

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**



Department of  
Management  
Science and  
Technology



DEPARTMENT  
OF INFORMATICS

Ειδικά Θέματα Τεχνολογίας Λογισμικού

Αναφορά Εργασίας Υλοποίησης

Έργο: TextSecure



Παπαδάς Κωνσταντίνος

Πεπολής Παναγιώτης

## Περιεχόμενα

1. Εισαγωγή .....	- 3 -
2. Κατανόηση και τεκμηρίωση του έργου .....	- 3 -
3. Πλάτος των αλλαγών .....	- 3 -
4. Ποιότητα υλοποίησης .....	- 4 -
5. Ολοκλήρωση.....	- 4 -
6. Έλεγχος.....	- 4 -
7. Συνεργασία με την ομάδα ανάπτυξης.....	- 4 -
8. Ποιότητα παραδοτέων .....	- 5 -
9. Οργάνωση στο GitHub (wiki, issues, commits, branching) .....	- 5 -
10. Επίλογος .....	- 5 -
11. Παραπομπές .....	- 5 -

## **1. Εισαγωγή**

Η ενασχόληση με το TextSecure ήταν η πρώτη μας επαφή με το λειτουργικό σύστημα android σε επίπεδο ανάπτυξης, καθώς και με το git σαν σύστημα διαχείρισης εκδόσεων. Για να ξεκινήσουμε να δουλεύουμε πάνω στο έργο χρειάστηκε πρώτα να κατανοήσουμε τις βασικές λειτουργίες του git καθώς και τα δομικά στοιχεία μιας android εφαρμογής, και τέλος τον τρόπο ανάπτυξης, μεταγλώττισης και εκτέλεσης. Επίσης ήταν η πρώτη μας επαφή με ολοκληρωμένα συστήματα μεταγλώττισης, όπως το gradle. Η επιλογή του συγκεκριμένου έργου έγινε επειδή ανήκει στο (ευρύ) πεδίο της ασφάλειας και των νέων σχετικά τεχνολογιών (Android) όπως και τα ενδιαφέροντά μας.

## **2. Κατανόηση και τεκμηρίωση του έργου**

Η τεκμηρίωση του έργου, από την πλευρά μας, κρίνεται ελλιπής. Δεν υπάρχει οργανωμένο documentation που να εξηγεί τον τρόπο λειτουργίας του έργου σε μορφή Javadoc, αλλά υπάρχει αρκετή τεκμηρίωση όσο αφορά το πρωτόκολλο ασφάλειας που έχουν σχεδιάσει και χρησιμοποιούν [1] και το μοντέλο κινδύνων (threat model) [2] που θα καλείται να αντιμετωπίσει το εν λόγω πρωτόκολλο. Στα πλαίσια της κατανόησης του πρωτοκόλλου, πριν ακόμα αποφασίσουμε ποια θα είναι η συνεισφορά μας, αντλήσαμε στοιχεία και από μια επιστημονική δημοσίευση [3] η οποία αποδεικνύει ότι η εφαρμογή είναι ευάλωτη σε επίθεση UKS (Unknown Key-Share attack), αλλά πρώτου καταλήξει σε αυτό κάνει μια εμπεριστατωμένη ανάλυση του πρωτοκόλλου ασφαλείας και των κρυπτογραφικών αλγορίθμων που χρησιμοποιούνται. Υπάρχουν επίσης οδηγίες για την μεταγλώττιση της εφαρμογής μέσα από eclipse ή AndroidStudio, τις οποίες αξιοποιήσαμε. Παρά την ελλείψη τεκμηρίωσης ο κώδικας αυτοτεκμηριώνεται μέσα από τον τρόπο συγγραφής του (ονόματα μεταβλητών και κλάσεων, οργάνωση σε πακέτα κλπ). Τελικά κατανοήσαμε ένα μεγάλο κομμάτι του έργου, κυρίως σε επίπεδο διαδικασιών και συγκεκριμένα την διαδικασία λήψης μηνύματος, αποκρυπτογράφησης, αποθήκευσης στην βάση, ανάκτησης ενός μηνύματος από την βάση και προβολής του στην οθόνη.

## **3. Πλάτος των αλλαγών**

Παρατηρώντας το αποθετήριο της OpenWhisperSystems παρατηρήσαμε ότι δεν δέχονται εύκολα pull request και ότι ο χρόνος ελέγχου είναι αρκετά μεγάλος, αναμενόμενο βάσει της φύσης του έργου. Γι αυτό το λόγο προσπαθήσαμε η αλλαγή μας να περιοριστεί όσο το δυνατόν περισσότερο σε πλάτος, ψάχνοντας για την πιο αποδοτική υλοποίηση. Αυτή εκφράζεται με την αλλαγή ενός αρχείου κώδικα στο οποίο προστέθηκε μια επιπλέον κλάση και 3 μέθοδοι.

## **4. Ποιότητα υλοποίησης**

Θεωρούμε ότι ο κώδικας που γράψαμε είναι σύμφωνα με τις αρχές ποιότητας λογισμικού. Έχουν χρησιμοποιηθεί έννοιες όπως η αφαίρεση και η ενθυλάκωση έχουμε φροντίσει να μην δημιουργήσουμε νέες εξαρτήσεις μεταξύ πακέτων(κυρίως περιορίζοντας τις αλλαγές μας σε μια μόνο κλάση) και έχουμε εξασφαλίσει την πληρότητα(πχ δεν ψάχνουμε μόνο για ένα αριθμό ανα μήνυμα)Εξετάσαμε τον κωδικα που γράψαμε και απο την πλευρά της απλότητας και αναγνωσιμότητας, φροντίζοντας για τα κατάλληλα ονόματα κλάσεων, μεθόδων και μεταβλητών.Μένοντας πιστοί σε μια απο τις βασικές αρχές της ασφάλειας πληροφοριών, την ακεραιότητα(Integrity) και κατ' επέκταση και του TextSecure, φροντίσαμε η αλλαγή μας να μην την καταπατά, αφήνοντας το αρχικό μήνυμα ανεπηρέαστο και κρυπτογραφημένο στην βάση δεδομένων, ενώ τροποποιούμε αυτό που βλέπει ο χρήστης στην οθόνη.

## **5. Ολοκλήρωση**

Ολοκληρώσαμε την αλλαγή μας, επανασχεδιάζοντας κάποια κομμάτια(σε σχέση με τον σχεδιασμό που παρουσιάσαμε στην πρώτη παρουσίαση), με γνώμονα την ποιότητα λογισμικού και είχαμε τελικά τα αναμενόμενα αποτελέσματα.Σε αυτήν την ενότητα είχαμε και την επιπλέον επιτυχία, ότι επιλέξαμε την συνεισφορά μας βάσει του διαθέσιμου χρόνου εκτιμώντας με ακρίβεια τον αναμενόμενο χρόνο υλοποίησης.

## **6. Έλεγχος**

Ελέγξαμε την νέα λειτουργικότητα με σενάρια ελέγχου σε emulator αλλά και σε φυσικές συσκευές, κάτι το οποίο οδήγησε στην διόρθωση κάποιων σφαλμάτων και στην προσθήκη ελέγχων για ακραίες τιμές(πχ άδεια λίστα με γνωστούς τηλεφωνικούς αριθμούς). Unit testing πραγματοποιήθηκε για προσωπική χρήση στην μέθοδο που ψάχνει τηλεφωνικούς αριθμούς για να εξασφαλιστεί η σωστή λειτουργία του regex, ενώ είναι στα σχέδιά μας, αν το pull request γίνει δεκτό, να συνεισφέρουμε και το unit test.Στα υπόλοιπα κομμάτια της συνεισφοράς μας λόγω της φύσης τους δεν ήταν εύκολο να γίνει unit testing(πχ αναζήτηση ενός αριθμού στις επαφές).

## **7. Συνεργασία με την ομάδα ανάπτυξης**

Η συνεργασία μας με την ομάδα ανάπτυξης ήταν ελάχιστη. Δεν λάβαμε απάντηση στην δήλωση ενδιαφέροντός μας για την λειτουργικότητα που επιλέξαμε, την οποία εκδηλώσαμε στα σχόλια του αντίστοιχου issue στο github. Λόγω της έλλειψης συνεργασίας αυτοσχεδιάσαμε σε κάποια κομμάτια, όπως στο αν θα αντικαθιστούμε το νούμερο με το όνομα της επαφής ή θα το προσθέτουμε δίπλα. Σε κάθε περίπτωση σχεδιάσαμε έτσι την υλοποίηση ώστε να είναι δυνατό και το αντιστροφο με μικρές αλλαγές.

## 8. Ποιότητα παραδοτέων

Φροντίσαμε τα παραδοτέα μας να ακολουθούν τα ενδεδειγμένα πρότυπα. Σε επίπεδο κώδικα φροντίσαμε να ακολουθούνται συμβάσεις συγγραφής όπως η σειρά με την οποία εμφανίζονται τα `import`, το αν θα χρησιμοποιείται `space` ή `tab` για τα κενά, το μέγεθος της εσοχής, το μέγιστο μήκος γραμμής κλπ. Επίσης ακολουθήσαμε το `LowerCamelCase` πρότυπο για την ονοματοδοσία μεταβλητών και μεθόδων όπως προτείνει η ομάδα ανάπτυξης. Σχόλια εκτός `Javadoc`, φροντίσαμε να μην βάλουμε ακολουθώντας τις υποδείξεις της ομάδας ανάπτυξης, η οποία προτρέπει για την συγγραφή `self-documenting` κώδικα, τις οποίες υιοθετήσαμε. Σε επίπεδο καταχώρησης αλλαγών στο αποθετήριο, φροντίσαμε τα `commit messages` να είναι σαφή, πλήρη και συνοπτικά.

## 9. Οργάνωση στο GitHub (wiki, issues, commits, branching)

Αξιοποιήσαμε πλήρως το `github` και τις ευκολίες που μας προσέφερε. Δηλώσαμε ενδιαφέρον για ένα ανοιχτό `issue` αφήνοντας σχόλιο. Δημιουργήσαμε έναν οργανισμό στο οποίο το αποθετήριο κάναμε `fork` το αντίστοιχο της `WhisperSystems`. Σε αυτό δημιουργήσαμε ένα `branch` με το όνομα `contactNames` όπου δουλεύαμε και πραγματοποιούσαμε τις αλλαγές. Λόγω ενός προβλήματος που προέκυψε με μια `third-party`, ανοιχτού κώδικα πάλι, βιβλιοθήκη, αναγκαστήκαμε να προσθέσουμε ένα δεύτερο `remote`, στα τοπικά μας αποθετήρια, ώστε να μπορούμε να παίρνουμε τυχόν αλλαγές από το επίσημο αποθετήριο της `WhisperSystems`. Χρησιμοποιήσαμε και την λειτουργία `merge` για την επίλυση του ιδίου προβλήματος (`merge` από το πλέον συγχρονισμένο, με αυτό της `WhisperSystems`, `master` στο `contactNames`).

## 10. Επίλογος

Όλες οι συνεισφορές μας φαίνονται από τους προσωπικούς μας λογαριασμούς . Το `pull request` βρίσκεται σε αυτόν τον σύνδεσμο:

<https://github.com/WhisperSystems/TextSecure/pull/3352>.

Το `branch` στο οποίο δουλέψαμε βρίσκεται εδώ: <https://github.com/codeBusters-cs/TextSecure/tree/contactNames>

## 11. Παραπομπές

1. Πρωτόκολλο Ασφάλειας  
<https://github.com/WhisperSystems/TextSecure/wiki/ProtocolV2>
2. Threat Model <https://github.com/WhisperSystems/TextSecure/issues/782>
3. How secure is TextSecure? <https://eprint.iacr.org/2014/904.pdf>

