Ανοικτό Πανεπιστήμιο Κύπρου
Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Ασφάλεια Υπολογιστών και Δικτύων*

Μεταπτυχιακή Διατριβή

Μετα-κβαντική Κρυπτογραφία : Ψηφιακές Υπογραφές

Χρήστος Ζησιμόπουλος

Επιβλέπων Καθηγητής
Κωνσταντίνος Λιμνιώτης

Μάιος 2020

Ανοικτό Πανεπιστήμιο Κύπρου
Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Ασφάλεια Υπολογιστών και Δικτύων*

Μεταπτυχιακή Διατριβή

Μετα-κβαντική Κρυπτογραφία: Ψηφιακές Υπογραφές

Χρήστος Ζησιμόπουλος

Επιβλέπων Καθηγητής
Κωνσταντίνος Λιμνιώτης

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για
απόκτηση μεταπτυχιακού τίτλου σπουδών
στην Ασφάλεια Υπολογιστών και Δικτύων
από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2020

Περίληψη

Η μελέτη των κρυπτογραφικών αλγόριθμων δημοσίου κλειδιού οι οποίοι θα είναι ασφαλείς και μετά την έλευση των κβαντικών υπολογιστών, αποτελεί σημαντικό ερευνητικό πεδίο, καθώς οι σημερινοί αλγόριθμοι δημοσίου κλειδιού θα είναι επισφαλείς στο νέο τεχνολογικό περιβάλλον. Αλγόριθμοι οι οποίοι θα είναι ασφαλείς ακόμα και μετά την έλευση κβαντικών υπολογιστών ανήκουν στο χώρο της λεγόμενης μετα-κβαντικής κρυπτογραφίας, ωστόσο προκύπτουν συχνά θέματα στην υλοποίησή τους ή/και στην απόδοσή τους.

Στόχος της παρούσας διατριβής, είναι να μελετήσει, τα κρυπτογραφικά σχήματα ψηφιακών υπογραφών, στην μετα-κβαντική κρυπτογραφία. Ιδιαίτερη έμφαση θα δοθεί σε σχήματα που βασίζονται σε κρυπτογραφικές συναρτήσεις κατακερματισμού, όπως τα σχήματα ψηφιακών υπογραφών μιας χρήσης των Lamport, Merkle, W-OTS+, Merkle Trees και XMSS. Τα σχήματα αυτά θα αναλυθούν ως προς τον τρόπο λειτουργίας τους, τα χαρακτηριστικά ασφαλείας τους και θα αξιολογηθούν ως προς την απόδοσή τους σε σημερινά συμβατικά υπολογιστικά συστήματα. Για την αξιολόγηση και συγκριτική αποτίμηση της απόδοσής τους, χρησιμοποιούμε τη προγραμματιστική βιβλιοθήκη Bouncy Castle για την υλοποίηση ορισμένων σεναρίων δοκιμών, για τη σύγκριση ενός συμβατικού αλγορίθμου (RSA ψηφιακής υπογραφής), για διάφορες παραμέτρους, και ενός γνωστού αλγορίθμου μετα-κβαντικής κρυπτογραφίας για ψηφιακή υπογραφή (XMSS).

Από τα αποτελέσματα των σεναρίων που πραγματοποιήθηκαν, συμπεραίνουμε ότι όσο μεγαλύτερη είναι η επεξεργαστική ισχύς του υπολογιστή, τόσο μικρότερος είναι ο χρόνος εκτέλεσης των αλγόριθμων που δοκιμάσαμε (όπως εξάλλου αναμενόταν). Ο αλγόριθμος RSA είναι ταχύτερος από τον αλγόριθμο XMSS. Το μέγεθος της υπογραφής, επηρεάζεται στον αλγόριθμο RSA, μόνο από το μήκος των κλειδιών, ενώ στον αλγόριθμο XMSS, επηρεάζεται από το αριθμό εξυπηρέτησης υπογραφών στο σχήμα (μεγαλύτερο ύψος) και την συνάρτηση κατακερματισμού που χρησιμοποιεί ο αλγόριθμος. Συμπερασματικά, οι αλγόριθμοι μετα-κβαντικής κρυπτογραφίας ψηφιακών υπογραφών παραμένουν σαφώς πιο αργοί, ωστόσο προκύπτει ότι είναι ρεαλιστικά υλοποιήσιμοι.

Summary

The study of public key cryptographic algorithms, that will remain secure after the advent of quantum computing is an important research field, as today's public key algorithms will be insecure in this new technological era. Cryptographic algorithms that will be secure after the advent of quantum computing lie in the so-called class of post-quantum cryptography; however, there are still some issues in their implementation and/or their performance.

The aim of this thesis is to study the cryptographic schemes of digital signatures in post-quantum cryptography. Special attention will be paid to one-time signature schemes based on the cryptographic hash functions, such as Lamport, Merkle, W-OTS+, Merkle Trees and XMSS. These schemes will be analyzed with respect to the security features work and will be evaluated with regard to their performance. To achieve this goal, we use the Bouncy Castle programming library to implement some test scenarios for comparing a conventional digital signature algorithm (RSA) and one of the post-quantum hashing schemes (XMSS). Comparing the performance of the two algorithms with different key sizes, underlying hash functions in two different environments with different computing powers computing, we observe that RSA is always faster. Moreover, as it is expected, the underlying hardware is important in terms of efficiency. The performance of the RSA signature scheme mainly rests with the key size, whereas the XMSS algorithm depends on the height of the corresponding tree (which in turn determines the number of possible distinct one-time signatures that can be created), as well as on the underlying hash function. Concluding, post-quantum hash-based signature schemes are indeed slower that conventional digital signatures; however, even today they can be implemented in conventional computing systems.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κωνσταντίνο Λιμνιώτη που με την καθοδήγησή του συνέβαλε τα μέγιστα για την ολοκλήρωση της διατριβής.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η κρυπτογραφία έχει χρησιμοποιηθεί εδώ και χιλιάδες χρόνια για τη μετατροπή της πληροφορίας (μηνυμάτων), από μια κανονική, κατανοητή μορφή σε μια ακατάληπτη, που χωρίς τη γνώση του κρυφού μετασχηματισμού θα παρέμενε ακατανόητη. Η κρυπτογραφία έχει σαν στόχο την ασφάλεια της πληροφορίας, που είναι σημαντικό για την επικοινωνία μεταξύ δυο η περισσοτέρων ακρών, που αυτά μπορεί να είναι άνθρωποι, υπολογιστές, προγράμματα κ.α. Η εμπιστευτικότητα της πληροφορίας επιτυγχάνεται μέσω της κρυπτογραφίας, που παρέχει, την εμπιστευτικότητα, την ακεραιότητα των δεδομένων, την πιστοποίηση της γνησιότητας χρηστών και πληροφοριών και την μη αποποίηση της αυθεντικότητας του τι έχει συμβεί.

Οι παλαιότερες μορφές κρυπτογράφησης, χρησιμοποιούσαν για την μετατροπή της πληροφορίας, την γλωσσική δομή του μηνύματος, ενώ οι νεότερες μορφές κρυπτογράφησης χρησιμοποιούν μαθηματικά προβλήματα που είναι δύσκολο να επιλυθούν. Οι εφαρμογές της κρυπτογραφίας στο σύγχρονο κόσμο ολοένα αυξάνεται καθιστώντας πλέον αξιόπιστη τη μεταφορά της πληροφορίας για διάφορους λειτουργικούς σκοπούς όπως: σε τραπεζικά δίκτυα – ebanking, ATM, στην τηλεφωνία - κινητή και σταθερή, στην διακίνηση πληροφοριών σε εταιρείες, σε στρατιωτικά/διπλωματικά δίκτυα, στην ηλεκτρονική ψηφοφορία, στο ηλεκτρονικό εμπόριο, στο ηλεκτρονικό ταχυδρομείο, στα συστήματα βιομετρικής αναγνώρισης, στα ιδιωτικά

δίκτυα (VPN), στο διαδίκτυο/ διαδίκτυο πραγμάτων (IoT) , στην τηλεφωνία μέσω διαδικτύου, στην δορυφορική/ καλωδιακή τηλεόραση, στα ασύρματα δίκτυα κ.α.

Ωστόσο, ο κβαντικός υπολογισμός ή η χρήση κβαντικών μηχανικών φαινομένων για την αναπαραγωγή και τον χειρισμό πληροφοριών, υπόσχεται να είναι μια τεχνολογία που θα αλλάξει την ιστορία της επιστήμης των υπολογιστών, όταν θα υλοποιείται πλήρως σε ευρεία κλίμακα. Πολλά προβλήματα που θεωρούνται πλέον πολύπλοκα για τους συμβατικούς υπολογιστές, ακόμα και τους πιο ισχυρούς υπερυπολογιστές, θα μπορούσαν να υπολογιστούν σε λεπτά ή δευτερόλεπτα αξιοποιώντας τις ιδιότητες της κβαντικής φυσικής (π.χ. εμπλοκή, υπέρθεση) για να εκμαιεύσουν πληροφορίες. Οι αρχικές εφαρμογές της κβαντικής πληροφορικής περιλαμβάνουν την προσομοίωση σύνθετων μοριακών συστημάτων στη χημεία και την επιστήμη των υλικών, την μηχανική μάθηση για την ταξινόμηση δεδομένων με μεγάλες διαστάσεις και τα προβλήματα βελτιστοποίησης σε ένα εξαιρετικά μεγάλο χώρο πιθανών λύσεων.

Ενώ ο κβαντικός υπολογισμός δημιουργεί ένα εντελώς νέο πρότυπο για την επίλυση πολύπλοκων προβλημάτων πληροφορικής, δυστυχώς αποτελεί επίσης ένα ισχυρό νέο εργαλείο για την επίθεση στους υφιστάμενους αλγόριθμους κρυπτογραφίας. Αυτό συνιστά μια σημαντική απειλή για την ασφάλεια της πληροφορίας, όπως την γνωρίζουμε σήμερα. Πιο συγκεκριμένα, η κρυπτογραφία δημόσιου κλειδιού (ασύμμετρη), που θα μελετήσουμε παρακάτω, βασίζεται σε μαθηματικές συναρτήσεις που επιτρέπουν τον εύκολο υπολογισμό, ενός δημόσιου κλειδιού από ένα ιδιωτικό κλειδί, αλλά κάνουν τον υπολογισμό ενός ιδιωτικού κλειδιού από ένα δημόσιο κλειδί είναι υπολογιστικά ανέφικτο.

Οι ευρέως χρησιμοποιούμενες συναρτήσεις στους σημερινούς αλγορίθμους δημοσίου κλειδιού βασίζονται, είτε στη δυσκολία της παραγοντοποίησης μεγάλων ακέραιων αριθμών, όπως στον αλγόριθμο του RSA, είτε και των παραλλαγών του λεγόμενου προβλήματος διακριτού λογαρίθμου, για την περίπτωση των ελλειπτικών καμπυλών. Και οι δύο αυτοί τύποι συναρτήσεων δεν έχουν γνωστή λύση για τον υπολογισμό ενός αντιστρόφου σε πολυωνυμικό χρόνο κάνοντας συμβατικούς υπολογισμούς. Στην κρυπτογραφία συμμετρικού κλειδιού, η ασφάλεια ενός κλειδιού που μοιράζεται μεταξύ δύο μερών εξαρτάται από το πόσο δύσκολο είναι το τυχαίο κλειδί να το υποθέσει κάποιος επιτιθέμενος. Εάν η τιμή δεν μπορεί να προσδιοριστεί άμεσα με κρυπτογράφηση, ο επιτιθέμενος μπορεί να εφαρμόσει μεθόδους αναζήτησης για να εξετάσει το χώρο των πιθανών κλειδιών αναζητώντας τη σωστή τιμή. Όμως, δεδομένου ενός επαρκώς μεγάλου χώρου πιθανών τιμών, η εύρεση ενός κλειδιού δεν είναι υπολογιστικά εφικτή

για το παράθυρο του χρόνου κατά το οποίο το πρόγραμμα χρησιμοποιείται για την προστασία των δεδομένων.

Το 1994, ο Peter Shor έδειξε πως ένας κβαντικός υπολογιστής (QC) θα μπορούσε να χρησιμοποιηθεί για να εκτελέσει την ακέραιη παραγοντοποίηση σε πολυωνυμικό χρόνο (πολυώνυμο σε λογαριθμικό N σε ακέραιο μέγεθος N) χρησιμοποιώντας σπονδυλωτή επέκταση (modular expansion) και ένα κβαντικό μετασχηματισμό Fourier που σχεδίασε (Shor 1997). Ο αλγόριθμος Shor, όπως ονομάζεται τώρα, έχει δειχθεί ότι γενικεύει επίσης την επίλυση των προβλημάτων διακριτού λογαρίθμου και ελλειπτικής καμπύλης διακριτών λογαρίθμων σε πολυωνυμικό χρόνο (Gerjuoy 2005: 73).

Για έναν εισβολέα με ένα επαρκώς ισχυρό QC, γίνεται σαφές ότι θα είναι σε θέση να «καταρρίψει» αποτελεσματικά την ασφάλεια, των κλασικών αλγορίθμων δημοσίου κλειδιού που χρησιμοποιούνται ευρέως μέχρι και σήμερα, αφού οι βασικές μαθηματικές ιδιότητες στις οποίες οι αλγόριθμοι στηρίζουν την ασφάλειά τους σήμερα δεν θα είναι πλέον επαρκείς να παρέχουν ασφάλεια.. Δηλαδή, ένας επιτιθέμενος θα μπορούσε να χρησιμοποιήσει ένα QC για να αποκτήσει ιδιωτικά κρυπτογραφικά κλειδιά από δημόσια κλειδιά γρήγορα και αποτελεσματικά. Το 1996, ο Lov Kumar Grover έδειξε επίσης ότι τα QCs θα μπορούσαν να χρησιμοποιηθούν για την επίλυση του προβλήματος της γραμμικής αναζήτησης σε ένα διαχωρισμένο N-στοιχείο χώρου σε λειτουργίες O (√N) χρησιμοποιώντας έναν ειδικό χειριστή διάχυσης που ανέπτυξε (Grover 1996).

Για τους αντιπάλους με QC, ο αλγόριθμος του Grover ουσιαστικά έρχεται να «πλήξει» την ασφάλεια των συμμετρικών αλγορίθμων, προτείνοντας έναν πιο αποτελεσματικό τρόπο αναζήτησης στο χώρο των πιθανών κλειδιών για να αποκτήσει το μυστικό κλειδί. Το 1999, οι Gilles Brassard et al. έδειξαν ότι τα QCs θα μπορούσαν να χρησιμοποιηθούν για την επίλυση του προβλήματος της εύρεσης συγκρούσεων σε κρυπτογραφικές συναρτήσεις κατακερματισμού (hash functions) σε λειτουργίες O (3√N) χρησιμοποιώντας τον αλγόριθμο του Grover (Brassard, Hoyer και Tapp 1998).

Οι συνέπειες αυτών των εκπληκτικών αποτελεσμάτων στην δημόσια και συμμετρική κρυπτογράφηση κλειδιών είναι καλά τεκμηριωμένες. Τόσο η έκθεση του οργανισμού NAD / CSS IAD [1] τον Ιανουάριο του 2016 (NSA/CSS Information Assurance Directorate 2016), όσο και η έκθεση NIST για την Μετα-κβαντική Κρυπτογραφία (Chen, Jordan, Liu, Moody, Peralta, Perlner

[1] Commercial National Security Algorithm Suite and Quantum Computing FAQ

και Smith-Tone 2016) τον Απρίλιο του 2016 υποδηλώνουν την ανάγκη για νέα πρότυπα αντικατάστασης κρυπτοσυστημάτων με βάση την παραγοντοποίηση ακέραιων αριθμών και τα διακριτά προβλήματα λογαρίθμου. Αυτό περιλαμβάνει την αντικατάσταση των ευρέως χρησιμοποιούμενων κρυπτοσυστημάτων RSA, ECDSA, ECDH και DSA με εναλλακτικές λύσεις μετά την κβαντική κρυπτογράφηση (PQC).

1.1 Σκοπός έρευνας και βασικά ερευνητικά ερωτήματα

Η παρούσα έρευνα αποσκοπεί στη μελέτη κρυπτογραφικών αλγορίθμων οι οποίοι θα μπορούν να είναι ανθεκτικοί μετά την έλευση των κβαντικών υπολογιστών. Αναλυτικότερα θα εξεταστούν κρυπτογραφικά σχήματα ψηφιακών υπογραφών, τα οποία εντάσσονται στο χώρο της μετα-κβαντικής κρυπτογραφίας, βασισμένα σε κρυπτογραφικές συναρτήσεις κατακερματισμού. Τέλος, η αξιολόγηση των χαρακτηριστικών ασφαλείας τους και η απόδοσή τους αποτελούν κριτήριο, για την υιοθέτησή τους μετά την έλευση των κβαντικών υπολογιστών. Άλλωστε, η μελέτη της απόδοσης τέτοιων σχημάτων σε σημερινά υπολογιστικά συστήματα έχει ιδιαίτερη αξία, λαμβάνοντας υπόψη ότι αναμένεται να υπάρξει μία «ενδιάμεση» περίοδος όπου αλγόριθμοι μετα-κβαντικής κρυπτογραφίας θα αρχίσουν να υλοποιούνται με μη κβαντικά συστήματα, προκειμένου η μετάβαση σε κβαντικά να μην απαιτεί αμέσως την αλλαγή όλων των υποκείμενων υλοποιήσεων.

Στην παρούσα διατριβή μελετώνται ορισμένα ερευνητικά ερωτήματα τα οποία θα γίνει προσπάθεια να διερευνηθούν και να απαντηθούν. Τα ερωτήματα αυτά είναι:

-Τι διαφορές υπάρχουν, αναφορικά με τη σχεδίαση, ανάμεσα στους σημερινούς κρυπτογραφικούς αλγόριθμους και στους μετα-κβαντικούς κρυπτογραφικούς αλγόριθμους;

-Είναι εφικτό ένας μετα-κβαντικός αλγόριθμος κρυπτογράφησης ψηφιακής υπογραφής να εκτελεστεί σε έναν υπάρχοντα συμβατικό ηλεκτρονικό υπολογιστή; Ποιοι άλλοι περιορισμοί ενδεχομένως ανακύπτουν (π.χ. πόσο επιδρούν παράμετροι όπως υπολογιστική ισχύς, μνήμη κτλ. στην απόδοση ενός μετα-κβαντικού αλγορίθμου ψηφιακής υπογραφής;)

-Πόσο μεγαλύτερη χρονική καθυστέρηση υπεισέρχεται με τη χρήση υπογραφής μετα-κβαντικής κρυπτογραφίας σε σημερινό συμβατικό υπολογιστή;

1.2 Αναγκαιότητα και σπουδαιότητα έρευνας

Η ανάπτυξη κρυπτογραφικών αλγορίθμων, που θα είναι ανθεκτικοί και μετά την έλευση κβαντικών υπολογιστών, αποτελεί πολύ σημαντικό ερευνητικό πεδίο, λόγω του ότι οι σημερινοί αλγόριθμοι (που βασίζονται σε δύσκολα μαθηματικά προβλήματα), δεν θα παρέχουν ασφάλεια στο νέο τεχνολογικό περιβάλλον. Εξάλλου, ο διαγωνισμός του οργανισμού NIST, ο οποίος είναι σε εξέλιξη για την εύρεση πρότυπων αλγορίθμων μετα-κβαντικής κρυπτογραφίας, αποδεικνύει πόσο αναγκαίο και σπουδαίο είναι το να προσδιοριστούν αποδοτικοί ισχυροί αλγόριθμοι μετα-κβαντικής κρυπτογραφίας ως νέα πρότυπα κρυπτογράφησης.

1.3 Μεθοδολογία

Η παρούσα διατριβή επικεντρώνεται, σε κρυπτογραφικά σχήματα ψηφιακών υπογραφών, τα οποία περιλαμβάνονται στην μετα-κβαντική κρυπτογραφία. Ιδιαίτερη προσοχή θα δοθεί σε σχήματα βασισμένα σε κρυπτογραφικές συναρτήσεις κατακερματισμού (Lamport, Merkle, Winternitz, W-OTS+, XMSS), εστιάζοντας ιδίως, ως προς την υλοποίηση και μέτρηση απόδοσης, στο σχήμα ψηφιακών υπογραφών. Στα παραπάνω σχήματα, θα γίνει ανάλυση ως προς την λειτουργία τους, την ασφάλειά τους και θα αξιολογηθούν ως προς την απόδοσή τους, σε σημερινά υπολογιστικά συστήματα.

1.4 Αποτελέσματα της έρευνας

Τα αποτελέσματα από τις δοκιμές που πραγματοποιήσαμε είναι ότι αν και ο αλγόριθμος XMSS είναι πιο αργός από τον αλγόριθμο RSA, ωστόσο δίνει αναμενόμενα – βάσει της θεωρίας - αποτελέσματα. Η ταχύτητα του αλγορίθμου RSA είναι αντιστρόφως ανάλογη, με το μήκος των κλειδιών που χρησιμοποιεί. Όσο αυξάνεται το μήκος κλειδιού που χρησιμοποιεί ο αλγόριθμος RSA τόσο μειώνεται η ταχύτητα εκτέλεσης του αλγόριθμου. Το μέγεθος της υπογραφής, επηρεάζεται στον αλγόριθμο RSA, μόνο από το μήκος των κλειδιών, ενώ στον αλγόριθμο XMSS, επηρεάζεται από το αριθμό εξυπηρέτησης υπογραφών στο σχήμα (μεγαλύτερο ύψος του αντίστοιχου δέντρου που δημιουργείται) και την συνάρτηση κατακερματισμού που χρησιμοποιεί ο αλγόριθμος. Τέλος, όσο πιο δυνατός είναι ο επεξεργαστής τόσο καλύτεροι χρόνοι παράγονται.

1.5 Δομή της εργασίας

Η δομή της διατριβής αποτελείται από τα παρακάτω κεφάλαια:

Το παρόν 1ο κεφάλαιο αποτελεί εισαγωγή. Το 2ο κεφάλαιο παρουσιάζει μία σύντομη βιβλιογραφική ανασκόπηση στην κρυπτογραφία (συμβατικών αλγορίθμων κρυπτογράφησης). Το 3ο κεφάλαιο αναλύει τα βασικά σχήματα μετα-κβαντικής κρυπτογραφίας και ειδικότερα σχήματα κατακερματισμού, στα οποία εστιάζει η παρούσα διατριβή. Το 4ο κεφάλαιο παρουσιάζει ως μελέτη περίπτωσης τη σύγκριση ενός συμβατικού αλγορίθμου και ενός μετα-κβαντικού αλγορίθμου: συγκρίνεται η απόδοση των αλγορίθμων εξετάζοντας διαφορετικούς συνδυασμούς κλειδιών, μεγέθους της εξόδου της υποκείμενης συνάρτησης κατακερματισμού και υπολογιστικής ισχύος. Το 5ο κεφάλαιο ολοκληρώνει την εργασία με τα συμπεράσματά της.

Κεφάλαιο 2
Κρυπτογραφία

2.1 Εισαγωγή

Η κρυπτογράφηση είναι η διαδικασία μετατροπής ενός μηνύματος (ή ενός απλού κειμένου) σε κρυπτοκείμενο (μη κατανοητό κείμενο) και η αποκρυπτογράφηση είναι η αντίστροφη διαδικασία, από ένα κρυπτοκείμενο στο αρχικό μήνυμα. Το κρυπτογραφημένο μήνυμα μεταδίδεται, από τον αποστολέα στον παραλήπτη μέσω ενός μη ασφαλούς καναλιού επικοινωνίας (δεδομένου ότι οποιοσδήποτε τρίτος μπορεί να παρακολουθήσει ή να υποκλέψει τα δεδομένα που μεταδίδονται μέσω αυτού του καναλιού). Ο αλγόριθμος που χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση ονομάζεται κρυπτογράφημα (cipher). Ένα κρυπτοσύστημα είναι ένα οποιοδήποτε σύστημα το οποίο περιλαμβάνει σύνολο αλγορίθμων, μαζί με σύνολα πιθανών παραμέτρων τα οποία χρησιμοποιούνται για την κρυπτογράφηση και αποκρυπτογράφηση μηνυμάτων (Λάσκαρη 2010).

Τα σύγχρονα κρυπτοσυστήματα χωρίζονται σε δύο κατηγορίες: σε εκείνα που εφαρμόζουν κρυπτογράφηση συμμετρικού κλειδιού, και σε εκείνα που εφαρμόζουν κρυπτογράφηση ασύμμετρου (ή δημόσιου) κλειδιού. Στην πρώτη κατηγορία, ο αποστολέας (Α) και ο παραλήπτης (Β) μοιράζονται ένα κλειδί, το οποίο δεν αποκαλύπτουν σε κάποιον τρίτο. Αυτό το κλειδί χρησιμοποιείται για την κρυπτογράφηση του απλού κειμένου, καθώς και την αποκρυπτογράφιση του κρυπτοκειμένου. Επιπλέον τα συστήματα αυτά διακρίνονται σε κρυπτοσυστήματα τμηματικής κρυπτογράφησης (block cipher), και σε κρυπτοσυστήματα ροής (stream cipher).

Εικόνα 1. Λειτουργία κρυπτογράφησης συμμετρικού κλειδιού

Στη δεύτερη κατηγορία, εκτός από το μυστικό κλειδί που έχει ο καθένας, από τους αποστολέα, παραλήπτη, υπάρχει και ένα δημόσιο κλειδί, που είναι διαθέσιμο σε όλους. Το δημόσιο κλειδί του Β χρησιμοποιείται από τον Α για να κρυπτογραφήσει το μήνυμα του και να το στείλει στον Β. Το μήνυμα αποκρυπτογραφείται από τον Β (και μόνο από τον Β) χρησιμοποιώντας το μυστικό κλειδί του. Πλεονέκτημα των συστημάτων δημοσίου κλειδιού, είναι ότι η ανταλλαγή μηνύματος μεταξύ αποστολέα παραλήπτη, μπορεί να πραγματοποιηθεί, χωρίς να υπάρχει εκ των προτέρων καμία ανταλλαγή μυστικής πληροφορίας μεταξύ τους.

Αντίστοιχα, η κρυπτανάλυση αποσκοπεί στην εφαρμογή διάφορων μεθόδων με σκοπό την αποκάλυψη του πρωτότυπου κειμένου. Αυτό μπορεί να επιτευχθεί με τη μορφή επιθέσεων και χωρίς να υπάρχει γνώση του μυστικού κλειδιού. Για παράδειγμα, οι εισβολείς αξιοποιούν τρωτά σημεία του κρυπτοσυστήματος. Οι συνήθεις μορφές επιθέσεων είναι (Λάσκαρη 2010):

- Επίθεση μόνο κρυπτογραφήματος: σε αυτή τη περίπτωση ο εισβολέας έχει υποκλέψει κρυπτογραφήματα τα οποία αναλύει με σκοπό να αναγνωρίσει το αρχικό μήνυμα ή να συνθέσει το μυστικό κλειδί με σκοπό να αποκρυπτογραφήσει μελλοντικά μηνύματα. Βασική παραδοχή σε αυτή τη περίπτωση είναι ότι τα κρυτογραφήματα είναι προϊόν του ίδιου αλγορίθμου.

- Επίθεση γνωστού πρωτότυπου κειμένου: σε αυτή τη περίπτωση ο εισβολέας έχει στη κατοχή του και τα κρυπτογραφήματα και τα πρωτότυπα μηνύματα. Αποσκοπεί έτσι να εξάγει το μυστικό κλειδί αλλά και τον αλγόριθμο κρυπτογράφησης που εφαρμόστηκε. Θα μπορεί έτσι να υποκλέψει μελλοντικά πρωτότυπα μηνύματα.

- Επίθεση επιλεγμένου πρωτότυπου κειμένου: είναι ανάλογη περίπτωση με την επίθεση γνωστού πρωτότυπου κειμένου αλλά θεωρείται πιο ισχυρή γιατί δίνει το πλεονέκτημα στον εισβολέα να επιλέξει τα πρωτότυπα κείμενα που θέλει να αναλύσει (για τα οποία θεωρούμε ότι είναι σε θέση να μάθει τα αντίστοιχα κρυπτογραφήματα).

- Επίθεση επιλεγμένου κρυπτογραφήματος: είναι η αντίστροφη περίπτωση της ανωτέρω. Σε αυτή, θεωρούμε ότι ο εισβολέας μπορεί να επιλέξει κρυπτογραφήματα, για τα οποία είναι θέση να μάθει τα αντίστοιχα αρχικά μηνύματα και να τα αναλύσει περαιτέρω.

2.2 Συμβατικοί αλγόριθμοι κρυπτογράφησης

Στην εποχή μας και με τη δεδομένη υπολογιστική ισχύ, έχουν επικρατήσει οι παρακάτω αλγόριθμοι κρυπτογράφησης για την ανταλλαγή δεδομένων και μηνυμάτων μεταξύ υπολογιστών, μέσω του διαδικτύου και άλλων μέσων (Gupta και Kaur Walia 2014).

1. Πρότυπο κρυπτογράφησης δεδομένων (DES)

Ο DES είναι ένας αλγόριθμος κρυπτογράφησης που εφαρμόζεται σε μέρη του μηνύματος (blocks). Ήταν το πρώτο πρότυπο κρυπτογράφησης που δημοσίευσε ο NIST. Είναι ένας συμμετρικός αλγόριθμος που σημαίνει ότι το ίδιο κλειδί χρησιμοποιείται για κρυπτογράφηση και αποκρυπτογράφηση. Χρησιμοποιεί ένα 64-bit κλειδί. Από τα 64 bit, τα 56 συνθέτουν το κλειδί και τα υπόλοιπα 8 bit χρησιμοποιούνται για την ανίχνευση σφαλμάτων.

Οι κύριες λειτουργίες του είναι οι μετατροπές των bits και η αντικατάσταση τους σε ένα γύρο (DES). Έξι διαφορετικές πράξεις μετατόπισης χρησιμοποιούνται τόσο για την επέκταση του κλειδιού όσο και κατά την ρουτίνα κρυπτογράφησης. Η αποκρυπτογράφηση του αλγόριθμου DES είναι παρόμοια με την κρυπτογράφηση, μόνο που τα ζυγά κλειδιά είναι σε αντίστροφη σειρά. Η έξοδος είναι ένα νέο (υπο-) μήνυμα (block) μήκους 64 bit. Πολλές επιθέσεις στο παρελθόν εντόπισαν τις αδυναμίες του DES, γεγονός που τον έχει κατατάξει στους μη ασφαλείς

αλγορίθμους. Εξάλλου, λόγω του μικρού μεγέθους κλειδιού, ήδη από το 1997 κατέστη σαφές ότι πρέπει να αντικατασταθεί ως πρότυπο κρυπτογράφησης.

2. 3DES (Triple DES)

Ο αλγόριθμος 3DES είναι μια βελτίωση του DES. Χρησιμοποιεί μέγεθος κλειδιού 192 bits που εφαρμόζεται σε ένα μέρος του μηνύματος σειριακά μήκους 64 bits. Η μέθοδος κρυπτογράφησης είναι παρόμοια με την αρχική DES αλλά εφαρμόζεται 3 φορές για να αυξηθεί το επίπεδο της ασφάλειας κατά την κρυπτογράφηση. Όμως έτσι καθίσταται πιο αργός από άλλες μεθόδους κρυπτογράφησης. Παρέχει ωστόσο μεγαλύτερη αξιοπιστία και μεγαλύτερο μήκος κλειδιού που εξαλείφει πολλές επιθέσεις σύντομης διάρκειας. Με άλλα λόγια, το 3DES μπορεί να χρησιμοποιηθεί για να μειωθεί ο χρόνος που χρειάζεται ένα κακόβουλο πρόγραμμα (εισβολέας) να σπάσει το κλειδί του DES.

Αν και ο 3DES ακόμα θεωρείται ασφαλής, εν τούτοις ο οργανισμός NIST επισημαίνει ότι υπάρχει κάποιο επίπεδο κινδύνου κατά τη χρήση του, ενώ αναμένεται στα προσεχή λίγα χρόνια να καταστεί παρωχημένος.

3. AES (Πρότυπο προηγμένης κρυπτογράφησης)

Ο σκοπός του NIST ήταν να οριστεί ένας αντικαταστάτης για το DES για να μπορεί να χρησιμοποιηθεί σε μη στρατιωτικές εφαρμογές ασφάλειας πληροφοριών από αμερικανικές κυβερνητικές υπηρεσίες. Έτσι αναπτύχθηκε ο AES, γνωστός και ως αλγόριθμος του Rijndael, που είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης. Ο αλγόριθμος κρυπτογραφεί ομάδες δεδομένων μήκους 128 bit χρησιμοποιώντας συμμετρικά κλειδιά. Έχει ένα μεταβλητό μήκος κλειδιού 128, 192 ή 256 bits: στις περισσότερες υλοποιήσεις επιλέγεται το τελευταίο μέγεθος των 256 bits (το οποίο, όπως θα δούμε και στη συνέχεια, παρέχει ασφάλεια και στην μετα-κβαντική εποχή) . Κρυπτογραφεί τα δεδομένα σε 10, 12 και 14 γύρους ανάλογα με το μέγεθος του κλειδιού. Μπορεί να εφαρμοστεί σε διάφορες πλατφόρμες καθώς οι μικρές συσκευές κρυπτογράφησης του AES είναι γρήγορες και ευέλικτες και έχει δοκιμαστεί για πολλές εφαρμογές ασφαλείας.

4. Blowfish

Είναι ένας από τους πιο ευρύτερα εφαρμοσμένους, για πολλά χρόνια, αλγορίθμους κρυπτογράφησης. Σχεδιάστηκε το 1993 από τον Bruce Schneier ως μια γρήγορη εναλλακτική λύση στους τότε υπάρχοντες αλγόριθμους κρυπτογράφησης. Εφαρμόζεται συμμετρικά σε μεγέθη δεδομένων 64 bit με ένα μεταβλητό μήκος κλειδιού από 32 bits έως 448 bits. Εκτελείται σε 16 γύρους ή λιγότερο, κάτι που τον καθιστά πολύ ασφαλή αλγόριθμο κρυπτογράφησης. Πράγματι δεν υπάρχουν αναφορές για επιθέσεις εναντίον του Blowfish, αν και πάσχει από προβλήματα λόγω της πιθανότητας δημιουργίας αδύναμων κλειδιών.

5. IDEA (Διεθνής Αλγόριθμος κρυπτογράφησης δεδομένων)

Ο IDEA υπήρξε ένας αλγόριθμος κρυπτογράφησης που εφαρμόζεται σε μέρος του κειμένου μήκους 64-bit. Το μέγεθος του κλειδιού είναι μήκους 128 bits. Η υλοποίηση του αλγορίθμου βασίζεται σε μείγμα συναρτήσεων από διαφορετικά αλγεβρικά σχήματα. Τρία αλγεβρικά σχήματα αναμειγνύονται και μπορούν εύκολα να υλοποιηθούν τόσο στο υλικό όσο και σε κάποιο πρόγραμμα λογισμικού: XOR, κύκλωμα πρόσθεσης 216, κύκλωμα πολλαπλασιασμού 216 + 1. Όλες αυτές οι λειτουργίες λειτουργούν με υπο-ομάδες των 16 bit καθώς ο αλγόριθμος είναι αποτελεσματικός σε επεξεργαστές των 16 bits.

6. RSA

Ο αλγόριθμος RSA είναι αλγόριθμος δημοσίου κλειδιού, εν αντιθέσει με τους ανωτέρω που είναι συμμετρικού κλειδιού. Ονομάστηκε έτσι από τους μαθηματικούς που τον δημιούργησαν: Ron Rivest, Adi Shamir και Leonard Adleman. Δημοσιεύθηκε για πρώτη φορά το 1997 και χρησιμοποιεί κλειδί μεταβλητού μεγέθους και διαδικασία κρυπτογράφησης σε τμήματα. Χρησιμοποιεί πρώτους αριθμούς για τη δημιουργία του δημόσιου και ιδιωτικού κλειδιού και στη συνέχεια πολλαπλασιάζει μεγάλους αριθμούς μαζί. Το μέγεθος του κλειδιού μπορεί να φτάσει έως τα 2048 bits (αυτή είναι η τιμή που θεωρείται ασφαλής σήμερα).

Περιγράφοντας την λειτουργία του RSA, για την δημιουργία δημόσιου και ιδιωτικού κλειδιού, συνοπτικά, έχει ως εξής:

-Επιλέγονται δυο μεγάλοι πρώτοι αριθμοί p και q από έναν χρήστη.

-Γίνεται πολλαπλασιασμός των δυο προηγουμένων αριθμών και το γινόμενό τους είναι ένας αριθμός N (N=pq) που για να είναι ασφαλές θα πρέπει να έχει τουλάχιστον 2048 δυαδικά ψηφία.

-Υπολογίζεται η συνάρτηση Euler φ(N) για τον αριθμό N που βρήκαμε παραπάνω και αποδεικνύεται ότι αν ισχύει N=pq, τότε ισχύει φ(N)=(p-1)(q-1).

-Επιλέγεται αριθμός e τυχαίος, τέτοιος ώστε gcd(e,φ(N))=1 (θεώρημα Euler, αν gcd(m,N)=1, τότε $m^{\varphi(N)} \equiv (\text{mod } N)$).

-Υπολογίζεται το $d=e^{-1} (\text{mod } \varphi(N))$

- Μετά από τους παραπάνω υπολογισμούς, το ζεύγος αριθμών (N,e) αποτελεί το δημόσιο κλειδί, ενώ αντίστοιχα, ο αριθμός d αποτελεί το ιδιωτικό κλειδί.

Ο αλγόριθμος RSA μπορεί να χρησιμοποιηθεί και για κρυπτογράφηση / αποκρυπτογράφηση μηνυμάτων. Καθώς ο αποστολέας γνωρίζει το κλειδί κρυπτογράφησης και ο παραλήπτης γνωρίζει το κλειδί αποκρυπτογράφησης, το κύριο πλεονέκτημα του αλγορίθμου RSA είναι η βελτιωμένη ασφάλεια και ευκολία. Για να διατηρήσει την ασφάλεια του ο αλγόριθμος RSA, η διαφορά των |p-q| δεν πρέπει να είναι πολύ μικρή, επίσης τα e και d θα πρέπει να είναι μεγάλου μεγέθους αριθμοί και τέλος το N να μην κοινοποιείται μεταξύ χρηστών. Η χρήση του μηχανισμού δημιουργίας ψηφιακών υπογραφών PKC είναι επίσης ένα πλεονέκτημα αυτού του αλγορίθμου. Παρόλο που το RSA δεν διαθέτει υψηλή ταχύτητα κρυπτογράφησης, μπορεί να θεωρηθεί αξιόπιστη λύση για τη δημιουργία ψηφιακής υπογραφής.

2.3 Συναρτήσεις κατακερματισμού

Οι συναρτήσεις κατακερματισμού (hash functions) επιστρέφουν στην έξοδό τους ένα μοναδικό, σταθερού μήκους αποτύπωμα, ενώ δέχονται στην είσοδό τους ένα οποιοδήποτε μεγέθους μήνυμα. Οι συναρτήσεις κατακερματισμού επιτυγχάνουν, την ακεραιότητα του μηνύματος, την αυθεντικοποίηση του αποστολέα και την μη αποποίηση ευθυνών (για τις τελευταίες δύο υπηρεσίες ασφαλείας, αυτό γίνεται μέσω πιο σύνθετων κρυπτογραφικών μετασχηματισμών οι οποίοι βασίζονται σε συναρτήσεις κατακερματισμού, όπως είναι οι ψηφιακές υπογραφές). Λόγω των ανωτέρω, χρησιμοποιούνται ευρύτερα σε συστήματα ψηφιακών υπογραφών και πιστοποίησης μηνυμάτων, για την αποθήκευση κατακερματισμένων κωδικών πρόσβασης,

εγκυρότητα αρχείων, προστασία συνθηματικών και την παραγωγή κλειδιών (Katz, Lindell 2014).

Οι συναρτήσεις κατακερματισμού για να είναι ασφαλείς πρέπει να έχουν τρεις βασικές ιδιότητες:

1. Να είναι μονόδρομες. Αυτό σημαίνει ότι δοθείσας μίας κατακερματισμένης τιμής c, πρέπει να είναι υπολογιστικά ανέφικτο να βρεθεί μήνυμα εισόδου m για το οποίο να ισχύει c= h(m), όπου h η συνάρτηση κατακερματισμού. Δηλαδή, από το ψηφιακό αποτύπωμα δεν θα πρέπει να μπορούμε να αναπαράγουμε το αρχικό μήνυμα.

2. Να ικανοποιούν τη λεγόμενη ασθενή αντίσταση σε συγκρούσεις: δοθέντος ενός μηνύματος m1, δεν θα πρέπει να είναι υπολογιστικά εφικτό η εύρεση άλλου μηνύματος m2 τέτοιο ώστε να ισχύει h(m1)=h(m2).

3. Να ικανοποιούν τη λεγόμενη ισχυρή αντίσταση σε συγκρούσεις. Δηλαδή μία συνάρτηση κατακερματισμού είναι ισχυρή εφόσον είναι ανέφικτο να βρεθούν δύο μηνύματα m1, m2 διαφορετικά μεταξύ τους για τα οποία να ισχύει h(m1)=h(m2).

Γνωστές συναρτήσεις κατακερματισμού είναι :

-Ο MD5[2] είναι ένας αλγόριθμος σύνθεσης και ανταλλαγής κρυπτογραφημένων μηνυμάτων. Προέρχεται από την προγενέστερη έκδοση του MD4 και σχεδιάστηκε από τον Ron Rivest το 1991. Το 2008 βρέθηκαν προβλήματα ασφαλείας στον MD5 σε επίθεση.

-Ο SHA-1 είναι αλγόριθμος που αναπτύχθηκε από το NIST (National Institute of Standards and Technology), στην είσοδό του παίρνει μήνυμα μήκους μικρότερο από το 2^{64} bits και παράγει μήνυμα 160 bits. Το 2017 (Marc, Bursztein, Karpman, Albertini, και Markov 2017) δημοσιεύτηκε μια επιτυχημένη επίθεση σε αυτόν.

-Ο SHA-2 που παρουσιάστηκε το 2002, με τρείς εκδόσεις μήκους σύνοψης 256, 384 και 512 bits, αλλά που βασίζονται στην ίδια μαθηματική μέθοδο.

[2] Message Digest

-Ο SHA-3 αλγόριθμος κατακερματισμού, που υιοθέτησε μετά από διαγωνισμό ο NIST. Ο διαγωνισμός ξεκίνησε το 2007 και ολοκληρώθηκε το 2012. Από 64 υποψηφιότητες επελέγη ο αλγόριθμος Keccak των Bertoni, Daeman, Peters και Van Assche. Ο αλγόριθμος αυτός θεωρείται ως εναλλακτική για «ελαφριές» () υλοποιήσεις και όχι σαν αντικατάσταση του SHA-2.

Στη περίπτωση των ψηφιακών υπογραφών που βασίζονται σε αλγορίθμους δημοσίου κλειδιού, μία συνάρτηση κατακερματισμού δουλεύει ως εξής:

Ο αποστολέας Α υπολογίζει τη κατακερματισμένη τιμή MD του αρχικού μηνύματος Μ, το MD κρυπτογραφείται με το ιδιωτικό κλειδί Κ του αποστολέα Α και παράγει την ψηφιακή υπογραφή DS. Το αρχικό μήνυμα Μ και η ψηφιακή υπογραφή DS αποστέλνονται στον παραλήπτη. Για την επαλήθευση της ψηφιακής υπογραφής από τον παραλήπτη Β, υπολογίζει τη κατακερματισμένη τιμή MD του αρχικού μηνύματος Μ, που παρέλαβε και επαληθεύει την ψηφιακή υπογραφή DS με το δημόσιο κλειδί PK του αποστολέα Α. Αυτό οδηγεί στον υπολογισμό του MD. Αν αυτά είναι ίδια τότε η ταυτότητα του αποστολέα επιβεβαιώνεται και επίσης πιστοποιείται και η ακεραιότητα του μηνύματος Μ.

2.4 Σχήματα ψηφιακής υπογραφής

Η ψηφιακή υπογραφή είναι μία ακολουθία ηλεκτρονικών δεδομένων (χαρακτήρων) που επισυνάπτονται σε ψηφιακά αρχεία ή άλλα δεδομένα και χρησιμεύουν ως μέθοδος απόδειξης της γνησιότητάς τους. Συνεπώς η ψηφιακή υπογραφή είναι άμεσα σχετιζόμενη με το περιεχόμενου του αρχείου ή μηνύματος και την ταυτότητα του υπογράφοντος. Αντίστοιχα ο παραλήπτης κατά την διαδικασία επαλήθευσης της υπογραφής πιστοποιεί ότι το πρωτότυπο αρχείο ή μήνυμα δεν έχει αλλοιωθεί και έχει παραληφθεί από τον αναμενόμενο αποστολέα.

Σήμερα το εύρος εφαρμογής των ψηφιακών υπογραφών είναι μεγάλο. Από την ηλεκτρονική ανταλλαγή δεδομένων και συναλλαγών (EDI[3]), τα ηλεκτρονικά τιμολόγια, τις ηλεκτρονικές προμήθειες π.χ. του Δημοσίου, ηλεκτρονικές ψηφοφορίες, συστήματα ηλεκτρονικών πληρωμών (Europay δίκτυο πιστωτικών καρτών, Mastercard, Visa μέσω του κοινού τους πρωτοκόλλου EMV). Επίσης εφαρμόζονται σε ψηφιακά διαβατήρια, υπηρεσίες ασφαλούς ηλεκτρονικού ταχυδρομείου, σε εφαρμογές ασφάλειας μεγάλων οργανισμών, πιστοποίηση ταυτότητας εξυπηρετητών διαδικτύου.

[3] Electronic Data Interchange

Όπως αναφέραμε παραπάνω, στη δημιουργία και επαλήθευση μιας ψηφιακής υπογραφής εμπλέκεται η εφαρμογή συναρτήσεων κατακερματισμού.

Γενικότερα, ένα σχήμα ψηφιακής υπογραφής περιγράφεται από μια πλειάδα αλγορίθμων (KeyGen, Sign, Verify) που ορίζονται ως εξής:

- Ο αλγόριθμος KeyGen για την παραγωγή κλειδιών είναι ένας αλγόριθμος που βασίζεται στη θεωρία πιθανοτήτων (δηλαδή παρουσιάζει τυχαιότητα) και εξάγει ένα δημόσιο κλειδί pk και ένα μυστικό κλειδί sk, δηλαδή ένα ζεύγος κλειδιών (pk, sk).

- Ο αλγόριθμος δημιουργίας υπογραφής Sign είναι ένας αλγόριθμος που επίσης βασίζεται στη θεωρία πιθανοτήτων και λαμβάνει ως είσοδο ένα μήνυμα m και ένα μυστικό κλειδί sk για να παράγει μια υπογραφή σ.

- Ο αλγόριθμος επαλήθευσης Verify είναι ένας ντετερμινιστικός αλγόριθμος που λαμβάνει ως είσοδο ένα μήνυμα m, μια υπογραφή σ και ένα δημόσιο κλειδί pk. Εκπέμπει τη λογική (Boolean) τιμή True για να υποδείξει ότι η υπογραφή είναι αποδεκτή ή False για να δηλώσει την απόρριψη της.

Για λόγους ορθότητας, απαιτείται ότι για όλα τα ζεύγη (pk, sk) ← KeyGen (), όλα τα μηνύματα m, και όλες οι υπογραφές σ ← Sign (m, sk), να ισχύει ότι Verify (m, σ, pk) = True. Αυτό εκφράζει ότι όλες οι υπογραφές που έχουν δημιουργηθεί σωστά είναι πράγματι αποδεκτές.

Για την αποτίμηση της ασφάλειας των σχημάτων των υπογραφών θεωρούμε ότι ο υποκλοπέας – αυτός που επιχειρεί μία επίθεση - μπορεί να επιλέγει απροσδιόριστα επιλεγμένα από αυτόν μηνύματα και να παρατηρεί τις υπογραφές τους (Goldwasser, Micali, και Rivest 1988: 281). Διαισθητικά, αυτή η έννοια αποτυπώνει την ιδέα ότι ο εισβολέας δεν θα πρέπει να μπορεί να δημιουργήσει έγκυρες υπογραφές για οποιοδήποτε μήνυμα m, ακόμη και μετά τη λήψη υπογραφών σε πολλά άλλα μηνύματα της επιλογή του. Ορίζουμε αυτό πιο τυπικά παρακάτω. Για ένα σχήμα ψηφιακής υπογραφής (KeyGen, Sign, Verify), εξετάζουμε το παρακάτω σενάριο μεταξύ ενός χρήστη C και ενός αντιπάλου (εισβολέα) A:

1. Ο C εκτελεί (pk, sk) ← KeyGen () και στέλνει το pk στον A.

2. Ο A στέλνει ένα ελεύθερα επιλεγμένο μήνυμα στον C.

3. Ο C απαντά με σi ← Sign (mi, sk).

4. Ο Α επαναλαμβάνει τα βήματα 2. και 3. για q επαναλήψεις, με q να ορίζεται ως ένα πολυώνυμο σχετικό με τα δεδομένα εισόδου.

5. Ο Α, αξιοποιώντας όλες τις πληροφορίες που συνέλλεξε από την προηγούμενη συνδιαλλαγή, εξάγει ένα ζεύγος (m ', σ'). Αυτό είναι έγκυρο εάν Verify (m ', σ', pk) = True και m '≠ mi για όλα τα i ∈ {1, . …, q}.

2.5 Έλευση μετα-κβαντικής κρυπτογραφίας

Αν σκεφτεί κάποιος ότι στο μέλλον (σε δέκα ή δεκαπέντε χρόνια), ανακοινωθεί η κατασκευή ενός μεγάλου κβαντικού υπολογιστή, αυτό θα είχε ως αποτέλεσμα το τέλος σημερινών ασφαλών αλγόριθμων κρυπτογραφίας. Τα δεδομένα των χρηστών στους υπολογιστές δεν θα είναι ασφαλή.

Οι σημερινοί ασφαλείς αλγόριθμοι που θα έσπαζαν, με την έλευση των κβαντικών υπολογιστών, είναι οι γνωστοί αλγόριθμοι δημοσίου κλειδιού. Αυτό θα συμβεί, γιατί οι αλγόριθμοι αυτοί, βασίζουν την ασφάλειά τους, σε γνωστά «δύσκολα» μαθηματικά προβλήματα, που όμως με τους κβαντικούς υπολογιστές, θα μπορούν να επιλυθούν πολύ γρήγορα. Συγκεκριμένα, ο αλγόριθμος Shor, μπορεί να χρησιμοποιηθεί για την αποτελεσματική διάσπαση των μεγάλων σύνθετων αριθμών στους πρώτους αριθμούς, καθώς και για την επίλυση του σχετικού προβλήματος διακριτού λογαρίθμου. Αυτά τα προβλήματα αξιοποιούνται κατάλληλα σε όλες τις σύγχρονες μεθόδους κρυπτογράφησης και ανταλλαγής κλειδιών που χρησιμοποιούνται σήμερα (Diffie-Hellman, RSA, τεχνικές ελλειπτικής καμπύλης). Μαζί, αυτά τα συστήματα περιλαμβάνουν ουσιαστικά όλη την κρυπτογραφία δημόσιου κλειδιού που χρησιμοποιείται ευρέως μέχρι αυτή τη στιγμή.

Η ασφάλεια των μηνυμάτων που κρυπτογραφούνται μέσω του αλγορίθμου RSA, βασίζεται στην τεράστια υπολογιστική προσπάθεια που απαιτείται για να βρεθούν οι πρώτοι αριθμοί ενός μεγάλου αριθμού N χρησιμοποιώντας συμβατικούς υπολογιστές. Το 1994 ο Peter Shor έδειξε ότι για αρκετά μεγάλο N, ένας κβαντικός υπολογιστής θα μπορούσε να εκτελέσει τους υπολογισμούς (παραγοντοποίηση) με πολύ λιγότερη υπολογιστική προσπάθεια(Gerjuoy 2005: 73).

Από την άλλη μεριά, οι συμμετρικοί αλγόριθμοι και οι συναρτήσεις κατακερματισμού, φαίνονται να είναι πιο ανθεκτικοί, στους κβαντικούς υπολογιστές. Οι γνωστοί αλγόριθμοι, του Grover, που μπορεί να κάνει γρήγορη αναζήτηση σε μη ταξινομημένες βάσεις δεδομένων και ο αλγόριθμος των Brassard et al, που μπορεί να δώσει κατά προσέγγιση μέτρηση, μπορούν να επηρεάσουν τους συμμετρικούς αλγόριθμους και τις συναρτήσεις κατακερματισμού, μέχρι κάποιο σημείο. Αυξάνοντας κατάλληλα το μέγεθος των κλειδιών, στους συμμετρικούς αλγόριθμους και αυξάνοντας το μήκος κατακερματισμού στην έξοδο, στις συναρτήσεις κατακερματισμού, θα μπορούν να είναι ασφαλείς, και μετά την έλευση των κβαντικών υπολογιστών. Οι μεταβολές στα μεγέθη του κλειδιού και κατακερματισμού, στην πράξη, έχουν μεγάλη επίδραση στην εκτεταμένη κρυπτογραφία για τα δεδομένα σε κίνηση και θα απαιτήσουν σημαντική υπολογιστική μηχανική για την υλοποίηση τους.

Παραδείγματα, ευρέως χρησιμοποιούμενων κρυπτογραφικών συστημάτων και των επιπέδων ασφαλείας τους ενάντια στις γνωστές προκβαντικές και μετα-κβαντικές επιθέσεις φαίνονται στον παρακάτω Πίνακα 1. Ο πίνακας συνοψίζει τις επιδράσεις των αλγορίθμων των Shor και Grover στους τυπικούς αλγορίθμους κρυπτογράφησης πρότυπα. Δείχνει ότι με την έλευση κβαντικών υπολογιστών, οι αλγόριθμοι κρυπτογραφίας δημοσίου κλειδιού, δεν θα είναι ασφαλείς, ενώ οι αλγόριθμοι της συμμετρικής κρυπτογραφίας (με μεγαλύτερα μεγέθη κλειδιών) θα μπορούν – τουλάχιστον βάσει της γνώσης μας σήμερα - να είναι ασφαλείς.

Όνομα	Λειτουργία	Προκβαντικό επίπεδο ασφάλειας	Μετακβαντικό επίπεδο ασφάλειας
Συμμετρική κρυπτογραφία			
AES-128	block cipher	128	64 (Grover)
AES-256	block cipher	256	128 (Grover)
Salsa20	stream cipher	256	128 (Grover)
GMAC	MAC	128	128 (χωρίς επίπτωση)
Poly1305	MAC	128	128 (χωρίς επίπτωση)
SHA-256	hash function	256	128 (Grover)
SHA-3	hash function	256	128 (Grover)
Κρυπτογραφία δημοσίου κλειδιού			
RSA-3072	Κρυπτογράφησης	128	Έσπασε (Shor)
RSA-3072	Υπογραφής	128	Έσπασε (Shor)
DH-3072	Αντ/γη κλειδιού	128	Έσπασε(Shor)
DSA-3072	Υπογραφής	128	Έσπασε (Shor)
256-bit ECDH	Αντ/γη κλειδιού	128	Έσπασε (Shor)
256-bit ECDSA	Υπογραφής	128	Έσπασε (Shor)

Πίνακας 1. Κρυπτογραφικοί αλγόριθμοι και ασφάλεια (Bernstein και Lange 2017: 3).

Ωστόσο, θα εστιάσουμε περισσότερο στην έρευνά μας γύρω από το νεότερο πρόβλημα της αντικατάστασης των αλγορίθμων κρυπτογραφίας του δημόσιου κλειδιού με εναλλακτικές λύσεις μετα-κβαντικών αλγορίθμων. Επιπρόσθετα, πολλοί επισημαίνουν ότι η εφαρμογή του αλγόριθμου του Grover σε QCs αναμένεται να είναι δύσκολη στην πράξη λόγω των μακρόχρονων σειριακών υπολογισμών και της ανάγκης για βαθιά κυκλώματα (NIST 2017, Grassl, Langenberg, Roetteler και Steinwandt 2016).

2.6 Διαγωνισμός του NIST για τη μετα-κβαντική κρυπτογραφία

Η έρευνα στη κρυπτογραφία, αν και συχνά αρκετά ακαδημαϊκή, συνδέεται στενά με την ανάπτυξη στον πραγματικό κόσμο. Αυτό αποδεικνύεται σαφέστερα από τα πανταχού παρόντα πρωτόκολλα όπως το TLS[4] και το EMV[5], με δισεκατομμύρια χρήσεις τους κάθε μέρα. Ίσως να μην είναι έντονα αντιληπτό, ωστόσο η κοινωνία εξαρτάται σε μεγάλο βαθμό από τη σωστή και

[4] Transport Layer Security: πρωτόκολλο ασφάλειας της ιδιωτικότητας και δεδομένων κατά την επικοινωνία τους στο διαδίκτυο

[5] Europay, Mastercard, Visa: πρότυπο των οργανισμών έκδοσης πιστωτικών καρτών

αποτελεσματική λειτουργία βασικών λειτουργιών της κρυπτογραφίας. Κανένα από αυτά δεν θα ήταν εφικτό χωρίς σαφείς συμφωνίες για τις αλληλεπιδράσεις μεταξύ όλων των εμπλεκόμενων συστημάτων. Για να εξασφαλιστεί η διαλειτουργικότητα, οι οργανισμοί τυποποίησης εκδίδουν κρυπτογραφικά πρότυπα. Ένα από τους οργανισμούς αυτούς είναι ο NIST (National Institute of Standards and Technology) - των Ηνωμένων Πολιτειών.

Αν και ο NIST είναι μια υπηρεσία της κυβέρνησης των Ηνωμένων Πολιτειών, τα πρότυπά του έχουν παγκόσμιο αντίκτυπο. Όλες οι ομοσπονδιακές υπηρεσίες πρέπει να συμμορφώνονται με τον NIST μέσω του FISMA[6] και πολλοί διεθνείς οργανισμοί και εταιρείες τηρούν τις ίδιες συστάσεις.

Στην κρυπτογραφία, ο NIST είναι «υπεύθυνος», μεταξύ άλλων, για το Advanced Encryption Standard (AES) (NIST 2001) και τους Secure Hashing αλγόρθμους (SHA-1, SHA-2 και πρόσφατα SHA-3) (Dang 2015, NIST 2015). Ενώ τα SHA-1 και SHA-2 σχεδιάστηκαν από την Εθνική Υπηρεσία Ασφαλείας (NSA), τα AES και SHA-3 είναι το αποτέλεσμα μιας πιο ανοιχτής διαδικασίας. Για να αντικαταστήσει το τότε κοινό πρότυπο κρυπτογράφησης δεδομένων (DES), ο NIST διεξήγαγε έναν ανοικτό διαγωνισμό και επέλεξε τον Rijndael (Daemen και Rijmen 1999) ως AES το 2001. Το Keccak(Bertoni, Daemen, Peeters και Van Assche 2011) επελέγη ως SHA-3 για να συμπληρώσει τα SHA-1 και SHA-2 2015 με τον ίδιο τρόπο.

Το 2016, ο NIST ανακοίνωσε ένα έργο με στόχο τον καθορισμό πρότυπων αλγορίθμων της μετα-κβαντικής κρυπτογραφίας (NIST 2016). Η ανακοίνωση αυτή ακολούθησε μετά από σεμινάριο στις αρχές του 2015, με ενημερωμένες συστάσεις του NSA που προωθούν τη μετάβαση στην κβαντική εποχή στη κρυπτογραφία (CNSA Suite 2015). Σε αντίθεση με τις προηγούμενες προσπάθειες τυποποίησης, ο NIST εξέφρασε την πρόθεση να δημοσιεύσει ένα ευρύτερο χαρτοφυλάκιο εγκεκριμένων αλγορίθμων και αποφεύγει ρητά την επισήμανση αυτού του έργου ως διαγωνισμό.

Ο NIST ζήτησε υποβολές σχεδίων κρυπτογράφησης δημόσιων κλειδιών, μηχανισμών κλειδώματος-ενθυλάκωσης και συστημάτων ψηφιακής υπογραφής. Στον πρώτο γύρο υποβλήθηκαν 82 αλγόριθμοι εκ των οποίων τα 69 θεωρήθηκαν πλήρη και σωστά (Moody 2018). Από αυτά, 26 «επιβίωσαν» στη συνέχεια στον δεύτερο γύρο (NIST 2019). Αξίζει να σημειωθεί ότι ο NIST ξεκίνησε ξεκάθαρα μια ξεχωριστή διαδικασία για την τυποποίηση

[6] Fedral Information Security Management Act: νομοθετικό πλαίσιο των ΗΠΑ για την προστασία κυβερνητικών συστημάτων, υπηρεσιών και πληροφοριών έναντι απειλών

υπογραφών που έχουν μνήμη (stateful) και βασίζονται στη τεχνική του κατακερματισμού (hashing), όπως και το IETF[7]. Το IETF έκτοτε δημοσίευσε τα RFC8391 (Hülsing, Butin, Gazdag, Rijneveld και Mohaisen 2018) και RFC8554 (Curcio, McGrew, και Fluhrer 2019).

Στην πρόσκληση υποβολής προτάσεων, ο NIST ορίζει πέντε κατηγορίες για να ταξινομήσει τα επίπεδα ασφαλείας των υποβολών (NIST 2016). Η ασφάλεια των AES, SHA(1-3) χρησιμοποιείται ως πλαίσιο αναφοράς, συγκρίνοντας την ασφάλεια με τις επιθέσεις αναζήτησης κλειδιών κατά της AES και τις επιθέσεις κατά της σύγκρουσης κατά των SHA-2 και SHA-3. Σε αυξανόμενη σειρά δυσκολίας, οι κατηγορίες 1 έως 5 ισοδυναμούν σε επίπεδο ασφάλειας σε AES-128, SHA-256 / SHA3256, AES-192, SHA-384 / SHA3-384 και AES-256 αντίστοιχα (NIST 2017, Grassl, Langenberg, Roetteler και Steinwandt 2016).

[7] Internet Engineering Task Force: οργανισμός για τη διαχείριση και ανάπτυξη πρωτοκόλλων διαδικτύου

Κεφάλαιο 3
Μετα-κβαντική κρυπτογραφία

3.1 Εισαγωγή

Για πολλά χρόνια, οι επιστήμονες στην Φυσική προέβλεπαν την επικείμενη κατασκευή ενός κβαντικού υπολογιστή μεγάλης κλίμακας, ένας υπολογιστής που χρησιμοποιεί ιδιότητες από την κβαντική μηχανική για να κάνει υπολογισμούς που ξεπερνάνε κατά πολύ τους κλασικούς υπολογιστές. Παρόλο που μερικές φορές παρουσιάζεται ως μια λύση για την επίλυση των πιο πιεστικών προβλημάτων στην ανθρωπότητα, μέχρι σήμερα, οι κβαντικοί αλγόριθμοι είναι εξαιρετικά εξειδικευμένοι. Τέτοιοι εξειδικευμένοι αλγόριθμοι (Shor 1999) λύνουν ακριβώς τα προβλήματα που η κλασσική κρυπτογραφία είναι δύσκολο να επιλύσει.

Η κρυπτογραφία που παραμένει ασφαλής παρά την παρουσία ενός αντιπάλου με πρόσβαση σε έναν κβαντικό υπολογιστή ονομάζεται μετα-κβαντική κρυπτογραφία (post-quantum cryptography). Η κυριότερη διαφορά μεταξύ της μετα-κβαντικής κρυπτογράφησης και της παραδοσιακής κρυπτογραφίας έγκειται στα προβλήματα στα οποία βασίζεται - προβλήματα για τα οποία κανένας αποτελεσματικός κβαντικός αλγόριθμος δεν είναι γνωστός. Τα προβλήματα

που έχουν οριστεί μέχρι τώρα μπορούν να ταξινομηθούν σε πέντε κατηγορίες, το καθένα με τα δικά του χαρακτηριστικά, δυνατά και αδύνατα σημεία. Αυτά σχετίζονται με τις λειτουργίες κατακερματισμού, τις πολυπαραγοντικές τετραγωνικές εξισώσεις, τα πλέγματα, τους κώδικες διόρθωσης σφαλμάτων και τα ισογενή γραφήματα (supersingular isogenies).

Την τελευταία δεκαετία, η έρευνα στον τομέα της μετα-κβαντικής κρυπτογραφίας προχώρησε με σταθερά βήματα. Παρόλο που αρχικά ξεκίνησε να χρησιμοποιείται σε πειραματικό στάδιο, η μετά-κβαντική κρυπτογραφία γίνεται σταδιακά όλο και πιο πρακτική. Αυτό είναι και συνέπεια των προσπαθειών τυποποίησης γύρω από αυτή. Ειδικότερα, το 2016, το Αμερικανικό Εθνικό Ινστιτούτο Προτύπων και Τεχνολογίας (NIST) ξεκίνησε ένα πολυετές πρόγραμμα για την προτυποποίηση της μετα-κβαντικής κρυπτογράφησης, όπως αναφέρθηκε στο προηγούμενο κεφάλαιο. Με εκατοντάδες συμμετέχοντες από τον ακαδημαϊκό χώρο και τη βιομηχανία, η πρακτική μετά-κβαντική κρυπτογραφία είναι πλέον πιο σχετική από ποτέ.

3.2 Κατηγορίες μετα-κβαντικών αλγόριθμων

Η μετα-κβαντική κρυπτογραφία βασίζεται σε μαθηματικά προβλήματα (παρόμοια με την τρέχουσα, προ-κβαντική κρυπτογραφία) αλλά αυτά τα μαθηματικά προβλήματα πρέπει να εντάσσονται σε συγκεκριμένες κατηγορίες, οι οποίες θα παρέχουν ασφάλεια ακόμα και σε κβαντικούς υπολογιστές, όταν αυτοί θα γίνουν πραγματικότητα (Chen 2017).

Οι κύριες κατηγορίες των μετα-κβανικών κρυπτογραφικών συστημάτων περιλαμβάνουν αλγορίθμους βασισμένους στον:

1. κατακερματισμό (hash-based)
2. κώδικα (code-based)
3. πλέγμα (lattice-based)
4. πολυπαραγοντικές τετραγωνικές εξισώσεις (multivariate quadratic equations) και
5. ισογενή γραφήματα.

Όπως φαίνεται στον Πίνακα 2, κάθε μία από αυτές τις κατηγορίες αντιστοιχεί σε ένα συγκεκριμένο μαθηματικό πρόβλημα που είναι δύσκολο να επιλυθεί όχι μόνο με παραδοσιακούς αλλά και με κβαντικούς υπολογιστές. Θα πρέπει επίσης να επισημανθεί ότι κάθε μια από τις

παραπάνω κατηγορίες έχει διαφορετικές απαιτήσεις όσον αφορά την αποτελεσματικότητα (π.χ. στο μέγεθος των δημόσιων και ιδιωτικών κλειδιών, τα μεγέθη των κωδικών κατακερματισμού, το υπολογιστικό κόστος κλπ.) και ωριμότητα .

Κατηγορία αλγορίθμου	**Τρόπος εφαρμογής**	**Παραδείγματα /Σχήματα**	**Σημαντικές ιδιότητες**
Τεχνική Κατακερματισμού	Ψηφιακή υπογραφή	XMSS, SPHINCS+	Καλά κατανοητή τεχνική. Απαιτεί σχήματα με δυνατότητα αποθήκευσης της ενδιάμεσης κατάστασης για τη μείωση των μεγάλων μεγεθών των υπογραφών.
Τεχνική Πλέγματος	KEM/Κρυπτογράφηση, Ψηφιακές Υπογραφές	FrodoKEM, NewHope, NTRU, FALCON, qTESLA	Ψηφιακές υπογραφές FrodoKEM, NewHope, NTRU, FALCON, qTESLA. Μικρού μεγέθους κρυπτογραφικό κείμενο και κλειδιά, με καλή απόδοση, αλλά μερικές φορές περίπλοκη τεχνική. Μικρού μεγέθους υπογραφές.
Τεχνική κωδικοποίησης	KEM/Κρυπτογράφηση	BIKE, Classic McEliece, HQC, NTS-KEM, RQC	Κρυπτογράφηση με μεγάλο βαθμό αξιοπιστίας. Ταχεία κρυπτογράφηση αλλά με μεγάλου μεγέθους κλειδιά.
Τεχνική πολυπαραγοντικών εξισώσεων	Ψηφιακές Υπογραφές	EMSS, LUOV, MQDSS, Rainbow	Μεγάλου μεγέθους κλειδιά (~1 MB / ~11 KB). Τα σχήματα της τεχνικής απαιτούν μεγαλύτερη ανάλυση.
Τεχνική ισογενών γραφημάτων	KEM/Κρυπτογράφηση	SIKE	Πολύ μικρού μεγέθους κλειδιά (λιγότερο από 500 B), μικρότερη απόδοση, σχετικά νέα τεχνική.

Πίνακας 2. Κατηγορίες (PQC) Μετα-κβαντικών Αλγορίθμων και βασικές τους ιδιότητες

3.3 Προβλήματα συναρτήσεων κατακερματισμού

Οι ιδιότητες των συναρτήσεων κατακερματισμού, οι οποίες περιγράφονται στο Κεφάλαιο 2, είναι ιδιαίτερα σημαντικές και για τη μετα-κβαντική εποχή. Για έναν ισχυρό αλγόριθμο κατακερματισμού, με επαρκές μέγεθος εξόδου (τουλάχιστον 224 bits), ούτε η αντιστροφή του ούτε η εύρεση συγκρούσεων είναι υπολογιστικά εφικτές, ακόμα και με την έλευση κβαντικών υπολογιστών. Συνεπώς, κάθε κρυπτογραφικός αλγόριθμος που βασίζει την ασφάλειά του σε αυτές τις ιδιότητες ανήκει στην πρώτη – από τις ανωτέρω πέντε – κατηγορίες μετα-κβαντικών αλγορίθμων.

Δεδομένου ότι οι αλγόριθμοι PQC, και ειδικά οι συναρτήσεις κατακερματισμού, γενικά έχουν μεγαλύτερες απαιτήσεις υπολογισμού, μνήμης, αποθήκευσης και επικοινωνίας (π.χ. μεγαλύτερα μεγέθη κλειδιών, πιο περίπλοκοι αλγόριθμοι ή και τα δύο), απαιτείται έρευνα για την καλύτερη κατανόηση και ποσοτικοποίηση των εκτιμήσεων απόδοσης σε ένα ευρύ φάσμα πλαισίων ανάπτυξης. Σε γενικές γραμμές, οι επιδόσεις αποτελούν βασικό μέλημα της βιομηχανίας και μια σημαντική δέσμη προκλήσεων που πρέπει να επιλυθούν πριν από την υιοθέτηση του PQC στην πράξη.

Η αύξηση του μεγέθους του κλειδιού, που είναι αναγκαία στους συμμετρικούς αλγορίθμους για ασφάλεια στη μετα-κβαντική εποχή, δημιουργεί προβλήματα απόδοσης, ενώ επηρεάζει και υπάρχουσες ρυθμίσεις σε συσκευές ή σε πρωτόκολλα δικτύου. Πράγματι, η τεχνική του κατακερματισμού, ως μια άλλη ασφαλής εναλλακτική λύση, προτάθηκε στη δεκαετία του '70 (Buchmann, Dahmen και Hülsing 2011) και προσφέρει καλά κατανοητή ασφάλεια κάτω από συγκριτικά λίγες υποθέσεις (οι υποθέσεις έγκεινται ακριβώς στις ιδιότητες των συναρτήσεων κατακερματισμού που αναφέρθηκαν ανωτέρω). Εντούτοις, εισάγει τη νέα πολυπλοκότητα της αποθήκευσης της ενδιάμεσης κατάστασης, δεδομένου ότι οι μιας χρήσης υπογραφές απαιτούν να μην χρησιμοποιείται ένα μυστικό κλειδί δύο φορές.

Για την διαχείριση της πολυπλοκότητας που παρουσιάζεται με τη χρήση κλειδιών μεγάλου μεγέθους, μπορούν να χρησιμοποιηθούν ειδικές δεντρικές δομές (π.χ. Merkle Trees) (Merkle 1979). Οι δομές αυτές υποστηρίζουν αντίθετα τη δημιουργία ενός μεγάλου αριθμού κλειδιών μικρού μεγέθους. Η μόνη πρόκληση μίας δεντρικής τεχνικής είναι ότι τα ενδιάμεσα κλειδιά πρέπει να αποθηκεύονται με ασφάλεια (Chen 2017).

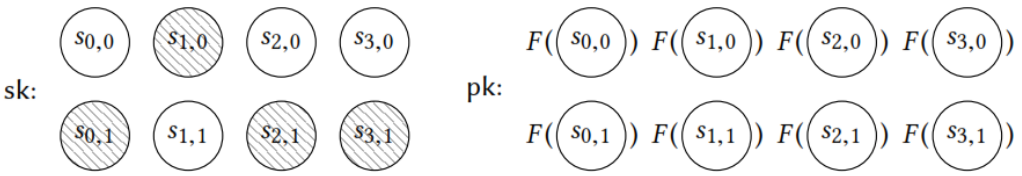
Αρκετοί ερευνητές επισημάνουν ότι τα σχήματα που αποθηκεύουν τα ενδιάμεσα κλειδιά (π.χ. XMSS) (Buchmann, Dahmen και Hülsing 2011) εμφανίζουν αρκετά μειονεκτήματα στην εφαρμογή τους, για παράδειγμα σε κατανεμημένα δίκτυα υπολογιστών ή σε περιπτώσεις που το σύστημα κρυπτογράφησης/αποκρυπτογράφησης υλοποιείται σε υλικό και όχι λογισμικό.

Παρόλα αυτά, νέα σχήματα κατακερματισμού όπως το XMSS (Buchmann, Dahmen και Hulsing 2011) και το SPHINCS (Bernstein, Hopwood, Hulsing, Lange, Niederhagen, Papachristodoulou, Schneider, Schwabe και Wilcox-O'Hearn 2015: 368) μπορούν να διαχειριστούν τα ενδιάμεσα κλειδιά με ασφάλεια. Το XMSS είναι πλέον ένα νέο RFC πρότυπο (Hulsing, Butin, Gazdag, Rijneveld και Mohaisen 2018: 17), ενώ το SPHINCS έχει συμπεριληφθεί στο έργο τυποποίησης της κβαντικής κρυπτογραφίας του NIST.

3.4 Σχήματα βασισμένα σε κρυπτογραφικές συναρτήσεις κατακερματισμού

3.4.1 Lamport

Το πρώτο σύστημα κρυπτογραφικής συνάρτησης κατακερματισμού για τη δημιουργία ψηφιακών υπογραφών πηγάζει από μια τεχνική έκθεση του Lamport (Lamport 1975: 43), αν και το ίδιο σύστημα αναφέρεται μερικές φορές ως "Lamport-Diffie σύστημα κρυπτογραφικής συνάρτησης μίας χρήσης" σε μία επόμενη εργασία του Merkle (Merkle 1990: 13). Η παρακάτω περιγραφή είναι μια μικρή αναδιατύπωση έτσι ώστε να ταιριάζει στην τρέχουσα βιβλιογραφία. Η Εικόνα 2, παρουσιάζει μια απεικόνιση μιας υπογραφής που παράγεται όπως περιγράφεται παρακάτω, ενώ οι Αλγόριθμοι 1, 2 και 3 παρακάτω επεξηγούν τον ψευδοκώδικα του σχήματος Lamport.



Εικόνα 2. Το σχήμα κατακερματισμού Lamport. Σε αυτό το παράδειγμα, το m = 1011 είναι το μήνυμα προς κρυπτογράφηση και οι γκρίζοι κόμβοι δείχνουν τα ενδιάμεσα μυστικά κλειδιά. (Rijneveld 2019).

Έστω ένα μήνυμα $m \in \{0,1\}^n$, δηλαδή δυαδικό μήνυμα m των n bits. Ορίζουμε τη μονόδρομη συνάρτηση $F: \{0,1\}^k \rightarrow \{0,1\}^k$, για την οποία ισχύει ότι, για κάθε y = F (x), είναι αδύνατο να υπολογιστεί x' για το οποίο να ισχύει F (x') = y.

1: **for** $i \in \{1, \ldots, n\}$ **do**
2: **for** $j \in \{0,1\}$ **do**
3: $s_{i,j} \xleftarrow{\$} \{0,1\}^k$
4: $p_{i,j} \leftarrow F(s_{i,j})$
5: **end for**
6: **end for**
7: **return** pk = $(p_{1,0}, p_{1,1}, \ldots, p_{n,0}, p_{n,1})$, sk = $(s_{1,0}, s_{1,1}, \ldots, s_{n,0}, s_{n,1})$

Εικόνα 3. KeyGen αλγόριθμος στο σχήμα Lamport. (Rijneveld 2019).

1: $(m_1, \ldots, m_n) = m$ ▷ Split m such that $m_i \in \{0,1\}$
2: **for** $i \in \{1, \ldots, n\}$ **do**
3: $\sigma_i = s_{i,m_i}$
4: **end for**
5: **return** $\sigma = (\sigma_1, \ldots, \sigma_n)$

Εικόνα 4. Sign (m, $s_{i,j} \in s_k$) αλγόριθμος στο σχήμα Lamport. (Rijneveld 2019).

Προτού προχωρήσει κάποιος στην παραγωγή υπογραφών, πρέπει να δημιουργήσει ένα ιδιωτικό κλειδί και το αντίστοιχο δημόσιο κλειδί. Το ιδιωτικό κλειδί αποτελείται από 2n τυχαίες τιμές, των k bits η κάθε μία. Οι τιμές αυτές συμβολίζονται ως $s_{i,j}$ για κάθε i ∈ {1, …, n} και j ∈ {0, 1}. Το δημόσιο κλειδί είναι το αποτέλεσμα της εφαρμογής της συνάρτησης F σε κάθε μυστική τιμή. Αυτό αναπαρίσταται ως $p_{i,j} \leftarrow F(s_{i,j})$ και εξάγει το δημόσιο κλειδί $pk = (p_{1,0}, p_{1,1}, \ldots, p_{n,0}, p_{n,1})$.

1: $(m_1, \ldots, m_n) = m$ ▷ Split m such that $m_i \in \{0,1\}$
2: **for** $i \in \{1, \ldots, n\}$ **do**
3: **if** $F(\sigma_i) \neq p_{i,m_i}$ **then**
4: **return** False
5: **end if**
6: **end for**
7: **return** True

Εικόνα 5. Verify (m, σi ε σ, pi,j ε pk) αλγόριθμος στο σχήμα Lamport. (Rijneveld 2019).

Έχοντας ως είσοδο ένα μήνυμα m, ο χρήστης επιλεκτικά εμφανίζει τα μυστικά κλειδιά $s_{i,j}$ που συσχετίζονται με τις τιμές των bits στο μήνυμα m. Πιο συγκεκριμένα, αυτές είναι οι τιμές s_{im_i} όπου m_i είναι το ιοστό bit του μηνύματος m. Το σύνολο αυτών των τιμών καθορίζουν την υπογραφή. Έχοντας ως είσοδο το περιεχόμενο της, ένας άλλος χρήστης στο στάδιο της

επιβεβαίωσης του γνησίου της υπογραφής εφαρμόζει τη συνάρτηση F σε ένα από τα μυστικά κλειδιά και παράγεται η συνάρτηση $F(s_i, m_i)$. Η συνάρτηση θεωρείται έγκυρη εφόσον $F(s_i, m_i) = p_{im_i}$ για όλα τα i bits του μηνύματος m.

Ως συνέπεια ορισμού της μονόδρομης συνάρτησης F, ένας τρίτος (π.χ. κακόβουλος) δεν μπορεί να δημιουργήσει μια τέτοια έγκυρη υπογραφή για οποιοδήποτε άλλο m': θα έπρεπε να έχει την γνώση των μυστικών $s_{i,j}$ που αντιστοιχούν ακριβώς στα δυαδικά ψηφία του m. Αλλάζοντας έστω και ένα bit στο m θα απαιτούσε να συμπεριληφθεί ένα επιπλέον μυστικό κλειδί στην υπογραφή.

Είναι φανερό από τα παραπάνω ότι ένα τέτοιο μυστικό κλειδί δεν μπορεί να χρησιμοποιηθεί για πολλαπλά μηνύματα. Εάν κάποιος πρόκειται να δημοσιεύσει μια υπογραφή για ένα διαφορετικό μήνυμα m' με βάση τα ίδια μυστικά κλειδιά $s_{i,j}$, ένας εξωτερικός παρατηρητής θα μάθαινε τα μυστικά κλειδιά s_{im_i} καθώς επίσης και $s_{im'_i}$. Αυτό όχι μόνο του επιτρέπει να αναπαράγει την υπογραφή για m και m' (πράγμα που δεν αποτελεί πρόβλημα, καθώς η δημοσίευσή τους θα ήταν νόμιμη), αλλά και για οποιοδήποτε άλλο μήνυμα m" που μπορεί να κατασκευαστεί συνδυάζοντας δυαδικά ψηφία των m και m', δηλαδή, όπου $m"_i \in \{m_i, m'_i\}$ για όλα τα i. Για αυτό το λόγο, οι υπογραφές αυτές λέγονται υπογραφές μίας χρήσης (one-time signatures).

Εκτός από το αδιαμφισβήτητο μειονέκτημα του ότι ένα ζεύγος κλειδιών υπογράφει μόνο ένα bit και μόνο για μία φορά, οι υπογραφές Lamport έχουν επίσης το μειονέκτημα ότι είναι επίσης αρκετά μεγάλες. Για να υπογραφούν ψηφιακά n bits, απαιτούνται 2kn bits (ιδιωτικό και δημόσιο κλειδί αντίστοιχα), για k=256, παράγονται υπογραφές των 128 KBit για κάθε ένα Kbit μηνύματος. Στο (Merkle 1990: 13), ο Merkle παρουσιάζει μία βελτίωση του σχήματος Lamport. Αντί να παράγονται μυστικά κλειδιά και για τα δύο bits, τα 1-bit και 0-bit, παράγονται μυστικά κλειδιά που αντιστοιχούν μόνο στα 1-bit. Αυτό μειώνει το μέγεθος της υπογραφής στο πενήντα τοις εκατό κατά μέσο όρο.

3.4.2 Winternitz

Επίσης, στο (Merkle 1990: 13), ο Merkle περιγράφει μια παραλλαγή που πιστώνεται στον Winternitz. Σε αυτή τη παραλλαγή μειώνεται το μέγεθος της υπογραφής έχοντας όμως το κόστος του επιπλέον χρόνου εκτέλεσης που απαιτείται. Σε αυτή τη λύση, ένα ιδιωτικό κλειδί αντιστοιχεί σε πολλά bits ενός μηνύματος (έναντι της αντιστοίχισης ένα-προς-ένα στις

παραπάνω λύσεις). Κατόπιν μία μονόδρομη συνάρτηση κατακερματισμού εφαρμόζεται διαδοχικά πολλές φορές, με έναν διαφορετικό κάθε φορά πλήθος εφαρμογών αυτής. Οι εφαρμογές της μονόδρομης συνάρτησης αναφέρονται συνήθως ως αλυσίδες και η ίδια η συνάρτηση ονομάζεται αλυσιδωτή συνάρτηση κατακερματισμού.

Για παράδειγμα, έστω η περίπτωση που έχουμε το μήνυμα για το οποίο ισχύει $w = 2^4 = 16$, δηλαδή το δημόσιο κλειδί προκύπτει από διαδοχική εφαρμογή της συνάρτησης κατακερματισμού F κατά w-1=15 φορές (δηλαδή, αν x είναι η τυχαία συμβολοσειρά που αποτελεί το ιδιωτικό κλειδί, το δημόσιο κλειδί ισούται με $F^{15}(x)$, όπου το F^{15} υποδηλώνει την εφαρμογή της συνάρτησης F κατά 1 διαδοχικές φορές). Με αυτόν τον τρόπο, μπορούμε να υπογράφουμε bits του μηνύματος σε ομάδες των 4 bits, με τον τρόπο που περιγράφεται στη συνέχεια. Αυτό σημαίνει ότι μηνύματα με n bits χρειάζονται μόνο n / 4 μυστικά (ιδιωτικά) κλειδιά για την υπογραφή τους. Η παράμετρος Winternitz (w) καθορίζει το αντιστάθμισμα στον σχεδιασμό του αλγορίθμου ανάμεσα στο μέγεθος του κλειδιού και το χρόνο δημιουργίας του.

Για κάθε ομάδα από bits, η διαδικασία κρυπτογράφησης εφαρμόζει τη συνάρτηση κατακερματισμού F, τόσες φορές όσες υποδεικνύει η τιμή (ερμηνευόμενη ως δυαδική αναπαράσταση αριθμού) της εκάστοτε ομάδας. Η μέγιστη τιμή, και επομένως ο μέγιστος αριθμός εφαρμογής της συνάρτησης, είναι w-1. Επακόλουθα, στο παράδειγμα μας τα δημόσια κλειδιά πρέπει να είναι $p_i = F^{w-1}(s_i) = F^{15}(s_i)$. Για παράδειγμα, για την κρυπτογράφηση της ομάδας m_i = 1101, δηλαδή της τιμής 13, η διαδικασία πρέπει να υπολογίσει και να διαθέσει ενδιάμεσα κλειδιά ίσα με $\sigma_i = F^{13}(s_i)$ (ήτοι η πληροφορία που απαιτείται για την επικύρωση της υπογραφής). Αυτό με τη σειρά του σημαίνει ότι απαιτούνται w-1-13 = 2 εφαρμογές της συνάρτησης F στη διαδικασία επαλήθευσης για να επαληθεύσει ένας χρήστης ότι πράγματι ισχύει $p_i = F^2(\sigma_i) = F^2(F^{13}(s_i))$.

Όπως και στο αρχικό σχήμα του Merkle έτσι και στην πρωταρχική έκδοση αυτού του σχήματος επιτρέπεται η δημιουργία ψεύτικων υπογραφών. Ένας πλαστογράφος θα μπορούσε να δημιουργήσει ένα μήνυμα m' παρόμοιο με το m με την εξαίρεση όμως της ι-οστής ομάδας από bits. Για αυτή την ομάδα μπορεί να επιλέξει ένα μήνυμα m_i' για το οποίο ισχύει $m_i' > m_i$. Κατόπιν υπολογίζει το σ_i' εφαρμόζοντας τη συνάρτηση $F(m_i)$ $m_i' - m_i$ φορές για να ολοκληρωθεί η δημιουργία του ψεύτικου κλειδιού. Η διαδικασία επαλήθευσης θα εκτελεστεί κατά μία λιγότερη επανάληψη και θα επαληθεύσει την σχέση $p_i = F^{w-1-m'_i}(F^{m'_i - m_i}(F^{m_i}(s_i)))$.

Η πιθανότητα μίας τέτοιας πλαστογράφησης εξαλείφεται με τη χρήση ενός αθροίσματος ελέγχου (με αρνητικό πρόσημο). Αν ο αριθμός των τιμών που πρέπει να κρυπτογραφηθούν είναι $l_1 = \left\lceil \frac{n}{\log(w)} \right\rceil$, τότε το άθροισμα ελέγχου υπολογίζεται ως $c = \sum_{i=1}^{l1}(w - 1 - m_i)$. Έχοντας ως βάση τη τιμή w, το άθροισμα ελέγχου απαιτεί $l_2 = \left\lfloor \frac{\log(l1\,(w-1))}{\log(w)} \right\rfloor$ τιμές, οι οποίες κρυπτογραφούνται με τον ίδιο τρόπο όπως το αρχικό μήνυμα. Αυτό σημαίνει ότι τα κλειδιά πρέπει να αποτελούνται από l = l1 + l2 τιμές s_i και p_i, προκειμένου να καλύψουν το μήνυμα και το άθροισμα ελέγχου (Merkle 1990: 13).

Το παραπάνω σχήμα ονομάζεται συνάρτηση του Winternitz δημιουργίας κλειδιών μίας χρήσης. Οι παρακάτω αλγόριθμοι επεξηγούν τα βήματα δημιουργίας κλειδιού, κρυπτογράφησης και επαλήθευσης, ενώ ο Πίνακας 3, αναλύει διαφορετικούς συνδυασμούς παραμέτρων με τη μέθοδο για τη δημιουργία κλειδιών.

1: **for** $i \in \{1, \ldots, \ell\}$ **do**
2: $\quad s_i \xleftarrow{\$} \{0,1\}^k$
3: $\quad p_i \leftarrow F^{w-1}(s_{i,j})$
4: **end for**
5: **return** pk = $(p_1, \ldots, p_\ell)$, sk = $(s_1, \ldots, s_\ell)$

Εικόνα 7. Διαδικασία KeyGen με τη μέθοδο του Winternitz(Rijneveld 2019).

1: $(m_1, \ldots, m_{\ell_1}) = m$ ▷ Convert m to base-w, s.t. $m_i \in \{0, \ldots, w-1\}$
2: **for** $i \in \{1, \ldots, \ell_1\}$ **do**
3: $\quad \sigma_i = F^{m_i}(s_i)$
4: **end for**
5: $c = \Sigma_{i=1}^{\ell_1}(w - 1 - m_i)$
6: $(c_1, \ldots, c_{\ell_2}) = c$ ▷ Convert c to base-w, s.t. $c_i \in \{0, \ldots, w-1\}$
7: **for** $i \in \{1, \ldots, \ell_2\}$ **do**
8: $\quad \sigma_{\ell_1+i} = F^{c_i}(s_{\ell_1+i})$
9: **end for**
10: **return** $\sigma = (\sigma_1, \ldots, \sigma_\ell)$

Εικόνα 8. Διαδικασία Sign με τη μέθοδο του Winternitz(Rijneveld 2019).

1: $(m_1, \ldots, m_{\ell_1}) = m$ ▷ Convert m to base-w, s.t. $m_i \in \{0, \ldots, w-1\}$
2: **for** $i \in \{1, \ldots, \ell_1\}$ **do**
3: **if** $F^{w-1-m_i}(\sigma_i) \neq p_i$ **then**
4: **return** False
5: **end if**
6: **end for**
7: $c \leftarrow \sum_{i=1}^{\ell_1}(w-1-m_i)$
8: $(c_1, \ldots, c_{\ell_2}) = c$ ▷ Convert c to base-w, s.t. $c_i \in \{0, \ldots, w-1\}$
9: **for** $i \in \{1, \ldots, \ell_2\}$ **do**
10: **if** $F^{w-1-c_i}(\sigma_{\ell_1+i}) \neq p_{\ell_1+i}$ **then**
11: **return** False
12: **end if**
13: **end for**
14: **return** True

Εικόνα 9. Διαδικασία Verify με τη μέθοδο του Winternitz(Rijneveld 2019).

n	**k**	**w**	**l1**	**l2**	**l**	**Bytes**
128	128	4	64	4	68	1088
128	128	16	32	3	35	560
128	128	256	16	2	18	288
192	192	4	96	5	101	2424
192	192	16	48	3	51	1224
192	192	256	24	2	26	624
256	256	4	128	5	133	4256
256	256	16	64	3	67	2144
256	256	256	32	2	34	1088

Πίνακας 3. Πίνακας βασικών παραμέτρων της συνάρτησης Winternitz (Rijneveld 2019).

3.4.3 W-OTS+

Το σχήμα του W-OTS+ είναι επέκταση του βασικού σχήματος Winternitz και αποτελεί μία καλή βάση για την δημιουργία κάποιου νέου σχήματος μετα-κβαντικής κρυπτογράφησης. Στο W-OTS+ σχήμα η μονόδρομη συνάρτηση που εκτελείται σε διαδοχικές επαναλήψεις στα τυπικά σχήματα κατακερματισμού, αντικαθίσταται από μία διαδικασία τυχαίας προσπέλασης μέσα σε μία σειρά συναρτήσεων κατακερματισμού.

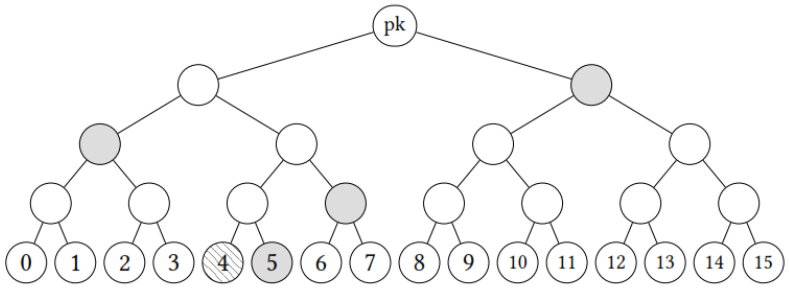
Η κεντρική ιδέα του αλγορίθμου είναι να μετατραπεί το αρχικό μήνυμα M σε μία δυαδική αναπαράσταση ενός φυσικού αριθμού και μετά να κανονικοποιήσει την αναπαράσταση του σε μία νέα αναπαράσταση με βάση τη παράμετρο w. Έχει το πλεονέκτημα ότι επιτυγχάνει μία καλή αντιστάθμιση ανάμεσα στην απόδοση του σχήματος και το μέγεθος του κλειδιού (το οποίο καθορίζεται από τη παράμετρο w). Ορίζει επίσης το μέγεθος των τμημάτων του συνολικού μηνύματος που κρυπτογραφούνται κάθε φορά.

3.4.4 Σύστημα Υπογραφών πολλαπλών χρήσεων (Merkle Trees)

Όλα τα παραπάνω σχήματα δημιουργίας μυστικού κλειδιού εφαρμόζονται για την δημιουργία κλειδιών μιας χρήσης (one time). Αυτό δεν είναι πάντα ωφέλιμο στη πράξη. Εναλλακτικά εξετάζονται σχήματα που επιτρέπουν τη δημιουργία κλειδιών πολλαπλών χρήσεων. Η διαδικασία βασίζεται στην αντίστοιχη διαδικασία δημιουργίας κλειδιών μίας μόνο χρήσης με αρκετές ωστόσο παραλλαγές. Για παράδειγμα, ο μέγιστος αριθμός κλειδιών είναι προκαθορισμένος και το μυστικό κλειδί συνέχεια μεταβάλλεται καθώς χρησιμοποιείται στη δημιουργία υπογραφών.

Υποθέτοντας ότι ο εγγεγραμμένος χρήστης έχει πρόσβαση στο απαιτούμενο δημόσιο κλειδί, μια υπογραφή μίας μόνο χρήσης παρέχει την ίδια λειτουργικότητα που περιμένουμε από οποιοδήποτε σχήμα ψηφιακής υπογραφής. Αυτή η υπόθεση παρουσιάζει το εξής πρόβλημα: αφού ένα ζεύγος κλειδιών μπορεί να χρησιμοποιηθεί μόνο μία φορά, κάθε λειτουργία επαλήθευσης μίας υπογραφής απαιτεί ένα διαφορετικό δημόσιο κλειδί. Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί με την ενσωμάτωση του δημόσιου κλειδιού στην υπογραφή μίας χρήσης. Σε αυτή τη περίπτωση παρουσιάζεται ένα διαφορετικό πρόβλημα: αφορά την επαλήθευση των συνημμένων δημόσιων κλειδιών.

Ο Merkle λύνει το πρόβλημα με την δημιουργία ενός δέντρου (Merkle Tree). Ένα δένδρο Merkle με φύλλα 2^k είναι σε θέση να υπογράψει k μηνύματα μήκους 1 (ή, ισοδύναμα, ένα μήνυμα μήκους μέχρι k bits). Η ρίζα του δέντρου είναι το δημόσιο κλειδί, ενώ τα φύλλα του δέντρου αποτελούν το ιδιωτικό κλειδί. Η λειτουργία του είναι απλή, ξεκινάει από τον κόμβο φύλλου που αντιστοιχεί σε μια δεδομένη υπογραφή και «προσθέτει» μόνο τους κόμβους που απαιτούνται για την διάσχιση προς τα πάνω και προς το δημόσιο κλειδί pk. Δεδομένου του κόμβου φύλλων που περιέχει το δημόσιο κλειδί μίας χρήσης pkj που αντιστοιχεί σε μια ειδική υπογραφή σ' αυτό σημαίνει ότι απαιτούμε από τον συγγενικό κόμβο-φύλλο να κατασκευάσει τον γονέα του. Κατόπιν, απαιτούμε από τον συγγενικό κόμβο να κατασκευάσει τον «πατέρα» του, κ.ο.κ. Σημειώνουμε ότι απαιτείται ακριβώς ένας κόμβος ανά επίπεδο του δέντρου - με άλλα λόγια, ο αριθμός των απαιτούμενων κόμβων είναι λογαριθμικός επί του αριθμού των κόμβων φύλλων, δηλαδή τον πιθανό αριθμό υπογραφών. Αυτοί οι κόμβοι αποτελούν το 'Μονοπάτι ταυτοποίησης' (Εικόνα 10).



Εικόνα 10. Ένα δυαδικό δέντρο κατακερματισμού με p = 16. Οι χρωματισμένοι κόμβοι περιλαμβάνονται στη διαδρομή επαλήθευσης που πιστοποιεί το δημόσιο κλειδί pk4 (Rijneveld 2019).

Ωστόσο το σύστημα υπογραφής πολλαπλών χρήσεων, όπως το δεντρικό σχήμα υπογραφών του Merkle, έχει τον εξής σημαντικό περιορισμό. Επειδή σε κάθε σχήμα μίας μόνο χρήσης (One Time Signature -OTS) τα ζεύγη κλειδιών μπορούν να χρησιμοποιηθούν μόνο μία φορά, ο υπογράφων πρέπει να παρακολουθεί ποια ζεύγη κλειδιών έχουν ήδη χρησιμοποιηθεί. Η πιο απλή προσέγγιση για να επιτευχθεί αυτό, είναι να συμπεριληφθεί σε ένα ευρετήριο το μυστικό κλειδί, υποδεικνύοντας ποιο ζεύγος κλειδιών πρόκειται να χρησιμοποιηθεί, ως επόμενο. Έτσι, ένα μυστικό κλειδί είναι του τύπου $sk_j = (j; sk_0, \ldots, sk_{p-1})$.

Ένα σχήμα ψηφιακών υπογραφών πολλαπλών χρήσεων (many time digital signature) είναι μία πλειάδα αλγορίθμων (KeyGen, Sign, Verify) που έχοντας υπόψη ένα μέγιστο αριθμό υπογραφών p ορίζεται ως εξής:

- Ο αλγόριθμος KeyGen είναι ένας πιθανοτικός αλγόριθμος και εξάγει ένα ζεύγος από δημόσιο κλειδί pk και ένα μυστικό κλειδί sk_0.

- Ο αλγόριθμος κατακερματισμού (δημιουργίας υπογραφών) Sign βασίζεται επίσης στη θεωρία πιθανοτήτων και λαμβάνει ως είσοδο ένα μήνυμα m και ένα μυστικό κλειδί sk_i για να δημιουργήσει την υπογραφή σ και ένα ενημερωμένο μυστικό κλειδί sk_{i+1}, εφόσον i+1 <p, αλλιώς αποτυγχάνει.

- Ο αλγόριθμος επαλήθευσης Verify είναι ένας ντετερμινιστικός αλγόριθμος που λαμβάνει ως είσοδο ένα μήνυμα m, μια υπογραφή σ και ένα δημόσιο κλειδί pk. Παράγει την τιμή True υποδηλώνοντας ότι η υπογραφή είναι αποδεκτή, ή False για να δηλώσει την απόρριψή της .

for $i \in \{0, \ldots, 2^h - 1\}$ **do**
    $pk_i, sk_i \leftarrow OTS.\text{KeyGen}()$
end for
$node_{i,j} \leftarrow \text{Merkle.BuildTree}(pk_0, \ldots, pk_{2^h-1})$ ▷ Let $0 \le i \le h$ and $0 \le j < 2^{h-i}$
$\text{pk} \leftarrow node_{h,0}$
return $\text{pk}, \text{sk}_0 = (0; sk_0, \ldots, sk_{2^h-1})$

Εικόνα 11. Διαδικασία KeyGen(Rijneveld 2019).

for $j \in \{0, \ldots, 2^h - 1\}$ **do**
    $node_{0,j} \leftarrow \text{leaf}_j$
end for
for $i \in \{1, \ldots, h\}$ **do**
    for $j \in \{0, \ldots, 2^{h-i} - 1\}$ **do**
        $node_{i,j} \leftarrow H(node_{i-1,2j}, node_{i-1,2j+1})$
    end for
end for
return $node_{i,j}$ **for** $i \in \{0, \ldots, h\}, j \in \{0, \ldots, 2^{h-i} - 1\}$

Εικόνα 12. Διαδικασία BuildTree (Rijneveld 2019).

$(idx; sk_0, \ldots, sk_{2^h-1}) = \mathsf{sk}_{idx}$
if $idx > 2^h - 1$ **then**
    return Fail
end if
$\mathsf{sk}_{idx+1} \leftarrow (idx + 1; sk_0, \ldots, sk_{2^h-1})$
$\sigma_{OTS} \leftarrow OTS.\mathsf{Sign}(m, sk_{idx})$
for $j \in \{0, \ldots, 2^h - 1\}$ **do**
    $pk_j \leftarrow OTS.\mathsf{RecoverPK}(sk_j)$
end for
$node_{i,j} \leftarrow \mathsf{Merkle.BuildTree}(pk_0, \ldots, pk_{2^h-1})$ ▷ Let $0 \le i \le h$ and $0 \le j < 2^{h-i}$
for $i \in \{0, \ldots, h - 1\}$ **do**
    $auth_i \leftarrow node_{i,\lfloor \frac{idx}{2^i} \rfloor \oplus 1}$
end for
$\sigma = (idx, pk_{idx}, \sigma_{OTS}, auth)$
return $\sigma, \mathsf{sk}_{idx+1}$

Εικόνα 13. Διαδικασία Sign (Rijneveld 2019).

$(idx, pk_{idx}, \sigma_{OTS}, auth) = \sigma$
if $\neg\, OTS.\mathsf{Verify}(m, \sigma_{OTS}, pk_{idx})$ **then**
    return False
end if
$node_0 \leftarrow pk_{idx}$
for $i \in \{0, \ldots, h - 1\}$ **do**
    if $\lfloor \frac{idx}{2^i} \rfloor \mod 2 = 0$ **then**
        $node_{i+1} \leftarrow H(node_i, auth_i)$
    else
        $node_{i+1} \leftarrow H(auth_i, node_i)$
    end if
end for
return $node_h \stackrel{?}{=} \mathsf{pk}$

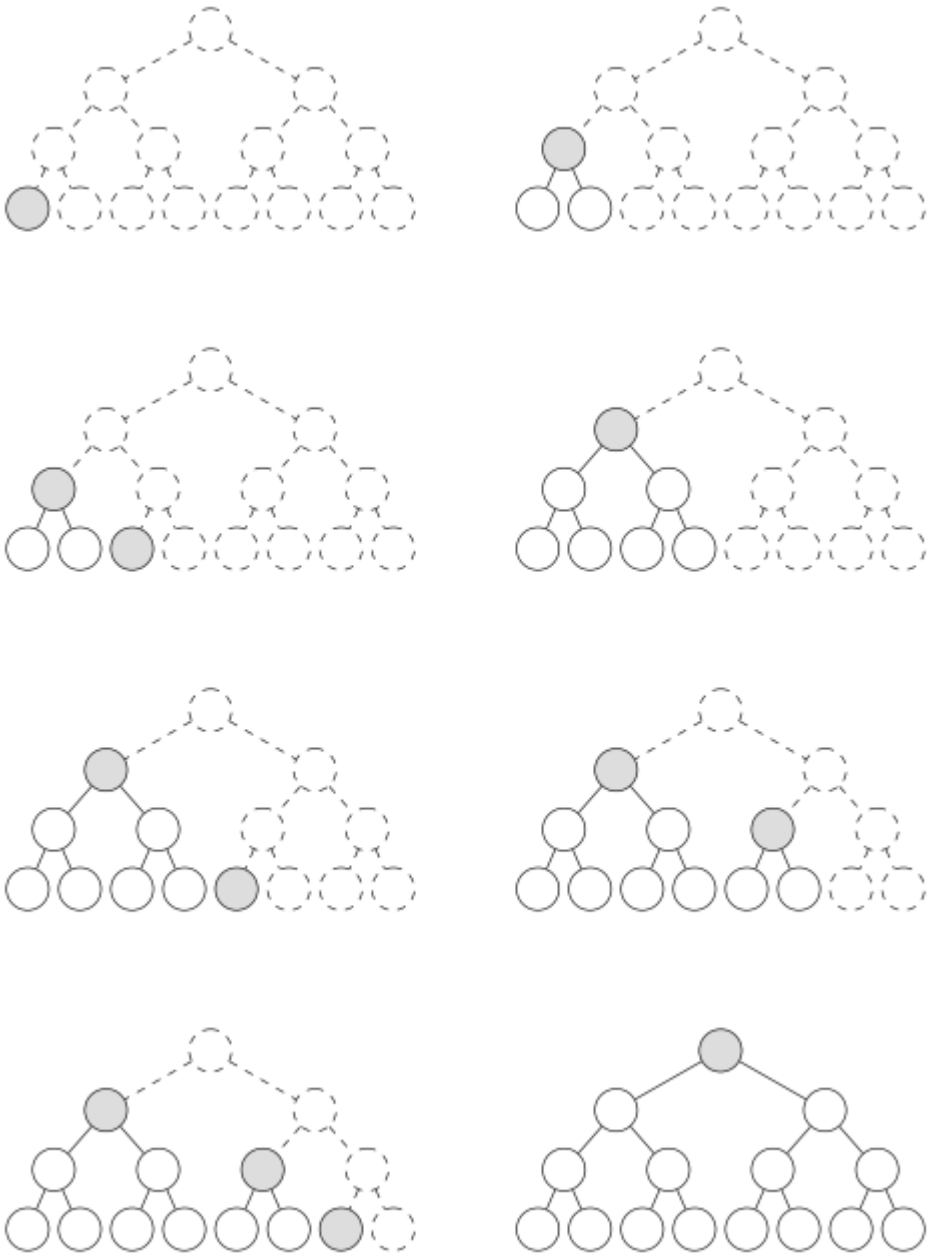
Εικόνα 14. Διαδικασία Verify (Rijneveld 2019).

TreeHash

Μία διαφοροποίηση των Merkle Trees είναι ο αλγόριθμος TreeHash. Ο αλγόριθμος αυτός αντί να χτίσει το δέντρο από επίπεδο σε επίπεδο, θεωρεί διαδοχικά έναν κόμβο-φύλλων κάθε φορά. Η βασική ιδέα είναι να δημιουργήσει υποδέντρα από τα φύλλα προς τα πάνω, σταδιακά συγχωνεύοντας τις ρίζες τους. Αυτό επιτρέπει να αποθηκεύονται μόνο οι κόμβοι-ρίζες του καθενός από τα υποδέντρα. Όλοι οι κόμβοι χρησιμοποιούνται ακριβώς μία φορά για να υπολογίσουν έναν πατέρα-κόμβο και έκτοτε δεν χρησιμοποιούνται ξανά.

Αρχικά, όταν ο πρώτος κόμβος-φύλλο προστίθεται, δεν έχουν οριστεί ακόμα υποδέντρα και οι κεφαλές τους, το νέο φύλλο θεωρείται ένα νέο υποδέντρο από μόνο του (n επίπεδο). Μόλις προστεθεί ο δεύτερος κόμβος, μπορεί να συγχωνευθεί με το τρέχον υποδέντρο για τη δημιουργία ενός νέου κόμβου στο n-1 επίπεδο. Το τρίτο φύλλο σχηματίζει ένα νέο υποδέντρο από μόνο του και πάλι, και ο τέταρτος κόμβος φύλλο προκαλεί όχι μία αλλά δύο συγχωνεύσεις, μπορεί να συγχωνευθεί με το τρίτο φύλλο στο n-1 επίπεδο, και με την ρίζα του υποδέντρου που έχει αναπτυχθεί ως στο n-2 επίπεδο. Αυτή η διαδικασία φυσικά συνεχίζεται μέχρι να έχουν προστεθεί όλοι οι κόμβοι-φύλλα (Εικόνα 15).

Μετά την ολοκλήρωση του αλγορίθμου και την επεξεργασία όλων των κόμβων φύλλων, το μόνο που παραμένει στη στοίβα είναι ο κόμβος ρίζα. Η παραπάνω διαδικασία είναι αρκετή για να αντικαταστήσει τον αλγόριθμο BuildTree , κατά τη δημιουργία του κλειδιού, αλλά όχι και για το στάδιο της υπογραφής. Εδώ, απαιτείται επίσης η διαδρομή επαλήθευσης που οδηγεί από έναν από τους κόμβους των φύλλων στη ρίζα. Αυτοί οι κόμβοι δημιουργούνται κατά τη διάρκεια του Treehash, αλλά δεν διατηρούνται εκ των προτέρων. Στο σημείο αυτό ο αλγόριθμος έχει τροποποιηθεί από τον Merkle για να αναγνωρίζει τους κόμβους καθώς παράγονται, αξιοποιώντας το γεγονός ότι η διαδρομή περιέχει ακριβώς έναν κόμβο σε κάθε επίπεδο του δέντρου.



Εικόνα 15. Αλγόριθμος Treehash, με ύψος h = 3. Οι τρέχουσες ρίζες των υποδέντρων είναι με γκρι χρώμα. Σε κάθε εκτέλεση εισάγεται ένας κόμβος-φύλλο και ενημερώνει τις ρίζες των υποδέντρων όπου είναι δυνατόν(Rijneveld 2019).

1: $stack \leftarrow []$
2: **for** $i \in \{0, \ldots, 2^h - 1\}$ **do**
3: $node \leftarrow leaf_i$ ▷ Leaves would typically be generated in-place.
4: **if** $idx = i \oplus 1$ **then** ▷ If this is the sibling of idx..
5: $auth_0 \leftarrow node$
6: **end if**
7: ▷ We use the abstract height() to keep track of node heights.
8: **while** height($stack$.head) = height($node$) **do**
9: $sibling \leftarrow stack$.pop()
10: **if** $\lfloor \frac{i}{2^{\text{height}(node)}} \rfloor \mod 2 = 0$ **then**
11: $node \leftarrow H(node, sibling)$
12: **else**
13: $node \leftarrow H(sibling, node)$
14: **end if**
15: **if** $\lfloor \frac{idx}{2^{\text{height}(node)}} \rfloor = \lfloor \frac{i}{2^{\text{height}(node)}} \rfloor \oplus 1$ **then**
16: $auth_{\text{height}(node)} \leftarrow node$
17: **end if**
18: **end while**
19: $stack$.push($node$)
20: **end for**
21: $root \leftarrow stack$.head
22: **return** $root, path = (auth_0, \ldots, auth_{h-1})$

Εικόνα 16. Αλγόριθμος Treehash (Rijneveld 2019).

3.4.5 XMSS

Όπως το σχήμα κρυπτογράφησης του Merkle, το σχήμα XMSS (eXtended Merkle Signature Scheme) χρησιμοποιεί ένα σχήμα δημιουργίας υπογραφής μίας μόνο χρήσης (OTS) που μπορεί να κρυπτογραφήσει μόνο ένα μήνυμα με ένα κλειδί. Για να ξεπεραστεί ο περιορισμός αυτός, χρησιμοποιείται ένα δέντρο κατακερματισμού, για να μειωθεί η ανάγκη ταυτοποίησης πολλαπλών κλειδιών ελέγχου OTS σε ένα δημόσιο κλειδί XMSS. Για να ελαχιστοποιηθούν οι απαιτήσεις αποθήκευσης, χρησιμοποιούνται γεννήτριες ψευδοτυχαίων αριθμών (PRG). Δημιουργούν τα κλειδιά υπογραφής OTS όπως απαιτείται (Buchmann, Dahmen και Hülsing 2011).

- $n \in N$, η παράμετρος ασφάλειας
- $w \in N, w > 1$, η παράμετρος Winternitz
- $m \in N$, το μήκος του μηνύματος (bits)

- Η ∈ N το ύψος του δέντρου, το σχήμα XMSS επιτρέπει τη δημιουργία 2^H υπογραφών με ένα ζεύγος κλειδιών

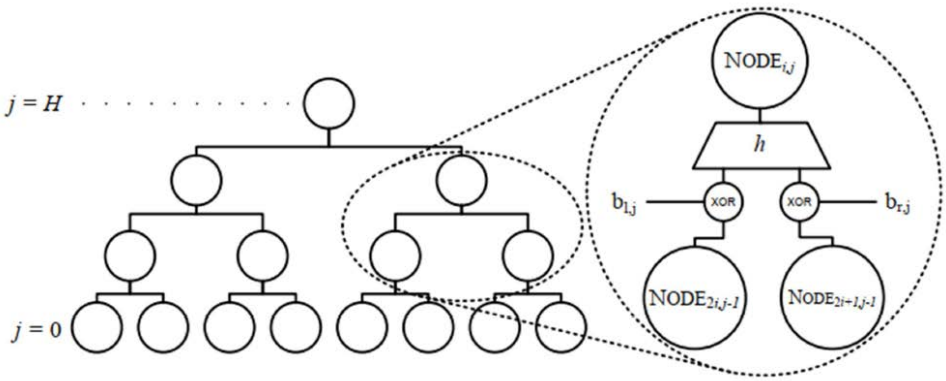
- h_k , μία συνάρτηση κατακερματισμού, που επιλέγεται τυχαία με ομοιόμορφη κατανομή από την ομάδα συναρτήσεων $H_n = \{h_k : \{0,1\}^{2n} \rightarrow \{0,1\}^n | K \in \{0,1\}^n\}$

- $x \in \{0,1\}^n$, επιλέγεται τυχαία με ομοιόμορφη κατανομή. Η συμβολοσειρά x χρησιμοποιείται για τη δημιουργία των κλειδιών επαλήθευσης μίας μόνο χρήσης.

Δημιουργία XMSS κλειδιού

Το σχήμα XMSS είναι κατά κάποιον τρόπο μια τροποποιημένη έκδοση μίας δεντρικής μορφής Merkle, όπου ωστόσο τα παραγόμενα φύλλα του υπολογίζονται με βάση το σχήμα Winternitz-OTS+ (W-OTS+). Ορίζοντας τη τιμή Η, ο αλγόριθμος δημιουργίας XMSS κλειδιού αρχικά δημιουργεί 2^H W-OTS+ ζεύγη κλειδιών ($sk_{W\text{-}OTS+,i}$, $pk_{W\text{-}OTS+,i}$) όπου $0 \leq i \leq 2^H$. Τα δημόσια κλειδιά W-OTS+ χρησιμοποιούνται κατόπιν για τη δημιουργία του XMSS δέντρου. Οι εσωτερικοί κόμβοι του XMSS δέντρου υπολογίζονται με βάση την φόρμουλα (Kannwischer, Genêt, Butin,Krämer και Buchmann 2018):

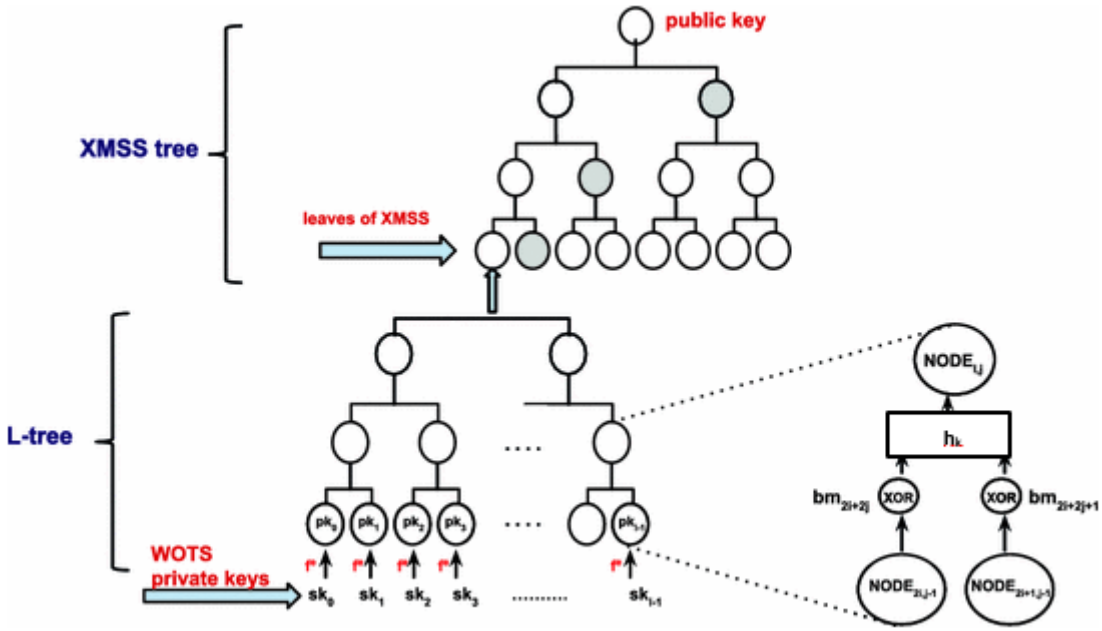
$NODE_{i,j} = h_k((NODE_{2i,j-1} \oplus b_{l,j}) \| (NODE_{2i+1,j-1} \oplus b_{r,j}))$ (Buchmann, Dahmen και Hülsing 2011: 117).



Εικόνα 17. Δημιουργία κλειδιού – δεντρικό σχήμα XMSS με βάση το σχήμα W-OTS+(Buchmann, Dahmen και Hülsing 2011: 117).

Όπου $b_{l,j}$ και $b_{r,j}$ αποθηκεύουν τυχαίες τιμές που παράγονται από μία γεννήτρια τυχαίων αριθμών (διαδικασία PRNG). Η αρχική κατάσταση αυτής της γεννήτριας αποτελεί επίσης τμήμα του δημόσιου κλειδιού, μαζί με τη ρίζα του δέντρου (αφού τη χρειάζεται όποιος επικυρώνει την υπογραφή, προκειμένου να αναπαράγει τις σωστές τυχαίες τιμές). Κάθε φύλλο του XMSS

δέντρου $u_0[i]$ παράγεται από τα αντίστοιχα δημόσια κλειδιά του W-OTS+ σχήματος χρησιμοποιώντας ένα άλλο XMSS δέντρο, το οποίο ονομάζεται L-tree. Με άλλα λόγια, τα φύλλα του XMSS δέντρου είναι ρίζες των L-δέντρων, τα οποία με τη σειρά τους έχουν ως φύλλα τα W-OTS+ δημόσια κλειδιά. Το υποδέντρο (L-tree) έχει αντίστοιχη δομή με το κυρίως XMSS δέντρο, συμπιέζει ένα n x l bit δημόσιο κλειδί σε μία τιμή που αποθηκεύεται σε n bits. Επαναλαμβάνεται το ίδιο για τους εσωτερικούς κόμβους.



Εικόνα 18. Δημιουργία XMSS δέντρου – χρήση L-υποδέντρων (Karina και López 2015).

Το μυστικό κλειδί του XMSS σχήματος είναι η σύνθεση όλων των μυστικών W-OTS+ κλειδιών $sk_{W-OTS+,i}$ και το ευρετήριο s από το επόμενο μη χρησιμοποιούμενο φύλλο.

Δημιουργία υπογραφής

Έχοντας το μυστικό κλειδί (sk_{W-OTS+}, s) και την δυαδική αναπαράσταση ενός μηνύματος M, $D \in \{0,1\}^n$, το σχήμα αρχικά υπολογίζει την υπογραφή σ_{W-OTS+} για το μήνυμα M χρησιμοποιώντας το μυστικό κλειδί sk_{W-OTS+}, s. Στη συνέχεια απαιτείται να αυξηθεί το s στο μυστικό κλειδί (δηλαδή ο δείκτης που υποδηλώνει ποιο ιδιωτικό κλειδί έχει ήδη χρησιμοποιηθεί) για να εξασφαλιστεί ότι το συγκεκριμένο ζεύγος κλειδιών, το οποίο είναι μόνο μίας χρήσης, δεν θα επαναχρησιμοποιηθεί. Πέρα από το δημόσιο κλειδί, η διαδικασία επαλήθευσης (verifier) απαιτεί αρκετούς κόμβους του XMSS δέντρου για να συνθέσει εκ νέου τη ρίζα του δέντρου κατακερματισμού (μέσα από ένα «μονοπάτι» κόμβων, κατ' αναλογία με ένα απλό δέντρο

Merkle). Αυτό επιτυγχάνεται προσθέτοντας το μονοπάτι ταυτοποίησης $A_s = (a_0, \ldots, a_{h-1})$ στην υπογραφή. Το μονοπάτι αυτό περιέχει ένα κόμβο σε κάθε επίπεδο του δέντρου κατακερματισμού . Οι κόμβοι a_h είναι είτε αριστεροί ή δεξιοί γείτονες των κόμβων στο μονοπάτι από το u[s] στο $u_h[0]$.

$$a_h = \begin{cases} v_h[s/2^h - 1], & \text{if } \lfloor s/2^h \rfloor \equiv 1 \pmod 2 \\ v_h[s/2^h + 1], & \text{if } \lfloor s/2^h \rfloor \equiv 0 \pmod 2 \end{cases}$$

Επομένως η XMSS υπογραφή είναι $\sigma = (s, \sigma_{W-OTS+}, pk_{W-OTS+}, s, A_s)$ (Kannwischer, Genêt, Butin,Krämer και Buchmann 2018).

3.5 Αξιολόγηση χαρακτηριστικών ασφάλειας & υλοποίησης

Στην παραπάνω ενότητα αναφερθήκαμε στα διάφορα σχήματα κατακερματισμού, τα πλεονεκτήματα και μειονεκτήματά τους. Σαφώς το κύριο μειονέκτημα της δημιουργίας υπογραφής μίας μόνο χρήσης (OTP), ειδικά στη περίπτωση του σχήματος Lamport, είναι ότι εάν η ίδια υπογραφή χρησιμοποιηθεί δύο φορές, για δύο διαφορετικά μηνύματα, ένας εισβολέας θα είναι σε θέση να δημιουργήσει ένα άλλο μήνυμα με μία έγκυρη υπογραφή. Επιπρόσθετα, το μέγεθος των υπογραφών και των κλειδιών στο σχήμα Lamport είναι αρκετά μεγάλα.

Στη προσέγγιση του σχήματος Merkle υπάρχει το μειονέκτημα της αύξησης του μεγέθους της υπογραφής κατά δύο παράγοντες σε σχέση με το προηγούμενο σχήμα. Ωστόσο, το κύριο δημόσιο κλειδί για το σύστημα είναι τώρα μόνο μία συγκεκριμένη τιμή κατακερματισμού, η οποία κάνει αυτή την προσέγγιση να κλιμακώνεται με ακόμα περισσότερη ευελιξία και σαφήνεια έναντι του προηγούμενου σχήματος. Αυτό σημαίνει ότι όλα τα διάφορα μυστικά κλειδιά μπορούν να ληφθούν μέσω μίας κεντρικής γεννήτριας κρυπτογραφημένων ψευδοτυχαίων αριθμών. Η γεννήτρια μπορεί να δημιουργήσει ένα αρκετά μεγάλο αριθμό (τυχαίων) δυαδικών ψηφίων από ένα μικρό εύρος αριθμών (seed).

Το σχήμα του Winternitz (W-OTS) εξετάζει όχι μόνο τη παράμετρο του μεγέθους της υπογραφής αλλά και την παράμετρο του χρόνου επαλήθευσης. Πάνω σε αυτό το σχήμα βασίστηκαν επόμενες επεκτάσεις όπως το σχήμα XMSS, XNYSS[8] (το οποίο δεν μελετάται εδώ) κοκ.

Για παράδειγμα, τα περισσότερα, αν όχι όλα, τα σχήματα κατακερματισμού κάνουν χρήση των δέντρων Merkle. Όπως είδαμε στη σχετική ενότητα, ο μέγιστος αριθμός μηνυμάτων που το βασικό σχήμα Merkle μπορεί να κρυπτογραφήσει είναι 2^h, όπου h είναι το ύψος του δέντρου. Επίσης, όλα τα φύλλα (κλειδιά) πρέπει να υπολογίζονται κατά τη διάρκεια της δημιουργίας του εκάστοτε κλειδιού για να σχηματιστεί η ρίζα. Λόγω των παραπάνω, για την κατασκευή ενός δέντρου ύψους h = 40, η δημιουργία κλειδιών θεωρείται ανέφικτη, διότι θα πρέπει να υπολογιστούν 2^{40} κλειδιά μίας χρήσης και κάθε κλειδί OTS εσωτερικά απαιτεί πολλές επικλήσεις κατακερματισμού (δηλ., 512 επικλήσεις κατακερματισμού στο σχήμα Lamport ή 67 για το σχήμα WOTS (w = 16) κατά τη χρήση του αλγορίθμου SHA256). Το τέχνασμα για τη διατήρηση του χρόνου δημιουργίας κλειδιών σε μία ρεαλιστική βάση, ενώ να επιτρέπεται ένας μεγάλος αριθμός υπογραφών, είναι να χρησιμοποιείται ένα δέντρο πολλαπλών επιπέδων (Merkle Trees, XMSS) (Chalkias, Brown, Hearn, Lillehagen, Nitto και Schroeter 2018).

Καθώς έχουν προταθεί και ερευνώνται διαφορετικά πρότυπα μετα-κβαντικών σχημάτων κρυπτογράφησης, αυτά διαφέρουν σημαντικά και αναφορικά με τους πόρους που απαιτούν. Ο Πίνακας 4, δείχνει μια σύγκριση του μεγέθους των κλειδιών και των μηνυμάτων των διαφορετικών σχημάτων. Σίγουρα όλα τα μετα-κβαντικά σχήματα απαιτούν μεγαλύτερου μεγέθους δημόσια κλειδιά και μεγαλύτερου μεγέθους σχήματα κρυπτογράφησης/ κωδικοποίησης/υπογραφής έναντι των παραδοσιακών σχημάτων. Ωστόσο, αυτό το κόστος αποσβένεται από το γεγονός ότι τα συστήματα γίνονται πιο ασφαλή από τις επιθέσεις που χρησιμοποιούν κβαντικούς υπολογιστές. Τα σχήματα RSA, ECC και DH δεν μπορούν πλέον να εφαρμοστούν σε σενάρια που λαμβάνουν υπόψη τη χρήση κβαντικών υπολογιστών(Niederhagen, Waidner 2017).

Σε πολλές περιπτώσεις, η προσπάθεια να μειωθούν οι απαιτήσεις σε πόρους με την εισαγωγή κάποιας ενδιάμεσης δομής στα συστήματα είχε ως αποτέλεσμα τη δημιουργία κενών που έγιναν στόχος πετυχημένων επιθέσεων. Επομένως, για ορισμένα σχήματα, η μείωση του μεγέθους του δημόσιου κλειδιού ή του μεγέθους των ενδιάμεσων δεδομένων ενδέχεται να είναι αδύνατη.

[8] eXtended Naor-Yung Signature Scheme

Αναμφισβήτητα, τα πιο αξιόπιστα συστήματα υπογραφής δημόσιου κλειδιού βασίζονται στα σχήματα κατακερματισμού. Απαιτούν μικρά δημόσια κλειδιά των 64-1.056 bytes που είναι στο εύρος των παραδοσιακών σχημάτων RSA και ECC. Ωστόσο, το μέγεθος των υπογραφών βάσει κατακερματισμού είναι 2.5-41 kB, το οποίο είναι πολύ μεγαλύτερο από τα μεγέθη των παραδοσιακών σχημάτων. Τα συστήματα πολλαπλών μεταβλητών είναι ακόμα σε ερευνητικό στάδιο, ωστόσο μέχρι τώρα δείχνει να απαιτούν μεγέθη δημόσιου κλειδιού 500 kB έως 1 MB, τα οποία είναι πολύ μεγαλύτερα από τα κλασσικά συστήματα αλλά έχουν πολύ μικρά μεγέθη υπογραφής.

Στα δε σχήματα κρυπτογράφησης δημόσιου κλειδιού, υπάρχει ισχυρή εμπιστοσύνη στα McEliece και Niederreiter συστήματα κρυπτογράφησης (χρησιμοποιώντας τους λεγόμενους κώδικες Goppa). Το μέγεθος των δημόσιων κλειδιών τους είναι περίπου 1 MB και συνεπώς πολύ μεγάλο σε σύγκριση με τα κλασσικά σχήματα. Το μέγεθος του κώδικα κρυπτογράφησης (χρησιμοποιείται, π.χ. για κλειδί ενθυλάκωσης) είναι μόνο περίπου 190 bytes το οποίο είναι στο εύρος των κλασικών σχημάτων. Τα συστήματα που βασίζονται σε πλέγματα (Lattices) είναι επίσης αρκετά ώριμα αλλά πιθανώς λιγότερο αξιόπιστα σε σύγκριση με τα συστήματα που βασίζονται σε κωδικοποίηση. Το σχήμα NTRUEncrpyt, για παράδειγμα, απαιτεί 1.5-2.0 kB συνολικά για κλειδιά και κώδικα κρυπτογράφησης.

Τέλος, τα συστήματα ανταλλαγής κλειδιών μπορούν εύκολα να δημιουργηθούν από τα σχήματα κρυπτογράφησης δημόσιου κλειδιού δημιουργώντας και μεταδίδοντας νέα κλειδιά για κάθε συναλλαγή. Τα ειδικά συστήματα ανταλλαγής κλειδιών παρέχουν καλύτερες επιδόσεις σε σχέση με το χρόνο δημιουργίας κλειδιού και τις απαιτήσεις σε χωρητικότητα του δικτύου (bandwidth). Το νέο σχήμα NewHope (τύπου Lattice) απαιτεί την αποστολή πακέτων μεγέθους 2 kB σε σύγκριση με μόνο 32-64 bytes του κλασσικού αλγορίθμου ECDH.

Σχήματα	Μέγεθος Δημόσιου Κλειδιού (bytes)	Μέγεθος δεδομένων (προς ενδιάμεση αποθήκευση /μετάδοση) (bytes)
Υπογραφές Δημόσιου κλειδιού		
◆Σχήματα κατακερματισμού: XMSS SPHINCS	 64 1056	 2500-2820 41000
◆Σχήματα πολλαπλών μεταβλητών: HFEv	500000 – 1000000	25 – 32
Κρυπτογράφηση Δημοσίου Κλειδιου		
◆ Σχήματα βάσει κώδικα: McEliece	958482 – 1046739	187 – 194
◆Σχήματα βασισμένα σε πλέγμα: NTRUEncrypt	1495 – 2062	1495 – 2062
Ανταλλαγής Κλειδιών:		
◆Σχήματα βασισμένα σε πλέγμα: NewHope	-	1824 – 2048
◆Σχήματα ισογενών γραφημάτων SIDH	-	564

Πίνακας 4. Μέγεθος κλειδιών και ενδιάμεσων δεδομένων ανά μετα-κβαντικό σχήμα (Niederhagen, Waidner 2017).

Κεφάλαιο 4
Μελέτη περίπτωσης

4.1 Εισαγωγή

Στο κεφάλαιο αυτό εστιάζουμε σε εξέταση αλγορίθμων ψηφιακής υπογραφής σε περιβάλλον δοκιμών, προκειμένου να γίνει μία συγκριτική αποτίμηση, βάσει πειραματικών μετρήσεων, μεταξύ σημερινού συμβατικού αλγορίθμου ψηφιακής υπογραφής και αλγορίθμου μετα-κβαντικής κρυπτογραφίας.

4.2 Βιβλιοθήκες κρυπτογράφησης

Για την μελέτη περίπτωσης εντοπίστηκαν στην βιβλιογραφία, βιβλιοθήκες κρυπτογράφησης ανοικτού κώδικα (open source) που μπορούν να χρησιμοποιηθούν στην ανάπτυξη των δοκιμών που θα πραγματοποιηθούν. Ορισμένες από αυτές παρουσιάζονται στον Πίνακα 5.

Βιβλιοθήκη	**Εταιρεία**	**Γλώσσα Προγραμματισμού**
Botan	Jack Lloyd	C++
Bouncy Castle	Legion of the Bouncy Castle Inc.	Java, C#
cryptlib	Peter Gutmann	C
Crypto++	The Crypto++ project	C++
GnuTLS	Nikos Mavrogiannopoulos, Simon Josefsson	C
Libgcrypt	GnuPG community and g10code	C
libsodium	Frank Denis	C
NaCl	Daniel J. Bernstein, Tanja Lange, Peter Schwabe	C
Nettle		C
Network Security Services (NSS)	Mozilla	C

Πίνακας 5. Βιβλιοθήκες κρυπτογράφησης

Η συντριπτική πλειοψηφία των βιβλιοθηκών κρυπτογράφησης, ο κώδικάς τους είναι γραμμένος σε γλώσσα προγραμματισμού C. Όμως πλέον η πλειονότητα των εμπορικών εφαρμογών αναπτύσσονται στην γλώσσα προγραμματισμού Java. Επίσης η γλώσσα προγραμματισμού Java μπορεί να «τρέξει» σε οποιοδήποτε λειτουργικό σύστημα και πλατφόρμα. Έτσι, επιλέχτηκε αυτή η γλώσσα προγραμματισμού για τη μελέτη περίπτωσης και θα χρησιμοποιηθεί η βιβλιοθήκη κρυπτογράφησης Bouncy Castle.

4.3 Bouncy Castle

Η Bouncy Castle βρίσκεται στην αγορά για περίπου 20 χρόνια προσφέροντας λύσεις κρυπτογράφησης. Οι προγραμματιστικές βιβλιοθήκες της (Bouncy Castle Crypto APIs) παρέχονται από έναν Αυστραλιανό φιλανθρωπικό οργανισμό (Legion of the Bouncy Castle Inc.), ο οποίος υποστηρίζει και προσπαθεί για την επέκτασή τους. Αυτές οι βιβλιοθήκες είναι ανοιχτού κώδικα και αποτελούνται από τα ακόλουθα συστατικά στοιχεία:

- ένα ελαφρύ API κρυπτογράφησης για Java και C#,
- έναν πάροχο για το Java Cryptography Extension (JCE) και το Java Cryptography Architecture (JCA),
- έναν πάροχο για το Java Secure Socket Extension (JSSE),

- μια καθαρή υλοποίηση για το JCE 1.2.1.
- μια βιβλιοθήκη για την ανάγνωση και την γραφή κωδικοποιημένων ASN.1 αντικειμένων,
- ελαφριά APIs για TLS (RFC 2246, RFC 4346) και DTLS (RFC 6347/ RFC 4347),
- γεννήτριες για X.509 πιστοποιητικά έκδοσης 1 και έκδοσης 3, έκδοσης 2 CRLs, και PKCS12 αρχεία,
- γεννήτριες για πιστοποιητικά χαρακτηριστικών X.509 έκδοσης 2,
- γεννήτριες/επεξεργαστές για S/MIME και CMS (PKCS7/RFC 3852),
- γεννήτριες/επεξεργαστές για OCSP (RFC 2560),
- γεννήτριες/επεξεργαστές για TSP (RFC 3161 & RFC 5544),
- γεννήτριες/επεξεργαστές για CMP και CRMF (RFC 4210 & RFC 4211),
- γεννήτριες/επεξεργαστές για OpenPGP (RFC 4880),
- γεννήτριες/επεξεργαστές για Extended Access Control (EAC),
- γεννήτριες/επεξεργαστές για Data Validation and Certification Server (DVCS) - RFC 3029,
- γεννήτριες/επεξεργαστές για DNS-based Authentication of Named Entities (DANE),
- γεννήτριες/επεξεργαστές για RFC 7030 Enrollment over Secure Transport (EST) και
- μία πιστοποιημένη/υπογεγραμμένη έκδοση JAR αρχείου συμβατή με JDK 1.4-1.11 και το Sun JCE.

Προδιαγραφές και άδεια χρήσης, για την βιβλιοθήκη Bouncy Castle υπάρχουν στον ιστότοπο https//www.bouncycastle.org/specifications.html

4.4 Σενάρια δοκιμών

Σκοπός της παρούσας μελέτης περίπτωσης είναι να συγκρίνουμε την ταχύτητα ενός μετα-κβαντικού αλγορίθμου κρυπτογράφησης, σε σχέση με έναν παραδοσιακό αλγόριθμο υπογραφής. Για αυτό το λόγο θα χρησιμοποιήσουμε:

- τον XMSS ως μετα-κβαντικό αλγόριθμο κρυπτογράφησης,
- τον RSA ως έναν απλό αλγόριθμο υπογραφής,
- την Bouncy Castle ως βιβλιοθήκη κρυπτογράφησης,
- την Java ως γλώσσα προγραμματισμού

Για να μπορέσουμε να πραγματοποιήσουμε την σύγκριση αξιοποιήσαμε με κατάλληλες προσαρμογές μία εφαρμογή σε Java η οποία περιγράφεται στο «Παράρτημα Α' – Υλοποίηση εφαρμογής» με βάση όσα αναφέρουμε παραπάνω. Αυτή η εφαρμογή μπορεί να εκτελεστεί σε οποιοδήποτε υπολογιστή και να δώσει αποτελέσματα τα οποία παρουσιάζονται στο «Παράρτημα Β – Αναλυτικά αποτελέσματα».

Συγκεκριμένα, οι αλγόριθμοι RSA και XMSS, θα συγκριθούν ως προς το χρόνο που χρειάζονται για να υπογράψουν το ίδιο μήνυμα. Τα μηνύματα που θα πραγματοποιηθούν τα τεστ θα είναι τρία. Ο εκάστοτε αλγόριθμος θα έχει τις δικές του παραμέτρους σύμφωνα με την υλοποίηση της βιβλιοθήκης που χρησιμοποιούμε (Bouncy Castle):

- Ο αλγόριθμος RSA θα παίρνει ως είσοδο:
    - το μήκος κλειδιού,
    - τη συνάρτηση κατακερματισμού και
    - το μήνυμα

- Ο αλγόριθμος XMSS θα παίρνει ως είσοδο:
    - το ύψος του δέντρου,
    - τη συνάρτηση κατακερματισμού και
    - το μήνυμα

4.5 Αποτελέσματα

Τα μηνύματα που χρησιμοποιήσαμε για να κρυπτογραφηθούν από τους αλγόριθμους που επιλέξαμε παρουσιάζονται στον Πίνακα 6. Αυτά τα μηνύματα χρησιμοποιήθηκαν ως είσοδοι στις δοκιμές που κάναμε.

ID	Μήνυμα
#A	Hello world!
#B	My name is test operator and I want to do some tests!
#Γ	Coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).[7] The disease was first identified in December 2019 in Wuhan, the capital of China's Hubei province, and has since spread globally, resulting in the ongoing 2019–20 coronavirus pandemic.[8][9] Common symptoms include fever, cough, and shortness of breath.[4] Other symptoms may include fatigue, muscle pain, diarrhea, sore throat, loss of smell, and abdominal pain.[4][10][11] The time from exposure to onset of symptoms is typically around five days but may range from two to fourteen days.[4][12] While the majority of cases result in mild symptoms, some progress to viral pneumonia and multi-organ failure.[8][13] As of 15 April 2020, more than 1.99 million[6] cases have been reported across 210 countries and territories,[14] resulting in over 128,000 deaths. More than 500,000 people have recovered.[6] The virus is primarily spread between people during close contact,[a] often via small droplets produced by coughing,[b] sneezing, or talking.[5][15][17] While these droplets are produced when breathing out, they usually fall to the ground or onto surfaces rather than being infectious over long distances.[5][18][19] People may also become infected by touching a contaminated surface and then their face.[5][15] The virus can survive on surfaces for up to 72 hours.[20] It is most contagious during the first three days after the onset of symptoms, although spread may be possible before symptoms appear and in later stages of the disease.[21] The standard method of diagnosis is by real-time reverse transcription polymerase chain reaction (rRT-PCR) from a nasopharyngeal swab.[22] Chest CT imaging may also be helpful for diagnosis in individuals where there is a high suspicion of infection based on symptoms and risk factors; however, it is not recommended for routine screening.[23][24] Recommended measures to prevent infection include frequent hand washing, maintaining physical distance from others (especially from those with symptoms), covering coughs and sneezes with a tissue or inner elbow, and keeping unwashed hands away from the face.[25][26] The use of masks is recommended for those who suspect they have the virus and their caregivers.[27] Recommendations for mask use by the general public vary, with some authorities recommending against their use, some recommending their use, and others requiring their use.[28][29][30] Currently, there is no vaccine or specific antiviral treatment for COVID-19.[5] Management involves treatment of symptoms, supportive care, isolation, and experimental measures.[31] The World Health Organization (WHO) declared the 2019–20 coronavirus outbreak a Public Health Emergency of International Concern (PHEIC)[32][33] on 30 January 2020 and a pandemic on 11 March 2020.[9] Local transmission of the disease has been recorded in many countries across all six WHO regions.[34]

Πίνακας 6. Μηνύματα κρυπτογράφησης – Πηγή en.wikipedia.org/wiki/Coronavirus_disease_2019

Καθένας από τους δύο αλγόριθμους που χρησιμοποιήσαμε, πέρα από τα μηνύματα που παρουσιάζονται στον Πίνακας 6, είχε τις ακόλουθες εισόδους:

- αλγόριθμος RSA:
    - μήκος κλειδιού, θα αναφέρεται ως ΜΚ εφεξής:
        - 512
        - 2048
        - 4096
    - συνάρτηση κατακερματισμού, θα αναφέρεται ως ΣΚ εφεξής:
        - SHA-1
        - SHA-224
        - SHA-256

Σημειώνεται ότι οι δύο τελευταίες είναι εκφάνσεις της SHA-2 συνάρτησης κατακερματισμού.

- αλγόριθμος XMSS:
    - ύψος του δέντρου, θα αναφέρεται ως Υ εφεξής:
        - 6
        - 8
        - 10
    - συνάρτηση κατακερματισμού, θα αναφέρεται ως ΣΚ εφεξής:
        - Digest 256

- Digest 512

Σημειώνεται ότι και οι δύο ανωτέρω είναι εκφάνσεις της SHA-2 συνάρτησης κατακερματισμού.

Πραγματοποιήσαμε δοκιμές σε δύο υπολογιστές (φορητοί υπολογιστές) με διαφορετική υπολογιστική ισχύ. Στον Πίνακα 7, παραθέτουμε τα χαρακτηριστικά των υπολογιστών.

Α/Α	**Χαρακτηριστικό**	**Υπολογιστής #1**	**Υπολογιστής #2**
1	Λειτουργικό Σύστημα	Microsoft Windows 10 Home	Microsoft Windows 10 Enterprise
2	Επεξεργαστής	Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz, 2201 Mhz, 4 Core(s), 8 Logical Processor(s)	Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz, 1800 Mhz, 4 Core(s), 8 Logical Processor(s)
3	Μνήμη	8.00 GB	16,0 GB
4	Σκληρός δίσκος	Samsung SSD 850 EVO 250GB	WDC PC SN520 SDAPNUW-256G-1006
5	Έτος κτήσης	2011	2019

Πίνακας 7. Χαρακτηριστικά υπολογιστών

Με βάσει όσα αναφέρουμε στο Παράρτημα Α σχετικά με την υλοποίηση της εφαρμογής, εκτελέσαμε την εφαρμογή σε καθέναν από τους δύο υπολογιστές και παραθέτουμε συγκεντρωτικά τα αποτελέσματα στον Πίνακα 8, που ακολουθεί.

	Υπολογιστής	**#1**			**#2**		
	Μήνυμα	**#Α**	**#Β**	**#Γ**	**#Α**	**#Β**	**#Γ**
RSA	**ΜΚ: 512, ΣΚ: SHA1withRSA**	133	15	53	17	13	12
	ΜΚ: 2048, ΣΚ: SHA1withRSA	3311	4009	2018	356	333	773
	ΜΚ: 4096, ΣΚ: SHA1withRSA	17768	30910	11450	3362	11535	4758
	ΜΚ: 512, ΣΚ: SHA224withRSA	122	33	16	18	14	10
	ΜΚ: 2048, ΣΚ: SHA224withRSA	1426	258	975	486	372	288
	ΜΚ: 4096, ΣΚ: SHA224withRSA	8618	17987	12622	4964	15614	2457
	ΜΚ: 512, ΣΚ: SHA256withRSA	26	12	42	19	11	23
	ΜΚ: 2048, ΣΚ: SHA256withRSA	431	1628	1281	290	330	211
	ΜΚ: 4096, ΣΚ: SHA256withRSA	10528	10550	1816	4062	7738	3411
XMSS	**Υ: 6, ΣΚ: SHA256Digest**	453	517	1670	224	253	222
	Υ: 8, ΣΚ: SHA256Digest	3754	1973	1966	1085	801	883
	Υ: 10, ΣΚ: SHA256Digest	9884	9707	7510	3579	3213	3276
	Υ: 6, ΣΚ: SHA512Digest	1250	2075	2769	647	628	597
	Υ: 8, ΣΚ: SHA512Digest	6784	9733	7666	3063	2457	2359
	Υ: 10, ΣΚ: SHA512Digest	28837	30629	22257	9263	9044	9071

Πίνακας 8. Αποτελέσματα δοκιμών - μέτρησης χρόνου σε millisecond.

keysize (bytes)	algorithm	messageId	message length (bytes)	signature length (bytes)
2048	SHA1withRSA	#A	12	256
2048	SHA256withRSA	#A	12	256
2048	SHA224withRSA	#A	12	256
2048	SHA1withRSA	#B	53	256
2048	SHA256withRSA	#B	53	256
2048	SHA224withRSA	#B	53	256
2048	SHA1withRSA	#Γ	2977	256
2048	SHA256withRSA	#Γ	2977	256
2048	SHA224withRSA	#Γ	2977	256
4096	SHA224withRSA	#A	12	512
4096	SHA256withRSA	#A	12	512
4096	SHA1withRSA	#A	12	512
4096	SHA224withRSA	#B	53	512
4096	SHA256withRSA	#B	53	512
4096	SHA1withRSA	#B	53	512
4096	SHA224withRSA	#Γ	2977	512
4096	SHA256withRSA	#Γ	2977	512
4096	SHA1withRSA	#Γ	2977	512
512	SHA224withRSA	#A	12	64
512	SHA1withRSA	#A	12	64
512	SHA256withRSA	#A	12	64
512	SHA224withRSA	#B	53	64
512	SHA1withRSA	#B	53	64
512	SHA256withRSA	#B	53	64
512	SHA224withRSA	#Γ	2977	64
512	SHA1withRSA	#Γ	2977	64
512	SHA256withRSA	#Γ	2977	64

Πίνακας 9. Δοκιμές RSA - Μέγεθος υπογραφών

height	digest	messageId	message length (bytes)	signature length (bytes)
6	SHA-256	#A	12	2372
6	SHA-256	#B	53	2372
6	SHA-256	#Γ	2977	2372
8	SHA-256	#A	12	2436
8	SHA-256	#B	53	2436
8	SHA-256	#Γ	2977	2436
10	SHA-256	#A	12	2500
10	SHA-256	#B	53	2500
10	SHA-256	#Γ	2977	2500
6	SHA-512	#A	12	8836
6	SHA-512	#B	53	8836
6	SHA-512	#Γ	2977	8836
8	SHA-512	#A	12	8964
8	SHA-512	#B	53	8964

8	SHA-512	#Γ	2977	8964
10	SHA-512	#A	12	9092
10	SHA-512	#B	53	9092
10	SHA-512	#Γ	2977	9092

Πίνακας 10. Δοκιμές XMSS – Μέγεθος υπογραφών.

Για καθεμία από τις δύο ομάδες δοκιμών που διενεργήσαμε, δημιουργήσαμε και αντίστοιχα γραφήματα που παρουσιάζονται στις Εικόνες 19 και 20. Αυτό που παρατηρούμε συγκρίνοντας τα δύο γραφήματα είναι:

1. Ο υπολογιστής #1 δίνει καλύτερους χρόνους, σε σχέση με τον υπολογιστή #2. Συγκρίνοντας τα χαρακτηριστικά τους παρατηρούμε ότι ο υπολογιστής #1 έχει δυνατότερο επεξεργαστή (i7, 2.20GHz, 2201 Mhz, 4), από τον υπολογιστή #2, καθώς και νεότερης τεχνολογίας δίσκο (SSD). Συνεπώς, συμπεραίνουμε ότι στους αλγόριθμους ψηφιακής υπογραφής, RSA και XMSS, τα ανωτέρω χαρακτηριστικά παίζουν σημαντικό ρόλο – κατ' αρχάς πιο σημαντικό από τη μνήμη (αφού ο υπολογιστής #1 έχει μικρότερη μνήμη).

2. Οι δοκιμές στον αλγόριθμο XMSS δίνουν περισσότερο προβλεπόμενα αποτελέσματα σε σχέση με τον αλγόριθμο RSA. Παρατηρείται ότι όσο αυξάνεται η τιμή του ύψους (Υ) του δέντρου και ο αριθμός των ψηφίων στην έξοδος της συνάρτησης κατακερματισμού (ΣΚ), τόσο μεγαλώνει ο χρόνος υπογραφής του μηνύματος.

3. Οι δοκιμές με τον αλγόριθμο RSA δεν δίνουν κάποιο πρότυπο σχετικά με τα αναμενόμενα αποτελέσματα. Αυτό που παρατηρείται είναι ότι για μέγεθος κλειδιού 512 και 2048 οι χρόνοι κρυπτογράφησης είναι πολύ μικροί. Οι χρόνοι, αυξάνονται όσο αυξάνεται και το μέγεθος των κλειδιών και υπάρχει μια απότομη αύξηση χρόνου, όταν το μέγεθος κλειδιού είναι 4096.

4. Στον αλγόριθμο RSA το μέγεθος της υπογραφής επηρεάζεται μόνο από το μήκος του κλειδιού. Από τις δοκιμές παρατηρήσαμε τα ακόλουθα (βλέπε πίνακα 9):

    a. Για ΜΚ 512 bytes το μέγεθος της υπογραφής είναι 64 bytes.

    b. Για ΜΚ 2048 bytes το μέγεθος της υπογραφής είναι 256 bytes.

    c. Για ΜΚ 4096 bytes το μέγεθος της υπογραφής είναι 512 bytes.

5. Στον αλγόριθμο XMSS το μέγεθος της υπογραφής επηρεάζεται πρωτίστως από το ύψος και δευτερευόντως από την συνάρτηση κατακερματισμού. Από τις δοκιμές παρατηρήσαμε τα ακόλουθα (βλέπε πίνακα 10):

    a. Για Υ 6 και ΣΚ SHA-256 το μέγεθος της υπογραφής είναι 2372 bytes.

    b. Για Υ 8 και ΣΚ SHA-256 το μέγεθος της υπογραφής είναι 2436 bytes.

    c. Για Υ 10 και ΣΚ SHA-256 το μέγεθος της υπογραφής είναι 2500 bytes.

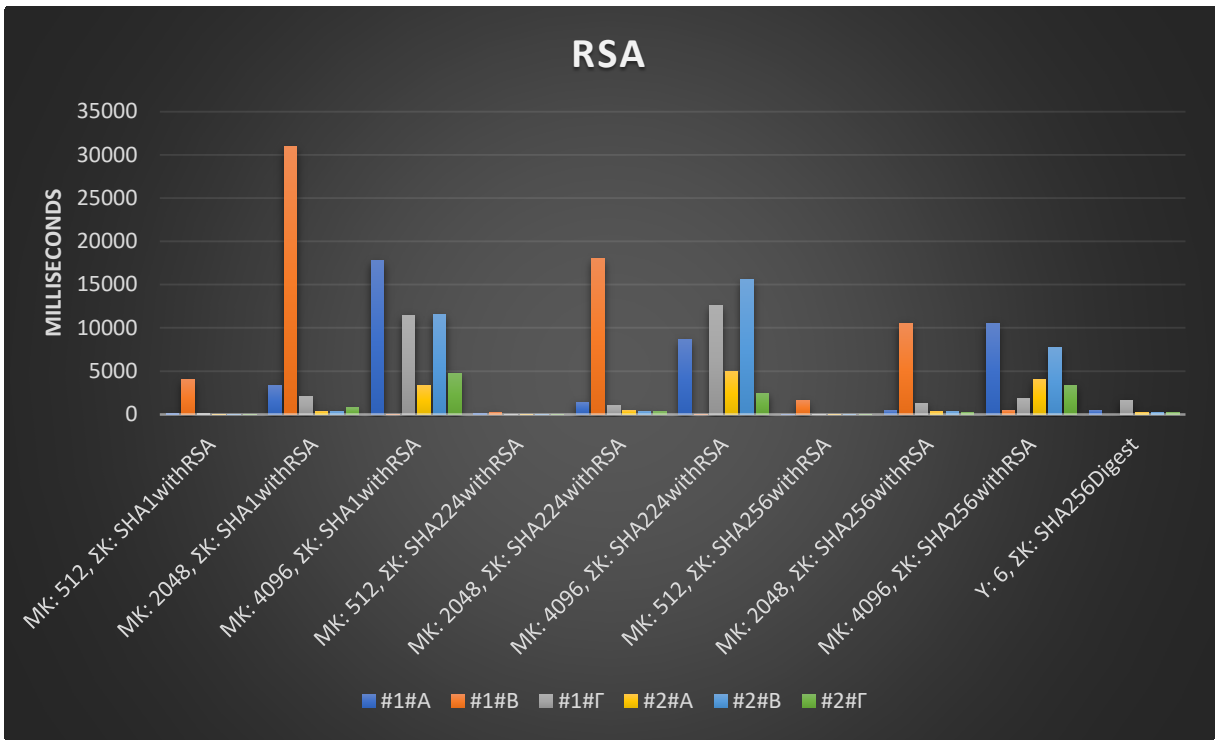
    d. Για Υ 6 και ΣΚ SHA-512 το μέγεθος της υπογραφής είναι 8836 bytes.

    e. Για Υ 8 και ΣΚ SHA-512 το μέγεθος της υπογραφής είναι 8964 bytes.

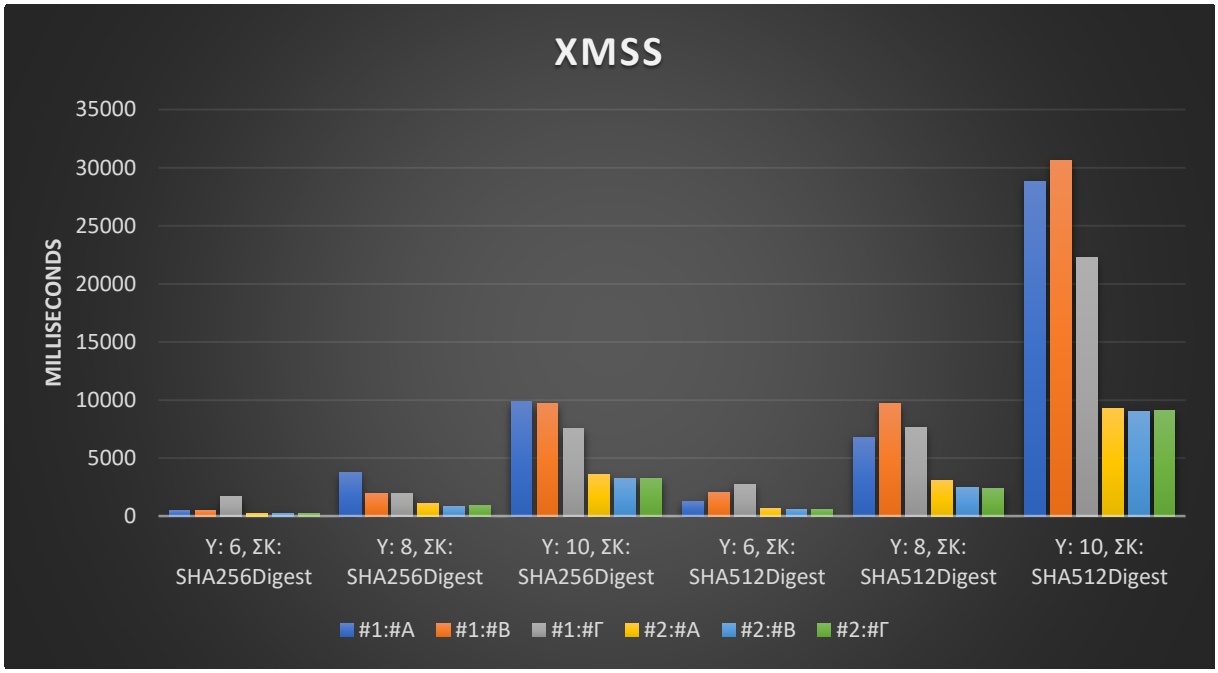
    f. Για Υ 10 και ΣΚ SHA-512 το μέγεθος της υπογραφής είναι 9092 bytes.

6. Συγκρίνοντας τον αλγόριθμο RSA με τον αλγόριθμο XMSS για ΣΚ SHA-256 παρατηρούμε τα ακόλουθα (βλέπε πίνακα 8):

    a. όσο μεγαλώνουν τα ΜΚ του RSA και το Υ του XMSS, τόσο ο XMSS βελτιώνεται συγκριτικά με τον RSA. Όμως ο XMSS παραμένει πιο αργός.



Εικόνα 19. RSA Δοκιμές



Εικόνα 20. XMSS Δοκιμές

Κεφάλαιο 5
Μελέτη περίπτωσης

5.1 Συμπεράσματα

Παρόλο που έχουν περάσει σχεδόν δύο δεκαετίες από την έναρξη της έρευνας στη περιοχή της μετα-κβαντικής κρυπτογραφίας, αυτό το ερευνητικό πεδίο εξακολουθεί να είναι σε πρώιμο στάδιο. Η έρευνα εστιάζει τόσο στην θεωρητική καινοτομία, όσο επίσης σε πειράματα με σκοπό την υλοποίηση νέων λύσεων και την τυποποίηση τους. Ίσως η πιο σημαντική κατεύθυνση σε μελλοντική έρευνα να περιλαμβάνει την κρυπτανάλυση και τις δυνατότητες ενός εισβολέα με πρόσβαση σε ένα κβαντικό υπολογιστή.

Στη παρούσα διατριβή παρουσιάσαμε τα βασικά σχήματα μετα-κβαντικής κρυπτογραφίας σε θεωρητικό επίπεδο εστιάζοντας στις ιδιότητες, τα πλεονεκτήματα, και μειονεκτήματα τους. Εστιάζουμε κυρίως σε πέντε σχήματα βασισμένα σε συναρτήσεις κατακερματισμού και αξιολογούμε τα χαρακτηριστικά ασφάλειας και υλοποίησης τους.

Οι συναρτήσεις κατακερματισμού έχουν σημαντική εφαρμογή σε συστήματα υπογραφών αλλά καθώς η μετα-κβαντική κρυπτογραφία γίνεται όλο και πιο σχετική, και η πρακτική εφαρμογή των συναρτήσεων κατακερματισμού σε αυτό το πεδίο κερδίζει όλο και περισσότερο ερευνητικό ενδιαφέρον. Ενώ τα σχήματα είναι εύκολα στην κατανόηση τους και λίγη δουλειά απομένει για να ολοκληρωθεί η διατύπωση τους σε ακαδημαϊκό επίπεδο, πρέπει να γίνουν ακόμα περισσότερα βήματα στην εφαρμογή τους. Για παράδειγμα, σχετικά εργαλεία να επιτρέπουν την ανάπτυξη διαφορετικών στρατηγικών διάσχισης δεντρικών δομών τύπου XMSS ή διαχείρισης ενδιάμεσων αποτελεσμάτων.

Με τη προγραμματιστική βιβλιοθήκη Bouncy Castle υλοποιήσαμε σενάρια σύγκρισης ενός συμβατικού αλγορίθμου (RSA) και ενός από τα πέντε παραπάνω σχήματα κατακερματισμού (XMSS). Συγκρίνοντας την απόδοση των αλγορίθμων με διαφορετικούς συνδυασμούς κλειδιών και υπολογιστικής ισχύος, παρατηρήσαμε ότι το μέγεθος του κλειδιού και η υπολογιστική ισχύς, ήταν ισχυροί παράγοντες κατά τη σύγκριση. Ωστόσο ο αλγόριθμος RSA είναι ταχύτερος από τον XMSS σε μικρότερα μήκη κλειδιών και ύψος για τον XMSS.

Εν κατακλείδι αυτό που συμπεραίνουμε από τις δοκιμές που πραγματοποιήσαμε είναι ότι, όπως αναμενόταν από τη θεωρία, ο αλγόριθμος XMSS είναι πιο αργός από τον αλγόριθμο RSA, ενώ η ταχύτητά του εξαρτάται από το μέγεθος του υποκείμενου δέντρου, το οποίο με τη σειρά του, όσο πιο μεγάλο είναι, τόσα πιο πολλά μηνύματα μπορεί να υπογράψει. Η ταχύτητα του αλγορίθμου RSA είναι αντιστρόφως ανάλογη, με το μήκος των κλειδιών που χρησιμοποιεί. Όσο αυξάνεται το μήκος κλειδιού που χρησιμοποιεί ο αλγόριθμος RSA τόσο μειώνεται η ταχύτητα εκτέλεσης του αλγόριθμου. Το μέγεθος της υπογραφής, επηρεάζεται στον αλγόριθμο RSA, μόνο από το μήκος των κλειδιών, ενώ στον αλγόριθμο XMSS, επηρεάζεται από το αριθμό εξυπηρέτησης υπογραφών στο σχήμα (μεγαλύτερο ύψος) και την συνάρτηση κατακερματισμού που χρησιμοποιεί ο αλγόριθμος. Φαίνεται ότι το προς κρυπτογράφηση μήνυμα δεν παίζει κάποιο ρόλο στα αποτελέσματα που προκύπτουν. Το μέγεθος του μηνύματος εισόδου, και στους δυο αλγόριθμους δεν επηρεάζει το μέγεθος της υπογραφής. Τέλος, είναι σαφές ότι σημαντική επίδραση στην ταχύτητα έχει η υποκείμενη τεχνολογία, όσο πιο δυνατός είναι ο επεξεργαστής τόσο καλύτεροι χρόνοι επιτυγχάνονται.

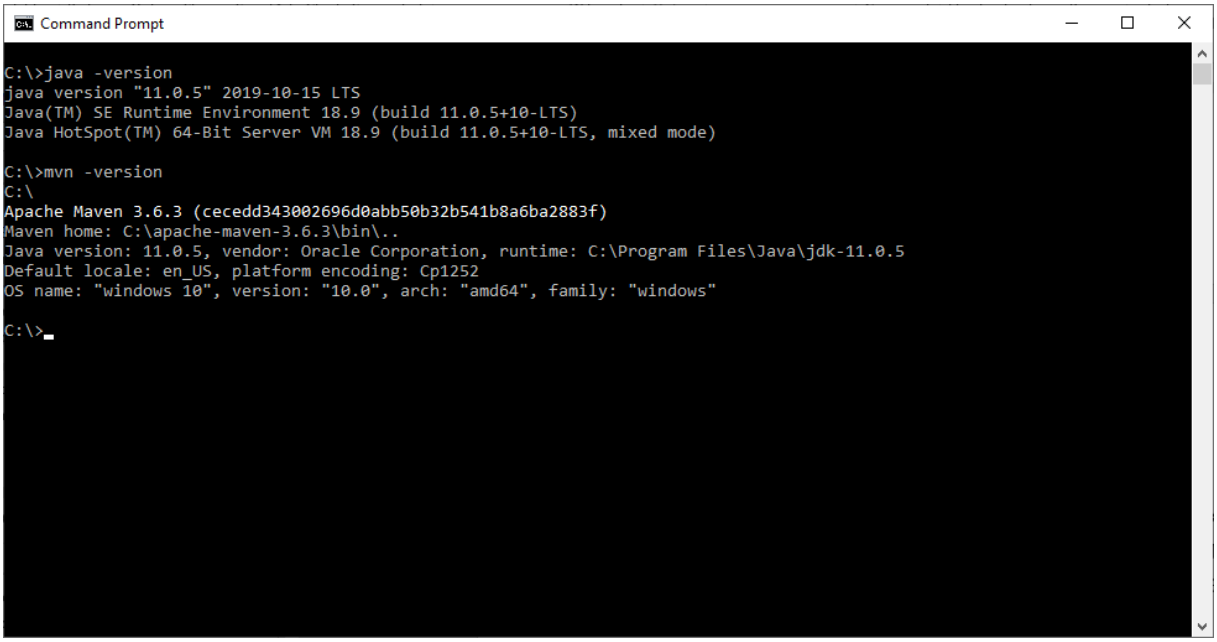
Η έρευνα στο χώρο της μετα-κβαντικής κρυπτογραφίας είναι συνεχής και φαίνεται ότι θα αποτελέσει βασικό πυλώνα της κρυπτογραφίας στα επόμενα χρόνια. Σε κάθε κρυπτογραφικό πρωτόκολλο (π.χ. TLS) γίνονται ήδη προσπάθειες ενσωμάτωσης αλγορίθμων μετα-κβαντικής κρυπτογραφίας (Bürstinghaus-Steinbach, Krauß, Niederhagen και Schneider 2020: 308). Επίσης,

ως μελλοντική διεύρυνση του χώρου που αναφέρεται στη μετα-κβαντική κρυπτογραφία στις ψηφιακές υπογραφές, με βάση τις συναρτήσεις κατακερματισμού, θα μπορούσε να είναι η τεχνολογία blockchain. Ήδη υπάρχουν τεχνικές που αξιοποιούν blockchain για την τήρηση «κόμβων» δέντρου υπογραφών (Chalkias, Brown, Hearn, Lillehagen, Nitto και Schroeter 2018), ενώ επίσης και αυτή καθ' αυτή η υλοποίηση τεχνολογιών blockchain οι οποίες να είναι ανθεκτικές στη μετα-κβαντική κρυπτογραφία αποτελεί ένα εξελισσόμενο ερευνητικά πεδίο. Σε κάθε περίπτωση, διαφαίνεται ότι είμαστε σε ένα μάλλον μεταβατικό στάδιο μετεξέλιξης της κρυπτογραφίας, από την τωρινή της μορφή στη μετα-κβαντική.

Παραρτήματα

Α' – Υλοποίηση εφαρμογής

Η υλοποίηση της παρούσας εφαρμογής έγινε με στην γλώσσα προγραμματισμού Java έκδοσης 11. Το Java project δημιουργήθηκε και αναπτύχθηκε με την χρήση του εργαλείου Apache Maven. Το λογισμικό που είναι απαραίτητο για την εκτέλεση της εφαρμογής είναι η Java έκδοσης 11 και το Apache Maven τελευταίας έκδοσης. Και τα δύο παραπάνω λογισμικά θα πρέπει να έχουν εγκατασταθεί και να τρέχουν από command line, όπως φαίνεται στην παρακάτω εικόνα.

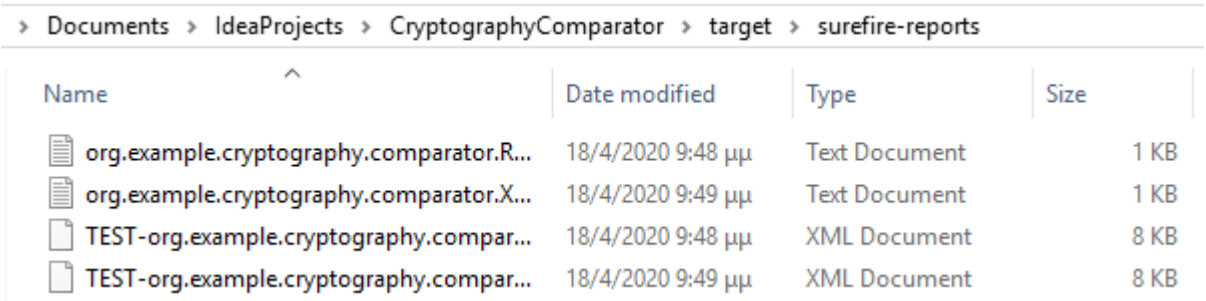


Εικόνα 21. Εκτέλεση Java και Apache Maven από command line

Το IDE που χρησιμοποιήθηκε είναι το IntelliJ IDEA 2019.3.4. Το IDE είναι απαραίτητο μόνο για την επισκόπηση του κώδικα, αλλαγές σε αυτόν ή στην παραμετροποίηση του project (αρχείο pom.xml). Επίσης, είναι απαραίτητη και η ύπαρξη σύνδεσης στο διαδίκτυο για το κατέβασμα όλων των απαραίτητων Java βιβλιοθηκών. Στο αρχείο pom.xml του Java project υπάρχουν όλες οι βιβλιοθήκες που είναι απαραίτητες για την εκτέλεση της εφαρμογής.

Η υλοποίηση της εφαρμογής έχει λάβει χώρα στο φάκελο \src\test\java ακολουθώντας τις κατευθυντήριες γραμμές του Apache Maven, καθώς αυτό που σκοπεύουμε να κάνουμε είναι να μετρήσουμε το χρόνο εκτέλεσης των δυο αλγόριθμων RSA και XMSS για μεταβλητό μήκος κλειδιού, σε διαφορετικά μηνύματα, συναρτήσεις κατακερματισμού και ύψος του δέντρου. Επίσης, έχουν γίνει οι κατάλληλες παραμετροποιήσεις στο αρχείο pom.xml, έτσι ώστε να εκτελείται από command line η εφαρμογή και να βγαίνουν τα αποτελέσματα αυτόματα στον φάκελο \target\surefire-reports.

Στις Java κλάσεις RSATest και XMSSTest υπάρχουν εκείνες οι μέθοδοι που θα εκτελεστούν για τα διαφορετικά μήκη κλειδιού. Για κάθε μία από τις κλάσεις θα δημιουργηθούν αποτελέσματα σε αρχείο XML και TXT Εικόνα 22.



Εικόνα 22. Αρχεία με αποτελέσματα δοκιμών

Για καθέναν αλγόριθμο (RSA και XMSS) δημιουργείται ένα Excel αρχείο με διάφορα στοιχεία και τιμές από τα οποία βγαίνουν τα αποτελέσματα της παραγράφου 4.5. Στο Παράρτημα Β' παρουσιάζονται αυτά τα Excel αρχεία. Η διαδρομή που θα αποθηκευτεί το εκάστοτε αρχείο καθορίζεται από τις τιμές που υπάρχουν στο αρχείο \test\resources\application.properties. Τα στοιχεία που αποθηκεύονται στο εκάστοτε Excel αρχείο είναι τα ακόλουθα:

- RSA:
    - order: σειρά δοκιμής
    - keysize (bytes): μέγεθος κλειδιού σε bytes
    - algorithm: συνάρτηση κατακερματισμού
    - messageId: id μηνύματος

- message length (bytes): μήκος μηνύματος σε bytes
- signature length (bytes): μήκος υπογραφής σε bytes
- initiate time (ms): χρόνος σε milliseconds για αρχικοποίηση δοκιμής
- sign time (ms): χρόνος σε milliseconds για υπογραφή μηνύματος
- verify time (ms): χρόνος σε milliseconds για πιστοποίηση δοκιμής
- total time (ms): συνολικός χρόνος σε milliseconds

- XMSS:
  - order: σειρά δοκιμής
  - height: μέγεθος κλειδιού σε bytes
  - digest: συνάρτηση κατακερματισμού
  - messageId: id μηνύματος
  - message length (bytes): μήκος μηνύματος σε bytes
  - signature length (bytes): μήκος υπογραφής σε bytes
  - initiate time (ms): χρόνος σε milliseconds για αρχικοποίηση δοκιμής
  - sign time (ms): χρόνος σε milliseconds για υπογραφή μηνύματος
  - verify time (ms): χρόνος σε milliseconds για πιστοποίηση δοκιμής
  - total time (ms): συνολικός χρόνος σε milliseconds

Για να εκτελέσει κάποιος την εφαρμογή θα πρέπει να ακολουθήσει τα εξής βήματα:

1. Να έχει εγκαταστημένα:

a. Java έκδοσης 11

b. Apache Maven τελευταίας έκδοσης

2. Να αποθηκεύσει το Java project σε ένα φάκελο στον υπολογιστή του

3. Να τρέξει από command line την εντολή mvn clean test site. Όταν εκτελεστεί αυτή η εντολή στον φάκελο που αναφέρουμε παραπάνω θα έχουν δημιουργηθεί τέσσερα αρχεία με τα αποτελέσματα.

Β' – Αναλυτικά αποτελέσματα

Β.1. Υπολογιστής #1

Β.1.1. RSA Test

order	keysize (bytes)	Algorithm	messageId	message length (bytes)	signature length (bytes)	initiate time (ms)	sign time (ms)	verify time (ms)	total time (ms)
6	4096	SHA224withRSA	#A	12	512	8535	83	<1	8618
4	512	SHA224withRSA	#A	12	64	111	11	<1	122
1	512	SHA1withRSA	#A	12	64	133	<1	<1	133
2	2048	SHA1withRSA	#A	12	256	3311	<1	<1	3311
8	2048	SHA256withRSA	#A	12	256	423	8	<1	431
9	4096	SHA256withRSA	#A	12	512	10467	61	<1	10528
3	4096	SHA1withRSA	#A	12	512	17726	42	<1	17768
7	512	SHA256withRSA	#A	12	64	26	<1	<1	26
5	2048	SHA224withRSA	#A	12	256	1404	16	1	1426
6	4096	SHA224withRSA	#B	53	512	17946	41	<1	17987
4	512	SHA224withRSA	#B	53	64	33	<1	<1	33
1	512	SHA1withRSA	#B	53	64	15	<1	<1	15
2	2048	SHA1withRSA	#B	53	256	4001	8	<1	4009
8	2048	SHA256withRSA	#B	53	256	1626	2	<1	1628
9	4096	SHA256withRSA	#B	53	512	10501	49	<1	10550
3	4096	SHA1withRSA	#B	53	512	30860	50	<1	30910
7	512	SHA256withRSA	#B	53	64	12	<1	<1	12

5	2048	SHA224withRSA	#B	53	256	256	2	<1	258
6	4096	SHA224withRSA	#Γ	2977	512	12574	48	<1	12622
4	512	SHA224withRSA	#Γ	2977	64	16	<1	<1	16
1	512	SHA1withRSA	#Γ	2977	64	37	<1	16	53
2	2048	SHA1withRSA	#Γ	2977	256	1956	62	<1	2018
8	2048	SHA256withRSA	#Γ	2977	256	1250	31	<1	1281
9	4096	SHA256withRSA	#Γ	2977	512	1747	69	<1	1816
3	4096	SHA1withRSA	#Γ	2977	512	11408	42	<1	11450
7	512	SHA256withRSA	#Γ	2977	64	42	<1	<1	42
5	2048	SHA224withRSA	#Γ	2977	256	975	<1	<1	975

Πίνακας 11. RSA test στον υπολογιστή #1 (χρόνος <1 σημαίνει κάτω από 1ms) .

B.1.2. XMSS Test

order	height	digest	messageId	message length (bytes)	signature length (bytes)	initiate time (ms)	sign time (ms)	verify time (ms)	total time (ms)
2	8	SHA-256	#A	12	2436	3722	16	16	3754
1	6	SHA-256	#A	12	2372	435	10	8	453
5	8	SHA-512	#A	12	8964	6745	29	10	6784
3	10	SHA-256	#A	12	2500	9877	7	<1	9884
4	6	SHA-512	#A	12	8836	1203	32	15	1250
6	10	SHA-512	#A	12	9092	28806	31	<1	28837
2	8	SHA-256	#B	53	2436	1958	15	<1	1973
1	6	SHA-256	#B	53	2372	501	16	<1	517
5	8	SHA-512	#B	53	8964	9693	28	12	9733
3	10	SHA-256	#B	53	2500	9656	31	20	9707
4	6	SHA-512	#B	53	8836	2029	31	15	2075
6	10	SHA-512	#B	53	9092	30582	31	16	30629
2	8	SHA-256	#Γ	2977	2436	1916	30	20	1966
1	6	SHA-256	#Γ	2977	2372	1601	51	18	1670
5	8	SHA-512	#Γ	2977	8964	7634	32	<1	7666

3	10	SHA-256	#Γ	2977	2500	7500	10	<1	7510
4	6	SHA-512	#Γ	2977	8836	2584	139	46	2769
6	10	SHA-512	#Γ	2977	9092	22156	71	30	22257

Πίνακας 12. XMSS test στον υπολογιστή #1 (χρόνος <1 σημαίνει κάτω από 1ms).

B.2. Υπολογιστής #2

B.2.1. RSA Test

order	keysize (bytes)	Algorithm	messageId	message length (bytes)	signature length (bytes)	initiate time (ms)	sign time (ms)	verify time (ms)	total time (ms)
6	4096	SHA224withRSA	#A	12	512	4914	49	1	4964
4	512	SHA224withRSA	#A	12	64	17	1	<1	18
1	512	SHA1withRSA	#A	12	64	17	<1	<1	17
2	2048	SHA1withRSA	#A	12	256	351	5	<1	356
8	2048	SHA256withRSA	#A	12	256	285	5	<1	290
9	4096	SHA256withRSA	#A	12	512	4035	27	<1	4062
3	4096	SHA1withRSA	#A	12	512	3333	28	<1	3362
7	512	SHA256withRSA	#A	12	64	18	1	<1	19
5	2048	SHA224withRSA	#A	12	256	480	6	<1	486
6	4096	SHA224withRSA	#B	53	512	15588	26	<1	15614
4	512	SHA224withRSA	#B	53	64	14	<1	<1	14
1	512	SHA1withRSA	#B	53	64	12	1	<1	13
2	2048	SHA1withRSA	#B	53	256	328	5	<1	333
8	2048	SHA256withRSA	#B	53	256	325	4	1	330
9	4096	SHA256withRSA	#B	53	512	7714	23	1	7738
3	4096	SHA1withRSA	#B	53	512	11506	28	1	11535
7	512	SHA256withRSA	#B	53	64	11	<1	<1	11

5	2048	SHA224withRSA	#B	53	256	367	5	<1	372
6	4096	SHA224withRSA	#Γ	2977	512	2431	24	2	2457
4	512	SHA224withRSA	#Γ	2977	64	10	<1	<1	10
1	512	SHA1withRSA	#Γ	2977	64	10	2	<1	12
2	2048	SHA1withRSA	#Γ	2977	256	768	5	<1	773
8	2048	SHA256withRSA	#Γ	2977	256	205	5	1	211
9	4096	SHA256withRSA	#Γ	2977	512	3387	23	1	3411
3	4096	SHA1withRSA	#Γ	2977	512	4733	25	<1	4758
7	512	SHA256withRSA	#Γ	2977	64	22	<1	1	23
5	2048	SHA224withRSA	#Γ	2977	256	284	4	<1	288

Πίνακας 13. RSA test στον υπολογιστή #2 (χρόνος <1 σημαίνει κάτω από 1ms).

B.2.2. XMSS Test

order	height	digest	messageId	message length (bytes)	signature length (bytes)	initiate time (ms)	sign time (ms)	verify time (ms)	total time (ms)
2	8	SHA-256	#A	12	2436	1073	9	3	1085
1	6	SHA-256	#A	12	2372	217	5	2	224
5	8	SHA-512	#A	12	8964	3038	17	8	3063
3	10	SHA-256	#A	12	2500	3572	5	2	3579
4	6	SHA-512	#A	12	8836	626	15	6	647
6	10	SHA-512	#A	12	9092	9245	13	5	9263
2	8	SHA-256	#B	53	2436	793	5	3	801
1	6	SHA-256	#B	53	2372	245	5	3	253
5	8	SHA-512	#B	53	8964	2438	14	5	2457
3	10	SHA-256	#B	53	2500	3205	5	3	3213
4	6	SHA-512	#B	53	8836	608	14	6	628
6	10	SHA-512	#B	53	9092	9026	13	5	9044
2	8	SHA-256	#Γ	2977	2436	875	5	3	883
1	6	SHA-256	#Γ	2977	2372	215	5	2	222
5	8	SHA-512	#Γ	2977	8964	2340	13	6	2359

3	10	SHA-256	#Γ	2977	2500	3269	5	2	3276
4	6	SHA-512	#Γ	2977	8836	579	13	5	597
6	10	SHA-512	#Γ	2977	9092	9051	14	6	9071

Πίνακας 14. XMSS test στον υπολογιστή #2.

Βιβλιογραφικές αναφορές

D. J. Bernstein and T. Lange. 2017."Post-quantum cryptography – dealing with the fallout of physics success," University of Illinois at Chicago p. 20.

Daniel J. Bernstein, Daira Hopwood, Andreas Hulsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. "SPHINCS: practical stateless hash-based signatures." In: Advances in Cryptology – EUROCRYPT 2015. Ed. by Marc Fischlin and Elisabeth Oswald. Vol. 9056. LNCS. Springer, 2015, pp. 368–397.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The Keccak reference.2011.url:http://keccak.noekeon.org (cit.onpp.33,138,159,168).

Brassard, G., Hoyer, P., and Tapp, A. "Quantum Cryptanalysis of Hash and Claw-Free Functions". Proceedings of LATIN '98. 1998.

Kevin Bürstinghaus-Steinbach and Christoph Krauß and Ruben Niederhagen and Michael Schneider, "Post-Quantum TLS on Embedded Systems", Cryptology ePrint Archive, Report 2020/308, 2020 (https://eprint.iacr.org/2020/308)

Johannes Buchmann, Erik Dahmen, and Andreas Hulsing. "XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions." In: Post-Quantum Cryptography – PQCrypto 2011. Ed. by Bo-Yin Yang. Vol. 7071. LNCS. Springer, 2011, pp. 117– 129.

J. Buchmann, E. Dahmen, and A. Hülsing, "XMSS—A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions," Proceedings of the 4th International Conference on Post-Quantum Cryptography (PQCrypto) 2011.

Buchmann, Johannes & Dahmen, Erik & Hülsing, Andreas. (2011). XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. IACR Cryptology ePrint Archive. 2011. 117-129. 10.1007/978-3-642-25405-5_8.

Buchmann, Johannes & Dahmen, Erik & Hülsing, Andreas. (2011). XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions. IACR Cryptology ePrint Archive. 2011. 117-129. 10.1007/978-3-642-25405-5_8.

K. Chalkias, J. Brown, M. Hearn, T. Lillehagen, I. Nitto and T. Schroeter, "Blockchained Post-Quantum Signatures," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 2018, pp. 1196-1203. doi: 10.1109/Cybermatics_2018.2018.00213

Chen, L. "Cryptography Standards in Quantum Time: New Wine in an Old Wineskin?" IEEE Security and Privacy. July/August 2017.

L. Chen, S. Jordan, Y-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, NIST "Report on Post-Quantum Cryptography" (NISTIR 8105), April 2016.

Commercial National Security Algorithm Suite. National Security Agency,2015. url:https://apps.nsa.gov/iaarchive/programs/ iad-initiatives/cnsa-suite.cfm (cit.onp.33).

Michael Curcio, David McGrew, and Scott Fluhrer. Leighton-Micali Hash-Based Signatures. Request for Comments 8554. IETF, 2019.url: https://tools.ietf.org/html/rfc8554 (cit.onpp.33,108).

Joan Daemen and Vincent Rijmen. "AES proposal: Rijndael." In: (1999). url: https://csrc.nist.gov/csrc/media/projects/ cryptographic-standards-and-guidelines/documents/aesdevelopment/rijndael-ammended.pdf (cit.onp.33).

FIPSPUB180-4: Secure Hash Standard. National Institute of Standards and Technology, 2015. url:http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf (cit.onpp.33,106,208).

E. Gerjuoy. Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers. American Journal of Physics 73, 521 (2005)

Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. “A digital signatureschemesecureagainstadaptivechosen-messageattacks.” In: Journal on Computing 17.2 (1988), pp. 281–308. url: https: //people.csail.mit.edu/silvio/Selected/Scientific/Papers/Digital/Signatures/A_Digital_Signature_Scheme_Secure_Against_Adaptive_Chosen-Message_Attack.pdf (cit.onp.28).

Grassl, M., Langenberg, B., Roetteler, M., and Steinwandt, R. “Applying Grover’s Algorithm to AES: Quantum Resource Estimates”. Proceedings of 7th Annual International Workshop on Post-Quantum Cryptography. 2016.

Grover, L.K. “A fast quantum mechanical algorithm for database search”. Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC). May 1996.

A. Gupta, N. Kaur Walia, Cryptography Algorithms: A Review. 2014 IJEDR | Volume 2, Issue 2 | ISSN: 232

Andreas Hülsing, Denis Butin, Stefan-Lukas Gazdag, Joost Rijneveld,andAzizMohaisen.XMSS:eXtendedMerkleSignatureScheme. Request for Comments 8391. IETF,2018. url:https://tools.ietf. org/html/rfc8391 (cit.onpp.17,33,44,50,52,65–67,70,72,75, 78,82,261).

Andreas Hulsing, Denis Butin, Stefan-Lukas Gazdag, Joost Rijneveld, and Aziz Mohaisen. XMSS: eXtended Merkle Signature Scheme. Request for Comments 8391. IETF, 2018. url:https://tools.ietf. org/html/rfc8391 (cit. on pp. 17, 33, 44, 50, 52, 65–67, 70, 72, 75, 78, 82, 261).

Kannwischer, Matthias J. & Genêt, Aymeric & Butin, Denis & Krämer, Juliane & Buchmann, Johannes. (2018). Differential Power Analysis of XMSS and SPHINCS. 10.1007/978-3-319-89641-0_10.

Ana Karina D.S. de Oliveira, Julio López J. (2015) An Efficient Software Implementation of the Hash-Based Signature Scheme MSS and Its Variants. In: Lauter K., Rodríguez-Henríquez F. (eds) Progress in Cryptology -- LATINCRYPT 2015. LATINCRYPT 2015. Lecture Notes in Computer Science, vol 9230. Springer, Cham

Jonathan Katz; Yehuda Lindell (6 November 2014). Introduction to Modern Cryptography, Second Edition. CRC Press. ISBN 978-1-4665-7026-9

Leslie Lamport. Constructing digital signatures from a one way function. Technical Report SRI-CSL-98. SRI International Computer Science Laboratory, 1979 (cit. on pp. 43, 45).

Stevens, Marc, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. 2017. 'The First Collision for Full SHA-1'. Σσ. 570–96 στο Advances in Cryptology – CRYPTO 2017. τ. 10401, Lecture Notes in Computer Science, επιμέλεια J. Katz και H. Shacham. Cham: Springer International Publishing.

R. Merkle, "Secrecy, Authentication and Public Key Systems—A Certified Digital Signature," PhD dissertation, Dept. of Electrical Eng., Stanford University, 1979.

Ralph Merkle. "A Certi_ed Digital Signature." In: Advances in Cryptology – CRYPTO '89. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, 1990, pp. 218–238. url: www.merkle.com/papers/Certified1979.pdf (cit. on pp. 13, 45, 47, 48, 52–54, 57).

DustinMoody.Let's Get Ready to Rumble – The NIST PQC "Competition".PresentationatPQCrypto2018.2018.url: https://csrc. nist.gov/Presentations/2018/Let-s-Get-Ready-to-RumbleThe-NIST-PQC-Competiti (cit.onp.33).

Dr. Ruben Niederhagen, Prof. Dr. Michael Waidner. Practical Post-Quantum Cryptography. FRAUNHOFER INSTITUTE FOR SECURE INFORMATION TECHNOLOGY. August 18, 2017

FIPS PUB 197: Advanced Encryption Standard (AES). National Institute of Standards and Technology,2001.url: http://nvlpubs. nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf (cit.onp.33).

FIPS PUB 202 – SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. National Institute of Standards and Technology, 2015. url: http://nvlpubs.nist.gov/nistpubs/ FIPS/NIST.FIPS.202.pdf (cit.onpp.26,33,106,208).

Submission Requirements and Evaluation Criteria for the PostQuantum Cryptography Standardization Process. National Institute of Standards and Technology, 2016. url: https : / / csrc . nist . gov / CSRC / media / Projects / Post - Quantum Cryptography/documents/call-for-proposals-final-dec2016.pdf (cit.onp.33)

“Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process”. NIST Post-Quantum Cryptography Call for Proposals. 2017.

Round Submissions. National Institute of Standards and Technology, 2019. url: https://csrc.nist.gov/projects/post-quantumcryptography/round-2-submissions (cit.onp.33).

NSA/CSS Information Assurance Directorate, “Commercial National Security Algorithm Suite and Quantum Computing FAQ” (MFQ U/OO/815099-15), January 2016.

Rijneveld, Joost. 2019. ‘Practical Post-Quantum Cryptography’. Radboud University 261.

Shor, P. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” SIAM Journal on Computing, vol. 26, no. 5. October 1997.

Ε. Λάσκαρη. Κρυπτογραφία και κρυπτανάλυση με μεθόδους υπολογιστικής νοημοσύνης και υπολογιστικών μαθηματικών και εφαρμογές. 2010 http://hdl.handle.net/10889/4134