Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή

Μελέτη και Σχεδιασμός Συστήματος Κρυπτογράφησης για Δορυφορικά Συστήματα Επικοινωνίας

Κωνσταντίνος Κουτσουρούμπης

Επιβλέπων Καθηγητής
Νικόλαος Σκλάβος

Δεκέμβριος 2016

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή

Μελέτη και Σχεδιασμός Συστήματος Κρυπτογράφησης για Δορυφορικά Συστήματα Επικοινωνίας

Κωνσταντίνος Κουτσουρούμπης

Επιβλέπων Καθηγητής
Νικόλαος Σκλάβος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στα Πληροφοριακά και Επικοινωνιακά Συστήματα από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Δεκέμβριος 2016

Περίληψη

Βασικός άξονας αυτής της μεταπτυχιακής διατριβής είναι η μελέτη και ο σχεδιασμός ενός σύγχρονου συστήματος κρυπτογραφίας για δορυφορικές επικοινωνίες. Αφορμή για την εκπόνηση της συγκεκριμένης μελέτης αποτέλεσαν οι αυξημένες ανάγκες κρυπτογράφησης των διαρκώς εξελισσόμενων δορυφορικών δικτύων, τα οποία αποτελούν τεχνολογίες αιχμής και παρουσιάζουν έντονο ερευνητικό ενδιαφέρον.

Στα πλαίσια της μεταπτυχιακής διατριβής μελετήθηκε και αναπτύχθηκε ένας επεξεργαστής ο οποίος εξυπηρετεί τις αυξημένες ανάγκες για κρυπτογράφηση δεδομένων τα οποία μεταδίδονται από ένα δορυφορικό σύστημα. Οι αλγόριθμοι κρυπτογράφησης που μας παρέχει ο επεξεργαστής – Advanced Encryption Standard (AES) με Counter Mode (CTR) για τη συμμετρική κρυπτογράφηση και Ελλειπτικές Καμπύλες (Elliptic Curves – EC) για την ασύμμετρη κρυπτογράφηση – επελέχθησαν για το επίπεδο ασφάλειας το οποίο προσφέρουν αλλά και για λόγους απόδοσης, καθώς η βελτιστοποίηση του συστήματος ως προς τις κρίσιμες παραμέτρους υλοποίησης (απόδοση, δεσμευμένοι πόροι, κατανάλωση ενέργειας κ.α.) ήταν, επίσης, ένας από τους στόχους της παρούσας μεταπτυχιακής διατριβής.

Η υλοποίηση του επεξεργαστή έγινε σε γλώσσα προγραμματισμού Java έτσι ώστε να μπορεί να χρησιμοποιηθεί από εφαρμογές, π.χ. μία εφαρμογή η οποία τρέχει σε ένα κινητό τηλέφωνο νέας τεχνολογίας (smartphone), ανεξάρτητα από την πλατφόρμα στην οποία είναι εγκατεστημένες. Ο επεξεργαστής που προτείνεται στην παρούσα μεταπτυχιακή διατριβή καλύπτει τις βασικές απαιτήσεις ασφάλειας, δηλαδή κρυπτογράφηση, αυθεντικοποίηση και ακεραιότητα, εξασφαλίζοντας παράλληλα την καλύτερη δυνατή απόδοση του συστήματος. Σχεδιασμένος να λειτουργεί σε διάταξη πομπού και δέκτη καλύπτοντας την ανάγκη για ασφαλή επικοινωνία και στις δύο περιπτώσεις.

Summary

The main axis of this master thesis is the study and design of a modern cryptography system for satellite communications. The reason of the preparation of this study were the increased encryption needs of constantly evolving satellite networks, which are the latest technologies and have strong research interest.

Within the master thesis a processor that serves the increased needs to encrypt data transmitted by a satellite system was studied and developed. The encryption algorithms that are provided by the processor - Advanced Encryption Standard (AES) with Counter Mode (CTR) for symmetric encryption and Elliptic Curves (EC) for asymmetric encryption - were selected for the security level that is offered and for reasons of efficiency, as the optimization of the system with respect to the critical implementation parameters (throughput, committed resources, energy consumption, etc.) was also one of the objectives of this master thesis.

The implementation of the processor was in Java programming language so that it can be used by applications, e.g. an application that runs on a mobile phone (smartphone), regardless of the platform on which it is installed. The processor that is proposed in this master thesis covers the essential safety requirements, i.e. encryption, authentication and integrity, while ensuring optimal system performance. Designed to operate as transmitter and receiver, addressing the need for secure communication in both cases.

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Οι δορυφορικές επικοινωνίες και η ασφάλεια των δεδομένων σε δίκτυα αυτής της κατηγορίας αποτελούν σύγχρονες τεχνολογίες αιχμής μεγάλου ερευνητικού ενδιαφέροντος. Σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη ενός επεξεργαστή ο οποίος θα καλύπτει την ανάγκη για ασφαλή μετάδοση των δεδομένων στα δίκτυα αυτά. Η φύση των δικτύων αυτών, και η σημαντική έκθεση τους σε κινδύνους που σχετίζονται με την ασφάλεια, καθιστά την αποτελεσματική προστασία των δεδομένων εξαιρετικά κρίσιμη και σημαντική. Επιπλέον, οι περιορισμένοι πόροι των δορυφόρων αλλά και άλλων συσκευών που χρησιμοποιούν τα δορυφορικά αυτά συστήματα, όπως κινητά τηλέφωνα, GPS συσκευές, τηλεοράσεις κλπ, αποτελούν μία επιπρόσθετη πρόκληση. Ζητούμενο είναι η διασφάλιση της ασφάλειας των δεδομένων που μεταδίδονται από ένα δορυφορικό σύστημα με το μικρότερο δυνατό κόστος σε πόρους (υπολογιστική ισχύς, κατανάλωση ενέργειας, μνήμη κ.α.) και την μεγαλύτερη δυνατή απόδοση.

Για την διασφάλιση της ασφάλειας των δεδομένων ο επεξεργαστής καλύπτει τις εξής βασικές λειτουργίες: κρυπτογράφηση/αποκρυπτογράφηση, πιστοποίηση και ακεραιότητα. Ο επεξεργαστής μπορεί να λειτουργεί τόσο σε διάταξη πομπού όσο και σε διάταξη δέκτη, καλύπτοντας τις ανάγκες των δορυφορικών εφαρμογών, είτε αυτές στέλνουν, είτε δέχονται δεδομένα.

Σήμερα ολοένα και περισσότερες συσκευές και υπηρεσίες (κινητά τηλέφωνα, υπολογιστές, τηλεοράσεις, σταθερή τηλεφωνία, GPS συσκευές κλπ) χρησιμοποιούν δορυφορικά συστήματα για τη μετάδοση των δεδομένων. Ο επεξεργαστής πρέπει να είναι ανεξάρτητος της πλατφόρμας στην οποία τρέχει έτσι ώστε να καλύψει τις ανάγκες όλων αυτών των συσκευών και των εφαρμογών που τρέχουν σε αυτές. Αυτό επιτυγχάνεται με την υλοποίηση του επεξεργαστή σε γλώσσα προγραμματισμού η

οποία είναι ανεξάρτητη από την πλατφόρμα στην οποία τρέχει, όπως η γλώσσα προγραμματισμού Java.

Η απόδοση είναι μία ακόμη σημαντική παράμετρος, στα δίκτυα γενικότερα, αλλά και στα δορυφορικά συστήματα ειδικότερα. Η επιλογή των αλγόριθμων κρυπτογράφησης και των τεχνικών κρυπτογράφησης έγινε με κριτήριο όχι μόνο τη διασφάλιση της μυστικότητας των δεδομένων, της ταυτότητας των επικοινωνούντων μερών και της ακεραιότητας των δεδομένων που μεταδίδονται μέσω του δορυφορικού συστήματος, αλλά και της μεγαλύτερης δυνατής απόδοσης (throughput, ταχύτητα κρυπτογράφησης και αποκρυπτογράφησης κλπ) του επεξεργαστή.

Στα επόμενα κεφάλαια θα παρουσιαστούν αναλυτικά οι επιλογές που έγιναν και ο αντίκτυπος που αναμένεται να έχουν αυτές στον επεξεργαστή όσον αφορά τις κρίσιμες παραμέτρους υλοποίησης, δηλαδή την απόδοση, τους δεσμευμένους πόρους, την υπολογιστική ισχύ, την κατανάλωση ενέργειας κλπ. Στο κεφάλαιο 2 θα γίνει μία συνοπτική παρουσίαση των δορυφορικών επικοινωνιών και των ιδιαιτεροτήτων που αυτές παρουσιάζουν. Στο κεφάλαιο 3 θα αναλυθούν οι κυριότεροι συμμετρικοί και ασύμμετροι αλγόριθμοι κρυπτογράφησης. Το κεφάλαιο 4 περιλαμβάνει την ανάλυση και τη σχεδίαση του επεξεργαστή, ενώ στο κεφάλαιο 5 θα παρουσιαστούν τα τμήματα εκείνα της υλοποίησης του επεξεργαστή που αφορούν τις λειτουργίες που αυτός επιτελεί. Το κεφάλαιο 6 περιέχει προτάσεις για μελλοντικές έρευνες που μπορούν να γίνουν με βάση τα συμπεράσματα της παρούσας μεταπτυχιακής διατριβής, τα οποία παρουσιάζονται στο κεφάλαιο 7.

Κεφάλαιο 2
Κρυπτογραφία

Με τον όρο κρυπτογραφία αναφερόμαστε στη μελέτη μαθηματικών τεχνικών που στοχεύουν στη διασφάλιση της ασφάλειας της πληροφορίας. Αυτό επιτυγχάνεται με την μετατροπή των δεδομένων σε ακατάληπτη μορφή. Η διαδικασία αυτή ονομάζεται κρυπτογράφηση, ενώ η αντίστροφη διαδικασία, κατά την οποία τα κρυπτογραφημένα ακατάληπτα δεδομένα μετατρέπονται στην αρχική τους μορφή, ονομάζεται αποκρυπτογράφηση.

Στόχος της κρυπτογραφίας είναι η διασφάλιση μίας σειράς ζητημάτων ασφάλειας. Τα κυριότερα ζητήματα ασφάλειας είναι η εμπιστευτικότητα, η ακεραιότητα των δεδομένων και η πιστοποίηση της ταυτότητας του αποστολέα. Η εμπιστευτικότητα αφορά τη μη εξουσιοδοτημένη πρόσβαση στην πληροφορία, δηλαδή τη διασφάλιση ότι η πληροφορία θα παραμείνει κρυφή σε κάθε μη εξουσιοδοτημένη οντότητα ακόμη και αν αυτή η οντότητα καταφέρει με κάποιο τρόπο να την υποκλέψει. Ο όρος ακεραιότητα αναφέρεται στη μη εξουσιοδοτημένη, και πιθανότατα κακόβουλη, τροποποίηση της πληροφορίας. Μέσω του ελέγχου ακεραιότητας διασφαλίζουμε ότι το μήνυμα δεν τροποποιήθηκε κατά τη μετάδοση του. Η πιστοποίηση της ταυτότητας του αποστολέα σχετίζεται με τη δυνατότητα εξακρίβωσης ότι ο αποστολέας της πληροφορίας είναι όντως αυτός ο οποίος ισχυρίζεται ότι είναι.

Στη συνέχεια θα μελετήσουμε δύο βασικές κατηγορίες αλγόριθμων κρυπτογράφησης, τους συμμετρικούς αλγόριθμους κρυπτογράφησης και τους ασύμμετρους αλγόριθμους κρυπτογράφησης.

2.1 Συμμετρική Κρυπτογράφηση

Σύμφωνα με τη συμμετρική κρυπτογράφηση, γνωστή και ως κρυπτογράφηση κρυφού κλειδιού, τόσο ο αποστολέας όσο και ο παραλήπτης μοιράζονται το ίδιο κρυφό κλειδί. Η

πληροφορία κρυπτογραφείται και αποκρυπτογραφείται με το κοινό κρυφό κλειδί. Το κλειδί της συμμετρικής κρυπτογράφησης το γνωρίζουν μόνο ο αποστολέας και ο παραλήπτης και πρέπει να παραμείνει κρυφό. Οι συμμετρικοί αλγόριθμοι κρυπτογράφησης χωρίζονται γενικά στους κρυπταλγόριθμους ροής (stream ciphers) και τους κρυπταλγόριθμους τμήματος (block ciphers). Η συμμετρική κρυπτογράφηση είναι γρήγορη και αποδοτική, στοιχεία που την καθιστούν ιδανική για την κρυπτογράφηση και αποκρυπτογράφηση μεγάλου όγκου δεδομένων. Στην συμμετρική κρυπτογράφηση η μεγαλύτερη πρόκληση έγκειται στην ασφαλή διανομή, ανάμεσα στα μέρη που θέλουν να επικοινωνήσουν, του κοινού μυστικού κλειδιού.

2.2 Ασύμμετρη Κρυπτογράφηση

Η ασύμμετρη κρυπτογράφηση, γνωστή και ως κρυπτογράφηση δημόσιου κλειδιού, χρησιμοποιεί ένα ζεύγος κλειδιών για την κρυπτογράφηση και την αποκρυπτογράφηση ενός μηνύματος. Κάθε ζεύγος κλειδιών αποτελείται από το δημόσιο κλειδί (public key), το οποίο διανέμεται ελεύθερα και είναι γνωστό σε όλους και το ιδιωτικό κλειδί (private key), το οποίο παραμένει κρυφό και το γνωρίζει μόνο ο κάτοχος του. Στην κρυπτογράφηση δημοσίου κλειδιού το δημόσιο κλειδί του παραλήπτη χρησιμοποιείται για την κρυπτογράφηση του μηνύματος και το αντίστοιχο ιδιωτικό κλειδί για την αποκρυπτογράφηση του. Με αυτό τον τρόπο οποιοσδήποτε έχει τη δυνατότητα να κρυπτογραφήσει ένα μήνυμα για κάποιο συγκεκριμένο παραλήπτη, χρησιμοποιώντας το δημόσιο κλειδί του παραλήπτη το οποίο είναι γνωστό σε όλους. Από την άλλη μόνο ο παραλήπτης, που έχει στην κατοχή του το αντίστοιχο ιδιωτικό κλειδί και το οποίο μόνο αυτός γνωρίζει, μπορεί να αποκρυπτογραφήσει το μήνυμα το οποίο κρυπτογραφήθηκε με το δημόσιο κλειδί του.

2.3 Συνδυασμός Συμμετρικής και Ασύμμετρης Κρυπτογράφησης

Η συμμετρική και η ασύμμετρη κρυπτογράφηση μπορούν να χρησιμοποιηθούν συμπληρωματικά έτσι ώστε να επιτύχουμε τα καλύτερα δυνατά αποτελέσματα. Η ασύμμετρη κρυπτογράφηση χρησιμοποιείται για την ασφαλή διανομή του κοινού κρυφού κλειδιού της συμμετρικής κρυπτογράφησης. Έπειτα, η συμμετρική κρυπτογράφηση, η οποία είναι πιο γρήγορη από την ασύμμετρη κρυπτογράφηση, χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση του κύριου όγκου

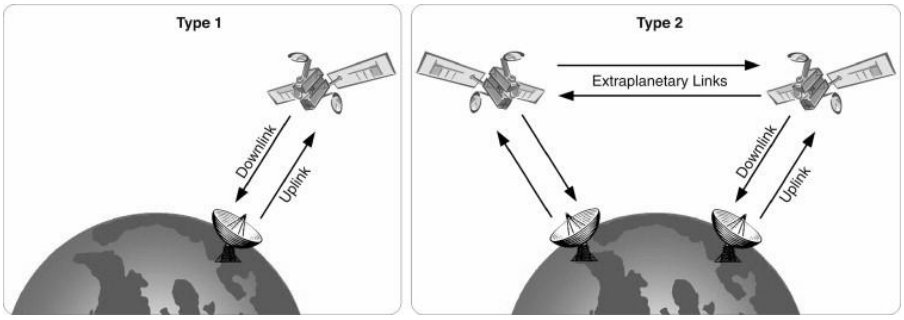
των δεδομένων τα οποία θα αποσταλούν. Επί της ουσίας, η ασύμμετρη κρυπτογράφηση λύνει το πρόβλημα της ασφαλούς διανομής του μυστικού κλειδιού της συμμετρικής κρυπτογράφησης, ενώ η συμμετρική κρυπτογράφηση η οποία είναι πιο γρήγορη από την ασύμμετρη κρυπτογράφηση χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων έτσι ώστε να επιτευχθεί η καλύτερη δυνατή απόδοση. Οι αλγόριθμοι ασύμμετρης κρυπτογράφησης χρησιμοποιούνται επίσης, σε συνδυασμό με κάποια συνάρτηση κατακερματισμού (hash function), για αυθεντικοποίηση και έλεγχο ακεραιότητας.

Κεφάλαιο 3
Δορυφορικές Επικοινωνίες

Οι δορυφορικές επικοινωνίες παίζουν ολοένα και μεγαλύτερο ρόλο στα σύγχρονα συστήματα. Η δυνατότητα τους να καλύπτουν περιοχές όπου η επίγεια υποδομή δεν είναι αρκετή για να ικανοποιήσει τις ανάγκες για επικοινωνία, όπως π.χ. στους ωκεανούς ή σε αναπτυσσόμενες χώρες, τις καθιστά ιδιαίτερα ελκυστικές. Σημαντικά στοιχεία των δορυφορικών επικοινωνιών είναι ακόμη η δυνατότητα λήψης δεδομένων σε άμεσο χρόνο χωρίς την παρέμβαση άλλων δικτύων καθώς και η προσιτή πλέον χρήση τους σε όλες τις εκφάνσεις της ανθρώπινης δραστηριότητας. Μέσω των δικτύων αυτών είναι πλέον εφικτή η επικοινωνία οποιουδήποτε, με οποιονδήποτε, οπουδήποτε και αν βρίσκονται και ανά πάσα στιγμή. Οι δορυφορικές επικοινωνίες έχουν εφαρμογή σε πολλές υπηρεσίες, όπως τα τηλεφωνικά δίκτυα, η τηλεόραση, οι δορυφορικές ραδιοφωνικές μεταδόσεις, οι ραδιοερασιτεχνικές μεταδόσεις μέσω δορυφόρων, το διαδίκτυο, ποικίλες στρατιωτικές εφαρμογές και πολλές άλλες.

3.1 Βασικά Χαρακτηριστικά

Με βάση την τροχιά τους οι δορυφόροι διακρίνονται σε χαμηλής τροχιάς (Low Earth Orbit – LEO), μεσαίας τροχιάς (Medium Earth Orbit – MEO) και γεωστατικής τροχιάς (Geostationary Earth Orbit). Επίσης, με βάση τους σκοπούς επικοινωνίας, οι σύγχρονοι δορυφόροι διακρίνονται σε δύο βασικές κατηγορίες, εκείνους που επικοινωνούν μόνο με σταθμούς στην επιφάνεια της γης και εκείνους που επικοινωνούν τόσο με σταθμούς στην επιφάνεια της γης όσο και με άλλους δορυφόρους (Soper 2009: 423), όπως φαίνεται και στην Εικόνα 1.



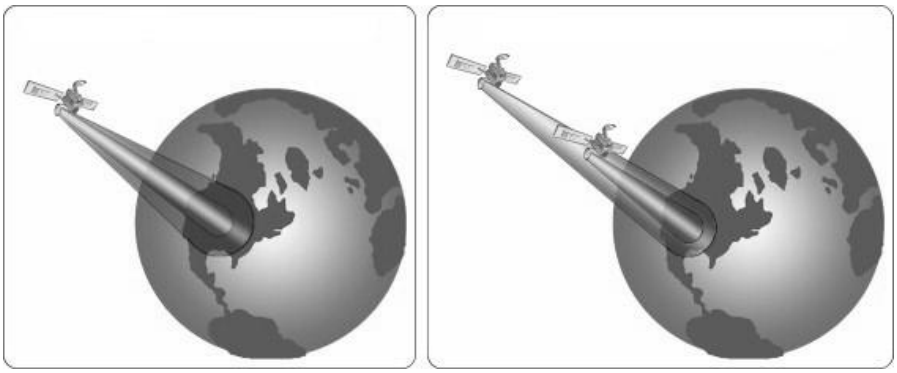
Εικόνα 1. Διάκριση δορυφόρων βάση των σκοπών επικοινωνίας.

Ένα δορυφορικό σύστημα αποτελείται από το επίγειο και το διαστημικό τμήμα. Για να επικοινωνήσει ένας επίγειος σταθμός βάσης με ένα δορυφόρο χρησιμοποιούνται δύο ζεύξεις. Η άνω ζεύξη (uplink) η οποία χρησιμοποιείται όταν ένας επίγειος σταθμός στέλνει δεδομένα σε ένα δορυφόρο και η κάτω ζεύξη (downlink) η οποία χρησιμοποιείται όταν ένας δορυφόρος στέλνει δεδομένα σε ένα επίγειο σταθμό.

3.2 Κρυπτογράφηση και Δορυφορικές Επικοινωνίες

Όπως συμβαίνει στα επίγεια δίκτυα, ασύρματα ή ενσύρματα, έτσι και στα δορυφορικά δίκτυα υπάρχει η ανάγκη να προστατευτεί η πληροφορία που μεταδίδεται μέσα σε αυτά. Μία ανάγκη η οποία πηγάζει αφενός από τη φύση της πληροφορίας, είτε μιλάμε π.χ. για μία τραπεζική συναλλαγή, μία τηλεφωνική κλήση, την αποστολή ενός ηλεκτρονικού μηνύματος μέσω μίας υπηρεσίας ηλεκτρονικού ταχυδρομείου ή τη μετάδοση ευαίσθητων δεδομένων μέσω στρατιωτικών δικτύων. Η αξία της πληροφορίας ενδέχεται να διαφέρει, ανάλογα με την περίπτωση, αλλά αυτό δεν αναιρεί την ανάγκη η πληροφορία αυτή να προστατευτεί.

Στα δορυφορικά δίκτυα είναι και η φύση των δικτύων τέτοια που καθιστά ακόμη πιο επιτακτική την ανάγκη για προστασία της μεταδιδόμενης μέσω αυτών πληροφορίας. Εάν για παράδειγμα ένας δορυφόρος θέλει να επικοινωνήσει με έναν επίγειο σταθμό βάσης, θα στρέψει την κεραία του ώστε να ευθυγραμμιστεί με τον συγκεκριμένο σταθμό. Αν και ο δορυφόρος θα εστιάσει την εκπομπή στο σταθμό βάσης με τον οποίο θέλει να επικοινωνήσει, το σήμα διασκορπίζεται, με αποτέλεσμα να καλύπτει μία ευρεία γεωγραφική περιοχή όταν φτάνει στην επιφάνεια της γης (βλέπε Εικόνα 2). Είναι απαραίτητο, επομένως, η πληροφορία που θα μεταδοθεί να είναι κρυπτογραφημένη, καθώς σε διαφορετική περίπτωση οποιοσδήποτε βρίσκεται εντός της περιοχής εκπομπής θα μπορεί να την διαβάσει (Soper 2009: 424).



Εικόνα 2. Διασκορπισμός σήματος.

Οι δορυφορικές επικοινωνίες είναι, επίσης, ευάλωτες σε επιθέσεις τύπου Άνθρωπος-Στη-Μέση (Man-In-The-Middle – MITM). Πρέπει, επομένως, να διασφαλιστεί, εκτός από τη μυστικότητα των δεδομένων, και η ταυτότητα των οντοτήτων που επικοινωνούν. Αυτό επιτυγχάνεται μέσω της αυθεντικοποίησης, κατά την οποία πιστοποιείται η ταυτότητα του αποστολέα του μηνύματος.

Η ακεραιότητα των δεδομένων, δηλαδή η εξασφάλιση ότι τα δεδομένα τα οποία φτάνουν στον παραλήπτη δεν έχουν παραποιηθεί από κάποιο επιτιθέμενο, είναι μία ακόμη πτυχή ασφάλειας που πρέπει να ληφθεί υπόψη. Αυτό επιτυγχάνεται με τη χρήση κατάλληλων συναρτήσεων κατακερματισμού (hash functions), οι οποίες μας επιτρέπουν να εντοπίσουμε μη εξουσιοδοτημένες αλλαγές στα δεδομένα που μεταδίδονται μέσω του συστήματος.

Κεφάλαιο 4
Ανάλυση και Σχεδίαση Επεξεργαστή

Σκοπός του επεξεργαστή είναι να καλύψει τις αυξημένες ανάγκες για κρυπτογράφηση δεδομένων, τα οποία μεταδίδονται από ένα δορυφορικό σύστημα. Ο επεξεργαστής παρέχει κρυπτογράφηση μέσω κατάλληλου συμμετρικού αλγόριθμου κρυπτογράφησης (Advanced Encryption Standard – AES), αυθεντικοποίηση μέσω κατάλληλου μηχανισμού πιστοποίησης της ταυτότητας των εμπλεκόμενων μερών (Ελλειπτικές Καμπύλες σε συνδυασμό με ένα Public Key Infrastructure – PKI) και διασφάλιση της ακεραιότητας των δεδομένων μέσω κατάλληλης συνάρτησης κατακερματισμού (Secure Hash Algorithm – SHA256 της οικογένειας αλγορίθμων SHA-2). Οι παραπάνω λειτουργίες παρέχονται από τον επεξεργαστή τόσο σε διάταξη πομπού όσο και σε διάταξη δέκτη. Στις επόμενες ενότητες θα αναλυθούν οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιήθηκαν καθώς και ο τρόπος λειτουργίας του επεξεργαστή τόσο σε διάταξη πομπού όσο και σε διάταξη δέκτη.

4.1 Κρυπτογράφηση και Αποκρυπτογράφηση

Δύο βασικοί τύποι αλγόριθμων κρυπτογράφησης υπάρχουν μέχρι σήμερα, οι συμμετρικοί αλγόριθμοι κρυπτογράφησης – όπου ένα μοναδικό κοινό μυστικό κλειδί χρησιμοποιείται από τον αποστολέα και τον παραλήπτη για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων – και οι ασύμμετροι αλγόριθμοι κρυπτογράφησης, όπου ένα ζεύγος δημόσιου/ιδιωτικού κλειδιού χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων.

Για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων προκρίθηκε η χρήση ενός συμμετρικού αλγόριθμου κρυπτογράφησης. Η συμμετρική κρυπτογράφηση

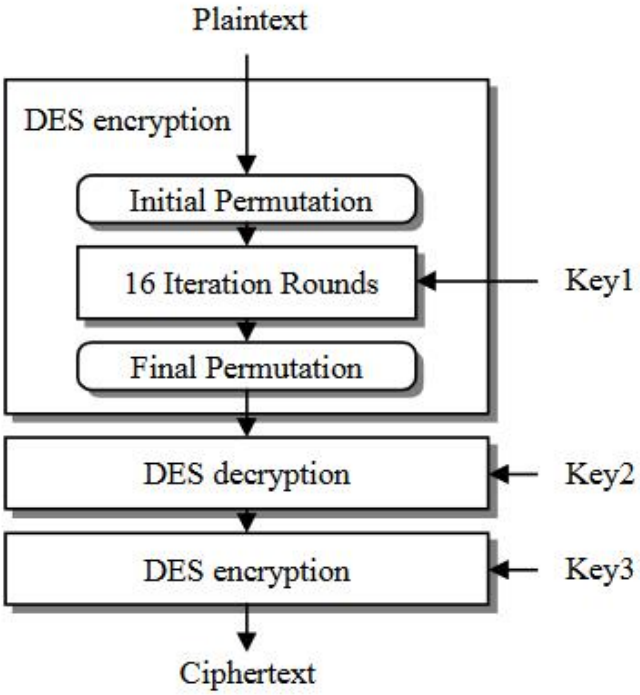
απαιτεί κατά αρκετές τάξεις μεγέθους λιγότερους υπολογισμούς, σε σχέση με την ασύμμετρη κρυπτογράφηση και ως εκ τούτου οι συμμετρικοί αλγόριθμοι κρυπτογράφησης είναι ταχύτεροι σε σύγκριση με τους ασύμμετρους αλγόριθμους κρυπτογράφησης (Preneel et al. 2003). Ο επεξεργαστής, για λόγους που αναλύονται παρακάτω, χρησιμοποιεί για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων τον συμμετρικό αλγόριθμο AES με μέγεθος κλειδιού 128 bits και τρόπο λειτουργίας μετρητή (CTR mode).

4.1.1 Συμμετρικοί Αλγόριθμοι Κρυπτογράφησης

Σε αυτή την ενότητα θα γίνει μία συνοπτική παρουσίαση μερικών εκ των βασικότερων αλγόριθμων συμμετρικής κρυπτογράφησης.

Triple DES (3DES): Ο 3DES είναι ένας συμμετρικός αλγόριθμος κρυπτογράφησης τμήματος ο οποίος αποτελεί μετεξέλιξη του Data Encryption Standard (DES). Στον 3DES έχουμε τριπλή διαδοχική εφαρμογή του αλγόριθμου DES, χρησιμοποιώντας κάθε φορά διαφορετικό κλειδί. Το μέγεθος του κλειδιού στον DES είναι 56 bits, οπότε το αντίστοιχο μέγεθος κλειδιού στον 3DES είναι 3 χ 56 = 168 bits. Στον 3DES το μεσαίο στάδιο επιτελεί αποκρυπτογράφηση έτσι ώστε ο 3DES να μπορεί να αποκρυπτογραφεί μηνύματα που έχουν κρυπτογραφηθεί με τον DES.

Ο 3DES θεωρείται γενικά ως αλγόριθμος απαιτητικός σε πόρους, ειδικά για υλοποιήσεις σε λογισμικό. Κρίνεται καταλληλότερος για υλοποιήσεις σε επίπεδο υλικού, καθώς τα κουτιά αντικατάστασης (S-boxes), τα οποία αποτελούν βασικό μέρος του αλγόριθμου, υλοποιούνται αποτελεσματικότερα σε επίπεδο υλικού (hardware). Στην Εικόνα 3 μπορούμε να δούμε τη δομή του 3DES σε υψηλό επίπεδο (Hamalainen et al. 2001: 1222).

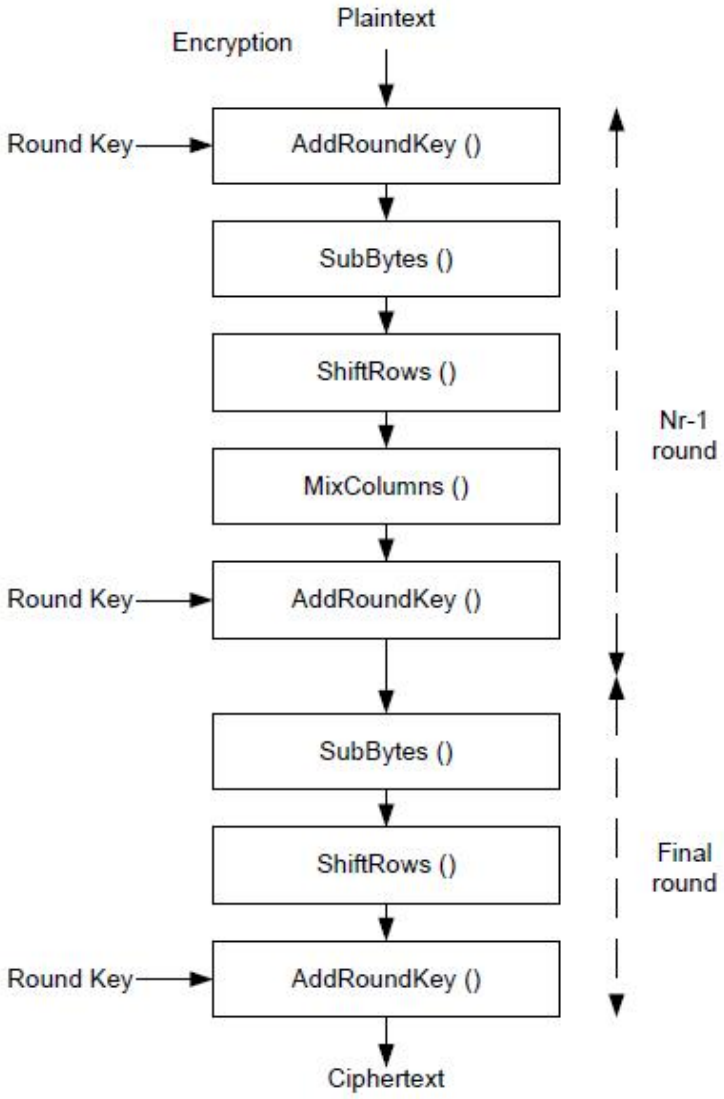


Εικόνα 3. Υψηλού επιπέδου δομή του αλγόριθμου κρυπτογράφησης 3DES.

RC6: Κρυπταλγόριθμος τμήματος ο οποίος προέρχεται από τον RC5. Σχεδιάστηκε από τους Ron Rivest, Matt Robshaw, Ray Sidney και Yiqun Lisa Yin και ήταν ένας από τους πέντε αλγόριθμους της τελικής φάσης στο διαγωνισμό για την επιλογή του Advanced Encryption Standard (AES), ο οποίος θα αντικαθιστούσε το μέχρι τότε πρότυπο, δηλαδή τον αλγόριθμο DES. Στον RC6 το μέγεθος του τμήματος που κρυπτογραφείται κάθε φορά είναι 128 bits και υποστηρίζονται κλειδιά μεγέθους 128, 192 και 256 bits. Η δομή του είναι παρόμοια με την δομή του RC5 από τον οποίο και προέρχεται άλλωστε. Χρησιμοποιεί, όπως και ο RC5, λειτουργίες όπως οι περιστροφές (rotations) οι οποίες υλοποιούνται αποτελεσματικά στους σύγχρονους επεξεργαστές. Βασική προσθήκη σε σχέση με τον RC5 αποτελεί η χρησιμοποίηση του πολλαπλασιασμού ακεραίων 32 bit, έτσι ώστε να επιτευχθεί καλύτερη και αποδοτικότερη διάχυση. Η διάχυση στον RC6 είναι ταχύτερη από ότι στον RC5, επιτρέποντας στον RC6 να επιτυγχάνει υψηλότερα επίπεδα ασφάλειας με λιγότερους γύρους και αυξημένη αποδοτικότητα (throughput), σε σύγκριση με τον RC5 (Rivest et al. 1998: 2).

Advanced Encryption Standard (AES ή Rijndael όπως ονομαζόταν πριν επιλεγεί ως AES): Πρόκειται για κρυπταλγόριθμο τμήματος ο οποίος αναπτύχθηκε από τους Joan Daemen και Vincent Rijmen και επιλέχθηκε τελικά από τον NIST (National Institute of Standards

and Technology) ως πρότυπο κρυπτογράφησης το 2002 για να αντικαταστήσει τον DES (Daemen & Rijmen 2013: 7). Υπάρχουν τρεις εκδοχές του αλγόριθμου ανάλογα με το μέγεθος του κλειδιού (AES128, AES192 και AES256). Το μέγεθος του τμήματος, στα οποία χωρίζονται τα δεδομένα, είναι 128 bits. Ο αριθμός των γύρων εξαρτάται από το μέγεθος του κλειδιού (10, 12 και 14 γύροι για μέγεθος κλειδιού 128, 192 και 256 bits αντίστοιχα). Σε κάθε γύρο εκτελούνται διαδοχικά τέσσερις διαδικασίες οι sub_bytes, shift_rows, mix_columns και add_round_key, όπως φαίνεται και στην Εικόνα 4 (Daemen & Rijmen 2013: 31, Heron 2009: 10, Rais & Qasim 2009: 305).



Εικόνα 4. Διαδικασίες που εκτελούνται σε κάθε γύρο του AES κατά την κρυπτογράφηση.

Σε κάθε γύρο του αλγόριθμου AES τα δεδομένα αναμιγνύονται με ένα κλειδί γύρου (round key), το οποίο δημιουργείται από το κλειδί κρυπτογράφησης (encryption key).

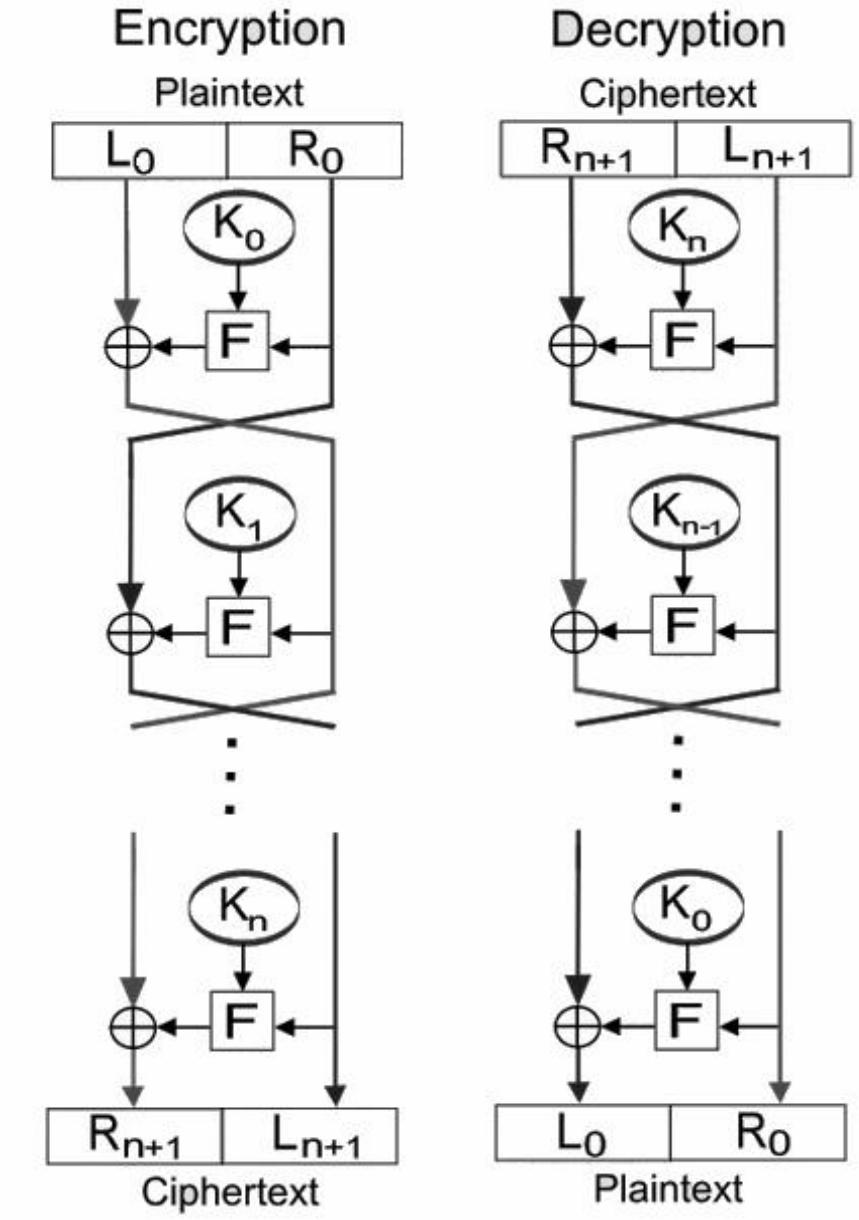
Κάθε μπλοκ δεδομένων, μεγέθους 128 bit, εκλαμβάνεται ως ένας 4χ4 πίνακας του οποίου η κάθε θέση είναι 1 byte. Όλες οι μετατροπές εφαρμόζονται πάνω σε αυτό τον πίνακα. Η sub_bytes επιτελεί αντικατάσταση byte με τη χρήση κατάλληλα διαμορφωμένων S-Boxes. Αποτελεί το μόνο μη γραμμικό τμήμα του αλγόριθμου και καθορίζει σε μεγάλο βαθμό την ασφάλεια του. Η shift_rows επιτελεί ολίσθηση, ολισθαίνοντας κυκλικά τη δεύτερη, τρίτη και τέταρτη γραμμή του πίνακα κατά ένα, δύο και τρία bytes αντίστοιχα. Η mix_columns διαδικασία επιτελεί ανάμειξη στηλών, μέσω του πολλαπλασιασμού κάθε στήλης του πίνακα με ένα σταθερό και κατάλληλα διαμορφωμένο πίνακα C διαστάσεων 4χ4, όπου κάθε στοιχείου του είναι ένα byte. Η mix_columns διαδικασία δεν εκτελείται στον τελευταίο γύρο του AES (Rais & Qasim 2009: 306).

Serpent: Συμμετρικός αλγόριθμος κρυπτογράφησης τμήματος ο οποίος σχεδιάστηκε από τους Ross Anderson, Eli Biham και Lars Knudsen. Κατέλαβε την δεύτερη θέση στον διαγωνισμό για την επιλογή του AES πίσω μόνο από τον Rijndael. Το μέγεθος του τμήματος που χρησιμοποιείται είναι 128 bits και υποστηρίζονται κλειδιά μεγέθους 128, 192 και 256 bits. Αποτελείται από ένα δίκτυο αντικατάστασης-αντιμετάθεσης (SP Network) 32 γύρων, κατά τους οποίους οι μετασχηματισμοί εφαρμόζονται σε ένα μπλοκ τεσσάρων λέξεων μεγέθους 32 bits η κάθε μία (οι οποίες αντιστοιχούν στο τμήμα μεγέθους 128 bits το οποίο υποστηρίζει ο αλγόριθμος). Πιο αναλυτικά ο αλγόριθμος αποτελείται από μία αρχική αντιμετάθεση, 32 γύρους κατά τους οποίους σε κάθε γύρο εφαρμόζεται μία λειτουργία ανάμιξης κλειδιού (key mixing operation), ένα πέρασμα από τα S-Boxes (σε όλους τους γύρους εκτός από τον τελευταίο, κατά τον οποίο γίνεται η αντιμετάθεση) και μία γραμμική μετατροπή, η οποία στον τελευταίο γύρο αντικαθίσταται από μία επιπλέον λειτουργία ανάμιξης κλειδιού. Μετά το πέρας των 32 γύρων ακολουθεί μία ακόμη τελική αντιμετάθεση (Anderson et al. 1998: 2).

Twofish: Κρυπταλγόριθμος τμήματος με μέγεθος τμήματος, όπως και οι προηγούμενοι αλγόριθμοι, 128 bits και κλειδί μεγέθους έως 256 bits. Σχεδιάστηκε από τους Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall και Niels Ferguson. Βασίζεται στα δίκτυα Feistel και αποτελείται από 16 γύρους.

Ο αλγόριθμος Feistel (ή δίκτυο Feistel) δημιουργήθηκε από τον Horst Feistel (Feistel 1973: 16, Feistel et al. 1975: 1546) και είναι βασισμένος στις αρχές του Shannon.

Πρόκειται για μία γενική μέθοδο μετατροπής μίας συνάρτησης F σε μία αντιμετάθεση. Η συνάρτηση F μπορεί να είναι οποιαδήποτε συνάρτηση, η οποία όμως πρέπει να αποτελείται από S-Boxes και P-Boxes. Τη δομή ενός δικτύου Feistel μπορούμε να δούμε στην Εικόνα 5. Στα δίκτυα Feistel το κύκλωμα της αποκρυπτογράφησης είναι το ίδιο με αυτό της κρυπτογράφησης και αυτό είναι ένα από τα σημαντικότερα πλεονεκτήματα των δικτύων αυτών.



Εικόνα 5. Κρυπτογράφηση και αποκρυπτογράφηση σε ένα δίκτυο Feistel.

Χαρακτηριστικά γνωρίσματα του Twofish αποτελούν τα προϋπολογισμένα εξαρτώμενα από το κλειδί S-Boxes, καθώς και ένα σχετικά περίπλοκο χρονοδιάγραμμα κλειδιού (key

schedule), κατά το οποίο τα bits του κλειδιού μετατρέπονται σε κλειδιά για τον εκάστοτε γύρο (round keys)(Schneier et al. 1998: 3).

4.1.2 Επιλογή Συμμετρικού Αλγόριθμου Κρυπτογράφησης

Για τον επεξεργαστή επελέγει τελικά ο συμμετρικός αλγόριθμος κρυπτογράφησης AES με μέγεθος κλειδιού 128 bits. Ο AES είναι αυτή τη στιγμή το πρότυπο συμμετρικού αλγόριθμου κρυπτογράφησης τμήματος, το οποίο έχει επιλεγεί από τον NIST, και αποτελεί έναν πολύ ισχυρό αλγόριθμο κρυπτογράφησης, ο οποίος έχει πολλά πλεονεκτήματα. Ο AES με μέγεθος κλειδιού 128 bits παρέχει επαρκή ασφάλεια για τα δεδομένα του συστήματος, ενώ είναι πιο αποδοτικός από τις αντίστοιχες υλοποιήσεις του αλγόριθμου με μεγέθη κλειδιών 192 και 256 bits (Dray 2000: 4, Sterbenz & Lipp 2000: 164).

Υπάρχουν πλέον 128-bits AES encryptors οι οποίοι μπορούν να κρυπτογραφήσουν σε ρυθμούς της τάξης του Gbit/s (Hodjat & Verbauwhede 2004: 309). Ο αλγόριθμος AES καταγράφει υψηλές επιδόσεις, σε σύγκριση πάντα με τους υπόλοιπους συμμετρικούς αλγόριθμους κρυπτογράφησης τμήματος, όσον αφορά την ταχύτητα κρυπτογράφησης, την ταχύτητα δημιουργίας του κλειδιού και το throughput (δηλαδή, το μέγεθος των δεδομένων τα οποία κρυπτογραφούνται ή αποκρυπτογραφούνται ανά δευτερόλεπτο) σε διάφορες πλατφόρμες υλοποίησης (Dandalis et al. 2000). Αντίστοιχες επιδόσεις έχουν καταγραφεί για τον AES, όχι μόνο για hardware αλλά και για software υλοποιήσεις (Dray 2000: 8, Sterbenz & Lipp 2000: 165).

4.2 Αλγόριθμος Ασύμμετρης Κρυπτογράφησης

Η συμμετρική κρυπτογράφηση χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση των δεδομένων αλλά η ασύμμετρη κρυπτογράφηση είναι απαραίτητη για την ασφαλή ανταλλαγή του συμμετρικού κλειδιού κρυπτογράφησης. Η ασύμμετρη κρυπτογραφία χρησιμοποιείται επίσης, για την πιστοποίηση της ταυτότητας του αποστολέα (αυθεντικοποίηση) και τον έλεγχο της ακεραιότητας των δεδομένων (σε συνδυασμό με μία συνάρτηση κατακερματισμού), λειτουργίες απαραίτητες για τη διασφάλιση της ασφάλειας κατά την επικοινωνία, είτε αυτή είναι επίγεια είτε δορυφορική.

Οι αλγόριθμοι ασύμμετρης κρυπτογράφησης βασίζονται σε δύσκολα προβλήματα η επίλυση των οποίων είναι, τουλάχιστον μέχρι σήμερα, υπολογιστικά αδύνατη. Οι πιο διαδεδομένοι ασύμμετροι αλγόριθμοι κρυπτογράφησης είναι ο RSA και οι Ελλειπτικές Καμπύλες (Elliptic Curve Cryptography – ECC). Ακολουθεί μία συνοπτική παρουσίαση των συγκεκριμένων αλγόριθμων ασύμμετρης κρυπτογράφησης.

RSA: Ο ασύμμετρος αλγόριθμος κρυπτογράφησης RSA πήρε το όνομα του από τα αρχικά των δημιουργών του, Rivest, Shamir και Adleman. Πρόκειται για τον πλέον διαδεδομένο αλγόριθμο ασύμμετρης κρυπτογράφησης. Το κρυπτοσύστημα RSA προτάθηκε από τους Rivest, Shamir και Adleman το 1977 και βασίζεται στο δύσκολο πρόβλημα της παραγοντοποίησης ενός σύνθετου ακεραίου σε γινόμενο πρώτων παραγόντων. Η ασφάλεια του RSA εξαρτάται από το πρόβλημα της παραγοντοποίησης. Εάν κάποια στιγμή βρεθεί αποδοτικός αλγόριθμος για την επίλυση του συγκεκριμένου δύσκολου προβλήματος τότε αυτόματα και ο αλγόριθμος RSA θα πάψει να θεωρείται ασφαλής. Μέχρι σήμερα τέτοιος αλγόριθμος δεν έχει βρεθεί.

Η κρυπτογράφηση στον αλγόριθμο RSA, ενός μηνύματος M και με δημόσιο κλειδί το ζεύγος (e, N) γίνεται μέσω του υπολογισμού $C = M^e (\mathrm{mod}\ N)$, όπου το N είναι το γινόμενο δύο μεγάλων πρώτων αριθμών. Για λόγους ασφάλειας το N πρέπει να αποτελείται τουλάχιστον από 1024 ψηφία. Η αποκρυπτογράφηση του μηνύματος γίνεται μέσω του υπολογισμού $M = C^d (\mathrm{mod}\ N)$, όπου το d είναι το ιδιωτικό κλειδί (Rivest et al. 1978: 6).

Ελλειπτικές Καμπύλες (Elliptic Curves - EC): Τα κρυπτοσυστήματα ελλειπτικών καμπύλων βασίζονται στο δύσκολο πρόβλημα διακριτού λογαρίθμου σε ελλειπτικές καμπύλες (Elliptic Curve Discrete Logarithm Problem – ECDLP). Σύμφωνα με αυτό, για δοθέντα σημεία P, Q μίας ελλειπτικής καμπύλης E τέτοιων ώστε να υπάρχει ακέραιος k για τον οποίο να ισχύει Q = kP, η εύρεση του k είναι υπολογιστικά αδύνατη. Το βασικότερο πλεονέκτημα των συστημάτων που βασίζονται στις ελλειπτικές καμπύλες είναι ότι χρησιμοποιούν σημαντικά μικρότερο μέγεθος κλειδιού σε σχέση με τον RSA. Επιπρόσθετα, το πρόβλημα επίλυσης του διακριτού λογάριθμου σε ελλειπτικές καμπύλες παρουσιάζει μεγαλύτερη πολυπλοκότητα σε σχέση με το πρόβλημα διακριτού λογάριθμου (Discrete Logarithm Problem – DLP), σύμφωνα με τους βέλτιστους, μέχρι σήμερα, αλγόριθμους.

4.2.1 Επιλογή Ασύμμετρου Αλγόριθμου Κρυπτογράφησης

Στον επεξεργαστή θα χρησιμοποιηθούν οι ελλειπτικές καμπύλες τόσο για την ασφαλή ανταλλαγή του μυστικού συμμετρικού κλειδιού όσο και για την αυθεντικοποίηση και τον έλεγχο ακεραιότητας των δεδομένων. Η επιλογή αυτή οφείλεται στην δυνατότητα των ελλειπτικών καμπύλων να παρέχουν αντίστοιχα επίπεδα ασφάλειας με τον RSA με τη χρήση σημαντικά μικρότερων σε μέγεθος κλειδιών κρυπτογράφησης. Ένα κλειδί μεγέθους 160 bits σε ένα κρυπτοσύστημα ελλειπτικής καμπύλης προσφέρει κρυπτογραφική ασφάλεια ίδιου επιπέδου με αυτή που επιτυγχάνεται με ένα κλειδί μεγέθους 1024 bits σε κρυπτοσυστήματα που χρησιμοποιούν τον αλγόριθμο RSA (Jurišic & Menezes 1997: 12). Αυτή η ιδιότητα των ελλειπτικών καμπύλων καθιστά τα κρυπτοσυστήματα ελλειπτικών καμπύλων αισθητά πιο αποδοτικά όσον αφορά σημαντικές παραμέτρους όπως η κατανάλωση ενέργειας, το μέγεθος των μεταδιδόμενων δεδομένων κλπ. Ένα κρυπτοσύστημα ελλειπτικών καμπύλων με κλειδί μεγέθους 160 bits χρειάζεται μόλις το 1/12 της ενέργειας του αντίστοιχου RSA κρυπτοσυστήματος με κλειδί μεγέθους 1024 bits (Wander et al. 2005: 1). Στις δορυφορικές επικοινωνίες, όπου οι πόροι είναι περιορισμένοι, εξοικονόμηση ενέργειας αυτής της τάξης μεγέθους είναι ιδιαίτερα σημαντική.

4.2.2 Ασφαλής Ανταλλαγή Συμμετρικού Κλειδιού – Πρωτόκολλο Diffie-Hellman

Η συμμετρική κρυπτογράφηση μπορεί να είναι πιο γρήγορη από την ασύμμετρη αλλά πάσχει από το πρόβλημα της ασφαλούς ανταλλαγής του κρυφού κλειδιού. Αυτό το πρόβλημα λύνεται με τη χρήση ασύμμετρων αλγόριθμων κρυπτογράφησης, οι οποίοι μας επιτρέπουν να κρυπτογραφήσουμε και να ανταλλάξουμε με ασφάλεια το μυστικό συμμετρικό κλειδί.

Για την ασφαλή ανταλλαγή του συμμετρικού κλειδιού χρησιμοποιήθηκε στον επεξεργαστή το πρωτόκολλο ανταλλαγής κλειδιών των Diffie και Hellman. Πρόκειται για το πρώτο ασύμμετρο πρωτόκολλο εδραίωσης κλειδιού. Η ασφάλεια του πρωτοκόλλου βασίζεται στο πρόβλημα διακριτού λογαρίθμου (DLP) (Diffie & Hellman 1976). Πιο συγκεκριμένα για την υλοποίηση του επεξεργαστή χρησιμοποιήθηκε η υλοποίηση του πρωτοκόλλου Diffie-Hellman σε ελλειπτικές καμπύλες (Elliptic Curve Diffie Hellman – ECDH), η οποία βασίζεται στο δύσκολο πρόβλημα του διακριτού λογάριθμου σε ελλειπτικές καμπύλες (ECDLP) (Koblitz et al. 2000: 179).

4.2.3 Αυθεντικοποίηση και Ακεραιότητα

Για την αυθεντικοποίηση, δηλαδή την πιστοποίηση της ταυτότητας του αποστολέα, χρησιμοποιήθηκε στον επεξεργαστή ο αλγόριθμος ψηφιακών υπογραφών Elliptic Curve Digital Signature Algorithm (ECDSA), ο οποίος αποτελεί την ανάλογη υλοποίηση του Digital Signature Algorithm (DSA) προσαρμοσμένη στις ελλειπτικές καμπύλες και η ασφάλεια του βασίζεται στο πρόβλημα υπολογισμού του διακριτού λογάριθμου σε ελλειπτικές καμπύλες (ECDLP). Ο συγκεκριμένος αλγόριθμος από το 2000 αποτελεί πρότυπο του Εθνικού Ινστιτούτου Τυποποίησης και Τεχνολογίας (NIST) καθώς και του IEEE (Institute of Electrical and Electronics Engineers) (Johnson et al. 2001: 3). Η συνάρτηση κατακερματισμού που χρησιμοποιήθηκε μαζί με τον ECDSA είναι η SHA256 (Secure Hash Algorithm), η οποία παρέχει στον επεξεργαστή ότι χρειάζεται ώστε να μπορεί να επιτελεί ελέγχους ακεραιότητας των δεδομένων, έτσι ώστε εκτός από την πιστοποίηση της ταυτότητας του αποστολέα να είμαστε σε θέση να πιστοποιήσουμε και την ακεραιότητα των δεδομένων, εξασφαλίζοντας ότι δεν τροποποιήθηκαν κατά τη μετάδοση (π.χ. από κάποιο επιτιθέμενο). Η συνάρτηση κατακερματισμού SHA256 ανήκει στην οικογένεια αλγορίθμων κατακερματισμού SHA-2, η οποία προτάθηκε το 2001 με σκοπό να αντικαταστήσει τον SHA, ο οποίος είχε δημοσιευτεί το 1993.

4.2.4 Πιστοποιητικά Τύπου X509 και PKI

Οι ψηφιακές υπογραφές από μόνες τους δεν επαρκούν για να διασφαλιστεί η αυθεντικότητα της ταυτότητας των οντοτήτων που επικοινωνούν. Πρέπει να διασφαλιστεί και η αυθεντικότητα των δημόσιων κλειδιών των οντοτήτων που επικοινωνούν μέσω του δορυφορικού συστήματος, καθώς σε διαφορετική περίπτωση το σύστημα θα είναι ευάλωτο σε επιθέσεις τύπου Man-In-The-Middle, κατά τις οποίες ο επιτιθέμενος προσποιείται ότι είναι π.χ. ο παραλήπτης, αφού πρώτα έχει φροντίσει να αναχαιτίσει το μήνυμα προτού αυτό φτάσει στον πραγματικό παραλήπτη. Τα δορυφορικά συστήματα, λόγω της φύσης τους, είναι ευάλωτα σε τέτοιου είδους επιθέσεις, καθώς το σήμα είναι διαθέσιμο σε μία μεγάλη γεωγραφική περιοχή, παρά το γεγονός ότι εστιάζεται η κεραία από τον δορυφόρο προς τον σταθμό βάσης. Επίσης, είναι πιθανό κάποιος άλλος δορυφόρος, ο οποίος γυρίζει σε χαμηλότερη τροχιά, να βρεθεί στην πορεία του καναλιού επικοινωνίας, ανάμεσα στο δορυφόρο και τον σταθμό βάσης με τον οποίο επιχειρεί ο δορυφόρος να επικοινωνήσει (Soper 2009: 427). Αυτό στον επεξεργαστή επιτυγχάνεται με τη χρήση πιστοποιητικών τύπου X509 version 1

και version 3 και την ύπαρξη ενός συστήματος δημόσιου κλειδιού (Public Key Infrastructure – PKI) (Gerck 2000: 3).

Ο επεξεργαστής έχει υλοποιηθεί ώστε να υποστηρίζει τη χρήση πιστοποιητικών τύπου X509 version 1 και version 3. Η σχεδίαση ενός συστήματος δημοσίου κλειδιού (PKI) δεν είναι μέρος της παρούσας εργασίας. Για να διασφαλιστεί, όμως, η ασφάλεια του συστήματος στο οποίο θα εγκατασταθεί ο επεξεργαστής, απαραίτητη προϋπόθεση είναι το σύστημα να υποστηρίζεται από ένα PKI ώστε να διασφαλίζεται η γνησιότητα των πιστοποιητικών που χρησιμοποιούν οι χρήστες του συστήματος.

Κεφάλαιο 5
Υλοποίηση Επεξεργαστή

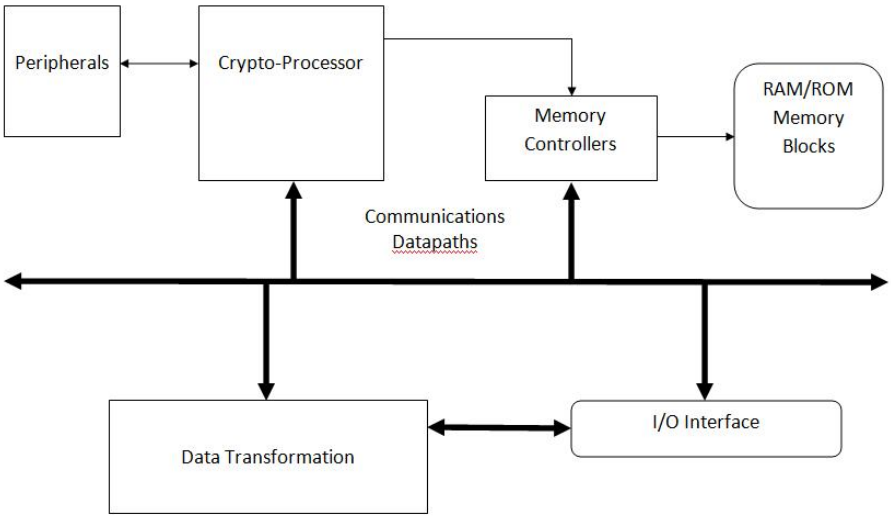
Η υλοποίηση των αλγόριθμων κρυπτογράφησης και των κρυπτογραφικών τεχνικών του επεξεργαστή έγινε στη γλώσσα προγραμματισμού Java. Χρησιμοποιήθηκε το JDK (Java Development Kit) 1.8 και ως πλατφόρμα υλοποίησης, η πλατφόρμα NetBeans IDE version 8.2.

Η προτεινόμενη αρχιτεκτονική του επεξεργαστή ακολουθεί τις βασικές αρχές σχεδίασης ενός τυπικού επεξεργαστή, αποτελούμενος μεταξύ άλλων από διαδρομές δεδομένων (datapaths), μπλοκ μνήμης (memory blocks), μονάδα ελέγχου (control unit) και διεπαφή εισόδου/εξόδου (I/O interface).

5.1 Αρχιτεκτονική Επεξεργαστή

Η προτεινόμενη για τον επεξεργαστή αρχιτεκτονική βασίστηκε σε μία System-On-A-Chip (SoC) αρχιτεκτονική σχεδιασμένη για συγκεκριμένου σκοπού εφαρμογές, όπως οι δορυφορικές επικοινωνίες (Sklavos 2012: 1). Στην Εικόνα 6 φαίνεται η βασική δομή μίας τέτοιας αρχιτεκτονικής.

Τα κύρια τμήματα μίας τέτοιας αρχιτεκτονικής είναι ο επεξεργαστής, ο οποίος μας παρέχει το σύνολο των κρυπτογραφικών αλγόριθμων και τεχνικών κρυπτογράφησης (τα οποία είναι απαραίτητα για τη διασφάλιση της ασφάλειας του συστήματος το οποίο υποστηρίζει), τα τμήματα μνήμης (RAM, ROM) και ελέγχου μνήμης (διαχειριστές μνήμης – memory controllers), διεπαφές εισόδου/εξόδου για την επικοινωνία με τα εξωτερικά συστήματα, καθώς και τις απαραίτητες διαδρομές δεδομένων (datapaths) για τη διασύνδεση των στοιχείων αυτών.



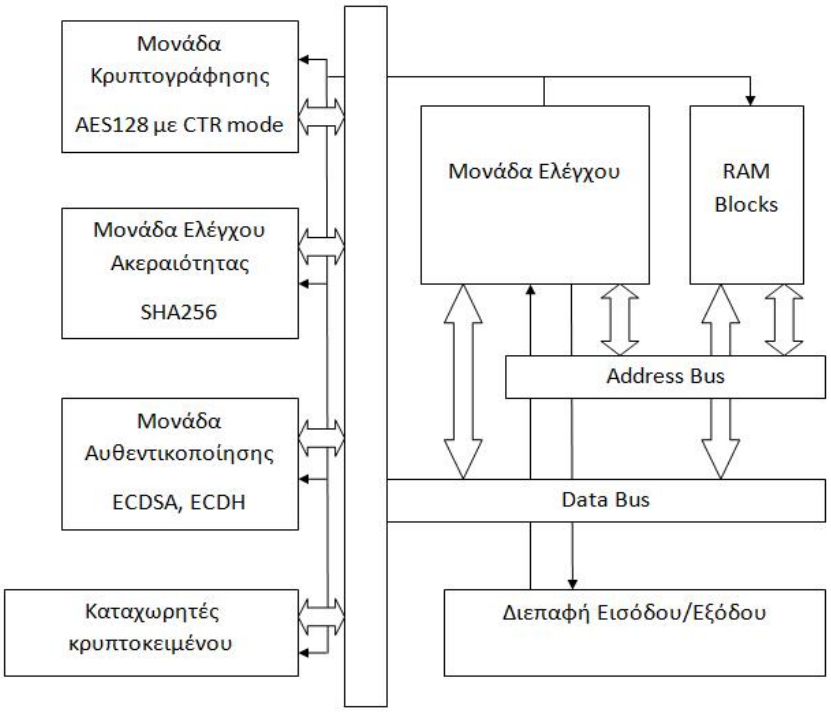
Εικόνα 6. System-On-A-Chip αρχιτεκτονική.

Ένα από τα κυριότερα πλεονεκτήματα της συγκεκριμένης αρχιτεκτονικής είναι η δυνατότητα της να τροποποιηθεί σε μία πιο σύνθετη δομή, η οποία θα μπορούσε π.χ. να περιλαμβάνει περισσότερους επεξεργαστές, να ενοποιεί περισσότερες διεπαφές εισόδου/εξόδου, να χρησιμοποιεί περισσότερες μονάδες μνήμης κλπ (Sklavos 2012: 1).

5.1.1 Ο Επεξεργαστής

Πιο συγκεκριμένα η μονάδα του επεξεργαστή αποτελείται από μία διαδρομή δεδομένων, μπλοκ μνήμης, μία διεπαφή εισόδου/εξόδου και μία μονάδα ελέγχου, ακολουθώντας τη σχεδίαση ενός τυπικού επεξεργαστή.

Η προτεινόμενη αρχιτεκτονική στηρίχθηκε σε μία WTLS αρχιτεκτονική για ένα Crypto-Processor (Sklavos et al. 2006: 35). Η συγκεκριμένη αρχιτεκτονική αποτελεί τη βάση για την αρχιτεκτονική του προτεινόμενου επεξεργαστή. Έχουν γίνει μικρές τροποποιήσεις ώστε να προσαρμοστεί στα δεδομένα και τις ανάγκες του επεξεργαστή και να υποστηρίζει τους αλγόριθμους κρυπτογράφησης που επιλέχθηκαν. Η προτεινόμενη αρχιτεκτονική παρουσιάζεται στην Εικόνα 7.



Εικόνα 7. Προτεινόμενη αρχιτεκτονική επεξεργαστή.

Ο επεξεργαστής υποστηρίζει τέσσερις διαφορετικούς αλγόριθμους κρυπτογράφησης. Ο συμμετρικός αλγόριθμος κρυπτογράφησης AES με μέγεθος κλειδιού 128 bits σε CTR mode χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων (Μονάδα Κρυπτογράφησης). Η μονάδα ακεραιότητας υποστηρίζει την SHA256 συνάρτηση κατακερματισμού (της οικογένειας αλγορίθμων κατακερματισμού SHA-2) και χρησιμοποιείται από τον επεξεργαστή για τον έλεγχο της ακεραιότητας των δεδομένων. Οι αλγόριθμοι κρυπτογράφησης ECDSA και ECDH εκτελούνται από τη μονάδα αυθεντικοποίησης, παρέχοντας στον επεξεργαστή τη δυνατότητα ταυτοποίησης του αποστολέα και ασφαλούς ανταλλαγής των συμμετρικών κλειδιών τα οποία θα χρησιμοποιηθούν για την κρυπτογράφηση των δεδομένων τα οποία θα μεταδοθούν μέσω του δορυφορικού συστήματος.

Στον επεξεργαστή χρησιμοποιείται επίσης ένα κοινό data bus 64 bits και ένα address bus 32 bits, τα οποία εξυπηρετούν τις εσωτερικές ανάγκες του επεξεργαστή όσον αφορά τη μετάδοση των δεδομένων. Τα μπλοκ μνήμης τεχνολογίας RAM χρησιμοποιούνται για την αποθήκευση των απαραίτητων για τη λειτουργία των κρυπτογραφικών αλγορίθμων κλειδιών, για όλους τους κρυπτογραφικούς αλγόριθμους τους οποίους ο επεξεργαστής υποστηρίζει. Η δεύτερη μονάδα αποθήκευσης του επεξεργαστή (καταχωρητές κρυπτοκειμένου) χρησιμοποιείται για την αποθήκευση των

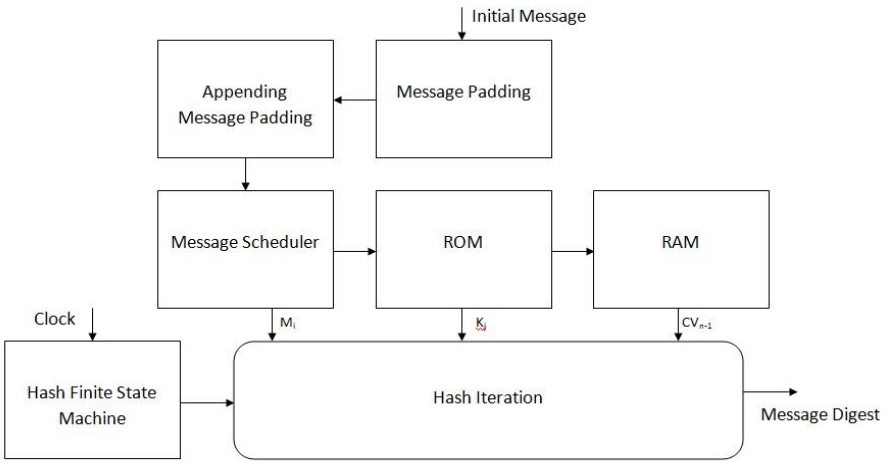
τροποποιημένων δεδομένων (είτε αυτά είναι κρυπτοκείμενο είτε το αρχικό κείμενο ανάλογα με το αν ο επεξεργαστής επιτελεί εκείνη τη στιγμή κρυπτογράφηση ή αποκρυπτογράφηση). Τα τροποποιημένα δεδομένα παραμένουν αποθηκευμένα στη συγκεκριμένη μονάδα για όσο είναι απαραίτητο.

Για την επικοινωνία με το εξωτερικό περιβάλλον υπεύθυνη είναι η Διεπαφή Εισόδου/Εξόδου, η οποία υποστηρίζει 32bit δεδομένα εισόδου και 32bit address buses. Μέσω της συγκεκριμένης μονάδας διασφαλίζεται η αποτελεσματική διασύνδεση του επεξεργαστή με τα υπόλοιπα εξωτερικά συστήματα.

5.1.2 Αρχιτεκτονική Συνάρτησης Κατακερματισμού

Οι συναρτήσεις κατακερματισμού είναι μία κατηγορία κρυπτογραφικών αλγόριθμων η οποία χαρακτηρίζεται από το μεγάλο αριθμό επαναλαμβανόμενων γύρων που πρέπει να εκτελεστούν για τη δημιουργία της σύνοψης του μηνύματος (message digest). Κάθε γύρος αποτελείται από τα ίδια βήματα. Το αποτέλεσμα κάθε γύρου δίνεται ως είσοδος στον επόμενο γύρο.

Η πιο κοινή αρχιτεκτονική για συναρτήσεις κατακερματισμού είναι η αρχιτεκτονική επαναλαμβανόμενου βρόχου (iterative loop architecture). Η αρχιτεκτονική αυτή προκρίθηκε για να χρησιμοποιηθεί στη μονάδα ακεραιότητας του επεξεργαστή. Την συγκεκριμένη αρχιτεκτονική βλέπουμε αναλυτικά στην Εικόνα 8 (Sklavos 2012: 4).



Εικόνα 8. Αρχιτεκτονική επαναλαμβανόμενου βρόχου.

Όπως φαίνεται και στην παραπάνω εικόνα, οι απαιτούμενες σταθερές και αρχικές τιμές της συνάρτησης κατακερματισμού φορτώνονται από τα ROM μπλοκ μνήμης, ενώ μπλοκ μνήμης τεχνολογίας RAM χρησιμοποιούνται για να αποθηκεύουμε τα τμήματα εκείνα κάθε γύρου τα οποία θα δοθούν ως είσοδος στον επόμενο γύρο. Η σύνοψη του μηνύματος παράγεται στο τέλος με XOR. Η συγκεκριμένη αρχιτεκτονική έχει καλή απόδοση και δεν απαιτεί τη δέσμευση πολλών πόρων (Sklavos 2012: 4).

5.2 Η Γλώσσα Προγραμματισμού Java

Η γλώσσα προγραμματισμού Java αποτελεί προϊόν της Sun Microsystems Inc. και παρουσιάστηκε επίσημα από τη Sun το 1995 στο συνέδριο Sun World 1995. Δημιουργός της Java είναι ο James Gosling, ο οποίος εκείνη την περίοδο εργαζόταν για την εταιρεία Sun. Η πρώτη έκδοση JDK 1.0 κυκλοφόρησε το 1996. Η εξέλιξη της από την επίσημη παρουσίαση της και έπειτα ήταν ραγδαία και σταθερά ανοδική. Σήμερα αποτελεί, πλέον, μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού.

Η γλώσσα προγραμματισμού Java παρουσιάζει μία σειρά από πλεονεκτήματα, όπως η απλότητα όσον αφορά το σχεδιασμό και τη χρήση της και η αξιοπιστία της. Η Java, η οποία σχεδιάστηκε με βάση τη C++, σχεδιάστηκε με στόχο να είναι πιο απλή και κατανοητή από τη C++ και για το λόγο αυτό αρκετά σπανίως χρησιμοποιούμενα και δυσκολονόητα χαρακτηριστικά της C++ παραλείφθηκαν. Διεργασίες όπως η αυτόματη συλλογή σκουπιδιών (automatic garbage collection) – αυτοματοποιημένη διαδικασία κατά την οποία μνήμη που δεν χρησιμοποιείται αποδεσμεύεται περιοδικά - εισήχθησαν στην Java έτσι ώστε να απλοποιηθεί η διαδικασία διαχείρισης της μνήμης, η οποία αποτελεί πηγή πολυπλοκότητας στις γλώσσες προγραμματισμού C και C++. Η λανθασμένη διαχείριση της μνήμης μπορεί να οδηγήσει και σε κενά ασφάλειας. Ένα από τα σημαντικότερα πλεονεκτήματα της Java είναι η δυνατότητα να τρέξουμε το ίδιο πρόγραμμα σε διαφορετικά συστήματα καθώς ως γλώσσα προγραμματισμού είναι ανεξάρτητη του λειτουργικού συστήματος και της πλατφόρμας στην οποία τρέχει. Η ιδιότητα της αυτή οφείλεται στον τρόπο με τον οποίο έχει σχεδιαστεί. Τα προγράμματα μεταφράζονται σε Java Virtual Machine (JVM) κώδικα, ο οποίος ονομάζεται bytecode και είναι ανεξάρτητος της μηχανής στην οποία τρέχει. Στη γλώσσα προγραμματισμού Java το πρόγραμμα μεταφράζεται μία φορά και ο κώδικας bytecode που δημιουργείται μπορεί να τρέξει σε οποιαδήποτε πλατφόρμα. Το μόνο που χρειάζεται να διαθέτει μία μηχανή είναι ένα διερμηνευτή (interpreter) Java για να μπορέσει να τρέξει ένα

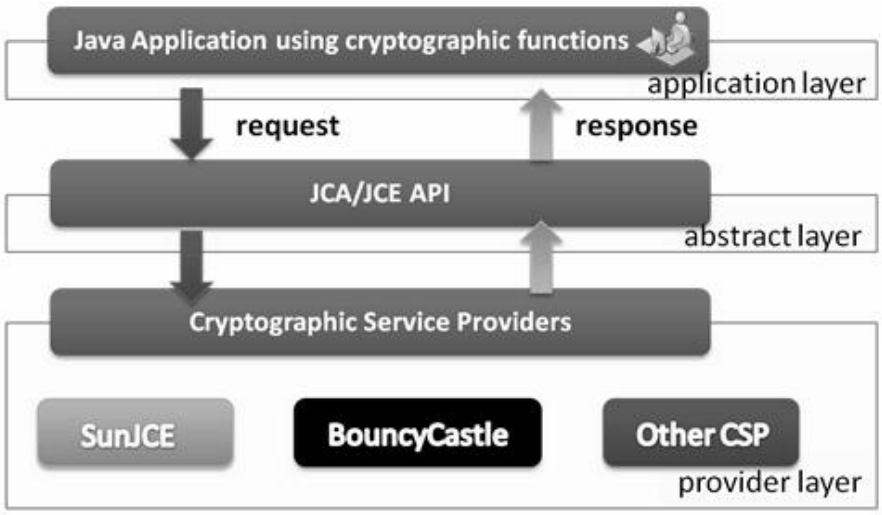
πρόγραμμα υλοποιημένο σε Java, δηλαδή τον κώδικα bytecode ο οποίος έχει παραχθεί από τον μεταφραστή (compiler) κατά τη μετάφραση του προγράμματος. Ως γλώσσα προγραμματισμού η Java θεωρείται ιδιαίτερα ασφαλής. Εν αντιθέσει με τις C και C++ η Java δεν χρησιμοποιεί δείκτες (pointers), γεγονός που εξαλείφει την πιθανότητα ακούσιας ή μη έγκυρης τροποποίησης της μνήμης.

Η Java ως αντικειμενοστραφής γλώσσα προγραμματισμού διαθέτει και όλα τα πλεονεκτήματα τα οποία μας προσφέρουν οι συγκεκριμένες γλώσσες προγραμματισμού, όπως είναι η συνέπεια στη γλώσσα που χρησιμοποιείται σε όλα τα στάδια της ανάπτυξης και η απόκρυψη των λεπτομερειών υλοποίησης, μέσω της κελυφοποίησης των δεδομένων και της συμπεριφοράς μέσα στις κλάσεις. Γενικά οι αντικειμενοστραφείς γλώσσες προγραμματισμού είναι αποτελεσματικές, γρήγορες και αφαιρετικές, στοιχεία που τις καθιστούν ιδανικές για τη δημιουργία μεγάλων και σύνθετων εφαρμογών.

Από την άλλη η Java παρουσιάζει και ορισμένα μειονεκτήματα, όπως στον τομέα των επιδόσεων όπου μπορεί να θεωρηθεί πιο αργή, κυρίως λόγω της ύπαρξης του διερμηνευτή, και πιο απαιτητική όσον αφορά την κατανάλωση μνήμης σε σύγκριση με γλώσσες προγραμματισμού όπως η C ή η C++. Χαρακτηριστικό μειονέκτημα της Java θεωρείται, επίσης, η δυσχρηστία που παρουσιάζει όσον αφορά τη διαχείριση του GUI (Graphics User Interface) κατά την ανάπτυξη εφαρμογών.

5.3 Ο Πάροχος Bouncy Castle

Η βασική λειτουργικότητα για τη χρήση κρυπτογραφικών τεχνικών στη Java παρέχεται μέσω των JCA (Java Cryptography Architecture) και JCE (Java Cryptography Extension). Οι δύο αυτοί όροι, σήμερα, χρησιμοποιούνται για να περιγράψουν την ίδια αρχιτεκτονική. Η αρχιτεκτονική JCA βασίζεται στους παρόχους (providers). Αποτελεί ένα ενδιάμεσο επίπεδο, ανάμεσα στην εφαρμογή που έχει υλοποιηθεί σε Java και στον πάροχο των κρυπτογραφικών τεχνικών, ο οποίος παρέχει το πλαίσιο το οποίο υλοποιεί τις κλάσεις και κατ' επέκταση τις συναρτήσεις που υλοποιούν τους αντίστοιχους κρυπτογραφικούς αλγόριθμους.



Εικόνα 9. Η αρχιτεκτονική JCA.

Το JCA έχει οριστεί ως ένα ανεξάρτητο επίπεδο. Τα JCA και JCE παρέχουν ένα σύνολο από κλάσεις και διεπαφές (interfaces), τα οποία χρησιμοποιούνται για να έχουμε πρόσβαση στους κρυπτογραφικούς αλγόριθμους. Αυτό σημαίνει ότι η κρυπτογράφηση, η αποκρυπτογράφηση και λοιπές κρυπτογραφικές λειτουργίες δεν υλοποιούνται από το JCA αλλά από τον εκάστοτε πάροχο. Μέσω του JCA έχουμε ένα μηχανισμό μέσω του οποίου μπορούμε να προσθέτουμε και να χρησιμοποιούμε συγκεκριμένους πάροχους. Η JCA αρχιτεκτονική, μάλιστα, είναι σχεδιασμένη με τέτοιο τρόπο ώστε να μπορούμε να χρησιμοποιήσουμε περισσότερους του ενός παρόχους ή να επιλέξουμε ένα πάροχο κατά την εκτέλεση της εφαρμογής.

Ένας από τους παρόχους κρυπτογραφικών αλγόριθμων σε Java, γνωστοί και ως Cryptography Service Providers (CSPs), είναι και ο Bouncy Castle πάροχος. Υπάρχουν και άλλοι CSPs, μεταξύ των οποίων είναι και ο SunJCE πάροχος, τον οποίο προσφέρει η εταιρεία Sun και περιέχεται στο JDK. Ο SunJCE όμως, δεν παρέχει, τουλάχιστον μέχρι σήμερα, παρά τις υλοποιήσεις για ορισμένους μόνο κρυπτογραφικούς αλγόριθμους, ενώ δεν υποστηρίζει την δημιουργία πιστοποιητικών. Επίσης, ο SunJCE πάροχος επιβάλλει μία σειρά από περιορισμούς, οι οποίοι προέρχονται από το σχετικό νομοθετικό πλαίσιο των Ηνωμένων Πολιτειών της Αμερικής, όσον αφορά τα μεγέθη κλειδιών των παρεχόμενων αλγόριθμων κρυπτογράφησης, π.χ. στον αλγόριθμο AES το μέγιστο επιτρεπόμενο μέγεθος κλειδιού είναι τα 128 bits. Οι παραπάνω περιορισμοί καθιστούν τον συγκεκριμένο πάροχο περισσότερο δυσλειτουργικό και λιγότερο ελκυστικό για χρήση σε εφαρμογές, όπου πολλές φορές η πλήρης αξιοποίηση των παρεχόμενων αλγόριθμων είναι απαραίτητη. Στον επεξεργαστή χρησιμοποιήθηκε ο Bouncy Castle

πάροχος. Πρόκειται για έναν από τους πιο γνωστούς και διαδεδομένους παρόχους, ο οποίος μας προσφέρει μία μεγάλη γκάμα κρυπτογραφικών αλγόριθμων και υπηρεσιών, όπως φαίνεται και στον Πίνακα1. Επίσης, οι κρυπτογραφικοί αλγόριθμοι του Bouncy Castle πάροχου δεν διέπονται από τους περιορισμούς που διέπουν τις αντίστοιχες υλοποιήσεις των αλγόριθμων αυτών από τον SunJCE πάροχο, καθώς έχουν δημιουργηθεί στην Αυστραλία.

Παρεχόμενη Υπηρεσία
A lightweight cryptography API.
A provider for the Java Cryptography Extension and the Java Cryptography Architecture.
A clean room implementation of the JCE 1.2.1.
A library for reading and writing encoded ASN.1 objects.
Lightweight APIs for TLS (RFC 2246, RFC 4346) and DTLS (RFC 4347).
Generators for Version 1 and Version 3 X.509 certificates, Version 2 CRLs, and PKCS12 files.
Generators for Version 2 X.509 attribute certificates.
Generators/Processors for S/MIME and CMS (PKCS7/RFC 3852).
Generators/Processors for OCSP (RFC 2560).
Generators/Processors for TSP (RFC 3161 & RFC 5544).
Generators/Processors for CMP and CRMF (RFC 4210 & RFC 4211).
Generators/Processors for OpenPGP (RFC 4880).
Generators/Processors for Extended Access Control (EAC).
Generators/Processors for Data Validation and Certification Server (DVCS) - RFC 3029.
Generators/Processors for DNS-based Authentication of Named Entities (DANE).
Signed jar versions suitable for JDK 1.4-1.8 and the Sun JCE.

Πίνακας 1. Υπηρεσίες και κρυπτογραφικοί αλγόριθμοι που παρέχονται από τον Bouncy Castle πάροχο.

Το API το οποίο μας παρέχει ο Bouncy Castle πάροχος είναι συμβατό με οποιαδήποτε έκδοση της Java από την J2ME μέχρι και το JDK 1.8, ενώ προσφέρει και τη δυνατότητα δημιουργίας πιστοποιητικών, η οποία είναι απαραίτητη στον επεξεργαστή, καθώς η σχεδίαση του προβλέπει όχι μόνο τη δυνατότητα χρήσης πιστοποιητικών τύπου X509 αλλά και δημιουργίας τους.

5.4 Processor Class

Ο επεξεργαστής υλοποιήθηκε ως ένα Java class (το Processor class), το οποίο μας παρέχει όλες τις απαραίτητες συναρτήσεις, οι οποίες υλοποιούν τις βασικές λειτουργίες του επεξεργαστή, δηλαδή την κρυπτογράφηση, την αποκρυπτογράφηση, την αυθεντικοποίηση και τον έλεγχο ακεραιότητας των δεδομένων. Επίσης, παρέχονται συναρτήσεις για τη δημιουργία και διαχείριση πιστοποιητικών (certificates) και αιτημάτων πιστοποιητικών (certificate requests). Κάθε στιγμιότυπο της κλάσης αυτής διαθέτει ένα συμμετρικό κλειδί, το οποίο χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των υπό μετάδοση δεδομένων, και ένα ζεύγος δημόσιου/ιδιωτικού κλειδιού, το οποίο χρησιμοποιείται για τη δημιουργία ψηφιακών υπογραφών. Όπως αναφέρθηκε και στο Κεφάλαιο 4 για τη συμμετρική κρυπτογράφηση χρησιμοποιείται ο αλγόριθμος AES και για την ασύμμετρη κρυπτογράφηση οι ελλειπτικές καμπύλες.

Μία σειρά από Constructors (βλέπε Πίνακα 2) είναι διαθέσιμοι για τη δημιουργία ενός στιγμιότυπου του επεξεργαστή. Κάθε στιγμιότυπο διαθέτει ένα ζεύγος κλειδιών για την ασύμμετρη κρυπτογράφηση, το οποίο ουσιαστικά είναι το ζεύγος κλειδιών της οντότητας η οποία έχει δημιουργήσει το στιγμιότυπο με σκοπό να επικοινωνήσει με ασφάλεια (κάνοντας χρήση των συναρτήσεων του επεξεργαστή) μέσω του δορυφορικού συστήματος. Επίσης, σε κάθε στιγμιότυπο του επεξεργαστή παρέχεται ένα συμμετρικό κλειδί, το οποίο χρησιμοποιείται κατά την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων. Τα πεδία αυτά, που περιέχουν τα κλειδιά της συμμετρικής και ασύμμετρης κρυπτογράφησης, έχουν ορισθεί ως ιδιωτικά (private) και μπορούν να χρησιμοποιηθούν μόνο από τις συναρτήσεις της κλάσης στην οποία ορίζονται.

Constructor
public Processor()
public Processor(PublicKey pubKey)
public Processor(KeyPair pair, PublicKey pubKey)
public Processor(X509Certificate cert)
public Processor(KeyPair pair, X509Certificate cert)

Πίνακας 2. Constructors του επεξεργαστή.

Κάθε constructor χρησιμοποιεί αρχικά την κλάση Security και τη μέθοδο addProvider που μας παρέχει ώστε να προσθέσει, δυναμικά και όχι στατικά, τον Bouncy Castle

πάροχο, τις κρυπτογραφικές συναρτήσεις του οποίου χρησιμοποιεί. Οι constructors που δέχονται ως όρισμα το δημόσιο κλειδί ή το πιστοποιητικό της οντότητας με την οποία θέλουμε να επικοινωνήσουμε, είτε για να αποστείλουμε δεδομένα είτε για να λάβουμε δεδομένα, χρησιμοποιούνται όταν κατά τη δημιουργία του στιγμιότυπου γνωρίζουμε με ποιον θα επικοινωνήσουμε. Αυτή η περίπτωση είναι και η επικρατέστερη. Στην Εικόνα 10 βλέπουμε τον κώδικα ενός τέτοιου constructor. Δύο βασικές λειτουργίες εκτελούνται στους constructors αυτούς, η προσθήκη του πάροχου και η δημιουργία του μυστικού συμμετρικού κλειδιού.

```
public Processor(KeyPair pair, PublicKey pubKey)
    throws  NoSuchAlgorithmException,
            InvalidKeyException,
            NoSuchProviderException,
            InvalidKeySpecException
{
    Security.addProvider(new BouncyCastleProvider());
    ecKeyPair = pair;

   /*Generate shared key for symmetric cryptography (AES)*/
   aesKey = generateSharedKeyForAES(ecKeyPair.getPrivate(), pubKey);
}
```

Εικόνα 10. Υλοποίηση ενός εκ των constructors της κλάσης Processor.

Η συνάρτηση generateSharedKeyForAES (την υλοποίηση της οποίας μπορούμε να δούμε στην Εικόνα 11) καλείται έτσι ώστε να δημιουργηθεί το κοινό μυστικό συμμετρικό κλειδί. Για την ασφαλή ανταλλαγή του κοινού μυστικού κλειδιού χρησιμοποιείται η υλοποίηση του πρωτοκόλλου Diffie-Hellman για ελλειπτικές καμπύλες (ECDH) που μας παρέχει ο Bouncy Castle πάροχος. Η κλάση KeyAgreement, η οποία παρέχει την λειτουργικότητα για πρωτόκολλα ανταλλαγής κλειδιού, χρησιμοποιείται έτσι ώστε να δημιουργηθεί ένα στιγμιότυπο της κλάσης το οποίο υλοποιεί την εκδοχή του Bouncy Castle πάροχου για το Diffie-Hellman πρωτόκολλο για ελλειπτικές καμπύλες. Έπειτα η μέθοδος init της κλάσης αυτής καλείται, με όρισμα το ιδιωτικό κλειδί της οντότητας που έχει δημιουργήσει το στιγμιότυπο του επεξεργαστή έτσι ώστε να ξεκινήσει η διαδικασία ανταλλαγής του κλειδιού. Η μέθοδος doPhase, η οποία δέχεται ως όρισμα το δημόσιο κλειδί της οντότητας με την οποία θέλουμε να συμφωνήσουμε το μυστικό κοινό συμμετρικό κλειδί, μας οδηγεί στην επόμενη και τελευταία φάση της ανταλλαγής. Η ανταλλαγή ολοκληρώνεται με τη χρήση της μεθόδου generateSecret, η οποία δέχεται ως όρισμα τον αλγόριθμο συμμετρικής

κρυπτογράφησης για τον οποίο θέλουμε να δημιουργήσουμε το μυστικό κλειδί. Στην υλοποίηση του αλγόριθμου όπως αυτός προσφέρεται από τον Bouncy Castle πάροχο το κλειδί που δημιουργείται έχει, εξ' ορισμού, μέγεθος 192 bits. Στην υλοποίηση του επεξεργαστή όμως χρησιμοποιούμε κλειδί μεγέθους 128 bits για την κρυπτογράφηση και την αποκρυπτογράφηση. Το κλειδί το οποίο χρησιμοποιείται τελικά, αποτελείται από τα πρώτα 128 bits του κλειδιού που δημιουργήθηκε από τον ECDH αλγόριθμο.

```
public static SecretKey generateSharedKeyForAES(PrivateKey privKey, PublicKey pubKey)
    throws  NoSuchAlgorithmException,
            InvalidKeyException,
            NoSuchProviderException,
            InvalidKeySpecException
{
        /*Agree on common symmetric session key using Diffie Hellman*/
        KeyAgreement keyAgr = KeyAgreement.getInstance("ECDH", "BC");
        keyAgr.init(privKey);
        keyAgr.doPhase(pubKey, true);

        SecretKey key = keyAgr.generateSecret("AES");

        /*Create from the 192 bits generated key a new 128 bits key by using
        the originals key's first 128 bits*/
        SecretKey newKey = new SecretKeySpec(key.getEncoded(), 0, 16, "AES");

        return newKey;
}
```

Εικόνα 11. Υλοποίηση της συνάρτησης generateSharedKeyForAES.

Εάν δεν παρέχεται το ζεύγος κλειδιών, μέσω των ορισμάτων, κατά τη δημιουργία του στιγμιότυπου (περίπτωση κατά την οποία χρησιμοποιείται ο κενός Constructor) τότε ο επεξεργαστής παράγει ένα ζεύγος κλειδιών για ελλειπτικές καμπύλες μέσω της συνάρτησης generateKeyPairForEC (βλέπε Εικόνα 12). Η προεπιλεγμένη ελλειπτική καμπύλη, η οποία χρησιμοποιείται σε αυτή την περίπτωση είναι η καμπύλη "brainpoolp256r1", η οποία είναι μία από τις ελλειπτικές καμπύλες τις οποίες μας διαθέτει ο πάροχος. Οι συγκεκριμένες καμπύλες χρησιμοποιούν τυχαίους πρώτους αριθμούς, γεγονός που τις καθιστά λιγότερο αποδοτικές, σε σύγκριση με καμπύλες οι οποίες δεν χρησιμοποιούν τυχαίους πρώτους αριθμούς. Ο λόγος που επελέγει μία τέτοια καμπύλη, αν και λιγότερο αποδοτική, είναι επειδή αυτές οι καμπύλες είναι περισσότερο ασφαλείς. Ο επεξεργαστής μπορεί, βέβαια, να λειτουργήσει και με οποιαδήποτε άλλη ελλειπτική καμπύλη. Για την δημιουργία του ζεύγους κλειδιών χρησιμοποιείται η κλάση KeyPairGenerator, η οποία είναι η κλάση της Java που χρησιμοποιείται για τη δημιουργία ζεύγους ιδιωτικού/δημόσιου κλειδιού. Η αρχικοποίηση του KeyPairGenerator στιγμιότυπου που έχει δημιουργηθεί γίνεται με κλήση της συνάρτησης initialize στην οποία παρέχεται και κάποια τυχαία (μέσω της κλάσης

SecureRandom) ποσότητα. Η γεννήτρια είναι έτοιμη για χρήση και με κλήση της μεθόδου generateKeyPair δημιουργείται το ζεύγος κλειδιών.

```
public static KeyPair generateKeyPairForEC()
    throws  NoSuchAlgorithmException,
            NoSuchProviderException,
            InvalidAlgorithmParameterException
{
    /*The brainpoolp256r1 curve from named curves is used for elliptic
    cryptography*/
    ECNamedCurveParameterSpec paramSpec = ECNamedCurveTable.getParameterSpec("brainpoolp256r1");

    /*Generate key pair using Elliptic Curves with Diffie Hellman*/
    KeyPairGenerator keyGen = KeyPairGenerator.getInstance("ECDH", "BC");

    SecureRandom random = new SecureRandom();

    keyGen.initialize(paramSpec, random);

    return keyGen.generateKeyPair();
}
```

Εικόνα 12. Υλοποίηση της συνάρτησης generateKeyPairForEC.

5.4.1 Κρυπτογράφηση/Αποκρυπτογράφηση

Για την κρυπτογράφηση και την αποκρυπτογράφηση χρησιμοποιείται ο αλγόριθμος AES σε CTR mode με μέγεθος κλειδιού 128 bits. Αρχικά καλείται η συνάρτηση generateSharedKeyForAES έτσι ώστε να γίνει μέσω του αλγόριθμου Diffie-Hellman για ελλειπτικές καμπύλες (ECDH) η ασφαλής ανταλλαγή του μυστικού κοινού συμμετρικού κλειδιού κρυπτογράφησης, το οποίο θα χρησιμοποιηθεί, κατά τη διάρκεια της επικοινωνίας, για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων τα οποία θα μεταδοθούν μέσω του δορυφορικού συστήματος. Η υλοποίηση του ECDH αλγόριθμου είναι εκείνη που μας παρέχεται από τον Bouncy Castle πάροχο. Η κλήση της μεθόδου αυτής γίνεται συνήθως στον constructor. Μοναδική εξαίρεση αποτελεί η δημιουργία στιγμιότυπου με χρήση του κενού constructor. Σε αυτή την περίπτωση πρέπει πρώτα να γίνει κλήση της μεθόδου setSharedKeyForAES πριν προχωρήσουμε στην κρυπτογράφηση και αποκρυπτογράφηση. Έπειτα για την κρυπτογράφηση των δεδομένων μία από τις συναρτήσεις encrypt (βλέπε Πίνακα 3) καλείται. Υπάρχουν τρεις διαφορετικές υλοποιήσεις της συνάρτησης encrypt διαθέσιμες, οι οποίες καλύπτουν την κρυπτογράφηση των δεδομένων σε διάφορες μορφές (κρυπτογράφηση byte[] array, κρυπτογράφηση String και κρυπτογράφηση αρχείου).

Συνάρτηση Κρυπτογράφησης
public byte[] encrypt(byte[] plaintext)
public String encrypt(String plaintext)
public void encrypt(File inputFile, File outputFile)

Πίνακας 3. Συναρτήσεις κρυπτογράφησης επεξεργαστή.

Την υλοποίηση μίας εκ των μεθόδων encrypt μπορούμε να δούμε στην Εικόνα 13. Η συνάρτηση createCtrIvForAES χρησιμοποιείται για τη δημιουργία του διανύσματος αρχικοποίησης (initialization vector) της Counter mode λειτουργίας του AES. Η λειτουργικότητα ενός κρυπτογραφικού αλγόριθμου στη Java προσφέρεται μέσω της κλάσης Cipher. Ένα στιγμιότυπο της συγκεκριμένης κλάσης δημιουργείται το οποίο υλοποιεί τον αλγόριθμο AES σε CTR mode όπως αυτός παρέχεται από τον Bouncy Castle πάροχο. Έπειτα το στιγμιότυπο της κλάσης Cipher που έχει δημιουργηθεί αρχικοποιείται μέσω της μεθόδου init, σε λειτουργία κρυπτογράφησης (ENCRYPT mode) και με κλειδί κρυπτογράφησης το κοινό μυστικό συμμετρικό κλειδί που διαθέτει ο επεξεργαστής. Για την κρυπτογράφηση των δεδομένων καλείται η μέθοδος doFinal της κλάσης Cipher.

```
public byte[] encrypt(byte[] plaintext)
    throws  NoSuchAlgorithmException,
            NoSuchPaddingException,
            InvalidKeyException,
            InvalidAlgorithmParameterException,
            IllegalBlockSizeException,
            BadPaddingException,
            NoSuchProviderException
{
    /*Create initialization vector for counter mode*/
    ivSpec = createCtrIvForAES();

    /*Initialize cipher on encryption mode. The algorithm to be used is AES
    with counter mode*/
    aesCipher = Cipher.getInstance("AES/CTR/PKCS5PADDING", "BC");
    aesCipher.init(Cipher.ENCRYPT_MODE, aesKey, ivSpec);

    return aesCipher.doFinal(plaintext);
}
```

Εικόνα 13. Υλοποίηση μίας εκ των παρεχόμενων μεθόδων κρυπτογράφησης.

Για την αποκρυπτογράφηση χρησιμοποιούνται οι αντίστοιχες συναρτήσεις decrypt (βλέπε Πίνακα 4). Και στην περίπτωση της αποκρυπτογράφησης εάν δεν έχει προηγουμένως δημιουργηθεί το κοινό μυστικό κλειδί, τότε πρώτα δημιουργείται με κλήση της συνάρτησης setSharedKeyForAES και έπειτα γίνεται η αποκρυπτογράφηση.

Συνάρτηση Αποκρυπτογράφησης
public String decrypt(String cipherText, IvParameterSpec ivSpec)
public byte[] decrypt(byte[] cipherText, IvParameterSpec ivSpec)
public void decrypt(File inputFile, File outputFile, IvParameterSpec ivSpec)

Πίνακας 4. Συναρτήσεις αποκρυπτογράφησης επεξεργαστή.

Όπως και κατά την κρυπτογράφηση έτσι και στην αποκρυπτογράφηση χρησιμοποιείται ένα στιγμιότυπο της κλάσης Cipher. Μοναδική αλλαγή είναι ότι αυτή τη φορά το στιγμιότυπο της κλάσης Cipher που δημιουργείται αρχικοποιείται σε λειτουργία αποκρυπτογράφησης (DECRYPT mode). Όπως και στην κρυπτογράφηση η μέθοδος doFinal καλείται για να γίνει η αποκρυπτογράφηση και να ανακτηθεί το αρχικό μήνυμα.

```
public byte[] decrypt(byte[] cipherText, IvParameterSpec ivSpec)
    throws  NoSuchAlgorithmException,
            NoSuchPaddingException,
            InvalidKeyException,
            InvalidAlgorithmParameterException,
            IllegalBlockSizeException,
            BadPaddingException,
            NoSuchProviderException
{
    /*Initialize cipher on decryption mode. The algorithm to be used is AES
    with counter mode*/
    aesCipher = Cipher.getInstance("AES/CTR/PKCS5PADDING", "BC");
    aesCipher.init(Cipher.DECRYPT_MODE, aesKey, ivSpec);

    return aesCipher.doFinal(cipherText);
}
```

Εικόνα 14. Υλοποίηση μίας εκ των παρεχόμενων μεθόδων αποκρυπτογράφησης.

5.4.2 Αυθεντικοποίηση/Ακεραιότητα

Για την αυθεντικοποίηση και τον έλεγχο ακεραιότητας των δεδομένων χρησιμοποιείται η υλοποίηση του αλγόριθμου ECDSA (Elliptic Curve Digital Signature Algorithm), όπως αυτή μας παρέχεται από τον πάροχο Bouncy Castle. Η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η SHA256.

Η συνάρτηση signData του επεξεργαστή (βλέπε Εικόνα 15) είναι εκείνη η οποία δημιουργεί την ψηφιακή υπογραφή για τα δεδομένα που θέλουμε να μεταδώσουμε. Η κλάση Signature της Java χρησιμοποιείται για την παροχή της λειτουργικότητας των αλγόριθμων ψηφιακών υπογραφών. Ένα στιγμιότυπο της κλάσης δημιουργείται δίνοντας την ανάλογη περιγραφή του κρυπτογραφικού αλγόριθμου που πρόκειται να χρησιμοποιηθεί (στην περίπτωση του επεξεργαστή SHA256withECDSA). Έπειτα το ιδιωτικό κλειδί της οντότητας που θέλει να υπογράψει τα δεδομένα χρησιμοποιείται για την αρχικοποίηση του στιγμιότυπου της κλάσης Signature που έχει δημιουργηθεί. Η μέθοδος update ενημερώνει τα δεδομένα για τα οποία στη συνέχεια δημιουργείται η

ψηφιακή υπογραφή με κλήση της μεθόδου sign της κλάσης Signature. Η μέθοδος sign επιστρέφει την ψηφιακή υπογραφή των δεδομένων σε μορφή byte[].

```
public byte[] signData(byte[] data)
    throws  NoSuchAlgorithmException,
            InvalidKeyException,
            SignatureException
{
    /*Sign data with SHA256withECDSA*/
    Signature signer = Signature.getInstance("SHA256withECDSA");
    signer.initSign(ecKeyPair.getPrivate());
    signer.update(data);

    return (signer.sign());
}
```

Εικόνα 15. Μέθοδος δημιουργίας ψηφιακής υπογραφής δεδομένων.

Για τον έλεγχο της εγκυρότητας της υπογραφής, αλλά και της ακεραιότητας των δεδομένων, χρησιμοποιείται μία από τις συναρτήσεις verifySig, οι οποίες επιστρέφουν Αληθές εάν ο αποστολέας είναι αυθεντικός και τα δεδομένα δεν έχουν παραποιηθεί και Ψευδές εάν είτε ο αποστολέας δεν είναι αυτός που ισχυρίζεται ότι είναι, είτε τα δεδομένα έχουν παραποιηθεί είτε και τα δύο. Όπως και κατά την υπογραφή των δεδομένων, έτσι και τώρα, ένα στιγμιότυπο της κλάσης Signature χρησιμοποιείται. Το στιγμιότυπο που έχει δημιουργηθεί, αυτή τη φορά, αρχικοποιείται με το δημόσιο κλειδί του αποστολέα και χρήση της μεθόδου initVerify (αντί της μεθόδου initSign η οποία χρησιμοποιήθηκε κατά την διαδικασία υπογραφής των δεδομένων). Το στιγμιότυπο είναι πλέον έτοιμο για έλεγχο αυθεντικοποίησης της ταυτότητας του αποστολέα και ακεραιότητας των δεδομένων. Η μέθοδος update ενημερώνει τα προς επιβεβαίωση δεδομένα και η μέθοδος verify τα ελέγχει.

```
public boolean verifySig(byte[] data, PublicKey key, byte[] sig)
    throws  NoSuchAlgorithmException,
            InvalidKeyException,
            SignatureException
{
    /*Verify signature signed with SHA256withECDSA algorithm*/
    Signature signer = Signature.getInstance("SHA256withECDSA");
    signer.initVerify(key);
    signer.update(data);
    return (signer.verify(sig));
}
```

Εικόνα 16. Μέθοδος ελέγχου ψηφιακής υπογραφής.

Στον παρακάτω πίνακα μπορούμε να δούμε τις υπογραφές όλων των διαθέσιμων συναρτήσεων του επεξεργαστή για δημιουργία και έλεγχο των ψηφιακών υπογραφών.

Συνάρτηση
public byte[] signData(byte[] data)
public boolean verifySig(byte[] data, PublicKey key, byte[] sig)
public static boolean verifySig(byte[] data, X509Certificate cert, byte[] sig)

Πίνακας 5. Συναρτήσεις δημιουργίας και ελέγχου ψηφιακών υπογραφών.

Ο επεξεργαστής έχει σχεδιαστεί με τέτοιο τρόπο ώστε να μπορεί να χειρίζεται πιστοποιητικά τύπου X509 version 1 και version 3. Η αυθεντικότητα των πιστοποιητικών αυτών πρέπει να διασφαλίζεται από ένα σύστημα δημόσιου κλειδιού (PKI). Πέρα από τις συναρτήσεις για την διαχείριση των πιστοποιητικών, όπως η συνάρτηση obtainX509CertificateFromFile η οποία διαβάζει ένα αρχείο και επιστρέφει ένα τύπου X509 πιστοποιητικό, παρέχονται συναρτήσεις για τη δημιουργία X509 πιστοποιητικών (βλέπε Πίνακας 6). Τα πιστοποιητικά μπορούν να είναι είτε version 1 είτε version 3 και είναι αυτό-υπογραφόμενα (self signed). Επίσης, δίνεται η δυνατότητα για δημιουργία αιτήματος πιστοποιητικού (certificate request) για πιστοποιητικά τύπου X509 version 1 και version 3 (βλέπε Πίνακας 6).

Συνάρτηση
public X509Certificate createSelfSignedV1X509Certificate (X500Name issuer, Date startDate, Date expiryDate, BigInteger serialNumber, X500Name subject, String signatureAlgorithm)
public X509Certificate createSelfSignedV3X509Certificate (X500Name issuer, Date startDate, Date expiryDate, BigInteger serialNumber, X500Name subject, String signatureAlgorithm, Extension[] extension)
public PKCS10CertificationRequest createV1X509CertificateRequest (X500Name subject, String signatureAlgorithm)
public PKCS10CertificationRequest createV3X509CertificateRequest (X500Name subject, String signatureAlgorithm, ASN1ObjectIdentifier oid, boolean critical, byte[] value, ASN1ObjectIdentifier attrType)
public X509Certificate obtainX509CertificateFromFile(String pathToFile)

Πίνακας 6. Συναρτήσεις δημιουργίας και διαχείρισης πιστοποιητικών και αιτημάτων.

5.5 Λειτουργία Επεξεργαστή

Παρακάτω θα παρουσιαστεί μία τυπική περίπτωση χρήσης του επεξεργαστή τόσο σε διάταξη πομπού, κατά την οποία τα δεδομένα που θέλουμε να μεταδώσουμε θα κρυπτογραφηθούν για να μεταδοθούν με ασφάλεια, όσο και σε διάταξη δέκτη, κατά την οποία τα ληφθέντα κρυπτογραφημένα δεδομένα θα αποκρυπτογραφηθούν και θα ελεγχθούν για να διασφαλιστεί η αυθεντικότητα του αποστολέα και η ακεραιότητα τους. Τα βήματα τα οποία θα ακολουθηθούν σε κάθε περίπτωση για την ασφαλή αποστολή και λήψη των δεδομένων θα αναλυθούν στις επόμενες ενότητες. Και στις δύο περιπτώσεις θεωρούμε ότι ο αποστολέας και ο παραλήπτης έχουν ήδη δημιουργήσει τα ζεύγη ιδιωτικού/δημόσιου κλειδιών τους. Επίσης, τα δημόσια κλειδιά τους είναι διαθέσιμα μέσω κάποιου μηχανισμού PKI, ο οποίος διασφαλίζει την αυθεντικότητα των πιστοποιητικών.

5.5.1 Διάταξη Πομπού

Στην περίπτωση κατά την οποία ο επεξεργαστής λειτουργεί ως πομπός θα δημιουργηθεί αρχικά ένα στιγμιότυπο του επεξεργαστή. Κατά τη δημιουργία του στιγμιότυπου του επεξεργαστή, δίνονται ως ορίσματα το ζεύγος κλειδιών του πομπού και το πιστοποιητικό του δέκτη, το οποίο περιέχει το δημόσιο κλειδί του. Κατά τη δημιουργία του στιγμιότυπου, όπως αναλύσαμε προηγουμένως ότι συμβαίνει με όλους τους constructors (μοναδική εξαίρεση όπως είδαμε αποτελεί ο κενός constructor), δημιουργείται και το μυστικό κοινό συμμετρικό κλειδί κρυπτογράφησης του AES, μέσω της κλήσης της συνάρτησης generateSharedKeyForAES. Μετά τη δημιουργία του στιγμιότυπου του επεξεργαστή και του κοινού μυστικού συμμετρικού κλειδιού, ο πομπός είναι πλέον σε θέση να κρυπτογραφήσει τα δεδομένα τα οποία θέλει να στείλει μέσω του δορυφορικού συστήματος. Πριν την κρυπτογράφηση, όμως, των δεδομένων, θα πρέπει να προηγηθεί η δημιουργία της υπογραφής τους. Επομένως, τα δεδομένα πρώτα υπογράφονται χρησιμοποιώντας τη συνάρτηση signData (βλέπε Εικόνα 15), η οποία θα μας επιστρέψει την ψηφιακή υπογραφή των δεδομένων. Έπειτα θα γίνει η κρυπτογράφηση των δεδομένων και της ψηφιακής υπογραφής τους. Για λόγους ασφάλειας θα πρέπει να κρυπτογραφείται και η ψηφιακή υπογραφή μαζί με τα δεδομένα. Η κρυπτογράφηση των δεδομένων και της ψηφιακής υπογραφής τους θα γίνει με χρήση μίας εκ των μεθόδων encrypt τις οποίες μας παρέχει ο επεξεργαστής (π.χ. της μεθόδου encrypt της Εικόνας 13). Μετά την κρυπτογράφηση τους τα δεδομένα είναι

πλέον έτοιμα για να σταλούν, με ασφάλεια, στο δέκτη μέσω του δορυφορικού συστήματος.

Βήμα
Βήμα 1: Ανάκτηση ζεύγους κλειδιών πομπού.
Βήμα 2: Ανάκτηση πιστοποιητικού δέκτη.
Βήμα 3: Δημιουργία στιγμιότυπου επεξεργαστή χρησιμοποιώντας το δημόσιο κλειδί του δέκτη και το ζεύγος κλειδιών του πομπού.
Βήμα 4: Δημιουργία υπογραφής για τα δεδομένα με χρήση της αντίστοιχης συνάρτησης.
Βήμα 5: Κρυπτογράφηση δεδομένων και υπογραφής με χρήση κατάλληλης συνάρτησης.
Βήμα 6: Τα κρυπτογραφημένα δεδομένα στέλνονται μέσω του δορυφορικού συστήματος με ασφάλεια στον δέκτη.

Πίνακας 7. Βήματα που ακολουθούνται σε μία τυπική περίπτωση λειτουργίας του επεξεργαστή ως πομπός.

5.5.2 Διάταξη Δέκτη

Στην περίπτωση κατά την οποία ο επεξεργαστής λειτουργεί ως δέκτης, όπως και όταν λειτουργεί ως πομπός, θα δημιουργηθεί αρχικά ένα στιγμιότυπο του επεξεργαστή με χρήση ενός εκ των constructors που μας παρέχονται. Όπως και προηγουμένως, αμέσως μετά την κλήση του constructor στον οποίο θα έχουμε εισάγει ως ορίσματα το ζεύγος κλειδιών του δέκτη και το πιστοποιητικό του πομπού, το οποίο περιέχει το δημόσιο κλειδί του, θα έχουμε δημιουργήσει και θα είναι διαθέσιμο προς χρήση ένα στιγμιότυπο του επεξεργαστή το οποίο θα διαθέτει το αντίστοιχο μυστικό κοινό συμμετρικό κλειδί. Ακολούθως, τα κρυπτογραφημένα δεδομένα και η υπογραφή τους, τα οποία θα έχουν προηγουμένως ληφθεί μέσω του δορυφορικού συστήματος, θα αποκρυπτογραφηθούν χρησιμοποιώντας μία εκ των διαθέσιμων μεθόδων decrypt του επεξεργαστή (όπως π.χ. η μέθοδος decrypt της Εικόνας 14). Μετά την αποκρυπτογράφηση των δεδομένων και της υπογραφής τους, θα ελεγχθεί η ψηφιακή υπογραφή των δεδομένων, χρησιμοποιώντας μία εκ των μεθόδων verifySig, τις οποίες μας παρέχει ο επεξεργαστής (όπως η μέθοδος ελέγχου ψηφιακής υπογραφής verifySig της Εικόνας 16). Η μέθοδος verifySig που θα χρησιμοποιηθεί θα ελέγξει τόσο την ταυτότητα του αποστολέα (αυθεντικοποίηση) όσο και την ακεραιότητα των δεδομένων (ακεραιότητα). Εφόσον, επιβεβαιωθεί επιτυχώς η ταυτότητα του αποστολέα και η ακεραιότητα των δεδομένων,

τα αποκρυπτογραφημένα πλέον δεδομένα θα έχουν ληφθεί επιτυχώς και θα μπορούν να χρησιμοποιηθούν από το δέκτη.

Βήμα
Βήμα 1: Ανάκτηση ζεύγους κλειδιών δέκτη.
Βήμα 2: Ανάκτηση πιστοποιητικού πομπού.
Βήμα 3: Δημιουργία στιγμιότυπου επεξεργαστή χρησιμοποιώντας το δημόσιο κλειδί του πομπού και το ζεύγος κλειδιών του δέκτη.
Βήμα 4: Λήψη κρυπτογραφημένων δεδομένων και της ψηφιακής υπογραφής τους μέσω του δορυφορικού συστήματος.
Βήμα 5: Αποκρυπτογράφηση των δεδομένων και της ψηφιακής υπογραφής τους με χρήση κατάλληλης συνάρτησης.
Βήμα 6: Έλεγχος ταυτότητας αποστολέα και ακεραιότητας των δεδομένων με χρήση της αντίστοιχης συνάρτησης.

Πίνακας 8. Βήματα που ακολουθούνται σε μία τυπική περίπτωση λειτουργίας του επεξεργαστή ως δέκτης.

Κεφάλαιο 6
Μελλοντική Έρευνα

Οι δορυφορικές επικοινωνίες βρίσκονται στο επίκεντρο του ενδιαφέροντος της ερευνητικής κοινότητας, λόγω των πολλών εφαρμογών που έχουν μέχρι σήμερα. Αναμένεται να έχουν σημαντικό ρόλο και στο μέλλον καθώς πρόκειται για ένα συνεχώς εξελισσόμενο τομέα. Η παρούσα μεταπτυχιακή διατριβή δίνει τη δυνατότητα για μελλοντικές έρευνες οι οποίες θα χρησιμοποιούν τεχνολογίες οι οποίες βρίσκονται ακόμη σήμερα σε πρώιμα στάδια, όσον αφορά την έρευνα, αλλά αναμένεται να διαδραματίσουν σημαντικό ρόλο στο μέλλον, όπως οι δορυφορικές επικοινωνίες μέσω κβαντικού καναλιού (Bacsardi 2007), η κβαντική κρυπτογραφία (Hunter 2002) και η κβαντική διανομή κλειδιού (Quantum Key Distribution - QKD) (Bascardi 2005, Toyoshima et al. 2008). Ιδιαίτερα η κβαντική διανομή κλειδιού είναι ένας τομέας που παρουσιάζει εξαιρετικό ενδιαφέρον, ενώ υπάρχουν ήδη κάποιες επιτυχημένες προσπάθειες - έστω και αν βρισκόμαστε ακόμα μακριά από τη χρήση της τεχνολογίας αυτής σε πρακτικό επίπεδο - διανομής του κλειδιού (Miao et al. 2007). Η συγκεκριμένη τεχνολογία θα μπορούσε στο μέλλον να χρησιμοποιηθεί για να αντικαταστήσει τον PKI μηχανισμό τον οποίο χρειαζόμαστε σήμερα για να διασφαλίσουμε την αυθεντικότητα των πιστοποιητικών τα οποία χειρίζεται ο επεξεργαστής.

Κεφάλαιο 7
Συμπεράσματα

Σκοπός αυτής της μεταπτυχιακής διατριβής ήταν η ανάπτυξη και ο σχεδιασμός ενός επεξεργαστή ο οποίος θα εξυπηρετεί τις αυξημένες ανάγκες για κρυπτογράφηση δεδομένων, τα οποία θα μεταδίδονται από ένα δορυφορικό σύστημα. Αφού πρώτα μελετήθηκε η φύση των δορυφορικών συστημάτων και οι ιδιαιτερότητες των δορυφορικών επικοινωνιών επελέγησαν οι αλγόριθμοι κρυπτογράφησης εκείνοι, οι οποίοι μετά από σχετική έρευνα και ανάλυση που έγινε, καλύπτουν καλύτερα τις ανάγκες των δικτύων αυτών. Ο στόχος ήταν να επιτύχουμε τη διασφάλιση της ασφάλειας των μεταδιδόμενων δεδομένων αλλά και να εξασφαλίσουμε παράλληλα την καλύτερη δυνατή απόδοση του επεξεργαστή. Η υλοποίηση του επεξεργαστή έγινε σε γλώσσα προγραμματισμού Java, εξασφαλίζοντας με αυτό τον τρόπο τη δυνατότητα ο επεξεργαστής να μπορεί να χρησιμοποιηθεί σε διαφορετικές πλατφόρμες (π.χ. κινητά τηλέφωνα, τηλεοράσεις, GPS συσκευές κλπ). Εκμεταλλευόμενοι τη σχετική ιδιότητα της συγκεκριμένης γλώσσας προγραμματισμού.

Ο επεξεργαστής που παρουσιάστηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής καλύπτει πλήρως τις βασικές απαιτήσεις ασφάλειας, δηλαδή κρυπτογράφηση, αποκρυπτογράφηση, αυθεντικοποίηση και έλεγχο ακεραιότητας δεδομένων, τόσο σε διάταξη πομπού όσο και σε διάταξη δέκτη. Η επιλογή των κρυπτογραφικών αλγόριθμων έγινε με γνώμονα την εξασφάλιση του απαραίτητου επίπεδου ασφάλειας για τα μεταδιδόμενα δεδομένα και την επίτευξη της καλύτερης δυνατής απόδοσης όσον αφορά τις επιδόσεις του συστήματος.

Κεφάλαιο 8
Επίλογος

Οι δορυφορικές επικοινωνίες αποτελούν ένα τομέα μεγάλου ερευνητικού ενδιαφέροντος. Λόγω της φύσης των δορυφορικών επικοινωνιών η ασφάλεια των μεταδιδόμενων σε αυτά δεδομένων είναι πρώτιστης σημασίας. Στην παρούσα μεταπτυχιακή διατριβή αναλύθηκε, σχεδιάστηκε και υλοποιήθηκε ένας επεξεργαστής ο οποίος καλύπτει τις αυξημένες ανάγκες για κρυπτογράφηση των μεταδιδόμενων δεδομένων (κρυπτογράφηση, πιστοποίηση και ακεραιότητα), τα οποία μεταδίδονται μέσω ενός δορυφορικού συστήματος. Πέρα από τη διασφάλιση της ασφάλειας των δεδομένων δόθηκε έμφαση και στην βελτιστοποίηση της απόδοσης του επεξεργαστή μέσω της επιλογής των κρυπτογραφικών αλγόριθμων εκείνων οι οποίοι εξασφαλίζουν το απαραίτητο επίπεδο ασφάλειας για τα μεταδιδόμενα δεδομένα και την καλύτερη δυνατή απόδοση του επεξεργαστή όσον αφορά βασικές παραμέτρους, όπως η απόδοση (throughput), οι δεσμευμένοι πόροι, η κατανάλωση ενέργειας κλπ. Η παρούσα μεταπτυχιακή διατριβή μπορεί να αποτελέσει τη βάση για μελλοντικές έρευνες οι οποίες θα εντάσσουν στον επεξεργαστή καινοτόμες τεχνολογίες, όπως η κβαντική κρυπτογραφία, οι οποίες σήμερα είναι ακόμη σε πρώιμο στάδιο και δεν μπορούν να χρησιμοποιηθούν άμεσα σε εφαρμογές.

Βιβλιογραφία

Anderson, R., Biham, E., & Knudsen, L. (1998). Serpent: A proposal for the advanced encryption standard. *NIST AES Proposal*, *174*, 1-23.

Bascardi, L. (2005). Using quantum computing algorithms in future satellite communications. *Acta Astronautica*, *57*, 224-229.

Bacsardi, L. (2007). Satellite communication over quantum channel. *Acta Astronautica*, *61*(1), 151-159.

Daemen, J., & Rijmen, V. (2013). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

Dandalis, A., Prasanna, V. K., & Rolim, J. D. (2000, August). A comparative study of performance of AES final candidates using FPGAs. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 125-140). Springer Berlin Heidelberg.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, *22*(6), 644-654.

Dray, J. (2000, April). NIST Performance Analysis of the Final Round Java™ AES Candidates. In *The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD* (pp. 149-160).

Feistel, H. (1973). Cryptography and computer privacy. *Scientific american*, *228*, 15-23.

Feistel, H., Notz, W. A., & Smith, J. L. (1975). Some cryptographic techniques for machine-to-machine data communications. *Proceedings of the IEEE*, *63*(11), 1545-1554.

Gerck, E. (2000). Overview of Certification Systems: X. 509, PKIX, CA, PGP & SKIP. *The Bell*, *1*(3), 8.

Hamalainen, P., Hannikainen, M., Hamalainen, T., & Saarinen, J. (2001). Configurable hardware implementation of triple-DES encryption algorithm for wireless local area network. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on* (Vol. 2, pp. 1221-1224). IEEE.

Heron, S. (2009). Advanced encryption standard (AES). *Network Security*, *2009*(12), 8-12.

Hodjat, A., & Verbauwhede, I. (2004, April). A 21.54 Gbits/s fully pipelined AES processor on FPGA. In *Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on* (pp. 308-309). IEEE.

Hunter,P. (2002) Quantum Cryptography Revisited, In *Network Security, Volume 2002, Issue 8*, 14-16.

Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security*, *1*(1), 36-63.

Jurišic, A., & Menezes, A. (1997). Elliptic curves and cryptography. *Dr. Dobb's Journal*, 26-36.

Koblitz, N., Menezes, A., & Vanstone, S. (2000). The state of elliptic curve cryptography. In *Towards a quarter-century of public key cryptography* (pp. 103-123). Springer US.

Miao, E. L., Han, Z. F., Zhang, T., & Guo, G. C. (2007). The feasibility of geostationary satellite-to-ground quantum key distribution. *Physics Letters A*, *361*(1), 29-32.

Preneel, B., Van Rompay, B., Örs, S., Biryukov, A., Granboulan, L., Dottax, E., ... & Biham, E. (2003). Performance of optimized implementations of the NESSIE primitives. *NESSIE Report, Deliverable D*, *12*.

Rais, M. H., & Qasim, S. M. (2009). A novel FPGA implementation of AES-128 using reduced residue of prime numbers based S-Box. *IJCSNS international journal of computer science and network security*, *9*(9), 305-309.

Rivest, R. L., Robshaw, M. J. B., Sidney, R., & Yin, Y. L. (1998, August). The RC6TM block cipher. In *First Advanced Encryption Standard (AES) Conference*.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., & Ferguson, N. (1998). Twofish: A 128-bit block cipher. *NIST AES Proposal*, *15*.

Sklavos, N. (2012, October). Cryptographic hardware & embedded systems for communications. In *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on* (pp. 1-6). IEEE.

Sklavos, N., Kitsos, P., Papadopoulos, K., & Koufopavlou, O. (2006). Design, architecture and performance evaluation of the wireless transport layer security. *The journal of supercomputing*, *36*(1), 33-50.

Soper, D. S. (2009). Satellite Encryption, In *Computer and Information Security Handbook* (pp. 423-431).

Sterbenz, A., & Lipp, P. (2000, April). Performance of the AES Candidate Algorithms in Java. In *AES Candidate Conference* (pp. 161-165).

Toyoshima, M., Takayama, Y., Klaus, W., Kunimori, H., Fujiwara, M., & Sasaki, M. (2008). Free-space quantum cryptography with quantum and telecom communication channels. *Acta Astronautica*, *63*(1), 179-184.

Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005, March). Energy analysis of public-key cryptography for wireless sensor networks. In *Third IEEE international conference on pervasive computing and communications* (pp. 324-328). IEEE.