Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Πληροφοριακά και Επικοινωνιακά Συστήματα*

Μεταπτυχιακή Διατριβή

Ανάπτυξη Αλγόριθμου Ενσωμάτωσης Κρυφού Κειμένου Μέσα Σε Αρχεία Εικόνας

Δημήτριος Ξυπολιάς

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Ιανουάριος 2017

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Πληροφοριακά και Επικοινωνιακά Συστήματα

Μεταπτυχιακό Πρόγραμμα Σπουδών

Μεταπτυχιακή Διατριβή

Ανάπτυξη Αλγόριθμου Ενσωμάτωσης Κρυφού Κειμένου Μέσα Σε Αρχεία Εικόνας

Δημήτριος Ξυπολιάς

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση
των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών
στις 28 Δεκεμβρίου 2016
από τη Σχολή Θετικών και Εφαρμοσμένων Επιστήμων
του Ανοικτού Πανεπιστημίου Κύπρου.

Ιανουάριος 2017

Περίληψη

Η στεγανογραφία αποτελεί μία τεχνική που εφαρμόζεται από την αρχαιότητα με σκοπό να κρύψει την ύπαρξη της επικοινωνίας. Παρουσιάζει σημαντικές διαφορές από τη κρυπτογραφία σε θεωρητικό και τεχνολογικό επίπεδο, ωστόσο οι δύο τεχνικές μπορούν να συνδυαστούν. Έχοντας ως κίνητρο το γεγονός ότι οι σύγχρονες τεχνικές στεγανογραφίας καλούνται να αξιοποιήσουν νέους τρόπους επικοινωνίας κυρίως μέσω του δικτύου κινητής τηλεφωνίας ή ασύρματων δικτύων που ωστόσο δεν διαθέτουν ακόμη σημαντικούς υπολογιστικούς πόρους, όπως κεντρικοί υπολογιστές, σκοπός της συγκεκριμένης μεταπτυχιακής διατριβής είναι η ανάπτυξη μίας τεχνικής υποκατάστασης τυχαίων λιγότερων σημαντικών θέσεων (bits) ενός ψηφιακού μέσου (εικόνας) με τις θέσεις (bits) του κρυφού μηνύματος.

Βασικές προκλήσεις του προτεινόμενου συστήματος στεγανογραφίας για εφαρμογές/ συσκευές Android, είναι η ενσωμάτωση κρυφών μηνυμάτων σε εικόνες χωρίς αυτά να γίνονται αντιληπτά και χωρίς να αλλοιώνεται η εικόνα. Επιπλέον, σημαντική πρόκληση αποτελεί το ζητούμενο η λήψη φωτογραφίας, η ενσωμάτωση μηνύματος (στεγανογραφία) και η αποστολή του μηνύματος να ολοκληρώνονται χωρίς καθυστέρηση. Σημαντική συνεισφορά της μεταπτυχιακής διατριβής αποτελεί ο αλγόριθμος στεγανογραφίας που προτείνεται (LSBv4AwRaC) και επιλέγει τις θέσεις αποθήκευσης του μηνύματος μέσα στο ψηφιακό μέσο της επικοινωνίας με τυχαίο τρόπο. Επιπλέον οι θέσεις αποθήκευσης κρυπτογραφούνται με κλειδί το οποίο είναι μοναδικό για κάθε ζεύγος αποστολέα-παραλήπτη.

Οι δοκιμές με το συγκεκριμένο σύστημα στεγανογραφίας επαλήθευσαν την ορθότητα των λειτουργιών του, την επίτευξη μεγαλύτερης ασφάλειας λόγω της τυχαιότητας, αλλά και την ικανοποιητική του απόδοση λαμβάνοντας υπόψη τους περιορισμούς που επιβάλλει το λειτουργικό σύστημα του Android. Συγκρίνοντας τον αλγόριθμο (LSBv4AwRaC) που έχει υλοποιηθεί με τον αλγόριθμο αντίστοιχου Android συστήματος (MobiStego) διαπιστώνεται ότι ο τελευταίος εμφανίζει καλύτερους χρόνους εκτέλεσης, καθώς χρησιμοποιεί στατικές και όχι τυχαίες θέσεις απόκρυψης του μηνύματος. Οι μετρήσεις όμως που αναδεικνύουν την υπεροχή της μεθοδολογίας που προτείνουμε είναι αυτές της ποιότητας της στεγανογραφημένης εικόνας, αλλά και της ασφάλειας του τελικού παραγόμενου αρχείου.

Summary

Steganography is a technique used since ancient times in order to hide the existence of communication. It presents significant differences from cryptography in theoretical and technological background, but the two techniques can be combined. This dissertation work is motivated by the fact that modern steganography techniques are invited to build new ways of communication mainly through the cellular network or wireless networks, however, do not yet have substantial computing resources such as servers. The aim of this project is the development of a technique of substitution of less important positions (bits) of a digital mean (image) with the positions (bits) of the hidden message.

Key challenges of the proposed steganography system applications/Android devices are the incorporation of hidden messages in images without being perceived and without altering the image. Furthermore, the integration (steganography) of message in the image and its communication should be without delay. An important contribution of this dissertation work is the steganography algorithm which is implemented (LSBv4AwRaC) and selects randomly the message storage locations within the digital medium of communication. Additional storage locations are encrypted with a key that is unique for each sender-receiver pair.

Tests with this steganography system verify the correctness of the functions, security assurance based on randomness and the satisfactory performance given the constraints imposed by the Android operating system. Comparing the algorithm (LSBv4AwRaC) which has been implemented in this dissertation with a relevant system for Android (MobiStego) it appears that the latter shows better runtimes since it uses static and not random positions for hiding messages. Finally the measurements which demonstrate the superiority of the methodology we propose are these for quality of stenographic image and safety of the generated file.

Περιεχομένα

Εικόνες

Πίνακες

Γλωσσάρι

JEE	Java Enterprise Edition
IP	Internet Protocol
IDE	Integrated Development Environment
RDBMS	Relational Database Management System
UAT	User Acceptance Testing
RDBMS	Relationship Database Management System
AVD	Android Virtual Device
ΕΥΔΧΚ	Εξυπηρετητή Υπηρεσιών Διαχείρισης Χρηστών & Κρυπτογράφησης

1 Εισαγωγή

Στεγανογραφία σημαίνει «καλυμμένη γραφή» και είναι η τέχνη και η επιστήμη της επικοινωνίας κατά τρόπο που να κρύβει την ύπαρξη της επικοινωνίας. Ο στόχος της στεγανογραφίας είναι να κρύψει τα μηνύματα μέσα σε άλλα αβλαβή μηνύματα με έναν τρόπο που δεν επιτρέπει οποιοδήποτε εχθρό να ανιχνεύσει ακόμη και αν είναι παρόν ένα δεύτερο κρυμμένο μυστικό μήνυμα (Kuhn , 1995). Αντίθετα, με τη κρυπτογραφία, ο εχθρός αφήνεται να ανιχνεύσει, ακούσει ή αναγνώσει και να τροποποιήσει τα μηνύματα χωρίς να είναι σε θέση να παραβιάσει ορισμένες προδιαγραφές ασφαλείας που προστατεύονται από ένα κρυπτογραφικό σύστημα. Με απλά λόγια, με την λέξη στεγανογραφία, περιγράφουμε την απόκρυψη του γεγονότος ότι κάποιο μήνυμα υπάρχει, το οποίο μπορεί να είναι και να μην είναι κρυπτογραφημένο. Η κρυπτογράφηση είναι η πρακτική της συστηματικής κωδικοποίησης της πληροφορίας για να μπορεί να αποκωδικοποιηθεί αργότερα, ενώ η στεγανογραφία είναι η πρακτική της απόκρυψης πληροφορίας. Οι δύο τεχνικές συχνά συνδυάζονται.

Η τεχνική της στεγανογραφίας χρησιμοποιείται από την αρχαιότητα μέσω διάφορων τεχνικών, και σήμερα αποτελεί σημαντικό εργαλείο των υπηρεσιών ασφαλείας των διάφορων χωρών. Ενδεικτικά, έχουν καταγραφεί κάποιες πρώτες χρήσεις της στεγανογραφίας στις Ιστορίες του Ηρόδοτου, όπου ο Δημάρατος ήθελε να ενημερώσει τους Σπαρτιάτες πως ο Ξέρξης σχεδίαζε να επιτεθεί στην Ελλάδα. Για να στείλει το μήνυμα χωρίς αυτό να ανακαλυφθεί, έξυσε το κερί από ένα ζευγάρι ξύλινες πτυσσόμενες πλάκες, έγραψε το μήνυμα στο ξύλο και στη συνέχεια το κάλυψε με κερί. Όταν οι πλάκες έφτασαν στον προορισμό τους, κανένας δεν γνώριζε τη σημασία τους μέχρις ότου μια γυναίκα, η Γοργώ, την αποκάλυψε. (Kahn ,1967)

Η στεγανογραφία σήμερα βρίσκει εκτεταμένη εφαρμογή στις ψηφιακές τεχνολογίες και επιτυγχάνεται με την ενσωμάτωση κρυφών μηνυμάτων σε κάποιο μέσο όπως σε ένα έγγραφο, μια εικόνα, ένα αρχείο ήχου ή βίντεο. Οποιοσδήποτε γνωρίζει ότι το μέσο περιέχει ένα μυστικό μήνυμα μπορεί να το πάρει αμέσως, εάν υποθέσουμε πως η μέθοδος κρυπτογράφησής του (και πιθανότατα ένα μυστικό κλειδί) είναι γνωστά. Για οποιονδήποτε άλλο το μήνυμα αυτό θα είναι τελείως αόρατο. (Denning 1999).

Στη σημερινή εποχή οι πιο συνήθεις μεθοδολογίες είναι είτε μέσω φυσικών εγγράφων ή μέσω ψηφιακών αρχείων. Οι πιο κοινοί τύποι ψηφιακών αρχείων έχουν τη κατάληξη .BMP, .PNG, .JPG, .MP3, .WAV. Με τον τρόπο αυτό το κρυμμένο μήνυμα είναι δύσκολο να εντοπιστεί μέσα στο ψηφιακό περιεχόμενο. Από την άλλη πλευρά οι παραπάνω μεθοδολογίες καλούνται να κρύψουν το μήνυμα σε θέσεις του περιεχομένου έτσι ώστε να μην γίνεται αντιληπτή η πιθανή αλλοίωση του αλλά και τελικά το παραγόμενο αρχείο να είναι διαθέσιμο προς χρήση (εικόνα, μουσική κτλ.).

Επίσης η στεγανογραφία έχει χρησιμοποιηθεί για την απόκρυψη εγκληματικών δραστηριοτήτων. Ένας κλέφτης πιστωτικών καρτών για παράδειγμα, τη χρησιμοποίησε για να κρύψει κλεμμένους αριθμούς πιστωτικών καρτών σε μια σελίδα του Ιστού στην οποία είχε εισβάλλει. Είχε αντικαταστήσει τις εικόνες της σελίδας αυτής με παρόμοιες εικόνες οι οποίες όμως περιείχαν τους αριθμούς των πιστωτικών καρτών τους οποίους με τον τρόπο αυτό παρέδωσε σε συνεργάτες του. Η περίπτωση αυτή παρουσιάζει την πιθανή χρήση εικόνων του Ιστού σαν μέσο στη διακίνηση πληροφοριών.

Η στεγανογραφία μπορεί να χρησιμοποιηθεί και για την απόκρυψη της ύπαρξης διαφόρων αρχείων στο σκληρό δίσκο ενός υπολογιστή. Οι Ross Anderson, Roger Needham και Adi Samir προτείνουν ένα στεγανογραφικό σύστημα αρχείων το οποίο θα μπορούσε να κάνει αόρατο ένα αρχείο σε οποιονδήποτε δεν γνωρίζει το όνομα και τον κωδικό του. Κάποιος που κάνει επίθεση και δεν γνωρίζει τις πληροφορίες αυτές δεν μπορεί να ξέρει εάν το αρχείο υπάρχει έστω και αν έχει πλήρη πρόσβαση τόσο στο συγκεκριμένο υπολογιστή όσο και στα προγράμματά του. Μια απλή προσέγγιση δημιουργεί ψεύτικα αρχεία έτσι ώστε τα κρυμμένα αρχεία του χρήστη είναι το XOR ενός υποτμήματος των ψεύτικων αρχείων. Το υποτμήμα αυτό επιλέγεται με τον κωδικό του χρήστη. (Denning 1999).

Στην εποχή μας κάθε πληροφορία δημιουργείται, αποθηκεύεται ή μεταδίδεται ως επί το πλείστο σε ψηφιακή μορφή. Αυτό συμβαίνει γιατί χρησιμοποιούμε συνεχώς διάφορες συσκευές που παράγουν ή χειρίζονται ψηφιακά αρχεία. Έτσι, έχουμε ψηφιακές φωτογραφικές μηχανές που παράγουν ψηφιακές εικόνες, κάμερες που παράγουν ψηφιακά αρχεία video και φυσικά υπολογιστές που παράγουν πλήθος ψηφιακών αρχείων. Λόγω της πληθώρας των ψηφιακών αρχείων οι σύγχρονες τεχνικές της Στεγανογραφίας επικεντρώνονται σε τεχνικές που μπορούν να κρύψουν ψηφιακό περιεχόμενο μέσα σε ένα ψηφιακό αρχείο χωρίς αυτό να γίνεται αντιληπτό.

Σκοπός της σύγχρονης στεγανογραφίας παραμένει ο ίδιος με αυτόν της κλασσικής στεγανογραφίας, δηλαδή η παντελής απόκρυψη του ευαίσθητου περιεχομένου μέσα σε ένα άλλο περιεχόμενο-κάλυμμα ώστε αυτό να μην γίνεται αντιληπτό. Οι σύγχρονες εφαρμογές στεγανογραφίας βρίσκουν σήμερα εκτεταμένη εφαρμογή (Maganbhai and Chouhan , 2015), (Judge , 2001) κυρίως στους παρακάτω τομείς:

(α) Μυστικές επικοινωνίες (Zhang, Wu και Zhou , 2009). Με χρήση στεγανογραφίας δεν είναι φανερή η απόκρυψη μηνυμάτων επιτυγχάνοντας την αποφυγή ελέγχου, τόσο της πλευράς του αποστολέα και του μηνύματος, αλλά και του λήπτη. Κρυφά μηνύματα ή ευαίσθητες πληροφορίες μπορούν να μεταδοθούν χωρίς να υποψιάζουν και να παρακινούν δυνητικές επιθέσεις.

(β) Tagging χαρακτηριστικών στοιχείων/ μεταδεδομένων, όπως στα κοινωνικά δίκτυα ή έξυπνες συσκευές. Μέσα σε ένα ψηφιακό μέσο π.χ. μια εικόνα, μπορούν

να ενσωματωθούν στοιχεία όπως τα ονόματα των ατόμων σε μια φωτογραφία ή η θέση λήψης σε ένα χάρτη. Τα ενσωματωμένα αυτά στοιχεία μπορούν να αναγνώσουν μόνο τα μέρη που κατέχουν το stego-κλειδί αποκωδικοποίησης.

(γ) Μηχανισμοί προστασίας πνευματικών δικαιωμάτων/ προστασίας αντιγραφής. Μέσω στεγανογραφικών μεθόδων (watermarking) εμποδίζεται η αντιγραφή ψηφιακών δεδομένων.

Ιδιαίτερα εκτεταμένη είναι και η μη νόμιμη χρήση της στεγανογραφίας στους παρακάτω τομείς:

- Επικοινωνία Ομάδων Τρομοκρατών
- Απάτη
- Hacking
- Ηλεκτρονικές Πληρωμές
- Τυχερά παιχνίδια και πορνογραφία
- Παρενόχληση
- Αδικήματα πνευματικής ιδιοκτησίας
- Κακόβουλο λογισμικό (malware, viruses)

Τελευταία, κακόβουλες επιθέσεις (Dharman and Thomas , 2016) που αξιοποιούν στεγανογραφικές μεθόδους συμβαίνουν κατ 'επανάληψη. Οι επιθέσεις αυτές είναι δύσκολο να αναγνωριστούν και να αποφευχθούν, δεδομένου ότι πραγματοποιούνται με τη χρήση κοινών ψηφιακών μέσων, όπως PNG, JPEG, MP3, MP4, κλπ κρύβοντας την παρουσία των κακόβουλων δεδομένων στον κώδικα. Στόχος των επιθέσεων αυτών είναι η παθητική διανομή κακόβουλου κώδικα κρυμμένου σε μια εικόνα ή άλλο αρχείο σε δίκτυα υπολογιστών. Οι επιθέσεις που βασίζονται σε στεγανογραφημένες εικόνες είναι αρκετά διαδεδομένες και στοχεύουν στην αναστολή λειτουργιών, την υποκλοπή πληροφοριών/ κωδικών. Για το λόγο αυτό τα λογισμικά antivirus και οι υποδομές IDS (intrusion detection system) θα πρέπει να εξελίσσονται συνεχώς ώστε να ενσωματώνουν τεχνικές ανίχνευσης κακόβουλου κώδικα σε εικόνες ή και άλλα ψηφιακά αρχεία (Shah , 2015).

Οι σύγχρονες τεχνικές στεγανογραφίας καλούνται να αξιοποιήσουν νέους τρόπους επικοινωνίας που έχουν αναπτυχθεί ραγδαία όπως η μετάδοση δεδομένων μέσω του δικτύου κινητής τηλεφωνίας (τρίτης, τέταρτης γενιάς, κ.ο.κ) ή ασύρματων δικτύων (IEEE 802.11). Ωστόσο οι συσκευές ασύρματης επικοινωνίας (κινητά τηλέφωνα) δεν διαθέτουν τους υπολογιστικούς πόρους (επεξεργαστική ισχύ, μνήμη, ταχύτητα μετάδοσης) που έχουν οι προσωπικοί ή κεντρικοί υπολογιστές. Επιπρόσθετα, το λογισμικό τους (πχ. Android, iOS) θέτει κάποιους περιορισμούς ως προς τη χρήση συγκεκριμένων τύπων αρχείων για την ενσωμάτωση κρυφών δεδομένων.

Σκοπός της συγκεκριμένης μεταπτυχιακής διατριβής είναι η ανάπτυξη μίας τεχνικής υποκατάστασης τυχαίων λιγότερων σημαντικών θέσεων (bits) ενός ψηφιακού μέσου (εικόνας) με τις θέσεις (bits) του κρυφού μηνύματος. Η τεχνική αυτή υλοποιείται σε μία Android εφαρμογή και έχει επιπλέον τις ακόλουθες δυνατότητες:

1. Ταυτοποίηση (login) χρήστη
2. Λήψη φωτογραφίας
3. Ενσωμάτωση μηνύματος στην ληφθείσα εικόνα (στεγανογραφία) και
4. Αποστολή σε μία επαφή (φίλο)

Η αντίστοιχη Android εφαρμογή του παραλήπτη λαμβάνει τη φωτογραφία και μπορεί να αποκωδικοποιήσει τις θέσεις στις οποίες έχει αποθηκευτεί το κρυμμένο μήνυμα προκειμένου να το συνθέσει και να το προβάλει στον χρήστη.

Κύριες απαιτήσεις για την σχεδιασμό της παραπάνω μεθόδου στεγανογραφίας η οποία θα ενσωματώνεται σε μία Android εφαρμογή είναι:

- Η ενσωμάτωση κρυφών μηνυμάτων σε εικόνες χωρίς αυτά να γίνονται αντιληπτά και χωρίς να αλλοιώνεται η εικόνα.
- Ο αλγόριθμος στεγανογραφίας που θα υλοποιηθεί θα πρέπει να επιλέγει τις θέσεις αποθήκευσης του μηνύματος με τυχαίο τρόπο.
- Οι θέσεις αποθήκευσης θα πρέπει να κρυπτογραφούνται με κλειδί το οποίο θα είναι μοναδικό για κάθε ζεύγος αποστολέα-παραλήπτη.
- Η λήψη φωτογραφίας, η ενσωμάτωση μηνύματος (στεγανογραφία) και η αποστολή του μηνύματος θα πρέπει να είναι χωρίς καθυστέρηση.
- Αντίστοιχα, η λήψη εικόνων και η παρουσίαση μηνυμάτων και εικόνων στο χρήστη θα πρέπει να εκτελείται επίσης χωρίς καθυστέρηση.

Η ενότητα 2 της μεταπτυχιακής διατριβής αποτελεί επισκόπηση της βιβλιογραφίας και του θεωρητικού υπόβαθρου αναφορικά με τις δύο κύριες κατηγορίες μεθοδολογιών στεγανογραφίας:

- υποκατάστασης (substitution/ spatial domain) και
- μετασχηματισμού (transform domain).

Στην ενότητα 3 παρουσιάζεται η προτεινόμενη μεθοδολογία για την υλοποίηση στεγανογραφίας και τη μετάδοση κρυφών μηνυμάτων μέσω εφαρμογών Android, καθώς και η προτεινόμενη αρχιτεκτονική και τα συστατικά της μέρη, αναδεικνύοντας την πολυπλοκότητα του εγχειρήματος μας.

Η ενότητα 4 περιγράφει την υλοποίηση του συστήματος στεγανογραφίας, τόσο ως προς την υλοποίηση της προτεινόμενης μεθοδολογίας, όσο και ως προς τις τεχνικές λεπτομέρειες υλοποίησης με περιγραφή των τεχνολογιών και των προγραμματιστικών εργαλείων που αξιοποιήθηκαν.

Η ενότητα 5 επαληθεύει την ορθότητα των λειτουργιών του συστήματος μέσω συγκεκριμένων δοκιμών (Unit Testing, UAT), καθώς και μετρήσεις ταχύτητας εκτέλεσης αλγορίθμου, ποιότητας στεγανογραφημένης εικόνας και επεξεργαστικής ισχύος.

Η ενότητα 6 αποτελεί σύνοψη της μεταπτυχιακής διατριβής και παρατίθενται τα συμπεράσματα που προέκυψαν. Επιπλέον, αναφέρονται πιθανές μελλοντικές επεκτάσεις της εφαρμογής που υλοποιήθηκε.

Στα Παραρτήματα παρατίθενται τα απαραίτητα εγχειρίδια για την παραμετροποίηση και εγκατάσταση του συστήματος που υλοποιήθηκε, οδηγός χρήσης του συστήματος, καθώς και εικόνες αποτύπωσης οθονών.

2 Θεωρητικό Υπόβαθρο

Στην παρούσα ενότητα παρατίθενται οι ευρέως χρησιμοποιούμενες μεθοδολογίες στεγανογραφίας που σύμφωνα με τη βιβλιογραφία (Shelke, Dongre and Soni , 2014) μπορούν να ομαδοποιηθούν στις δύο κύριες κατηγορίες μεθοδολογιών:

- υποκατάστασης (substitution/ spatial domain) και
- μετασχηματισμού (transform domain).

Στην πρώτη κατηγορία ανήκουν οι αλγόριθμοι, όπου τα κρυφά μηνύματα ενσωματώνονται με άμεσο τρόπο στο αρχείο εικόνας. Για κάθε εικόνα επιλέγονται συγκεκριμένα pixels στα οποία εκτελούνται εργασίες αλλαγής των τιμών τους, με βάση συγκεκριμένους αλγορίθμους. Σε αυτή την κατηγορία ο πιο δημοφιλής και απλός αλγόριθμος είναι ο αλγόριθμος εισαγωγής στο λιγότερο σημαντικό bit (least significant bits insertion method - LSB) (Johnson και Jajodia , 2008), αλλά και όσοι βασίζονται σε αυτόν, οι οποίοι περιγράφονται στην επόμενη ενότητα. Οι αλγόριθμοι αυτής της κατηγορίας χρησιμοποιούνται συχνά λόγω της δυνατότητας ενσωμάτωσης πλήθους κρυφών μηνυμάτων, αλλά και λόγω της εύκολης υλοποίησης τους.

Στην κατηγορία των αλγορίθμων μετασχηματισμού (transform domain), το κρυφό μήνυμα ενσωματώνεται στους συντελεστές μετασχηματισμού της εικόνας κάλυψης. Παραδείγματα αυτής της κατηγορίας οι Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT) και Discrete Wavelet Transform (DWT) (Shelke, Dongre and Soni , 2014). Οι αλγορίθμοι της κατηγορίας αυτής είναι πιο κατάλληλοι για εφαρμογές που απαιτούν την ανθεκτικότητα της μεθόδου σε δεδομένες επεξεργασίες. Στην πρώτη φάση εφαρμόζεται μια συνάρτηση μετασχηματισμού π.χ. DCT (Discrete Cosine Transform) (Walia and Jain , 2010). Τα δεδομένα εισάγονται μεταβάλλοντας τους επιλεγμένους συντελεστές. Η επιλογή των συντελεστών γίνεται έτσι ώστε το μήνυμα να είναι ανθεκτικό σε πιθανές αλλαγές της εικόνας. Η στεγανογραφημένη εικόνα παράγεται με την εφαρμογή του αντίστροφου μετασχηματισμού.

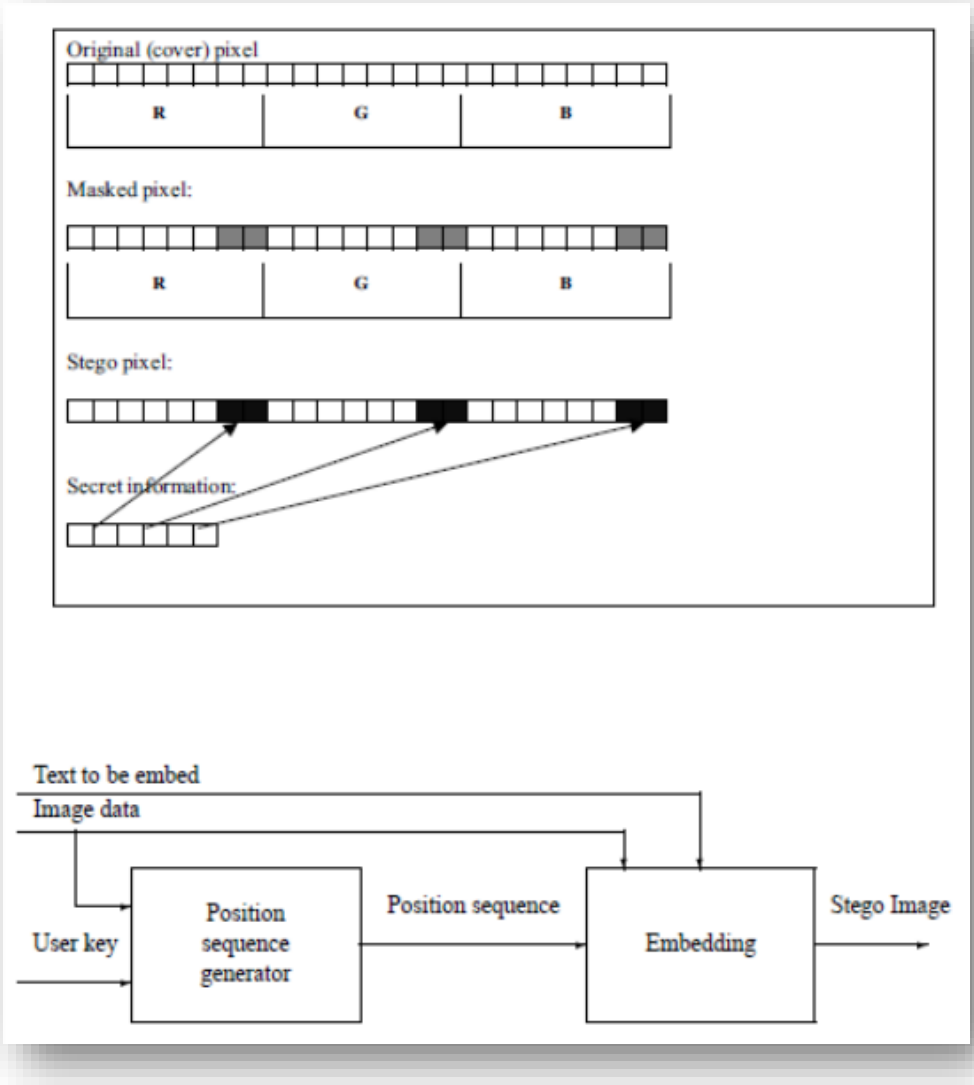
2.1 Μεθοδολογίες στεγανογραφίας

2.1.1 Μέθοδος LSB

Η μέθοδος LSB (Least Significant Bit) είναι μία από τις πιο δημοφιλής μεθόδους στεγανογραφίας (Johnson and Jajodia , 1998). Έστω ότι διαθέτουμε ένα αρχείο εικόνας. Σε ένα τέτοιο αρχείο υπάρχει τεράστιο ποσό άχρηστου ή περιττού χώρου. Το χώρο αυτό, χρησιμοποιεί η στεγανογραφία για να κρύψει κάποιο μήνυμα μέσα στο αρχείο. Η αρχή στην οποία στηρίζεται ακολουθεί την λογική ότι

το λιγότερο σημαντικό ψηφίο από κάποια ή και από όλα τα pixels μία εικόνας κάλυψης αντικαθίστανται από ένα bit του κρυφού μηνύματος.

Συγκεκριμένα, αν ένα αρχείο εικόνας που αποτελείται από κάποια bytes και κάθε byte από 8 bits, αυτά τα bits καθορίζουν την απόχρωση (πράσινου, κόκκινου, μπλε). Τα αριστερότερα bits που αποτελούν ένα byte είναι μεγαλύτερης σημαντικότητας από τα δεξιότερα. Αυτό συμβαίνει γιατί τα αριστερότερα bits καθορίζουν την απόχρωση του χρώματος με αποτέλεσμα αν αλλάξουμε το πρώτο bit της ακολουθίας, τότε το θα έχουμε μεγάλη αλλαγή χρώματος, ενώ αλλάζοντας το τελευταίο, τότε το χρώμα παραμένει σχεδόν ίδιο.



Εικόνα 1, Σχηματική απεικόνιση αλγόριθμου LSB

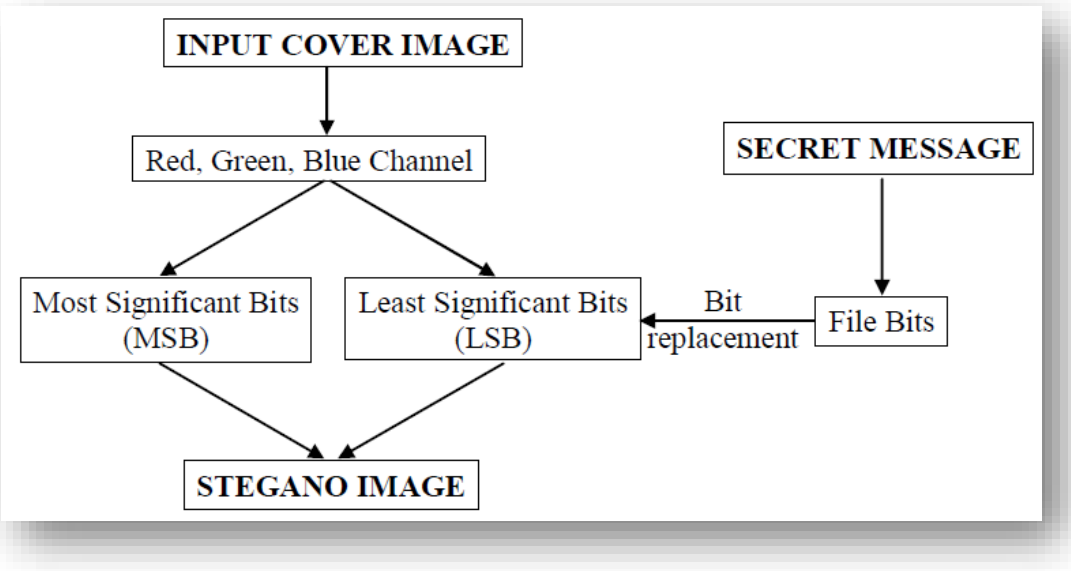
Με βάση το γεγονός ότι η μέθοδος απόκρυψης είναι γνωστή, εύκολα μπορεί κάποιος να εξάγει το κρυμμένο μήνυμα. Σε ένα πιο ασφαλές σύστημα ο αποστολέας και ο παραλήπτης χρησιμοποιούν ένα κρυφό κλειδί με το οποίο καθορίζονται τα pixels που θα χρησιμοποιηθούν. Έτσι εξασφαλίζεται ότι κάποιος

ανεπιθύμητος που δεν έχει στην κατοχή του το κλειδί δε μπορεί να γνωρίζει σε ποια pixels είναι κρυμμένο το μήνυμα και κατά συνέπεια δεν μπορεί να το αποσπάσει (Anderson and Petitcolas ,1998).

Με την εξέλιξη της στεγανογραφίας, έχουν προταθεί και αξιοποιηθεί πλειάδα παραλλαγών και βελτιωμένων εκδοχών του βασικού αλγορίθμου LSB, οι σημαντικότερες από τις οποίες παρουσιάζονται στη συνέχεια.

2.1.2 Ενισχυμένος Αλγόριθμος LSB (L2LSB)

Μία από τις πρώτες τεχνικές στεγανογραφίας ψηφιακής εικόνας είναι μια απλή τεχνική που ενσωματώνει πληροφορίες στο λιγότερο σημαντικό bit (LSB). Έστω G είναι μια εικόνα με MxN pixels, όπου κάθε εικονοστοιχείο αναπαρίσταται ως μία ακολουθία 8-bit. Η τεχνική ενσωματώνει το «μυστικό» μέρος των δεδομένων στο τελευταίο κομμάτι (δηλαδή, στο λιγότερο σημαντικό bit) των επιλεγμένων εικονοστοιχείων της εικόνας που χρησιμοποιείται προς κάλυψη. Για το ανθρώπινο μάτι, η εικόνα που προκύπτει ως αποτέλεσμα της διαδικασίας θα είναι ίδια με την αρχική εικόνα (Design of image steganography algorithms with secure message routing for p2p networks n.d.).



Εικόνα 2, Παραδοσιακή τεχνική στεγανογραφίας του λιγότερο σημαντικού δυαδικού στοιχείου (Least Significant Bit)

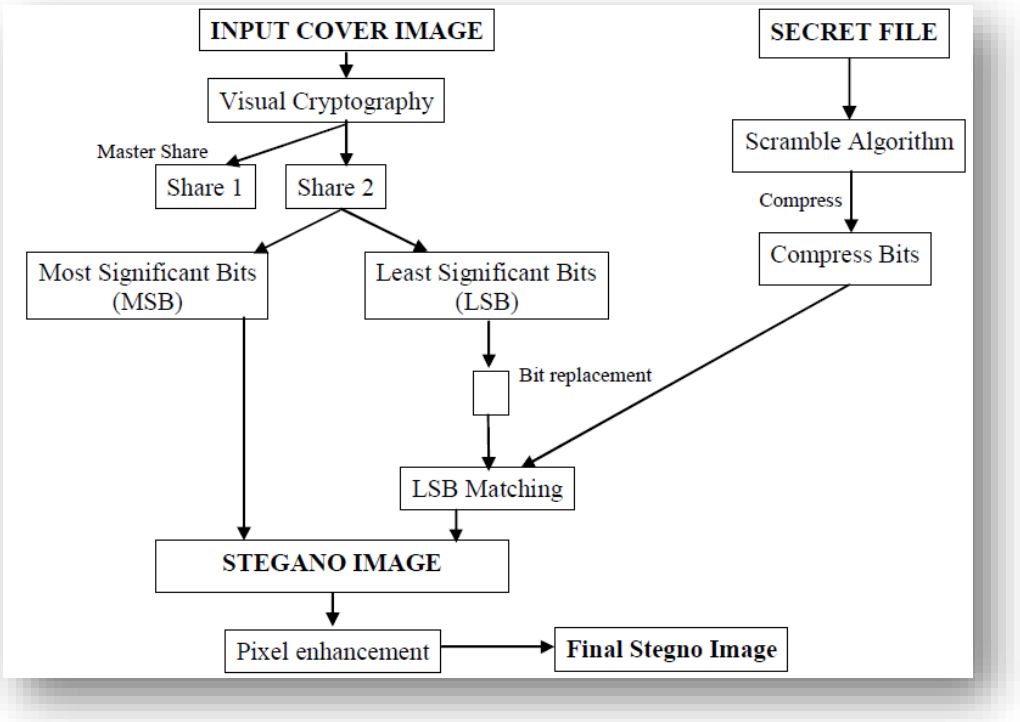
Μειονεκτήματα:

- Δεν είναι πολύ ανθεκτική στις επιθέσεις
- Η τεχνική μπορεί να ενσωματώσει μόνο σταθερού μήκους μυστικά μηνύματα

- Κατά τη διάρκεια ενσωμάτωσης, η τεχνική αυτή χρησιμοποιεί τα τρία κανάλια χρώματος για να ενσωματώσει τα bits (ένα δυαδικό στοιχείο από το R, G και B).
- Ακόμη και αν η τεχνική του Ανθρώπινου Οπτικού Συστήματος (Human Visual System- HVS) δεν παρουσιάζει μεταβολή, η στατιστική ανάλυση παρουσιάζει διαφορά και ως εκ τούτου είναι λιγότερο ασφαλής.

Για την επίλυση των παραπάνω προβλημάτων, η παραδοσιακή τεχνική LSB ενισχύεται με τον παρακάτω τρόπο:

- Αλγόριθμος κρυπτογράφησης (scrambling) χρησιμοποιείται για να τροποποιήσει το αρχικό μήνυμα, το οποίο συμπιέζεται πριν ενσωματωθεί στην εικόνα για να αποφευχθεί το πρόβλημα του σταθερού μήκους και να αυξήσει την ασφάλεια του αρχείου με το μυστικό μήνυμα
- Υλοποίηση της τεχνικής οπτικής κρυπτογράφησης για να αυξηθεί η ασφάλεια κατά τη διάρκεια επιθέσεων
- Εφαρμόζεται η διαδικασία περιορισμού των pixels (Restrict Pixel Procedure - RPP) προκειμένου να καθοριστεί ποιο κανάλι χρώματος χρησιμοποιείται για να ενσωματώσει το μυστικό μήνυμα
- Μια διαδικασία βελτίωσης των pixel χρησιμοποιείται ως ένα στάδιο επεξεργασίας εκ των υστέρων για να αυξήσει την ποιότητα της στεγανογραφημένης εικόνας
- Η τεχνική LSB ενισχύεται για να χρησιμοποιεί τρία εικονοστοιχεία αντί ενός



Εικόνα 3, Βελτιωμένη Έκδοση της τεχνικής LSB (L2LSB)

2.1.1 Βελτιωμένη μέθοδος LSB για στεγανογραφία εικόνων χρησιμοποιώντας Henon Chaotic Map

Η στεγανογραφία των εικόνων αποκρύπτει μυστικές πληροφορίες στο κάλυμμα των εικόνων και χρησιμοποιείται για ασφαλή μεταφορά δεδομένων. Διάφορες LSB (λιγότερων σημαντικών bit) προσεγγίσεις έχουν χρησιμοποιηθεί όλα αυτά τα χρόνια στην στεγανογραφία. Έχει προταθεί η χρήση μια νέας τεχνικής που βελτιώνει την συμβατική LSB τεχνική που χρησιμοποιείται στην στεγανογραφία των εικόνων (Raghava, et al. , 2014). Αυτή η καινούργια τεχνική χρησιμοποιεί μια ψευδό γεννήτρια τυχαίων αριθμών βασιζόμενη στο Henon Chaotic Map. Οι τυχαίοι αριθμοί χρησιμοποιούνται για την κρυπτογράφηση της μυστικής εικόνας η οποία ενσωματώνεται στο κάλυμμα της εικόνας. Αυτή η κρυπτογράφηση με την χρήση της ψευδό γεννήτριας τυχαίων αριθμών παρέχει επαρκή ασφάλεια για τα δεδομένα, γιατί το ίδιο σύνολο των τυχαίων αριθμών δεν μπορεί να δημιουργηθεί ξανά χωρίς κάποιος να γνωρίζει την ακριβή λειτουργία της ψευδό γεννήτριας τυχαίων αριθμών (Leung and Chen , 2009). Συνεπώς, δεν είναι εύκολο να ανακτηθούν τα μυστικά δεδομένα. Αυτή η τεχνική έχει δοκιμασθεί με επιτυχία σε διάφορα jpg αρχεία.

2.1.2 Προσαρμοστική Μέθοδος Στεγανογραφίας, LSB-PBSM

Η μέθοδος αυτή (Ms.G.S.Sravanthi, et al. , 2012) αντί να αποθηκεύει τα δεδομένα σε κάθε λιγότερο σημαντικό bit των εικονοστοιχείων (pixels), προσπαθεί να χρησιμοποιήσει περισσότερα από ένα bit σε ένα pixel με τέτοιο τρόπο ώστε η αλλαγή αυτή δεν θα επηρεάσει την οπτική εμφάνιση της εικόνας. Χρησιμοποιεί παράπλευρη πληροφορία από γειτονικά εικονοστοιχεία για την εκτίμηση του αριθμού των bits τα οποία μπορούν να ενσωματωθούν σε pixel της εικόνας προκειμένου να κρύψουν τα μυστικά δεδομένα. Η τεχνική αυτή ονομάζεται PBSM.

2.1.2.1 Αποστολή των Δεδομένων

Τα βήματα που ακολουθούνται για την αποστολή των δεδομένων είναι τα παρακάτω:

1. Μετατροπή της εικόνας σε δυαδική μορφή.
2. Διαίρεση του μυστικού μηνύματος σε τμήματα, κάθε τμήμα αποτελείται από 16 χαρακτήρες (128 bits).
3. Εφαρμογή της διαδικασίας κρυπτογράφησης για τη μετατροπή κάθε τμήματος απλού κειμένου σε ένα τμήμα κρυπτογραφημένο κειμένου.
4. Όλα τα κρυπτογραφημένα τμήματα κειμένου τίθενται σε μία σειρά για να σχηματίσουν το ενιαίο κρυπτογραφημένο κείμενο.
5. Το κρυπτογραφημένο κείμενο μετατρέπεται σε δυαδικό.
6. Το κρυπτογραφημένο κείμενο ενσωματώνεται στη δυαδική μορφή της εικόνας, σύμφωνα με τη διαδικασία ενσωμάτωσης που αναφέρθηκε

παραπάνω και έτσι προκύπτει αρχικά η στεγανογραφημένη (stego) δυαδική μορφή της εικόνας. Κατόπιν η στεγανογραφημένη (stego) δυαδική μορφή της εικόνας μετατρέπεται σε στεγανογραφημένη εικόνα για να αποσταλεί στον παραλήπτη.

2.1.2.2 Ενσωμάτωση Κρυπτογραφημένου Μηνύματος

Για την κρυπτογράφηση του μηνύματος χρησιμοποιούνται τα ακόλουθα.

Στοιχεία εισόδου:

- Το μυστικό μήνυμα M αποτελούμενο από bits μήκους l
- Μία ασυμπίεστη εικόνα IPE
- Τα κλειδιά κρυπτογράφησης και στεγανογράφησης Ps

Παράμετροι:

- το υψηλότερο επίπεδο bit (Imax)
- το μέγεθος m × n του συρόμενου παραθύρου

Τα βήματα που ακολουθούνται για την κρυπτογράφηση του μηνύματος είναι:

1. Μετατροπή των l bits σε l' bits1 σύμφωνα με τη παραπάνω διαδικασία
2. Διαίρεση της δυαδικής σειράς l' σε τμήματα (επίπεδα bit) μήκους N
3. Συμπίεση και κρυπτογράφηση μηνύματος με το κλειδί Ke
4. Αρχικοποίηση της γεννήτριας τυχαίων αριθμών με το κλειδί Ks
5. Για i από 1 έως Imax
    a. Εντοπίζονται όλες οι m × n επίπεδες περιοχές στο κάθε τμήμα (επίπεδο) Bi με κατώφλι t, σύμφωνα με το 4
    b. Το μήνυμα ενσωματώνεται στα bits των μη-επίπεδων επιφανειών και σε μία ψευδό-τυχαία ακολουθία
6. Αν κάποια bits του μηνύματος δεν έχουν ενσωματωθεί επιστρέφεται σφάλμα
7. Μετατροπή της δυαδικής σειράς l' σε στεγανογραφημένη δυαδική εικόνα σύμφωνα με το 2

Στοιχεία κατά την έξοδο:

- Κρυπτογραφημένη (στεγανογραφημένη) εικόνα

2.1.2.3 Λήψη της Εικόνας

Η λήψη της εικόνας στεγανογραφημένης εικόνας γίνεται ως εξής:

Στοιχεία εισόδου:

- μία στεγανογραφημένη εικόνα l' Pe
- τα κλειδιά κρυπτογράφησης και στεγανογράφησης Ps

Παράμετροι:

- το υψηλότερο επίπεδο bit Imax
- το μέγεθος m × n του συρόμενου παραθύρου

Τα βήματα που ακολουθούνται είναι:

1. Μετατροπή της εικόνας l’ σε δυαδική μορφή
2. Διαίρεση της δυαδικής μορφής l’ σε επίπεδα bit μήκους N
3. Αρχικοποίηση της γεννήτριας τυχαίων αριθμών με το κλειδί Ks
4. Για i από Imax έως 1
    a. Βρίσκουμε όλες τις επίπεδες περιοχές bit Bi μεγέθους m × n
    b. Εξαγωγή του μηνύματος M στις μη –επίπεδες περιοχές της σειρά Bi χρησιμοποιώντας τη ψευδο-τυχαία ακολουθία bits
5. Αποκρυπτογράφηση του μηνύματος M με το κλειδί Ke και αποσυμπίεση του

Στοιχεία κατά την έξοδο:

- το μυστικό μήνυμα M μήκους l

2.1.3 Αλγόριθμοι L2LSB-EDCT

Αρχικά προετοιμάζονται οι εικόνες κάλυψης και στεγανογράφησης με τη τεχνική DCT (Design of image steganography algorithms with secure message routing for p2p networks n.d.).

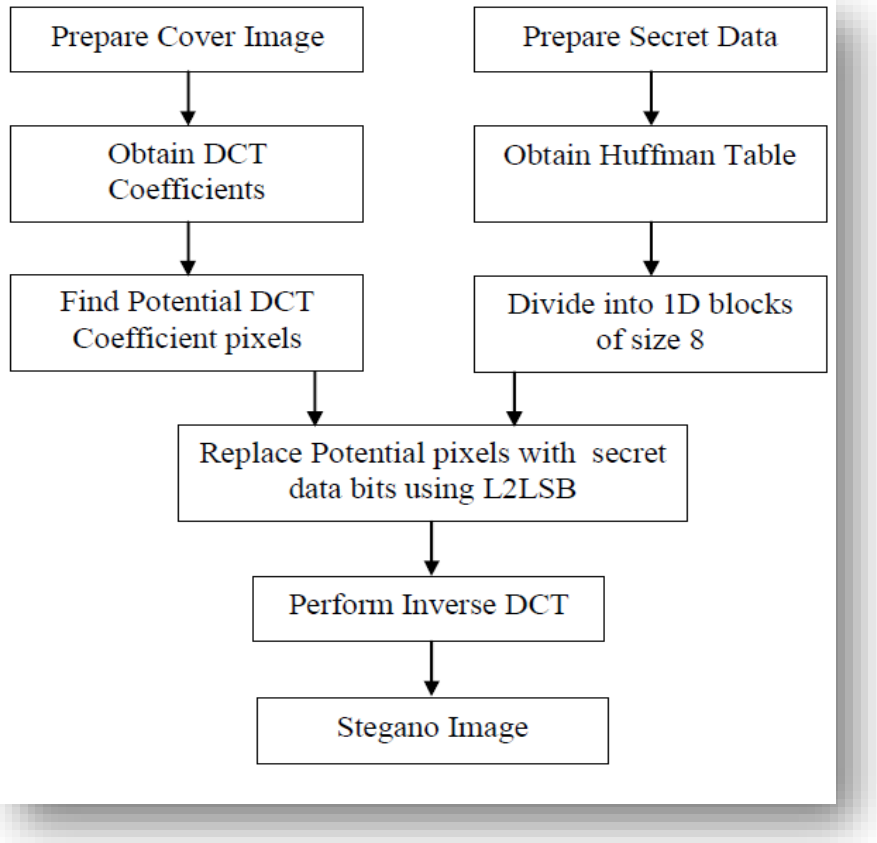
Η τεχνική DCT (Discrete Cosine Transformation) μεταφράζει το περιεχόμενο της εικόνας από τη χωροταξική της μορφή (spatial domain) σε μορφή συχνοτήτων (frequency domain) για να αναπαρασταθεί σε μια πιο συμπαγή μορφή. Οι στοχαστικές της ιδιότητες είναι παρόμοιες με τον μετασχηματισμό Fourier η οποία θεωρεί πλέον την εικόνα χρονικά αναλλοίωτη ή ως ένα σταθερό σήμα. Η τεχνική DCT είναι μία ειδική περίπτωση Διακριτού Μετασχηματισμού Fourier (DFT) στον οποίο τα ημίτονα συστατικά απαλείφθηκαν για να μείνουν μόνο τα συνημίτονα.

Ο διακριτός μετασχηματισμός συνημίτονου (DCT) εκφράζει μια πεπερασμένη ακολουθία από σημεία δεδομένων τα οποία ορίζονται ως το άθροισμα των συναρτήσεων συνημίτονου οι οποίες ταλαντώνονται σε διαφορετικές συχνότητες.

Κατόπιν η κωδικοποίηση Huffman εκτελείται πάνω από την εικόνα με το μυστικό μήνυμα πριν την ενσωμάτωση του μηνύματος και κάθε bit του Huffman κωδικού της εικόνας με το μυστικό μήνυμα ενσωματώνεται στο εσωτερικό της εικόνας-κάλυμμα μεταβάλλοντας το λιγότερο σημαντικό bit (LSB) που καθορίζει την ένταση των εικονοστοιχείων της εικόνας-κάλυψης. Το μέγεθος της κωδικοποιημένης ροής δυαδικών στοιχείων Huffman όπως και ο πίνακας

Huffman ενσωματώνονται επίσης στο εσωτερικό της εικόνας-κάλυψης, έτσι ώστε η στεγανογραφημένη εικόνα αποτελεί αυτόνομη πληροφορία για τον παραλήπτη.

Σε επόμενο βήμα εφαρμόζονται DCT συντελεστές. Στη περίπτωση αυτή μπορεί να χρησιμοποιηθούν τέσσερις τύποι DCT: 1D DCT, 2D DCT, 3D και 4D DCT. Όλοι οι τέσσερες τύποι DCT διαιρούν την εικόνα σε τμήματα 8 x 8, από τις οποίες οι συντελεστές DCT μπορούν να εκτιμηθούν. Κατόπιν εφαρμόζεται η διαδικασία του αντίστροφου DCT για να ληφθεί η στεγανογραφημένη εικόνα.



Εικόνα 4, Αλγόριθμος L2LSB-DCT

Ο γενικός τύπος του 1-DCT είναι:

$$C(u) = \propto (u) \sum_{i=0}^{N-1} f(x) cos \left[\frac{(2x+1)u\pi}{2N} \right]$$

Όπου u=0, 1, 2… N-1 και N=7

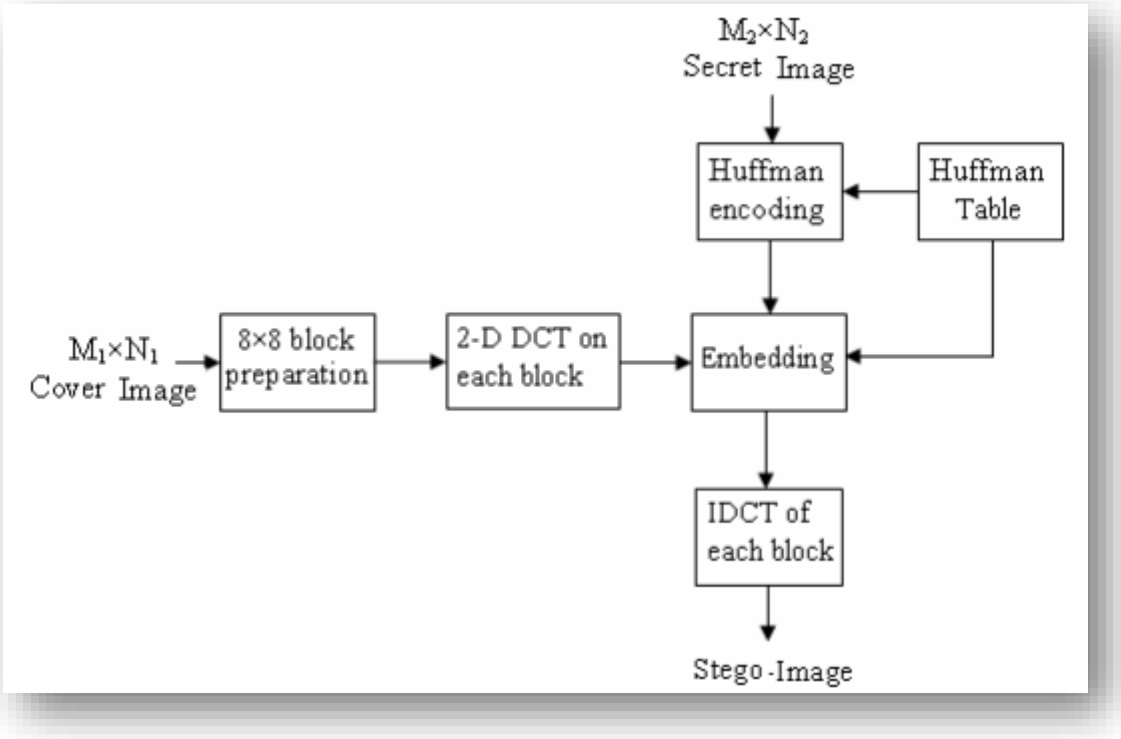
Ενώ ο τύπος του 2-DCT είναι:

$$C(u) = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(x,y)cos\left[\frac{(2x+1)u\pi}{2N}\right]cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Όπου u, v=0, 1, 2… N-1 και N=7

2.1.4 Αλγόριθμοι LSB, 3D-DCT - στεγανογραφία βασισμένη στην κωδικοποίηση Huffman

Στα δομημένα P2P συστήματα η παράδοση του μηνύματος μπορεί να γίνει αναγνωρίζοντας τα ID των κόμβων των ανεξάρτητων συστημάτων. Ο αποστολέας του μηνύματος πρέπει να αποφασίσει τον παραλήπτη του μηνύματος και μπορεί να το δρομολογήσει μέσω ενός ή περισσότερων hops. Το μήνυμα περνάει από το ένα hop στο άλλο σωστά, προσδιορίζοντας την IP διεύθυνση, και στο τέλος φτάνει στον προορισμό του. Μπορεί να εφαρμοσθεί μια αποτελεσματική στρατηγική δρομολόγησης για τον έλεγχο της διαδρομής δρομολόγησης και την αναγνώριση κακόβουλων κόμβων (Satyavathy and Punithavalli , 2011). Με αυτή την στρατηγική δρομολόγησης εξαλείφονται τα μειονεκτήματα της κρυπτογράφησης με την εφαρμογή στεγανογραφίας στην παράδοση του μηνύματος. Επίσης, μπορεί να εφαρμοσθεί ένα στεγανογραφικό σχήμα κωδικοποίησης το οποίο διαχωρίζει τα κανάλια χρώματος των windows εικόνων bitmap και το οποίο στην συνέχεια αποθηκεύει τυχαία μηνύματα στο LSB του ενός συστατικού χρώματος ενός επιλεγμένου pixel (Vazquez , 2000), όπου τα συστατικά χρώματος των άλλων δύο pixel βρέθηκαν να είναι ίσα με το επιλεγμένο κλειδί. Επιπλέον, εφαρμόζεται στεγανογραφία βασισμένη στον αλγόριθμο 3D-DCT η οποία ενσωματώνει το κείμενο του μηνύματος στο LSB του Discrete Cosine (DC) συντελεστή της ψηφιακής εικόνας. (Aggoun , 2006) Τότε η κωδικοποίηση Huffman εκτελείται επίσης στα μυστικά μηνύματα / εικόνες πριν την ενσωμάτωση. Κάθε bit του κώδικα Huffman του μυστικού μηνύματος / εικόνας ενσωματώνεται στο πεδίο των συχνοτήτων μεταβάλλοντας το LSB όλων των DCT συντελεστών του καλύμματος της εικόνας.



Εικόνα 5, Αλγόριθμος DCT βασισμένος στην κωδικοποίηση Huffman

Τα αποτελέσματα των πειραμάτων δείχνουν ότι ο αλγόριθμος επιτυγχάνει υψηλή παραγωγικότητα και μεγάλη κάλυψη (A.Nag, et al. , 2010). Επιπλέον, το PSNR του καλύμματος της εικόνας, στην οποία έχει εφαρμοσθεί στεγανογραφία, επιτυγχάνει καλύτερα αποτελέσματα σε σχέση με άλλες υπάρχουσες στεγανογραφικές προσεγγίσεις. Η ασφάλεια διατηρείται σε ικανοποιητικά επίπεδα δεδομένου ότι το μυστικό μήνυμα / εικόνα δεν μπορεί να εξαχθεί χωρίς την γνώση των κανόνων αποκωδικοποίησης και του πίνακα Huffman.

2.1.5 Αλγόριθμος Στεγανογραφίας βασισμένος στη τεχνική Wavelet Transform (DWT)

Η τεχνική Wavelet Transform (DWT) χρησιμοποιεί wavelets τα οποία είναι ειδικές λειτουργίες για τον ορισμό και επεξεργασία σημάτων. Οι DWT συντελεστές του μετασχηματισμένου καλύμματος της εικόνας θα ποσοτικοποϊηθούν και στην συνέχεια θα τροποποιηθούν σύμφωνα με τα μυστικά δεδομένα (Rani Lakshmi and Pradeepa , 2014). Μία από τις εξειδικεύσεις της τεχνικής είναι η Haar-DWT, η απλούστερη μορφή της τεχνικής DWT.

Οι τιμές της έντασης σε έναν δυσδιάστατο πίνακα μια εικόνας απεικονίζονται από n bits, το οποίο καλείται ως το βάθος bit της εικόνας. Το βάθος bit μπορεί να κυμαίνεται από 2 έως 32 και εξαρτάται από την υποστήριξη που προσφέρει το υλικό και το λογισμικό. Μεταξύ των διαφορετικών πλεονασμών η κωδικοποίηση

του πλεονασμού είναι εκείνη που μπορεί να χρησιμοποιηθεί με βεβαιότητα για την επίτευξη συμπίεσης χωρίς σφάλματα ή απώλειες. (Thanikaiselvan, et al. , 2013)Η κωδικοποίηση Huffman είναι μια τεχνική μεταβλητού μεγέθους χωρίς απώλειες, η οποία μπορεί να εφαρμοσθεί σε οποιαδήποτε οντότητα που μπορεί να αναπαρασταθεί ψηφιακά. Αρκετές τεχνικές κωδικοποίησης μπορούν να χρησιμοποιηθούν σε συνδυασμό με την κωδικοποίηση Huffman για να συμπιεστεί το ωφέλιμο φορτίο. Η ενσωμάτωση όλων των παραπάνω σε ένα ενιαίο σκέλος καταλήγει σε υψηλότερα αποτελέσματα συμπίεσης στις χωρητικότητες μεγάλης ενσωμάτωσης ενός αλγορίθμου στενογραφίας.

Δημιουργήθηκε ένα καινούργιο στεγανογραφικό σχήμα κωδικοποίησης που διαχωρίζει τα κανάλια χρώματος των εικόνων bitmap των Windows και στην συνέχεια αποκρύπτει με τυχαίο τρόπο μηνύματα μέσα στο LSB του ενός συστατικού από το επιλεγμένο pixel, όπου οι χρωματικοί συντελεστές των άλλων δύο βρέθηκαν να είναι ίσοι με τα επιλεγμένα κλειδιά. Επιπλέον, εισήχθη και στεγανογραφία βασισμένη στο DWT που ενσωματώνει το μήνυμα του κειμένου στο LSB των DC συντελεστών. Υλοποιήθηκαν στεγανογραφία βασισμένη στο LSB, στεγανογραφία βασισμένη στο 2D-DWT (Panigrahi and Reddy , 2014) σε συνδυασμό με την κωδικοποίηση Huffman και υπολογίσθηκε ο δείκτης PSNR. Ο PSNR είναι το σήμα κορυφής προς θόρυβο, σε ντεσιμπέλ, μεταξύ δύο εικόνων. Αυτή η αναλογία χρησιμοποιείται ως μέτρηση της ποιότητας μεταξύ των δύο εικόνων. Αν ο λόγος PSNR είναι υψηλός τότε οι εικόνες είναι καλύτερης ποιότητας.

2.2 Σύγκριση Αλγορίθμων

Στη συνέχεια παρατίθεται πίνακας με τα κυριότερα σημεία σύγκρισης των σημαντικότερων αλγορίθμων που συναντήσαμε στη βιβλιογραφία καθώς και πλεονεκτήματα/ μειονεκτήματα (Hussain , 2013) που καταγράφηκαν ως αποτέλεσμα μελέτης.

Τεχνική / Αλγόριθμος	Αποδοτικότητα	Ανθεκτικότητα	Ταχύτητα	Τυχαιότητα	Φορητότητα	Θετικά	Αρνητικά
L2LSB (MobiStego)	Y		Y		Y	• Εύκολα υλοποιήσιμος, • Ταχύτητα εκτέλεσης.	• Ευάλωτος σε επιθέσεις, εύκολα διακριτός.
LSB-PBSM		Y				• Χρήση παράπλευρων	• Υπολογιστικοί πόροι κατά την λειτουργία.

						bits για αύξηση ασφάλειας.	• Ταχύτητα εκτέλεσης.
L2LSB-EDCT		Υ	Υ	Υ		• Υψηλή ποιότητα στεγανογρα-φημένης εικόνας.	• Χρήση σε αρχεία με lossy data compression.
LSB 3D-DCT		Υ	Υ	Υ		• Υψηλή ποιότητα στεγανογρα-φημένης εικόνας. • Αυξημένη ασφάλεια.	• Αυξημένη πολυπλοκότητα υλοποίησης. • Χρήση σε αρχεία με lossy data compression
LSB HCM	Υ			Υ		• Τυχαιότητα σημείων στεγανο-γραφίας.	• Χαμηλή ασφάλεια. • Χρήση γεννήτριας συχνοτήτων στις δυο πλευρές επικοινωνίας.
DWT	Υ	Υ		Υ		• Πολύ υψηλή ποιότητα στεγανογρα-φημένης εικόνας. • Αυξημένη ασφάλεια	• Αυξημένη πολυπλοκότητα υλοποίησης. • Υπολογιστικοί πόροι κατά την λειτουργία.

Πίνακας 1, Πίνακας Σύγκρισης Αλγορίθμων

Όπως προκύπτει από τη μελέτη της βιβλιογραφίας οι παραπάνω ευρέως χρησιμοποιούμενοι αλγόριθμοι L2LSB, L2LSB-EDCT, LSB 3D-DCT, DWT κατορθώνουν τη διατήρηση της υψηλής ποιότητας της εικόνας μετά τη στεγανογράφησή της, ενώ στον LSB HCM εισάγεται ο παράγοντας τυχαιότητας στην επιλογή σημείων στεγανογραφίας. Μεγάλη σημασία έχει ο βαθμός ασφάλειας κάθε αλγορίθμου με πιο ασφαλής τους LSB-PBSM, LSB 3D-DCT, DWT. Κάθε αλγόριθμος ανάλογα με την πολυπλοκότητα υλοποίησής του απαιτεί ανάλογους υπολογιστικούς πόρους, στοιχείο που αποτελεί σημείο έρευνας και βελτιστοποίησης.

2.3 Εφαρμοσμένη Στεγανογραφία

Η αξιοποίηση των αλγορίθμων στεγανογραφίας στην πράξη εισάγει ολοένα και περισσότερα σημεία ελέγχου και βελτίωσης των στεγανογραφικών μεθόδων. Τέτοια σημεία ελέγχου και βελτιστοποίησης απαιτεί η αποδοτικότητα που στηρίζεται στους απαιτούμενους και διαθέσιμους υπολογιστικούς πόρους, αλλά και η ασφάλεια που επιτυγχάνεται αξιοποιώντας και τα εξωγενή συνεργαζόμενα συστήματα, όπως το υπολογιστικό νέφος (cloud) ή οι σχέσεις που αναπτύσσονται μεταξύ peer-to-peer εφαρμογών και κοινωνικών δικτύων (social media)/ συνεργατικότητας (collaboration). Στη συνέχεια παρατίθενται ενδεικτικές περιπτώσεις πρακτικών εφαρμογών στεγανογραφίας.

2.3.1 Μεθοδολογία Στεγανογραφίας με χρήση ενός επιπλέον επίπεδου ασφάλειας στο Cloud

Ο συγκεκριμένη μεθοδολογία βασίζεται σε αλγόριθμο, ο οποίος διαβάζει τα δεδομένα από το αρχείο-πηγή εισόδου και τα μετατρέπει σε ASCII-BCD βάση δεδομένων με τη χρήση κλειδιών ασφαλείας. Η ενότητα με τη διαδικασία της μετατροπής παράγει το κρυπτογραφημένο κείμενο και στη συνέχεια, το μερικώς κρυπτογραφημένο κείμενο μετατρέπεται σε ένα πλήρες κρυπτογραφημένο κείμενο χρησιμοποιώντας ένα άλλο κλειδί. Ένα πλήρες κρυπτογραφημένο κείμενο αποστέλλεται στο υπολογιστικό σύννεφο για αποθήκευση χρησιμοποιώντας την ενότητα κρυπτογράφησης. Αν ένας χρήστης πρέπει να αντλήσει τα δεδομένα από το υπολογιστικό σύννεφο, πρώτα διαβάζει μια εικόνα, προσδιορίζει τη θέση στην εικόνα και στη συνέχεια μετατρέπει την εικόνα σε μορφή BCD (C.Saravanakumar and C.Arun , 2014).

Το μερικώς κρυπτογραφημένο κείμενο στη συνέχεια μετατρέπεται σε ένα ισοδύναμο ASCII κείμενο χρησιμοποιώντας το κλειδί αποκρυπτογράφησης (Shrote and Chouragade , 2015). Το πρωτότυπο κείμενο ανακτάται με τη χρήση άλλου κλειδιού με τη διαδικασία αποκρυπτογράφησης.

2.3.1.1 Διαδικασία Κρυπτογράφησης

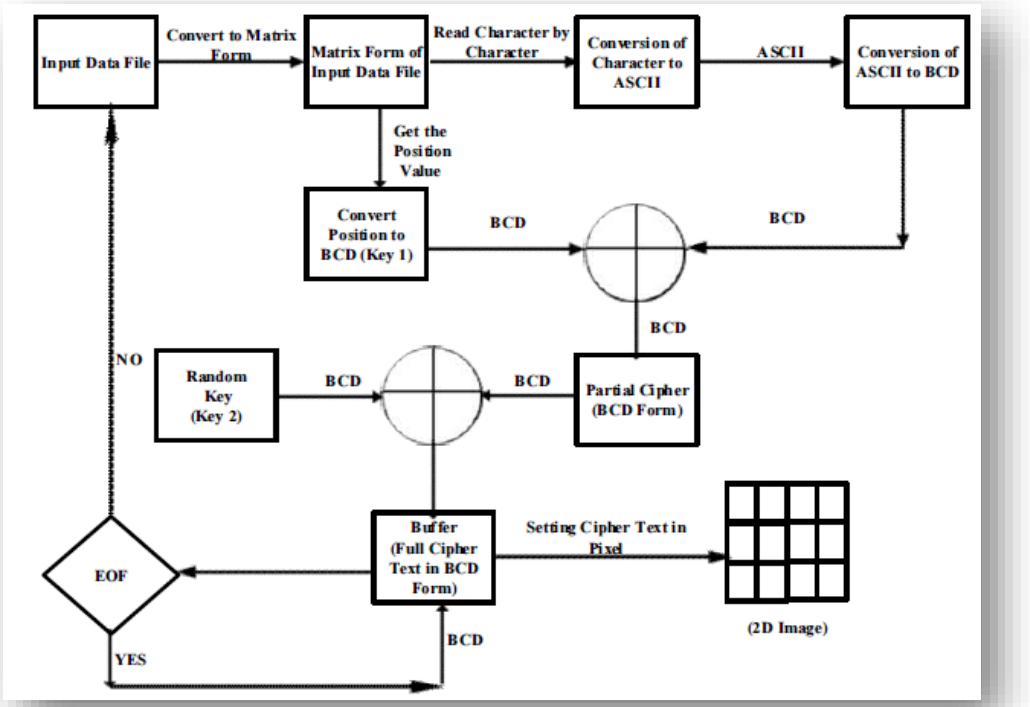
Η διαδικασία κρυπτογράφησης χρησιμοποιεί στεγανογραφία βασισμένη σε εικόνες για την ασφαλή διακίνηση των δεδομένων των χρηστών στο υπολογιστικό σύννεφο. Η δισδιάστατη εικόνα χρησιμοποιείται για να κρύψει τις πληροφορίες και σε οποιαδήποτε άλλη μορφή αρχείου εικόνας. Τα δεδομένα των χρηστών τα οποία πρόκειται να διακινηθούν αποθηκεύονται σε ένα κείμενο ASCII. Αυτό το αρχείο δεδομένων μετατρέπονται σε κωδικοποιημένους χαρακτήρες με τη χρήση ειδικού πίνακα όπου η κατάλληλη θέση στον πίνακα καθορίζει ποιος κωδικοποιημένος χαρακτήρας θα εκχωρηθεί σε ποιο ASCII χαρακτήρα. Ο κωδικός BCD δημιουργείται για κάθε ASCII χαρακτήρα χρησιμοποιώντας κωδικοποίηση της τάξης των 16-bit. Η θέση του χαρακτήρα

καθορίζεται από τον πίνακα και κατόπιν η θέση αυτή μετατρέπεται σε 16-bit BCD κωδικό.

Εφαρμόζοντας τη συνάρτηση XOR στον κωδικό BCD της θέσης και στα δεδομένα προκύπτει η τελική 16-bit BCD τιμή. Η τιμή που προκύπτει είναι το μερικώς κρυπτογραφημένο κείμενο. Επίσης εφαρμόζεται η συνάρτηση XOR ανάμεσα στο ιδιωτικό κλειδί XORed και το μερικώς κρυπτογραφημένο κείμενο παρέχοντας ένα πλήρες κρυπτογραφημένο κείμενο το οποίο θα σταλεί στο υπολογιστικό σύννεφο από τον χρήστη. Ο χρήστης θα να είναι σε θέση να ανακτήσει τις ίδιες πληροφορίες με τη χρήση ιδιωτικών κλειδιών. Ένα πλήρες κρυπτογραφημένο κείμενο είναι επίσης στη μορφή BCD.

Η θέση του κρυπτογραφημένου κειμένου στην εικόνα σχετίζεται με την θέση των εικονοστοιχείων. Η κάθε θέση pixel χαρακτηρίζεται επιπλέον από τις BCD τιμές που έχουν υπολογιστεί παραπάνω και αποθηκεύονται σε μορφή πίνακα (σειρά, στήλη): το πρώτο μέρος καθορίζει τη σειρά του pixel ενώ το δεύτερο μέρος καθορίζει τη στήλη του pixel. Η διεύθυνση και η τιμή του πρώτου κρυπτογραφημένου κειμένου στην εικόνα προσδιορίζεται από άλλο ιδιωτικό κλειδί. Το πρώτο κρυπτογραφημένο κείμενο τοποθετείται στην εικόνα με σχετικό κωδικό θέσης δηλαδή τη τιμή του κλειδιού.

Αυτή η διαδικασία επαναλαμβάνεται μέχρις ότου όλα τα μέρη του κρυπτογραφημένου κειμένου τοποθετηθούν στην εικόνα. Αυτή η εικόνα κατέχει το σύνολο του κρυπτογραφημένου κειμένου σε διάφορες θέσεις pixel χωρίς να αλλοιωθεί η εμφάνιση και ποιότητα και χρησιμοποιείται για να μπορεί ο χρήστης να διατηρεί και να ανταλλάσσει πληροφορίες με τον πάροχο υπηρεσιών υπολογιστικού νέφους και ως ένα άλλο επίπεδο ασφαλείας.



Εικόνα 6, Διαδικασία Κρυπτογράφησης

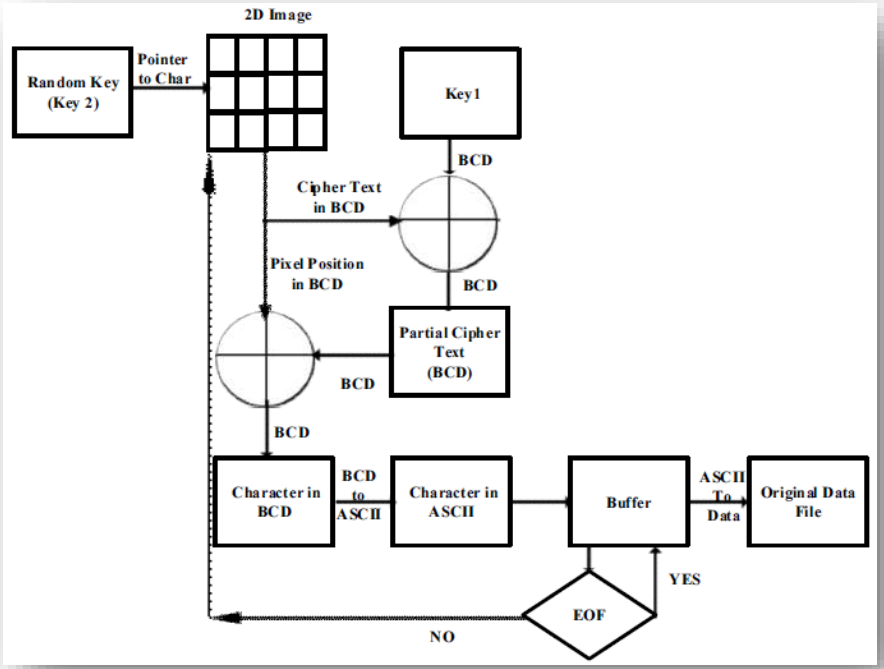
2.3.1.2 Διαδικασία Αποκρυπτογράφησης

Ο χρήστης θα πρέπει να ανακτήσει τις πληροφορίες που είναι ήδη αποθηκευμένες στο σύννεφο ως μια εικόνα. Ο πάροχος επαληθεύει την απαίτηση του πελάτη, επιλέγει την εικόνα και τη στέλνει στον χρήστη. Η λαμβανόμενη εικόνα αποκρυπτογραφείται χρησιμοποιώντας το κλειδί που έχει ήδη χρησιμοποιηθεί για τη διαδικασία κρυπτογράφησης.

Το πρώτο κλειδί είναι ένα κλειδί που χρησιμοποιείται τόσο για τη διαδικασία κρυπτογράφησης όσο και της αποκρυπτογράφησης, ενώ ένα άλλο κλειδί χρησιμοποιείται μόνο για τη διαδικασία αποκρυπτογράφησης. Το δεύτερο κλειδί χρησιμοποιείται για να προσδιορίσει τη θέση του pixel και επίσης να εντοπίσει το πρώτο χαρακτήρα κρυπτογραφημένου κειμένου στην εικόνα. Μόλις εντοπιστεί η θέση του πρώτου χαρακτήρα του κρυπτογραφημένου κειμένου θα μετατραπεί σε BCD κωδικό. Το δεύτερο κρυπτογραφημένο κείμενο ταυτοποιείται με βάση το πρώτο κρυπτογραφημένο κείμενο και ούτω καθεξής.

Το μερικώς κρυπτογραφημένο κείμενο παράγεται εφαρμόζοντας τη συνάρτηση XOR ανάμεσα στο BCD κωδικό και ένα άλλο κλειδί. Η τιμή BCD του μερικώς κρυπτογραφημένου κειμένου μετατρέπεται σε ASCII χαρακτήρα. Αυτή η διαδικασία συνεχίζεται έως ότου δεν υφίσταται πλέον θέση. Το αρχικό κείμενο ανακτάται με την ένωση όλων των επιμέρους κρυπτογραφημένων κειμένων σε

ένα ολοκληρωμένο κρυπτογραφημένο κείμενο το οποίο είναι διαθέσιμο στην εικόνα.



Εικόνα 7, Διαδικασία Αποκρυπτογράφησης

Η παραπάνω μεθοδολογία εφαρμόζεται ως ένα επιπλέον επίπεδο ασφάλειας στο υπολογιστικό νέφος, ωστόσο εισάγει το κόστος της χρησιμοποίησης δύο κλειδιών τόσο για την κρυπτογράφηση όσο και την αποκρυπτογράφηση.

2.3.2 Μηχανισμοί Ασφάλειας με χρήση στεγανογραφίας σε Peer-to-Peer εφαρμογές

Σήμερα χρησιμοποιούνται πολλά διαφορετικά είδη peer-to-peer εφαρμογών (Parashar, et al. , 2001). Μερικές επιτρέπουν την επικοινωνία μεταξύ ατόμων, όπως βίντεο και μηνύματα κειμένου, ενώ άλλες παρέχουν δυνατότητες κοινής χρήσης δεδομένων. Επίσης, μερικές λειτουργούν και σαν ένας μεγάλος εικονικός υπολογιστής, παρέχοντας κατανεμημένες υπολογιστικές υπηρεσίες σε μία κεντρική πηγή. Μια σημαντική δυσκολία στην ανάπτυξη των συστημάτων αυτών είναι η επιλογή του κατάλληλου μηχανισμού ασφαλείας από το ευρύ φάσμα των διαθέσιμων τεχνολογιών.

Στις αποκεντρωμένες εφαρμογές, όπως είναι οι peer-to-peer εφαρμογές, η ασφάλεια είναι κάτι πολύ περισσότερο από μία απλή ταυτοποίηση του χρήστη. Πρέπει να παρέχει υπηρεσίες ελέγχου προέλευσης και ακεραιότητας δεδομένων, καθώς και την διαχείριση των δικαιωμάτων πνευματικής ιδιοκτησίας. Μία από

τις κύριες δυσκολίες στο σχεδιασμό της ασφάλειας για peer-to-peer εφαρμογές είναι η απόφαση για τις τεχνολογίες ασφάλειας που ταιριάζουν καλύτερα στην υπό-ανάπτυξη εφαρμογή. Οι peer-to-peer εφαρμογές μπορούν να ομαδοποιηθούν στις ακόλουθες τρεις κατηγορίες: κατανεμημένη χρήση αρχείων, επικοινωνίες σε πραγματικό χρόνο και κατανεμημένα υπολογιστικά συστήματα. Οι κύριες τεχνολογίες ασφάλειας που χρησιμοποιούνται είναι η κρυπτογραφία, οι έξυπνες κάρτες και η στεγανογραφία.

2.3.3 Κρυμμένη Επικοινωνία στα P2P Δίκτυα: Στεγανογραφική Χειραψία και Αναμετάδοση

Το καίριο ερώτημα είναι με ποιο τρόπο σε μια ομάδα συνωμοτών κόμβων σε ένα p2p δίκτυο θα μπορέσει να βρει ο ένας τον άλλο και να επικοινωνήσει χωρίς να προκαλέσει υποψίες μεταξύ των τακτικών κόμβων ή οποιασδήποτε άλλης αρχής που ελέγχει το δίκτυο. Ουσιαστικά το πρόβλημα ενάγεται στο πως ένας συνωμότης κόμβος θα μπορέσει να μεταδώσει κρυφά ένα μήνυμα σε όλους τους άλλους συνωμότες κόμβους. Ένα υπό-πρόβλημα το οποίο εξετάζεται ξεχωριστά είναι το πώς ένας συνωμότης θα μπορέσει να εξακριβώσει με ασφάλεια το τύπο ενός συνδεδεμένου κόμβου, π.χ. να μάθει εάν ο συνδεδεμένος κόμβος είναι ένας συνωμότης ή ένας κανονικός κόμβος χωρίς όμως να προδώσει ο ίδιος τον τύπου του (ότι είναι συνωμότης). Όταν υπάρχουν πολλά επίπεδα παρακολούθησης μπορούν να χρησιμοποιηθούν κατανεμημένοι και αποδοτικοί αλγόριθμοι που μπορούν να μεταδώσουν τις κρυμμένες πληροφορίες μεταβάλλοντας κατάλληλα την ακολουθία του μπλοκ της αίτησης (Raphael, Thomas and Roger , 2011). Το p2p πρωτόκολλο προσφέρει πολλούς στεγανογραφικούς τύπους μέσω των οποίων μπορούν να μεταδοθούν οι κρυφές πληροφορίες. Τα p2p δίκτυα είναι επιρρεπή σε κρυμμένη πληροφορία ακόμα και αν παρακολουθούνται πλήρως.

Με βάση τα παραπάνω και την βιβλιογραφική μελέτη που προηγήθηκε, προκύπτει πως η στεγανογραφία στη σύγχρονη ψηφιακή εποχή βρίσκει εκτεταμένη εφαρμογή σε πλήθος εφαρμογών, τόσο αναφορικά με την ενίσχυση της ασφάλεια των εφαρμογών και των δεδομένων που αυτές μεταχειρίζονται, όσο και σε εφαρμογές όπου η στεγανογραφία αξιοποιείται για την προστασία δικαιωμάτων (πνευματικά δικαιώματα) (Maganbhai and Chouhan , 2015), (Judge , 2001). Όπως αναφέρεται και παραπάνω (Ενότητα 1), οι τομείς όπου η στεγανογραφία εφαρμόζεται πλέον είναι:

(α) Μυστικές επικοινωνίες (Zhang, Wu και Zhou , 2009).
(β) Tagging χαρακτηριστικών στοιχείων/ μεταδεδομένων, όπως στα κοινωνικά δίκτυα ή έξυπνες συσκευές.
(γ) Μηχανισμοί προστασίας πνευματικών δικαιωμάτων/ προστασίας αντιγραφής.

3 Προτεινόμενη Μεθοδολογία

Στην παρούσα ενότητα παρουσιάζεται η προτεινόμενη μεθοδολογία για την υλοποίηση στεγανογραφίας και τη μετάδοση κρυφών μηνυμάτων μέσω εφαρμογών Android, καθώς και η προτεινόμενη αρχιτεκτονική και τα συστατικά της μέρη.

3.1 Ανάλυση Απαιτήσεων

Με βάση τους ευρέως γνωστούς αλγορίθμους που παρουσιάστηκαν στην ενότητα 2 διαπιστώνουμε τα ακόλουθα:

1. Όλες οι μέθοδοι (βλ. 2.2) εμφανίζουν κλιμακούμενη πολυπλοκότητα με τις περισσότερες να εμφανίζουν μεγάλη πολυπλοκότητα επενδύοντας στην ποιότητα και την ασφάλεια (LSB-PBSM, LSB 3D-DCT, DWT) με αποτέλεσμα να απαιτούν αυξημένους υπολογιστικούς πόρους, στοιχείο που απαιτεί εκτέλεση τους σε κεντρικούς υπολογιστές με πολλούς διαθέσιμους πόρους. Συνεπώς, μία πρόκληση από μόνη της είναι η υλοποίηση αλγορίθμου που εκτελείται σε κινητά τηλέφωνα π.χ. Android.
2. Ελάχιστες από τις παραπάνω μεθόδους λαμβάνουν υπόψη την τυχαιότητα(βλ. 2.2). Όπως προκύπτει και από τη μελέτη της βιβλιογραφίας (βλ. ενότητα 2) μόνο στον LSB HCM εισάγεται ο παράγοντας τυχαιότητας στην επιλογή σημείων στεγανογραφίας.
3. Καμία μέθοδος δεν λαμβάνει υπόψη, με στόχο την εγγύηση υψηλού επιπέδου ασφάλειας τον τρόπο διαμοιρασμού των κλειδιών και τη συνεργατικότητα (collaboration) μεταξύ των χρηστών.

Με βάση τις ανωτέρω διαπιστώσεις, πρωταρχικός σκοπός της μεταπτυχιακής διατριβής είναι η ανάπτυξη μίας τεχνικής στεγανογραφίας με δυνατότητα αξιοποίησης σε Android εφαρμογή, μιας και η μετάδοση δεδομένων μέσω του δικτύου κινητής τηλεφωνίας (τρίτης, τέταρτης γενιάς, κ.ο.κ) ή ασύρματων δικτύων (IEEE 802.11) αποτελεί ραγδαία αναπτυσσόμενη τεχνολογία, αλλά και τρόπο ζωής της σημερινής κοινωνίας. Πρόκληση στην προσπάθεια μας αποτελεί το γεγονός ότι οι συσκευές ασύρματης επικοινωνίας (κινητά τηλέφωνα) δεν διαθέτουν τους υπολογιστικούς πόρους (επεξεργαστική ισχύ, μνήμη, ταχύτητα μετάδοσης) που έχουν οι προσωπικοί ή κεντρικοί υπολογιστές στους οποίους αξιοποιούνται, επί το πλείστο, σήμερα οι τεχνικές στεγανογραφίας (Ενότητα 2). Επιπρόσθετα, υπόψη λήφθηκε το γεγονός ότι το λειτουργικό σύστημα των κινητών συσκευών (πχ. Android, iOS) θέτει περιορισμούς ως προς τη χρήση συγκεκριμένων τύπων αρχείων για την ενσωμάτωση κρυφών δεδομένων.

Με βάση τα παραπάνω, στόχος μας τέθηκε η υλοποίηση ενός συστήματος στεγανογραφίας που αξιοποιεί την τυχαιότητα και την κρυπτογράφηση και ο οποίος να μπορεί να εκτελείται σε Android συσκευές χωρίς υψηλές απαιτήσεις σε πόρους συστήματος (υλικό, χωρητικότητα). Ο προτεινόμενος αλγόριθμος

εφαρμόζεται σε εικόνες τύπου PNG αξιοποιώντας τη μη απωλεστική μέθοδο συμπίεσης του τύπου αυτού.

Επιπλέον, στα πλαίσια της ανάλυσης των προδιαγραφών της εφαρμογής που θα αποτελέσει απόδειξη εφικτότητας (proof of concept) της προτεινόμενης μεθοδολογίας μας, η υπό υλοποίηση εφαρμογή έχει τις παρακάτω δυνατότητες/ χαρακτηριστικά:

- Ταυτοποίηση (login) χρήστη
- Λήψη φωτογραφίας και ενσωμάτωση μηνύματος στην ληφθείσα εικόνα (στεγανογραφία)
- Αποστολή σε μία επαφή (φίλο)

Η αντίστοιχη Android εφαρμογή του παραλήπτη λαμβάνει τη στεγανογραφημένη φωτογραφία και μπορεί να αποκωδικοποιήσει τις θέσεις στις οποίες έχει αποθηκευτεί το κρυμμένο μήνυμα προκειμένου να το συνθέσει και να το προβάλει στον χρήστη.

Η προτεινόμενος αλγόριθμος καλύπτει τις βασικές αρχές κρυπτογραφίας (Panasenko and Smagin , 2011) και συγκεκριμένα:

- Εμπιστευτικότητα αλγορίθμου (Confidentiality): Για την διαδικασία αποστολής της στεγανογραφημένης εικόνας, αν δεν υπάρχει υφιστάμενη φιλία (relationship) ανάμεσα στα άτομα επικοινωνίας η εικόνα δεν είναι δυνατόν να αποκρυπτογραφηθεί, συνθήκη που προσδίδει στον προτεινόμενο αλγόριθμο την αρχή της εμπιστευτικότητας.
- Ακεραιότητα (Integrity): Η διαδικασία αποκρυπτογράφησης μεσολαβεί για τον προσδιορισμό των τυχαίων LSB στοιχείων που έχουν χρησιμοποιηθεί στην στεγανογραφημένη εικόνα. Το κλειδί που ανακτάται από την βάση δεδομένων διασφαλίζει την έγκυρη και αναλλοίωτη αναδιάρθρωση του κρυφού μηνύματος.
- Αυθεντικότητα (Authenticity): Η λειτουργικότητα της εφαρμογής που υλοποιήθηκε τηρεί την διαδικασία κρυπτογράφησης σαν ένα ενδιάμεσο βήμα πριν την τελική αποστολή της στεγανογραφημένης εικόνας. Το ενδιάμεσο αυτό βήμα εκτελεί έναν αυστηρό έλεγχο ανάμεσα στους λογαριασμούς που προσπαθούν να επικοινωνήσουν και μόνο αν η συνθήκη φιλίας (relationship) ικανοποιηθεί προβαίνει σε μοναδική αποστολή. Η αυθεντικότητα του μηνύματος που αποστέλλεται ανάμεσα στους δυο χρήστες είναι κάτι που διασφαλίζεται από το κλειδί κρυπτογράφησης που βρίσκεται στο κεντροποιημένο σύστημα διαχείρισης αποτελούμενο από την JEE εφαρμογή και την βάση δεδομένων.
- Μη άρνηση ενεργειών (Non-repudiation): Η συνδυαστική υλοποίηση στεγαρνογραφίας και κρυπτογραφίας στον προτεινόμενο αλγόριθμο

υποστηρίζεται από δυο βασικές δομές που καταγράφουν τις μοναδικές κινήσεις αποστολής και λήψης της κάθε μιας στεγανογραφημένης εικόνας. Στη προτεινόμενη διαδικασία μεσολαβεί ένα πλήρες αρχείο καταγραφής μέσω του εξυπηρετητή εφαρμογών (Jboss) αλλά και στην βάση δεδομένων, καθιστώντας αδύνατη την άρνηση του αποστολέα για την δημιουργία και αποστολή του στεγανογραφημένου μηνύματος.

Συνοψίζοντας, οι προδιαγραφές για τον σχεδιασμό της παραπάνω μεθόδου στεγανογραφίας η οποία θα ενσωματώνεται σε μία Android εφαρμογή είναι:

- Η ενσωμάτωση κρυφών μηνυμάτων σε εικόνες χωρίς αυτά να γίνονται αντιληπτά και χωρίς να αλλοιώνεται η εικόνα.
- Ο αλγόριθμος στεγανογραφίας που θα υλοποιηθεί θα πρέπει να επιλέγει τις θέσεις αποθήκευσης του μηνύματος με τυχαίο τρόπο.
- Οι θέσεις αποθήκευσης θα πρέπει να κρυπτογραφούνται με κλειδί το οποίο θα είναι μοναδικό για κάθε ζεύγος αποστολέα-παραλήπτη.
- Η λήψη φωτογραφίας, η ενσωμάτωση μηνύματος (στεγανογραφία) και η αποστολή του μηνύματος θα πρέπει να είναι χωρίς καθυστέρηση.
- Αντίστοιχα, η λήψη εικόνων και η παρουσίαση μηνυμάτων και εικόνων στο χρήστη θα πρέπει να εκτελείται επίσης χωρίς καθυστέρηση.

Στο σημείο αυτό αναφορικά με την τεχνολογία υλοποίησης, επισημαίνεται ότι η δυνατότητα αποστολής και λήψης φωτογραφιών σε Android συσκευές, μας οδήγησε την επιλογή socket programming, δηλαδή την αντιστοίχιση με εναλλαγές των ρόλων «εξυπηρετητή» και «πελάτη» ανάμεσα σε δύο συσκευές που επικοινωνούν. Την πολυπλοκότητα της υλοποίησης μας αυξάνουν η τυχαιότητα και η κρυπτογράφηση που αποτελούν βασικές προδιαγραφές της πρότασής μας, όπως περιγράφεται στη συνέχεια.

3.2 Προτεινόμενος Αλγόριθμος Στεγανογραφίας

Το μήνυμα εισάγεται σε τυχαίες θέσεις μέσα στην εικόνα. Οι τυχαίες θέσεις του μηνύματος μέσα στην εικόνα ενσωματώνονται (με στεγανογραφία) από το πρώτο pixel της εικόνας και μετά. Αυτό γίνεται για να γνωρίζει ο παραλήπτης ποιες θέσεις της εικόνας πρέπει να διαβάσει για να μπορέσει να πάρει όλο το μήνυμα. Επίσης, για λόγους μεγαλύτερης ασφάλειας τόσο το μήνυμα όσο και οι θέσεις που αυτό βρίσκεται μέσα στην εικόνα είναι κωδικοποιημένες με ένα κλειδί που το γνωρίζουν αποκλειστικά και μόνο ο αποστολέας και ο παραλήπτης. Το κλειδί βρίσκεται σε μία βάση δεδομένων και είναι μοναδικό για κάθε ζεύγος αποστολέα και παραλήπτη. Το μήνυμα και οι θέσεις που αυτό βρίσκεται μέσα στην εικόνα ενσωματώνονται σε αυτήν (με στεγανογραφία) με τέτοιο τρόπο ώστε να μην φαίνονται ορατές αλλοιώσεις.

Στις εφαρμογές που τρέχουν σε Android υπάρχουν οι ακόλουθοι περιορισμοί που λάβαμε υπόψη μας κατά τον σχεδιασμό της αρχιτεκτονικής της λύσης μας:

1. Στο Android δεν χρησιμοποιούνται οι ίδιες βιβλιοθήκες για επεξεργασία και διαχείριση εικόνων με την Java (Grønli, Hansen and Ghinea , 2010) . Συνεπώς, δεν μπορούμε να επαναχρησιμοποιήσουμε λογικές, αρχιτεκτονικές και υλοποιήσεις στεγανογραφίας που έχουν γίνει με επιτυχία σε Java εφαρμογές.
2. Όταν μία εικόνα αποθηκεύεται σε JPEG μορφή υπάρχει αλλοίωσή της (εξ' ορισμού (Taylor , 2016) (Chen and Huang , 2011)). Αυτό έχει σαν αποτέλεσμα το μήνυμα που έχει ενσωματωθεί σε αυτήν (με στεγανογραφία) να χάνεται ή να αλλοιώνεται. Γι' αυτό το λόγο θα χρησιμοποιήσουμε εικόνες που αποθηκεύονται σε PNG μορφή.
3. Σε μία εικόνα που είναι αποθηκευμένη σε JPEG μορφή μπορούμε να βάλουμε επιπλέον δεδομένα ως EXIF (Microsoft Corporation , 2002). Εάν η εικόνα είναι αποθηκευμένη σε PNG μορφή (που εμείς θα χρησιμοποιήσουμε) δεν μπορούμε να βάλλουμε επιπλέον δεδομένα ως EXIF.
4. Λαμβάνοντας υπόψη την βιβλιογραφία (Google, Bitmap.Config n.d.) θα χρησιμοποιήσουμε ARGB_8888 bitmap configuration. Με αυτό το bitmap configuration έχουμε την μεγαλύτερη δυνατότητα παρέμβασης στην εικόνα, γιατί κάθε pixel αποθηκεύεται σε 4 bytes και κάθε κανάλι (RGB και alpha για την διαύγεια) αποθηκεύεται με ακρίβεια 8 bits (Wells , 2009). Έτσι ώστε να μπορέσουμε να ενσωματώσουμε τις πληροφορίες που θέλουμε (μήνυμα και θέσεις που αυτό αποθηκεύεται) και να μην αλλοιώσουμε την εικόνα.
5. Πριν την αποστολή της εικόνας (αφού έχουν ενσωματωθεί τα δεδομένα σε αυτήν) και αμέσως μετά την λήψης της αυτή θα αποθηκεύεται στο εκάστοτε κινητό τηλέφωνο. Αυτό θα γίνεται για να μπορέσει οποιοσδήποτε να πιστοποιήσει ότι η εικόνα που έχει αποσταλεί και ληφθεί με ενσωματωμένα δεδομένα σε αυτή (με στεγανογραφία) ανοίγει με οποιοδήποτε εφαρμογή προβολής φωτογραφιών.
6. Για να μπορεί ο παραλήπτης να γνωρίζει που τελειώνουν τα δεδομένα που έχει εισάγει ο αποστολέας με στεγανογραφία στην εικόνα, κάθε αλληλουχία δεδομένων που εισάγεται στην εικόνα θα έχει στην αρχή και στο τέλος της ένα διευκρινιστικό αλφαριθμητικό.

Με βάση τους παραπάνω περιορισμούς γίνεται αντιληπτό ότι εάν δεν εφαρμόσουμε κάποιον αλγόριθμο που δεν μεταβάλλει στο ελάχιστο δυνατό την πληροφορία που απαιτείται για να εμφανιστεί σωστά η εικόνα, τότε θα έχουμε φαινόμενα που θα εμφανίζονται παράταιρα pixel στην εικόνα (αλλοιωμένη εικόνα).

Συνεπώς, δημιουργούμε έναν αλγόριθμο αποκλειστικά για χρήση σε Android συσκευές που βασίζεται **στις αρχές του LSB** ώστε να έχουμε όσο τον δυνατόν λιγότερη αλλοίωση σε συνδυασμό με την τυχαιότητα και κρυπτογράφηση. Αυτή την παραλλαγή του αλγορίθμου LSB την ονομάσαμε **LSBv4AwRaC** (LSB variant for Android with Randomness and Cryptography).

3.3 Πρόταση Υλοποίησης Συστήματος

Με στόχο την υποστήριξη του προτεινόμενου αλγορίθμου LSBv4AwRaC σχεδιάστηκε και υλοποιήθηκε σύστημα εφαρμογών με τα ανωτέρω χαρακτηριστικά. Το σύστημα αυτό στηρίζεται στην αποστολή και λήψη εικόνας με ενσωματωμένο κρυφό μήνυμα, σύμφωνα με τις παρακάτω ροές εργασίας. Στις ακόλουθες ενότητες παρουσιάζονται και τα σχετικά UML διαγράμματα.

3.3.1 Αποστολή Εικόνας

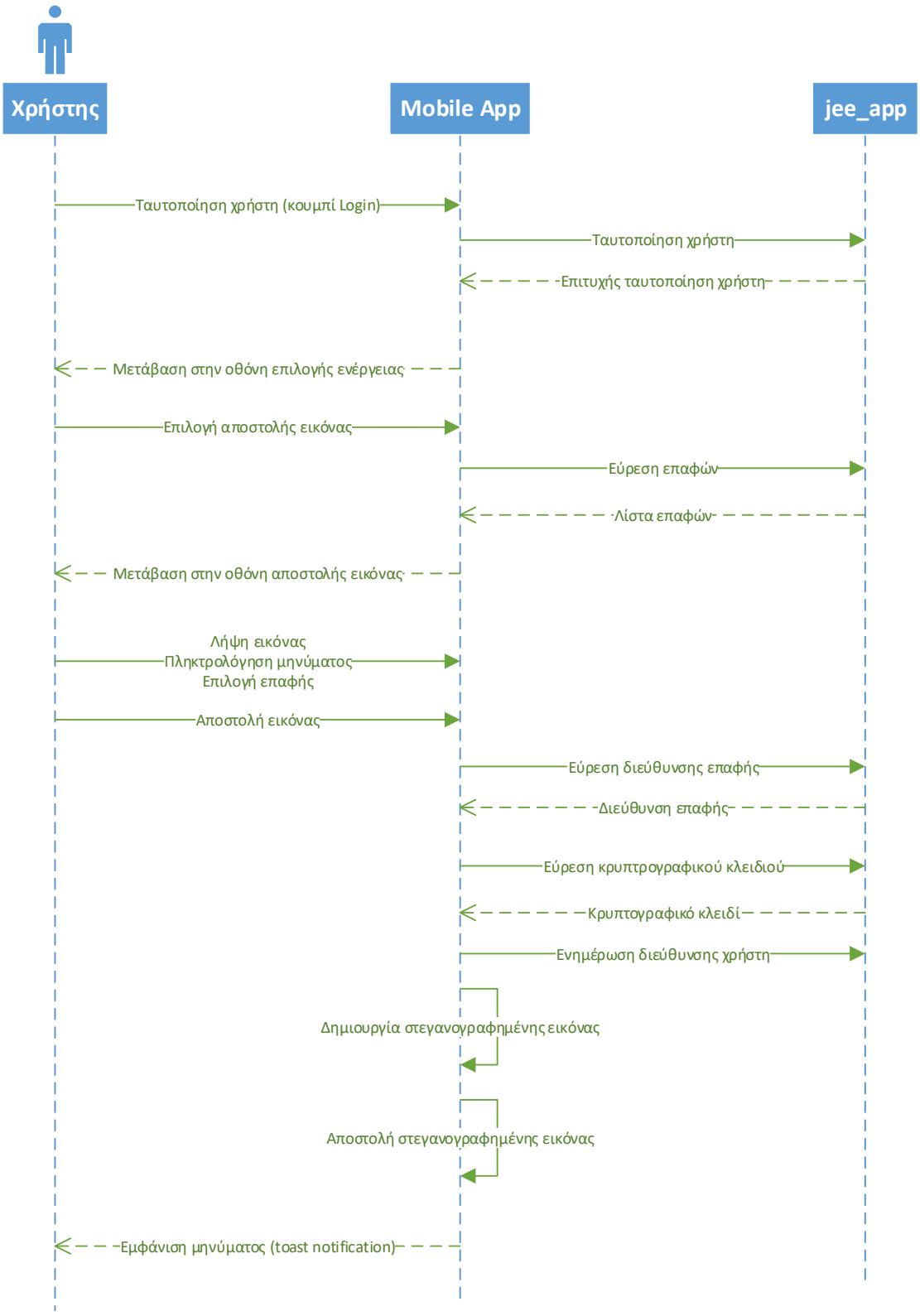
Για την αποστολή της εικόνας, τα βήματα που ακολουθούνται είναι τα ακόλουθα:

1. Ο χρήστης εκτελεί τις ακόλουθες ενέργειες κατά σειρά:
    a. Σύνδεση σε portal και δημιουργία νέου χρήστη
    b. Επιλογή «φιλίας» με υφιστάμενους χρήστες
    c. Τακτοποιείται στην Android εφαρμογή
    d. Προχωράει στην λήψη μιας φωτογραφίας
    e. Πληκτρολογεί το μήνυμα που θέλει να στείλει
    f. Επιλέγει την επαφή θέλει να στείλει την στεγανογραφημένη εικόνα
    g. Ενημερώνει την Android εφαρμογή (π.χ. πατώντας το αντίστοιχο κουμπί) ότι θέλει να στείλει στεγανογραφημένη εικόνα
2. Η Android εφαρμογή από την στιγμή που ο χρήστης έχει κάνει τα παραπάνω βήματα ανακτά από την βάση δεδομένων:
    a. την διεύθυνση της επαφής
    b. το κλειδί κρυπτογράφησης
3. Η Android εφαρμογή ενημερώνει την βάση δεδομένων με την τρέχουσα διεύθυνση του χρήστη
4. Η Android εφαρμογή δημιουργεί την στεγανογραφήμενη εικόνα:
    a. Μετατρέπει το μήνυμα που έχει εισάγει ο χρήστης σε μια σειρά (array) από bytes
    b. Δημιουργεί τόσες τυχαίες θέσεις όσες είναι τα bytes του μηνύματος του χρήστη
    c. Κρυπτογραφεί τις θέσεις των bytes του μηνύματος του χρήστη (χρησιμοποιώντας το κλειδί που ανάκτησε παραπάνω)
    d. Κρυπτογραφεί το κάθε byte του μηνύματος του χρήστη (χρησιμοποιώντας το κλειδί που ανάκτησε παραπάνω)

e. Εισάγει τις κρυπτογραφημένες θέσεις του μηνύματος του χρήστη στην αρχή της εικόνας
f. Εισάγει κάθε byte του μηνύματος του χρήστη σε καθεμία από τις τυχαία δημιουργημένες θέσεις με αύξουσα σειρά
g. Αποθηκεύει την στεγανογραφημένη εικόνα στο κινητό

5. Η Android εφαρμογή αποστέλλει την εικόνα στην παραλήπτη (με socket programming) και εμφανίζει σχετικό μήνυμα στον χρήστη (toast notification)

Στην Εικόνα 8 παρουσιάζεται το UML διάγραμμα ακολουθίας της διαδικασία αποστολής εικόνας.



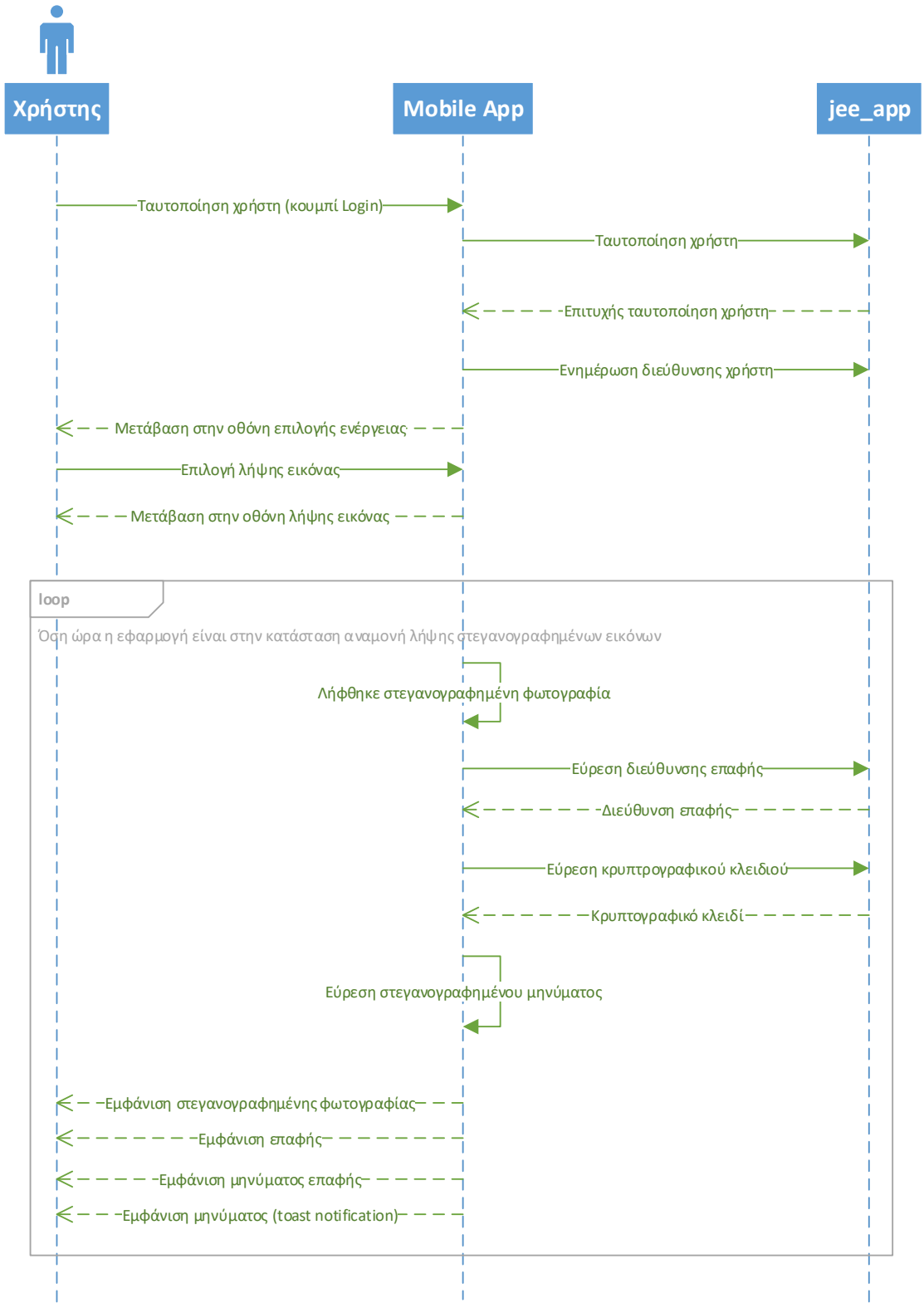
Εικόνα 8, UML διάγραμμα ακολουθίας - διαδικασία αποστολής εικόνας

3.3.2 Λήψη Εικόνας

Για τη λήψη εικόνας, τα βήματα που ακολουθούνται είναι τα ακόλουθα:

1. Προϋποθέτουμε πως ο χρήστης υπάρχει ήδη στην βάση με χρήση του Portal.
2. Ο χρήστης τακτοποιείται στην Android εφαρμογή
3. Η Android εφαρμογή ενημερώνει την βάση δεδομένων με την τρέχουσα διεύθυνση του χρήστη
4. Η Android εφαρμογή λαμβάνει μια στεγανογραφημένη εικόνα (με socket programming) και εμφανίζει σχετικό μήνυμα στον χρήστη (toast notification)
5. Όταν η Android εφαρμογή λάβει μια στεγανογραφημένη εικόνα κάνει τα παρακάτω βήματα για να ανακτήσει το κλειδί κρυπτογράφησης:
    a. Βρίσκει το όνομα της επαφής που έστειλε την εικόνα. Αυτό γίνεται με βάση την διεύθυνση της συσκευής που έστειλε την φωτογραφία. Η Android εφαρμογή αναζητά στην βάση δεδομένων το όνομα της επαφής στην οποία ανήκει η διεύθυνση που έστειλε την φωτογραφία.
    b. Ανακτά από την βάση δεδομένων το κλειδί κρυπτογράφησης
6. Η Android εφαρμογή βρίσκει το κρυπτογραφημένη μήνυμα:
    a. Εξάγει από την αρχή της εικόνας τις κρυπτογραφημένες θέσεις των bytes του μηνύματος της επαφής
    b. Αποκρυπτογραφεί τις θέσεις των bytes του μηνύματος της επαφής (χρησιμοποιώντας το κλειδί που ανάκτησε παραπάνω)
    c. Εξάγει κάθε κρυπτογραφημένο byte του μηνύματος της επαφής από τις θέσεις με αύξουσα σειρά
    d. Αποκρυπτογραφεί το κάθε byte του μηνύματος της επαφής (χρησιμοποιώντας το κλειδί που ανάκτησε παραπάνω)
    e. Συνθέτει το μήνυμα της επαφής ενώνοντας όλα τα bytes που αποκρυπτογράφησε παραπάνω
7. Από την στιγμή που η Android εφαρμογή βρήκε το κρυπτογραφημένο μήνυμα της επαφής εμφανίζει:
    a. Την εικόνα που έστειλε η επαφή
    b. Το όνομα της επαφής
    c. Το μήνυμα που έστειλε η επαφή
    d. Σχετικό μήνυμα στον χρήστη (toast notification)

Στην Εικόνα 9 παρουσιάζεται το UML διάγραμμα ακολουθίας της διαδικασία λήψης εικόνας.



Εικόνα 9, UML διάγραμμα ακολουθίας - διαδικασία λήψης εικόνας

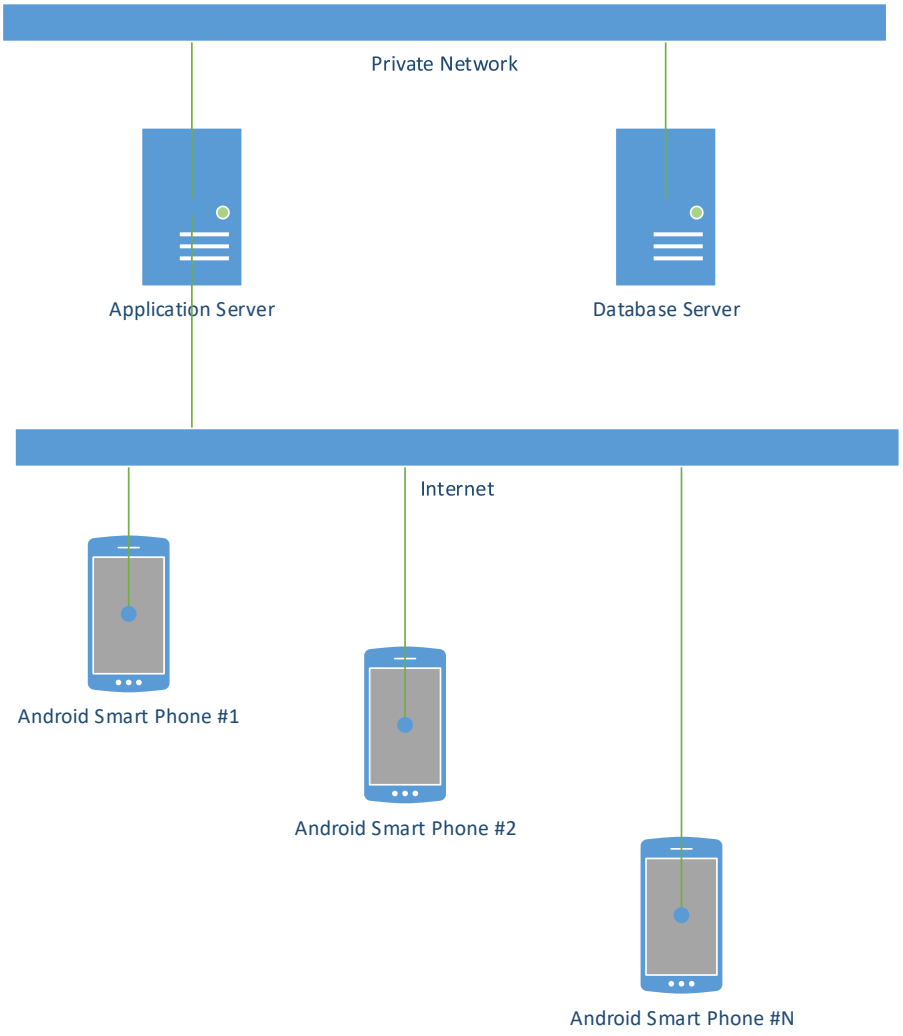
Αναλυτικά, περιγράφεται η υλοποίηση της ανωτέρω μεθοδολογίας/ συστήματος στην ενότητα 4.

3.4 Αρχιτεκτονική Υλοποίησης

Στην παρούσα ενότητα παρουσιάζεται η αρχιτεκτονική υλοποίησης του προτεινόμενου συστήματος, από την άποψη της λογικής αρχιτεκτονικής, αποτυπώνονται οι τεχνολογίες που αξιοποιήθηκαν για την ανάπτυξη του συστήματος, αλλά και η πολυπλοκότητα της υλοποίησης.

Η συνολική λύση που αναπτύχθηκε αποτελείται από τρία (3) κύρια αρχιτεκτονικά μέρη (components), όπως φαίνεται παρακάτω:

1. Βάση δεδομένων
2. JEE εφαρμογή και Web Based Portal
3. Mobile εφαρμογή



Εικόνα 10, Network and Peripherals Diagram

Στην βάση δεδομένων (database server) είναι αποθηκευμένα όλα τα δεδομένα των χρηστών. Τα δεδομένα που είναι αποθηκευμένα στην βάση δεδομένων προσπελαύνονται αποκλειστικά και μόνο από την JEE εφαρμογή μέσω ενός ιδιωτικού δικτύου (private network).

Η JEE εφαρμογή είναι αυτή που διαθέτει τα δεδομένα προς τον έξω κόσμο μέσω του internet (public network) και εκτελείται στον εξυπηρετητή υπηρεσιών διαχείρισης χρηστών και κρυπτογράφησης (application server).

Η mobile εφαρμογή που εκτελείται σε android smart phones επικοινωνεί με την JEE εφαρμογή για να ανακτήσει τα δεδομένα των χρηστών.

Στην Εικόνα 10 παρουσιάζεται το διάγραμμα του δικτύου και των περιφερειακών (Network and Peripherals Diagram). Ως περιφερειακά θεωρούμε την βάση δεδομένων, τον εξυπηρετητή υπηρεσιών διαχείρισης χρηστών και κρυπτογράφησης (ΕΥΔΧΚ) και τα έξυπνα κινητά τηλέφωνα με λειτουργικό Android.

3.4.1 Εφαρμογές

Σε συνέχεια των παραπάνω, οι εφαρμογές που αναπτύχθηκαν στα πλαίσια του προτεινόμενου συστήματος είναι:

1. JEE εφαρμογή
2. Web Based Steganography Portal
3. Mobile εφαρμογή

Ο ρόλος της JEE εφαρμογής είναι να προσπελαύνει την βάση δεδομένων και να εκθέτει τα δεδομένα της στην mobile εφαρμογή. Γι' αυτό το λόγο αποτελείται από τα ακόλουθα δύο (2) συστατικά:

1. **Database Services**: Χρησιμοποιείται για την προσπέλαση της βάσης δεδομένων.
2. **RESTful Services**: Εκθέτει τα δεδομένα της βάσης δεδομένων μέσω RESTful Web Services (Louvel, Templier and Boileau 2012).

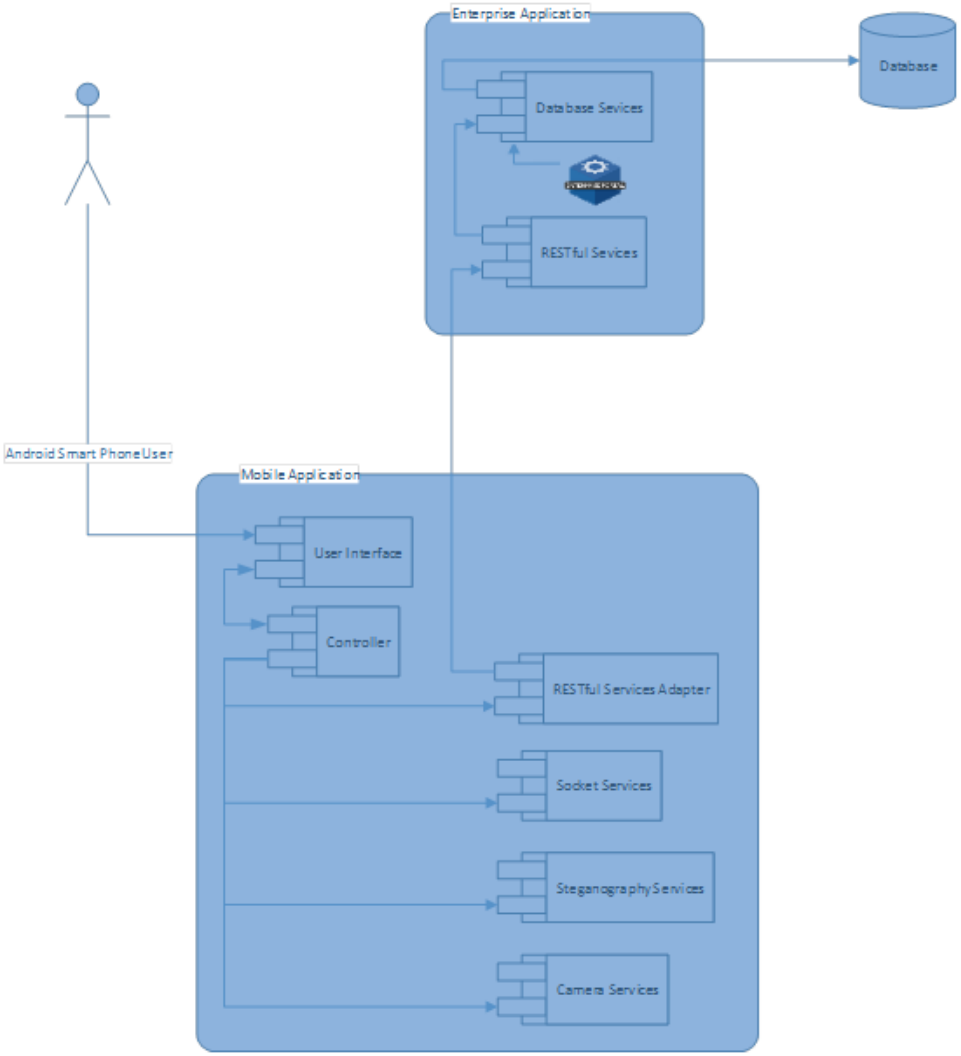
Ο ρόλος του web based Steganography Portal καθιστά δυνατή την δημιουργία νέων χρηστών για την υποστήριξη της εφαρμογής, καθώς και την ενσωμάτωση φιλίας μεταξύ αυτών.

Η mobile εφαρμογή χρησιμοποιείται αποκλειστικά από τον χρήστη και τρέχει σε android smart phones. Η mobile εφαρμογή αναπτύχθηκε με την χρήση του **MVC pattern** και αποτελείται από τα ακόλουθα έξι (6) συστατικά:

1. **User Interface**: Αυτό το συστατικό αναλαμβάνει την διεπαφή με τον χρήστη.

2. **Controller**: Είναι το συστατικό που υλοποιεί το MVC pattern. Ουσιαστικά αναλαμβάνει τον «συντονισμό» όλων των συστατικών της mobile εφαρμογής.
3. **RESTful Services Adapter**: Επικοινωνεί με το συστατικό RESTful Services της JEE εφαρμογής (Arroquic, et al. , 2012).
4. **Socket Services**: Αναλαμβάνει την αποστολή και λήψη των εικόνων μεταξύ των mobile εφαρμογών.
5. **Steganography Services**: Είναι το συστατικό που ασχολείται αποκλειστικά και μόνο με την στεγανογραφία την εικόνων (προσθήκη και ανάγνωση κρυφών μηνυμάτων).
6. **Camera Services**: Χρησιμοποιεί το **Camera API** του android για την λήψη φωτογραφιών.

Στην Εικόνα 11 παρουσιάζεται το διάγραμμα των εφαρμογών (Applications Diagram) που αναπτύχθηκαν.



Εικόνα 11, Applications Diagram

3.4.1.1 Frameworks & Βιβλιοθήκες ανάπτυξης

Για την ανάπτυξη της JEE εφαρμογής χρησιμοποιήσαμε τα ακόλουθα JEE frameworks:

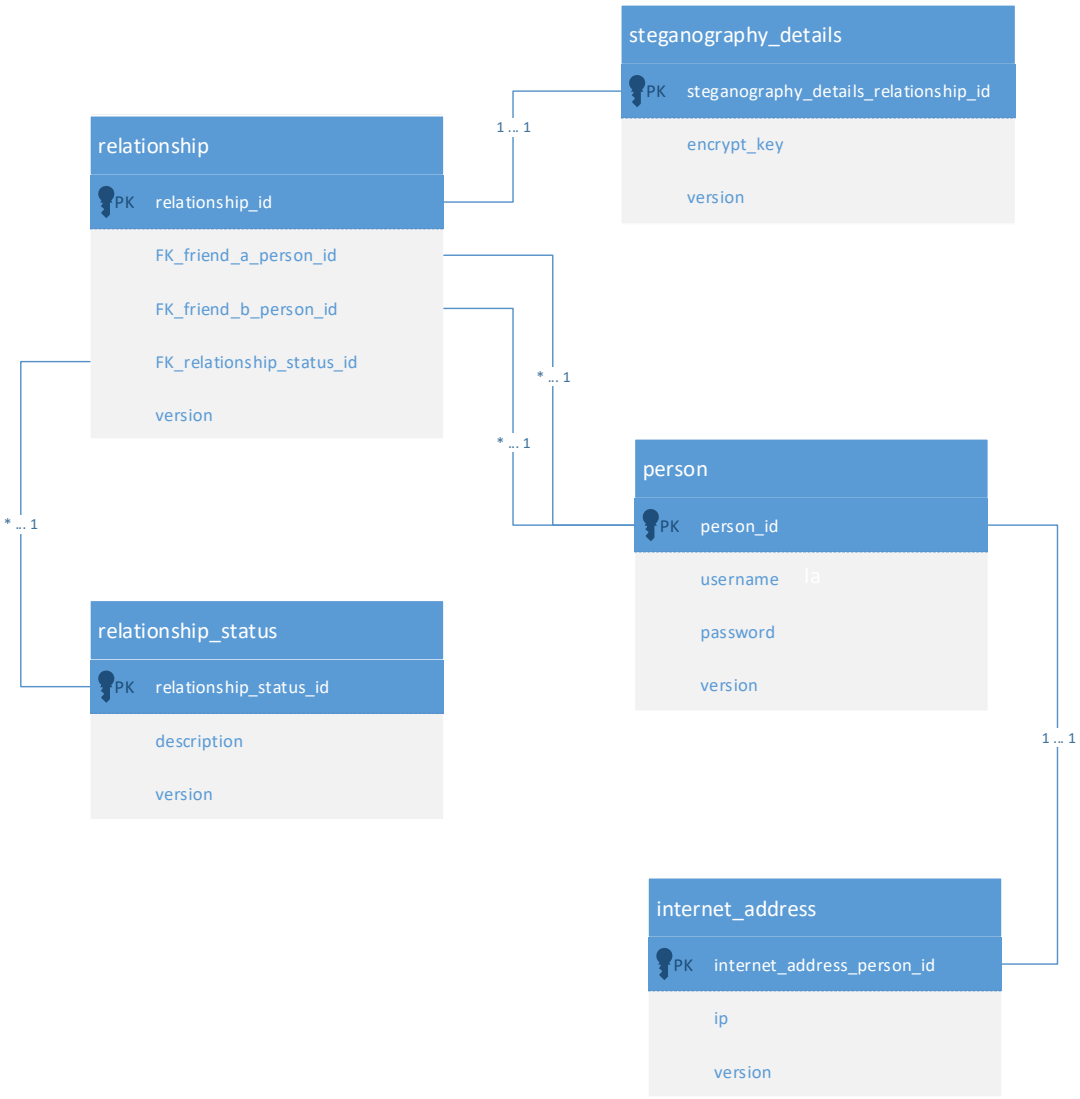
1. **RESTEasy**: δημιουργία RESTful Web Services
2. **Hibernate**: διεπαφή με την βάση δεδομένων
3. **JUnit**: δημιουργία δοκιμών για Java κλάσεις
4. **Arquilian**: δημιουργία δοκιμών για εφαρμογές που τρέχουν σε containers
5. **REST Assured**: δοκιμές RESTful Web Services

3.4.2 Βάση δεδομένων

Η βάση δεδομένων αποθηκεύει πληροφορίες σχετικά με τους χρήστες. Οι πληροφορίες που αποθηκεύονται έχουν σχέση με:

- Στοιχεία χρηστών
- Συσχετίσεις μεταξύ τους (ποιος είναι φίλος με ποιον)
- Κατάσταση φιλίας
- Τελευταία γνωστή IP διεύθυνση

Στην βάση δεδομένων υπάρχει πίνακας audit για κάθε πίνακα για να καταγράφονται με ακρίβεια πότε έγιναν οι αλλαγές και από ποιον. Σε αυτή την νοοτροπία υπάρχει και η στήλη version που χρησιμοποιείται από την JEE εφαρμογή για να καταγράφεται η έκδοση της κάθε εγγραφής του εκάστοτε πίνακα. Στην Εικόνα 12 παρουσιάζουμε το διάγραμμα Οντοτήτων-Συσχετίσεων (E-R Diagram).



Εικόνα 12, Διάγραμμα Οντοτήτων-Συσχετίσεων (E-R Diagram)

4 Υλοποίηση Προτεινόμενου Συστήματος

Στην παρούσα ενότητα παρέχονται λεπτομερείς πληροφορίες σχετικά με την τεχνική υλοποίηση του προτεινόμενου συστήματος στεγανογραφίας.

4.1 Μεθοδολογία Υλοποίησης

Η μεθοδολογία που χρησιμοποιήσαμε για την υλοποίηση της συνολικής λύσης είναι η Agile. Για να εφαρμόσουμε την μεθοδολογία Agile αποτελεσματικά χρησιμοποιήσαμε το framework Scrum (Singh , 2008).

Μέχρι την ολοκλήρωση της συνολικής λύσης πραγματοποιήθηκαν τα ακόλουθα τέσσερα (4) Sprint(s) με τις εξής φάσεις (increments):

1. **Sprint #1 increment**: υλοποίηση JEE εφαρμογής (RESTful Web Services & διεπαφή με την βάση δεδομένων και το Steganography Web Portal)
2. **Sprint #2 increment**: μερική υλοποίηση mobile εφαρμογής (1/3 δυνατοτήτων) με δυνατότητες δημιουργίας και εύρεσης χρηστών (πλήρης αλληλεπίδραση mobile εφαρμογής με JEE εφαρμογή)
3. **Sprint #3 increment**: μερική υλοποίηση mobile εφαρμογής (2/3 δυνατοτήτων) με δυνατότητα λήψης φωτογραφιών και προσθήκης μηνύματος σε αυτές (δυνατότητες στεγανογραφίας)
4. **Sprint #4 increment**: πλήρης υλοποίηση mobile εφαρμογής με προσθήκη δυνατότητας αποστολής στεγανογραφημένης εικόνας και ανάγνωση του μηνύματος από τον παραλήπτη

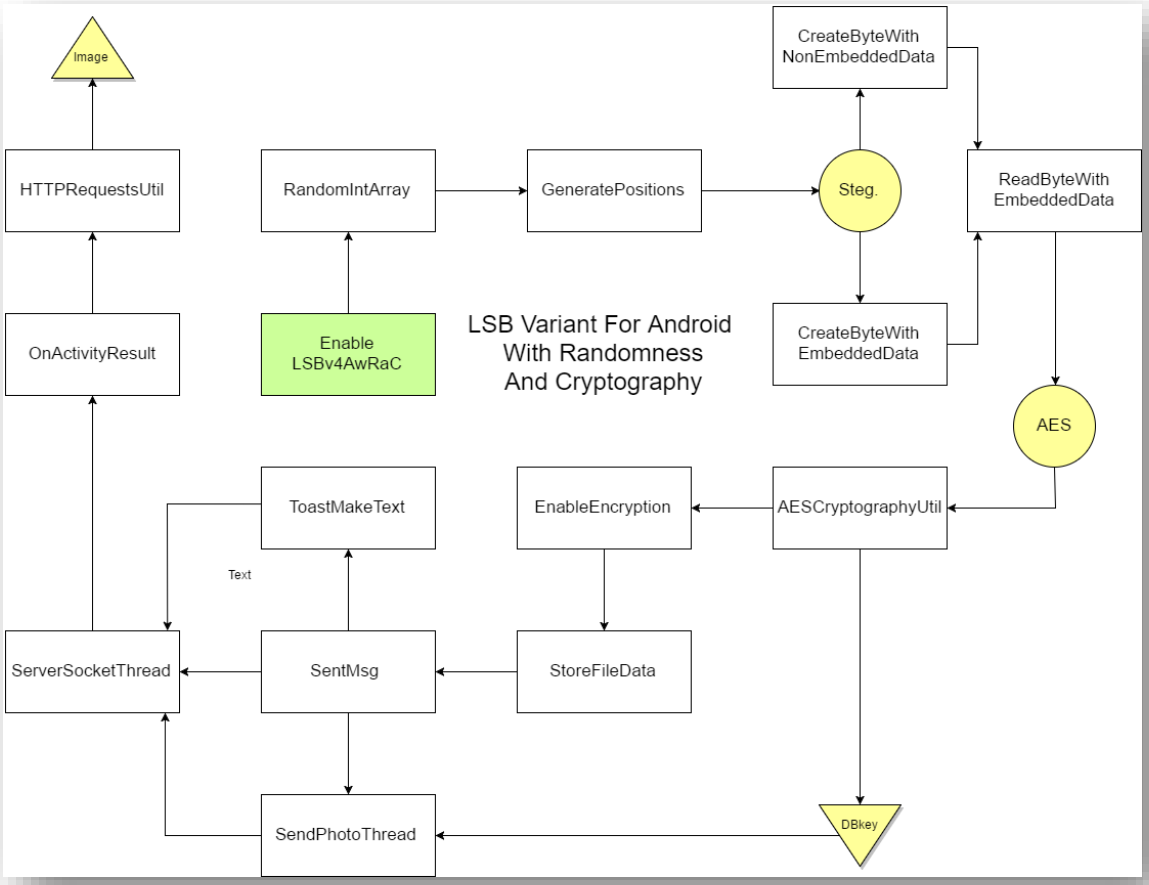
4.2 Υλοποίηση Προτεινόμενης Μεθοδολογίας Στεγανογραφίας

Στη συνέχεια αναλύεται η τεχνική υλοποίηση της προτεινόμενης μεθοδολογίας στεγανογραφίας μέσω του αλγόριθμου LSBv4AwRaC.

4.2.1 Υλοποίηση Αλγορίθμου LSBv4AwRaC

Η ενσωμάτωση του αλγόριθμου LSBv4AwRaC στο προτεινόμενο σύστημα, βασίστηκε στις επιμέρους υλοποιήσεις των βασικών χαρακτηριστικών του: τυχαιότητα, στεγανογραφία και κρυπτογραφία, οι οποίες περιγράφονται αναλυτικά στη συνέχεια.

Στο παρακάτω διάγραμμα παρουσιάζεται η σχηματική απεικόνιση του προτεινόμενου αλγορίθμου LSBv4AwRaC.



Εικόνα 13, Προτεινόμενος αλγόριθμος LSBv4AwRaC

4.2.1.1 Υλοποίηση Τυχαιότητας

Η υλοποίηση της τυχαιότητας προϋποθέτει την δημιουργία τόσων τυχαίων θέσεων όσα είναι και τα bytes του μηνύματος του χρήστη που πρέπει να ενσωματωθούν στην φωτογραφία. Έπειτα από εκτεταμένες δοκιμές που πραγματοποιήσαμε καταλήξαμε στα ακόλουθα συμπεράσματα:

1. Οι θέσεις του μηνύματος του χρήστη που ενσωματώνονται στην αρχή της φωτογραφίας καταλαμβάνουν το πολύ 2000 bytes.
2. Ένα κρυπτογραφημένο byte μηνύματος χρήστη καταλαμβάνει το πολύ 200 bytes.
3. Μία φωτογραφία που απεικονίζεται στην οθόνη ενός Android device έχει μέγεθος το πολύ 40000 bytes.
4. Ένα μήνυμα χρήστη σε Android συσκευή δεν έχει μέγεθος μεγαλύτερο των 200 bytes.

Με βάση τα παραπάνω συμπεράσματα πραγματοποιούνται στον κώδικα που έχουμε αναπτύξει οι ακόλουθες ενέργειες:

1. Δημιουργία λίστας τυχαίων θέσεων που κυμαίνονται μεταξύ των αριθμών 2000 και 3000. Ο λόγος που επιλέξαμε αυτό το εύρος είναι γιατί:
    a. 2000 είναι το μέγιστο πλήθος των bytes που απαιτούνται για την ενσωμάτωση των θέσεων του μηνύματος του χρήστη στην αρχή της φωτογραφίας. Συνεπώς, το 2000 είναι το κατώτατο όριο των τυχαίων θέσεων του μηνύματος του χρήστη.
    b. Επειδή ένα μήνυμα χρήστη δεν έχει πάνω από 200 bytes το 3000 είναι το μέγιστο άνω όριο των τυχαίων θέσεων του μηνύματος του χρήστη.

    Ο κώδικας που υλοποιεί την πρώτη (1η) ενέργεια παρατίθεται στην συνέχεια.

```
/**
 * @param size
 * @param min
 * @param max
 * @return
 */
public static List<Integer> randomIntArray(int size, int min, int
max) {
    Random random = new Random();
    final Set<Integer> intSet = new HashSet<>();

    while (intSet.size() < size) {
        intSet.add(random.nextInt((max - min) + 1) + min);
    }

    List<Integer> integerList = new ArrayList<>();
    Iterator<Integer> iter = intSet.iterator();

    for (int i = 0; iter.hasNext(); ++i) {
        integerList.add(i, iter.next());
    }

    Log.d(TAG, "Random int array: " + integerList.toString());

    return integerList;
}
```

2. Τοποθετούνται σε αύξουσα σειρά οι τυχαίες θέσεις του μηνύματος του χρήστη.
3. Επειδή κάθε κρυπτογραφημένο byte του μηνύματος του χρήστη μπορεί να καταλαμβάνει το πολύ 200 bytes, πολλαπλασιάζουμε κάθε μία από τις τυχαίες θέσεις με τον αριθμό 1,0 με βήμα 0,1. Στην συνέχεια η τυχαία θέση αντικαθίσταται από το ακέραιο μέρος του γινομένου. Π.χ. πολλαπλασιάζουμε την πρώτη τυχαία θέση με τον αριθμό 1,0, την δεύτερη τυχαία θέση με τον αριθμό 1,1, την τρίτη τυχαία θέση με τον αριθμό 1,2 κ.ο.κ.. Ο λόγος που το κάνουμε αυτό είναι για να είμαστε σίγουροι ότι κάθε τυχαία θέση θα έχει διαφορά από την προηγούμενη τουλάχιστον κατά

200. Π.χ. έστω ότι έχουμε τις τρεις (3) πρώτες τυχαίες θέσεις 2001, 2010 και 2011:

a. Πρώτη θέση: το 2001 πολλαπλασιαζόμενο με το 1,0 γίνεται 2001
b. Δεύτερη θέση: το 2010 πολλαπλασιαζόμενο με το 1,1 γίνεται 2211, διαφέρει από το προηγούμενο κατά 210
c. Τρίτη θέση: το 2011 πολλαπλασιαζόμενο με το 1,2 γίνεται 2413, διαφέρει από το προηγούμενο κατά 202

Ο κώδικας που υλοποιεί την δεύτερη (2η) και (3η) ενέργεια παρατίθεται στην συνέχεια. Ουσιαστικά σε αυτόν το κώδικα καλείται πρώτα η μέθοδος **randomIntArray** που υλοποιεί την πρώτη (1η) ενέργεια και στην συνέχεια εκτελούνται οι ενέργειες δύο (2) και τρία (3).

```
/**
 *
 * @param size
 * @param min
 * @param max
 * @param multiplierOffset
 * @return
 */
public static List<Integer> generatePositions(int size, int min, int
max, double multiplierOffset) {
    List<Integer> integerList = randomIntArray(size, min, max);

    Collections.sort(integerList);

    List<Integer> positions = new ArrayList<>();

    double multiplier = 1;

    for(int i: integerList) {
        positions.add((int) (i * multiplier));

        multiplier += multiplierOffset;
    }

    return positions;
}
```

4.2.1.2 Υλοποίηση Στεγανογραφίας

Για να μπορέσουμε να εφαρμόσουμε στεγανογραφία στην φωτογραφία που θέλουμε να αποστείλουμε σε μία επαφή πρέπει να μεταβάλλουμε κάθε byte αυτής της φωτογραφίας. Στην συνέχεια παρατίθεται ο κώδικας για τις τρεις (3) σημαντικότερες μεθόδους της εφαρμογής που αναλαμβάνουν την ενσωμάτωση του μηνύματος στην φωτογραφία και στο διάβασμα αυτού του μηνύματος. Οι τρεις (3) αυτοί μέθοδοι για τις λειτουργίες που επιτελούν βασίζονται στην άλγεβρα Boole για να εκτελέσουν πράξεις σε bytes και λειτουργούν για μηνύματα ανεξαρτήτως εάν είναι κωδικοποιημένα ή όχι.

- **createByteWithNonEmbeddedData**: δημιουργείται ένα byte της στεγανογραφημένης εικόνας χωρίς να έχει ενσωματωθεί κανένα μήνυμα σε αυτό.

```
/**
 * @param origBitmapPixels
 * @param element
 * @param channelIndex
 * @return
 */
private byte createByteWithNonEmbeddedData(
        int[] origBitmapPixels, int element, int channelIndex) {
    return (byte) ((origBitmapPixels[element] >>
Constants.BINARY[channelIndex]) & 0xFF);
}
```

- **createByteWithEmbeddedData**: δημιουργείται ένα byte της στεγανογραφημένης εικόνας με μερική ή πλήρη ενσωμάτωση ενός byte από το κωδικοποιημένο μήνυμα του χρήστη.

```
/**
 *
 * @param origBitmapPixels
 * @param element
 * @param channelIndex
 * @param currentMessage
 * @param currentMessageIndex
 * @param shiftIndex
 * @return
 */
private byte createByteWithEmbeddedData(
        int[] origBitmapPixels, int element, int channelIndex, byte[]
currentMessage, int currentMessageIndex, int shiftIndex) {
    return (byte) (((createByteWithNonEmbeddedData(origBitmapPixels,
element, channelIndex)) & 0xFC) |
            ((currentMessage[currentMessageIndex] >>
Constants.TO_SHIFT[(shiftIndex) % Constants.TO_SHIFT.length]) &
0x3));// 6
}
```

- **readByteWithEmbeddedData**: διαβάζεται μερικώς ή πλήρως ένα byte από το κωδικοποιημένο μήνυμα του χρήστη.

```
/**
 *
 * @param tmp
 * @param stegBitmapBytes
 * @param i
 * @param shiftIndex
 * @return
 */
private byte readByteWithEmbeddedData(byte tmp, byte[]
stegBitmapBytes, int i, int shiftIndex) {
    return (byte) (tmp | ((stegBitmapBytes[i] <<
Constants.TO_SHIFT[shiftIndex % Constants.TO_SHIFT.length]) &
```

```
            Constants.AND_BYTE[shiftIndex %
Constants.TO_SHIFT.length]));
}
```

4.2.1.3 Υλοποίηση Κρυπτογραφίας

Όλες οι μέθοδοι για την υλοποίηση της κρυπτογραφίας βρίσκονται στην κλάση **AESCryptographyUtil**. Για να μπορέσουμε να κρυπτογραφήσουμε ή να αποκρυπτογραφήσουμε ένα μήνυμα του χρήστη αυτό που πρέπει πρώτα να γίνει είναι να ανακτήσουμε από την βάση δεδομένων το κλειδί. Εφόσον, ανακτηθεί από την βάση δεδομένων το κλειδί δημιουργούμε ένα στιγμιότυπο της παραπάνω κλάσης περνώντας σαν παράμετρο το Context της εφαρμογής και το κλειδί. Στην συνέχεια καλούμε:

1. την μέθοδο **encrypt** για την κρυπτογράφηση ενός array από bytes και

```
/**
 * @param clear
 * @return
 */
public byte[] encrypt(byte[] clear) {
    byte[] encrypted = null;

    Log.v(TAG, "Trying to encrypt " + Arrays.toString(clear) + " with
key " + Arrays.toString(rawKey));

    try {
        SecretKeySpec skeySpec = new SecretKeySpec(rawKey, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding",
"BC");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        encrypted = cipher.doFinal(clear);
    } catch (NoSuchProviderException ex) {
        Log.e(TAG, "NoSuchProviderException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (NoSuchPaddingException ex) {
        Log.e(TAG, "NoSuchPaddingException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (NoSuchAlgorithmException ex) {
        Log.e(TAG, "NoSuchAlgorithmException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (InvalidKeyException ex) {
        Log.e(TAG, "InvalidKeyException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (IllegalBlockSizeException ex) {
        Log.e(TAG, "IllegalBlockSizeException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (BadPaddingException ex) {
        Log.e(TAG, "BadPaddingException: " + ex.getMessage());

        ex.printStackTrace();
    }
```

```
    Log.v(TAG, "Encrypted: " + Arrays.toString(encrypted));

    return encrypted;
}
```

2. την μέθοδο **decrypt** για την αποκρυπτογράφηση ενός array από bytes.

```
/**
 * @param encrypted
 * @return
 */
public byte[] decrypt(byte[] encrypted) {
    byte[] decrypted = null;

    Log.v(TAG, "Trying to decrypt " + Arrays.toString(encrypted) + "
with key " + Arrays.toString(rawKey));

    try {
        SecretKeySpec skeySpec = new SecretKeySpec(rawKey, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding",
"BC");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        decrypted = cipher.doFinal(encrypted);
    } catch (NoSuchProviderException ex) {
        Log.e(TAG, "NoSuchProviderException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (NoSuchPaddingException ex) {
        Log.e(TAG, "NoSuchPaddingException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (NoSuchAlgorithmException ex) {
        Log.e(TAG, "NoSuchAlgorithmException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (InvalidKeyException ex) {
        Log.e(TAG, "InvalidKeyException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (IllegalBlockSizeException ex) {
        Log.e(TAG, "IllegalBlockSizeException: " + ex.getMessage());

        ex.printStackTrace();
    } catch (BadPaddingException ex) {
        Log.e(TAG, "BadPaddingException: " + ex.getMessage());

        ex.printStackTrace();
    }

    Log.v(TAG, "Decrypted: " + Arrays.toString(encrypted));

    return decrypted;
}
```

4.2.2 Υλοποίηση Αποστολής & Λήψης Φωτογραφιών

Η αποστολή και η λήψη φωτογραφιών έχει υλοποιηθεί με **socket programming**. Μετά το πέρας της αποστολής και της λήψη φωτογραφιών εμφανίζεται στην Android συσκευή ένα μήνυμα (**toast notification**):

1. Κώδικας εμφάνισης toast notification κατά την επιτυχημένη αποστολή στεγανογραφημένης εικόνας:

```
final String sentMsg = "Successful sent photo";

activity.runOnUiThread(new Runnable() {

    @Override
    public void run() {
        Toast.makeText(activity,
                sentMsg,
                Toast.LENGTH_LONG).show();
    }
});
```

2. Κώδικας εμφάνισης toast notification κατά την επιτυχημένη λήψη στεγανογραφημένης εικόνας:

```
Toast.makeText(activity,
        "Successful receive photo",
        Toast.LENGTH_LONG).show();
```

Κατά την λήψη φωτογραφιών η Android συσκευή λειτουργεί σαν server. Όταν λάβει μία εισερχόμενη σύνδεση, δηλαδή όταν μία επαφή θέλει να στείλει μία φωτογραφία, τότε η Android συσκευή αποδέχεται την εισερχομένη σύνδεση και λαμβάνει την φωτογραφία. Η λήψη φωτογραφιών υλοποιείται από την κλάση **ServerSocketThread** που υλοποιεί την λειτουργικότητα του server στην εκάστοτε Android συσκευή. Η κλάση αυτή καλείται με το ακόλουθο κομμάτι κώδικα.

```
serverSocketThread = new
ServerSocketThread(ReceivePhotoActivity.this);
serverSocketThread.start();
```

Αντίστοιχα κατά την αποστολή φωτογραφιών η Android συσκευή λειτουργεί σαν client που προσπαθεί να συνδεθεί στην επαφή, στην οποία επαφή η Android συσκευή λειτουργεί σαν server, και να στείλει την φωτογραφία (μετά την επιτυχημένη σύνδεση). Για την αποστολή μίας φωτογραφίας καλείται η κλάση **SendPhotoThread** σύμφωνα με το ακόλουθο κομμάτι κώδικα.

```
// Send picture
SendPhotoThread sendPhotoThread =
        new SendPhotoThread(
```

```
                internetAddress,
                Constants.SOCKET_SERVER_PORT,
                SendPhotoActivity.this,
                pictUtil);

sendPhotoThread.start();
```

4.2.3 Υλοποίηση Κλήσεων RESTful Web Services

Όλες οι σχετικές μέθοδοι για τις κλήσεις των RESTful Web Services υλοποιούνται αποκλειστικά στην Java κλάση **HTTPRequestsUtil**. Η μέθοδος που καλείται για την εκτέλεση κάθε RESTful Web Service είναι η **performCall**.

Η παραπάνω μέθοδος έχει τις ακόλουθες δυνατότητες:

1. Πραγματοποίηση GET και POST HTTP κλήσεων

```
if (httpMethod.equals(SupportedHTTPMethods.POST)) {
    conn.setDoInput(true);
    conn.setDoOutput(true);

    os = conn.getOutputStream();
    writer = new BufferedWriter(
            new OutputStreamWriter(os, "UTF-8"));

    writer.write(getPostDataString(postDataParams));
}
```

2. Επιστροφή της απάντησης του RESTful Web Service σαν αλφαριθμητικό (String) σε περίπτωση επιτυχίας.
3. Έλεγχος εάν ο HTTP κωδικός απάντησης του RESTful Web Service είναι 200. Εάν δεν είναι 200 ο HTTP κωδικός απάντησης του RESTful Web Service τότε επιστρέφει κενή απάντηση

```
if (responseCode == HttpsURLConnection.HTTP_OK) {
    String line;
    BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream()));

    while ((line = br.readLine()) != null) {
        Log.d(TAG, "line: " + line);

        response += line;
    }
} else {
    response = "";
}
```

4.2.4 Επεξεργασία Φωτογραφίας σε Επίπεδο Thumbnail

Ένα από τα βασικά χαρακτηριστικά της εφαρμογής είναι ότι επεξεργαζόμαστε τις φωτογραφίες που εμφανίζονται στην συσκευή της οθόνης του Android device που χρησιμοποιούμε. Στον παρακάτω κομμάτι κώδικα βλέπουμε ότι η μεταβλητή

bitmap που επεξεργαζόμαστε είναι αυτή που εμφανίζουμε στην οθόνη της Android συσκευής (μεταβλητή **imageView**).

```
@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    Bitmap photo = (Bitmap) data.getExtras().get("data");
    bitmap = photo.copy(Constants.BITMAP_CONFIG, true);

    if(bitmap != null) {
        Log.d(TAG, "Successful bitmap copy creation from captured
photo");
    } else {
        Log.e(TAG, "Unsuccessful bitmap copy creation from captured
photo");
    }

    imageView.setImageBitmap(bitmap);
}
```

Ουσιαστικά επεξεργαζόμαστε την φωτογραφία που φαίνεται στην οθόνη και όχι αυτή που έχει αποθηκευτεί στον χώρο αποθήκευσης του κινητού τηλέφωνου. Οι λόγοι που το κάνουμε αυτό είναι:

1. Μείωση της επεξεργαστικής ισχύς που απαιτείται, π.χ. άλλη επεξεργαστική ισχύ θέλει μία φωτογραφία που έχει αποθηκευτεί σαν 8 megapixels με διαστάσεις 3264 x 2448 και άλλο μία φωτογραφία διαστάσεων 204 x 153 (όσο έχουμε επιλέξει προγραμματιστικά να φαίνεται στην οθόνη του Android device).
2. Μείωση του όγκου του δεδομένων που θα μεταφέρεται μέσω του δικτύου, π.χ. μία φωτογραφία 8 megapixels με διαστάσεις 3264 x 2448 έχει μέγεθος 1,38 MB ενώ μια φωτογραφία διαστάσεων 204 x 153 έχει μέγεθος 56 KB.
3. Επειδή υπάρχει περίπτωση οι χρήστες των εφαρμογών μας να μην θέλουν να αποθηκεύσουν την φωτογραφία στο Android device τους, οπότε να μην μπορούμε να την ανακτήσουμε από εκεί.

4.2.5 Εναλλαγή Αλγορίθμων LSBv4AwRaC και MobiStego

Με σκοπό την εύκολη εναλλαγή ανάμεσα σε διαφορετικούς αλγόριθμους (LSBv4AwRaC και MobiStego) κατά την εκτέλεση των πειραματικών μετρήσεων, υλοποιήσουμε διαφορετικούς αλγόριθμους στεγανογραφίας στην Android εφαρμογή δημιουργήσαμε το ακόλουθο Java interface που πρέπει να ακολουθούν όλοι οι αλγόριθμοι στεγανογραφίας που υλοποιούνται σε αυτήν.

```
public interface SteganographyUsageInterface {

    public Bitmap encode(Bitmap sourceBitmap, String message);
```

```
    public String decode(PictUtil pictUtil);
}
```

Η μέθοδος **encode** λαμβάνει ως παράμετρο μία φωτογραφία (Bitmap) και ένα μήνυμα (String) και επιστρέφει την στεγανογραφημένη εικόνα. Η μέθοδος **decode** λαμβάνει ως παράμετρο ένα αντικείμενο τύπου PictUtil που περιέχει πληροφορίες για την στεγανογραφημένη φωτογραφία που είναι αποθηκευμένη στην Android συσκευή και επιστρέφει ως παράμετρο το μήνυμα της επαφής που έχει ενσωματωθεί με στεγανογραφία στην φωτογραφία. Οι αλγόριθμοι στεγανογραφίας που έχουμε υλοποιήσει είναι:

1. ο LSBv4AwRaC (ο αλγόριθμος που αναπτύξαμε) και
2. ο αλγόριθμος LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego.

Εξ΄ ορισμού στην Android εφαρμογή είναι ενεργοποιημένος ο αλγόριθμος που έχουμε αναπτύξει:

1. Java κλάση **SendPhotoActivity**

```
// Steganography
long lStartTime = System.currentTimeMillis();

// LSBv4AwRaC Algorithm
Bitmap stegBitmap = new LSBv4AwRaCUsage(this,
encryptKey).encode(bitmap, message);

// LSB2bit Algorithm
//Bitmap stegBitmap = new LSB2bitCUsage(this).encode(bitmap,
message);

long lEndTime = System.currentTimeMillis();
long difference = lEndTime - lStartTime;

Log.i(TAG, "Elapsed milliseconds for encoding: " + difference);
```

Εάν θέλουμε να χρησιμοποιήσουμε τον αλγόριθμο LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego αρκεί να κάνουμε comment in την γραμμή κώδικα

```
Bitmap stegBitmap = new LSBv4AwRaCUsage(this,
encryptKey).encode(bitmap, message);
```

και comment out την γραμμή κώδικα

```
//Bitmap stegBitmap = new LSB2bitCUsage(this).encode(bitmap,
message);
```

2. Java κλάση **ReceicePhotoThread**

```
// Steganography
long lStartTime = System.currentTimeMillis();

// LSBv4AwRaC Algorithm
String message = new
LSBv4AwRaCUsage(activity.getApplicationContext(),
encryptKey).decode(pictUtil);

// LSB2bit Algorithm
//String message = new
LSB2bitCUsage(activity.getApplicationContext()).decode(pictUtil);

long lEndTime = System.currentTimeMillis();
long difference = lEndTime - lStartTime;

Log.i(TAG, "Elapsed milliseconds for decoding: " + difference);
```

Εάν θέλουμε να χρησιμοποιήσουμε τον αλγόριθμο LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego αρκεί να κάνουμε comment in την γραμμή κώδικα

```
String message = new
LSBv4AwRaCUsage(activity.getApplicationContext(),
encryptKey).decode(pictUtil);
```

και comment out την γραμμή κώδικα

```
//String message = new
LSB2bitCUsage(activity.getApplicationContext()).decode(pictUtil);
```

4.3 Ανάπτυξη Εφαρμογών

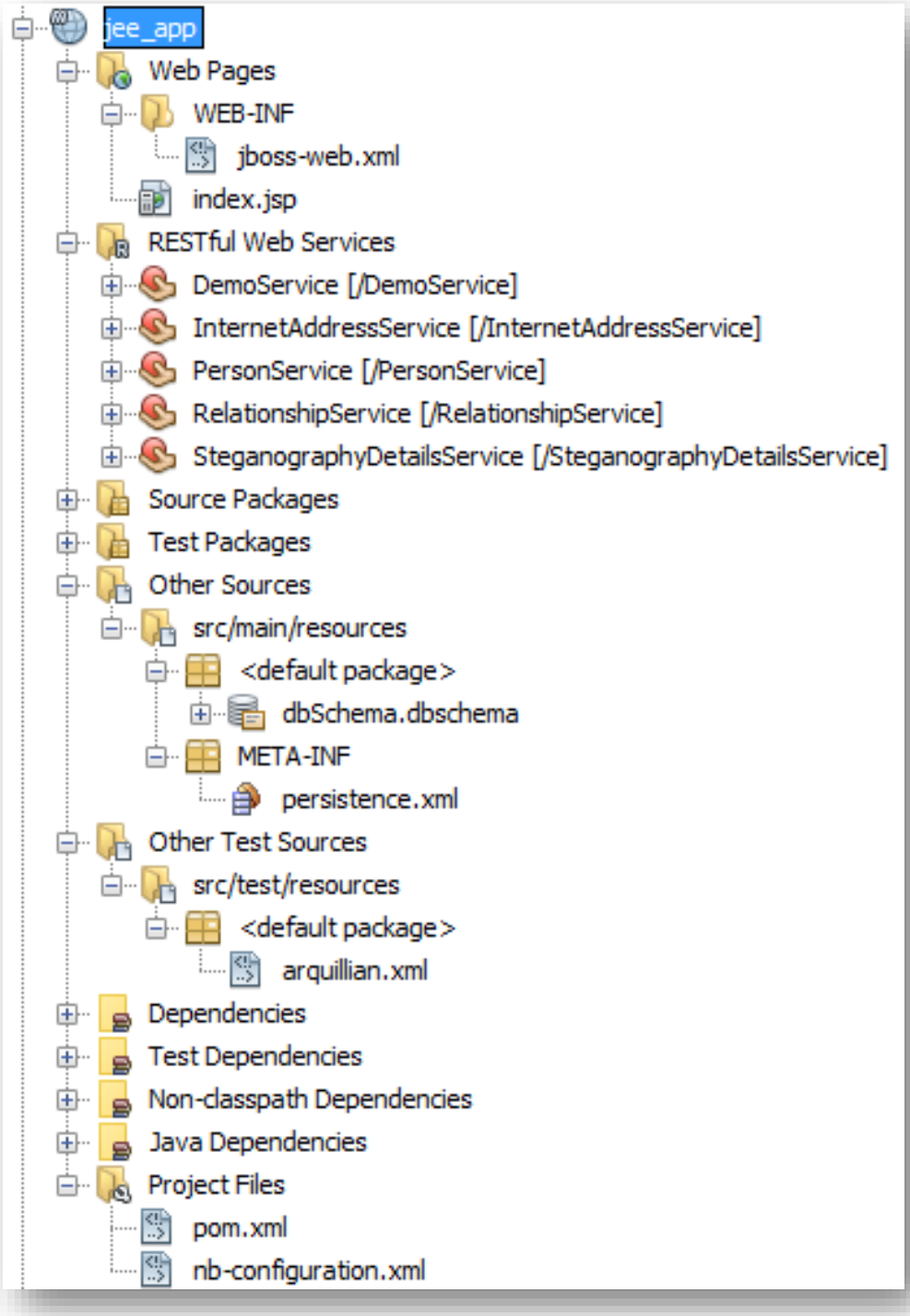
Όπως αναφέρεται και παραπάνω το προτεινόμενο σύστημα συντίθεται από πλειάδα εφαρμογών, των οποίων οι τεχνικές λεπτομέρειες παρουσιάζονται ακολούθως.

4.3.1 Desktop Εφαρμογές

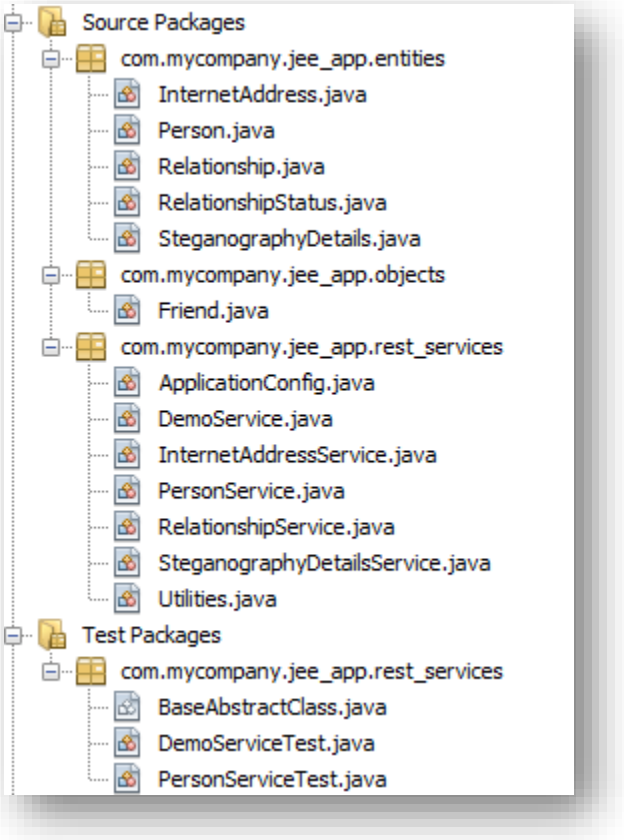
4.3.1.1 JEE Εφαρμογή

Η JEE εφαρμογή που θα τρέχει στον JBoss EAP επικοινωνεί με την βάση δεδομένων με το framework **Hibernate** (Ang and Wen-xue , 2005). Η JEE εφαρμογή εκθέτει την λειτουργικότητά με RESTful Web Services χρησιμοποιώντας το framework **RESTEasy**. Το όνομα της εφαρμογής θα είναι jee_app.

Στην Εικόνα 15 παρουσιάζεται η διάρθρωση των source και test java packages της εφαρμογής jee_app.



Εικόνα 14, Διάρθρωση maven project εφαρμογής jee_app



Εικόνα 15, Διάρθρωση source και test packages εφαρμογής jee_app

Στον Πίνακας 2 που ακολουθεί παρουσιάζονται αναλυτικά τα Rest Services που δημιουργήσαμε. Στην Εικόνα 14 παρουσιάζεται η διάρθρωση του maven project της εφαρμογής jee_app.

Rest Service	HTTP Μέθοδοι	Τύπος	Παράμετροι	Περιγραφή
DemoService	Hello	GET	String	Επίδειξη λειτουργικό-τητας
InternetAddress Service	Internet Addresses	GET	Integer	Εύρεση IP χρήστη
		PUT	Integer String	Δημιουργία IP χρήστη
		POST	Integer String	Ενημέρωση IP χρήστη

	/internet Addresses/ ip	GET	String	Εύρεση id χρήστη
PersonService	persons	GET	String	Εύρεση χρήστη
		PUT	String String	Δημιουργία χρήστη
		POST	String String	Ενημέρωση κωδικού χρήστη
	persons/login	POST	String String	Ταυτοποίηση χρήστη
	persons/id	GET	Integer	Εύρεση ονόματος χρήστη
RelationshipService	relationships	GET	Integer	Εύρεση επαφών χρήστη
		PUT	Integer Integer	Εύρεση id φιλίας
		POST	Integer String	Δημιουργία φιλίας
Steganography DetailsService	steganography Details	GET	Integer	Εύρεση λεπτομερειών στεγανο-γραφίας
		PUT	Integer String	Δημιουργία λεπτομερειών στεγανο-γραφίας
		POST	Integer String	Ενημέρωση λεπτομερειών στεγανο-γραφίας

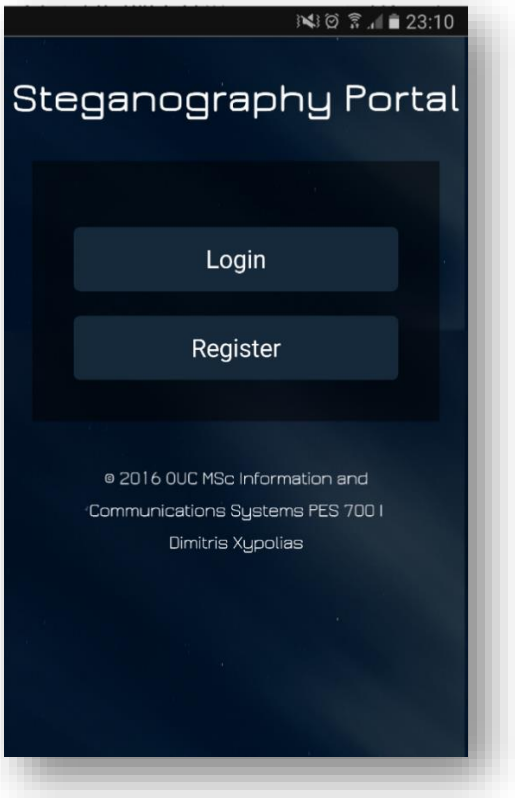
Πίνακας 2, Πίνακας Rest Services

4.3.1.2 Web Based Portal - Διάρθρωση Source Code

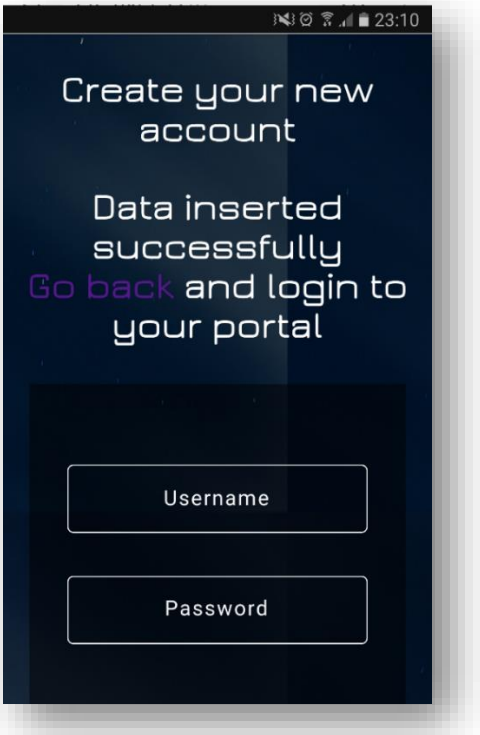
Η δημιουργία του Portal κρίθηκε απαραίτητη με σκοπό την δημιουργία νέων χρηστών για την εφαρμογή, καθώς και την σύνδεση μεταξύ τους για την δυνατότητα αποστολής των φωτογραφιών.

Στην συγκεκριμένη δικτυακή εφαρμογή που δημιουργήθηκε ο χρήστης έχει την δυνατότητα να περιηγηθεί αρχικά σε δυο επιλογές, την σύνδεση (Login) και την εγγραφή (Register). Επιλέγοντας την σύνδεση ο χρήστης μπορεί να προσπαθήσει να συνδεθεί με κάποιον από τους υφιστάμενους χρήστες της βάσης δεδομένων αλλιώς με την επιλογή εγγραφή να δημιουργήσει έναν νέο χρήστη. Από την στιγμή που ο χρήστης είναι συνδεδεμένος στο Steganography Portal η διαδικασία χωρίζεται σε τρεις βασικές κατηγορίες, την κατηγορία που εμφανίζονται οι χρήστες που έχει ο συνδεδεμένος χρήστης διασύνδεση μεταξύ τους (relationship), την κατηγορία προσθήκης νέας φιλίας, καθώς και την κατηγορία download της mobile εφαρμογής που θα αναπτυχθεί παρακάτω. Η επιλογή προσθήκης νέας φιλίας είναι μια εύκολη διαδικασία όπως διακρίνεται και στις παρακάτω αποτυπώσεις οθόνης επιλέγοντας τους χρήστες που επιθυμούμε, δημιουργούμε αυτόματα νέο relationship καθώς και την δημιουργία του κρυπτογραφικού κλειδιού που απαιτείται για την σύναψη σχέσης και αποστολής της στεγανογραφημένης φωτογραφίας.

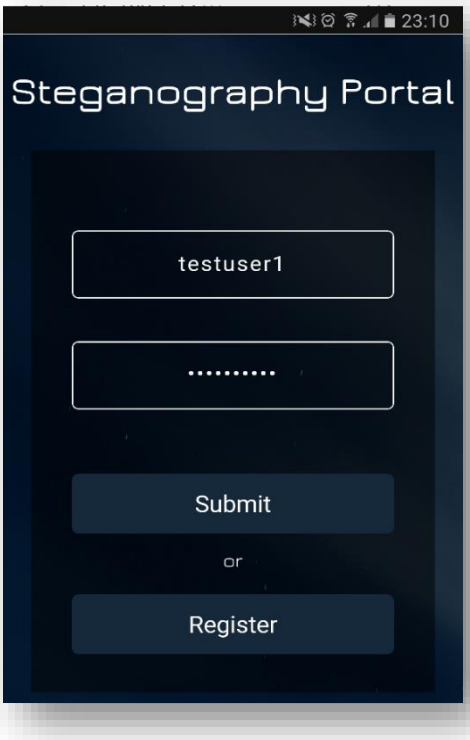
4.3.1.2.1 Οθόνες Χρήσης Εφαρμογής



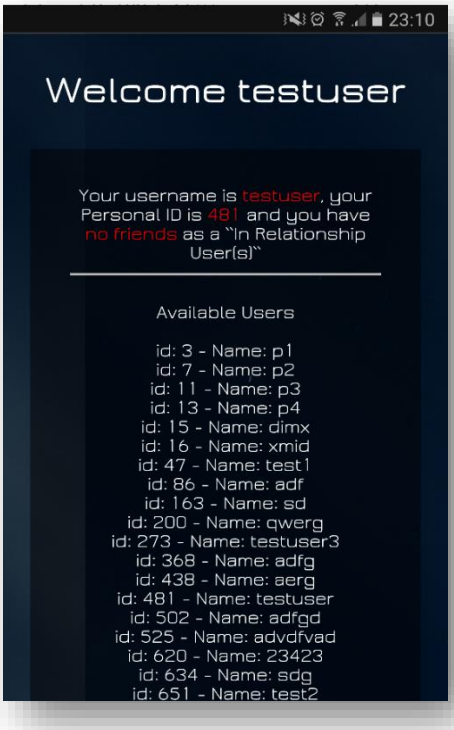
Εικόνα 16, Εισόδου Web Based εφαρμογής



Εικόνα 17, Εγγραφή νέου χρήστη

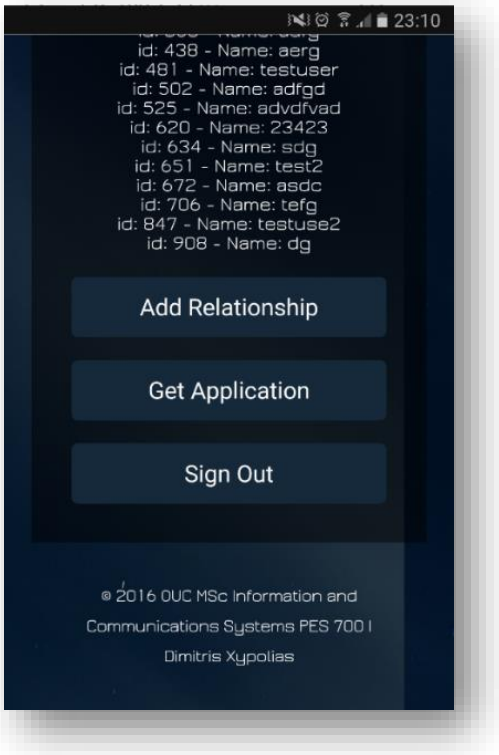


Εικόνα 18, Εισόδου testuser1, υφιστάμενου χρήστη.

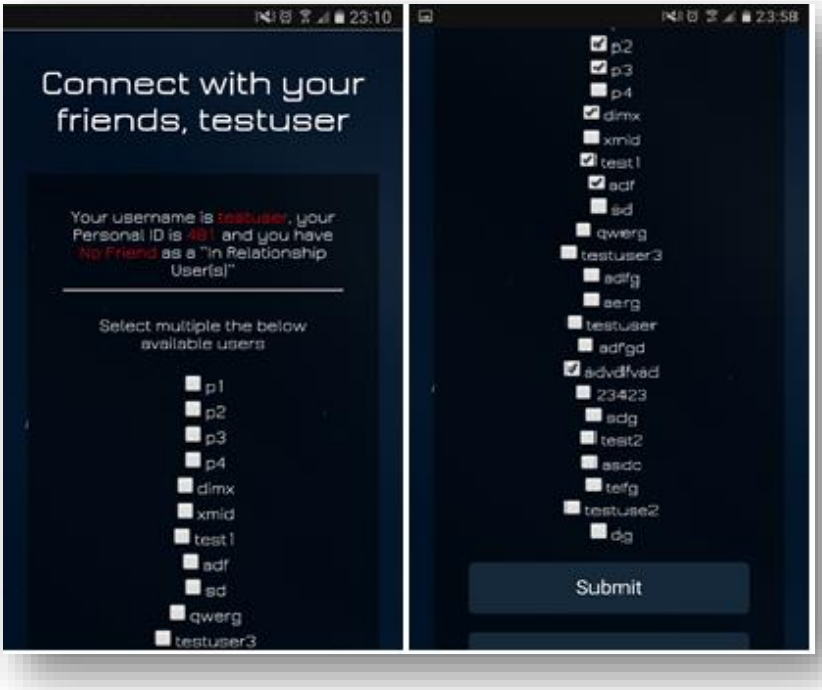


Εικόνα 19, Είσοδος χρήστη και πληροφορίες.

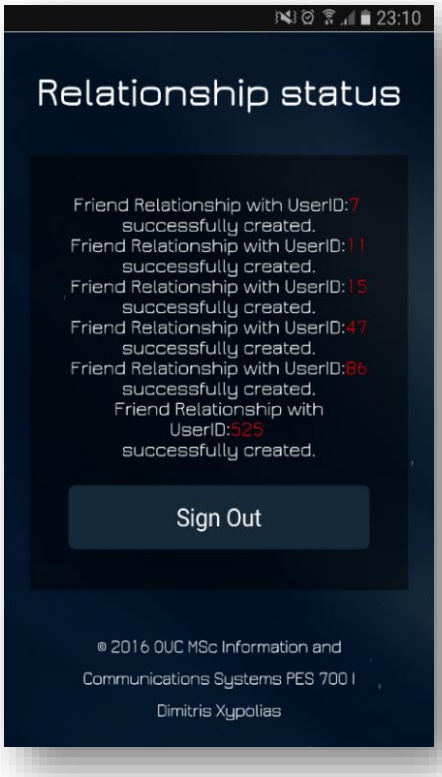
Παρακάτω μπορούμε να δούμε τις επιλογές που εχει ο συνδεδεμένος χρήστης στην δικτυακή εφαρμογή Steganography Portal. Δίνεται η δυνατότητα να δημιουργήσει νεα φιλία αλλα και να κάνει λήψη της mobie εφαρμογή. Είναι κατανοητό πως η δημιουργία νέου χρήστη αλλα και η συναψη φιλίας με τουλαχιστον έναν διαφορετικό χρήστη είναι απαραίτητη προυπόθεση για την εναρξη της διαδικασίας αποστολής ενός κρυφού μυνηματος μεσω της στεναγροφαμηνένης φωτογραφίας.



Εικόνα 20, Επιλογές σελίδας Welcome.



Εικόνα 21, Δυνατότητα Relationship.



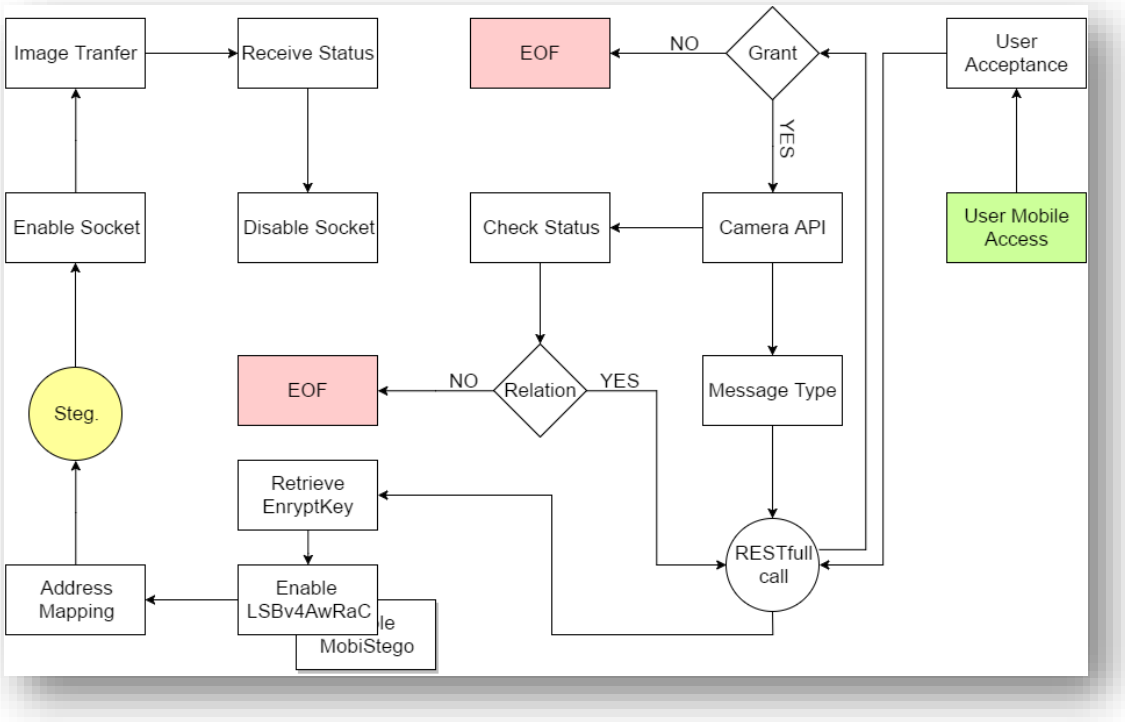
Εικόνα 22, Προσθήκη επαφής με άλλους χρήστες.

4.3.2 Mobile Εφαρμογή

Η mobile εφαρμογή που αναπτύξαμε ονομάζεται mobile_app και έχει τις ακόλουθες δυνατότητες:

1. Επικοινωνία με jee_app
    a. Ταυτοποίηση χρήστη
    b. Εύρεση επαφών χρήστη
    c. Ενημέρωση διεύθυνσης χρήστη
    d. Εύρεση διεύθυνσης χρήστης
    e. Ανάκτηση κρυπτογραφικού κλειδιού
2. Λήψη φωτογραφιών
3. Εμφάνιση φωτογραφιών
4. Αποστολή φωτογραφιών
5. Αποθήκευση φωτογραφιών
6. Διάβασμα φωτογραφιών
7. Εισαγωγή κρυφού μηνύματος σε φωτογραφία
8. Εξαγωγή κρυφού μηνύματος σε φωτογραφία

Στο παρακάτω διάγραμμα παρουσιάζεται η ροή εργασίας της mobile εφαρμογής που υλοποιήθηκε.



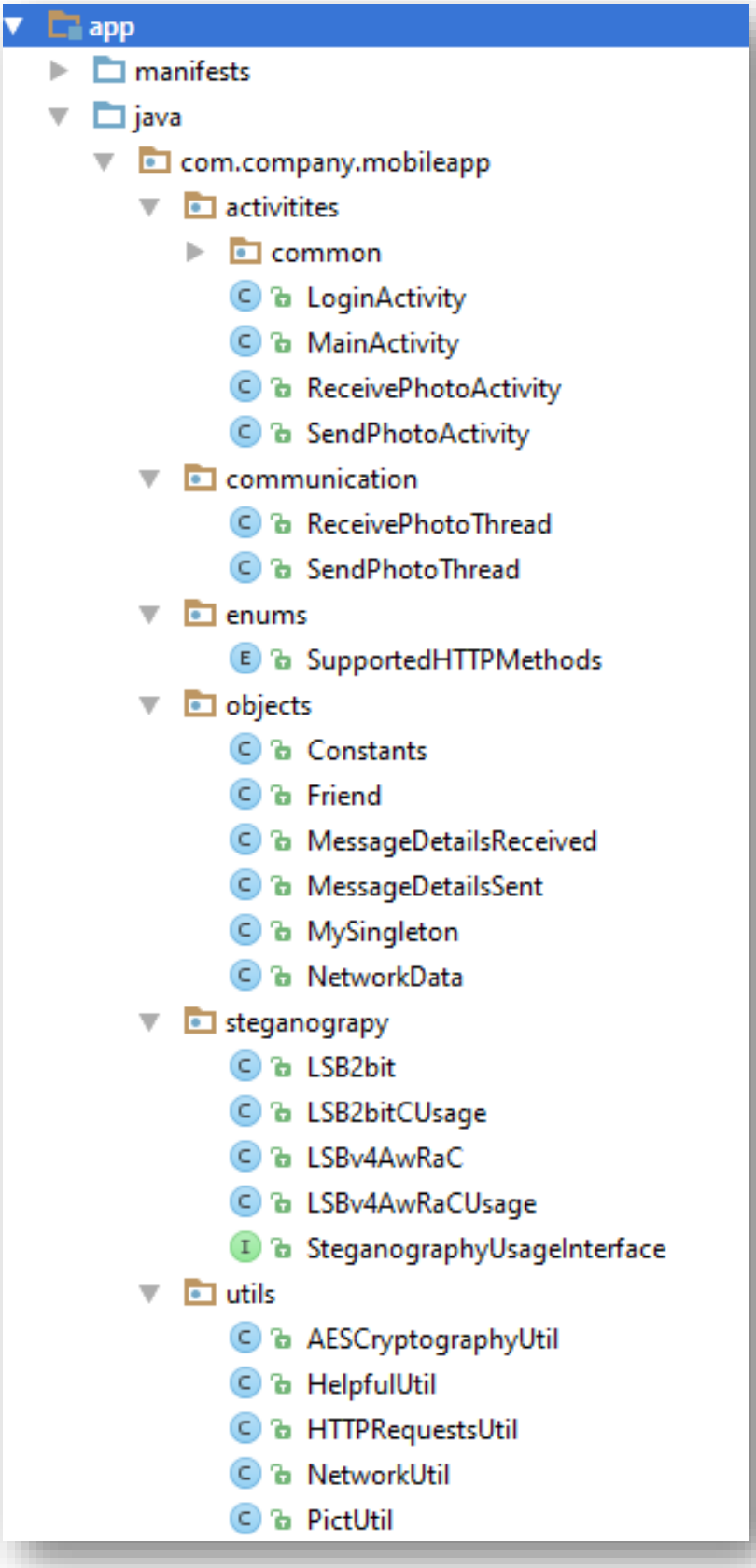
Εικόνα 23, Ροή εργασίας mobile εφαρμογής

Για να μπορέσει να υλοποιηθεί η εφαρμογή γρήγορα και αποτελεσματικά έχουμε κάνει τις ακόλουθες παραδοχές:

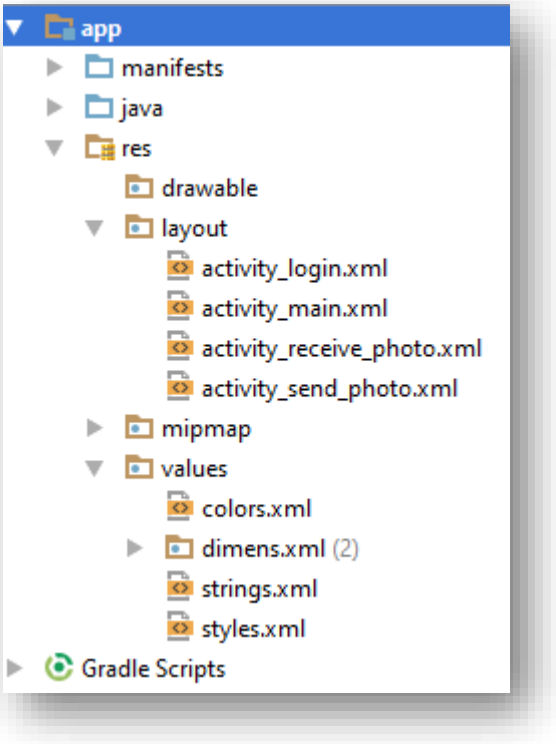
- Επειδή το σύνολο των εφαρμογών και λογισμικών που χρησιμοποιούμε μπορεί να τρέξουν σε διαφορετικά δίκτυα, ο χρήστης της Android εφαρμογής πρέπει κατά την ταυτοποίησή του να εισάγει την διεύθυνση του ΕΥΔΧΚ.
- Δεν έχουμε υλοποιήσει κάποιο service στην Android εφαρμογή που να περιμένει να λάβει στεγανογραφημένες εικόνες από επαφές. Αντί για κάποιο service ο χρήστης μπορεί να επιλέξει (πατώντας αντίστοιχο κουμπί) να βάλει την Android εφαρμογή σε κατάσταση αναμονή λήψης στεγανογραφημένων εικόνων.

4.3.2.1 Διάρθρωση Source Packages & Resources

Στις Εικόνα 24 και Εικόνα 25 παρουσιάζεται η διάρθρωση των source packages και των resources αντίστοιχα της mobile εφαρμογής.



Εικόνα 24, Διάρθρωση source packages mobile εφαρμογής



Εικόνα 25, Διάρθρωση resources mobile εφαρμογής

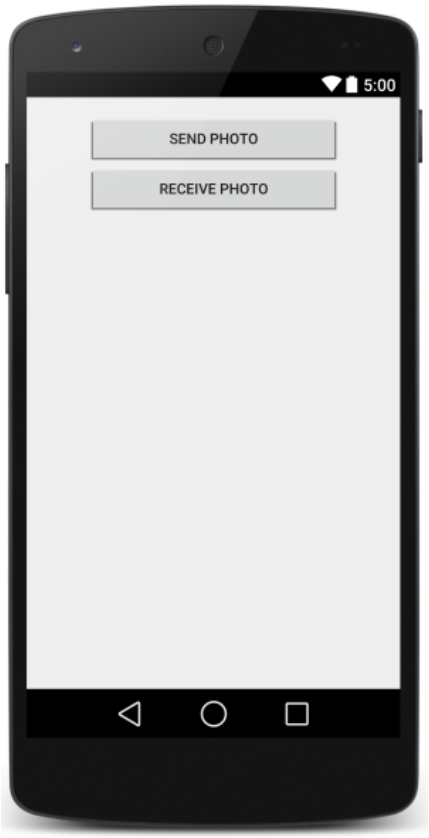
4.3.2.2 Οθόνες Εφαρμογές

Στην mobile εφαρμογή που υλοποιήθηκε και τρέχει σε Android συσκευές δημιουργήθηκαν τις ακόλουθες τέσσερις (4) οθόνες:

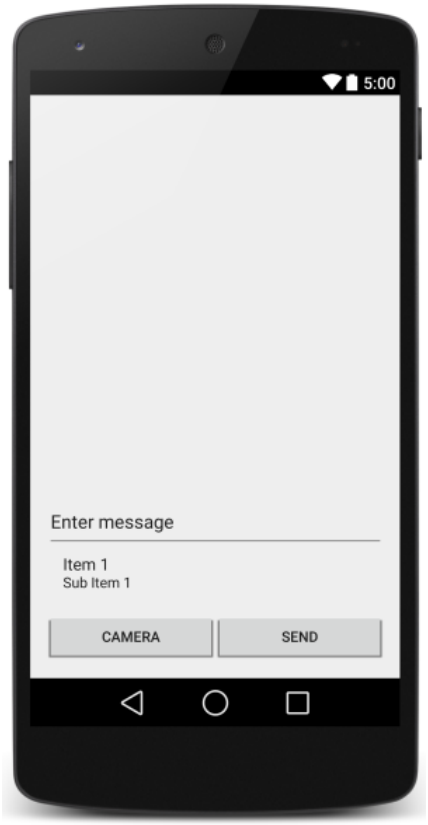
1. **Οθόνη ταυτοποίησης χρήστη** (Εικόνα 26): Ο χρήστης εισάγει username και password για να ταυτοποιηθεί στην mobile εφαρμογή. Η mobile εφαρμογή για να ταυτοποιήσει τον χρήστη καλεί το αντίστοιχο RESTful Service από τον Πίνακας 2. Όπως έχουμε αναφέρει και πρωτύτερα ο χρήστης πρέπει να βάλει και την διεύθυνση του ΕΥΔΧΚ.
2. **Οθόνη επιλογής ενέργειας χρήστη** (Εικόνα 27): Ο χρήστης επιλέγει την ενέργεια που θέλει να κάνει.
3. **Οθόνη αποστολής φωτογραφίας** (Εικόνα 28): Ο χρήστης σε αυτή την οθόνη αλληλεπιδρά με την mobile εφαρμογή ως εξής:
    a. Τραβάει μια φωτογραφία πατώντας το κουμπί με το λεκτικό «CAMERA».
    b. Πληκτρολογεί το μήνυμα που θέλει να εισάγει με στεγανογραφία στην εικόνα στην θέση του λεκτικού «Enter message».
    c. Επιλέγει την επαφή που θέλει να στείλει την στεγανογραφημένη εικόνα από την πτυσσόμενη λίστα «Item 1».
    d. Αποστέλλει την στεγανογραφημένη εικόνα πατώντας του κουμπί «SEND».

4. **Οθόνη λήψης φωτογραφίας** (Εικόνα 29): Ο χρήστης σε αυτή την οθόνη δεν αλληλεπιδρά με την mobile εφαρμογή. Η mobile εφαρμογή είναι σε κατάσταση αναμονής για να λάβει στεγανογραφημένες εικόνες. Η mobile εφαρμογή όταν λαμβάνει μια εικόνα τότε:
    a. Εμφανίζει την εικόνα στον χώρο πάνω από τις ετικέτες που έχουν το λεκτικό «Waiting…».
    b. Στην θέση του πρώτης ετικέτας με το λεκτικό «Waiting…» κάτω από την οθόνη εμφανίζει το όνομα της επαφής που έστειλε την φωτογραφία.
    c. Στην θέση της δεύτερης ετικέτας με το λεκτικό «Waiting…» (στο κάτω μέρος της οθόνης) εμφανίζει το μήνυμα που εισήγαγε με στεγανογραφία η επαφή στην εικόνα.

Εικόνα 26, Οθόνη ταυτοποίησης



Εικόνα 27, Οθόνη επιλογής ενέργειας χρήστη

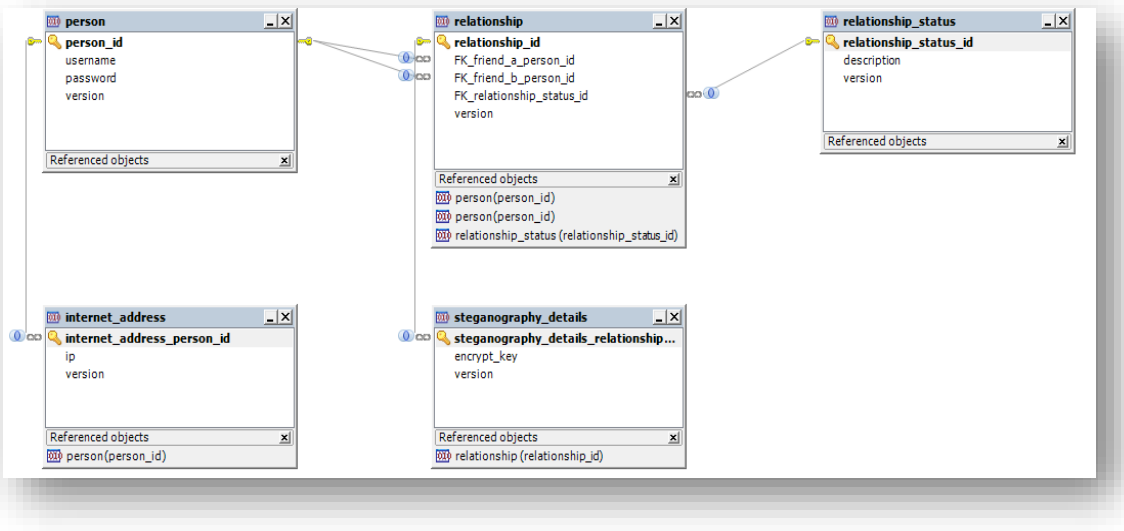


Εικόνα 28, Οθόνη αποστολής φωτογραφίας

Εικόνα 29, Οθόνη λήψης φωτογραφίας

4.3.3 Δημιουργία Βάσης Δεδομένων

Για την υποστήριξη του συστήματος δημιουργήθηκε βάση δεδομένων με βάση το E-R διάγραμμα της § 3.4.2 και όνομα project_db. Για την δημιουργία της βάσης δεδομένων χρησιμοποιήσαμε τα SQL scripts που υπάρχουν στα παραδοτέα. Στην Εικόνα 30 παρουσιάζουμε την βάση δεδομένων από το εργαλείο Toad αμέσως μετά την δημιουργία της.



Εικόνα 30, Διάγραμμα Βάσης Δεδομένων

4.4 Λογισμικά Υλοποίησης

Με τον όρο λογισμικά αναφερόμαστε στο σύνολο των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη, δοκιμή και εκτέλεση των συστατικών της συνολικής λύσης. Τα λογισμικά που χρησιμοποιήθηκαν για την ανάπτυξη της συνολικής λύσης είναι open source και ως εκ' τούτου δεν υπάρχει κάποιο κόστος. Βασιστήκαμε σε λογισμικά που είναι διεθνώς αναγνωρισμένα και έχουν αποδοχή από την κοινότητα (community) των προγραμματιστών. Αυτό θα μας βοηθήσει στην επίλυση τυχόν προβλημάτων που θα προκύψουν κατά την λειτουργία κάποιου ή κάποιων εκ των συστατικών (components) της συνολικής λύσης σε ολόκληρο τον κύκλο ζωής της προτεινόμενης λύσης.

1. **IDEs**: Για την ανάπτυξη των εφαρμογών χρησιμοποιήθηκαν τα ακόλουθα δύο εργαλεία:
    a. **Android Studio**: για την ανάπτυξη της mobile εφαρμογής
    b. **Netbeans**: για την ανάπτυξη της JEE εφαρμογής
2. **RDBMS**: Το RDBMS εγκαταστάθηκε χρησιμοποιώντας την εφαρμογή **MySQL Installer** της Oracle (mysql-installer-community-5.7.10.0.msi). Η έκδοση του **MySQL Server** που εγκαταστάθηκε είναι η **5.7.10**.

    Για την διαχείριση της βάσης δεδομένων χρησιμοποιήθηκε το εργαλείο **Toad for MySQL**. Προτιμήθηκε αυτό το εργαλείο έναντι του εργαλείου που προσφέρεται από την Oracle (MySQL Workbench) λόγω των πολλών περισσότερων δυνατοτήτων που προσφέρει. Επίσης η χρήση του Open Source εργαλείου **HeidiSQL** είναι μια εναλλακτική πρόταση για γρήγορη και εύκολη παραμετροποίηση της βάσης δεδομένων της συγκεκριμένης μεταπτυχιακής διατριβής.

3. **ΕΥΔΧΚ**: Ο ΕΥΔΧΚ εγκαταστάθηκε χρησιμοποιώντας την εφαρμογή **JBoss EAP 6.4.0 Installer** που παρέχει το Red Hat Developers (jboss-eap-6.4.0-installer.jar). Η έκδοση του **JBoss EAP** που εγκαταστάθηκε είναι η **6.4.0**.
4. **Δοκιμές**: Για να μπορέσουμε να δοκιμάσουμε τα HTTP calls που γίνονται στην εφαρμογή jee_app που τρέχει στον ΕΥΔΧΚ εγκαταστήσαμε το εργαλείο **Postman**.

5 Πειραματικά Αποτελέσματα

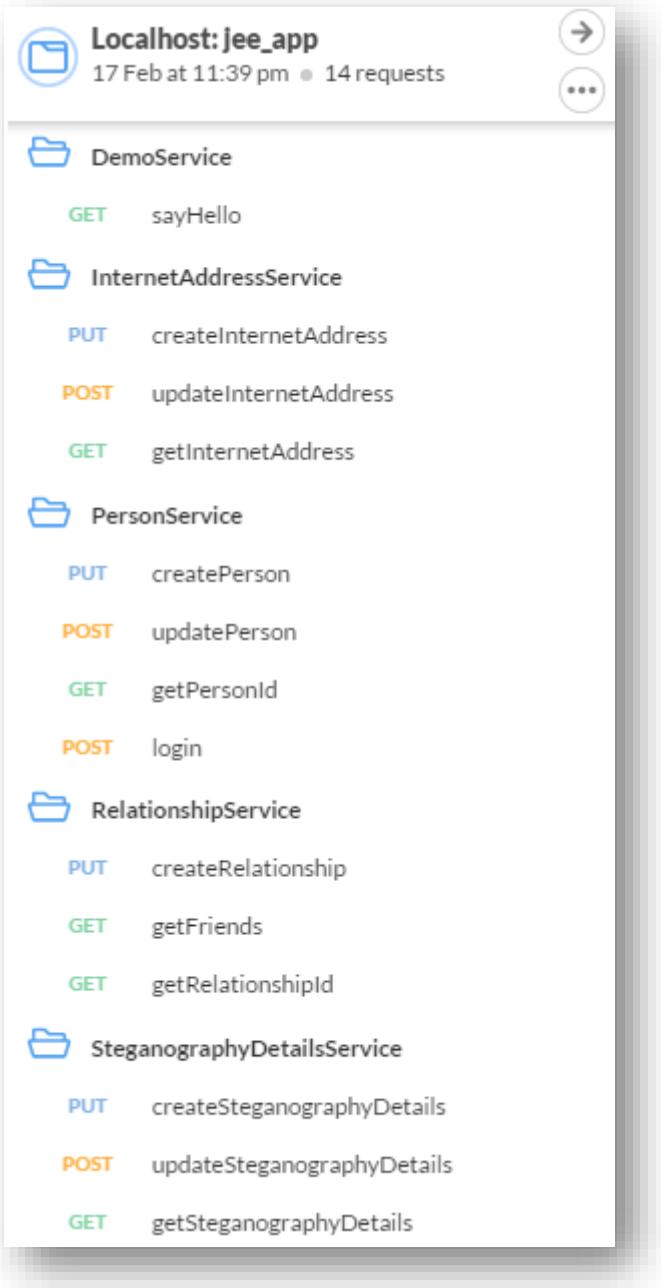
Το σύστημα που υλοποιήθηκε στα πλαίσια της παρούσας μεταπτυχιακής διατριβής, υποβλήθηκε στις παρακάτω δοκιμασίες ελέγχου.

5.1 Unit Testing

Κατά την υλοποίηση οποιασδήποτε καινούργιας λειτουργικότητας σχεδιάσαμε και υλοποιήσαμε τα αντίστοιχα unit tests, έτσι ώστε να διαπιστώσουμε ότι η καινούργια λειτουργικότητα έχει την επιθυμητή συμπεριφορά.

Σε ότι αφορά την εφαρμογή που τρέχει στον ΕΥΔΧΚ (jee_app) χρησιμοποιήσαμε τα frameworks **JUnit**, **Arquilian** & **REST Assured** (Massol and Husted , 2003) (Siikarla, et al. , 2008). Λόγω της χρήσης του Postman για την εκτέλεση δοκιμών στην εφαρμογή jee_app, δημιουργήσαμε JUnit tests μόνο για το Rest Service Person Service, ως επίδειξη της σχετικής λειτουργικότητας.

Στο Postman δημιουργήσαμε μία συλλογή (collection) με όνομα Localhost: jee_App στην οποία συμπεριλάβαμε όλες τις δυνατές κλήσεις στην εφαρμογή jee_app. Στα παραδοτέα της διατριβής έχουμε συμπεριλάβει αυτή την συλλογή, στην περίπτωση που κάποιος θέλει να την χρησιμοποιήσει για να δοκιμάσει την εφαρμογή jee_app. Στην Εικόνα 31 παρουσιάζουμε την διάρθρωση αυτής της συλλογής.

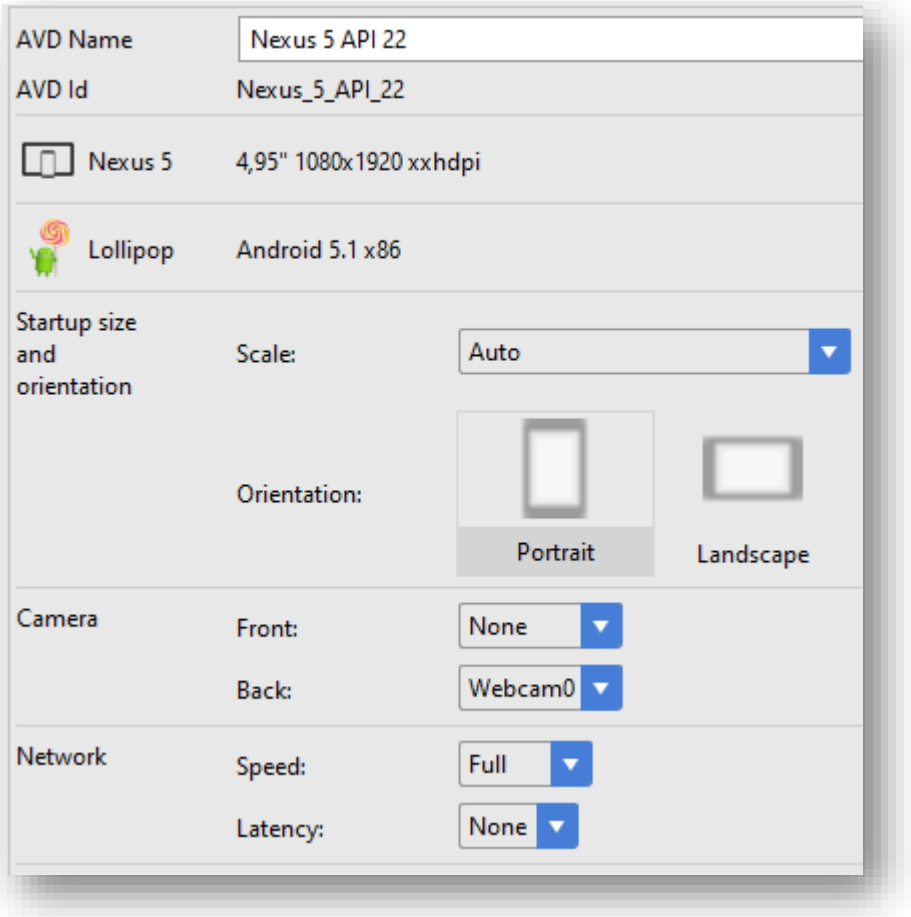


Εικόνα 31, Διάρθρωση συλλογής Localhost: jee_App του Postman

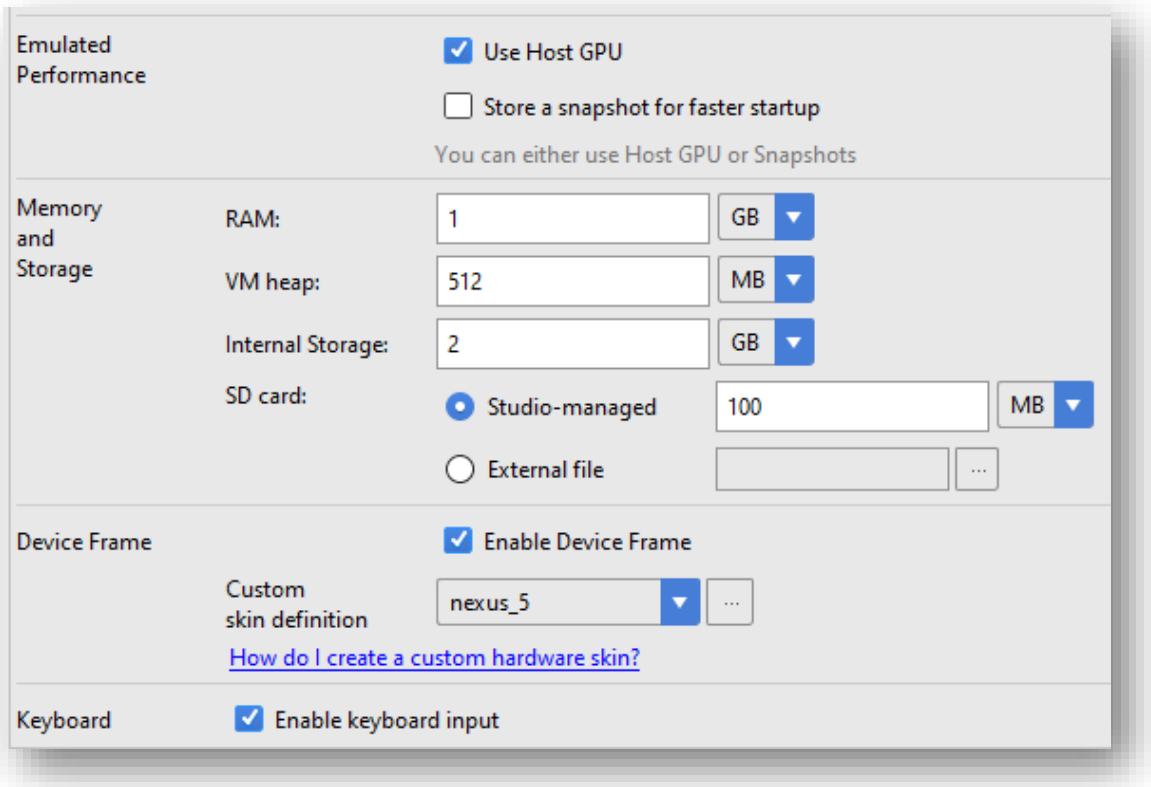
Η mobile εφαρμογή (mobile_app) δοκιμαζόταν κατά την υλοποίησή της σε ένα AVD που με τα χαρακτηριστικά που περιγράφονται στις Εικόνα 32 και Εικόνα 33. Πρέπει να σημειώσουμε ότι δεν μπορέσαμε να χρησιμοποιήσουμε το AVD για να δοκιμάσουμε το σύνολο της λειτουργικότητας, γιατί υπάρχουν οι ακόλουθοι περιορισμοί:

1. Το εκάστοτε AVD εκτελείται σε ιδιωτικό δίκτυο με το workstation που το φιλοξενεί και δεν μπορεί να δει άλλα AVDs που τρέχουν στο ίδιο workstation. (για να εκτελεστούν πολλαπλά AVDs απαιτείται haxm intel παραμετροποίηση σε υπολογιστές με CPU Intel).

2. Τα AVDs δεν υποστηρίζουν WiFi δικτύωση και οι IP διευθύνσεις που λαμβάνουν είναι ίδιες, πράγμα που καθιστά αδύνατη την λειτουργία της εφαρμογής (αποστολή/λήψη υλικού).
3. Ορισμένες φορές δεν είναι δυνατό το AVD να πάρουμε φωτογραφίες από το AVD (χρησιμοποιώντας είτε την web camera του workstation είτε emulated camera).



Εικόνα 32, Χαρακτηριστικά AVD (1)



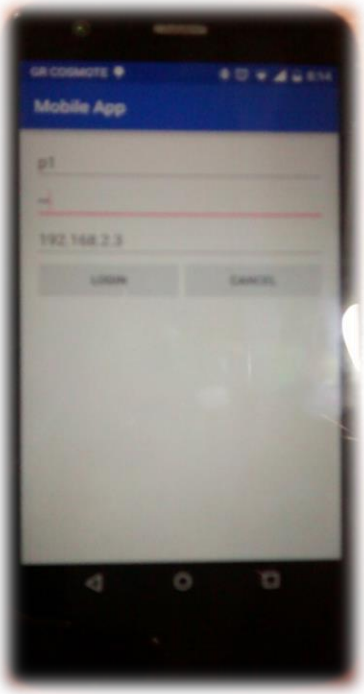
Εικόνα 33, Χαρακτηριστικά AVD (2)

5.2 User Acceptance Testing

Για να μπορέσουμε να εκτελέσουμε UAT εισήγαμε στην βάση δεδομένων χρησιμοποιώντας το POSTMAN δύο χρήστες (**χρήστης Α** & **χρήστης Β**).

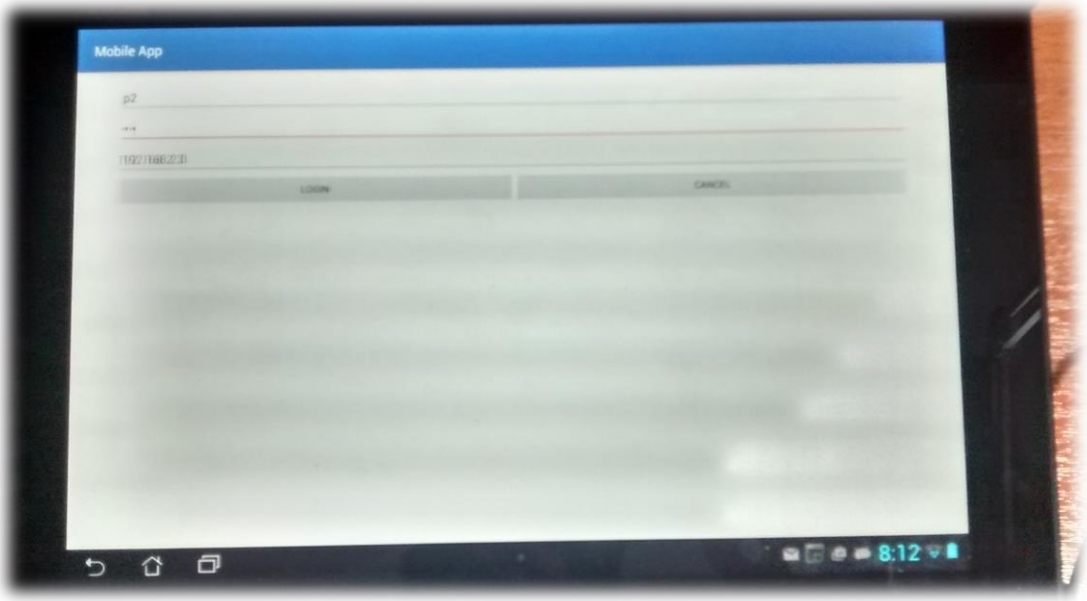
Εκτελέστηκε το ακόλουθο σενάριο με επιτυχία χρησιμοποιώντας δύο Android συσκευές (**Android συσκευή Α** & **Android συσκευή Β**):

1. Ο **χρήστης Α** ταυτοποιήθηκε στην **Android συσκευή Α**



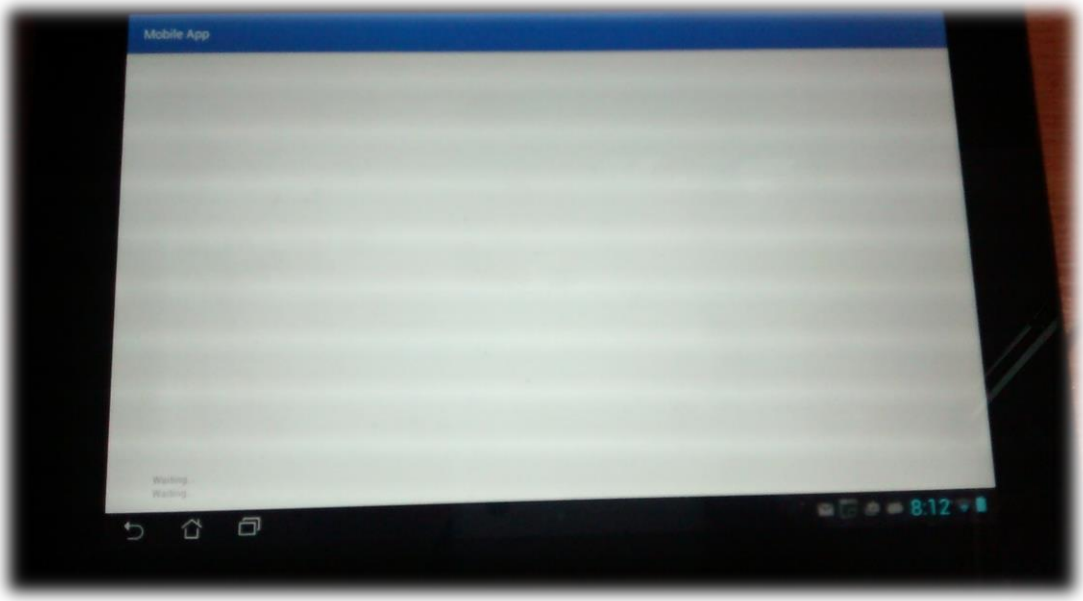
Εικόνα 34, Ταυτοποίηση χρήστη Α

2. Ο **χρήστης Β** ταυτοποιήθηκε στην **Android συσκευή Β**



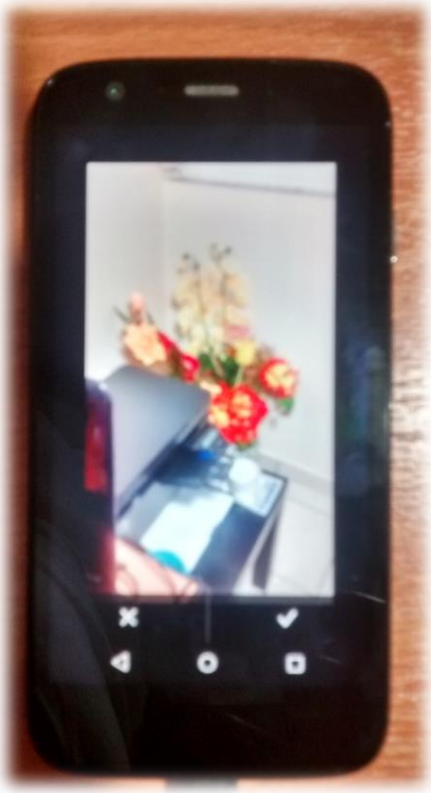
Εικόνα 35, Ταυτοποίηση χρήστη Β

3. Ο **χρήστης Β** έθεσε την **Android συσκευή Β** σε θέση αναμονής για λήψη στεγανογραφημένων εικόνων



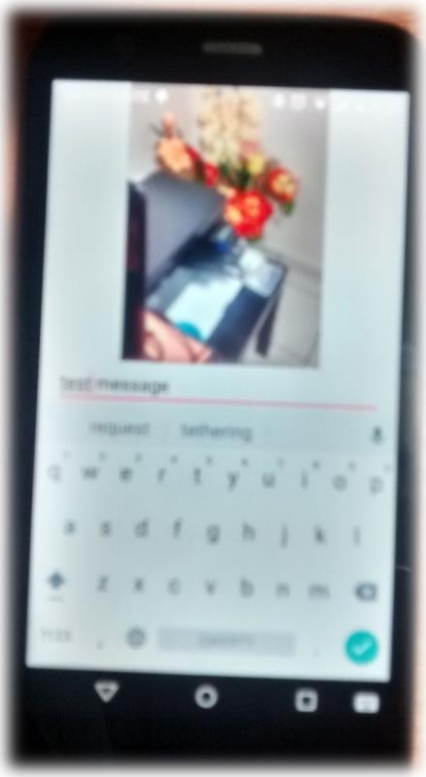
Εικόνα 36, Η Android συκσευή Β είναι θέση αναμονής για λήψη στεγανογραφημένων φωτογραφιών

4. Ο **χρήστης Α**:
    a. Τραβάει μία εικόνα



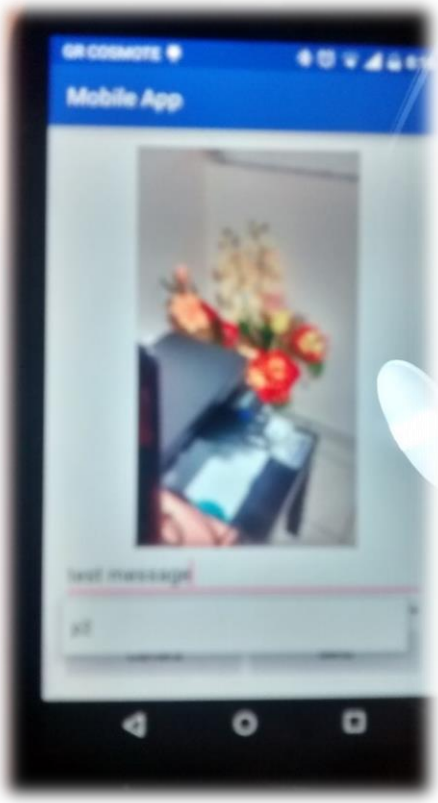
Εικόνα 37, Ο χρήστης Α έχει μόλις τραβήξει μία φωτογραφία

    b. Εισάγει ένα κρυπτογραφημένο μήνυμα



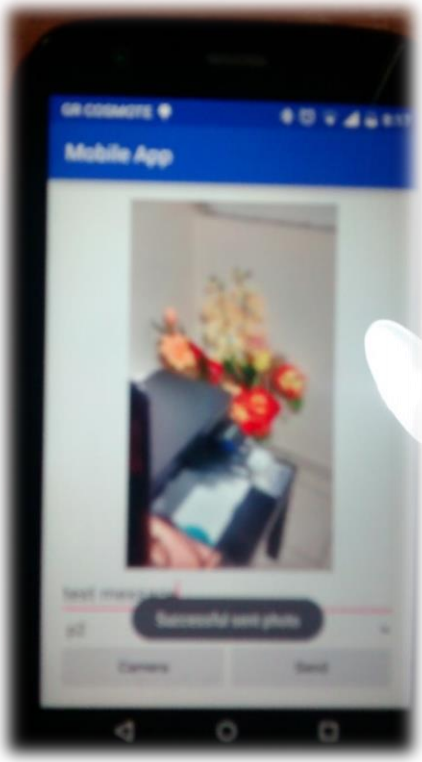
Εικόνα 38, Ο χρήστης Α πληκτρολογεί ένα μήνυμα

c. Επιλέγει σαν παραλήπτη τον **χρήστη Β**



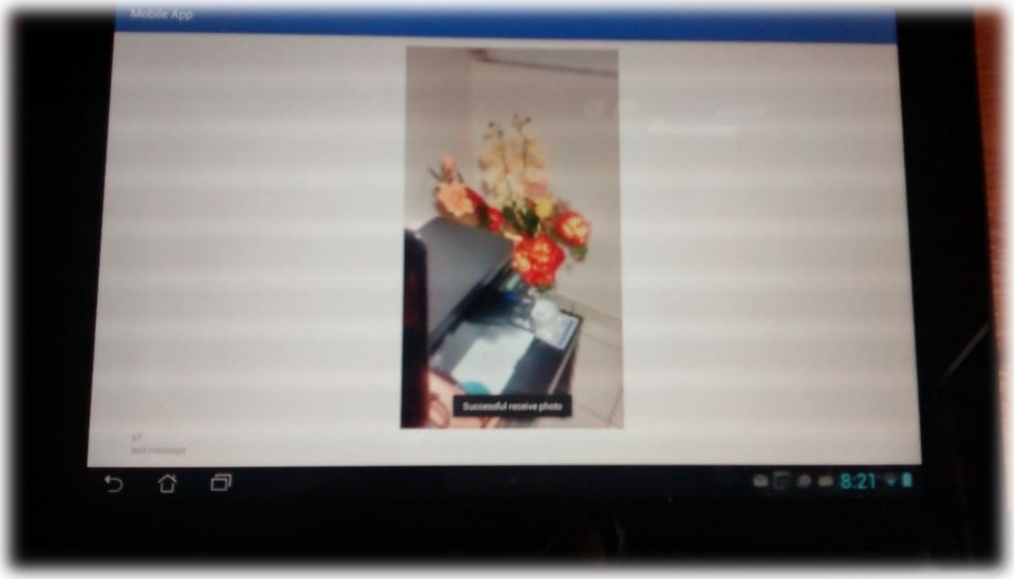
Εικόνα 39, Ο χρήστης Α επιλέγει επαφή για αποστολή στεγανογραφημένης εικόνας

d. Στέλνει την στεγανογραφημένη εικόνα



Εικόνα 40, Ο χρήστης Α αποστέλλει την στεγανογραφημένη εικόνα

5. Η **Android συσκευή Β** λαμβάνει την στεγανογραφημένη εικόνα και κάνει τις ακόλουθες ενέργειες:
    a. Εμφανίζει την στεγανογραφημένη εικόνα
    b. Εμφανίζει το μήνυμα που έγραψε ο **χρήστης Α**
    c. Εμφανίζει το όνομα του **χρήστη Α**



Εικόνα 41, η Android συσκευή εμφανίζει μία στεγανογραφημένη εικόνα

5.3 Μετρήσεις

Στα πλαίσια των πειραματικών δοκιμασιών του προτεινόμενου αλγορίθμου, πραγματοποιήθηκαν μετρήσεις ταχύτητας, ποιότητας, κατανάλωσης επεξεργαστικής ισχύος μέσω του συστήματος που υλοποιήθηκε στα πλαίσια της παρούσας διατριβής.

5.3.1 Ταχύτητα

Για να μπορέσουμε να μετρήσουμε την ταχύτητα του αλγορίθμου στεγανογραφίας που υλοποιήσαμε πραγματοποιήθηκαν δέκα (10) μετρήσεις χρησιμοποιώντας δύο (2) Android συσκευές:

- Για την αποστολή στεγανογραφημένων φωτογραφιών χρησιμοποιήθηκε μία Android συσκευή XT1032 (Android Smartphone Motorola Moto G 1st Generation)
- Για την αποστολή στεγανογραφημένων φωτογραφιών χρησιμοποιήθηκε μία Android συσκευή ASUS Transformer Pad TF300T (Android Tablet)

Στο σημείο αυτό πρέπει να επισημάνουμε ότι κάθε Android συσκευή εκτελεί στο παρασκήνιο και άλλες λειτουργίες. Αυτό έχει σαν αποτέλεσμα να υπάρχει πιθανότητα εκχώρησης της CPU και σε άλλες λειτουργίες κατά την διάρκεια λήψης των μετρήσεων. Μία τέτοια εκχώρηση της CPU θα οδηγήσει σε αύξηση των «καθαρών μετρήσεων» χρόνου. Λόγω της φύσης του Android δεν μπορούμε να πάρουμε απόλυτα αξιόπιστες μετρήσεις, εκτός και αν χρησιμοποιήσουμε Android συσκευές με τα ακόλουθα χαρακτηριστικά:

1. Δεν είναι συνδεδεμένες σε δίκτυο τηλεφωνίας
2. Δεν έχουν καμία εγκατεστημένη εφαρμογή πέρα από τις default και την δικιά μας
3. Δεν έχουν συνδεθεί σε κανένα account (Google, Yahoo, κλπ)
4. Δεν τρέχει καμία λειτουργία στο παρασκήνιο
5. Είναι ακριβώς ίδιες σε hardware και software (έκδοση Android OS που τρέχουν)

Επιπλέον, η διασύνδεση των Android συσκευών με workstation (μέσω ADB) ενδέχεται να επηρεάζει τα αποτελέσματα των μετρήσεων.

Όλες οι μετρήσεις πάρθηκαν χρησιμοποιώντας την ίδια φωτογραφία (Εικόνα 42) και το ίδιο μήνυμα (this is a test message for steg). Η φωτογραφία που παρουσιάζεται στην Εικόνα 42 είναι η στεγανογραφημένη εικόνα όπως λήφθηκε και αποθηκεύτηκε στο κινητό της επαφής.

Εικόνα 42, Φωτογραφία Μετρήσεων

Οι μετρήσεις που κάναμε παρουσιάζονται στον Πίνακας 3 και αφορούν μόνο το χρόνο που χρειάζεται η εφαρμογή:

1. να δημιουργήσει μία στεγανογραφημένη εικόνα και
2. να διαβάσει το μήνυμα από μία στεγανογραφημένη εικόνα.

Οι μετρήσεις γίνονται από κώδικα που έχει γραφεί μέσα στην εφαρμογή και οι τιμές εμφανίζονται στο logcat του tab Android Monitor του Android Studio ως εξής:

- Για την ενσωμάτωση του μηνύματος σε μία φωτογραφία ο κώδικας είναι:

```
Log.i(TAG, "Elapsed milliseconds for encoding: " + difference);
```

Η εγγραφή στο logcat έχει την ακόλουθη μορφή:

```
04-17 20:03:46.196 22464-22464/com.company.mobileapp
I/com.company.mobileapp.activitites.SendPhotoActivity: Elapsed milliseconds
for encoding: 106
```

- Για το διάβασμα του μηνύματος σε μία στεγανογραφημένη φωτογραφία ο κώδικας είναι:

```
Log.i(TAG, "Elapsed milliseconds for decoding: " + difference);
```

Η εγγραφή στο logcat έχει την ακόλουθη μορφή:

```
04-17 20:03:48.752 7080-7080/com.company.mobileapp
I/com.company.mobileapp.communication.ReceivePhotoThread: Elapsed
milliseconds for decoding: 251
```

Οι μετρήσεις έγιναν τόσο χρησιμοποιώντας τον αλγόριθμο που αναπτύξαμε (LSBv4AwRaC), όσο και χρησιμοποιώντας τον αλγόριθμο LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego (Pasquale , 2016).

# Μέτρηση	**Δημιουργία εικόνας**		**Διάβασμα μηνύματος**	
Αλγόριθμος	***LSBv4AwRaC***	***MobiStego***	***LSBv4AwRaC***	***MobiStego***
1	115 ms	104 ms	236 ms	20 ms
2	134 ms	75 ms	167 ms	23 ms
3	81 ms	75 ms	251 ms	33 ms
4	206 ms	117 ms	102 ms	15 ms
5	75 ms	112 ms	100 ms	23 ms
6	75 ms	112 ms	110 ms	25 ms
7	82 ms	62 ms	196 ms	33 ms
8	85 ms	72 ms	230 ms	52 ms
9	115 ms	81 ms	114 ms	19 ms
10	63 ms	60 ms	248 ms	14 ms

Πίνακας 3, Μετρήσεις ταχύτητας

Με βάση τις παραπάνω μετρήσεις που καταγράφηκαν συμπεραίνουμε τα ακόλουθα για τον αλγόριθμο LSBv4AwRaC:

1. Ο χρόνος δημιουργίας μιας στεγανογραφημένης εικόνας ποικίλει μεταξύ 63 ms και 206 ms.
2. Ο χρόνος διαβάσματος ενός μηνύματος από μία στεγανογραφημένη εικόνα ποικίλει μεταξύ 100 ms και 251 ms.

Με βάση τις μετρήσεις που έγιναν για τον αλγόριθμο LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego συμπεραίνουμε τα ακόλουθα:

1. Ο χρόνος δημιουργίας μιας στεγανογραφημένης εικόνας ποικίλει μεταξύ 60 ms και 104 ms.
2. Ο χρόνος διαβάσματος ενός μηνύματος από μία στεγανογραφημένη εικόνα ποικίλει μεταξύ 14 ms και 52 ms.

Συγκρίνοντας τις δύο υλοποιήσεις διαπιστώνουμε ότι ο αλγόριθμος LSB που έχει υλοποιηθεί στην Android εφαρμογή MobiStego είναι γρηγορότερος από τον

αλγόριθμο μας LSBv4AwRaC. Αυτό είναι λογικό και αναμενόμενο λόγω της τυχαιότητας και της κρυπτογραφίας που έχουμε υλοποιήσει στον αλγόριθμο LSBv4AwRaC, παρατήρηση που μας οδηγεί στο συμπέρασμα ότι η επιπλέον ασφάλεια που προσφέρει ο αλγόριθμος LSBv4AwRaC έχει επίπτωση στην ταχύτητα.

5.3.2 Ποιότητα

Για την μέτρηση της ποιότητας της στεγανογραφημένης εικόνας χρησιμοποιήθηκε το εργαλείο ImageMagick (Cristy , 1990). Πρόκειται για ένα πανίσχυρο Command Line εργαλείο διαθέσιμο υπό GPL άδεια χρήσης το οποίο είναι ικανό να μας δώσει συνοπτική και περιγραφική πληροφορία σχετικά με την ποιότητα των φωτογραφιών στις οποίες εφαρμόσαμε τον αλγόριθμο στεγανογράφησης. Η έξοδος (output) του προγράμματος βασίζεται στην τεχνική PSNR δίνοντάς μας πληροφορίες για τέσσερες βασικές μονάδες σύγκρισης:

- σφάλμα απόκλισης τετραγώνου (MSE),
- τυπική απόκλιση (standard deviation),
- κύρτωση (kurtosis),
- ασυμμετρία (skewness) και
- εντροπία (entropy)

Στη συνέχεια παρουσιάζονται οι στατιστικές έννοιες που χρησιμοποιούνται ως βασικές μονάδες σύγκρισης ποιότητας μεταξύ των εικόνων.

Το **σφάλμα απόκλισης τετραγώνου** (MSE) μετρά το μέσο όρο των τετραγώνων των σφαλμάτων ή αποκλίσεων στο οποίο διαφαίνεται η διαφορά μεταξύ της εκτιμήτριας τιμής και της τιμής που τελικά εκτιμάται. Το εργαλείο ImageMagick αναλύοντας την τεχνική PSNR χρησιμοποιεί το κριτήριο του μέσου τετραγωνικού σφάλματος (mean square error) της εκτίμησης ως το άθροισμα της διασποράς του εκτιμητή και του τετραγώνου της μεροληψίας.

Η **τυπική απόκλιση** (standard deviation) είναι ένα μέτρο που χρησιμοποιείται για να υπολογιστεί το ποσό της μεταβολής ή της διασποράς ενός συνόλου τιμών δεδομένων. Μια χαμηλή τυπική απόκλιση υποδηλώνει ότι τα σημεία των δεδομένων τείνουν να είναι κοντά στο μέσο όρο του συνόλου, ενώ μία υψηλή τυπική απόκλιση υποδεικνύει ότι τα στοιχεία απλώνονται πάνω από ένα ευρύτερο φάσμα των τιμών.

Το εργαλείο ImageMagick που χρησιμοποιήσαμε καταμετρά επίσης, το **τυπικό σφάλμα κύρτωσης** (kurtosis) το οποίο δείχνει την απόκλιση που θα μπορούσε να υπάρχει στις τιμές κύρτωσης μεταξύ πολλαπλών τυχαίων δειγμάτων που θα προερχόντουσαν από την ίδια κατανομή που προήλθε και το δείγμα ανάλυσης. Μία χαμηλή τιμή τυπικού σφάλματος δείχνει ότι η απόκλιση στις τιμές κύρτωσης

μεταξύ πολλαπλών τυχαίων δειγμάτων που θα προερχόντουσαν από την ίδια κατανομή που προήλθε και το δείγμα ανάλυσης είναι μηδενική. Υψηλές τιμές δείχνουν μεγαλύτερη απόκλιση της κατανομής από όπου προήλθε το δείγμα από μία κατανομή με μηδενική κύρτωση.

Αναφορικά με την **ασυμμετρία** (skewness), το εργαλείο ImageMagick δίνει τη δυνατότητα μέτρησης της κατανομής ενός πλήθους ψηφιακών στοιχείων που μπορεί να είναι είτε συμμετρική, είτε μη συμμετρική. Στην πρώτη περίπτωση η μέση τιμή, η διάμεσος και η κορυφή της κατανομής συμπίπτουν. Στις άλλες περιπτώσεις ένα από τα τμήματα στα οποία χωρίζει την κατανομή η κορυφή, περιέχει περισσότερες παρατηρήσεις από το άλλο. Η ασυμμετρία χωρίζεται σε δυο κατηγορίες, την θετική και την αρνητική. Στους ελέγχους ποιότητας εικόνας μέσω της τεχνικής PSNR παρουσιάζεται η αρνητική συμμετρία η οποία υποδηλώνει ότι οι περισσότερες παρατηρήσεις των ψηφιακών στοιχείων που ολοκληρώνουν την ανάλυση μιας εικόνας, όπως και η μέση τιμή τους καθώς και η διάμεσος τους, βρίσκονται αριστερά της κορυφής.

Τέλος, η **εντροπία** (entropy) είναι η έννοια μέσω της οποίας μετράται η αταξία, η μέγιστη τιμή της οποίας, αντικατοπτρίζει την πλήρη αποδιοργάνωση (ομογενοποίηση των πάντων). Αρχικά βασίστηκε σαν έννοια στην θερμοδυναμική, πλέον όμως, χρησιμοποιείτε εκτενέστατα στην ανάλυση ψηφιακών αρχείων και στην εκτίμηση συμπίεσης ή κρυπτογράφησης.

Με βάση τις παραπάνω έννοιες και με χρήση του εργαλείου ImageMagick προχωρήσαμε σε μετρήσεις ποιότητας χρησιμοποιώντας το ίδιο αρχείο εικόνας που χρησιμοποιήθηκε στην ενότητα 5.3.1 και συγκεκριμένα την Εικόνα 41.

Οι αρχικές τιμές για το αρχείο stego1.png με την εκτέλεση της εντολής:

```
identify -verbose -features 1 -moments -unique stego1.png
```

είναι οι ακόλουθες:

Μονάδα	Τιμή
MSE	65.758
Std. Deviation	62.3584
Kurtosis	3.48741
Skewness	-0.528865
Entropy	0.922405

Πίνακας 4, Ποιότητα αρχικής εικόνας

Με την εκτέλεση της παραπάνω εντολής στα αρχεία εικόνας που προκύπτουν από την εκτέλεση των δυο συγκρινόμενων αλγορίθμων (LSBv4AwRaC, MobiStego) καταγράφηκαν τα παρακάτω αποτελέσματα:

Μονάδα	**Τιμές**	
Αλγόριθμος	***LSBv4AwRaC***	***MobiStego***
MSE	106.131	109.394
Std. Deviation	72.0052	72.0641
Kurtosis	-1.42776	-1.42776
Skewness	0.0472146	0.0479657
Entropy	0.920125	0.920115

Πίνακας 5, Μετρήσεις ποιότητας εικόνων

Οι τιμές του σφάλματος του τετραγώνου (MSE) των αποκλίσεων αλλά και της ασυμμετρίας (skewness) υποδεικνύουν την μεταβολή στην συνολική μονάδα PSNR (Instruments , 2013).

$$PSNR = 20\log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right)$$

Ο MSE αντιπροσωπεύει το μέσο όρο του τετραγώνου των «λαθών» που παρουσιάζονται στο πλήθος των εικονοστοιχείων μιας εικόνας.

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\|f(i,j) - g(i,j)\|^2$$

Η τιμή m αντιπροσωπεύει τον αριθμό των σειρών και η τιμή n την τιμή των στηλών των εικονοστοιχείων της εικόνας.

Με βάση τα παραπάνω, η υπόριζη ποσότητα των «λαθών» (MSE) που βρίσκεται στον παρονομαστή του τύπου PSNR καθορίζει την συνολική ποιότητα της εικόνας μας.

Με τη βοήθεια του εργαλείου ImageMagick προκύπτουν οι παραπάνω μετρήσεις για το μέσο τετραγωνικό σφάλμα (MSE) μεταξύ της στεγανογραφημένης εικόνας και της αρχικής εικόνας. Οι μετρήσεις έγιναν για την ίδια αρχική εικόνα με χρήση των δύο αλγορίθμων LSBv4AwRaC και MobiStego. Όπως έχει καταγραφεί στον Πίνακας 5, στην περίπτωση εφαρμογής του προτεινόμενου αλγορίθμου LSBv4AwRaC, η τιμή MSE βρίσκεται σε χαμηλότερα επίπεδα και σε ψηλότερα με την εφαρμογή του MobiStego. Συνεπώς, το συνολικό PSNR της εικόνας που

προέκυψε από στεγανογραφία με τη χρήση LSBv4AwRaC είναι ψηλότερο και η ανάλυση της συγκριτικά καλύτερη(αντίστροφη σχέση μεταξύ του MSE και του PSNR).

Σημαντικό είναι να επισημάνουμε στο σημείο αυτό, πως η μέτρηση MSE της αρχικής εικόνας με τιμή 65.758 είναι αισθητά μικρότερη, αλλά αυτό αποτελεί ξεκάθαρη αποδοχή όταν εφαρμόζεται στεγανογραφία σε αρχεία εικόνας.

Μελετώντας τις παραπάνω μετρήσεις (Πίνακας 5) συμπεραίνουμε πως η παραμετροποίηση σε συνεχόμενα LSB στοιχεία που εφαρμόζει ο MobiStego δημιουργούν μεγαλύτερη αλλοίωση στην τελική εικόνα (χαμηλότερο PSNR). Αυτό είναι πολύ σημαντικό και για την Ασφάλεια, καθώς για μια εικόνα στην οποία γίνεται αντιληπτό πως έχει υποστεί επεξεργασία δημιουργούνται αυτόματα μεγαλύτερες πιθανότητες για υποβολή της σε μεθόδους αποκρυπτογράφησης και στεγανάλυσης, καθιστώντας κατ' επέκταση αυξημένο των κίνδυνο υποκλοπής μηνυμάτων.

Η καλύτερη ποιότητα με χρήση του αλγορίθμου LSBv4AwRaC μπορεί να ερμηνευτεί και με βάση την επεξεργασία των bit planes. Τα bit planes παρουσιάζουν την ακολουθία των bits για κάθε pixel σε ένα bitmap. Ο αριθμός των bits στο κάθε ένα bit plane καθορίζει τον αριθμό των χρωμάτων που εμφανίζονται σε μια εικόνα. Με την χρήση ενός απλού αλγορίθμου LSB όπως είναι ο MobiStego η μεταβολή των λιγότερο σημαντικών στοιχείων είναι σειριακή, δηλαδή από την αρχή των δεδομένων που παρουσιάζουν την εικόνα γίνεται αλλαγή με το στοιχείο που επιθυμούμε να αντικαταστήσουμε για να συνθέσουμε το κρυφό μήνυμά μας και άρα του bitplane. Αυτό έχει ως αποτέλεσμα, μεγάλη αλλοίωση στα πρώτα bitplane8 και λιγότερη στα επόμενα. Η διαφορά αυτή μπορεί να γίνει αισθητή ακόμη και με φυσικό μέσο (μάτι). Αντίθετα, με τον προτεινόμενο αλγόριθμο και τη διάσπαρτη επιλογή των LSBs έχουμε τυχαία και διάσπαρτη αλλαγή των bit planes (Ker , 2007) του αρχείου της εικόνας, με συνέπεια οποιαδήποτε αλλοίωση να γίνεται λιγότερο αντιληπτή λόγω της φαινομενικής ομοιομορφίας.

Συνοψίζοντας, με την υλοποίηση του αλγόριθμου **LSBv4AwRaC** και την διάσπαρτη επέμβαση σε LSB στοιχεία μιας εικόνας, το αποτέλεσμα είναι η μικρότερη αλλοίωσή της.

5.3.3 Επεξεργαστική Ισχύς

Μια από τις σημαντικότερες παραμέτρους στο χώρο των mobile εφαρμογών είναι η κατανάλωση επεξεργαστικής ισχύος των κινητών συσκευών.

Για την ανάλυση του κώδικα που υλοποιήσαμε σε επίπεδο επεξεργαστικής κατανάλωσης της συσκευής παίρνουμε σαν παράμετρο δυο σημαντικές μονάδες καταμέτρησης:

- Τον χρόνο ένταξης «inclusion time» ο οποίος μας παρουσιάζει τον χρόνο λειτουργίας που δαπανήθηκε για την συνολική εκτέλεση ενός block κώδικα συμπεριλαμβανομένου και των objects που έχουν ήδη εκτελεστεί ή βρίσκονται σε αναμονή.
- Τον χρόνο αποκλεισμού «exclusion time» ο οποίος είναι ο καθαρός χρόνος εκτέλεσης ενός object χωρίς την καταμέτρηση των αλληλεξαρτώμενων block κώδικα.

Παρακάτω παρουσιάζονται οι τιμές που καταγράφηκαν με τη βοήθεια του εργαλείου Android Studio Performance Monitor για τις δυο αυτές παραμέτρους ανά object κώδικα που εκτελείται στις διάφορες λειτουργίες, καθώς και το πλήθος εκτέλεσης του (invocation count) σε διάρκεια μιας πλήρης λειτουργίας που περιλαμβάνει τα παρακάτω:

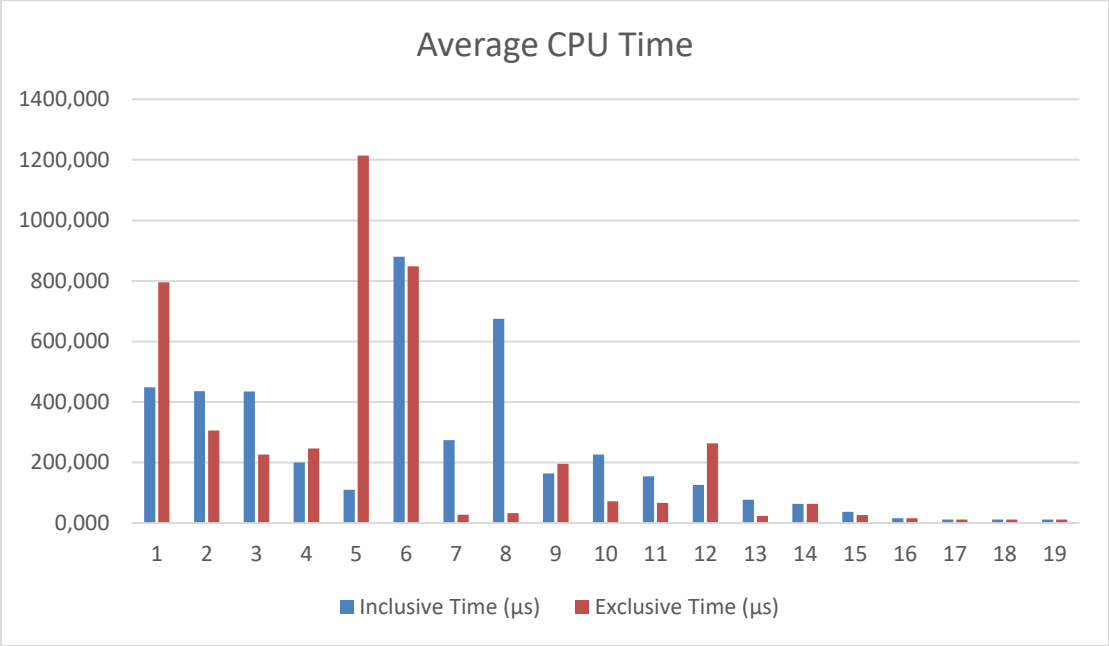
- άνοιγμα,
- σύνδεση,
- λήψη εικόνας,
- εισαγωγή μηνύματος,
- στεγανογράφηση,
- αποστολή

Object Name	Invocation Count	Inclusion (μs)	Exclusion (μs)
activitites.LoginActivity.loginMethod	1	448,494	795
utils.HTTPRequestsUtil.login	1	435,955	305
utils.HTTPRequestsUtil.performPostCall	1	434,471	226
activitites.common.Application.initNetData	1	200,053	246
utils.NetworkUtil.getMACAddress	2	109,623	1214
utils.NetworkUtil.getIPAddress	2	879,980	848
activitites.LoginActivity.init	1	273,816	27
utils.HTTPRequestsUtil.clinit	1	674,710	32
objects.NetworkData.toString	1	163,730	195
enums.SupportedHTTPMethods.clinit	1	226,500	72
enums.SupportedHTTPMethods.init	2	154,300	66
activitites.LoginActivity.onCreate	1	125,188	263

utils.HTTPRequestsUtil.init	1	77,100	23
objects.MySingleton.getInstance	4	63,100	63
objects.NetworkData.init	1	36,800	26
objects.MySingleton.setServer	1	16,000	16
objects.MySingleton.getNetworkData	1	11,000	11
objects.MySingleton.getServer	1	11,000	11
objects.MySingleton.setNetworkData	1	11,000	11

Πίνακας 6, Μετρήσεις Inclusion και Exclusion ανά object κώδικα σε διάρκεια μιας πλήρης λειτουργίας

Σύμφωνα με τους παραπάνω χρόνους, καθώς και με το παρακάτω διάγραμμα παρατηρούμε ότι η ανώτερη τιμή καθυστέρησης κυμαίνεται στα 1204 μs και η μικρότερη παρουσιάζεται στα 11 μs που υποδηλώνει διεργασία σε αναμονή (idle object).



Εικόνα 43, Διάγραμμα CPU ανά Object

Οι βασικές δομές που εκτελούνται σε κάθε πλήρη λειτουργία της εφαρμογής διακρίνονται στα πρώτα τέσσερα στάδια του παραπάνω διαγράμματος. Οι τιμές αυτές δεν ξεπερνάνε τα 2000 μs (συγκεκριμένα 1572 μs) γεγονός που επιβεβαιώνει την ομαλή λειτουργία της εφαρμογής.

Αξίζει να σημειωθεί ότι σύμφωνα με τα επίσημα Performance and Stability tests (Google n.d.) που χρησιμοποιεί η Google για την δημοσίευση των εφαρμογών στο Google Play οι τιμές που οριοθετούν την δοκιμή PM-1 με χρήση του mode StrictMode.ThreadPolicy.Builder δεν πρέπει να ξεπερνούν τα 3000 μs.

Κατά συνέπεια, η εφαρμογή που αναπτύχθηκε καλύπτει πλήρως τις βασικές προϋποθέσεις από πλευρά επεξεργαστικής ισχύος που καταναλώνεται σε εμπορικές Android συσκευές.

5.3.4 Ασφάλεια

Η ασφάλεια του προτεινόμενου αλγόριθμου στεγανογραφίας ενισχύεται, σε σύγκριση με τον αλγόριθμο LSB, μέσω της κρυπτογράφησης της θέσης των LSB αξιοποιώντας ένα μέρος της τεχνολογίας της κρυπτογράφησης μέσω του αλγορίθμου AES.

5.3.4.1 Εντροπία και κρυπτογράφηση

Για την μέτρηση της ασφάλειας του αλγορίθμου χρησιμοποιείται η έννοια της εντροπίας (entropy). Η εντροπία είναι η μέτρηση της τυχαιότητας. Η ιδέα προήλθε από την μελέτη της θερμοδυναμικής, όμως ο Claude E. Shannon προχώρησε στην εφαρμογή της έννοιας της εντροπίας σε ψηφιακές τεχνολογίες (Weaver και Shannon n.d.) και ασχολήθηκε με τον προσδιορισμό της μέγιστης τιμής στην οποία θα μπορούσε να συμπιεστεί ένα ψηφιακό αρχείο. Ο ίδιος θεωρείται από πολλούς ένας από τους βασικότερους μελετητές της συμπίεσης δεδομένων, αλλά και της κρυπτογραφίας.

Ο τρόπος συμπίεσης αρχείου που εισήγαγε ο Shannon βασίζεται σε αντικατάσταση κάποιων μοτίβων bits με μικρότερα πρότυπα bits. Κατ' αναλογία, η κρυπτογράφηση βασίζεται στην πολλαπλή αλλαγή των ίδιων μοτίβων με μεγαλύτερα μη πρότυπα bits. Δεδομένου ότι η εντροπία αποτελεί μέτρηση της τυχαιότητας, όσο πιο μεγάλη η εντροπία σε ένα αρχείο δεδομένων, τόσο λιγότερο μπορεί να συμπιεστεί και όσο μικρότερη η εντροπία τόσο πιο εύκολη η αποκρυπτογράφησή του. Με βάση τα παραπάνω ένα αρχείο με μεγάλη εντροπία δεν μπορεί να προσπελαστεί παρά μόνο αν αποκωδικοποιηθεί. Στην περίπτωσή μας η εφαρμογή της κωδικοποίησης εφαρμόζεται σε πολύ συγκεκριμένο σημείο και εξυπηρετεί την αρτιότερη χρήση της στεγανογραφίας.

Με την χρήση της γλώσσας προγραμματισμού Python προχωρήσαμε σε ελέγχους εντροπίας στο τελικό στεγανογραφημένο αρχείο εικόνας τόσο με την χρήση του αλγορίθμου που υλοποιήσαμε LSBv4AwRaC αλλά και του αλγορίθμου MobiStego.

Κάνοντας χρήση του παρακάτω κώδικα (Hartman n.d.) υπολογίζεται η εντροπία στα παραγόμενα αρχεία και στη συνέχεια αναπαρίσταται η συνάρτηση συχνότητας και bytes σε γραφήματα.

```
import sys
import math

if len(sys.argv) != 2:
    print "Usage: entropy.py [path]filename"
    sys.exit()
```

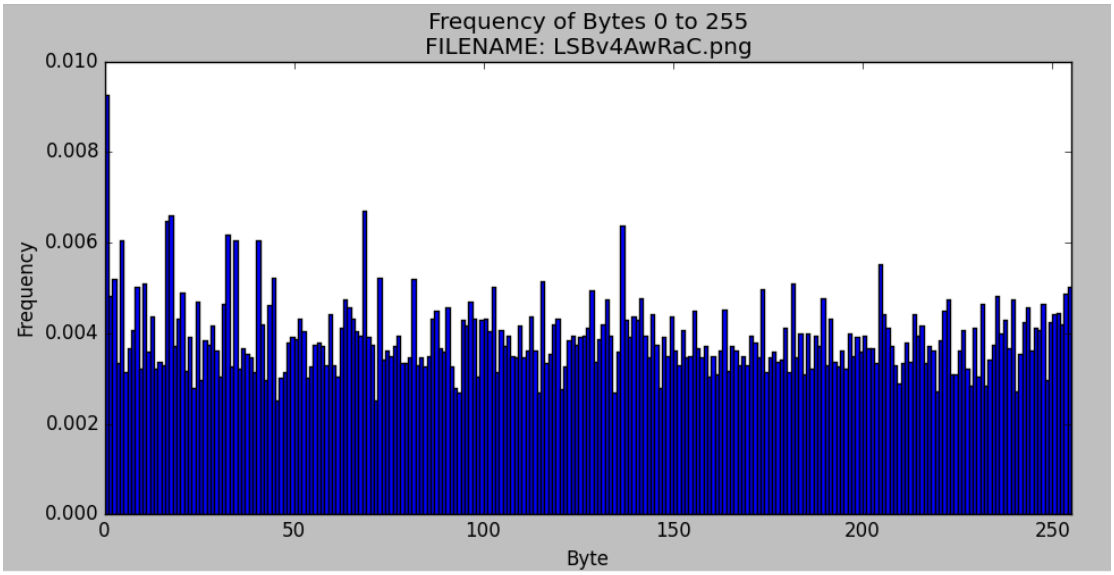
```
f = open(sys.argv[1], "rb")
byteArr = map(ord, f.read())
f.close()
fileSize = len(byteArr)
print 'File size in bytes:'
print fileSize
print

freqList = []
for b in range(256):
    ctr = 0
    for byte in byteArr:
        if byte == b:
            ctr += 1
    freqList.append(float(ctr) / fileSize)

# Shannon entropy
ent = 0.0
for freq in freqList:
    if freq > 0:
        ent = ent + freq * math.log(freq, 2)
ent = -ent
print 'Shannon entropy (min bits per byte-character):'
print (ent / 8)
print ' Shannon entropy (total bits per character):'
print ent
print
print 'Min size assuming max theoretical compression efficiency:'
print (ent * fileSize), 'in bits'
print (ent * fileSize) / 8, 'in bytes'

import numpy as np
import matplotlib.pyplot as plt

N = len(freqList)

ind = np.arange(N)
width = 1.00

fig = plt.figure(figsize=(11,5),dpi=100)
ax = fig.add_subplot(111)
rects1 = ax.bar(ind, freqList, width)
ax.set_autoscalex_on(False)
ax.set_xlim([0,255])

ax.set_ylabel('Frequency')
ax.set_xlabel('Byte')
ax.set_title('Frequency of Bytes 0 to 255\nFILENAME: ' +
sys.argv[1])

plt.show()
```

Με την ολοκλήρωση της παραπάνω διαδικασίας λαμβάνουμε τα παρακάτω αποτελέσματα αναφορικά με την εντροπία του κάθε στεγανογραφημένου αρχείου αλλά και τις πληροφορίες πιθανής συμπίεσης.

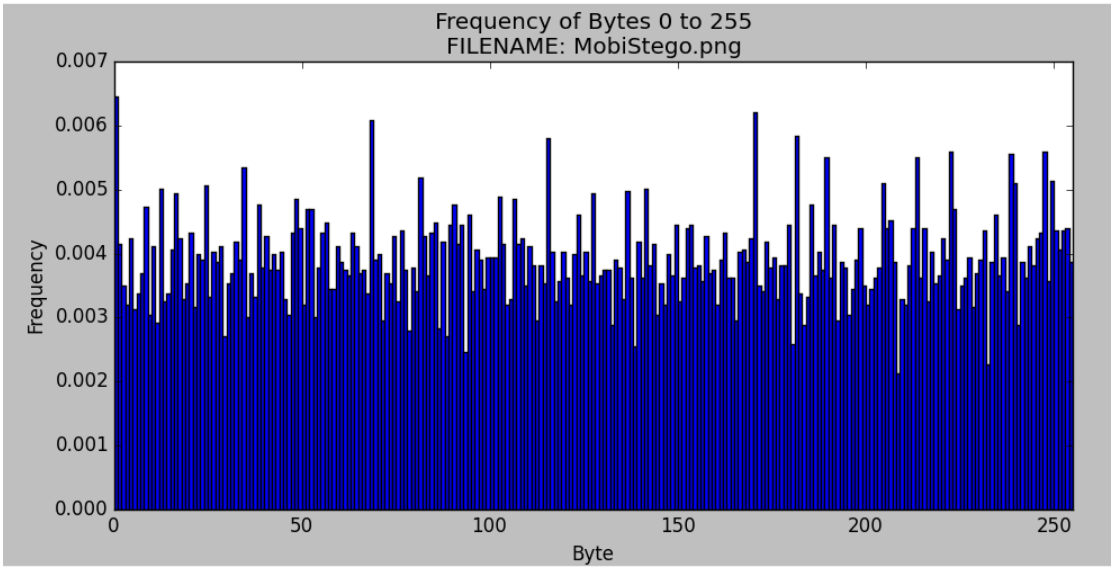
@> python entropy.py **LSBv4AwRaC.png**
File size in bytes:
94761
Shannon entropy (min bits per byte-character):
0.92012591938
Shannon entropy (total bits per character):
7.36100735504
Min size assuming max theoretical compression efficiency:
756361.3098856 in bits
94545.1637357 in bytes

@> python entropy.py **MobiStego.png**
File size in bytes:
94012
Shannon entropy (min bits per byte-character):
0.92011500779
Shannon entropy (total bits per character):
7.36092006232
Min size assuming max theoretical compression efficiency:
751388.834801 in bits
93923.6043502 in bytes

Τα αποτελέσματα των μετρήσεων μας παρουσιάζουν μια διαφορά στην εντροπία ανάμεσα στα δυο παραγόμενα αρχεία της τάξης των 0.00001091159 στο ελάχιστο bit και 0.00008729272 bits στην συνολική εντροπία που είναι ικανή να προσθέσει το επιμέρους στοιχείο της κρυπτογράφησης των θέσεων στο στεγανογραφημένο μας αρχείο. Στα παρακάτω διαγράμματα παρουσιάζεται η συνολική μορφή δεδομένων στα δυο υπό εξέταση αρχεία ανάμεσα στην συχνότητα και τα bytes.



Εικόνα 44 Συχνότητα/Byte αρχείου LSBv4AwRac.png



Εικόνα 45 Συχνότητα/Byte αρχείου MobiStego.png

Στην συνέχεια πραγματοποιούμε επιπλέον μετρήσεις με την παραπάνω μέθοδο σε 10 διαφορετικές στεγανογραφημένες φωτογραφίες.

# Εικόνα	**Entropy**	
Αλγόριθμος	***LSBv4AwRaC***	***MobiStego***
1	7,36100735504	7,36092006232
2	7,62316134513	7,42456434200
3	7,36534563450	7,29345236345
4	7,52346134534	7,42456134533
5	7,43451345223	7,40034523423
6	7,33452452340	7,31084523420
7	7,60013451340	7,53234245634
8	7,39992345234	7,31423423223
9	7,59345122232	7,51111534520
10	7,38345441214	7,29994524423

Πίνακας 7, Μετρήσεις Εντροπίας

Από τις παραπάνω μετρήσεις παρατηρούμε πως όλες οι εικόνες που δοκιμάστηκαν παρουσιάζουν μεγαλύτερη εντροπία με την χρήση του αλγόριθμου μας. Ένας από τους κύριους λόγους που συμβαίνει αυτό είναι γιατί η εναλλαγή των bit στοιχείων που αλλάζουμε κατανέμονται σε μεγαλύτερο εύρος στο σύνολο του αρχείο με αποτέλεσμα να επηρεάζεται η συνολική εντροπία του παραγόμενου αρχείου.

Αρχεία με μεγαλύτερη εντροπία δεν είναι ζητούμενα στα πλαίσια της διατριβής αυτής. Ο στόχος μας είναι η παραγωγή ενός αρχείου στο οποίο τα κύρια χαρακτηριστικά θα παραμείνουν αναλλοίωτα, για την περίπτωση μας δηλαδή, το αρχείο θα παραμείνει αναγνώσιμο ως αρχείο φωτογραφίας. Αν το αρχείο μας ήταν κρυπτογραφημένο δεν θα μπορούσε να είναι προσπελάσιμο χωρίς την αποκωδικοποίησή του. Αρχεία τα οποία είναι πλήρως κωδικοποιημένα ή συμπιεσμένα παρουσιάζουν πολύ μεγάλες τιμές εντροπίας με αποτέλεσμα να μην είναι άμεσα προσπελάσιμα. Για το λόγο αυτό η κρυπτογράφηση πραγματοποιείται μόνο για τις θέσεις των LSB όπου κρύβεται το μήνυμα και όχι στα δεδομένα της εικόνας.

Η χρήση της κρυπτογραφικής μεθόδου που υλοποιήθηκε βασίζεται στον αλγόριθμο AES. Όπως αναφέρεται και στο κεφάλαιο 4.2.1.3 η πληροφορία για το ποια είναι τα ελάχιστα σημαντικά bits που έχουν τροποποιηθεί ώστε σε αυτά να αποκρυφθεί το μήνυμα μας, υποβάλλεται σε κρυπτογράφηση μέσω της κλάσης encrypt. Μέσω του μοναδικού κλειδιού που ανακτάται από την βάση δεδομένων ανάμεσα σε χρήστες που επικοινωνούν με την χρήση της εφαρμογής παράγεται ένα getRawKey των 256bit. Η πληροφορία για το πού τελικά βρίσκονται τα LSB στοιχεία υποβάλλεται σε κρυπτογράφηση, γεγονός που καθιστά αδύνατο τον εντοπισμό του κρυφού μηνύματός (στεγανάλυση) μέσω των συνηθισμένων τεχνικών εξάλειψης στεγανογραφίας.

5.3.4.2 Ανθεκτικότητα σε επιθέσεις

Στα πλαίσια των ελέγχων ασφάλειας για τον προτεινόμενο αλγόριθμο δοκιμάζεται μέσω του εργαλείου StegExpose, η ανθεκτικότητά του σε επιθέσεις (attacks). Το εργαλείο StegExpose είναι ένα εργαλείο στεγανάλυσης που διακρίνει την στεγανογραφία τυπου LSB σε αρχεία εικόνας χωρίς απωλεστική μέθοδο συμπίεσης (lossless compression). Τέτοια αρχεία είναι και τα PNG που χρησιμοποιούμε στην παρούσα μεταπτυχιακή διατριβή. Το συγκεκριμένο εργαλείο πσρέχει διεπαφή γραμμής εντολών και έχει σχεδιαστεί για να αναλύει τις εικόνες σε ογκώδη διάσταση (bulk while) παρέχοντας παράλληλα δυνατότητες αναφοράς και παραμετροποίησης. Ο αλγόριθμος που χρησιμοποιείται για την αξιολόγηση μιας εικόνας προέρχεται από δοκιμασμένες τεχνικές των γειτονικών και αλλαγμένων pixel (μέσω της διαβάθμισης των bit planes) με βάση τις μεθόδους:

- Sample Pairs (Dumitrescu, Wang and Wu , 2003)
- RS Analysis (Fridrich and Goljan , 2004)
- Chi Square Attack (Westfeld and Pfitzmann , 2000)
- Primary Sets (Dumitrescu, Wu and Memon , 2002)

Για τις ανάγκες των μετρήσεων μας, χρησιμοποιήθηκε η εφαρμογή που αναπτύχθηκε και περιγράφεται στην Ενότητα 4.3.2. Μέσω της εφαρμογής έγιναν δεκαέξι (16) λήψεις και δημιουργήθηκαν 16 αρχεία εικόνας, εκ των οποίων οκτώ

(8) δημιουργήθηκαν με την εφαρμογή του προτεινόμενου αλγόριθμου LSBv4AwRaC και οκτώ (8) με την εφαρμογή του απλού LSB αλγορίθμου MobiStego. Για κάθε μία από τις φωτογραφίες χρησιμοποιήθηκε το ίδιο μήνυμα ανά ζεύγος οκτάδας. Άρα για κάθε μια περίπτωση χρήσης έγινε εισαγωγή ίδιου μηνύματος και για τους δυο αλγόριθμους. Σε ιδανικές συνθήκες οι οκτώ φωτογραφίες που χρησιμοποιήθηκαν με τον έναν αλγόριθμο θα έπρεπε να ήταν κοινές με τις υπόλοιπες οκτώ φωτογραφίες του δευτέρου. Η χρήση της φωτογραφικής μηχανής του κινητού τηλεφώνου καθιστά αυτό το γεγονός εξαιρετικά απίθανο. Έγινε προσπάθεια όλες οι φωτογραφίες να απεικονίζουν σχεδόν το ίδιο καθώς και ο φωτισμός του δωματίου να μην μεταβάλλεται. Το αποτέλεσμα σε επίπεδο μεγέθους (MB) της εκάστοτε φωτογραφίας ήταν ικανοποιητικό και δόθηκε έτσι η δυνατότητα να υλοποιηθούν οι παρακάτω μετρήσεις.

Η σύνταξη της εντολής που χρησιμοποιήθηκε για τις μετρήσεις μας είναι η παρακάτω:

```
$>java -jar StegExpose.jar /home/dimx/stegex/imgs default default
results.csv
```

File name	Above stego threshold?	Secret message size in bytes (ignore for clean files)
LSBv4AwRaC (1).png	FALSE	16762
LSBv4AwRaC (2).png	FALSE	1088
LSBv4AwRaC (3).png	FALSE	22421
LSBv4AwRaC (4).png	FALSE	851
LSBv4AwRaC (5).png	FALSE	22425
LSBv4AwRaC (6).png	TRUE	1132
LSBv4AwRaC (7).png	FALSE	13416
LSBv4AwRaC (8).png	FALSE	2354
MobiStego (1).png	TRUE	29
MobiStego (2).png	TRUE	97
MobiStego (3).png	TRUE	71
MobiStego (4).png	TRUE	44
MobiStego (5).png	FALSE	18974
MobiStego (6).png	TRUE	79
MobiStego (7).png	FALSE	673
MobiStego (8).png	FALSE	13094

Πίνακας 8, Οι πρώτες 3 στήλες του results.csv

Μελετώντας τα αποτελέσματα που προέκυψαν από το εργαλείο StegExpose παρατηρείται υπεροχή στην απόκρυψη της εφαρμοσμένης στεγανογραφίας στις περιπτώσεις που γίνεται χρήση του προτεινόμενου αλγόριθμου. Η μοναδική περίπτωση όπου το εργαλείο στεγανάλυσης εμφανίζει αληθή την πιθανότητα εφαρμογής στεγανογραφίας είναι στο αρχείο LSBv4AwRaC (6).png. Δεδομένου όμως ότι το μήνυμα που επιχειρείται να ενσωματωθεί δεν είναι 1132 bytes, αλλά

79 bytes, οδηγούμαστε στο συμπέρασμα ότι το εργαλείο υπολόγισε λανθασμένα το κρυφό μήνυμα και άρα η εν λόγω μέτρηση δεν είναι έγκυρη.

Παρακάτω παρουσιάζεται ο πίνακας αντιστοίχησης των μηνυμάτων που χρησιμοποιήθηκαν για τις μετρήσεις και το μέγεθός τους σε bytes.

Εικόνα	Μήνυμα στεγανογραφίας	Bytes
1	message for mob1 to mob2 test	29
2	You cant win darth strike me down and I will become more powerful than you could possibly imagine	97
3	he present is theirs the future for which I really worked is mine.tesla	71
4	Imagination is more important than knowledge	44
5	Everything is theoretically impossible until it is done. Heinlein	65
6	Nothing is too wonderful to be true if it be consistent with the laws of nature	79
7	run Forest run	14
8	Art is I,science is we.Bernard	30

Πίνακας 9, Πίνακας Μηνυμάτων

Τέλος, με το εργαλείο stegExpose μπορεί να γίνει εξαγωγή των τιμές για τις στατιστικές έννοιες Primary Sets, Chi Square, Sample Pairs, RS analysis και Fusion (mean) που εμφανίζονται παρακάτω. Σε μελλοντική έρευνα, με χρήση των τιμών αυτών σε στατιστικό δείγμα που ξεπερνά τις 2000 μετρήσεις μπορούμε να προβούμε στην απαραίτητη στατιστική ανάλυση και την δημιουργία ROC διαγραμμάτων.

Primary Sets	Chi Square	Sample Pairs	RS analysis	Fusion (mean)
0.008091719013158376	6,56E+11	null	0.029506171349486195	0.012751212679568172
0.352075102336637	0.2737126728972277	null	0.2091230824353674	0.2783036192230774
0.11329240178885477	0.1423569884449125	null	0.048735256189453034	0.10146154880774011
0.09372108615660697	2,25E-01	0.019343694656741996	0.046452957968860474	0.03987943469555237
0.11342008932949325	0.14357315271470206	null	0.049390600090339964	0.10212794737817843
NaN	0.09371291967741853	0.7812415352997496	0.7680715936124942	0.5476753495298875
0.01653992591872635	0.0020576187477316554	null	0.036616633877419855	0.01840472618129262
0.028480431325644658	0.03596650763921852	0.080529823771301	0.08440531345457021	0.05734551904768359
NaN	0.032714920140734266	null	0.900069710099886	0.46639231512031015
0.014035814076028972	1,53E+03	0.05621926369452168	0.0680779606771208	0.03458325961195617
0.11416125219829386	0.14175864542544572	null	0.05031568315520302	0.1020785269263142
0.07596224742087106	null	null	0.03403872160357525	0.055000484512223155
0.07430830039525692	null	null	0.034743251591151006	0.05452577599320396
0.15712894630704333	null	null	0.05292629759985697	0.10502762195345014
0.11329240178885477	0.1423569884449125	null	0.048735256189453034	0.10146154880774011
0.2573883057346056	0.002651049049337065	0.09692103046233898	0.10675110693744194	0.11592787304593088

Πίνακας 10, Οι υπόλοιπες στήλες του results.csv

6 Συμπεράσματα

Στα πλαίσια της παρούσας διπλωματικής διατριβής στόχος ήταν η πρόταση μιας νέας προσέγγισης στεγανογραφίας και η ανάδειξη των συγκριτικών πλεονεκτημάτων του προτεινόμενου αλγορίθμου έναντι των υφιστάμενων στον χώρο της στεγανογραφίας. Η βιβλιογραφική μελέτη και η πρακτική δοκιμασία στην οποία υποβλήθηκε ο προτεινόμενος αλγόριθμος LSBv4AwRaC συνοψίζεται στα παρακάτω.

- Αξιοποίηση και επέκταση του αλγορίθμου LSB.
- Τοποθέτηση πληροφορίας στο σύνολο της εικόνας, σε τυχαία και όχι σειριακά ελάχιστα σημαντικά bits (LSB).
- Κρυπτογράφηση των σημείων αυτών με μοναδικό κλειδί ανά συσχετισμό επαφών (relationship «encryptKey»).
- Υλοποίηση αλγορίθμου σε λογισμικό Android.
- Κεντροποιημένη διαχείριση εφαρμογής με χρήση RESTfull service, βάσης δεδομένων και Registration Portal.
- Χρήση μορφής αρχείου PNG με μη απωλεστική μέθοδο συμπίεσης (lossless compression)

Στη συνέχεια παρατίθενται τα ανωτέρω χαρακτηριστικά σε αντιστοιχία με τους ευρέως γνωστούς αλγόριθμους που βρίσκουν χρήση στη στεγανογραφία, αλλά και οι μελλοντικές επεκτάσεις και πρακτικές εφαρμογές της προσέγγισης μας.

6.1 Συγκριτική Αξιολόγηση

Όπως αναλύεται σε προηγούμενο κεφάλαιο, η προσαρμοστική μέθοδος βασίζεται στην τεχνική PBSM Plane Bit Substitution Method (Johnson and Katzenbeisser , 2000) η οποία όμως καταναλώνει μεγάλους υπολογιστικούς πόρους στην προσπάθεια της να χρησιμοποιήσει πληροφορία, η οποία βρίσκεται αποθηκευμένη σε παράπλευρα bits και να δημιουργηθεί έτσι μια εκτίμηση του αριθμού των συνολικών bits στα οποία μπορεί τελικά να ενσωματωθεί το κρυφό μήνυμα. Ο αλγόριθμος αυτός μπορεί να προσφέρει καλύτερη ασφάλεια συγκριτικά με ένα αλγόριθμο LSB στεγανογραφίας, αλλά σε υλοποίηση για Android συσκευή είναι απαγορευτικός τόσο λόγω του απαιτούμενου χρόνου εκτέλεσης του αλγόριθμου, όσο και λόγω της αλλοίωσης που τελικά επιφέρει στο στεγανογραφημένο αρχείο που δημιουργείται κατά την λήψη μιας φωτογραφίας.

Με την αξιοποίηση ενός αλγορίθμου L2LSB ο προτεινόμενος αλγόριθμος προσφέρει κρυπτογράφηση σε σταθερό σημείο μιας στεγανογραφημένης εικόνας. Η κρυπτογράφηση σε σταθερό σημείο καθιστά τον αλγόριθμο L2LSB ευάλωτο σε επιθέσεις, καθώς το σημείο αυτό βρίσκεται σε κάθε τελευταίο bit σε σειριακή μορφή δίνοντας την δυνατότητα εύκολων επιθέσεων με στόχο την εύρεση του κρυπτογραφημένου κειμένου, μέσω της μαζικής λήψης όλων των

τελευταίων μη σημαντικών bits από τα εικονοστοιχεία με χρήση μεθόδων τύπου Viterbi (Forney , 2005) αλγορίθμου. Αντίθετα, το μήνυμα στον LSBv4AwRaC αλγόριθμο, που προτάθηκε και αναπτύχθηκε στην συγκεκριμένη μεταπτυχιακή διατριβή, δεν αποθηκεύεται σε σειριακά LSB στοιχεία παρά μόνο σε τυχαία. Επιπλέον, η τοποθεσία των τυχαίων αυτών σημείων κρυπτογραφείται μέσω του μοναδικού relationship κλειδιού «encryptKey» που ανακτάται από την βάση δεδομένων της εφαρμογής.

Οι αλγόριθμοι στεγανογραφίας που βασίζονται στην τεχνική Discrete Cosine Transform (DCT) όπως οι L2LSB-EDCT και 3D-DCT χρησιμοποιούνται σε εικόνες μορφής JPEG που ενώ παρουσιάζουν ποιοτικά πλεονεκτήματα (υψηλός PSNR) και δίνουν τη δυνατότητα ενσωμάτωσης υδατογράφηματος (Thongkor, et al. , 2013) για τη διασφάλιση των πνευματικών δικαιωμάτων, αποτελούν πολύπλοκες υλοποιήσεις. Επιπλέον, κατά την ενσωμάτωσή τους σε εφαρμογή Android καταγράφηκαν ατέλειες που οφείλονται σε σφάλματα που πραγματοποιήθηκαν κατά τη διαδικασία μετατροπής του εύρους χρωμάτων (changing color space process). Τα σφάλματα αυτά επαφίονται στη χρήση εικόνων JPEG, όπου γίνεται χρήση απωλεστικής μεθόδου συμπίεσης (lossy data compression), οδηγώντας σε απώλεια δεδομένων, καθιστώντας ένα στεγανογραφημένο μήνυμα αδύνατο να κρυπτογραφηθεί σε μια ήδη αλλοιωμένη με τις παραπάνω τεχνικές φωτογραφία (JPEG). Επιπλέον, περιορισμοί τίθενται από το λειτουργικό σύστημα Android, λόγω ελλιπούς υποστήριξης του χρωματικού μοντέλου YCbCr σε ορισμένες εκδόσεις (είναι διαθέσιμο στην έκδοση Android 4.4 Kitkat (API level 19) και έπειτα) (Rahmanto, Faisal and Nugroho , 2016). Ο αλγόριθμος που προτείναμε ξεπερνάει το παραπάνω πρόβλημα κάνοντας χρήση της μορφής PNG μιας πλέον ευρύτατα διαδεδομένης μορφής αποθήκευσης φωτογραφίας βασισμένη σε μη απωλεστική μέθοδο συμπίεσης (lossless compression) και ενσωμάτωσης πάνω από 16 εκατομμύρια χρώματα.

Αναφορικά με την βελτιωμένη μέθοδο LSB η οποία κάνει χρήση μιας γεννήτριας τυχαίων αριθμών βασιζόμενη στο Henon Chaotic Map, πρόκειται επίσης για μέθοδο που χρησιμοποιείται σε αρχεία τύπου JPEG που όπως αναφέραμε παραπάνω δεν ενδείκνυνται για στεγανογραφία σε Android συσκευή χωρίς η συμπίεση να αλλοιώσει το τελικό αποτέλεσμα. Επιπλέον, μειονέκτημα της μεθόδου αυτής θεωρείται το χαμηλό επίπεδο ασφάλειας που προσαρμόζει πάνω στο αρχείο, μιας και σημαντική προϋπόθεση για την αποκωδικοποίηση του αρχείου με την μέθοδο αυτή είναι η κατοχή της γεννήτριας τυχαίων αριθμών τόσο στην συσκευή του αποστολέα όσο και στην συσκευή του παραλήπτη. Στην περίπτωση του προτεινόμενου αλγορίθμου LSBv4AwRaC που αναπτύξαμε η χρήση ενός κλειδιού που ανακτάται από την βάση δεδομένων της εφαρμογής με μοναδικό κριτήριο την ύπαρξη σχέσης μεταξύ των συσκευών, και κατ' επέκταση χρηστών που επικοινωνούν, καθιστά σαφή την απλοποιημένη μέθοδο αποκωδικοποίησης χωρίς επιπλέον λογισμικό ή κώδικα γεννήτριας τύπου Henon Chaotic Map που εξ ορισμού θα δημιουργούσε κενό ασφάλειας. Στην

προτεινόμενη υλοποίηση το κλειδί αποθηκεύεται σε ασφαλές περιβάλλον (server-side) και καλείται από την java κλάση μέσω της παραμέτρου «encryptKey» κατά το encode και decode από τις φορητές συσκευές. Η στεγανογραφημένη φωτογραφία δεν μπορεί να αποκωδικοποιηθεί αν δεν μεσολαβήσει ο κεντρικός εξυπηρετητής πράγμα που την καθιστά ασφαλέστερη σε σύγκριση με τον παραπάνω μηχανισμό τυχαίων αριθμών που θα βρισκόταν αποθηκευμένος στην εκάστοτε συσκευή.

Οι δοκιμές με το προτεινόμενο σύστημα στεγανογραφίας που αναπτύχθηκε επαλήθευσαν την ορθότητα των λειτουργιών του, όπως αναλύεται στις Ενότητες4 και 5, αλλά και την ικανοποιητική του απόδοση λαμβάνοντας υπόψη τους περιορισμούς που επιβάλλει το λειτουργικό σύστημα του Android. Συγκρίνοντας τον αλγόριθμο (LSBv4AwRaC) που έχει υλοποιηθεί με τον αλγόριθμο (LSB) που χρησιμοποιεί αντίστοιχο Android σύστημα (MobiStego) διαπιστώνεται ότι ο τελευταίος εμφανίζει καλύτερους χρόνους εκτέλεσης καθώς χρησιμοποιεί στατικές και όχι τυχαίες θέσεις απόκρυψης του μηνύματος. Η σύγκριση αυτή προβάλλει το δίλημμα της ταχύτητας έναντι της μεγαλύτερης ασφάλειας που εγγυάται η τυχαιότητα και η ενσωμάτωση κρυπτογράφησης όπως συμβαίνει στον αλγόριθμο μας LSBv4AwRaC. Αν βασιστούμε στο γεγονός πως η εξέλιξη της τεχνολογίας στις φορητές συσκευές είναι ραγδαία και οι χρόνοι λειτουργίας της υλοποίησης μας σε σύντομο χρονικό διάστημα θα μειωθούν κατά πολύ σε συνάρτηση με το πλήθος των πληροφοριών που υποκλέπτονται καθημερινά η απάντηση στο δίλλημα αυτό είναι σαφώς η επιλογή της μεγαλύτερης ασφάλειας.

6.2 Μελλοντικές Επεκτάσεις

Στα πλαίσια της μελέτης και υλοποίησης της συγκεκριμένης μεταπτυχιακής διατριβής δημιουργήθηκαν ιδέες για ποικίλες διαφορετικές ανάγκες που θα μπορούσε να καλύψει ο αλγόριθμος LSBv4AwRaC.

Στην μεταπτυχιακή αυτή διατριβή επικεντρωθήκαμε στην υλοποίηση μίας νέας τεχνικής στεγανογραφίας για το λειτουργικό σύστημα Android που κλήθηκε να αντιμετωπίσει με καινοτόμο τρόπο τις προκλήσεις του λειτουργικού συστήματος και συγχρόνως να ενσωματώσει τυχαιότητα και κρυπτογράφηση στη διαδικασία της στεγανογραφίας. Η εφαρμογή που υλοποιήθηκε στο πλαίσιο της παρούσας διατριβής, μπορούμε να ισχυριστούμε ότι αποτελεί μια αυτόνομη δικτυακή υπηρεσία ανταλλαγής κρυφών μηνυμάτων που εκπληρώνει τους στόχους της δημιουργίας της, αλλά που αναμφισβήτητα επιδέχεται πολλών αλλαγών, βελτιώσεων και επεκτάσεων προκειμένου να αποτελέσει μια πλήρη υπηρεσία και να μπορεί να εισέλθει ανταγωνιστικά στο χώρο του Android Market.

Οι αρχικοί στόχοι πραγματώθηκαν στην πλειοψηφία τους, όμως κατά την ενασχόληση με την υλοποίηση και τη συνεχή επαφή με τις τάσεις που επικρατούν

στο λογισμικό του λειτουργικού συστήματος Android, αλλά και στις κεντροποιημένες υπηρεσίες (JEE, Portal), δημιουργήθηκαν στόχοι για περαιτέρω επεκτάσεις και αλλαγές που όμως ξεφεύγουν από τις προθέσεις του επικείμενου σκοπού της μεταπτυχιακής διατριβής.

Όσων αφορά στα τεχνικά χαρακτηριστικά της εφαρμογής, η δημιουργία ενός ενοποιημένου (bundle) πακέτου λογισμικού που θα εγκαθιστά εξολοκλήρου το λογισμικό Red Hat JBoss ή κάποια μελλοντική επέκταση του σε Apache Tomcat ή Glassfish θα αποτελούσε σημαντική βελτίωση στην προσφερόμενη υπηρεσία. Με την δημιουργία μιας πλήρους έκδοσης σε επίπεδο εγκατάστασης, ο χρήστης θα μπορούσε εύκολα να κάνει εύκολα χρήση της εφαρμογής που αναπτύχθηκε στην συγκεκριμένη διπλωματική εργασία. Στο πακέτο αυτό, θα πρέπει να εμπεριέχεται και η μηχανή λογισμικού Oracle Java, καθώς και οι έτοιμες ρυθμίσεις των μεταβλητών περιβάλλοντος (environmental variables) που είναι απαραίτητες για την λειτουργία της εφαρμογής. Απαραίτητα στοιχεία του ενοποιημένου πακέτου είναι, επιπλέον, η προσθήκη της JEE εφαρμογής καθώς και του Portal, μέσω του οποίου θα είναι διαθέσιμη και η μεταφόρτωση της mobile εφαρμογής, όπως και η εγκατάσταση της βάσης δεδομένων (MySQL).

Στα πλαίσια βελτιωτικών ενεργειών της εφαρμογής που αναπτύχθηκε για συσκευές android, μπορεί να σχεδιαστεί η ενσωμάτωση ενός πληρέστερου τρόπου εγγραφής (registration) και εισόδου (login) στην εφαρμογή. Οι συνήθεις πρακτικές που εφαρμόζονται στις περισσότερες εφαρμογές που υπάρχουν σε Mobile εκδόσεις και είναι κατάλληλες για instant messaging (whatsapp, viber, FB messanger, κ.α.) αξιοποιούν την μοναδικότητα του αριθμού κινητού τηλεφώνου για πιστοποίηση του εκάστοτε χρήστη, καθιστώντας ιδιαίτερα ασφαλή τη σύνδεση και επικοινωνία μεταξύ χρηστών.

Επιπρόσθετα, στα πλαίσια ενίσχυσης της ασφάλειας εισόδου στην εφαρμογή, προτείνεται η χρήση μοναδικού κλειδιού (token) βασισμένο σε χρονική σφραγίδα. Η λειτουργία των One Time Passwords (OTP) (Tilborg , 2005) είναι ευρύτατα διαδεδομένη και αποτελεί μια από τις κορυφαίες λύσεις σε επίπεδο ασφάλειας.

Άλλες προτάσεις βελτίωσης, βρίσκουν εφαρμογή σε λειτουργίες όπως αναζήτηση του παραλήπτη του στεγανογραφημένου μηνύματος από τη λίστα επαφών του κινητού τηλεφώνου, δημιουργία ομάδας (group messaging) για αποστολή στεγανογραφημένων εικόνων σε περισσότερων του ενός παραλήπτη, προτάσεις που θα προσδώσουν χαρακτηριστικά πληρότητας επιπέδου εμπορικής εφαρμογής.

Επιπλέον, σε μελλοντικό στάδιο, θα μπορούσαν να ενσωματωθούν λειτουργίες όπως οι παρακάτω στην πλατφόρμα Android.

- Ενσωμάτωση του Steganography Portlal και διαχείριση φίλων, σύναψη relationship μέσω του mobile app
- Υπηρεσίες άμεσης ενημέρωσης του χρήστη για λήψη νέου μηνύματος (instant notification service).
- Διαχείριση κλειδιών – επιλογή αλγορίθμων κρυπτογράφησης κατά την αποστολή του μηνύματος (on demand)

Σημαντική πρόκληση θα αποτελούσε, τέλος, η επέκταση και υλοποίηση αντίστοιχου συστήματος στο ευρέως διαδεδομένο λειτουργικό σύστημα iOS.

Συνοψίζοντας, στην μεταπτυχιακή αυτή διατριβή επικεντρωθήκαμε στην υλοποίηση μίας νέας τεχνικής στεγανογραφίας για το λειτουργικό σύστημα Android που κλήθηκε να αντιμετωπίσει με καινοτόμο τρόπο τις προκλήσεις του λειτουργικού συστήματος και συγχρόνως να ενσωματώσει τυχαιότητα και κρυπτογράφηση στη διαδικασία της στεγανογραφίας. Το συγκεκριμένο στοιχείο (component) μπορεί να αποτελέσει τον πυρήνα μελλοντικών συστημάτων στεγανογραφίας σε ακαδημαϊκό ή εμπορικό επίπεδο που υλοποιούν επίσης τα παραπάνω πιθανά μελλοντικά λειτουργικά χαρακτηριστικά.

Βιβλιογραφία

A.Nag, S. Biswas, D. Sarkar, and P.P. Sarkar. , 2010. "A novel technique for image steganography based on Block-DCT and Huffman Encoding."

Aggoun, A. , 2006. *A 3D Dct Compression Algorithm For Omnidirectional Integral Images.*

Anderson, R. J., και F. A. P. Petitcolas. ,1998. «On the limits of steganography.»

Ang, GAO, και WEI Wen-xue. , 2005. *Application of Java data persistence with Hibernate and Struts framework.*

Arroquic, Mauricio, Cristian Mateosa, Claudio Machadoc, και Alejandro Zunino. , 2012. *RESTful Web Services improve the efficiency of data transfer of a whole-farm simulator accessed by Android smartphones.*

C.Saravanakumar, and C.Arun. , 2014. "AN EFFICIENT ASCII-BCD BASED STEGANOGRAPHY FOR CLOUD SECURITY USING COMMON DEPLOYMENT MODEL." *Journal of Theoretical and Applied Information Technology* Vol. 65 No.3, pp. 687-694.

Chen, Yueh-Hong, and Hsiang-Cheh Huang. , 2011. "A Copyright Information Embedding System for Android Platform."

Cristy, John. , 1990. http://www.imagemagick.org/script/index.php.

Denning, Dorothy E. 1999. *Information Warfare and Security.*

n.d. "Design of image steganography algorithms with secure message routing for p2p networks ." http://shodhganga.inflibnet.ac.in:8080/jspui/bitstream/10603/38614/9/09_chapter4.pdf.

Dharman, Reshma, και Ajoy Thomas. , 2016. «Steganalysis Application to Detect Image with Malicious Code.»

Dumitrescu, S., Zhe Wang, and Xiaolin Wu. , 2003. *Detection of LSB steganography via sample pair analysis.*

Dumitrescu, Xiaolin Wu, and N. Memon. , 2002. *On steganalysis of random LSB embedding in continuous-tone images.*

E., Varsaki, V.Fotopoulos, και Skodras A. , 2011. «Data Hiding Based On Texture Classification.»

Falesh, Mr., M. Shelke, Miss. Ashwini, A. Dongre, Mr. Pravin, και D. Soni. 2014. «Comparison of different techniques for Steganography in images.»

Forney, G.D. , 2005. «The viterbi algorithm.»

Fridrich, Jessica, and Miroslav Goljan. , 2004. "On estimation of secret message length in LSB steganography in spatial domain."

Google. n.d. https://developer.android.com/distribute/essentials/quality/core.html.

—. n.d. *Bitmap.Config.* Accessed 03 01, , 2016. http://developer.android.com/reference/android/graphics/Bitmap.Config.html.

Grønli, Tor-Morten, Jarle Hansen, and Gheorghita Ghinea. , 2010. "Android vs Windows Mobile vs Java ME: a comparative study of mobile development environments."

Hartman, Kenneth G. n.d. http://www.kennethghartman.com/.

Hussain, Mehdi. , 2013. «A Survey of Image Steganography Techniques.»

Instruments, N. , 2013. «PSNR as an Image Quality Metric.»

Johnson, N.F., και S. Jajodia. , 1998. «Exploring Steganography: Seeing the Unseen.»

Johnson, N.F′, και S. Jajodia. , 2008. «Exploring Steganography:Seeing the Unseen.»

Johnson, Neil F., and Stefan C. Katzenbeisser. , 2000. "A servery of steganographic techniques."

Judge, James C. , 2001. «Steganography: Past, Present, Future.»

Kahn, David. ,1967. *The Codebreakers.*

Ker, Andrew D. , 2007. "Steganalysis of Embedding in Two Least-Significant Bits."

Kuhn, M. , 1995. *Steganography and Watermarking.*

Leung, Henry, and Siyue Chen. , 2009. *Chaotic watermarking for a digital image.*

Louvel, Jerome, Thierry Templier, and Thierry Boileau. 2012. "Restlet in Action: Developing RESTful web APIs in Java."

Maganbhai, Parmar Ajit Kumar, και Prof. Krishna Chouhan. , 2015. «A Study and literature Review on Image Steganography.»

Massol, Vincent, και Ted Husted. , 2003. *JUnit in Action .*

Microsoft Corporation. , 2002. *Transformation of EXIF images.* https://www.google.com/patents/US7676118.

Ms.G.S.Sravanthi, Mrs.B.Sunitha Devi, S.M.Riyazoddin, and M.Janga Reddy. , 2012. "A Spatial Domain Image Steganography Technique Based on Plane Bit Substitution Method." *Global Journal of Computer Science and Technology Graphics & Vision.*

Panasenko, Sergey, και Sergey Smagin. , 2011. «Lightweight Cryptography: Underlying Principles and Approaches.»

Panigrahi, Bikash Kumar, και P. Sanjeeva Reddy. , 2014. «High Quality High Capacity Robust DWT Based Steganography.»

Parashar, Manish, Manish Agarwal, Arbeeny Steele, Viraj Bhat, and Rangini Chowdhury. , 2001. "Evaluating Security Mechanisms in Peer-to-Peer Applications."

Pasquale, Paola. , 2016. *MobiStego.* 03 01. https://github.com/paspao/MobiStego.

Raghava, N., Kumar Ashish, Deep Aishwarya, and Chahal Abhilasha. , 2014. "Improved LSB method for Image Steganography using Henon Chaotic Map." *OPEN JOURNAL OF INFORMATION SECURITY AND APPLICATIONS.*

Rahmanto, Oky, M Reza Faisal, and Radityo Adi Nugroho. , 2016. "Robustness Test of Discrete Cosine Transform Algorithm in Digital Image Watermarking on Android Platform."

Rani Lakshmi, S, and D. Pradeepa. , 2014. "Efficient Routing Strategy for Structured Peer to Peer System Using DWT Based Steganography." *International Journal of Science and Research (IJSR).*

Raphael, Eidenbenz, Locher Thomas, and Wattenhofer Roger. , 2011. "Hidden Communication in P2P Networks: Steganographic Handshake and Broadcast."

Satyavathy, G., and M. Punithavalli. , 2011. "LSB, 3D-DCT and Huffman Encoding based Steganography in Safe Message Routing and Delivery for Structured Peer-to-Peer Systems." *IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications".*

Shah, Saumil. , 2015. «Hacking with pictures.»

Shelke, Mr. Falesh M., Miss. Ashwini A. Dongre, και Mr. Pravin D. Soni. , 2014. «Comparison of different techniques for Steganography in images.»

Shrote, Khushboo R., και Pushpanjali Chouragade. , 2015. «A Review Paper for Storage and Computation on Enterprise Data in the Cloud.»

Siikarla, Mika, Markku Laitkorpi, Petri Selonen, και Tarja Systä. , 2008 . *Transformations Have to be Developed ReST Assured .*

Singh, Mona. , 2008. *U-SCRUM: An Agile Methodology for Promoting Usability.*

Taylor, Dave. , 2016. *WHY IS JPEG CONSIDERED A "LOSSY" GRAPHICS FORMAT?* 04 09. https://www.askdavetaylor.com/why_is_jpeg_considered_a_lossy_graphics_format/.

Thanikaiselvan, V., P. Arulmozhivarman, S. Subashanthini, και Rengarajan Amirtharajan. , 2013. *A Graph Theory Practice on Transformed Image: A Random Image Steganography.*

Thongkor, Kharittha, Narong Mettripun, Thitiporn Pramoun, and Thumrongrat Amornraksa. , 2013. "Image watermarking based on DWT coefficients modification for social networking services."

Tilborg, Henk C. A. van. , 2005. "One-Time Password."

Vazquez, Carlos Lopez. , 2000. *Method of inserting hidden data into digital archives comprising polygons and detection methods.*

Walia, E., και P. Jain. , 2010. «An Analysis of LSB & DCT based Steganography.»

Weaver, και Shannon. n.d. *Model of Communication.* , 1948.

Wells, Michael T. , 2009. *Mobile Image Processing on the Google Phone with the Android Operating System .*

Westfeld, Andreas, and Andreas Pfitzmann. , 2000. "Attacks on Steganographic Systems."

Zhang, Lin, Jianhua Wu, και Nanrun Zhou. , 2009. «Image Encryption with Discrete Fractional Cosine Transform and Chaos.»

Παράρτημα Α': Οδηγός χρήσης

Α.1 Εγχειρίδιο Εγκατάστασης Εφαρμογών

Για να εγκατασταθούν οι εφαρμογές θα πρέπει να γίνουν οι ενέργειες που περιγράφονται στην συνέχεια με την σειρά που αναφέρονται.

Α.1.1 Εγκατάσταση Βάσης Δεδομένων

Η βάση δεδομένων μπορεί εγκατασταθεί με τους ακόλουθους τρόπους:

1. Εκτελώντας τα SQL scripts που βρίσκονται στον φάκελο **\project\db_scripts** με την ακόλουθη σειρά:
    a. **create_database_and_tables_v0.1.sql**
    b. **create_constraints.sql**
    c. **insert_values.sql**

    Μετά την επιτυχημένη εκτέλεση των SQL scripts θα πρέπει να εισαχθούν τεστ δεδομένα στην βάση δεδομένων.

2. Εκτελώντας το SQL script Dump20160406.sql που βρίσκεται στον φάκελο **\project\db_scripts**. Με την εκτέλεση αυτού του SQL script θα δημιουργηθεί μία πανομοιότυπη βάση δεδομένων με αυτή που χρησιμοποιήσαμε κατά την ανάπτυξη της εφαρμογής μας.

Α.1.2 Παραμετροποίηση ΕΥΔΧΚ

Α.1.2.1 Αποδοχή απομακρυσμένων συνδέσεων

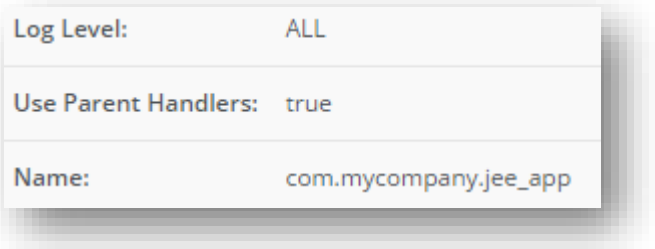
Για να μπορέσει ο ΕΥΔΧΚ να μπορεί να δέχεται απομακρυσμένες συνδέσεις (remote connections) από οποιαδήποτε διεύθυνση, θα πρέπει στο xml αρχείο **standalone.xml** που βρίσκεται στον φάκελο **standalone\configuration** να αντικαταστήσετε την εγγραφή

<inet-address value="${jboss.bind.address:127.0.0.1}"/>
για το **public interface** με την εγγραφή
<inet-address value="${jboss.bind.address:0.0.0.0}"/>.

Α.1.2.2 Logging

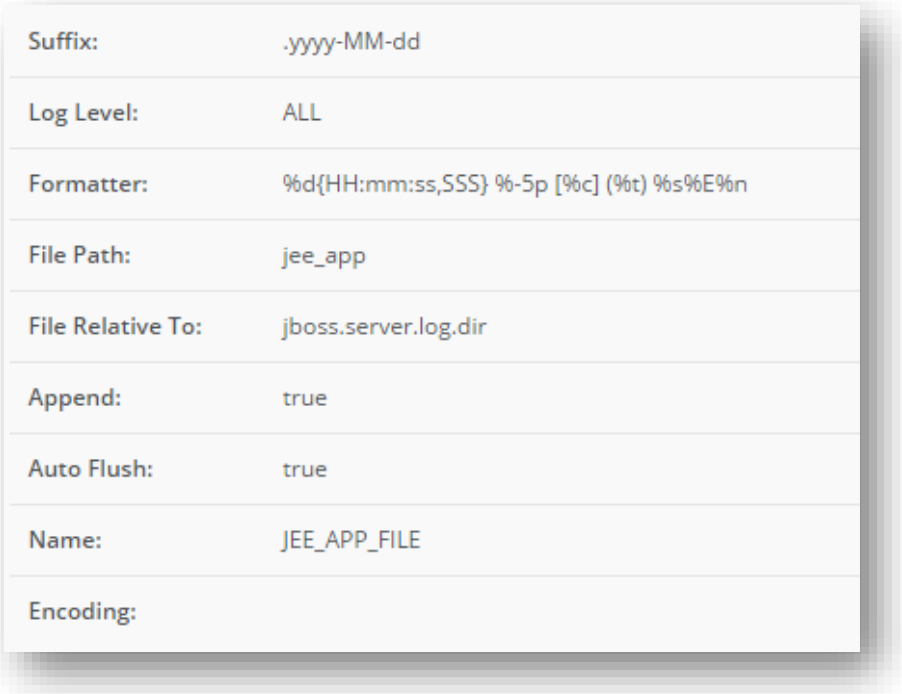
Στον ΕΥΔΧΚ (JBoss EAP 6.4.0) εκτελείται η εφαρμογή jee_app της οποίας όλες οι κλάσεις βρίσκονται κάτω από το πακέτο **com.mycompany.jee_app**. Αυτό μας δίνει την ευχέρεια να δημιουργήσουμε στον JBoss EAP το log category που

παρουσιάζεται στην Εικόνα 46 για να καταγράφονται όλα τα logs της εφαρμογής jee_app.



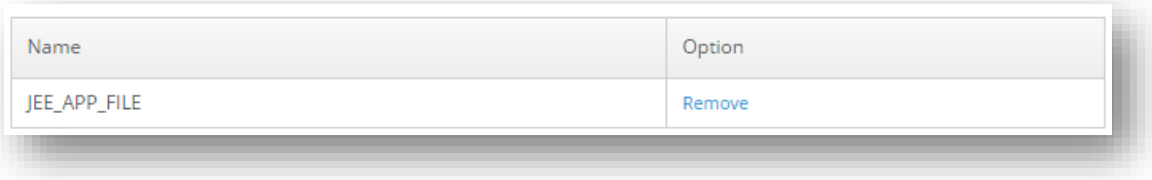
Εικόνα 46, com.mycompany.jee_app log category

Σε αυτό log category προσθέσαμε τον periodic handler που παρουσιάζεται στην Εικόνα 47.



Εικόνα 47, JEE_APP_FILE handler

Συνεπώς, όλα τα logs της εφαρμογής jee_app θα γράφονται τόσο στους handlers του root logger, γιατί είναι ενεργοποιημένη η επιλογή Use Parent Handlers του log category com.mycompany.jee_app, όσο και στον periodic handler JEE_APP_FILE (Εικόνα 48).

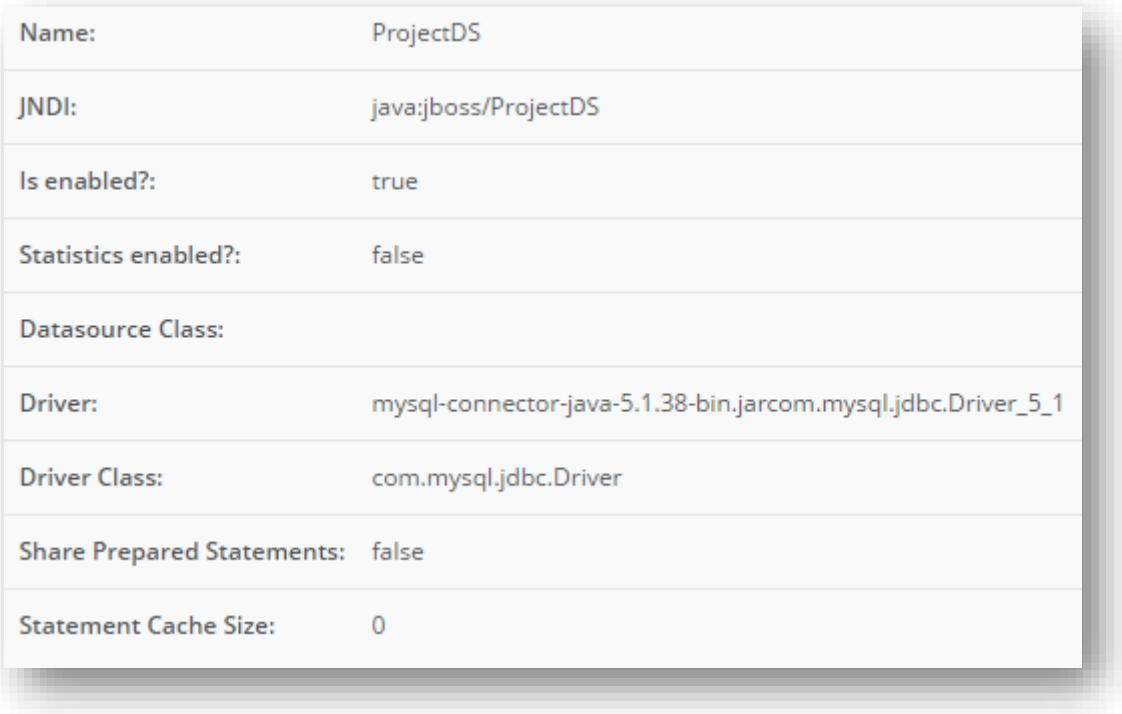


Name	Option
JEE_APP_FILE	Remove

Εικόνα 48, Handlers για Log Category com.mycompany.jee_app

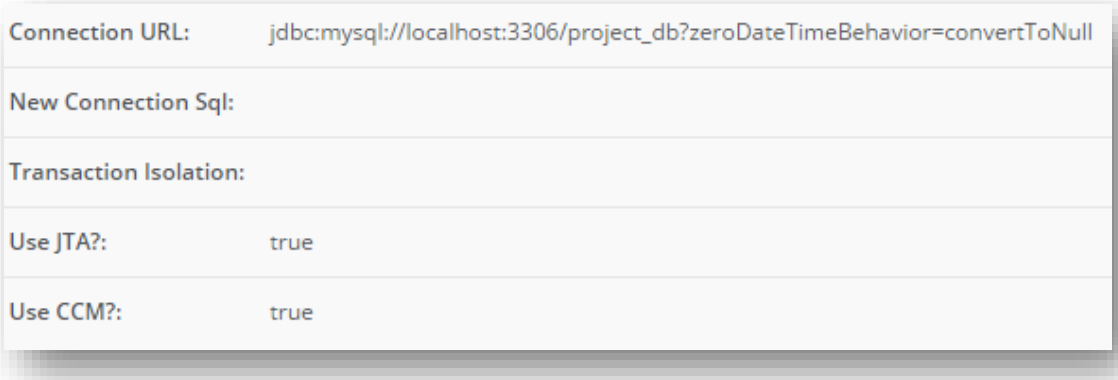
A.1.2.3 Datasource

Η εφαρμογή jee_app χρησιμοποιεί την βάση δεδομένων project_db. Ωστόσο, δεν χρησιμοποιεί την βάση δεδομένων απευθείας, αλλά διαμέσου του ΕΥΔΧΚ ο οποίος αναλαμβάνει την διαχείριση της επικοινωνίας με το RDBMS. Στις Εικόνα 49 και Εικόνα 50 παραθέτουμε την παραμετροποίηση του Datasource στον ΕΥΔΧΚ.



Name:	ProjectDS
JNDI:	java:jboss/ProjectDS
Is enabled?:	true
Statistics enabled?:	false
Datasource Class:	
Driver:	mysql-connector-java-5.1.38-bin.jarcom.mysql.jdbc.Driver_5_1
Driver Class:	com.mysql.jdbc.Driver
Share Prepared Statements:	false
Statement Cache Size:	0

Εικόνα 49, Datasource attributes



Εικόνα 50, Datasource connection

A.1.3 Εγκατάσταση JEE Εφαρμογής

Το όνομα της JEE εφαρμογής είναι **jee_app-0.1.war** και βρίσκεται στον φάκελο **\project\jee_app\v0.1\jee_app\target**. Η εφαρμογή εγκαθίσταται στον ΕΥΔΧΚ με βάση τις οδηγίες που υπάρχουν στην παράγραφο **10.2. DEPLOY WITH THE MANAGEMENT CONSOLE** του URL

https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.4/html-single/Administration_and_Configuration_Guide/#sect-Deploy_with_the_Management_Console.

A.1.4 Εγκατάσταση Mobile Εφαρμογής

Η mobile εφαρμογή **app-debug.apk** που βρίσκεται στο φάκελο **\project\mobile_app\v0.1\MobileApp\app\build\outputs\apk** θα πρέπει να εγκατασταθεί σε δύο Android συσκευές σύμφωνα με τις παρακάτω οδηγίες:

- Για την εγκατάσταση της Mobile εφαρμογής απαιτείται η ενεργοποιήση των “Unknown sources» (Συνήθης τοποθεσία ρύθμισης: Settings > Applications)
- Απαιτείται η ενεργοποίηση του USB Debugging Mode
  (Συνήθης τοποθεσία ρύθμισης: Settings > Applications > Development)
  Σε περιπτώσεις που δεν υπάρχει η επιλογή Development πρέπει να ακολουθήσουμε τα εξής βήματα: Settings > About phone > Build Number > επιλογή πολλαπλές φορές πάνω στο πλαίσιο μέχρι να εμφανιστεί το μήνυμα "You are now a developer!"
- Όταν ολοκληρώσουμε τα παραπάνω ακολουθούμε έναν από τους τρεις παρακάτω τρόπους λήψης

1) Κάνοντας χρήση της εφαρμογής του Steganography Portal στην επιλογή Get Application εγκαθιστούμε την εφαρμογή στο εκάστοτε τηλέφωνο
2) Το αρχείο app-debug.apk έχουμε δυνατότητα επίσης να το στείλουμε με την χρήση Bluetooth από τον ηλεκτρονικό υπολογιστή ή από ένα κινητό σε κάποιο άλλο
3) Με χρήση καλωδίου USB και διασύνδεση της συσκευής με τον ηλεκτρονικό υπολογιστή κάνοντας χρήση της βιβλιοθήκης adb που εμπεριέχεται στα JDK εργαλεία και εκτελώντας adb install <path/app-debug.apk>

- Επιλέγουμε το αρχείο και προχωράμε με την επιλογή Install
- Βρίσκουμε την εφαρμογή στο πλαίσιο εφαρμογών και επιλέγουμε για άνοιγμα

Α.2 Εγχειρίδιο Χρήστη

Η πρώτη επαφή του χρήστη βασίζεται στην δημιουργία νέου λογαριασμού μέσω της διαδικτυακής εφαρμογής Steganography Portal. Για είσοδο στην συγκεκριμένη ενότητα απαιτούνται δικαιώματα τα οποία μπορεί να αποκτήσει ο χρήστης, όπως περιγράφεται στην ενότητα **Error! Reference source not found.**4.3.1.2. Στην ίδια ενότητα περιγράφεται η δημιουργία «νέας φιλίας» με τουλάχιστον έναν διαφορετικό χρήστη η οποία αποτελεί προαπαιτούμενο για την ανταλλαγή στεγανογραφημένων εικόνων.

Αφού πραγματοποιηθεί η σύνδεση στην διαδικτυακή εφαρμογή Steganography Portal, δίνεται η δυνατότητα επιλογής λήψης της εφαρμογής Mobile App και αποθήκευσης στην κινητή συσκευή. Για την εγκατάσταση της εφαρμογής ακολουθούμε τα βήματα που περιγράφονται στο Παράρτημα Α.1.4 Αφού ολοκληρωθεί η εγκατάσταση της εφαρμογής, ο χρήστης μπορεί άμεσα να τη χρησιμοποιήσει. Η χρήση της mobile εφαρμογής δεν απαιτεί κάποια ιδιαίτερη δεξιότητα από τον χρήστη, πέρα από την προσοχή του κατά την ταυτοποίηση (login) (Εικόνα 26). Συγκεκριμένα, κατά την εκκίνηση της εφαρμογής ο χρήστης θα πρέπει να εισαγάγει το όνομα (username) το κωδικό πρόσβασης (password), καθώς και την IP διεύθυνση του server που φιλοξενεί την Εφαρμογή Υπηρεσιών Διαχείρισης Χρηστών & Κρυπτογράφησης (ΕΥΔΧΚ).

Αφού πιστοποιηθεί ο χρήστης και εισέλθει στη κεντρική οθόνη της mobile εφαρμογής μπορεί να επιλέξει ανάμεσα στις δυο βασικές λειτουργίες:

- Αποστολή φωτογραφίας (Send Photo)
- Λήψη φωτογραφίας (Receive Photo)

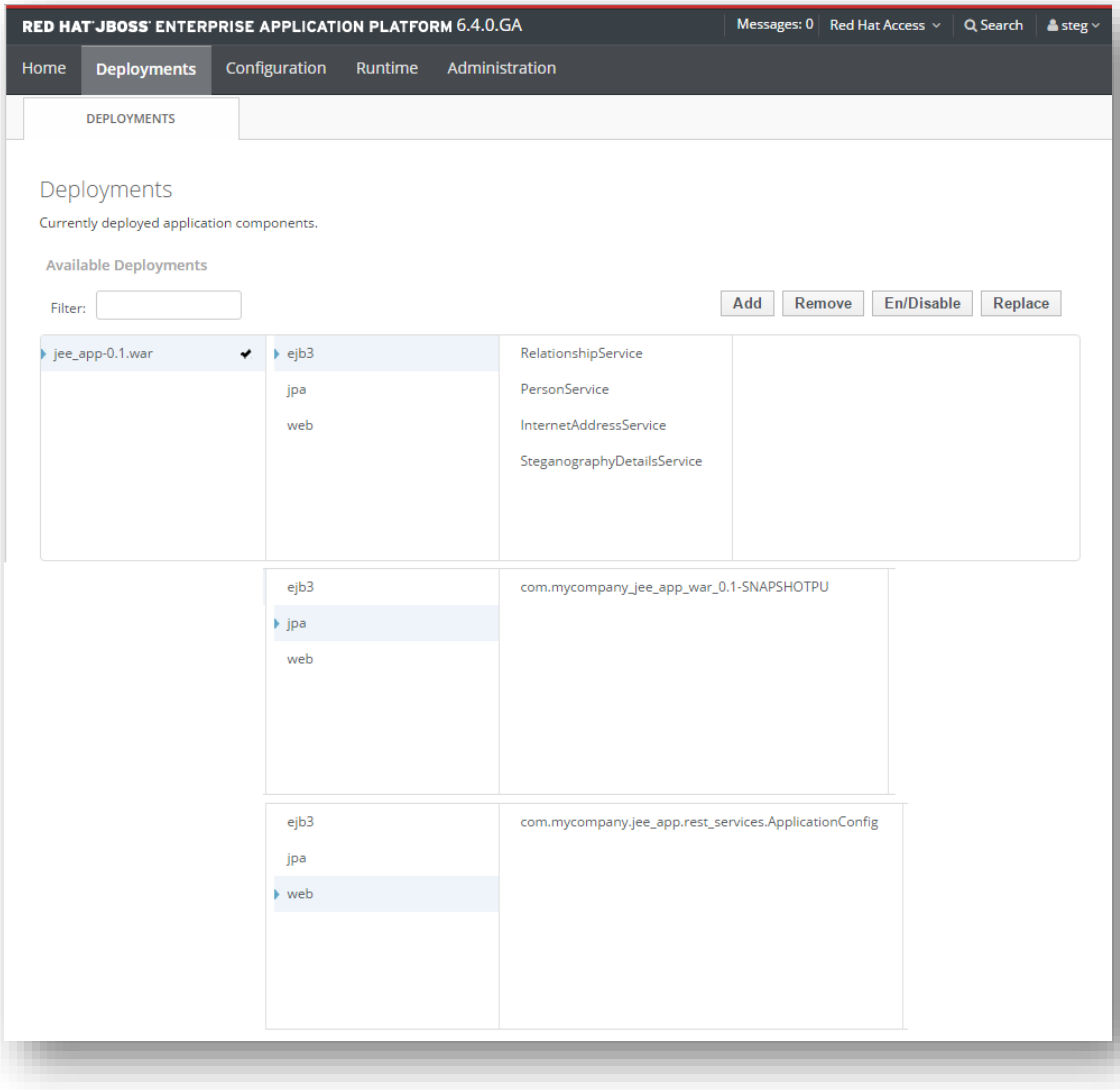
Με βάση την επιθυμία του χρήστη προχωρά στην επιλογή της ανάλογης λειτουργίας και μεταβαίνει στην επόμενη οθόνη.

Στην περίπτωση που ο χρήστης επιλέξει αποστολή μιας στεγανογραφημένης εικόνας με κρυφό μήνυμα, στην οθόνη «Send Photo» επιλέγει το πλήκτρο Camera ενεργοποιώντας τη λειτουργία λήψης εικόνας μέσω της ενσωματωμένης κάμερας του κινητού τηλεφώνου. Αφού επιλεγεί η εικόνα ο χρήστης πληκτρολογεί το μήνυμα στο πεδίο “Enter Message”. Κάτω από το μήνυμα δίνεται η δυνατότητα επιλογής του χρήστη στον οποίο θα αποσταλεί η στεγανογραφημένη εικόνα με τη βοήθεια ενός επιλογέα κίνησης (scroll down menu).

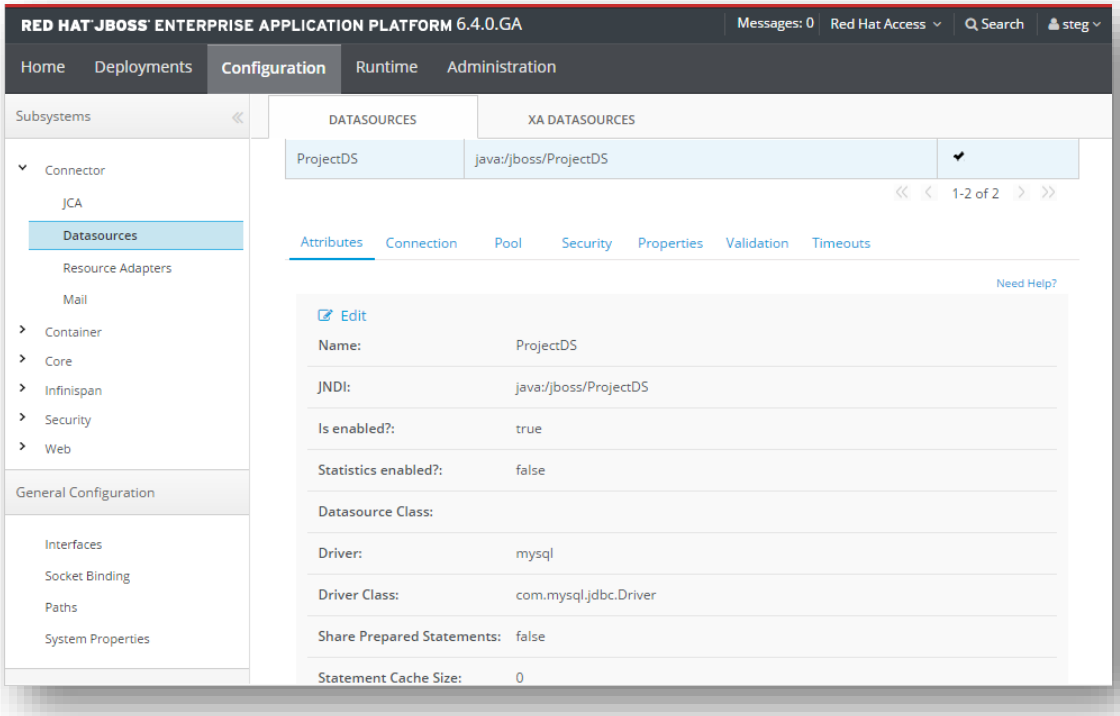
Αν η λειτουργία που επιλέξει ο χρήστης είναι η λήψη μιας στεγανογραφημένης εικόνας “Receive Photo”, τότε η συσκευή τίθεται σε κατάσταση αναμονής για λήψη στεγανογραφημένων φωτογραφιών από χρήστες με τους οποίους υπάρχει σχέση φιλίας (relationship).

Παράρτημα Β': Αποτυπώσεις

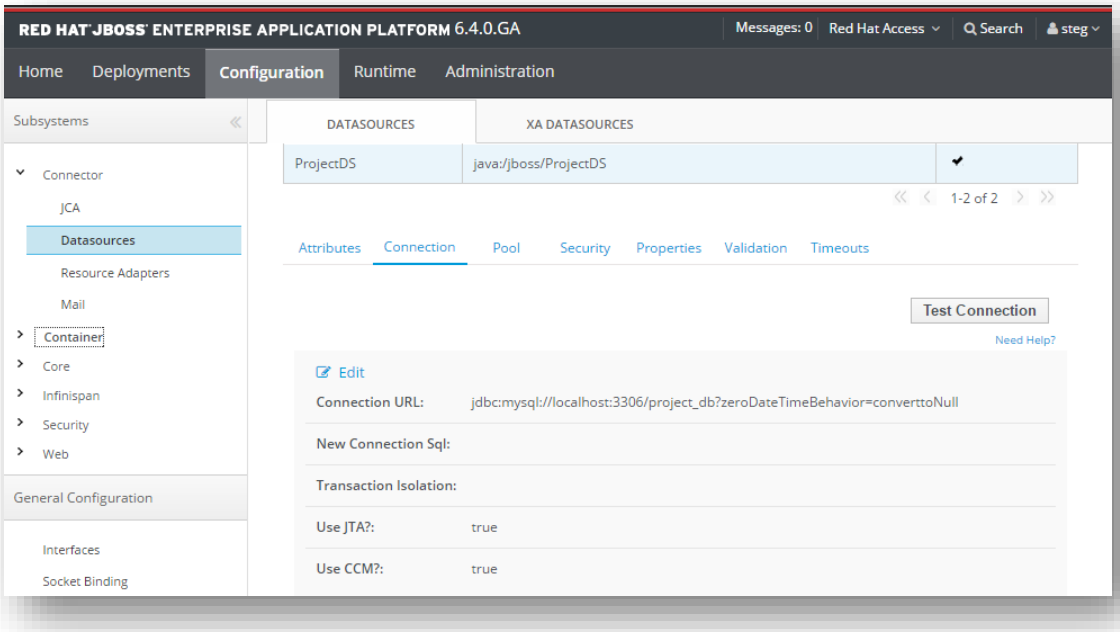
Στη συνέχεια παρατίθενται ενδεικτικές αποτυπώσεις των οθονών ρυθμίσεων του λογισμικού εξυπηρετητή εφαρμογών JBoss που χρησιμοποιήσαμε, καθώς και του προσομοιωτή Android Studio.



Εικόνα 51, Redhat jboss Deployments Page.

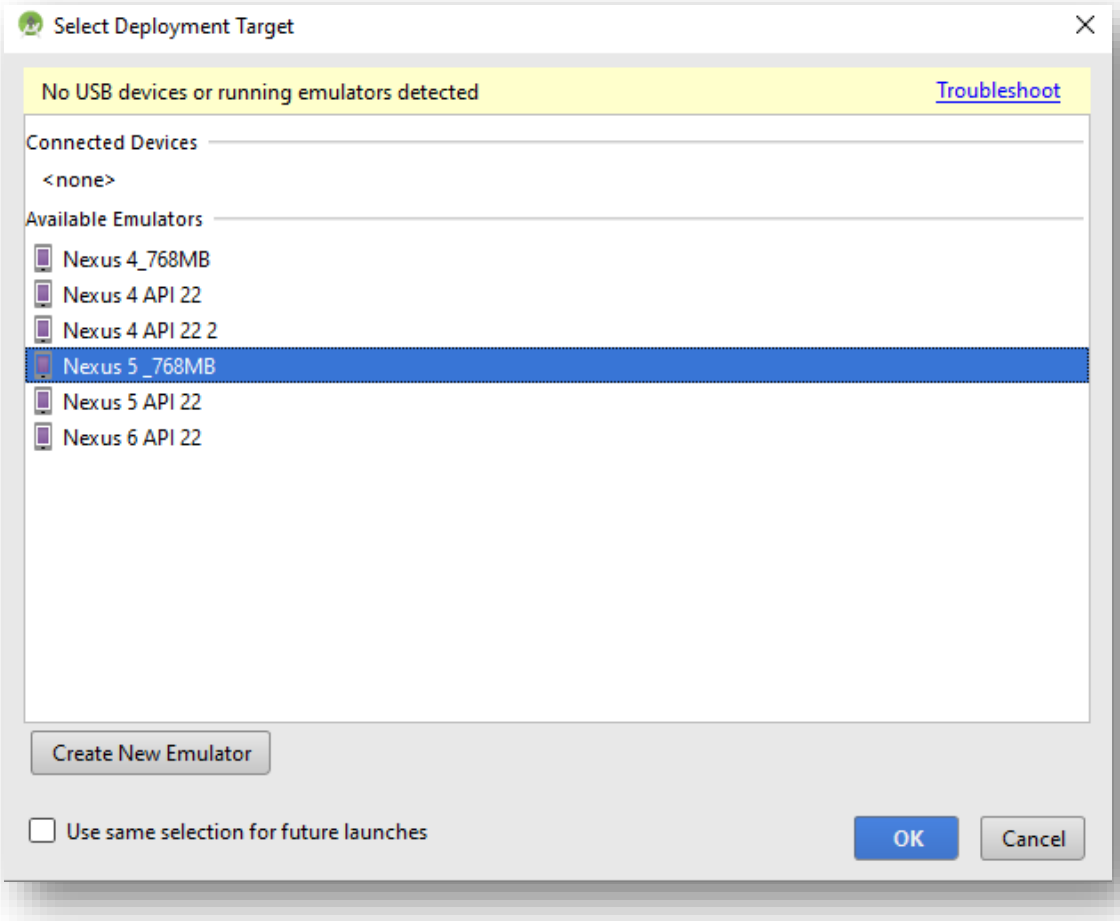


Εικόνα 52, Redhat jboss Settings 1.

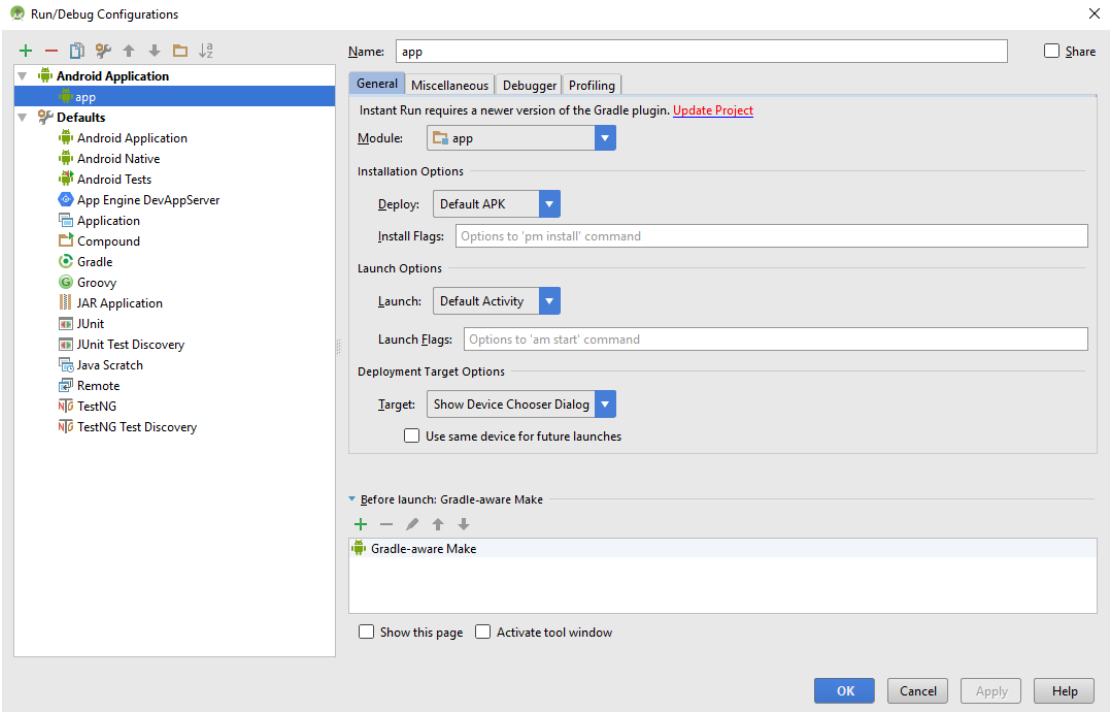


Εικόνα 53, Redhat jboss Settings 2.

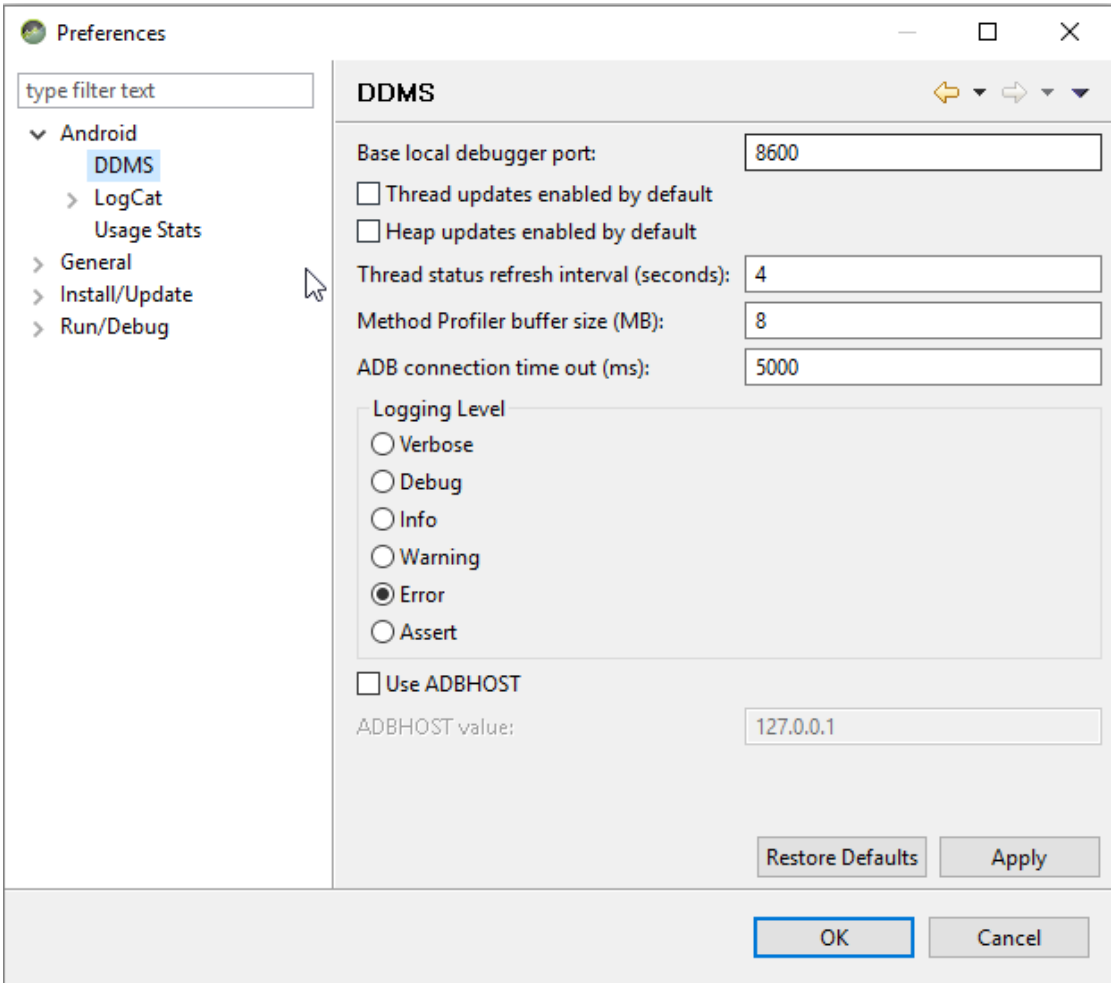
Εικόνα 54, Έκδοση Android Studio



Εικόνα 55, Android Studio Emulators



Εικόνα 56, Debug Configuration



Εικόνα 57, Android Device Monitor preferences