis();
```

```
        String signature = sign(message,
pair.getPrivate());

        final long endTimeSign =
System.currentTimeMillis();

        boolean isCorrect = verify(message,
signature, pair.getPublic());

        final long endTimeVerify =
System.currentTimeMillis();

        System.out.println("Υπογραφή: " + signature);

        System.out.println("Χρόνος Υπογραφής (DSA): "
+ (endTimeSign - startTime) );

        System.out.println("Επαλήθευση: " +
isCorrect);

        System.out.println("Χρόνος Επαλήθευσης (DSA):
" + (endTimeVerify - endTimeSign) );
    }

    public static KeyPair generateKeyPair() throws
Exception {
        KeyPairGenerator generator =
KeyPairGenerator.getInstance("DSA");
        generator.initialize(2048, new
SecureRandom());
        KeyPair pair = generator.generateKeyPair();

        return pair;
    }

    public static String sign(String plainText,
PrivateKey privateKey) throws Exception {
        Signature privateSignature =
Signature.getInstance("SHA256withDSA");
        privateSignature.initSign(privateKey);
```

```
privateSignature.update(plainText.getBytes(UTF_8));

        byte[] signature = privateSignature.sign();

          return
Base64.getEncoder().encodeToString(signature);
     }

     public static boolean verify(String plainText,
String signature, PublicKey publicKey) throws
Exception {
        Signature publicSignature =
Signature.getInstance("SHA256withDSA");
        publicSignature.initVerify(publicKey);

publicSignature.update(plainText.getBytes(UTF_8));

        byte[] signatureBytes =
Base64.getDecoder().decode(signature);

        return
publicSignature.verify(signatureBytes);
    }

}
```

Το μήκος κλειδιού του παραγόμενου ζεύγους είναι 2048 bits.

Το κείμενο που υπογράφεται είναι το «ΤΕΙ Πελοποννήσου, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών Πληροφορικής».

Η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η SHA256.

Ο χρόνος απόκρισης δημιουργίας μετριέται ακριβώς πριν, και ακριβώς μετά από την εντολή δημιουργίας της υπογραφής. Ο χρόνος απόκρισης επαλήθευσης μετριέται ακριβώς πριν, και ακριβώς μετά από την εντολή επαλήθευσης της υπογραφής. Η δημιουργία του ζεύγους κλειδιών έχει προηγηθεί.

Ο μέσος όρος απόκρισης δημιουργίας της υπογραφής , μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **21 ms** (βλέπε παράρτημα Α, στήλη SignDSA).

Ο μέσος όρος απόκρισης επαλήθευσης της υπογραφής , μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **22 ms** (βλέπε παράρτημα Α, στήλη VerifyDSA).

5.4 Συγκεντρωτικά αποτελέσματα

Πίνακας 1: Χρόνος απόκρισης προγραμμάτων σε msec

(μέσος όρος μετά από 100 εκτελέσεις)

KeysRSA	KeysDSA	SignRSA	VerifyRSA	SignDSA	VerifyDSA
1.143	236	34	5	21	22

Από τον παρακάτω πίνακα προκύπτουν τα ακόλουθα

- Στην παραγωγή κλειδιών ο DSA είναι κατά **382,4%** ταχύτερος από τον RSA
- Στην δημιουργία ψηφιακής υπογραφής ο DSA είναι κατά **61,38%** ταχύτερος από τον RSA
- Αντίθετα στην επαλήθευση ψηφιακής υπογραφής ο RSA είναι κατά **78.69%** ταχύτερος από τον DSA

6 Συμπεράσματα

6.1 Σύγκριση RSA με DSA

Στην κρυπτογραφία και στους αλγόριθμους κρυπτογράφησης οι έννοιες των μη συμμετρικών αλγορίθμων RSA and DSA εμφανίζονται συχνά. Και οι δύο χρησιμοποιούνται κατά κόρον σε θέματα ασφάλειας με πολύ καλά αποτελέσματα. Παρ' όλα αυτά υπάρχουν μεταξύ τους αρκετές διαφορές όπως αυτές που αναφέρονται παρακάτω.

Ο αλγόριθμος DSA αναφέρεται στα αρχικά της ονομασίας «Digital Signature Algorithm». Ο αλγόριθμος RSA, από την άλλη πλευρά αναφέρεται στα αρχικά των ονομάτων των ανθρώπων που τον δημιούργησαν, συγκεκριμένα στους Ron Rivest, Adi Shamir, και Leonard Adleman. Ο RSA σχεδιάστηκε για να είναι ένας αλγόριθμος κρυπτογράφησης. Ο DSA αναπτύχθηκε από την NSA (National Security Agency) για να χρησιμοποιηθεί από την κυβέρνηση των ΗΠΑ ως standard για ψηφιακές υπογραφές. Ο DSA στηρίχτηκε κυρίως στον αλγόριθμο «ElGamal Signature Algorithm» από το οποίο δανείστηκε πολλές μαθηματικές τεχνικές. Ο RSA, από την άλλη πλευρά στηρίζεται στην δυσκολία της παραγοντοποίησης πρώτων αριθμών, σαν την κύρια τεχνική του.

Το ίδιο το όνομα DSA υποδηλώνει την κύρια λειτουργία του. Είναι ένα πρόγραμμα που σχεδιάστηκε κυρίως για τη δημιουργία υπογραφής, και επομένως είναι ιδιαίτερα δημοφιλές όταν χρειάζεται κάποιος να υπογράψει ψηφιακά ένα κείμενο. Παρόλη τη δημοφιλία της η τεχνική αυτή δεν μπορεί να επεκταθεί παραπέρα. Ο RSA, από την άλλη πλευρά, καλύπτει θέματα υπογραφής αλλά και επιπλέον θέματα κρυπτογράφησης - αποκρυπτογράφησης.

Σαν αποτέλεσμα όταν κάποιος ενδιαφέρεται για ψηφιακές υπογραφές και μόνον, ο DSA υπερτερεί γιατί και γρηγορότερα παράγει τα κλειδιά του αλλά και γρηγορότερα δημιουργεί την ψηφιακή υπογραφή.

Από την άλλη πλευρά όσον αφορά στην επαλήθευση της ψηφιακής υπογραφής ο RSA σαφώς υπερτερεί του DSA. Αυτό οφείλεται κυρίως στο γεγονός ότι ο RSA αποκρυπτογραφεί ένα κείμενο πολύ γρηγορότερα από ότι το κρυπτογραφεί (βλέπε απόδειξη με πρόγραμμα σε Java, στο Παράρτημα Β).

Ένας βέλτιστος συνδυασμός που να χρησιμοποιεί και τις δύο προσεγγίσεις, είναι δυνατόν να βρεθεί κάθε φορά, ad hoc, ανάλογα με το πρόβλημα που εμφανίζεται. Και οι δύο προσεγγίσεις έχουν παρόμοιο κρυπτογραφικά πλεονεκτήματα και μπορούν να χρησιμοποιηθούν εξίσου καλά σε θέματα ασφάλειας σε περιβάλλοντα server – client.

Σε γενικές γραμμές μπορεί να ειπωθεί ότι ο DSA είναι καταλληλότερος για θέματα υπογραφής, ενώ για επαλήθευση και κρυπτογράφηση καταλληλότερος είναι ο RSA.

6.2 Σύνοψη

- Ο DSA και ο RSA είναι δύο περίπου ισοδύναμοι μη συμμετρικοί αλγόριθμοι κρυπτογράφησης
- Ο DSA είναι πιο γρήγορος κατά τη δημιουργία κλειδιών από τον RSA
- Ο RSA από την άλλη πλευρά είναι σχεδιασμένος και για κρυπτογράφηση ενώ ο DSA δεν είναι
- Ο DSA είναι πιο γρήγορος κατά την δημιουργία ψηφιακής υπογραφής

- Κατά την επαλήθευση υπογραφής, ο RSA είναι ταχύτερος, κυρίως λόγω της εξαιρετικής ικανότητας του στην αποκρυπτογράφηση

Εάν χρειάζεται κανείς ψηφιακή υπογραφή, ο DSA είναι ο αλγόριθμος που πρέπει να επιλέξει.

Για την επαλήθευση της ψηφιακής υπογραφής, η RSA είναι η καλύτερη επιλογή.

Οι DSA και RSA έχουν περίπου ίσες δυνατότητες σε θέματα ασφαλείας και κάθε φορά πρέπει να επιλεγεί η λύση εκείνη που καταναλώνει του λιγότερους πόρους από το σύστημα.

ΠΑΡΑΡΤΗΜΑ Α

Πίνακας Αποτελεσμάτων

Χρόνος απόκρισης προγραμμάτων σε msec

(μέσος όρος μετά από 100 εκτελέσεις)

Tries	KeysRSA	KeysDSA	SignRSA	VerifyRSA	SignDSA	VerifyDSA		Encryption RSA	Decryption RSA
1	2.423	473	32	5	33	18		677	43
2	2.158	452	25	4	19	20		677	16
3	1.086	484	35	5	18	20		677	20
4	865	470	23	4	16	21		677	17
5	1.903	462	32	3	22	29		677	19
6	870	458	39	5	21	24		677	23
7	952	488	33	4	27	20		677	17
8	866	468	31	4	26	21		677	17
9	1.553	416	34	5	33	31		677	20
10	908	277	21	3	22	16		677	20
11	2.108	262	28	3	21	16		677	29
12	790	267	45	5	20	22		677	17
13	1.118	261	28	4	27	29		677	16
14	1.017	257	37	4	25	35		677	17
15	864	260	28	5	23	34		677	22
16	1.120	268	36	7	21	20		480	15
17	1.388	277	43	3	21	23		479	16
18	2.250	270	27	4	17	15		458	16
19	569	258	33	5	20	26		465	20
20	1.008	260	44	5	24	20		505	18
21	1.224	263	29	3	20	25		472	18
22	911	269	37	4	24	22		485	16
23	1.184	264	26	8	16	28		477	19
24	1.365	275	31	5	18	17		475	16
25	649	273	35	5	18	30		477	18
26	1.256	263	43	4	23	21		500	17
27	990	258	25	5	25	47		486	19
28	784	261	37	4	18	13		493	19
29	738	264	37	5	20	31		478	37
30	922	261	44	7	24	22		543	15
31	1.247	261	29	3	23	25		476	17
32	840	262	30	4	20	20		504	17
33	864	258	63	7	21	19		458	15
34	615	262	35	5	20	15		485	17
35	706	257	37	6	19	26		505	17
36	1.215	248	34	5	19	14		474	28
37	849	253	29	5	16	17		476	26
38	1.609	253	32	4	18	23		490	19

39	1.001	243	37	4	26	14		563	19
40	968	252	36	6	17	15		476	16
41	829	255	31	5	18	25		518	23
42	598	256	33	5	23	20		495	20
43	1.073	260	40	8	19	22		516	15
44	600	249	30	7	18	45		497	20
45	803	260	30	5	21	15		491	15
46	1.331	257	31	4	19	26		489	15
47	983	263	34	5	21	23		557	20
48	1.464	262	35	5	22	21		497	18
49	2.188	246	37	5	23	23		530	22
50	741	252	35	5	21	23		523	15
51	791	240	35	7	23	21		505	18
52	1.307	245	38	5	18	15		492	20
53	610	280	37	4	25	29		488	24
54	1.127	250	30	6	24	20		531	20
55	906	248	21	4	23	25		484	18
56	757	246	24	5	20	16		492	16
57	1.776	256	29	3	24	21		487	16
58	1.730	249	36	5	20	21		514	22
59	689	243	27	4	20	23		481	18
60	966	248	28	4	23	23		509	18
61	589	250	32	4	25	15		503	16
62	1.351	252	39	4	17	33		479	18
63	947	244	34	5	21	20		481	22
64	969	249	34	4	19	21		485	22
65	1.394	242	32	5	19	17		471	20
66	850	249	30	3	16	18		485	15
67	919	252	35	5	19	21		538	27
68	1.485	246	39	5	22	22		539	26
69	1.435	261	28	4	19	29		484	18
70	1.273	250	37	5	18	42		467	20
71	691	246	36	4	21	22		492	17
72	1.103	247	32	5	19	25		470	26
73	961	249	27	7	21	24		485	16
74	1.081	247	37	5	24	19		485	18
75	1.111	247	29	5	20	21		470	15
76	1.625	255	55	4	21	21		489	23
77	838	249	32	4	22	22		490	15
78	826	247	30	4	20	13		515	21
79	1.270	254	42	4	18	14		491	17
80	1.376	248	36	5	17	19		485	16
81	669	254	33	5	19	27		491	22
82	1.660	283	28	5	19	28		489	17

83	1.695	266	35	6	21	22		496	17
84	1.882	248	33	4	20	15		500	17
85	1.070	252	31	5	20	17		506	19
86	1.533	248	32	5	18	14		528	18
87	507	248	32	3	21	20		512	24
88	1.218	239	31	4	17	48		512	22
89	998	254	22	4	21	15		510	17
90	973	248	29	6	21	20		498	18
91	1.022	243	35	4	17	19		524	20
92	1.399	251	29	5	22	17		579	18
93	1.327	247	29	4	20	21		483	17
94	660	246	47	6	19	14		502	20
95	3.332	249	32	4	17	28		531	17
96	1.932	249	34	5	22	16		505	19
97	752	250	32	7	23	21		516	17
98	1.126	249	31	4	24	22		537	18
99	1.251	252	40	6	18	15		504	23
100	2.093	255	26	4	20	21		518	20
Average	**1.162**	**236**	**33**	**5**	**21**	**22**		**525**	**19**

ΠΑΡΑΡΤΗΜΑ Β

Κρυπτογράφηση και αποκρυπτογράφηση με τον RSA

Πρόγραμμα σε Java

```
import static
java.nio.charset.StandardCharsets.UTF_8;

import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.SecureRandom;
import java.util.Base64;

import javax.crypto.Cipher;

public class EncryptDecryptRSA {

    public static void main(String[] args) throws
Exception {

        KeyPair pair = generateKeyPair();

        String message = "ΤΕΙ Πελοποννήσου, Σχολή
Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών
Πληροφορικής";

        final long startTime =
System.currentTimeMillis();

        String cipherText = encrypt(message,
pair.getPublic());

        final long endTimeEncrypt =
System.currentTimeMillis();

        String decipheredMessage =
decrypt(cipherText, pair.getPrivate());

        final long endTimeDecrypt =
System.currentTimeMillis();

        System.out.println("Κρυπτογράφημα: " +
cipherText);
```

```
        System.out.println("Χρόνος Κρυπτογράφησης
(RSA): " + (endTimeEncrypt - startTime) );

System.out.println("Αποκρυπτογράφημα: " +
decipheredMessage);

        System.out.println("Χρόνος Αποκρυπτογράφησης
(RSA): " + (endTimeDecrypt - endTimeEncrypt) );
    }

    public static KeyPair generateKeyPair() throws
Exception {
        KeyPairGenerator generator =
KeyPairGenerator.getInstance("RSA");
        generator.initialize(2048, new
SecureRandom());
        KeyPair pair = generator.generateKeyPair();

        return pair;
    }

    public static String encrypt(String plainText,
PublicKey publicKey) throws Exception {
        Cipher encryptCipher =
Cipher.getInstance("RSA");
        encryptCipher.init(Cipher.ENCRYPT_MODE,
publicKey);

        byte[] cipherText =
encryptCipher.doFinal(plainText.getBytes(UTF_8));

        return
Base64.getEncoder().encodeToString(cipherText);
    }

    public static String decrypt(String cipherText,
PrivateKey privateKey) throws Exception {
        byte[] bytes =
Base64.getDecoder().decode(cipherText);
```

```
        Cipher decriptCipher =
Cipher.getInstance("RSA");
        decryptCipher.init(Cipher.DECRYPT_MODE,
privateKey);

        return new
String(decryptCipher.doFinal(bytes), UTF_8);
    }


}
```

Το μήκος κλειδιού του παραγόμενου ζεύγους είναι 2048 bits.

Το κείμενο που κρυπτογραφείται και αποκρυπτογραφείται είναι το «ΤΕΙ Πελοποννήσου, Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Μηχανικών Πληροφορικής».

Ο χρόνος απόκρισης κρυπτογράφησης μετριέται ακριβώς πριν, και ακριβώς μετά από την εντολή δημιουργίας του κρυπτογραφήματος υπογραφής. Ο χρόνος απόκρισης αποκρυπτογράφησης μετριέται ακριβώς πριν, και ακριβώς μετά από την εντολή αποκρυπτογράφησης του κρυπτογραφήματος. Η δημιουργία του ζεύγους κλειδιών έχει προηγηθεί.

Ο μέσος όρος απόκρισης για κρυπτογράφηση, μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε **525 ms** (βλέπε Παράρτημα Α, στήλη Encryption RSA).

Ο μέσος όρος απόκρισης για αποκρυπτογράφηση, μετά από εκατό εκτελέσεις του προγράμματος, ανέρχεται σε μόλις **19 ms** (βλέπε Παράρτημα Α, στήλη Decryption RSA).

Βιβλιογραφία

- «Ασφάλεια της Πληροφορίας», Α. Σουρής, Δ. Πατσός, Ν. Γρηγοριάδης, Εκδόσεις Νέων Τεχνολογιών, 2004
- «Βασικές Αρχές Ασφάλειας Δικτύων», W. Stallings, Εκδόσεις Κλειδάριθμος, 2008
- «Κρυπτογραφία & Εφαρμογές», Κ. Πατσάκης, Ε. Φούντας, Εκδόσεις Βαρβαρήγου, 2009
- «Κρυπτογραφία, η Επιστήμη της Ασφαλούς Επικοινωνίας», Δ. Πουλάκης, Εκδόσεις Ζήτη, 2004
- «Ασφάλεια Πληροφοριακών Συστημάτων», Σ. Κάτσικας, Δ. Γκρίτζαλης, Σ. Γκρίτζαλης, Εκδόσεις Νέων Τεχνολογιών, 2004
- «Σύγχρονη Κρυπτογραφία», Π. Νάστου, Π. Σπυράκης, Γ. Σταματίου, Εκδόσεις Ελληνικά Γράμματα, 2003
- «Τεχνικές Κρυπτογραφίας & Κρυπτανάλυσης», Β. Κάτος, Γ. Στεφανίδης, 2003
- «Εφαρμογή Σύγχρονων Τεχνικών Κρυπτογράφησης σε Σήματα μιάς ή Περισσοτέρων Διαστάσεων», Κ. Ασκητόπουλος, Γ. Καραγιάννης, ΤΕΙ Μεσολογγίου, 2004
- «Εφαρμογές της Κρυπτογραφίας», Κ. Αποστολίδου, Διπλωματική εργασία, ΤΕΙ Δυτικής Μακεδονίας, 2008