ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής κρυπτογραφημένης P2P επικοινωνίας για συσκευές Android

Σπουδαστής: Γεώργιος Γκόγκας

Εισηγητής: Πάρις Μαστοροκώστας, *Καθηγητής*

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΩΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Γεώργιος Γκόγκας**, με αριθμό μητρώου **39100**, φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσής της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου προς τον καθηγητή μου Πάρη Μαστοροκώστα για τη δυνατότητα που μου έδωσε να πραγματοποιήσω την πτυχιακή μου εργασία και για την πολύτιμη βοήθειά του σε κάθε στάδιο που την χρειάστηκα καθώς και για τον χρόνο που διέθεσε για την περάτωση της παρούσας εργασίας. Οι υποδείξεις του και οι συμβουλές του ήταν για μένα εξαιρετικά πολύτιμες.

Επίσης όλους τους καθηγητές μου του τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής για τις γνώσεις που μου μεταλάμπαδευσαν καθ όλη τη διάρκεια της φοίτησης μου, γνώσεις οι οποίες θα είναι πραγματικά και ισχυρά εφόδια για την μετέπειτα πορεία μου.

Τέλος να ευχαριστήσω την οικογένεια μου για τη στήριξη, την εμπιστοσύνη και την πίστη που μου έδειξαν όλα αυτά τα χρόνια.

ΠΕΡΙΛΗΨΗ

Αντικείμενο της εργασίας είναι η υλοποίηση μιας εφαρμογής για κρυπτογραφημένη P2P επικοινωνία σε συσκευές Android. Η εφαρμογή θα υλοποιηθεί με χρήση της γλώσσας Java και θα χρησιμοποιηθούν τα πακέτα java.security, JXTA/JXSE και JavaFX. Ο χρήστης θα έχει τη δυνατότητα να επιλέγει (α) συνομιλητές με χρήση ψευδώνυμων και (β) τη μέθοδο κρυπτογράφησης (ενδεικτικά αναφέρονται οι RSA, AES, TripleDES – DESede και Blowfish).

ABSTRACT

The aim of this paper is the implementation of an application for encrypted P2P communication for Android devices. The application will be implemented with the use of Java programming language and the packages java.security, JXTA/JXSE and JavaFX. The user will be able to select (a) other users through their aliases to communicate with and (b) the encryption method (for example RSA, AES, TripleDES - DESede και Blowfish)

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή

Στη σύγχρονη εποχή η ανάγκη για επικοινωνία έχει αυξηθεί δραστικά και επηρεάζει τόσο τους πολίτες στην καθημερινότητα τους, όσο και τους διάφορους οργανισμούς και επιχειρήσεις οι οποίες πρέπει να βρίσκονται σε διαρκεί επαφή για την ανάπτυξη τους και εξυπηρέτηση του κοινού τους.

Η ανάγκη για επικοινωνία μπορεί να καλυφθεί εύκολα από την μεγάλη πληθώρα εφαρμογών που υπάρχουν. Οι εφαρμογές αυτές απευθύνονται σε διάφορα είδη κοινού σύμφωνα με τις παράπλευρες ανάγκες που καλύπτουν.

Μία από τις πιο σημαντικές ανάγκες που επηρεάζουν την επικοινωνία είναι η ασφάλεια. Η εξασφάλιση μιας ασφαλούς επικοινωνίας είναι τόσο σημαντική που από μόνη της μπορεί να αποτελέσει ένα κριτήριο για την επιτυχία οποιασδήποτε δραστηριότητας.

Στα πλαίσια της εξασφάλισης μιας ασφαλούς επικοινωνίας, έχουν πραγματοποιηθεί πολλές βελτιώσεις με πρωτοβουλίες τόσο από τον δημόσιο τομέα όσο και από τον ιδιωτικό τομέα. Ένα γεγονός που μπορεί να εφησυχάσει τους χρήστες διαδικτυακών εφαρμογών επικοινωνίας, αλλά και αναμφισβήτητη απόδειξη για την διαρκή εξέλιξη και ανάπτυξη νέων μεθόδων ασφάλειας.

Η ανάπτυξη θα πρέπει να είναι πολύμορφη και να στοχεύει παράλληλα στη δημιουργία νέων επιστημονικών προτύπων αλλά και στην ανάπτυξη νέων τρόπων βελτίωσης και αξιοποίησης υπάρχοντών τεχνολογιών.

Ο καλύτερος τρόπος για την εξασφάλιση μίας ασφαλούς επικοινωνίας είναι να γίνει προσωπική ευθύνη όλων των συμμετεχόντων. Αυτό επιτυγχάνεται δίνοντας στους χρήστες τη δυνατότητα να προσαρμόσουν τον τρόπο που επικοινωνούν.

Ένας τρόπος προσαρμογής των επικοινωνιών είναι η εγκατάσταση πιστοποιητικών όπως τα **SSL** και **TLS** ή χρήση κάποιων **Cypher** για την κρυπτογράφηση της μεταδιδόμενης πληροφορίας. Ένας άλλος τρόπος για την τροποποίηση των επικοινωνιών είναι η χρήση ιδιωτικών εικονικών δικτύων, τα οποία επιτρέπουν την άμεση επικοινωνία μεταξύ χρηστών χωρίς μεσάζοντες να παρακολουθούν τη συζήτηση και σταθερές συνδέσεις που μπορούν να γίνουν το μέσο παρακολούθησης.

Ο πιο σύγχρονος και αποδοτικός τρόπος επικοινωνίας με ασφάλεια είναι η χρήση ιδιωτικών δικτύων. Η λύση αυτή είναι ο πιο σίγουρος τρόπος για την διασφάλιση της επικοινωνίας και γίνεται πιο προσιτός αφού το κόστος αγοράς και διατήρησης ισχυρού οικιακού εξοπλισμού ή επαγγελματικού εξοπλισμού μειώνεται με κάθε μέρα που περνά.

Σε αυτή την εργασία μελετάται η χρήση ιδιόκτητων φυσικών και εικονικών δικτύων, P2P συνδέσεων και κρυπτογράφησης με προσαρμοζόμενα κλειδιά που δημιουργούνται από τους χρήστες. Η μελέτη αυτή αποδεικνύει πως η δημιουργία και χρήση κρυπτογραφικών μεθόδων και δομών είναι εφικτή τόσο για τους οικιακούς υπολογιστές όσο και στα σύγχρονα **Smartphones** με επίκεντρο τα **Android** συστήματα.

Για την περάτωση της εργασίας και την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν τα **Gluon** και **Gradle** για γρήγορη ανάπτυξη και δημοσίευση της εφαρμογής και μελετήθηκε η βιβλιοθήκη **JXTA/JXSE** ως προς τη βιωσιμότητα στο **Android** περιβάλλον

Κεφάλαιο 2. Ιστορικές αναφορές

Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού (OOPs) για ενσωματωμένα συστήματα **(embedded systems)** και δημοσιεύτηκε στις 23 Ιανουαρίου 1996 με το πρώτο **Java Development Kit (JDK)** από τους **James Gosling**, **Mike Sheridan** και **Patrick Naughton** της **Green Team** των **Sun Microsystems**.

Αντικειμενοστραφής σημαίνει πως το πρόγραμμα οργανώνεται σε οντότητες που ονομάζονται αντικείμενα. Τα αντικείμενα περιέχουν δεδομένα και συμπεριφορά (ενέργειες και λειτουργίες). Ένα αντικείμενο μπορεί να είναι μια αυτοτελή οντότητα ή ο συνδυασμός άλλων αντικειμένων για την υλοποίηση περιπλοκότερων λειτουργιών.

Στη Java τα αντικείμενα δημιουργούνται από ένα προσχέδιο το οποίο ονομάζεται Κλάση **(Class)**. Η Κλάση περιέχει τη δομή που χρειάζεται το αντικείμενο για να συγκρατήσει τα απαραίτητα για την λειτουργία δεδομένα του, καθώς και τις μεθόδους **(functions)** οι οποίες απαρτίζουν τη λειτουργία και συμπεριφορά του αντικειμένου.

Κύριο στοιχείο κάθε σύγχρονης γλώσσας προγραμματισμού είναι η δυνατότητα να δημιουργεί γραφικές διεπαφές για τον χρήστη. Η Java το κατάφερε μέσω του Abstract Window Toolkit (AWT) API. Το API χρησιμοποιούσε τη μηχανή γραφικών του λειτουργικού για την αποτύπωση του GUI. Αυτό είχε σαν αποτέλεσμα την έλλειψη ομοιογένειας μεταξύ λειτουργικών συστημάτων.

Με την δεύτερη επίσημη έκδοση της Java (V1.2) η Sun Microsystems παρουσίασε στον κόσμο το Swing API τον Δεκέμβριο του 1998. Το Swing API είναι κομμάτι του Java Foundation Classes, το οποίο μεταξύ άλλων περιλαμβάνει και το AWT API. Τα δύο αυτά συστήματα

σχεδιάστηκαν να δουλεύουν συνεργατικά. Το AWT επέτρεπε στους προγραμματιστές να σχεδιάζουν εφαρμογές με μεγαλύτερη απόκριση λόγο των events του συστήματος και το Swing τους πρόσφερε την ποικιλία και προσαρμοστικότητα που μέχρι τότε έλειπε.

JavaFX

Η εξέλιξη στη σχεδίαση και ανάπτυξη διεπαφών ήρθε με την ανάπτυξη και δημοσίευση του JavaFx τον Δεκέμβρη του 2008. Αρχικά δημιουργήθηκε για να διευκολύνει τη σχεδίαση διεπαφών για τα Java Applets. Ενώ εξετέλεσε τον σκοπό του, η δεύτερη επίσημη έκδοση του API διακόπτει όλα τα στοιχεία που έχουν να κάνουν με την ανάπτυξη Java Applet.

Η απόφαση της Sun Microsystems να διακόψει το JavaFx στους φυλλομετρητές οδήγησε στην ανάπτυξη του JavaFX ως API για τη Java στα συστήματα γενικής χρήσης της εποχής. Αυτό από μόνο του ήταν αρκετό για να ενοποιήσει τα δύο αυτά συστήματα και να επιταχύνει την ανάπτυξη εφαρμογών αφού τα στοιχεία διεπαφής και λειτουργικότητας μοιράζονταν την ίδια γλώσσα και σύνταξη.

Το νέο API είχε πολλές δυνατότητες όπως hardware accelerated graphics, υποστήριξη αναπαραγωγής διαδικτυακών μέσω και σελίδων. Το πιο σημαντικό από όλα είναι η περιγραφική γλώσσα FXML, μια εκδοχή της XML η οποία επιτρέπει την ταχεία ανάπτυξη εφαρμογών.

Java on mobile

Όπως έχουμε ήδη αναφέρει η Java είναι μια αντικειμενοστραφής γλώσσα για embedded systems. Ο όρος embedded system αναφέρεται σε συστήματα τα οποία φέρουν το λειτουργικό τους στον εξοπλισμό που τα απαρτίζει. Αυτά τα συστήματα συνήθως είναι φορητές συσκευές και σχεδιάζονται για μια πληθώρα λειτουργιών.

Τέτοιες συσκευές είναι και τα κινητά τηλέφωνα, στα οποία η Java είχε από νωρίς μια πολύ ενεργή παρουσία. Μάλιστα από το 1996 υπήρξαν αρκετές εκδόσεις της γλώσσας που παρείχαν τη δυνατότητα ανάπτυξης εφαρμογών για κινητά τηλέφωνα. Την άνοιξη του 1998 η Sun Microsystems δημιούργησε το Java Mobile Edition **(J2ME)** το οποίο ομογενοποιεί τις προηγούμενες προσπάθειές τους για ένα Java API για τις κινητές συσκευές.

Σήμερα η Oracle Java Mobile Edition υποστηρίζει μια μεγάλη ποικιλία συσκευών που στηρίζονται στους νέους μικροεπεξεργαστές όπως οι Cortex M4/M7 και development boards όπως το Intel Galileo Gen. 2 και το Raspberry Pi.

JavaFX Mobile – OpenJFX

Ενώ η Java χρησιμοποιούνταν στα κινητά από το 1996, το JavaFX API δημιουργήθηκε αποκλειστικά για χρήση στο διαδίκτυο και τους σταθερούς υπολογιστές, αφήνοντας το J2ME για τα κινητά. Με την έκδοση 1.2 του JavaFx η Oracle εκδίδει το JavaFX Mobile για Symbian και Windows κινητά.

Το 2018 η Oracle μετέφερε το JavaFX API στο OpenJDK πακέτο με την ονομασία OpenJFX. Το πακέτο αυτό καθώς και το καινούργιο API αποτελούν Open Source Community Projects. Αυτή της η κίνηση επιτάχυνε την ανάπτυξή τους και την εισαγωγή νέων στοιχείων.

Η υιοθέτηση του JavaFX από την Open Source κοινότητα είχε ξεκινήσει νωρίτερα με την ανάπτυξη του JavaFXPorts, το οποίο επέτρεπε την χρήση του API σε συσκευές iOS της Apple και Android. Σαν αποτέλεσμα αυτής της ανάπτυξης ήταν η εμφάνιση εργαλείων όπως το **Gluon** το οποίο χρησιμοποιεί το **Gradle** και αναλαμβάνει να μετατρέψει τον κώδικα που απευθύνεται στους υπολογιστές σε κώδικα για κινητά τηλέφωνα.

Smartphones

Ο όρος smartphone είναι πιο παλιός απ` όσο ο περισσότερος κόσμος γνωρίζει. Τα πρώτα smartphones εμφανίστηκαν το 1992. Τα τηλέφωνα αυτά είχαν κάποια στοιχεία τα οποία ακόμα φέρουν τα σύγχρονα κινητά. Τα πιο σημαντικά στοιχεία είναι η οθόνη επαφής **(touch screen),** ηλεκτρονικό ταχυδρομείο **(email)** και εφαρμογές **(applications)**.

Blackberry

Αν και τα πρώτα smartphones φέρουν πολλά από τα γνώριμα χαρακτηριστικά των σύγχρονων κινητών, αλλά δεν ήταν οι πολυμεσικές συσκευές **(multimedia devices)** που κυκλοφορούν σήμερα. Δε θα ήταν έτσι αν η Blackberry δεν είχε παρουσιάσει το **BlackBerry 5810** στις 4 Μαρτίου 2002. Το πρώτο κινητό με Java λειτουργικό, δυνατότητες αναπαραγωγής

μουσικής και σύνδεσης στο διαδίκτυο **(internet)**.

Android

Την επιτυχία της Blackberry θα ακολουθήσει η Google με το Android το 2008 μέσα από το **T-Mobile G1** της **HTC**. Η εξέλιξη της τεχνολογίας οδήγησε στο Android V1.5 το 2009, την πρώτη έκδοση η οποία πρόσφερε τη δυνατότητα εγκατάστασης και χρήσης εφαρμογών κατασκευαστών πέρα της Google.

Android SDK

Με την έκδοση του Android V1.5 αρχίζει η ανάπτυξη εφαρμογών από τους χρήστες του λειτουργικού. Αυτό καθίσταται εφικτό από την έκδοση του Android SDK από την Google τον Οκτώβριο του 2009. Το SDK παρέχει τον απαραίτητο debugger, βιβλιοθήκες καθώς και τον εξομοιωτή για τις αναπτυσσόμενες εφαρμογές. Επίσης το SDK παρέχει πολλά εργαλεία που προσφέρουν διάφορες μεθόδους ανάπτυξης και αποσφαλμάτωσης, όπως Android Development Tools (ADT) plugin για το Eclipse IDE.

Αξίζει να σημειωθεί πως η ανάπτυξη των εφαρμογών στις πρώτες μέρες του Android SDK στηρίζονταν αποκλειστικά στη χρήση των επεκτάσεων του ADT για τα διάφορα IDEs που κυκλοφορούσαν εκείνη την περίοδο. Αυτό ήρθε ν᾽ αλλάξει όταν τον Δεκέμβρη του 2014 η Google σε συνεργασία με την IntelliJ κυκλοφόρησαν το πρώτο IDE αποκλειστικά για το Android με το όνομα Android Studio.

Gradle

Το Gradle είναι ένα εργαλείο που ενώ κυκλοφορούσε από το 2007, δεν είδε την πρώτη επίσημη κυκλοφορία του μέχρι το 2011. Παρόλη την πολύχρονη ανάπτυξη του, δεν άργησε να γίνει το επικρατέστερο εργαλείο αυτοματοποίησης της μεθόδου ανάπτυξης εφαρμογών.

Ο λόγος που επικράτησε είναι η ικανότητα του να διευκολύνει και να επιταχύνει της επαναλαμβανόμενες διαδικασίες. Διαδικασίες όπως η διαχείριση των dependencies, την εκτέλεση δοκιμαστικών μεθόδων και μεθόδων αποσφαλμάτωσης καθώς και του πακεταρίσματος (Packaging/building) της εφαρμογής για οποιοδήποτε λειτουργικό σύστημα.

Gluon

Σχεδιασμένο από την ομώνυμη εταιρία το Gluon αξιοποιεί τις δυνατότητες του Gradle για να μετατρέπει τις εφαρμογές των σταθερών υπολογιστών σε εφαρμογές κινητών. Το Gluon παρέχει εργαλεία για τον ευκολότερο σχεδιασμό εφαρμογών για όλες τις πλατφόρμες.

Το Gluon δρα σαν τον πυρήνα της εφαρμογής και γύρω του στήνονται τα υπόλοιπα στοιχεία της εφαρμογής. Η ανταλλαγή δεδομένων και εντολών από και προς τον Unix πυρήνα του Android περνάει από το Gluon κατευθυνόμενη στα υπόλοιπα επίπεδα της εφαρμογής. Επίσης αν η εφαρμογή χρειάζεται πρόσβαση στους πόρους και τις υπηρεσίες του πυρήνα θα πρέπει να καλέσει το Gluon.

Όλα αυτά τα χαρακτηριστικά είναι που κάνουν το Gluon το ιδανικό εργαλείο για την γρήγορη ανάπτυξη εφαρμογών για υπολογιστές και κινητά. Παρόλα αυτά, το Gluon παραμένει ένα προϊόν και επαγγελματικό εργαλείο το οποίο σε αντίθεση με όλα το προαναφερθέντα είναι επί πληρωμή. Συνεπώς η ανάπτυξη και έκδοση εφαρμογών κοστίζει μια ετήσια συνδρομή. Η συνδρομή μπορεί να αγνοηθεί αλλά θα παρουσιάζεται μια οθόνη ειδοποίησης όπου θα παροτρύνει στην αγορά μιας άδειας.

Κρυπτογράφηση

Η κρυπτογράφηση είναι η επιστήμη που ασχολείται με την απόκρυψη της πληροφορίας. Όποια στιγμή υπάρχουν αυξημένες απαιτήσεις για ασφάλεια η κρυπτογράφηση είναι η πιο συνετή κίνηση. Οι χρήστες των κρυπτογραφικών μεθόδων έχουν τη δυνατότητα να αποθηκεύουν και να ανταλλάσσουν κρυπτογραφημένα αρχεία και μηνύματα εξασφαλίζοντας έτσι την ασφάλεια και αποτρέποντας την κλοπή τους.

Υπάρχουν πολλές ιστορικές αναφορές για την κρυπτογράφηση καθ` όλη την ιστορία του ανθρώπου. Ο Ρωμαίος ιστορικός Suetonius αναφέρεται στον αλγόριθμο του Ιούλιου Καίσαρα, έναν απλό στην ουσία αλγόριθμο που λέγετε ότι χρησιμοποιούσε ο Καίσαρας για να επικοινωνεί με τους αξιωματικούς του. Η μέθοδος κρυπτογράφησης είναι ένας αλγόριθμος μετατόπισης. Απαρτίζεται από δύο μέρη, το πρώτο είναι το μήνυμα προς κρυπτογράφηση και το δεύτερο είναι το ποσοστό μετατόπισης. Αν για παράδειγμα το μήνυμα είναι “ABC” και το ποσοστό μετατόπισης είναι 5, τότε το κάθε γράμμα του μηνύματος θα πρέπει να μετατοπιστεί

5 θέσεις δεξιά και το αποτέλεσμα θα είναι “EFG”.

Ένας άλλος γνωστός ιστορικός αλγόριθμος, είναι ο αλγόριθμος του Vigenere, ο οποίος δημιουργήθηκε από τον Γάλλο Blaise de Vigenere τον 16° αιώνα. Ο αλγόριθμος αποτελεί μια βελτίωση του αλγόριθμου του Καίσαρα. Η διαδικασία κρυπτογράφησης απαιτεί ένα μήνυμα και μια λέξη-φράση κλειδί. Για παράδειγμα, έστω το μήνυμα (A) είναι “Hello” και η λέξη κλειδί (B) είναι “Apple”, τότε το κρυπτογραφημένο μήνυμα υπολογίζεται από την πρόσθεση των A και B από τον πίνακα [A1+B1,A2+B2,A3+B3,A4+B4]. Οπότε το κρυπτογραφημένο μήνυμα είναι “iubxd”.

Σε αντίθεση με τον αλγόριθμο του Καίσαρα, ο αλγόριθμος του Vigenere απαιτεί το μήνυμα και το κλειδί να έχουν το ίδιο μήκος. Οπότε αν το μήνυμα είναι “Hello world” και το κλειδί είναι “Apple”, δε μπορεί να γίνει κρυπτογράφηση με αυτόν τον αλγόριθμο. Το πρόβλημα μπορεί να λυθεί αυξάνοντας το μήκος του κλειδιού. Ένας τρόπος να αυξηθεί το κλειδί χωρίς την εισαγωγή νέας πληροφορίας είναι η επανάληψη του μέχρι και τα δύο μήκη να είναι ίσα. Έτσι η κρυπτογράφηση του μηνύματος θα γίνει μέσα από το κλειδί “AppleApple” και το κρυπτογραφημένο μήνυμα θα είναι “iubxtxehxi”.

Σημείωση*: τα κενά δεν χρησιμοποιούνται σε κανέναν από τους δύο αλγορίθμους.*

Java Security Package

Το Java Security πακέτο προστέθηκε στη Java με την έκδοση 1.2 και περιέχει όλες τις μεθόδους για την κρυπτογράφηση και την πιστοποίηση των στοιχείων του χρήστη. Το πακέτο προσφέρει κάποιες αφηρημένες (Abstract) Classes που λειτουργούν ως οδηγίες για την υλοποίηση κρυπτογραφικών εργαλείων.

Την υλοποίηση αναλαμβάνουν συστήματα που ονομάζονται Providers. Η Sun Microsystems έχει δύο υλοποιήσεις. Την προεπιλεγμένη υλοποίηση ονομαζόμενη “SUN provider” η οποία υποστηρίζει τον DSA αλγόριθμο και την Java Cryptography Extension (JCE) η οποία υποστηρίζει DiffieHellman αλγόριθμο.

Πέρα από τους δύο Providers που προσφέρει η Java, οι προγραμματιστές μπορούν να σχεδιάσουν του δικούς τους Providers ή να χρησιμοποιήσουν άλλων προγραμματιστών. Κάθε κρυπτογραφικός αλγόριθμος μπορεί να έχει τους δικούς του Providers και ο προγραμματιστής

μπορεί να χρησιμοποιήσει όποιους και όσους χρειάζεται. Η δυνατότητα της Java να χρησιμοποιεί πολλαπλούς Providers για τις κρυπτογραφικές μεθόδους επιτρέπει στους χρήστες να προσαρμόζουν τις εφαρμογές τους και να αυξάνουν την απόδοσή τους.

Peer to Peer (P2P)

Τα P2P δίκτυα, είναι δίκτυα υπολογιστικών συστημάτων (Η/Υ, Κινητά τηλέφωνα, tablets κτλπ) στα οποία οι κόμβοι (υπολογιστικά συστήματα) λειτουργούν ταυτόχρονα ως Clients και Servers. Τα P2P δίκτυα χαρακτηρίζονται ως Overlay δίκτυα, γιατί στηρίζονται σε άλλα δίκτυα, όπως το Internet και το τηλεφωνικό δίκτυο.

Τα Overlay δίκτυα στηρίζονται επί το πλείστον στο internet και τις διάφορες παραλλαγές του. Τα δίκτυα αυτά διατηρούν τους δικούς τους κανόνες λειτουργίας και επικοινωνίας. Η επικοινωνία στηρίζεται στη χρήση "Εικονικών" και "Λογικών" Συνδέσμων μεταξύ των κόμβων. Αυτοί οι σύνδεσμοι αντιστοιχούν στο μονοπάτι το οποίο πρέπει να διανύσουν τα δεδομένα από τον έναν κόμβο στον άλλον. Τα μονοπάτια που περιγράφουν οι σύνδεσμοι αντιστοιχούν σε φυσικούς συνδέσμους στο δίκτυο στο οποίο στηρίζεται το P2P δίκτυο.

JXTA/JXSE

Με την άνοδο των P2P δικτύων και εφαρμογών η Java δημιούργησε το JXTA πρωτόκολλο. Το πρωτόκολλο επιτρέπει στις συνδεδεμένες συσκευές να επικοινωνούν μέσω XML μηνυμάτων.

Οι εφαρμογές που υλοποιούνται με το JXTA μπορούν να λειτουργούν ως Clients και Servers. Ασχέτως του ρόλου που έχει η εφαρμογή στο δίκτυο, μπορεί αυτόνομα να τον αλλάξει ή να τους χρησιμοποιήσει και τους δύο παράλληλα ανάλογα με τις ανάγκες της.

Όλοι οι κόμβοι που έχουν τον ρόλο του Client χρησιμοποιούν ένα σύστημα "διαφήμισης" για να προωθήσουν την ταυτότητα τους στους Servers στους οποίους είναι συνδεδεμένοι. Μέσω αυτού του συστήματος οι εφαρμογές και οι χρήστες μπορούν να δημιουργήσουν και να χρησιμοποιήσουν διάφορα κανάλια επικοινωνίας.

Παρόλη την εξέλιξή και την ανάπτυξη του, στο σημείο να μετατραπεί σε βιβλιοθήκη του Android με το όνομα PeerDroid, η Oracle αποφάσισε να αποσύρει το πρωτόκολλο.

Κεφάλαιο 3. Τεχνικοί περιορισμοί

1. Μέσο μετάδοσης

Σε αυτό το στάδιο ανάπτυξης της εφαρμογής η επικοινωνία είναι περιορισμένη μέσω **WIFI** και **Ethernet**. Αυτό οφείλεται στο γεγονός πως η εφαρμογή αναπτύσσεται με την χρήση **οικιακού εξοπλισμού**, ο οποίος υπόκειται σε υψηλά στάνταρ ασφάλειας και περιορισμένων διαχειριστικών δυνατοτήτων για λόγους κόστους.

Η χρήση ενός οικιακού δρομολογητή εμποδίζει την εφαρμογή από το να εγκαθιδρύσει συνδέσεις με συστήματα έξω από το οικιακό δίκτυο αφού δεν μπορεί να προσδιορίσει το τοπικό σύστημα στόχο στο οποίο απευθύνονται τα εισερχόμενα αιτήματα. Το πρόβλημα εμφανίζεται μόνο στις συνδέσεις μεταξύ χρηστών και μπορεί να λυθεί με τη χρήση **στατικής θύρας** επικοινωνίας και **Port Forwarding** ρυθμίσεων στον δρομολογητή.

Αφού λυθεί το πρόβλημα προώθησης των αιτημάτων στον σωστό τοπικό παραλήπτη θα πρέπει να μελετηθεί η συμπεριφορά των συστημάτων όταν χρησιμοποιούν φορητές συνδέσεις όπως το **GSM** δίκτυο. Σε μια συσκευή με φορητή σύνδεση μπορεί να αλλάξει η διεύθυνση σε απρόσμενη χρονική στιγμή και να χαθούν οι ενεργές συνδέσεις με τους συμμετέχοντες στο δίκτυο.

2. Ασυμβατότητα του JXTA/JXSE με το Android

Το πρωτόκολλο αναπτύχθηκε και μεταφέρθηκε στο Android. Η χρήση του επιτρέπει τη σύνδεση μεταξύ χρηστών μέσω τεχνικών όπως το Invitation και το Broadcasting. Το Broadcasting απαιτεί την ύπαρξη ενός δημοσίου καναλιού το οποίο θα φέρει τα μηνύματα.

Το Broadcasting κομμάτι του πρωτοκόλλου δεν είναι απαραίτητο για την λειτουργία της εφαρμογής, αφού η εφαρμογή θα μπορούσε να λειτουργήσει μέσω του Invitation

συστήματος. Παρόλα αυτά η υλοποίηση του πρωτοκόλλου απαιτεί τη σύνδεση στο δημόσιο κανάλι ασχέτως αν χρησιμοποιείται ή όχι το Broadcasting σύστημα.

Όταν μια εφαρμογή που χρησιμοποιεί το πρωτόκολλο δεν βρει δημόσιο κανάλι να συνδεθεί θα πετάξει μια **NullPointException.** Αυτό δεν αποτελεί πρόβλημα στους υπολογιστές αφού κατά την εκτέλεση η εφαρμογή γίνεται **Optimized** και η **Exception** μετατρέπετε σε **Warning** και η λειτουργία της συνεχίζει κανονικά. Στο Android όμως αυτή η διαδικασία δεν ακολουθείται αφού η εφαρμογή πρέπει να μετατραπεί από Java σε Android, με αποτέλεσμα να δημιουργείται σφάλμα λόγω του **Exception** και να είναι αδύνατη η σύνδεση μεταξύ χρηστών.

Το πρόβλημα με το δημόσιο κανάλι θα μπορούσε να λυθεί τροποποιώντας το πακέτο του πρωτοκόλλου. Όμως, οποιαδήποτε τροποποίηση στο πακέτο μπορεί να δημιουργήσει άλλα σφάλματα τα οποία μπορούν να οδηγήσουν στην τροποποίηση του μεγαλύτερου μέρους του πακέτου.

Όποια τροποποίηση και να γίνει στο JXTA/JXSE πακέτο, το δημόσιο κανάλι δε θα λειτουργήσει και χωρίς αυτό απαραίτητες λειτουργίες, όπως η αναζήτηση χρηστών και η ανταλλαγή κρυπτογραφικών κλειδιών, δε θα μπορέσουν να υλοποιηθούν. Οπότε η καλύτερη λύση είναι η δημιουργία ενός νέου πακέτου το οποίο θα αναλάβει τον ρόλο του δημοσίου καναλιού (Server).

3. JavaFx – Gluon

Το Android χρησιμοποιεί ένα εξειδικευμένο σύστημα σχεδίασης το οποίο αποτελείται από τη δική του έκδοση της Java και ένα σύστημα σχεδίασης γραφικών. Από τη στιγμή που ξεκίνησαν να δημοσιεύονται εφαρμογές για τον Android, ήταν απαραίτητο οι εφαρμογές να σχεδιάζονται με την έκδοση της Java που συμπεριλαμβάνει το σύστημα. Δεν ισχύει το ίδιο για τα γραφικά περιβάλλοντα, τα οποία αρχικά μπορούσαν να σχεδιαστούν μέσω του JavaFx, αλλά στην πορεία το JavaFx εγκαταλείφθηκε για το πιο εξιδεικευμένο σύστημα το οποίο χρησιμοποιείται σήμερα από το Android.

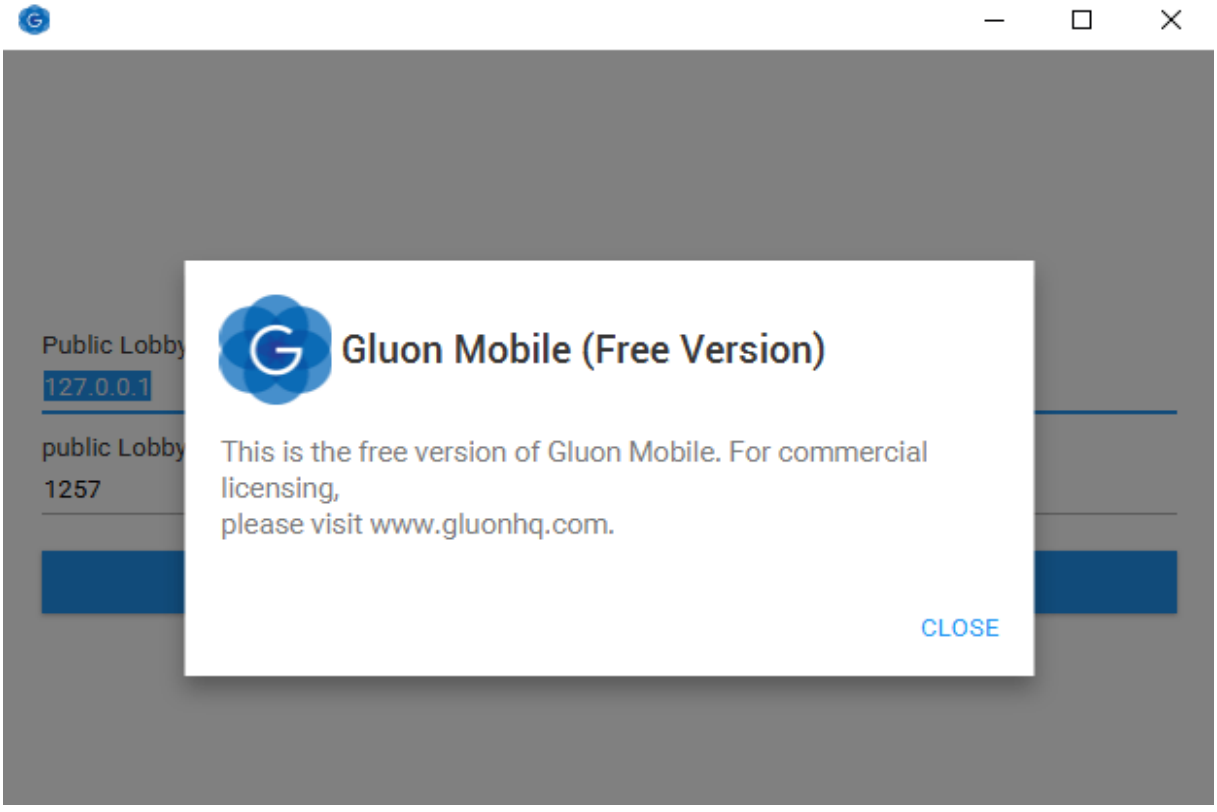
Η ασυμβατότητα του Android με το JavaFx και τη Java για υπολογιστές μπορεί να ξεπεραστεί χρησιμοποιώντας εργαλεία, όπως το Gradle, τα οποία μπορούν να μετατρέψουν τις εφαρμογές υπολογιστών σε Android. Το Gradle και τα άλλα εργαλεία όπως αυτό

χρησιμοποιούν scripts και προγράμματα τα οποία συμπληρώνουν βιβλιοθήκες και αντιστοιχούν εντολές στην εφαρμογή με βάση την πλατφόρμα/γλώσσα στόχο.

Οι συνεχόμενα αυξανόμενοι ρυθμοί ανάπτυξης του Android καθιστούν αυτές τις τεχνικές περιττές, λόγο της περιπλοκότητας τους και την ανάγκη τους για διαρκή τροποποίηση ώστε να συμβαδίζουν με το Android και τις απαιτήσεις του. Για αυτό τον λόγο αναπτυχθήκαν προγράμματα όπως το Gluon, το οποίο είναι ένα **framework** από βιβλιοθήκες, μεθόδους και αυτοματισμούς.

Το Gluon πέρα από μεθόδους συμβατότητας προσφέρει isolation. Αυτό σημαίνει πως αν κάποια μέθοδος της εφαρμογής δεν λειτουργεί σε όλες τις πλατφόρμες, μπορεί να δημιουργηθεί ξεχωριστή μέθοδος για κάθε πλατφόρμα στην οποία παρουσιάζεται κάποιο πρόβλημα.

Το μειονέκτημα του Gluon είναι ότι, στην ουσία είναι ένα επί πληρωμή σύστημα, το οποίο προσφέρει μια δωρεάν δοκιμαστική έκδοση. Η δωρεάν έκδοση δεν έχει περιορισμούς και επιτρέπει την ανάπτυξη των εφαρμογών χωρίς άδεια, αλλά χρησιμοποιεί μια **Nag Screen** η οποία ζητά την αγορά και εγκατάσταση μιας πληρωμένης άδειας. Το κόστος της άδειας είναι 461€ (τη μέρα που γράφεται αυτό το κείμενο) για ένα χρόνο, αλλά δεν είναι απαραίτητο για την ολοκλήρωση της εργασίας.



Εικόνα 1: Gluon Nag Screen

4. Character encoding

Παρόλο που το framework προσφέρει πολλά βοηθήματα, βασίζεται σε παλιά έκδοση της Java. Η έκδοση αυτή δεν υποστηρίζει **UTF-8** χαρακτήρες μέσω του Java Socket. Για την αποστολή μέσω του Socket χρησιμοποιείται το **PrintWriter** αντικείμενο το οποίο χρησιμοποιεί το Default encoding του περιβάλλοντος εργασίας.

Η έκδοση 9 της Java ορίζει το **UTF-8 Character Set** ως το προεπιλεγμένο **Character Encoding**. Ο ορισμός του Character Set από την Java μπορεί να λύνει κάποια από τα προβλήματα κωδικοποίησης αλλά δεν εξασφαλίζει την κάλυψη όλων τον περιπτώσεων.

Ο μόνος τρόπος να καλυφθούν όλες οι περιπτώσεις είναι να οριστεί το Character Set κατά τη δημιουργία του PrintWriter. Αυτή η δυνατότητα πρωτοεμφανίστηκε στην έκδοση 10 της Java.

Η επικοινωνία μέσα από έναν Java Socket Server

```
public class Main {

    public static void main(String[] args) throws IOException {
        String msg = "This is a test – Αυτή είναι μια δοκιμή";
        ServerSocket serverSocket = new ServerSocket(9999);

        Thread host = new Thread(new Runnable() {

            @Override
            public void run() {
                while (true) {
                    try {
                        Socket socket = serverSocket.accept();

                        if (socket != null) {
                            BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(socket.getInputStream(), StandardCharsets.UTF_8));

                            while (true) {
                                String line = bufferedReader.readLine();

                                if (line != null) {
                                    System.out.println(line);
                                }
                            }
                        }
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
            }
        });

        host.start();

        Socket socket = new Socket("127.0.0.1", 9999);

        PrintWriter printWriter = new PrintWriter(socket.getOutputStream(), true);
        printWriter.println(msg);
    }
}
```

Στο παραπάνω πρόγραμμα βλέπουμε μια Java Class, η οποία δημιουργεί ένα Socket Server/Client ζευγάρι και επικοινωνεί ένα δοκιμαστικό μήνυμα σε δύο γλώσσες. Σύμφωνα με το πρόγραμμα θα δημιουργηθεί ένας Server ο οποίος παρακολουθεί τη **localhost** διεύθυνση

“127.0.0.1” και τη θύρα 9999. Αυτές οι ρυθμίσεις μπορούν να διαφέρουν ανάλογα με τις Hosting επιλογές και ρυθμίσεις.

Αφού εκκινήσει ο Server δημιουργείται ένα Thread το οποίο τον παρακολουθεί για εισερχόμενες συνδέσεις και μηνύματα. Αυτή η διαδικασία γίνεται μέσω του Thread ώστε ο Server να λειτουργεί αδιάκοπα. Ο Server διακόπτεται και σταματά να δέχεται νέες συνδέσεις μόλις συνδεθεί ένας χρήστης, διότι παρακολουθεί τη νέα σύνδεση για εισερχόμενα μηνύματα.

Με τον Server και το Thread σε λειτουργία ο χρήστης μπορεί να συνδεθεί και να στείλει μηνύματα σ' αυτόν. Η αποστολή γίνεται μέσω του Stream writer **PrintWriter,** ο οποίος γράφει τα μηνύματα του χρήστη στο **OutputStream** του Socket.

Σύμφωνα με το πρόγραμμα ό,τι μήνυμα στείλει ο χρήστης θα εκτυπωθεί στην κονσόλα. Οπότε αν η πλατφόρμα και η εφαρμογή έχουν ρυθμιστεί στο σωστό Encoding στη κονσόλα θα πρέπει να πάρουμε το δοκιμαστικό κείμενο “This is a test – Αυτή είναι μια δοκιμή”. Αν η Κωδικοποίηση Χαρακτήρων δεν είναι σωστή το κείμενο στην κονσόλα θα γράφει “This is a test –???? ????? ??? ???”.

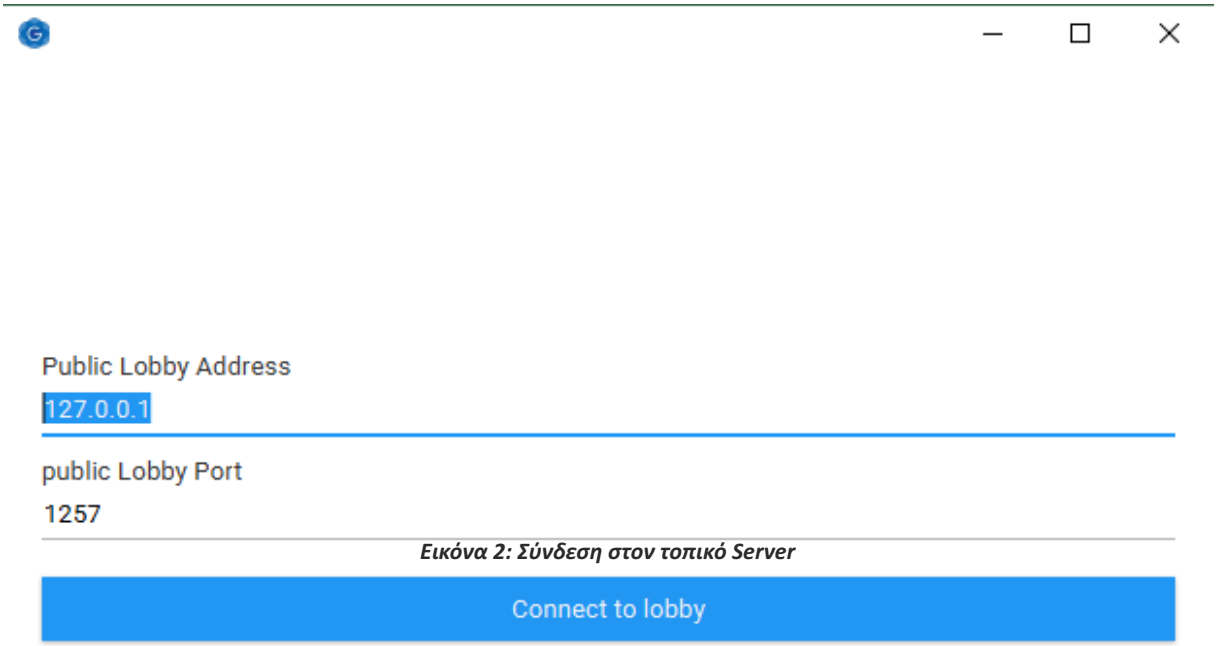
Στο παρακάτω πλαίσιο φαίνεται ο σωστός τρόπος για την κάλυψη όλων των περιπτώσεων όπως ορίζεται από την έκδοση 10 της Java. Χρησιμοποιώντας αυτή τη μέθοδο η ορθή μεταφορά μηνυμάτων είναι εγγυημένη.

```
PrintWriter printWriter = new PrintWriter(socket.getOutputStream(), true, StandardCharsets.UTF_8);
printWriter.println(msg);
```

5. Σύνδεση στον Public Server

Η σύνδεση στον δημόσιο Server απαιτεί την ύπαρξη ενός χώρου φιλοξενίας για την εφαρμογή και ένα Domain ή μια στατική IP. Κατά τη δημιουργία και την έκδοση της εφαρμογής θα πρέπει να οριστεί στις ρυθμίσεις η διεύθυνση του Server. Επειδή όμως ο Server εκτελείται από έναν τοπικό υπολογιστή η διεύθυνσή του δεν είναι σταθερή και πρέπει να βρεθεί ένας έμμεσος τρόπος σύνδεσης στον Server.

Ο ευκολότερος τρόπος αντιμετώπισης αυτού του προβλήματος είναι ο ορισμός της διεύθυνσης κατά την εκκίνηση της εφαρμογής μέσω μιας φόρμας. Στη φόρμα ο χρήστης θα πρέπει να συμπληρώσει την IP και τη θύρα επικοινωνίας του τοπικού δημόσιου Server. Τα στοιχεία σύνδεσης θα αποθηκευτούν στη συσκευή και κατά την επόμενη σύνδεση ο χρήστης θα μπορέσει να συνδεθεί χωρίς να χρειάζεται να τα συμπληρώσει.



Εικόνα 2: Σύνδεση στον τοπικό Server

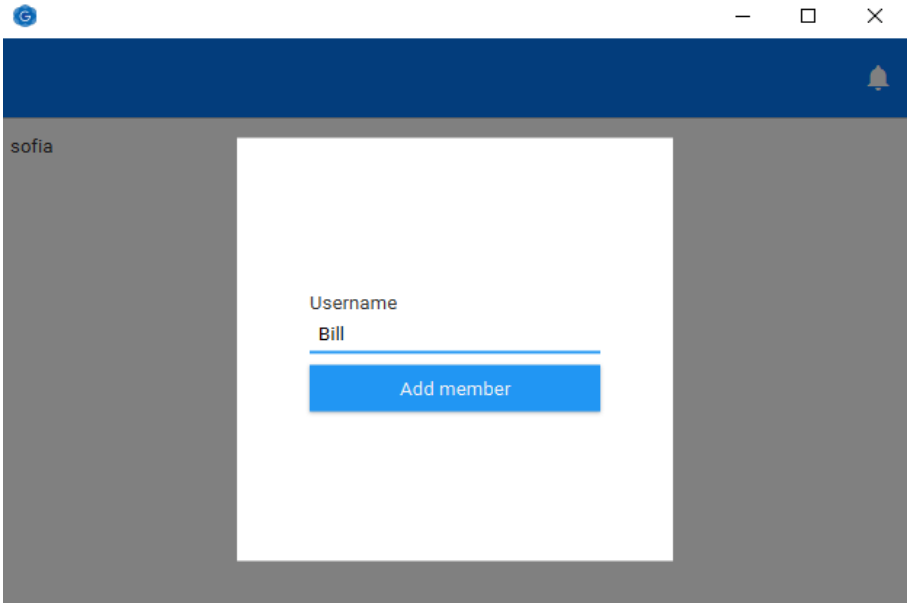
Κεφάλαιο 4. Τρόπος λειτουργίας

Η εφαρμογή αποτελείται από δύο στοιχεία, τον Server και τον Client. Μπορούν να υπάρχουν πολλαπλοί Servers και Clients. Οι Servers ελέγχουν αν οι χρήστες έχουν το δικαίωμα να συνδεθούν στο δίκτυο τους και τους επιτρέπει να αναζητήσουν και να συνδεθούν με άλλους χρήστες. Οι Clients αναλαμβάνουν την σύνδεση στον Server και στους άλλους χρήστες/Clients και την κωδικοποίηση/αποκωδικοποίηση των μηνυμάτων.

Για την ορθή λειτουργία του δικτύου οι Clients πάντα συνδέονται σ έναν κοινό Server, ο οποίος έχει το ρόλο του δημόσιου δρομολογητή. Έτσι όταν η εφαρμογή ενεργοποιείται θα πρέπει να συνδεθεί στον δημόσιο Server. Μ αυτόν τον τρόπο ο χρήστης ταυτοποιείται και μπορεί να βρει άλλους χρήστες με τους οποίους μπορεί να συνδεθεί και να συνομιλήσουν.

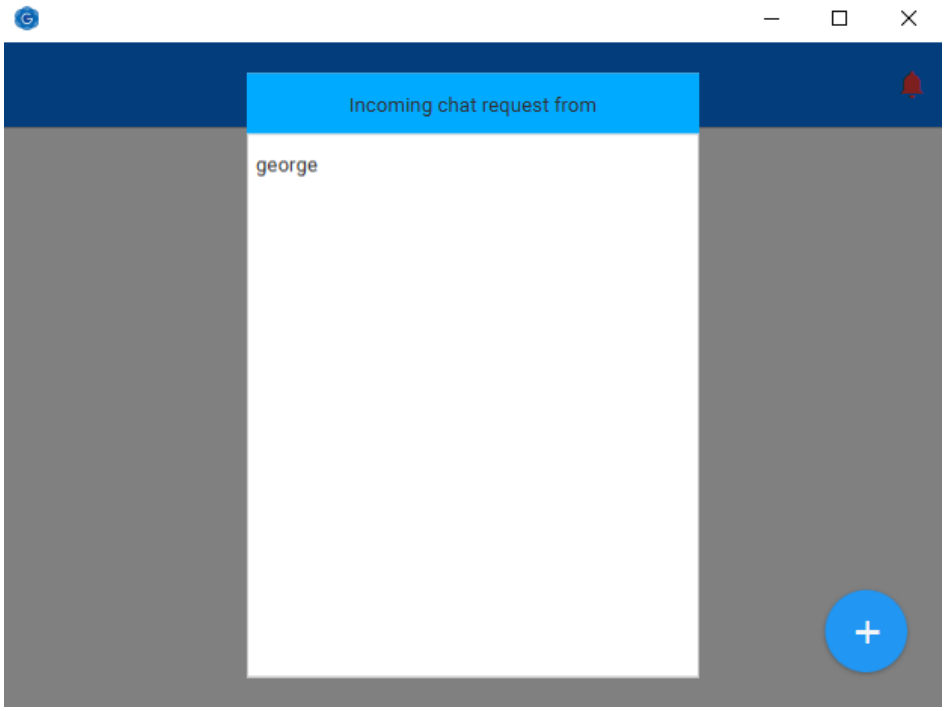
Αφού η εφαρμογή συνδεθεί με τον Server ένα Socket δημιουργείται μεταξύ της συσκευής Client και του Server. Το επόμενο βήμα είναι η ταυτοποίηση του χρήστη, η οποία γίνεται στέλνοντας στον Server τα στοιχεία του χρήστη (Client και Password). Μετά την ταυτοποίηση ο χρήστης μπορεί να αναζητήσει άλλους χρήστες και να συνδεθεί μαζί τους.

Υποθέτοντας ότι υπάρχουν οι χρήστες **A** και **B**. Οι δύο χρήστες είναι συνδεδεμένοι και ταυτοποιημένοι στον δημόσιο Server. Αν ο χρήστης A θέλει να επικοινωνήσει με τον χρήστη B θα πρέπει να τον αναζητήσει στον Server, ο οποίος θα ελέγξει αν υπάρχει ο χρήστης στη βάση του και θα επιστρέψει την αντίστοιχη απάντηση.

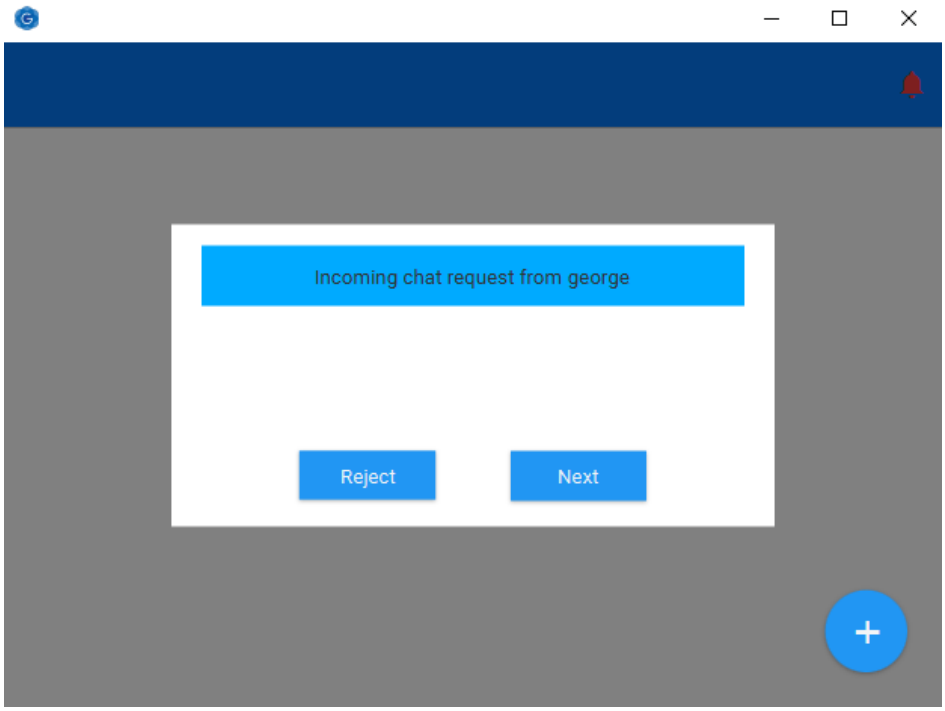


Εικόνα 3: Αναζήτηση ενός χρήστη στον Server

Στην περίπτωση που ο χρήστης Β βρεθεί θα του στείλει ένα μήνυμα το οποίο θα τον ενημερώνει ότι ο χρήστης Α θέλει να επικοινωνήσει μαζί του. Τότε ο χρήστης Β θα απαντήσει στον Server και εκείνος με τη σειρά του θα μεταφέρει την απάντηση στον χρήστη Α.

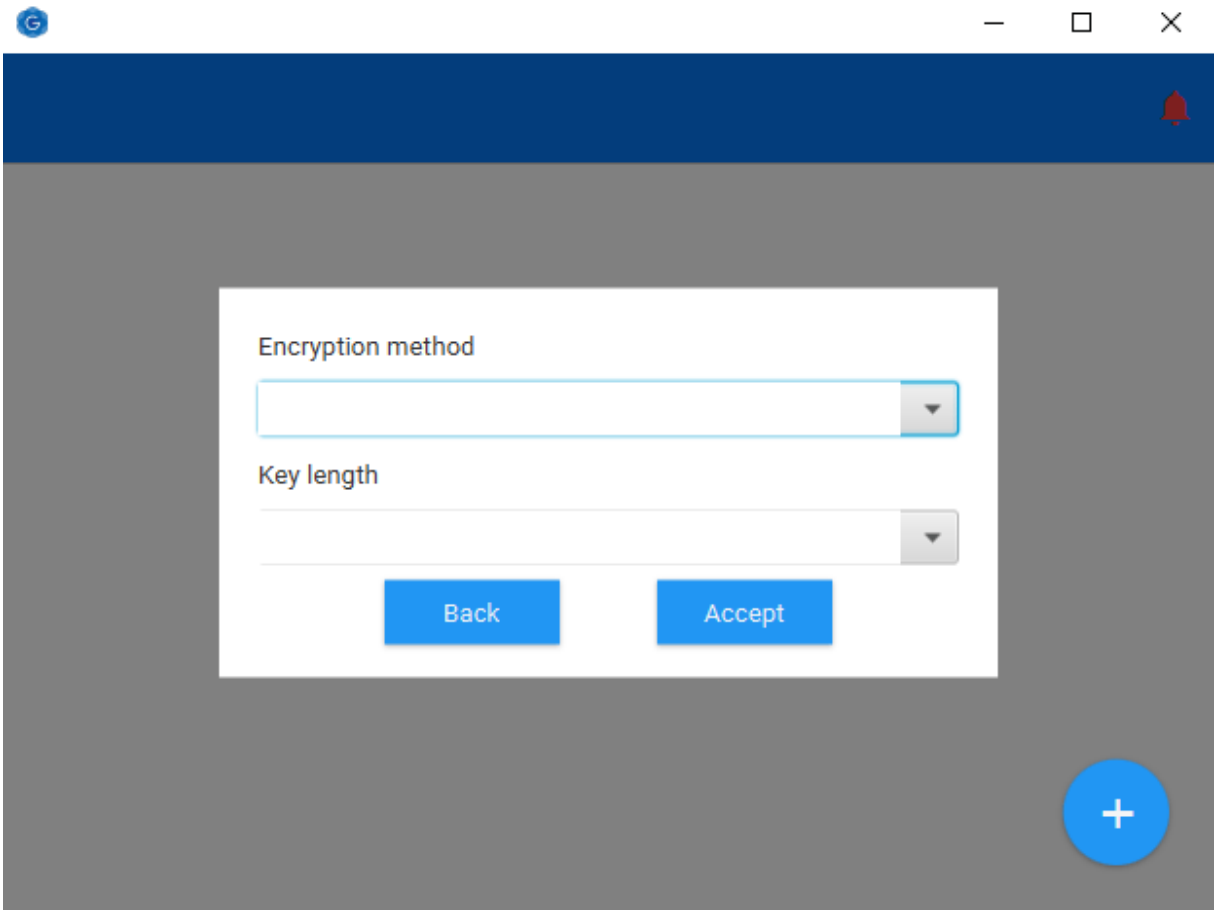


Εικόνα 4: Η λίστα προσκλήσεων



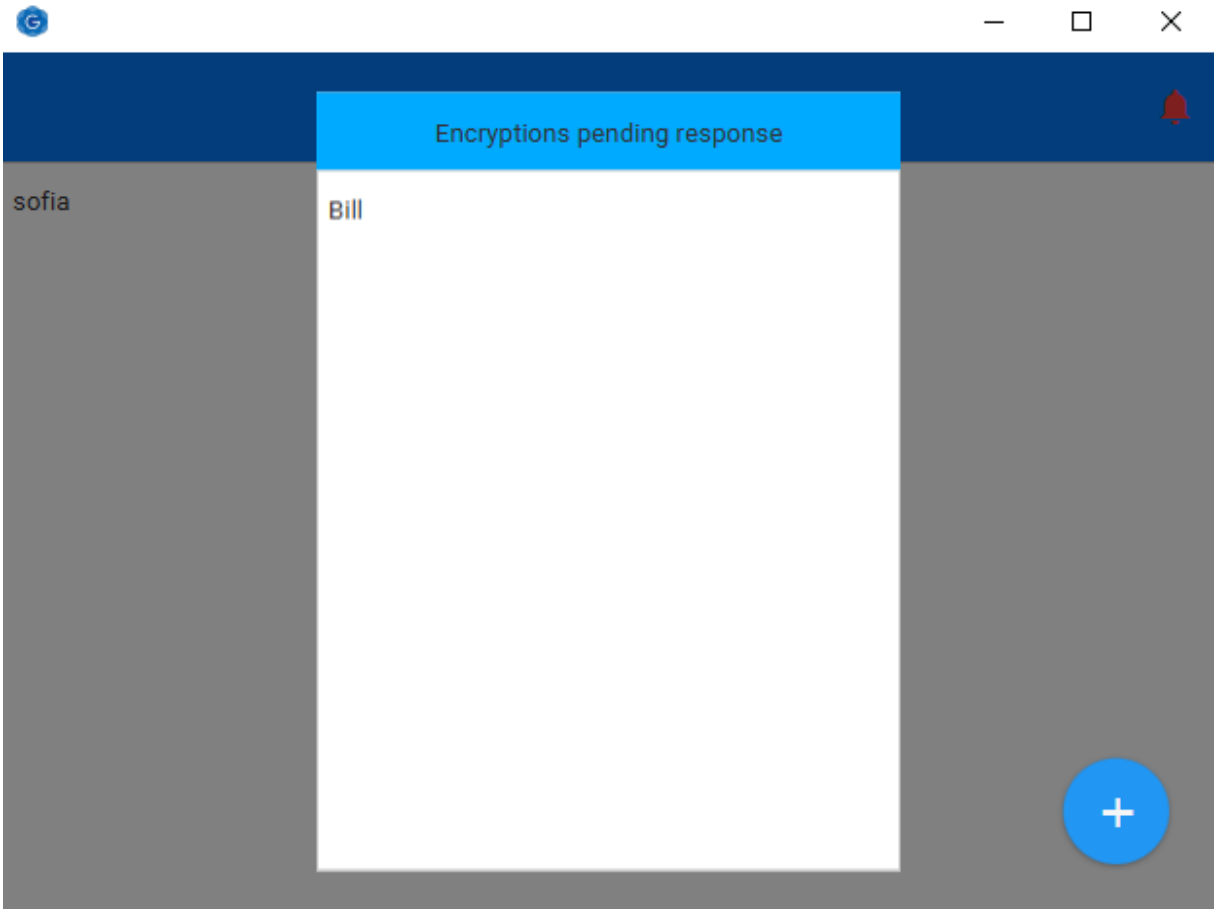
Εικόνα 5: Χειρισμός της εισερχόμενης πρόσκλησης

Αν ο χρήστης επιθυμεί να επικοινωνήσει με τον χρήστη Α, θα πρέπει να στείλει στον Server το είδος της κρυπτογράφησης, το μέγεθος του κλειδιού και το δημόσιο κλειδί που θα χρησιμοποιηθεί στην επικοινωνία. Ο Server θα προωθήσει τα στοιχεία στον χρήστη Α το οποίο θεωρείται θετική απάντηση.

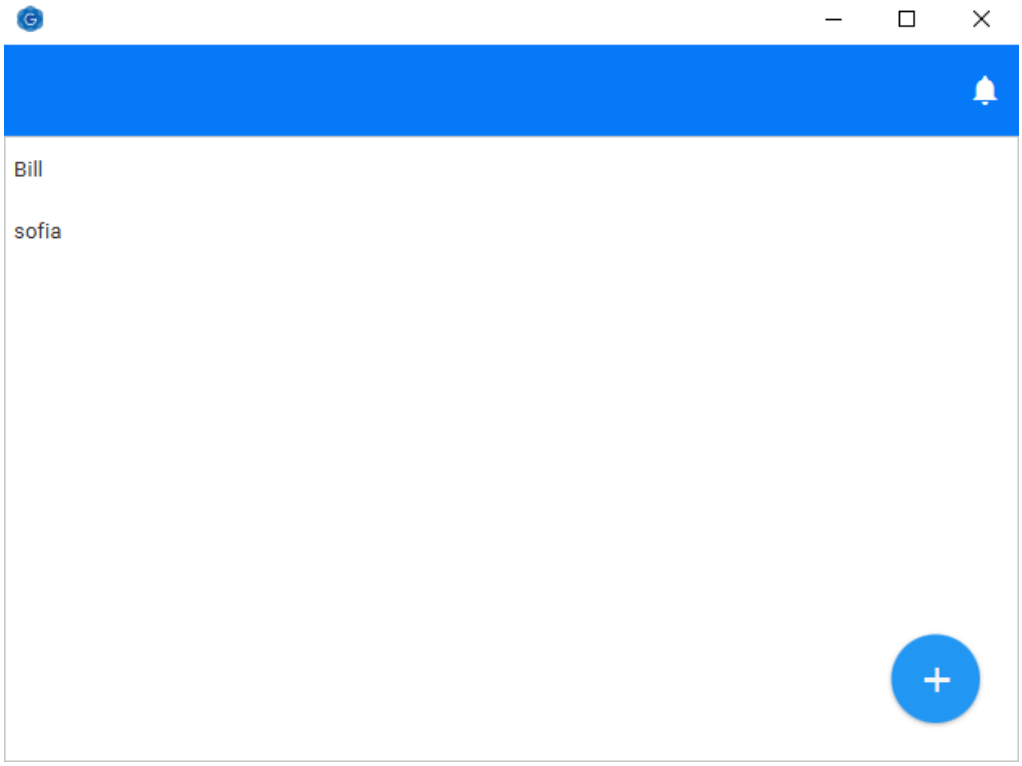


Εικόνα 6: Ρύθμιση της εξερχόμενης κρυπτογράφισης

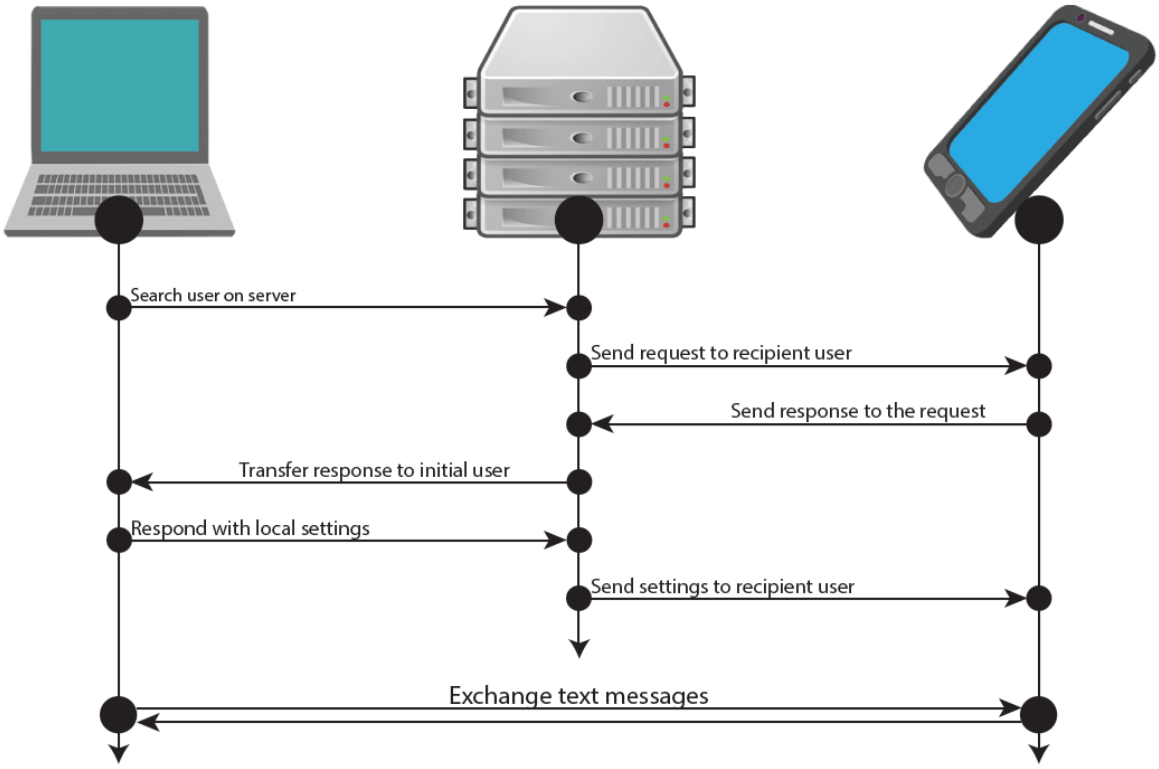
Αφού έχει λάβει θετική απάντηση ο χρήστης Α θα πρέπει να δημιουργήσει και να στείλει το δικό του κλειδί μέσω του Server στον χρήστη Β και να ολοκληρώσει την εγκατάσταση της επικοινωνίας.



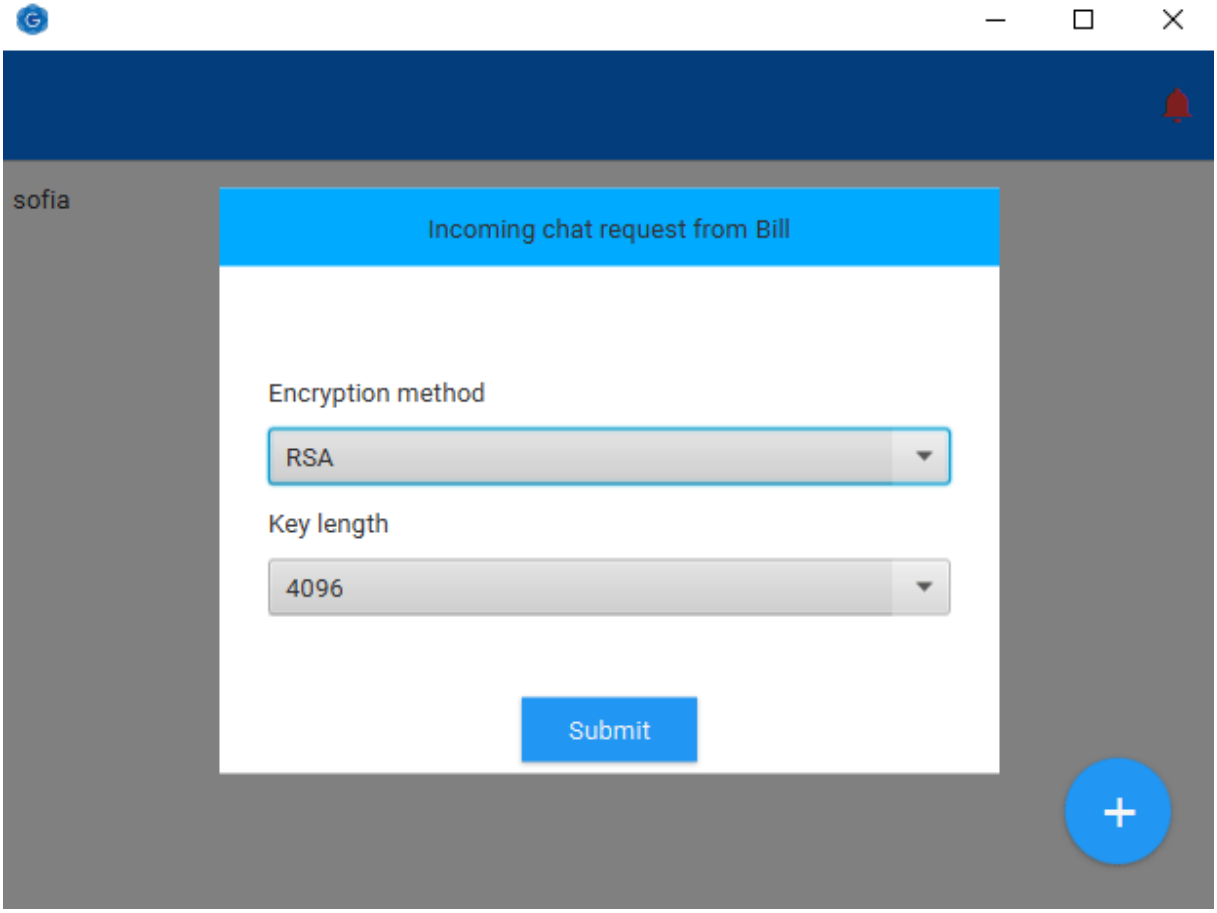
Εικόνα 7: Η λίστα εισερχόμενων κρυπτογραφικών ρυθμίσεων από θετικές απαντήσεις



Εικόνα 9: Η λίστα συνομιλητών



Εικόνα 10: Το διάγραμμα εγκαθίδρυσης επικοινωνίας



Εικόνα 8 Ρύθμιση της εξερχόμενης κρυπτογράφισης για θετική απάντηση

Κατά τη διαδικασία της εγκατάστασης οι χρήστες μπορούν να δημιουργήσουν όποιον συνδυασμό κλειδιών θέλουν, επιτρέποντας έτσι την επικοινωνία μεταξύ ένα χρήστη που χρησιμοποιεί ένα κλειδί τύπου **RSA** και ενός που χρησιμοποιεί **TripleDES**.

Κεφάλαιο 5. Συμπεράσματα

Στόχος της εργασίας ήταν η σχεδίαση μιας εφαρμογής που ανταλλάσσει κρυπτογραφημένα μηνύματα μεταξύ συστημάτων με επίκεντρο τους προσωπικούς υπολογιστές και τα Android κινητά. Την διαδικασία επίσπευσε η χρήση του Gluon, το οποίο βοηθά στη σχεδίαση του γραφικού περιβάλλοντος της εφαρμογής ώστε να μπορεί να λειτουργεί σε μια πληθώρα λειτουργικών συστημάτων (συμπεριλαμβανομένου και του IOS το οποίο δεν αποτελεί κομμάτι της εργασίας).

Ένα ακόμα σημαντικό χαρακτηριστικό του Gluon είναι η ικανότητα του να μεταγλωττίζει τον **Source Code** σε **Native Code** του συστήματος στόχου (Android ή IOS). Για να γίνει η μεταγλώττιση από το ένα σύστημα στο άλλο θα πρέπει να καθοριστεί η πηγαία γλώσσα και το API στόχος. Η συσχέτιση των δύο θα πρέπει να είναι ξεκάθαρη και κάθε εντολή ή δομή να αντιστοιχηθεί μεταξύ πηγής και στόχου.

Η σχεδίαση μεταγλωττιστών είναι μια περίπλοκη διαδικασία που από τη φύση της μπορεί να προκαλέσει την καθυστέρηση αφομοίωσης νεότερων στοιχείων και χαρακτηριστικών. Συνήθως αυτό δεν αποτελεί πρόβλημα διότι αυτή η μέθοδος ανάπτυξης απευθύνεται σε εφαρμογές με απλούστερη λειτουργία. Στη συγκεκριμένη εργασία όμως οι καθυστερήσεις αυτές είναι εξαιρετικά εμφανείς, γεγονός που θέτει αρκετούς τεχνικούς περιορισμούς στην ανάπτυξη της εφαρμογής.

Παρά τους περιορισμούς που επιφέρει αυτή η μέθοδος σχεδίασης προσφέρει πολλά πλεονεκτήματα όπως η παράλληλη ανάπτυξη σε πολλαπλές πλατφόρμες με κοινό κώδικα και προτυποποιημένες δομές σχεδίασης από τη βιβλιοθήκη γραφικών του Gluon.

Διεκπεραιώνοντας αυτού του είδους την εργασία πέρα από τους περιορισμούς που εκπίπτουν από τις επιλεγμένες τεχνολογίες θα πρέπει να ληφθούν υπόψη και οι επιμέρους

τεχνικοί περιορισμοί που προέρχονται από την εκάστοτε υλικοτεχνική υποδομή.

Σε προηγούμενο κεφάλαιο αναφέρθηκε το πρόβλημα εγκαθίδρυσης εισερχόμενης P2P σύνδεσης λόγο του εξοπλισμού. Το πρόβλημα αυτό είναι ένα χαρακτηριστικό παράδειγμα του ρόλου που παίζει η υλικοτεχνική υποδομή στην ανάπτυξη εφαρμογών.

Κατά τη ανάπτυξη της εφαρμογής θα πρέπει να ληφθούν υπόψη αυτοί οι τεχνικοί περιορισμοί είτε δημιουργώντας τις απαραίτητες δομές για την παράκαμψη του προβλήματος όπως ο Proxy Server είτε μεθόδους να παραμετροποιήσουν την εφαρμογή και το δίκτυο τους.

Παράλληλα θα πρέπει να οριστεί ποιος είναι ο χρήστης στόχος. Αν στόχος της εφαρμογής είναι κάποιος αρχάριος χρήστης, η εφαρμογή θα πρέπει να χρησιμοποιεί ένα απλοποιημένο Γραφικό Περιβάλλον και περιορισμένες δυνατότητες παραμετροποίησης της εφαρμογής ώστε να αποτρέψει τον χρήστη από το να πάρει λάθος αποφάσεις και να επηρεάσουν αρνητικά το σύστημά του και την εμπειρία του, χρησιμοποιώντας την εφαρμογή.

Μελετώντας το θέμα χρηστών, σχεδιάζοντας την εφαρμογή πέρα από τους αρχάριους χρήστες θα πρέπει να ληφθούν υπόψη και οι προχωρημένοι χρήστες ή μεγάλες εταιρίες και όμιλοι. Αυτού του είδους η ανάλυση μπορεί να οδηγήσει στη δημιουργία πρωτοκόλλων για την ανάπτυξη νέων παρένθετων υπηρεσιών, κατ᾽ επέκταση επεκτείνοντας το κοινό της εφαρμογής.

Κεφάλαιο 6. Μελλοντικά σχέδια

Η εργασία έχει επιτύχει στην εγκαθίδρυση επικοινωνίας μέσω ενσύρματων και ασύρματων τοπικών μέσων χρησιμοποιώντας έναν τοπικό server για την δρομολόγηση των αιτημάτων. Στα πλαίσια της επικοινωνίας μέσω του Διαδικτύου είναι απαραίτητο να γίνουν μία σειρά από τροποποιήσεις.

1. Εγκαθίδρυση συνδέσεων σε συστήματα πίσω από οικιακό δρομολογητή (router)

Για να επιτευχθεί η P2P σύνδεση δύο συστημάτων μέσω του οικιακού εξοπλισμού σύνδεσης στο διαδίκτυο θα πρέπει να παραμετροποιηθεί ο οικιακός δρομολογητής ώστε να μπορεί να δρομολογεί την εισερχόμενη κίνηση στο σύστημα στόχο.

Η διαδικασία παραμετροποίησης του δρομολογητή λέγεται Port Forwarding και απαιτεί από τον χρήστη να ορίσει στον τοπικό δρομολογητή μια δημόσια Θύρα στην οποία θα δέχεται την εισερχόμενη πληροφορία και μια ιδιωτική Θύρα σε μία τοπική διεύθυνση στην οποία θα την προωθεί.

Επειδή δεν είναι δυνατόν να πραγματοποιηθεί Port Forwarding για κάθε σύνδεση που πραγματοποιεί ο χρήστης θα πρέπει να δημιουργηθεί μια εναλλακτική μέθοδος για τη σύνδεση των συστημάτων, όπως η χρήση Proxy Server.

Ως Proxy Server ορίζεται ένα υπολογιστικό σύστημα το οποίο λαμβάνει αιτήματα από τα διάφορα συνδεδεμένα μ΄ αυτό συστήματα και τα προωθεί στον τελικό παραλήπτη. Ο proxy server μπορεί να είναι τοπικός ή απομακρυσμένος. Στην περίπτωση που ο proxy είναι τοπικός, απαιτείται να γίνουν ρυθμίσεις Port Forwarding, ώστε να μπορεί να προσπελάσει τον οικιακό εξοπλισμό.

Ανεξάρτητα με την γεωγραφική τοποθεσία του, ο Server μπορεί να έχει έναν χαρακτηριστικό ρόλο. Οι διαθέσιμοι ρόλοι για έναν Proxy server είναι ο Tunneler και ο Supervisor. Στόχος και των δύο ρόλων είναι η προώθηση των μηνυμάτων παρακάμπτοντας τους περιορισμούς του οικιακού εξοπλισμού.

Από τους δύο ρόλους ο Tunneler είναι ο πιο απλός. Επιτρέπει την επικοινωνία προωθώντας την εισερχόμενη κίνηση στη σύνδεση στόχο. Ο Supervisor μπορεί να προσφέρει περισσότερες λειτουργίες όπως η επιβολή κανόνων και η παραμετροποίηση της σύνδεσης.

2. Διατήρηση των συνδέσεων στα κινητά μέσω του Διαδικτύου

Ανάλογα με τον τύπο δικτύου ή την σύνδεση, η διεύθυνση/IP του συστήματος μπορεί να αλλάξει χωρίς προειδοποίηση. Αυτό έχει σαν αποτέλεσμα την μείωση της σταθερότητας και αξιοπιστίας της υπηρεσίας. Ένα πολύ πιθανό παράδειγμα αυτού, είναι ένα κινητό τηλέφωνο μέσα σ ένα όχημα να ταξιδεύει μέσα στην πόλη, το οποίο μπορεί να αλλάξει πολλαπλές κεραίες καθώς περνάει από κυψέλη σε κυψέλη.

Η αλλαγή της διεύθυνσης θα οδηγήσει στη διακοπή των συνδέσεων και την κατάρρευση του εικονικού δικτύου. Το σύστημα θα πρέπει να ελέγχει την ίδια του τη διεύθυνση και όταν εντοπίσει κάποια αλλαγή θα πρέπει να ειδοποιήσει το πρωτεύων δίκτυο, το οποίο θα αναλάβει να ενημερώσει τα συμμετέχοντα συστήματα ώστε να επανεκινήσουν τις συνδέσεις τους.

3. Ιδιωτικά δίκτυα πολλαπλών επιπέδων

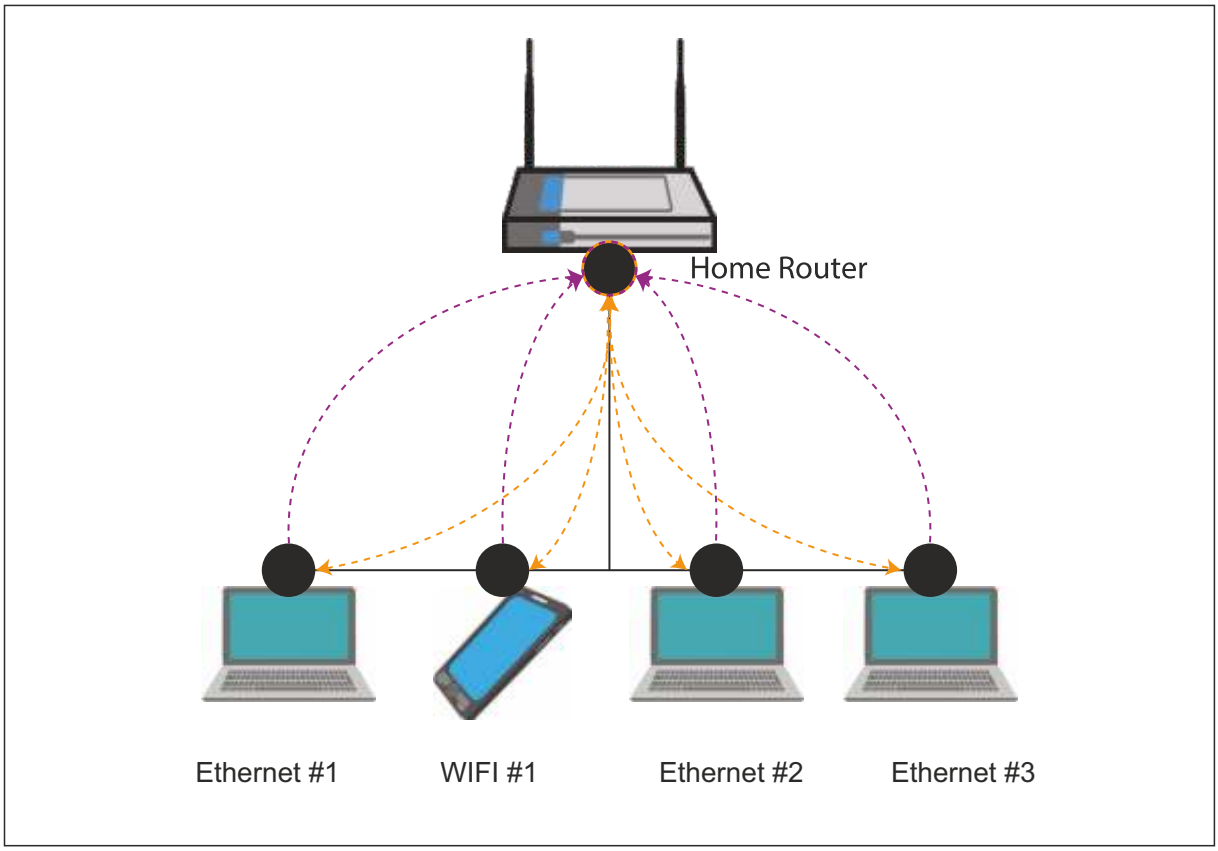
Ένα από τα σημαντικότερα σημεία μιας ασφαλούς επικοινωνίας είναι η χρήση ιδιωτικών δικτύων. Ένα ιδιωτικό δίκτυο μπορεί να υλοποιηθεί χρησιμοποιώντας είτε έναν ιδιωτικό Proxy Server είτε έναν ιδιωτικό Server της εφαρμογής.

Με τη χρήση ενός Proxy Server ένα σύστημα ή ένα ιδιωτικό δίκτυο μπορεί να αποκτήσει πρόσβαση στο δημόσιο δίκτυο της εφαρμογής. Με τη χρήση ενός Supervisor Proxy Server ο διαχειριστής του δικτύου μπορεί να έχει έλεγχο στις συνδέσεις και την κατάσταση του δικτύου. Αξίζει να σημειωθεί πως ο Proxy Server είναι ή προτεινόμενη λύση όταν το ιδιωτικό δίκτυο βρίσκεται πίσω από έναν μικρό οικιακό δρομολογητή και θέλουμε να προωθήσουμε όλη την

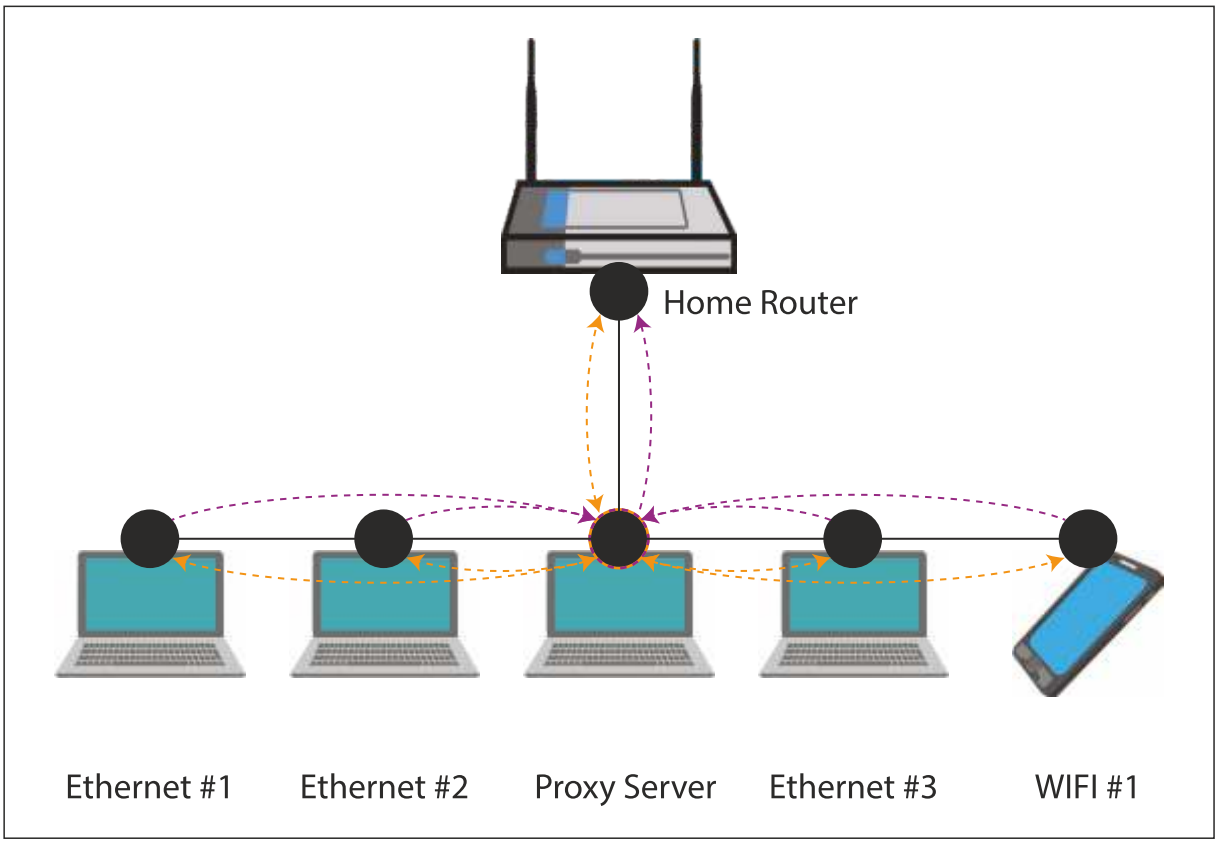
επικοινωνία μέσα από ένα Port Forward.

Μια εναλλακτική στον Proxy Server είναι η χρήση του ιδιωτικού Server. Ένας ιδιωτικός Server μπορεί να απομονώσει το δίκτυο από τη δημόσια κίνηση της υπηρεσίας, έτσι όλα τα αιτήματα δρομολόγησης θα εξυπηρετηθούν από αυτόν. Το μεγαλύτερο πλεονέκτημα της χρήσης ενός τοπικού Server είναι η δυνατότητα χρήσης ιδιωτικής υπηρεσίας ονοματοδοσίας η οποία προσφέρει τον έλεγχο στο ποιες συσκευές μπορούν να συνδεθούν στο δίκτυο.

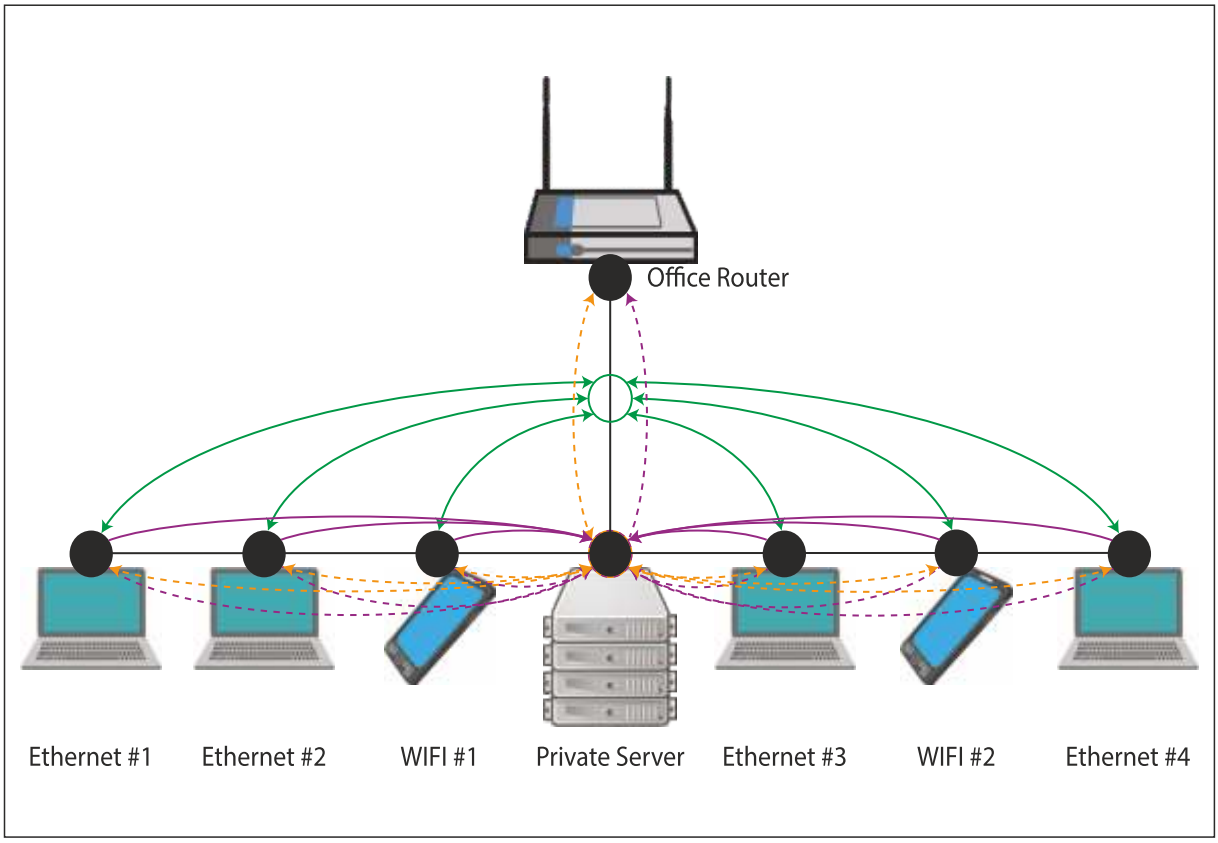
ΠΑΡΑΡΤΗΜΑ



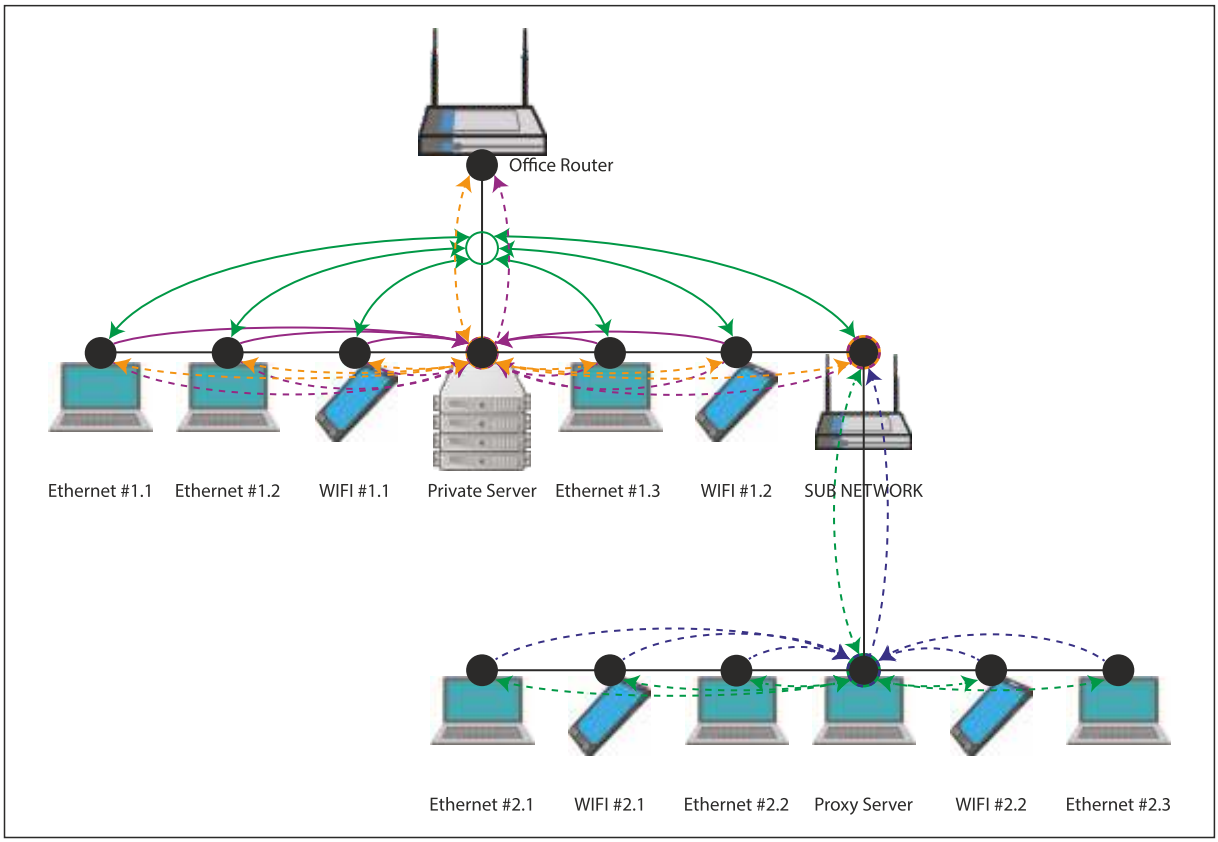
Εικόνα 1. Οικιακό δίκτυο



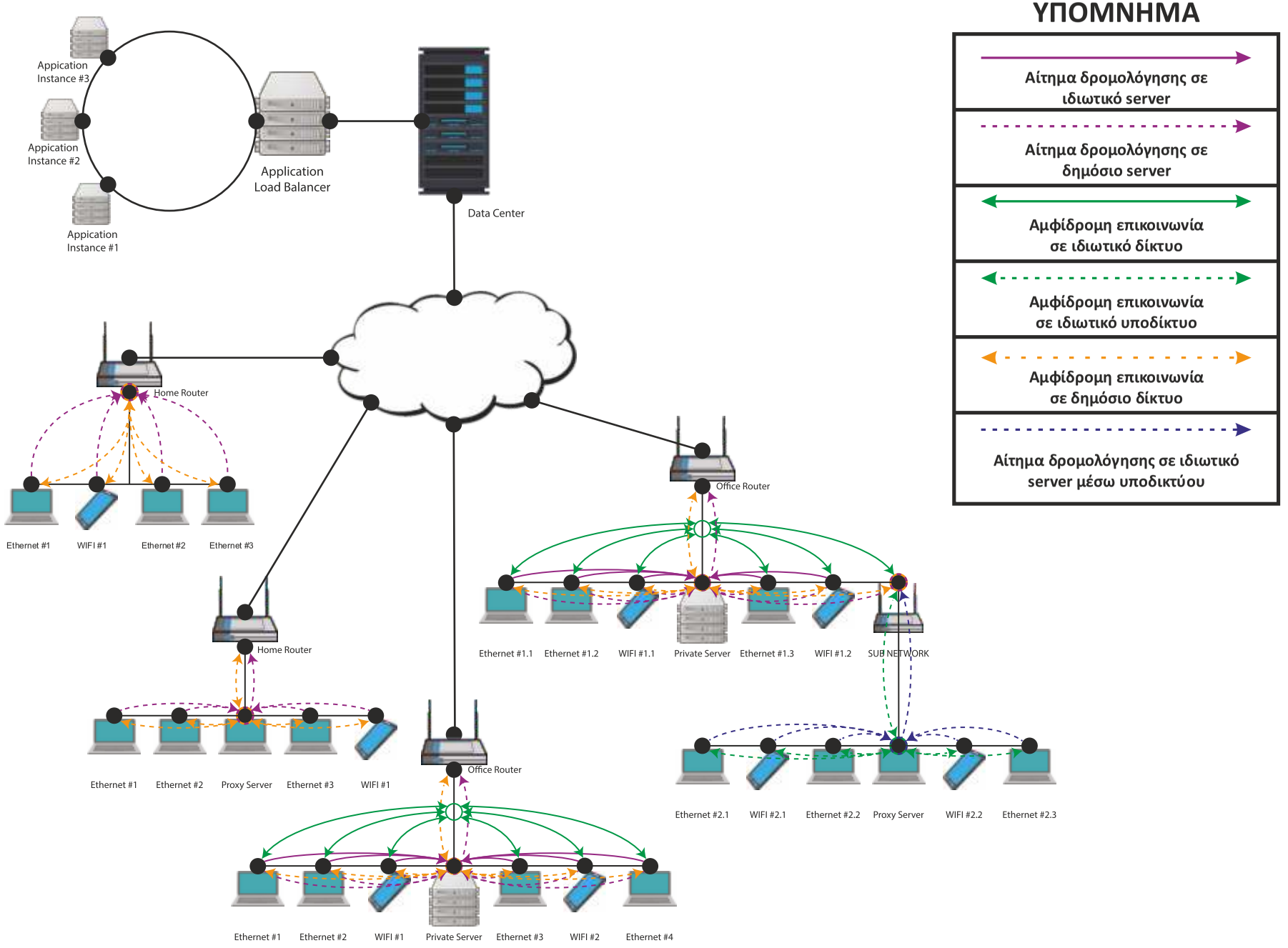
Εικόνα 2. Οικιακό δίκτυο με χρήση Proxy Server



Εικόνα 3. Επαγγελματικό δίκτυο με ιδιωτικό Server



Εικόνα 4. Επαγγελματικό δίκτυο με ιδιωτικό Server και ένα υποδίκτυο



Εικόνα 5. Το οικοσύστημα της εφαρμογής

Βιβλιογραφία:

Serious Cryptography - A Practical Introduction to Modern Encryption by Jean – Philippe Aumasson

Java Cryptography - Jonathan B. Knudsen – O`REILLY

Δικτυογραφία:

https://en.wikipedia.org/wiki/Abstract_Window_Toolkit

https://en.wikipedia.org/wiki/JavaFX

https://en.wikipedia.org/wiki/Peer-to-peer

https://en.wikipedia.org/wiki/Overlay_network

https://www.javatpoint.com/history-of-java

https://www.javatpoint.com/java-me

https://en.wikibooks.org/wiki/J2ME_Programming/The_J2ME_Platform

https://www.textrequest.com/blog/history-evolution-smartphone/

https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html

https://www.csie.ntu.edu.tw/~r93020/eBook/OReilly.Java.Swing.2nd.2002.pdf

https://www.thoughtco.com/what-is-javafx-2034192

https://gradle.org/releases/

https://gluonhq.com/

https://gluonhq.com/pricing/