Ατομική Διπλωματική Εργασία

ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΑΝΤΑΛΛΑΓΗ ΔΕΔΟΜΕΝΩΝ ΜΕ ΑΣΦΑΛΕΙΑ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ

Κωνσταντίνος Κυπριανού

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση Εφαρμογής για Αποθήκευση και Ανταλλαγή Δεδομένων με Ασφάλεια στο Υπολογιστικό Νέφος

Κωνσταντίνος Κυπριανού

Επιβλέπουσα Καθηγήτρια
Άννα Φιλίππου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2015

Ευχαριστίες

Για την ολοκλήρωση αυτής της εργασίας δεν ήμουν μόνος σε καμία περίπτωση. Έτυχα καθοδήγησης, συμπαράστασης, εκπαίδευσης, συμβουλής και διόρθωσης από άτομα χωρίς τα οποία δεν θα έφερνα εις πέρας το έργο αυτό.

Αρχικά, θέλω να ευχαριστήσω την κ. Άννα Φιλίππου για την εξαιρετική καθοδήγηση και την ευκαιρία που μου παρείχε όσον αφορά την μελέτη ενός θέματος που τελικά βρήκα πολύ ενδιαφέρον. Όσες φορές ήρθαμε σε επαφή για κάποια επεξήγηση, διευκρίνιση ή διόρθωση, με αντιμετώπισε με κατανόηση και υπομονή.

Στη συνέχεια θα ήθελα επίσης να ευχαριστήσω τον κ. Βάσο Βασιλείου για τον εξαιρετικό χειρισμό και παράδοση του μαθήματος Ασφάλειας Πληροφοριών και Δικτύων. Παρακολούθησα αυτό το μάθημα το προηγούμενο εξάμηνο και είχα πολύ γερές βάσεις βοήθησαν τόσο στη μελέτη μου όσο και στην υλοποίηση.

Τέλος, θερμές ευχαριστίες προς τους φίλους και την οικογένειά μου για την ψυχική υποστήριξη και την πίστη τους σε μένα καθ' όλη τι διάρκεια των σπουδών μου. Δεν θα μπορούσα να παραλείψω το διαδίκτυο για την ψυχαγωγία και βοήθεια που μου προσφέρει, με διάφορους τρόπους. Ποιοτικά διαλείμματα ενισχύουν την απόδοση, ειδικά στην δική μου περίπτωση.

Περίληψη

Το γενικό θέμα που ανέλαβα να μελετήσω στην ΑΔΕ μου είναι το υπολογιστικό νέφος και θέματα ασφάλειας που το αφορούν. Το υπολογιστικό νέφος είναι πολύ διαδεδομένο στις μέρες μας· ακόμα και αν ο τυπικός ορισμός «υπολογιστικό νέφος» δεν είναι πολύ γνωστός, σχεδόν όλοι έχουμε να κάνουμε με αυτό καθημερινά είτε το καταλαβαίνουμε είτε όχι, ακόμη και μέσω υπηρεσιών όπως το Facebook. Το νέφος αφορά την χρήση πόρων μέσω του διαδικτύου από κάποιο προμηθευτή και οι οποίοι μπορεί να βρίσκονται οπουδήποτε. Οι πόροι αυτοί μπορεί να χρησιμοποιηθούν για οποιοδήποτε σκοπό όπως αποθήκευση δεδομένων, ή εκτελέσεις μικρών και μεγάλων υπολογισμών, και είναι θεωρητικά απεριόριστοι. Επομένως, αυτή η συναναστροφή του νέφους με ξένα δεδομένα τα οποία πιθανώς να είναι ευαίσθητα, επιφέρει την άμεση ανάγκη για μελέτη της ασφάλειάς τους. Ο γενικός όρος ασφάλεια περιλαμβάνει διάφορα πιο συγκεκριμένα θέματα τα οποία ασχολούνται με κακόβουλους τρίτους παράγοντες, και έχουν να κάνουν με ένα ευρύτερο σύνολο τομέων της Πληροφορικής, στο οποίο βρίσκεται και το υπολογιστικό νέφος. Ωστόσο, ο όρος αυτός περιλαμβάνει επίσης και ένα θέμα που σχετίζεται αποκλειστικά με το υπολογιστικό νέφος: κατά πόσο το ίδιο το νέφος αποτελεί κάποιο είδος κινδύνου, σημαντικού ή όχι, και σε ποιες περιπτώσεις. Η ιδιαιτερότητα του νέφους είναι ότι με το που οι χρήστες το χρησιμοποιούν προς όφελός τους δίνοντάς του πρόσβαση σε προσωπικά τους δεδομένα, ταυτόχρονα του δίνουν την ενδεχόμενη δυνατότητα να τους βλάψει. Ως αποτέλεσμα, ανέλαβα να ερευνήσω την ασφάλεια που έχουν ή δεν έχουν οι χρήστες όσον αφορά το υπολογιστικό νέφος, τα χαρακτηριστικά της και πώς μπορούν να την κατανοήσουν/πετύχουν. Επιπλέον, αφού μελέτησα τη σχετική βιβλιογραφία και σύγκρινα τις διάφορες ιδέες μου, κατέληξα σε ένα γενικό σύστημα που έχει προοπτικές για πιο συγκεκριμένη χρήση. Πρόκειται για ένα μοντέλο το οποίο έχει άμεση σχέση με την ιδέα της κοινωνικής δικτύωσης, πάντα προσφέροντας την απαραίτητη ασφάλεια απέναντι στο υπολογιστικό νέφος. Θα επιτρέπει στους χρήστες να ανταλλάσσουν ή να αποθηκεύουν δεδομένα κάτω από συνθήκες ασφαλείας, με προστασία των δεδομένων και των σχέσεων των χρηστών. Επιδεικνύει μια προσέγγιση που θα μπορούσε να εφαρμοστεί ως ένα κομμάτι μεγαλύτερου συστήματος, εφόσον τα χαρακτηριστικά της είναι επιθυμητά.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

Το υπολογιστικό νέφος (cloud computing) διαδίδεται με εξαιρετικά μεγάλους ρυθμούς στις μέρες μας· όλο και περισσότερος κόσμος κατανοεί και βλέπει τα οφέλη που είναι δυνατό να κερδίσει από όσα μπορεί να προσφέρει, ενώ πολλοί μπορεί να το χρησιμοποιούν χωρίς να το γνωρίζουν. Τα οφέλη αυτά βρίσκονται γύρω από την χρήση πόρων μέσω του διαδικτύου από κάποιο προμηθευτή οι οποίοι μπορεί να χρησιμοποιηθούν για οποιοδήποτε σκοπό όπως αποθήκευση δεδομένων, ή εκτελέσεις μικρών και μεγάλων υπολογισμών. Επιπλέον, δεν γνωρίζουμε την ακριβή φυσική τους τοποθεσία, και είναι θεωρητικά απεριόριστοι, γεγονότα που απαλείφουν ευθύνες από εμάς. Θεωρείται μία από τις πιο σημαντικές στροφές στην τεχνολογία της Πληροφορικής και ανοίγει πόρτες προς νέα είδη υπηρεσιών για όλο τον κόσμο με κάθε άλλο παρά πολύπλοκο τρόπο.

Οι υπηρεσίες στο νέφος (cloud services) είναι προσβάσιμες από οπουδήποτε και μέσω οποιασδήποτε συσκευής (π.χ. τάμπλετς, έξυπνα τηλέφωνα κλπ.), εφόσον έχουμε συνδεθεί στο διαδίκτυο χωρίς να απαιτούν καμία επαφή πρόσωπο με πρόσωπο. Δεν ενδιαφέρει κατ' ουδέν λόγο τον χρήστη η τοποθεσία που βρίσκονται τα δεδομένα του, ή αν υπάρχει περίπτωση να χαθούν αφού τέτοιου είδους προστασία είναι εγγυημένη. Για παράδειγμα, μια πολύ γνωστή υπηρεσία που διατίθεται στο ανοιχτό κοινό είναι η παροχή θεωρητικά άπειρου χώρου αποθήκευσης στο νέφος. Ακόμα όπως αναφέρεται στο [2], με το υπολογιστικό νέφος απαλείφεται και το τυχόν πρόβλημα έλλειψης υπολογιστικής

δύναμης. Το προαναφερθέν πρόβλημα υπό κανονικές συνθήκες θα έπρεπε να τύχει επίλυσης από το αντίστοιχο άτομο/α ή εταιρεία, με ρίσκο για άσχημες οικονομικές συνέπειες αν προμηθευτεί λιγότερη ή ακόμα και περισσότερη υπολογιστική δύναμη από αυτή που τελικά χρειαζόταν. Είναι εύκολη η σύλληψη του γεγονότος ότι με την απαλοιφή αυτού του ρίσκου και με την εμφάνιση τέτοιων υπηρεσιών, αποκτούμε πρόσβαση σε κάτι που θα θεωρούνταν όνειρο πριν μερικά χρόνια.

Η εισαγωγή αυτών των εννοιών στην ήδη ανεπτυγμένη διαδικτυακή κοινωνία, μας ωθούν ακόμα πιο μπροστά με πιο γοργούς ρυθμούς αφού όλο και περισσότερος κόσμος δύναται να αρπάξει ευκαιρίες. Ένα πολύ κύριο χαρακτηριστικό της χρήσης των υπηρεσιών του υπολογιστικού νέφους είναι η ελαστικότητά τους όσον αφορά οικονομικά θέματα, πράγμα που ολοκληρώνει περεταίρω το σύνολο των θετικών σημείων που ήδη υπάρχουν.

Με την ταχύτατη ανάπτυξη και διάδοση της έννοιας του υπολογιστικού νέφους, όλο και περισσότερα νέφη και υπηρεσίες δημιουργούνται από διάφορους οργανισμούς τα οποία γίνονται διαθέσιμα στο κοινό, με απώτερο στόχο την βέλτιστη ικανοποίηση των αναγκών του.

1.1 Κίνητρο

Με την εισαγωγή του υπολογιστικού νέφους στο πεδίο της Πληροφορικής, δημιουργείται η εντατική ανάγκη για μελέτη των ποικίλων θεμάτων ασφάλειας που το συνοδεύουν. Το νέφος διαχειρίζεται δεδομένα πελατών, και η ασφάλεια πληροφοριών είναι ένας πολύ σημαντικός τομέας της Πληροφορικής όπως και το κύριο θέμα συζήτησης αυτού του έγγραφου. Λαμβάνοντας υπόψη και την ραγδαία διάδοση του υπολογιστικού νέφους, είναι υποχρεωτικό να αυξηθεί ταυτόχρονα, ή πολύ περισσότερο, και η σημασία που δίνουμε στα θέματα ασφαλείας. Είναι αναμενόμενο όταν κάτι επεκταθεί πολύ και με μεγάλους ρυθμούς, να γίνει επίσης πιο εύκολα και συχνά στόχος από διάφορους κακόβουλους. Επομένως, αξίζει η μελέτη επί αυτού του θέματος για περεταίρω κατανόηση και αντιμετώπισή του εκεί που χρειάζεται.

1.2 Στόχος

Στόχος μας σε αυτό το έγγραφο είναι αρχικά να μελετήσουμε το υπολογιστικό νέφος σε βάθος, πριν να στρέψουμε το ενδιαφέρον μας προς τα θέματα ασφαλείας. Από τα παραπάνω θέματα, θα προχωρήσουμε στο να τα κατανοήσουμε και να αναγνωρίσουμε αυτά που χρειάζονται αντιμετώπιση. Μετά από την απαιτούμενη προηγούμενη διαδικασία θα ασχοληθούμε σε βάθος συγκεκριμένα με την ασφάλεια των χρηστών απέναντι στο ίδιο το υπολογιστικό νέφος. Αφού μελετήσουμε το πρόβλημα, θα ακολουθήσει περιγραφή ενός συστήματος το οποίο δημιουργήσαμε και υλοποιήσαμε που έχει ως κύριο στόχο να προσεγγίσει όσο γίνεται το πρόβλημα ασφαλείας που αναφέραμε.

Το σύστημα αυτό έχει χαρακτηριστικά κοινωνικής δικτύωσης και επιτρέπει στους χρήστες να ανταλλάσσουν πληροφορίες της επιλογής τους με άλλους χρήστες που επιθυμούν. Το εν λόγω σύστημα υλοποιήθηκε σε λειτουργικό Android. Όλα αυτά γίνονται κάτω από προστασία από το υπολογιστικό νέφος, με χρήση τεχνικών που θα συζητήσουμε αναλυτικά στη συνέχεια. Επιπλέον, μέσα σε αυτή τη συζήτηση θα περιλαμβάνονται και τρόποι που μπορεί να χρησιμοποιηθούν γενικότερα σε εφαρμογές που θέλουν να κάνουν χρήση του υπολογιστικού νέφους, και ταυτόχρονα να έχουν την ασφάλεια που επιθυμούν.

1.3 Μεθοδολογία

Ξεκινώντας, έκανα έρευνα πάνω στο νέφος για να κατανοήσω πλήρως την έννοια, αφού μέχρι τότε είχα έρθει σε αμυδρή επαφή με την έννοια και ήξερα μόνο ό,τι ήταν προφανές. Επομένως, για να επεκτείνω την επιφανειακή εντύπωση που είχα για το όλο θέμα, μελέτησα άρθρα που μιλούσαν για το υπολογιστικό νέφος και το ρόλο του σήμερα στη διαδικτυακή ζωή μας. Αρχικά ήταν ενδεικτικά και βασικά άρθρα που μου δόθηκαν από την κ. Φιλίππου, και στη συνέχισα με προσωπική μου έρευνα για διεύρυνση των γνώσεων μου σε αυτό το θέμα. Ακολούθως, προχώρησα στα θέματα ασφαλείας που αφορούν το υπολογιστικό νέφος, τα οποία δίνουν μια πολύ ολοκληρωμένη εικόνα του ότι τίποτα δεν είναι τέλειο και ότι παντού υπάρχουν τα «bad guys». Μετά από αυτό το στάδιο, έστρεψα το ενδιαφέρον μου στο συγκεκριμένο θέμα της ασφάλειας των χρηστών

από το ίδιο το νέφος αντί από τρίτους παράγοντες, αφού αυτό είναι μια σημαντική πτυχή που πολλοί αγνοούν, είτε γνωρίζοντάς το είτε όχι. Κατά τη διάρκεια αυτού του σταδίου, μελετούσα και το θέμα της ασφάλειας πληροφοριών (κρυπτογράφηση δεδομένων κλπ.) μέσω του μαθήματος Ασφάλειας Πληροφοριών και Δικτύων που παρακολουθούσα και προσωπικής μελέτης. Με αυτές της γνώσεις ήμουν σε θέση να κατανοήσω διάφορες τεχνικές που μπορούν να χρησιμοποιηθούν για να αντιμετωπίσουν σε κάποιο βαθμό αυτό το πρόβλημα. Έπειτα, μελέτησα υλοποιήσεις αυτών των τεχνικών σε υπάρχοντα πρωτόκολλα, με στόχο να αξιολογήσω κατά πόσο μπορούν να εφαρμοστούν στο δικό μου σύστημα το οποίο θα είχε χαρακτηριστικά κοινωνικής δικτύωσης. Σε αυτή μου την προσπάθεια, πειραματίστηκα με διάφορες προσεγγίσεις μέχρι να καταλήξω σε αυτή που κρίνω τόσο ορθή όσο και απαιτητική. Τέλος, έφτασα σε ένα σύστημα σε λειτουργικό Android που υλοποιεί αρκετές από τις πολιτικές που μελέτησα σε αυτόν τον τομέα.

1.4 Δομή Εργασίας

Σε αυτό το έγγραφο αρχικά ξεκινώντας με το κεφάλαιο 2 θα μιλήσουμε αναλυτικά για το υπολογιστικό νέφος, τα χαρακτηριστικά του, και τα οφέλη που προσφέρει. Στη συνέχεια, προχωρώντας στο κεφάλαιο 3 θα επικεντρωθούμε στο θέμα της ασφάλειας γύρω από το υπολογιστικό νέφος και τις προσεγγίσεις τους, πριν αφοσιωθούμε πλήρως στην ασφάλεια αποκλειστικά από το υπολογιστικό νέφος. Σε αυτό το σημείο θα μελετήσουμε διάφορες τεχνικές που προτείνονται για αντιμετώπιση του προβλήματος, όπως επίσης και ένα σύστημα που εφαρμόζει κάποιες από αυτές. Το συγκεκριμένο σύστημα είναι και η βάση μας για την δική μας υλοποίηση. Έπειτα, μεταβαίνοντας στο κεφάλαιο 4 θα συζητήσουμε τα διάφορα εργαλεία και τεχνικές που μας είναι απαραίτητα στην υλοποίηση, αφού προϋποτίθενται γνώσεις από το πεδίο Ασφάλεια Πληροφοριών για τέτοιο σύστημα. Τέλος, στο κεφάλαιο 5 θα ακολουθήσει πλήρης περιγραφή και ανάλυση του συστήματος, και παρουσίαση των λειτουργιών.

Κεφάλαιο 2

Το Υπολογιστικό Νέφος (Cloud Computing)

Μετά από την προηγούμενη περιληπτική περιγραφή του υπολογιστικού νέφους, καταλαβαίνουμε ότι με αυτές και άλλες δυνατότητες στη διάθεσή μας, μας παρέχονται ευέλικτοι τρόποι για ανάπτυξη πολλών διαδικτυακών υπηρεσιών ή επεξεργασίας μαζών δεδομένων κ.ά., απλά, οικονομικά και αποδοτικά. Ωστόσο, πρέπει να έχουμε κατανοήσει καλά το υπολογιστικό νέφος πριν να μπορέσουμε να το χρησιμοποιήσουμε σωστά, έτσι ώστε να πετύχουμε τους στόχους μας με σωστό και ασφαλή τρόπο. Στη συνέχεια, θα συζητήσουμε σε βάθος τι εστί υπολογιστικό νέφος και πιο τυπικά αυτά που μπορεί να μας προσφέρει.

2.1 Σημασιολογία και Ορισμός

Για να εξηγήσουμε τον ορισμό υπολογιστικό νέφος θα ξεκινήσουμε σπάζοντάς τον σε δυο μέρη, με πληροφορίες από τα [2,9]. Το πρώτο μέρος, το νέφος (cloud), είναι βασικά το υλικό (πολλοί υπολογιστές) και λογισμικό (π.χ. ειδικά virtual machines) που αποτελούν το κέντρο δεδομένων (data center) του προμηθευτή του νέφους (cloud provider) ο οποίος το παραθέτει για χρήση από το ανοιχτό κοινό. Όταν λέμε προμηθευτής του νέφους, δεν μιλάμε συγκεκριμένα για ένα άτομο, αλλά για την εταιρία ή οργανισμό που είναι υπεύθυνη και διαχειρίζεται το εν λόγω νέφος. Αυτή η διαχείριση περιλαμβάνει τον τρόπο και τις πολιτικές βάσει των οποίων δεσμεύονται/χρησιμοποιούνται/απελευθερώνονται οι πόροι, και άλλα θέματα πάνω στα οποία δεν έχουν δικαιοδοσία οι χρήστες. Ο προμηθευτής του νέφους είναι αυτός στον

οποίο αποτίθεται κάποιος που θέλει να χρησιμοποιήσει τους πόρους που προσφέρει για να διευθετήσουν όλες τις λεπτομέρειες που μπορεί να είναι απαραίτητες. Τα κέντρα δεδομένων ενός προμηθευτή λειτουργούν με την λογική του grid computing, δηλαδή σε κάθε κόμβο μπορεί να ανατεθεί διαφορετική διεργασία και οι κόμβοι δεν είναι καν απαραίτητο να βρίσκονται στο ίδιο γεωγραφικό σημείο.

Πάνω στο υλικό που υπάρχει στα κέντρα δεδομένων είναι εγκατεστημένες οι υπηρεσίες που παρέχονται από κάποιο χρήστη στο κοινό (άλλους χρήστες) μέσω ενός συγκεκριμένου νέφους. Αυτές οι υπηρεσίες, σύμφωνα με το [5], ονομάζονται Software as a Service (SaaS), είτε ο χρήστης είναι ο παροχέας είτε απλός χρήστης μιας υπηρεσίας. Οι παροχείς των υπηρεσιών είναι αυτοί που προμηθεύονται αυτά που χρειάζονται από έναν προμηθευτή νέφους όπως αναφέρθηκε πιο πάνω, άρα είναι αυτοί που πληρώνουν. Οι απλοί χρήστες μιας υπηρεσίας δεν προϋποτίθεται να πληρώνουν για κάτι εκτός και αν το απαιτεί η συγκεκριμένη υπηρεσία, αφού υπάρχουν οι δωρεάν υπηρεσίες και αυτές που είναι εκ πληρωμής (π.χ. για εγγραφή).

Τώρα με το υπολογιστικό (computing) μέρος του ορισμού, εννοούμε τον σκοπό που δημιουργήθηκε η συγκεκριμένη ορολογία, που είχε και ως επακόλουθο να υπάρχουν όλα αυτά τα νέφη σήμερα. Αυτός ο σκοπός, που αναφέρθηκε γενικά με παραδείγματα και πιο πάνω, είναι για να χρησιμοποιούνται οι υπηρεσίες του νέφους από αυτούς που τις χρειάζονται. Με αναφορά στο [2], η υπηρεσία αυτή που πωλείται από τους προμηθευτές χαρακτηρίζεται ως υπολογιστική ωφέλεια (utility computing) και παρέχεται στους πελάτες με ένα pay-as-you-go τρόπο, παρόλο που δεν είναι ακριβώς υπολογιστική ωφέλεια. Όταν λέμε υπολογιστική ωφέλεια εννοούμε την παροχή πόρων (υπολογιστικών, αποθηκευτικών κλπ.) και υποδομής (αξιοπιστία, ασφάλεια) σε άτομα για προσωπική χρήση. Αυτοί είναι και οι ορισμοί των Infrastructure as a Service (IaaS) και Platform as a Service (PaaS) αντίστοιχα.

Η υπολογιστική ωφέλεια ως ξεχωριστός ορισμός δεν προϋποθέτει νέφος αφού μπορεί να υλοποιηθεί σε οποιοδήποτε περιβάλλον, ακόμα και έναν τοπικό εξυπηρετητή (server). Ένα καλό παράδειγμα υπολογιστικής ωφέλειας είναι το ηλεκτρικό ρεύμα που έχουμε στα σπίτια μας, αφού ο τρόπος που το αποκτούμε και πληρώνουμε για αυτό είναι ακριβώς υπολογιστική ωφέλεια. Οι εμπλεκόμενοι πόροι/υπηρεσίες παρέχονται με ένα

πολύ ευέλικτο τρόπο πληρωμής, συνήθως χωρίς αρχικό κόστος και με το τελικό ποσό να βασίζεται εξ' ολοκλήρου στη συνολική χρήση των πόρων αντί σε σταθερό ποσό ανά μονάδα χρόνου, γεγονός που κάνει πολύ πιο κοινή και εύκολη την χρήση των νεφών.

Με τον όρο δημόσιο νέφος (public cloud) αναφερόμαστε στο νέφος που διατίθεται στο κοινό επί πληρωμής, πράγμα που στην ουσία φαντάζεται και αναμένει κάποιος με το που γίνεται οικείος με το θέμα. Αυτό είναι και το είδος νέφους που όντως υπάρχει έξω στο διαδίκτυο διαθέσιμο σε εμάς, αλλά υπάρχουν και τα ιδιωτικά νέφη (private clouds) τα οποία αναφέρονται σε εσωτερικά κέντρα δεδομένων κάποιας εταιρίας ή οργανισμού που δεν είναι στη διάθεση του γενικού κοινού. Όπως αναφέρεται στο [3] όμως, για να μπορεί να θεωρηθεί το κέντρο δεδομένων ενός οργανισμού ως νέφος πρέπει να είναι σε θέση που να ικανοποιεί τα πρότυπα που πληροί και ένα δημόσιο νέφος σε κάποιο βαθμό σε όλους τους τομείς, με περισσότερη έμφαση να πηγαίνει στο μέγεθος και στην υπολογιστική δύναμη.

Συνεπώς, το υπολογιστικό νέφος μπορεί να χαρακτηριστεί ως το άθροισμα της υπολογιστικής ωφέλειας (IaaS και PaaS) και SaaS. Η όλη αυτή έννοια εξελίχθηκε και διαδόθηκε αρκετά προς το καλύτερο και προς το όφελος του χρήστη, με όλο και περισσότερη ευκολία στην πρόσβαση από τον οποιονδήποτε. Η εν λόγω ευκολία, εκτός του ότι βοηθά τον καθένα (άτομο ή οργανισμό) ξεχωριστά, έχει σαν ολικό επακόλουθο και την ανάπτυξη ολόκληρης της διαδικτυακής κοινότητας.

2.2 Χρήσεις και Ωφελήματα

Αυτοί τους οποίους οι εισαγωγή του υπολογιστικού νέφους επηρέασε ριζικά είναι κοινώς οι προγραμματιστές ή πιο γενικά σχετικοί οργανισμοί. Αυτό ισχύει επειδή είναι αυτοί που έχουν ανάγκη τους πόρους και την υποδομή για ανάπτυξη/υποστήριξη υπηρεσιών ή χρήση υπολογιστικής δύναμης για μεγάλα προγράμματά τους που έχουν πολύ φόρτο είναι πολύ καλή περίπτωση.

Ένα παράδειγμα που αναφέρεται στο [9], είναι αν κάποιος θέλει να διαθέσει μια διαδικτυακή υπηρεσία που ανέπτυξε ανοιχτή για χρήση από το κοινό, μπορεί πολύ απλά

να προμηθευτεί τους πόρους που χρειάζεται για την εγκατάστασή της (hosting) εύκολα και οικονομικά σε ένα νέφος. Ακόμα, είναι δυνατό να χρησιμοποιήσει το νέφος για την ανάπτυξη της εφαρμογής αφού μπορεί να του παρασχεθούν και τα εργαλεία που χρειάζεται (γλώσσες προγραμματισμού κ.ά.) από το ίδιο το νέφος. Επίσης, θα έχει την ευελιξία για επέκτασή των πόρων εφόσον η ζήτηση και επιτυχία της εφαρμογής του επιφέρει αυτή την ανάγκη. Αυτό τον αποδεσμεύει από το να επιβαρυνθεί ο ίδιος με αυτή την ευθύνη που ταυτόχρονα αποτελεί και μεγάλο ρίσκο σε περίπτωση όπου αποτύχει αυτή του η απόπειρα για κέρδος από την υπηρεσία που ανέπτυξε. Ή αν ένας οργανισμός θέλει να εκτελέσει διεργασίες μεγάλης πολυπλοκότητας ή πολλών δεδομένων που απαιτεί τεράστια υπολογιστική δύναμη, δεσμεύει από ένα νέφος τους πόρους που χρειάζεται για όσο τους χρειάζεται, με την οικονομική ελαστικότητα που περιγράψαμε. Και όλα αυτά, με εγγύηση ασφάλειας ως επίσης και σχεδόν μηδενικής πιθανότητας για σφάλματα, αφού οι προμηθευτές του νέφους μας παρέχουν υποδομή ποιότητας με τα απαιτούμενα αντίγραφα ασφαλείας των δεδομένων κλπ., χωρίς καν να πρέπει να διευκρινίσουμε κάτι εμείς.

Μετά την περιγραφή του τι είναι το υπολογιστικό νέφος, όπως και αυτών που μας προσφέρει μέσω παραδειγμάτων, θα καταγράψουμε πιο τυπικά τα χαρακτηριστικά του, κάποια από τα οποία δεν είναι καν διακριτά στον χρήστη αλλά αυτονόητο ότι υπάρχουν. Οι πληροφορίες προέρχονται κυρίως από τα [3,5]:

- Κάθε χρήστης μπορεί να αποκτήσει πρόσβαση στις υπηρεσίες/πόρους του υπολογιστικού νέφους εξ' αποστάσεως και χωρίς καθόλου επαφή πρόσωπο με πρόσωπο με οποιονδήποτε προμηθευτή.
- Όλα αυτά δεν προϋποθέτουν κάποιο εξειδικευμένο εξοπλισμό για να μας δοθεί πρόσβαση, αφού όλα είναι εφικτά και μέσω της πιο κοινής κινητής τηλεφωνικής συσκευής.
- Ένα υπολογιστικό νέφος πρέπει να είναι σε θέση να ικανοποιήσει πολλαπλούς χρήστες, με σωστές πολιτικές για διαμοιρασμό των πόρων σωστά. Αυτό είναι κάτι για το οποίο μπορεί να μην έχει επίγνωση ένας χρήστης, αφού για παράδειγμα οι πόροι που χρησιμοποιεί μπορεί να μην είναι καν στην ίδια περιοχή ή ακόμα και χώρα.

- Ο χρήστης θα πρέπει να είναι σε θέση ανά πάσα στιγμή να δεσμεύσει περισσότερους πόρους στο νέφος. Άρα, δεν πρέπει σε καμία περίπτωση να του δοθεί η εντύπωση ότι αυτό δεν είναι εφικτό. Για αποφυγή του προηγούμενου, κάποια αυτοματοποίηση και «εξυπνάδα» θα πρέπει να είναι υλοποιημένη για να φτάσουμε την επιθυμητή ελαστικότητα του συστήματος.
- Τα δεδομένα του χρήστη πρέπει να είναι ασφαλισμένα σε πολλαπλά επίπεδα, τόσο από εξωτερικούς κακόβουλους παράγοντες προς το νέφος όσο και από τυχόν βλάβες του συστήματος. Η παροχή και των δύο ειδών ασφάλειας επιτυγχάνεται με αντίγραφα ασφαλείας των δεδομένων, firewalls κλπ. Επιπλέον, μετά από κάποια αποκατάσταση του συστήματος έπειτα από κάποια βλάβη, θα πρέπει να δημιουργείται καινούργιο ενημερωμένο αντίγραφο ασφαλείας.

Κεφάλαιο 3

Ασφάλεια και Νέφος

Σε αυτό το κεφάλαιο θα δούμε το θέμα της ασφάλειας από πολλές οπτικές γωνίες. Θα μιλήσουμε αρχικά γενικά με κατανοητή προσέγγιση εξηγώντας περιληπτικά κάποια θέματα ασφαλείας, πριν επικεντρωθούμε στο σημείο που μας αφορά περισσότερο. Για το συγκεκριμένο θα δούμε διάφορες τεχνικές όπως επίσης και ένα παράδειγμα συστήματος που εφαρμόζει κάποιες από αυτές για να πετύχει τον στόχο του όσον αφορά την ασφάλεια, και τέλος την δική μας προσέγγιση.

3.1 Θέματα Ασφάλειας και Αντιμετώπιση

Ο όρος ασφάλεια είναι πολύ γενικός και διαχωρίζεται σε άλλους υποτομείς πιο συγκεκριμένους από τους οποίους ο καθένας αναφέρεται σε μια συγκεκριμένη πτυχή του θέματος. Θα αναφερθούμε και θα εξηγήσουμε τους πιο κοινούς από αυτούς, όσον αφορά

την αμεσότητα στη συσχέτιση που έχουν με το υπολογιστικό νέφους, με αναφορά κυρίως στα [4,7]:

- **Κακοποίηση και δόλια χρήση του υπολογιστικού νέφους (Abuse and Nefarious Use of Cloud Computing):**

  Πολλοί προμηθευτές προσφέρουν ένα πολύ «αδύναμο» και κάποτε ανώνυμο σύστημα εγγραφής για χρήση των υπηρεσιών τους· ο καθένας που έχει μια έγκυρη πιστωτική κάρτα μπορεί να πάρει πρόσβαση, ή ακόμα και μέσω «free trial». Πολλά κακόβουλα άτομα εκμεταλλεύονται αυτό το γεγονός και εγκαθιστούν το κακόβουλο λογισμικό τους στο νέφος απαρατήρητοι, ή χρησιμοποιούν τους πόρους για σπάσιμο κλειδιών και άλλες κακόβουλες ενέργειες. Αυτά έχουν ως αποτέλεσμα να απειλούν τους ίδιους τους προμηθευτές και την ακεραιότητα των υπηρεσιών τους, ως επίσης και τα δεδομένα πελατών. Αρκετά μέτρα λαμβάνονται ήδη στην προσπάθεια για πρόληψη αυτού του θέματος όπως πιο αυστηρό και λεπτομερές σύστημα εγγραφής, έλεγχο στα πακέτα δεδομένων των πελατών, τις διεργασίες τους, κ.ά.

- **Ανασφαλείς διεπιφάνειες και APIs (Insecure Interfaces and APIs):**

  Οι προμηθευτές παρέχουν κάποιες διεπιφάνειες και APIs για να χρησιμοποιούν οι πελάτες στην αλληλεπίδρασή τους με τις υπηρεσίες που παρέχονται. Επομένως, η ασφάλεια/διαθεσιμότητα αυτών των APIs και των διεπιφανειών αντανακλάται στην ασφάλεια των υπηρεσιών, επομένως πρέπει να είναι σχεδιασμένες σωστά. Για να αποφευχθούν προβλήματα τέτοιας κατηγορίας λαμβάνεται πολύ σοβαρά η σχεδίαση όσον αφορά το μοντέλο ασφάλειας αυτών των διεπιφανειών/APIs, προσοχή σε άλλες δεσμεύσεις που μπορεί να χρειάζονται με τρίτους (παροχείς κάποιων υπηρεσιών), κ.ά.

- **Κακόβουλοι εσωτερικοί παράγοντες (Malicious Insiders):**

  Κακόβουλο εσωτερικό παράγοντα σε ένα οργανισμό ορίζουμε ένα νυν ή πρώην υπάλληλο, ή άλλο συνεργάτη που είχε/έχει εξουσιοδότηση και πρόσβαση σε εμπιστευτικές πληροφορίες. Η έκταση της ζημιάς που

μπορεί να προκαλέσει ένας τέτοιος παράγοντας εξαρτάται πλήρως από το επίπεδο της εξουσιοδότησης που έχει. Για πρόληψη αυτού του είδους κινδύνου όλες οι εξουσιοδοτημένες ενέργειες καταγράφονται, καθορίζονται περιορισμοί στα συμβόλαια, αφαιρούνται από πρώην εξουσιοδοτούμενους κ.ά.

- **Ευπάθεια κοινόχρηστης τεχνολογίας (Share Technology Vulnerabilities):**

  Οι υπηρεσίες που προσφέρονται στο νέφος είναι βασισμένες στο ότι το υλικό και λογισμικό τους μοιράζονται ανάμεσα στους πελάτες. Πολλές επιθέσεις έγιναν προσπαθώντας να εκμεταλλευτούν το γεγονός ότι τα διάφορα μέρη του υλικού δεν σχεδιάστηκαν για να προσφέρουν ιδιωτικότητα στα δεδομένα κάθε χρήστη. Για αντιμετώπιση του παραπάνω προβλήματος απαιτούνται περισσότερα μέτρα ασφαλείας από τους προμηθευτές.

- **Απώλεια δεδομένων (Data Loss):**

  Με τη χρήση του υπολογιστικού νέφους εμπιστευόμαστε τα δεδομένα μας στο νέφος και επομένως θα είναι αποθηκευμένα μακριά από μάς. Απώλεια δεδομένων στο cloud ονομάζουμε την ακούσια διαγραφή τους από τον προμηθευτή, ή τον χαμό τους από κάποια φυσική καταστροφή στο κέντρο δεδομένων. Ένας απλός και αποδοτικός τρόπο αντιμετώπισης των πρώτων δύο περιπτώσεων είναι ή εφαρμογή αντιγράφων λειτουργιών από τον προμηθευτή όπως περιεγράφηκε προηγουμένως.

- **Διαρροές δεδομένων (Data Breaches):**

  Το πρόβλημα αυτό αναφέρεται στην έκθεση των δεδομένων μας σε λάθος άτομα παράνομα. Όμως με το υπολογιστικό νέφος ήρθαν στην επιφάνεια κάποια ενδεχόμενα για παραβίαση της ιδιωτικότητας των πελατών μέσω αδυναμιών των εφαρμογών που χρησιμοποιούν οι ίδιοι ή των virtual machines στους servers του cloud. Για αντιμετώπιση αυτού του κινδύνου θέτονται σε ισχύ ισχυρές κρυπτογραφικές τεχνικές για προστασία της ιδιωτικότητας και ακεραιότητας των δεδομένων των χρηστών.

- **Παραβίαση/πειρατεία λογαριασμών ή υπηρεσιών (Account or Service Hijacking):**

  Αυτό το πρόβλημα δεν είναι καινούργιο, επειδή το γεγονός παραβίασης των διακριτικών (username, password) μας πάντα ήταν απειλή. Με το υπολογιστικό νέφος όμως, εάν τύχει κάτι τέτοιο μπορεί να φτάσει σε ακόμη χειρότερο βαθμό για τα δεδομένα μας, ή ακόμα περισσότερο να χρησιμοποιηθούμε «εμείς» ως βάση για νέες επιθέσεις. Στην προσπάθεια για αποφυγή τέτοιων περιστατικών εφαρμόζονται 2-way τεχνικές ταυτοποίησης και δίνεται μεγάλη προσοχή στην διαχείριση των διακριτικών.

- **Ασαφής βαθμός ασφάλειας και άγνοια ρίσκου (Unknown Risk Profile):**

  Πολλοί οργανισμοί προχωρούν στον ισχυρισμό ότι προσφέρουν όλα όσα θα έπρεπε να προσφέρει ένας προμηθευτής αγνοώντας πολλές πτυχές του όλου θέματος. Χωρίς να έχουν διαθέσιμους τους απαραίτητους πόρους που χρειάζονται υπάρχει το ενδεχόμενο να υστερούν σε θέματα ασφάλειας ή άλλα, που ούτε οι ίδιοι έχουν επίγνωση με αποτέλεσμα αυτό να επιφέρεται και στους πελάτες τους. Αυτό το πρόβλημα αν δεν τακτοποιηθεί από τον υπεύθυνο οργανισμό, θα μπορούσε να δημιουργήσει προβλήματα γι’ αυτό και γίνονται ελέγχει ασφαλείας των διάφορων νεφών.

- **Άρνηση Υπηρεσίας (Denial of Service)**

  Οι επιθέσεις άρνησης υπηρεσίας έχουν ως απώτερο στόχο να αποτρέψουν τους χρήστες των διάφορων υπηρεσιών από το να έχουν πρόσβαση στα δεδομένα ή τις εφαρμογές τους. Αυτές οι επιθέσεις αναγκάζουν το σύστημα να χρησιμοποιεί σχεδόν όλους τους πόρους του με αποτέλεσμα το σύστημα να επιβραδύνεται πολύ και αυτό έχει τεράστιο αντίκτυπο στους νόμιμους χρήστες. Αυτές οι επιθέσεις αντιμετωπίζονται μέσω διάφορων μηχανισμών που εξετάζουν την κυκλοφορία δεδομένων προς ένα σύστημα και από πού έρχεται.

Όλοι από τους αναφερόμενους υποτομείς ασφάλειας έχουν υπάρξει αρκετά πριν από την εισαγωγή και ακμή του υπολογιστικού νέφους, και ως επακόλουθο, έτυχαν μελέτης νωρίτερα. Σύμφωνα με το [6], λόγω αυτού υπάρχει ήδη ένας σεβαστός αριθμός διαθέσιμων λύσεων και τρόπων άμυνας, όσον αφορά αυτά τα προβλήματα, με αποτέλεσμα να θεωρούνται λυμένα. Κάποιες από τις λύσεις και τους τρόπους άμυνας αναφέρθηκαν περιληπτικά πιο πάνω όταν περιγράψαμε σε συντομία τα προβλήματα. Επίσης, παρόλο που όλοι αυτοί οι υποτομείς είναι πολύ κρίσιμοι για τη σωστή και ασφαλή λειτουργία των υπηρεσιών στο νέφος, δεν είναι προβλήματα που αφορούν αποκλειστικά το υπολογιστικό νέφος, αφού εμφανίζονται και σε υπηρεσίες άλλων πεδίων της πληροφορικής. Για να είναι σε θέση ένας προμηθευτής να προσφέρει μια υγιή υπηρεσία στους πελάτες του πρέπει να φροντίσει να πάρει τα απαραίτητα μέτρα για την προφύλαξη απέναντι σε αυτά τα προβλήματα, αφού οι πελάτες δεν πρέπει να έχουν τίποτα να κάνουν με αυτά, παρά να τα θεωρούν ήδη εγγυημένα λυμένα.

3.2 Ασφάλεια από το Νέφος

Το πρόβλημα που εμφανίζεται σε αυτό το σημείο είναι πέρα όλων αυτών, δεν αναφέρθηκε προηγουμένως, και επίσης έχει το μοναδικό χαρακτηριστικό ότι αφορά αποκλειστικά το υπολογιστικό νέφος και όλα όσα έχουν να κάνουν με αυτό. Μπορεί να θεωρηθεί μια επέκταση του προβλήματος που αφορά τους κακόβουλους εσωτερικούς παράγοντες σε έναν οργανισμό, αφού στην ουσία έχει να κάνει με το γεγονός ότι ο εν λόγω προμηθευτής του νέφους έχει πρόσβαση στα δεδομένα των χρηστών. Η διαφορά όμως όπως αναφέρεται στο [6], είναι στο ότι αν το όλο σύνολο του προμηθευτή (όλος ο οργανισμός) είναι διεφθαρμένος και εσκεμμένα θέλει να πράξει κακόβουλα, μπορεί πολύ εύκολα να το κάνει.

Για περεταίρω εξήγηση, είναι κατανοητό ότι με το που εκμεταλλευόμαστε τις υπηρεσίες του υπολογιστικού νέφους για τους δικούς μας σκοπούς, έμμεσα θέτουμε τον εαυτό μας και τα πιθανώς σημαντικά δεδομένα μας σε κίνδυνο. Δηλαδή, με το που εμπιστευόμαστε πληροφορίες σε εξωτερικό παράγοντα για τον ένα ή τον άλλο λόγο, ταυτόχρονα βασιζόμαστε και στην αξιοπιστία του ίδιου του παράγοντα, και αυτό δεν είναι καθόλου υπό τον έλεγχό μας. Αν για παράδειγμα ο προμηθευτής του υπολογιστικού

νέφους είναι ένας οργανισμός ή μια εταιρεία, τότε ουσιαστικά πολλοί αν όχι όλοι οι παράγοντές του μπορεί να έχουν πρόσβαση στις πληροφορίες/δεδομένα μας. Κάποιος μπορεί να ισχυριστεί ότι αν αλλοιωθεί το οτιδήποτε τότε μάλλον ο χρήστης θα το καταλάβει, ωστόσο η απλή πρόσβαση και αποκάλυψη των δεδομένων μας σε τρίτους δεν θα γίνει αντιληπτή.

Επιπλέον, δυστυχώς όλα τα παραπάνω ούτε καν περιορίζονται στον προμηθευτή του νέφους αφού όπως περιγράψαμε νωρίτερα πολλοί μπορεί να εγκαθιστούν υπηρεσίες τους στο νέφος με τους πόρους που προσφέρει το νέφος. Με το να χρησιμοποιούμε τέτοιες υπηρεσίες θέτουμε τον εαυτό μας και πιθανά σημαντικές πληροφορίες σε ίσο ή μεγαλύτερο κίνδυνο. Αυτό ισχύει επειδή σε τέτοιες περιπτώσεις ο παροχέας της υπηρεσίας βρίσκεται σε παρόμοια θέση με το υπολογιστικό νέφος όσον αφορά πρόσβαση στα δεδομένα των χρηστών, αφού τα δεδομένα βρίσκονται σε «δικούς» του πόρους οι οποίοι με τη σειρά τους ανήκουν στο νέφος. Συνεπώς, είδαμε ότι το ρίσκο όσον αφορά την ασφάλεια και εμπιστευτικότητα των δεδομένων των χρηστών αυξάνεται με το παραμικρό, πράγμα που προκαλεί και συνεχόμενη ασχολία επί του θέματος.

3.3 Υπάρχουσες Τεχνικές

Η κλίμακα του ρίσκου της ασφάλειάς μας απέναντι στο υπολογιστικό νέφος όπως περιεγράφηκε προηγουμένως είναι ανάλογη προς τις απαιτήσεις που έχουμε από το υπολογιστικό νέφος. Αυτές περιλαμβάνουν οποιαδήποτε ενέργεια το υπολογιστικό νέφος πρέπει να κάνει στα δεδομένα/πληροφορίες μας έτσι ώστε να καλύπτει τις απαιτούμενες ανάγκες που έχουμε ως χρήστες συγκεκριμένης υπηρεσίας. Επιπρόσθετα, μπορούμε να κατηγοριοποιήσουμε τις ανάγκες αυτές.

Η απλούστερη ανάγκη θα ήταν η απλή αποθήκευση δεδομένων στο νέφος, χωρίς περιορισμό για το τι είναι τα δεδομένα, με απλή προσωπική χρήση. Σε αυτή την περίπτωση το νέφος δεν είναι υποχρεωμένο ούτε υπάρχει η ανάγκη να κάνει οποιοδήποτε υπολογισμό στα δεδομένα των χρηστών, όπως λέει και το [3]. Κατά συνέπεια, ο χρήστης είναι σε θέση να κρυπτογραφήσει οτιδήποτε θέλει να αποθηκεύσει στο υπολογιστικό νέφος και να κρατήσει ο ίδιος τα κλειδιά, αναλαμβάνοντας πλήρη ευθύνη της ασφάλειάς

τους. Με το που θέλει να προβεί σε οποιαδήποτε ενέργεια στα δεδομένα του, αποκρυπτογραφεί ό,τι είναι απαραίτητο, πράττει αυτά που επιθυμεί και κρυπτογραφεί ξανά πριν να αποσταλεί οτιδήποτε πίσω στο νέφος.

Ωστόσο, στο παραπάνω σενάριο δεν έχουμε πρόσβαση σε υπολογιστική δύναμη που μπορεί να μας προσφέρει το νέφος, ούτε το εκμεταλλευόμαστε για να πράξει κάποιους υπολογισμούς για καλύτερη λειτουργία κάποιου συστήματος. Αυτό οφείλεται στο ότι οι πράξεις/υπολογισμοί στους οποίους αναφερόμαστε εξαρτούνται από τα καθαρά δεδομένα και δεν μπορούν να εφαρμοστούν σε κρυπτογραφημένα. Αυτό είναι εντελώς αναμενόμενο αφού ο σκοπός των κρυπτογραφημένων είναι ουσιαστικά να μην βγάζουν νόημα παρά μόνο εάν έχουμε στην κατοχή μας το ορθό κλειδί. Λαμβάνοντας υπόψη και το γεγονός ότι οι περισσότερες υπηρεσίες στο υπολογιστικό νέφος έχουν κάποια ανάγκη που βρίσκεται στην υπολογιστική κατηγορία του νέφους, έναντι στην απλά αποθηκευτική, το πιο πάνω σενάριο δεν θα χαρακτηριζόταν και το πιο συνηθισμένο. Για να γίνει πιο ξεκάθαρο αυτό το θέμα θα αναφέρουμε μερικά παραδείγματα τα οποία είναι σε αυτήν την κατηγορία και αυξάνουν την πρόκληση για την αντιμετώπιση του προβλήματος, τα οποία καταγράφονται στο [6]:

- Έλεγχος για ανεπιθύμητα ηλεκτρονικά μηνύματα (spam filtering) σε μια υπηρεσία ηλεκτρονικού ταχυδρομείου, ή οποιοδήποτε άλλο είδος φιλτραρίσματος γίνεται σε τέτοιες υπηρεσίες.
- Σύγκριση αρχείων, όπως για παράδειγμα φωτογραφιών για κατηγοριοποίηση ανά θέμα.
- Για θέματα οικονομικά, μπορεί το νέφος να πρέπει να υπολογίσει κάποια στατιστικά ή διαφοροποιήσεις σε κάποιους τομείς για παρουσίαση στο κοινό.

Σε αρχικό και συγκεκριμένα στο παρόν στάδιο, το θέμα αυτό αντιμετωπίζεται μέσω πολιτικών. Με τον όρο πολιτικές εννοούμε κάποιο είδος εγγύησης που παραχωρεί το υπολογιστικό νέφος στου πελάτες/χρήστες, όπως για παράδειγμα με κάποιο συμβόλαιο εγγυάται την εμπιστευτικότητα των δεδομένων. Ακόμα, γίνονται απόπειρες έτσι ώστε όσο το δυνατό πιο μειωμένος αριθμός προσωπικού να έχει πρόσβαση σε ένα συγκεκριμένο σύνολο δεδομένων ανά πάσα στιγμή. Επιπλέον μέρος αυτού του συμβολαίου μπορεί να περιλαμβάνει και απαγόρευση όσον αφορά έκθεση πληροφοριών

σε τρίτους. Σε γενικές γραμμές, λαμβάνονται κάποια μέτρα αλλά το πρόβλημα δεν απαλείφεται ούτε εξαφανίζει το ενδεχόμενο για παραβίαση της ιδιωτικότητας του χρήστη χωρίς την αντίληψή του.

Επομένως, προχωρώντας θα συζητήσουμε, θα αναλύσουμε και θα συγκρίνουμε διάφορες τεχνικές που χρησιμοποιούνται για αντιμετώπιση του παραπάνω θέματος, έστω και σε κάποιο βαθμό. Ένα σύνολο αυτών δημιουργήθηκε και αποκλειστικά για αυτό τον σκοπό· αν όχι για το υπολογιστικό νέφος συγκεκριμένα, τότε σίγουρα για αντίστοιχο περιβάλλον υπολογιστικής ωφέλειας.

Για να είναι ξεκάθαρο, οι αλγόριθμοι κρυπτογραφίας χρησιμοποιούν κλειδί για να κρυπτογραφήσουν τα δεδομένα. Επίσης, είναι μέσα στις προδιαγραφές/απαιτήσεις τους το ότι η οποιαδήποτε αλλαγή στο καθαρό κείμενο (plaintext) πριν την κρυπτογράφηση, παράγει εντελώς διαφορετικό κρυπτογραφημένο κείμενο (ciphertext), χωρίς να εγγυάται τίποτα για αυτή την αλλαγή ούτε να είναι «εφικτό» να την υπολογίσουμε. Ταυτόχρονα από την άλλη πλευρά, οποιαδήποτε αλλαγή στο κρυπτογραφημένο κείμενο θα επηρεάσει στα σίγουρα και το καθαρό κείμενο, πάλι με μη-υπολογίσιμο και μη-προβλέψιμο τρόπο· δεδομένου ότι έχουμε το κλειδί για να αποκτήσουμε το καθαρό κείμενο.

3.3.1 Ομομορφικά Κρυπτοσυστήματα (Homomorphic Encryption)

Ως πρώτο παράδειγμα για λύση θα αναφερθούμε στην Ομομορφική Κρυπτογράφηση. Αυτό το είδος κρυπτογράφησης προσεγγίζει το πρόβλημα που αντιμετωπίζουμε με τους κοινούς και συνηθισμένους αλγόριθμους κρυπτογράφησης. Έχει ένα μοναδικό χαρακτηριστικό το οποίο ακούγεται μαγικό: υπάρχουν πράξεις που μπορούν να γίνουν πάνω σε κρυπτογραφημένο κείμενο το οποίο όταν αποκρυπτογραφηθεί μας δίνει το ισοδύναμο που θα είχαμε αν κάναμε τις αντίστοιχες πράξεις στο καθαρό κείμενο. Υπάρχουν διάφορα επίπεδα Ομομορφικής Κρυπτογράφησης, με πολλά να υποστηρίζουν ένα σύνολο ενεργειών (πρόσθεση και πολλαπλασιασμό κειμένων). Ωστόσο, η Πλήρως Ομομορφική Κρυπτογράφηση (Fully Homomorphic Encryption) φτάνει στο σημείο να μας επιτρέπει να έχουμε αντιστοιχίες

ενεργειών για αυθαίρετες πράξεις πάνω στα καθαρά ή κρυπτογραφημένα κείμενα. Επομένως, για κάθε ενέργεια που θα θέλαμε να κάνει το νέφος εκ μέρους μας, θα μπορούσαμε να κτίσουμε ένα «πρόγραμμα» (κάποια ακολουθία εντολών) για να το πράττει αυτό μέσω του κρυπτογραφημένου κειμένου. Δίνοντας πρόσβαση του νέφους σε αυτά τα «προγράμματα» αποτελεί σε κάποιο βαθμό λύση. Για περισσότερες πληροφορίες και συγκεκριμένες τεχνικές πάνω στην Ομομορφική Κρυπτογραφία γίνεται αναφορά στο [8].

Αυτή η τεχνική εκ πρώτης όψεως ακούγεται σαν να είναι όντως η λύση για το πρόβλημα, αφού μπορεί να δώσει την δυνατότητα στο νέφος να ενεργήσει πάνω στα δεδομένα του χρήστη χωρίς να παραβιάσει την εμπιστευτικότητά της κρυπτογράφησης. Όμως, έχει κάποια σημαντικά μειονεκτήματα που δυστυχώς την καθιστούν να μην είναι η τέλεια λύση όπως υπόσχεται έμμεσα, παρόλο που σίγουρα υπάρχουν πεδία στα οποία μπορεί να εφαρμοστεί επιτυχώς.

Τα προβλήματα που αναφερόμαστε είναι κατ' αρχήν, το γεγονός ότι παρόλο που με αυτή την τεχνική το νέφος θα είναι σε θέση να ενεργήσει παίρνοντας αποφάσεις πάνω στα δεδομένα του χρήστη, αυτό δεν υπερπηδά το εμπόδιο της ανάγκης που έχουν κάποιες ενέργειες (email spam filtering) να γίνουν πάνω στο καθαρό κείμενο. Δηλαδή, σε τέτοιες περιπτώσεις αυτή η τεχνική μπορεί να μας φτάσει ως ένα σημείο, δηλαδή να μας δώσει το ορθό αποτέλεσμα κρυπτογραφημένο, άρα να μην μπορεί να πάρει κάποια απόφαση χωρίς τον χρήστη (κάτοχο του κλειδιού). Για να προχωρήσει περισσότερο όμως, θα χρειάζεται η μεσολάβηση του χρήστη για πιθανά ενδιάμεσες αποκρυπτογραφήσεις, πράγμα επικίνδυνο αφού κατά λάθος μπορεί το νέφος να αποκτήσει πρόσβαση σε δεδομένα που δεν θα έπρεπε.

Επιπρόσθετα, ένα επίσης σημαντικό ελάττωμα είναι το μέγεθος των κλειδιών που χρησιμοποιούνται σε αυτό το κρυπτοσύστημα όπως επίσης και τα μεγέθη των κρυπτογραφημένων κειμένων. Αυτό καθιστά τις τυχόν εφαρμοζόμενες ενέργειες μη-πρακτικές. Επίσης, το σύστημα είναι δύσκολο στην υλοποίηση και υπάρχουν πολλαπλοί τρόποι να υλοποιηθεί. Οι ρίζες του εν λόγω συστήματος προέρχονται από τον κρυπτοσύστημα RSA, το οποίο ενεργεί με πολύ μεγάλα κλειδιά και μεγάλα κρυπτογραφημένα κείμενα εξ' ορισμού. Το γεγονός αυτό έχει ως αποτέλεσμα οι

κρυπτογραφικές ενέργειες αυτού του αλγόριθμου να είναι αργές. Στην περίπτωση του ίδιου του RSA όμως, δεν μας ενοχλεί αυτό αφού ο τρόπος χειρισμού του και η θέση του στο πεδίο της ασφάλειας εκμηδενίζουν αυτό το ελάττωμα. Ως συμπέρασμα, οι τυχόν απαιτήσεις/προδιαγραφές κάποιων εφαρμογών προς το υπολογιστικό νέφος για μεγάλες μάζες δεδομένων καθιστούν αυτή την τεχνική μη-πρακτική λύση. Αυτό ισχύει τουλάχιστον για προβλήματα που θα αναμέναμε ότι μπορεί να επιλύσει με τις υποσχέσεις που δίνονται από τα χαρακτηριστικά της.

Σε συνδυασμό αυτών με το προηγούμενο ελάττωμα, συμπεραίνουμε ότι αυτή η τεχνική δεν λύνει εντελώς το πρόβλημά μας, παρόλο που υπάρχουν πολλές περιπτώσεις που πιθανώς να είναι η ιδανική. Γίνονται προσπάθειες για αντιμετώπιση αυτών των ελαττωμάτων, αλλά κρίνοντας από την μέχρι τώρα εξέλιξη δεν δίνονται πολλές υποσχέσεις για το πόσο θα μπορεί να χρησιμοποιηθεί αυτή η τεχνική στο υπολογιστικό νέφος, ειδικά όταν θέλουμε ισχυρή ασφάλεια και πολλούς υπολογισμούς. Συνεπώς, θα προσπαθήσουμε να εξετάσουμε και άλλους τρόπους προσέγγισης.

3.3.2 Μετάφραση Κλειδιών στον Πλοηγητή (Key Translation in the Browser)

Στη συνέχεια, θα συζητήσουμε μια άλλη τεχνική που ονομάζεται Μετάφραση Κλειδιών στον Πλοηγητή, και από το όνομα κάποιος μπορεί να υποθέσει περί τίνος πρόκειται, με πληροφορίες από το [6]. Αυτή η προσέγγιση περιλαμβάνει την κλασσική κρυπτογράφηση των δεδομένων πριν την αποστολή τους στο υπολογιστικό νέφος από τον χρήστη, όπως επίσης και «μετάφραση κλειδιών» η οποία θα εξηγηθεί αμέσως μετά. Η κρυπτογράφηση ακόμα μπορεί να γίνεται και με διαφορετικά κλειδιά για κάθε μέρος των δεδομένων, ανάλογα με την υπηρεσία, το σύστημα που υλοποιήσαμε και τις απαιτήσεις του. Επιπλέον, θα μπορούσε να δίνεται στο νέφος πρόσβαση σε επιλεγμένα σύνολα δεδομένων τα οποία θα χρησιμοποιούνταν για τυχόν επιθυμητούς υπολογισμούς ή για την ορθή λειτουργία του συστήματος.

Η μετάφραση κλειδιών αναφέρεται στην ιδεολογία ότι οι χρήστες μπορούν να κρυπτογραφούν κάποια κλειδιά με τρόπο τέτοιο ώστε να απευθύνονται σε άλλους συγκεκριμένους χρήστες ή σύνολα από χρήστες, πάντοτε κρυμμένα από το νέφος. Ή

ακόμα, σε μια υπηρεσία/πρωτόκολλο μπορεί να υπάρχουν συγκεκριμένοι εξειδικευμένοι χρήστες (Trusted Third Party) που να είναι χρεωμένοι τις επιπλέον ευθύνες της μετάφρασης κλειδιών και να θεωρούνται η καρδία του συστήματος, αντί η μετάφραση να γίνεται κατανεμημένα. Φυσικά, αυτές οι προσεγγίσεις και άλλες που δεν αναφέραμε εξαρτούνται πάντα από τη φύση του συστήματος που αναλάβαμε να ενισχύσουμε και τις απαιτήσεις του. Ωστόσο, τα παραδείγματα αυτά συνείσφεραν στην κατανόηση του ορισμού «μετάφραση κλειδιών».

Αυτή η τεχνική δεν επιτρέπει στο υπολογιστικό νέφος να ενεργεί αυθαίρετα πάνω στα δεδομένα του χρήστη όπως η Πλήρως Ομομορφική Κρυπτογράφηση, αλλά αν χρησιμοποιηθεί ορθά παρέχει ασφαλείς λύσεις σε υπηρεσίες που είναι εφικτό να εφαρμοστεί. Ο αριθμός αυτών των υπηρεσιών όμως είναι περιορισμένος και γενικώς χαρακτηρίζονται ως «store-and-forward», αν και γίνεται συνεχής έρευνα για διεύρυνση αυτού του αριθμού. Επομένως, δεν παρέχει τόσες υποσχέσεις σε χρήση όσες θα μπορούσαμε να σκεφτούμε για την Πλήρως Ομομορφική Κρυπτογράφηση, όμως σε περιπτώσεις που μπορεί να χρησιμοποιηθεί δίνει πολύ ικανοποιητικά αποτελέσματα.

Παρόλα αυτά όμως υπάρχουν και σε αυτή την προσέγγιση κάποια θέματα που αξίζει να αναφέρουμε, εκτός του περιορισμένου αριθμού υπηρεσιών που εφαρμόζεται επιτυχώς. Συνήθως σε πρωτόκολλα που υλοποιούν τέτοιους μηχανισμούς, ανταλλάσσονται περισσότερα δεδομένα μεταξύ των αναμειγμένων κόμβων για να επιτευχθεί κάτι από όσα θα ήταν εάν δεν μας ενδιέφερε η ασφάλεια. Αυτό είναι και ένα «trade-off» που πληρώνουμε για να κερδίσουμε κάτι άλλο, από τα οποία είναι γεμάτη η επιστήμη της Πληροφορικής. Με σωστή υλοποίηση μπορούμε να αποτρέψουμε οποιοδήποτε πρόβλημα είναι δυνατόν να προκύψει από αυτό.

Τέλος, εφαρμογές αυτής της κατηγορίας εκμεταλλεύονται το υπολογιστικό νέφος για αποστολή δεδομένων και για επίβλεψη της σωστής ροής των πληροφοριών του συστήματος. Οι ενέργειες που πράττει το νέφος όμως είναι καθαρά ενέργειες δρομολόγησης από πιθανώς ένα χρήστη προς κάποιον άλλο, ή οτιδήποτε. Παρόλο που είναι σημαντικό και όντως γίνεται χρήση του νέφους, το νέφος δεν έχει πρόσβαση στα καθαρά δεδομένα στις περισσότερες των περιπτώσεων άρα δεν μπορεί να κάνει υπολογισμούς και πράξεις που βασίζονται στο καθαρό κείμενο.

Λαμβάνοντας υπόψη όλα τα πιο πάνω, καταλήγουμε ότι η προσέγγιση αυτή έχει τη δυνατότητα να φανεί χρήσιμη σε αυτές τις συγκεκριμένες εφαρμογές τις οποίες ικανοποιεί παρ' όλους τους περιορισμούς που αντιμετωπίζει. Στη συνέχεια θα εξετάσουμε μια αρκετά θεμελιώδη διαφορετική λύση, για σύγκριση με όσα είδαμε μέχρι τώρα.

3.3.3 Ασφάλεια Υλοποιημένη σε Υλικό (Hardware-Anchored Security)

[6] Αυτή η προσέγγιση δεν έχει να κάνει αποκλειστικά με τρόπο κρυπτογράφησης των δεδομένων, αλλά προχωρά ακόμα πιο χαμηλά για να αντιμετωπίσει το ζήτημά μας. Συγκεκριμένα φτάνει μέχρι και το υλικό (hardware) από την πλευρά του προμηθευτή του νέφους. Περιλαμβάνει πολύ εξειδικευμένο υλικό το οποίο προσφέρει μηχανισμούς και λειτουργίες προς την ασφάλεια του χρήστη, και παρέχει την δυνατότητα του χρήστη να το επιβεβαιώνει.

Το εξειδικευμένο υλικό που αναφέρθηκε πιο πάνω, επιτρέπει στο υπολογιστικό νέφος να αποκρυπτογραφεί τα δεδομένα του χρήστη, αλλά με εγγυήσεις και περιορισμούς ως προς την πρόσβαση. Οι εγγυήσεις επιτρέπουν στον χρήστη να ξέρει πότε έχει γίνει η οποιαδήποτε αλλαγή στα δεδομένα του, όπως και να υπάρχει τρόπος να βεβαιωθεί ότι όντως το σύστημα είναι ασφαλές. Επιπλέον, η αποκρυπτογράφηση που γίνετε είναι πλήρως ελεγμένη και περιορισμένη μόνο σε αυτά που πρέπει να γίνουν, με το τέλος των οποίων γίνεται ξανά αυτόματα κρυπτογράφηση. Κάποιος θα διερωτάτο το τι έχει να κάνει το υλικό με αυτή την περιγραφή, κάτι που θα εξηγήσουμε αμέσως μετά.

Η όλη ιδέα βασίζεται στο υλικό να παρέχει την δυνατότητα στο νέφος να αποθηκεύσει κλειδιά κρυπτογραφημένα και, το πιο σημαντικό, να τα συνδέσει με ένα συγκεκριμένο «πρόγραμμα» (κάποια ακολουθία εντολών), ένα για κάθε κλειδί. Όταν λέμε συνδέσει, εννοούμε ότι μόνο το συγκεκριμένο πρόγραμμα θα μπορεί να ανακτήσει το κλειδί, να το αποκρυπτογραφήσει, να αποκρυπτογραφήσει τα δεδομένα, να κάνει τις απαιτούμενες αλλαγές/υπολογισμούς, να τα κρυπτογραφήσει ξανά, και να κρυπτογραφήσει το κλειδί ξανά. Το διαφορετικό είναι ότι όλα αυτά γίνονται στην πλευρά

του νέφους, στην ουσία όλες οι ενέργειες που πιθανώς να έκανε ο χρήστης σε λύσεις όπως την προηγούμενη, γίνονται από το ίδιο το νέφος με θεωρητικά ασφαλές τρόπο.

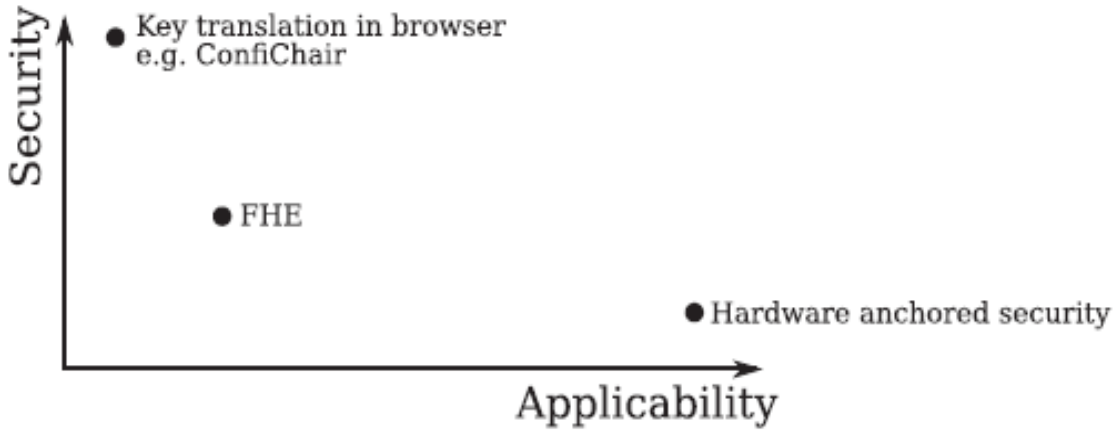
Το εν λόγω πρόγραμμα που έχει την πρόσβαση στα δεδομένα καθορίζεται από την αρχή μεταξύ του νέφους και του χρήστη, δηλαδή κατά κάποιο τρόπο δημιουργούν το συγκεκριμένο πρόγραμμα (ή προγράμματα) που χρειάζεται για την περίπτωσή τους. Στη συνέχεια, θα συνδέσουν το πρόγραμμα αυτό με το κλειδί το οποίο θα ανεβάσει ο χρήστης στο νέφος με ασφαλή τρόπο. Όταν τώρα ο χρήστης θέλει να χρησιμοποιήσει το νέφος για υπολογισμούς μέσω του προγράμματος που δημιούργησε, ανεβάζει τα δεδομένα που θέλει να παραχωρήσει κρυπτογραφημένα με το ίδιο κλειδί που έχει πρόσβαση το πρόγραμμα, για να γίνει η δουλειά που πρέπει. Ο τρόπος που συνδέεται το υλικό με αυτό το πρόγραμμα μας παρέχει την βεβαίωση ότι όντως είναι το πρόγραμμα αυτό που εκτελείται πάνω στα δεδομένα μας την δεδομένη στιγμή.

Όπως όλες οι άλλες προσεγγίσεις που περιγράψαμε, έτσι και αυτή πάσχει από κάποιες αδυναμίες. Η κυριότερη είναι ότι το πρόγραμμα που δημιουργείται για μια συγκεκριμένη ενέργεια στα δεδομένα, πολλές φορές είναι δύσκολο να ελεγχθεί η ασφάλειά του. Αυτό συμβαίνει επειδή για υπηρεσίες που δεν είναι απλές και περιέχουν ένα σεβαστό βαθμό πολυπλοκότητας το πρόγραμμα αυτό καταλήγει να είναι πολύπλοκο και πολύ μεγάλο σε μέγεθος. Επομένως ο χρήστης δεν έχει τρόπο να ελέγξει αν η ασφάλεια που υποτίθεται παρέχει το σύστημα όντως υφίσταται αφού έχει να κάνει με πολύπλοκο κώδικα. Το μόνο που ξέρει σίγουρα είναι ότι το συγκεκριμένο πρόγραμμα είναι αυτό που εκτελείται, όχι ότι σίγουρα είναι ασφαλές.

3.3.4 Σύγκριση

Σε αυτό το σημείο έχουμε συζητήσει και αναλύσει ξεχωριστά τρεις προσεγγίσεις στο θέμα της ασφάλειας στο υπολογιστικό νέφος. Λαμβάνοντας υπόψη τα θετικά και αρνητικά της κάθε προσέγγισης, όπως επίσης και τις εφαρμογές τους, καταλήγουμε στην πιο κάτω αναλογία ασφάλειας και εφαρμοσιμότητας για την καθεμία. Στο ένα άκρο έχουμε την Μετάφραση Κλειδιών στον Πλοηγητή αφού είναι σε θέση να εφαρμόσει τους πιο δυνατούς αλγόριθμους κρυπτογράφησης με αντάλλαγμα πολύ περιορισμένο σύνολο

εφαρμογών που μπορεί να χρησιμοποιηθεί. Από την άλλη, έχουμε την Ασφάλεια Υλοποιημένη στο Υλικό αφού με τις προδιαγραφές του ικανοποιεί πολύ μεγάλο μέρος εφαρμογών, αν όχι όλες, αλλά υστερεί στο να βεβαιώνει τον χρήστη για την ασφάλεια που προσφέρει. Τέλος, ενδιάμεσα βρίσκεται η Πλήρως Ομομορφική Κρυπτογράφηση που παραχωρεί πρόσβαση σε μέτρια επίπεδα εφαρμοσιμότητας και ασφάλειας, λόγω των χαρακτηριστικών που περιεγράφηκαν νωρίτερα.



Σχήμα 3.1: Οι τεχνικές οργανωμένες ανάλογα με ασφάλεια και εφαρμοσιμότητα. [6]

3.4 Σχετική Δουλειά

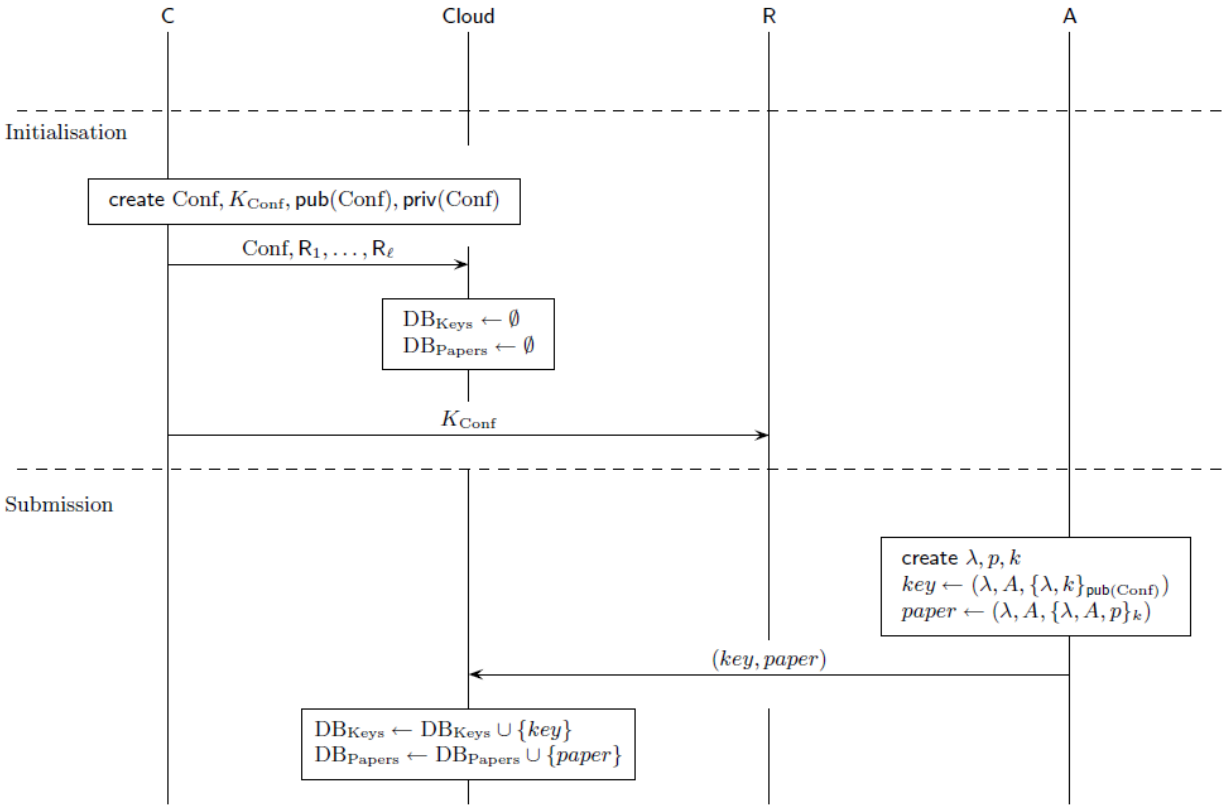
Πέραν από όλα όσα μελετήθηκαν και συζητήθηκαν μέχρι τώρα, η κυριότερη σχετική δουλειά που μελετήθηκε και στην οποία βασίστηκε σε μεγάλο βαθμό η συνέχεια αυτής της εργασίας είναι το πρωτόκολλο [1] ConfiChair. Το συγκεκριμένο πρωτόκολλο υλοποιεί ένα σύστημα διαχείρισης συνεδρίων (conference management system) με ασφάλεια από το υπολογιστικό νέφος. Το συγκεκριμένο παράδειγμα χρησιμοποιεί Μετάφραση Κλειδιών στον Πλοηγητή σε συνδυασμό με έξυπνες τεχνικές και πετυχαίνει το στόχο του σε πολύ αποδοτικά επίπεδα. Αποκρύπτει όλες τις σημαντικές πληροφορίες που αφορούν την λειτουργία του συστήματος από το νέφος. Ταυτόχρονα, το εκμεταλλεύεται για να ελέγχει την σωστή ροή δεδομένων προς και από το σύστημα. Στη συνέχεια θα περιγράψουμε τι είναι ένα σύστημα διαχείρισης συνεδρίων και τον τρόπο λειτουργίας του ConfiChair χρησιμοποιώντας διαγράμματα τα οποία παρείχαν οι

δημιουργοί του πρωτοκόλλου. Για πλήρη λεπτομερή περιγραφή της όλης ιδεολογίας, προεργασίας και λειτουργίας τους ConfiChair, ο αναγνώστης παραπέμπεται στο [1].

Ένα σύστημα διαχείρισης συνεδρίων, και συγκεκριμένα το μοντέλο που μελετά και ασχολείται το ConfiChair, περιλαμβάνει σε γενικές γραμμές τους χρήστες και το νέφος. Ο σκοπός αυτής της υπηρεσίας είναι η διαχείριση και αξιολόγηση των άρθρων που έχουν υποβληθεί σε κάποιο συνέδριο και η επιλογή του επιστημονικού προγράμματος. Σε κάθε συνέδριο εμπλέκονται τέσσερα είδη οντοτήτων, από τα οποία τα δύο είναι μοναδικά ανά συνέδριο. Αυτές οι οντότητες είναι οι εξής:

- **Το Νέφος:** Το νέφος είναι ουσιαστικά παρόν σε όλα τα συνέδρια, και δεν υπάρχει κάτι διαφορετικό όσον αφορά τον ρόλο του σε αυτό το σενάριο, ισχύουν όλα όσα είπαμε για το νέφος· η αποθήκευση δεδομένων, ο απαιτούμενος χειρισμός, η ροή τους κλπ.
- **Ο Διαχειριστής του Συνέδριου (Conference Chair):** Ο διαχειριστής ενός συνέδριου είναι μια μοναδική οντότητα σε κάθε συνέδριο και μπορεί να είναι ένας οποιοσδήποτε χρήστης. Οι ευθύνες του περιλαμβάνουν τις απαραίτητες ενέργειες για την αρχικοποίηση του συνέδριου, την επιλογή όλων των κριτών και υποσυνόλων τους για κάθε έγγραφο, και άλλες ενέργειες κατά τη διάρκεια της διεξαγωγής ενός συνέδριου με τον ρόλο του «Έμπιστου Τρίτου» (Trusted Third Party).
- **Οι Συγγραφείς (Authors):** Σε κάθε συνέδριο είναι δυνατό να υπάρχουν πολλοί συγγραφείς, αλλά πρακτικά χρειάζεται τουλάχιστον ένας για να υπάρχει νόημα στον όρο συνέδριο. Ο ρόλος ενός συγγραφέα είναι να καταθέσει ένα ή περισσότερα έγγραφά του στο συγκεκριμένο συνέδριο για να βαθμολογηθούν/κριθούν από τους κριτές που θα τα αναλάβουν. Επίσης, ο κάθε συγγραφέας μπορεί να είναι και κριτής άλλου/ων εγγράφου/ων.
- **Οι Κριτές (Reviewers):** Όπως και συγγραφείς, μπορούν να συμμετέχουν πολλοί κριτές σε κάθε συνέδριο, και είναι ουσιαστικά απαραίτητο να υπάρχουν περισσότεροι από έναν. Ο κάθε κριτής αναλαμβάνει να διαβάσει ένα ή περισσότερα έγγραφα, να τα κρίνει και να καταθέσει την κριτική του πίσω στον διαχειριστή. Δεν έχει πρόσβαση σε άλλα έγγραφα παρά μόνο εκείνα που του διαθέτει ο διαχειριστής.

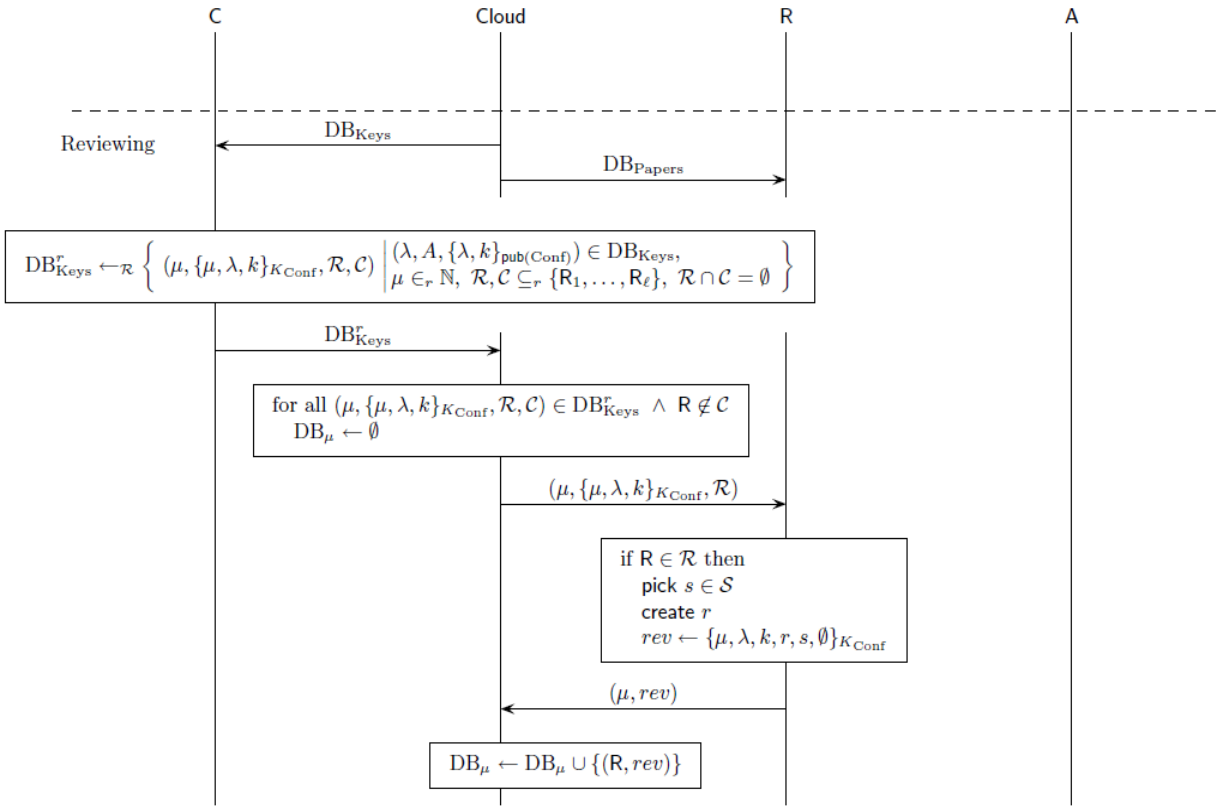
Για να μπουν όλα σε μια σειρά, κάθε συνέδριο ξεκινά μεταξύ του διαχειριστή και του νέφους έτσι ώστε να γίνουν οι απαραίτητες αρχικοποιήσεις. Σε αυτό το σημείο ορίζονται όλοι όσοι θα λάβουν μέρος στο συνέδριο (ουσιαστικά οι κριτές), από τους οποίους όποιοι θέλουν μπορούν να είναι και συγγραφείς και να καταθέσουν έγγραφα για κριτική. Στη συνέχεια ο διαχειριστής αντιστοιχεί κριτές με έγγραφα για να διεξαχθεί η κριτική, και στο τέλος επιστρέφει σε κάθε συγγραφές τις κριτικές για το έγγραφό/ά του.



Σχήμα 3.2: Αρχικοποίηση του συστήματος και κατάθεση εγγράφων. [1]

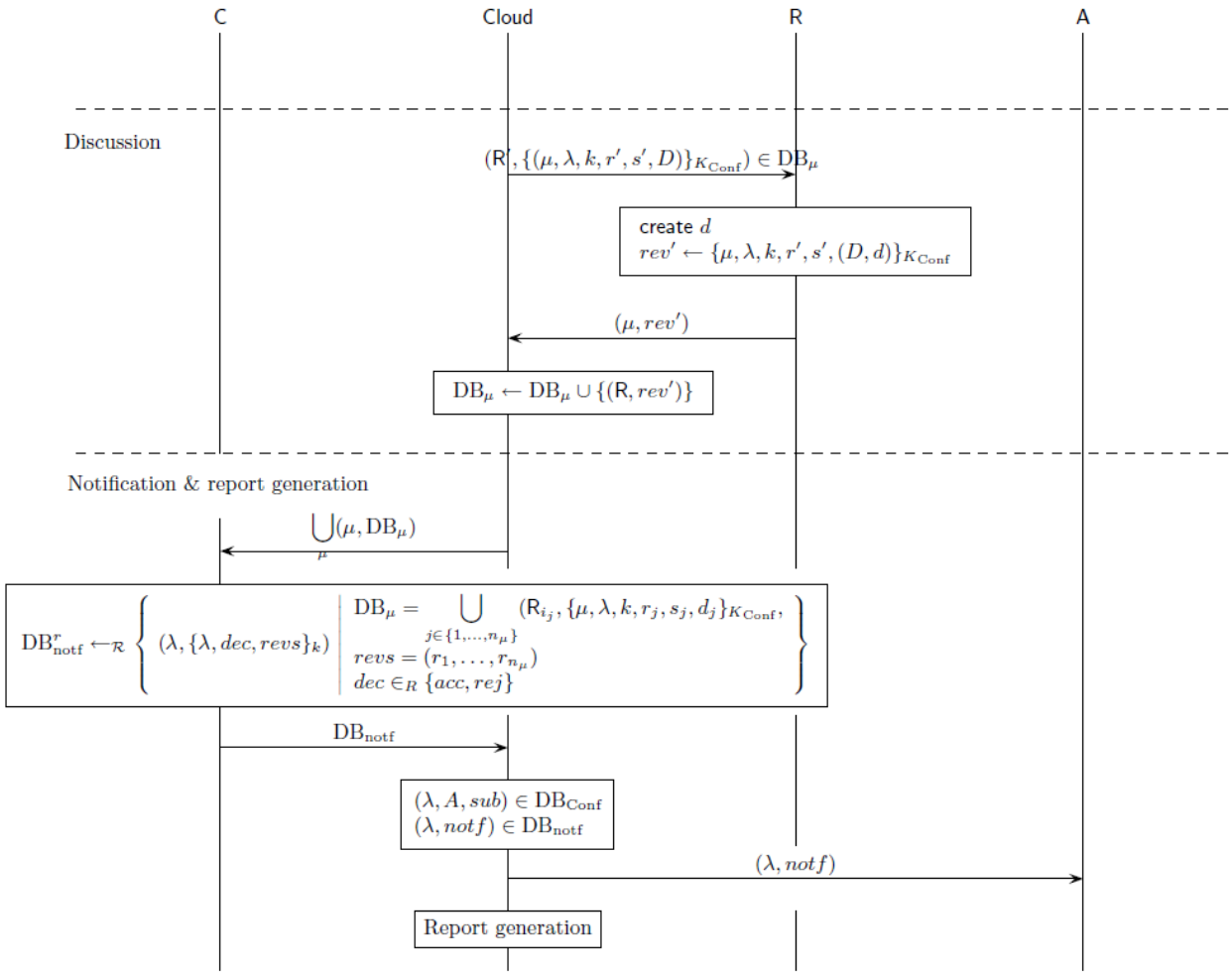
Αυτό το διάγραμμα παρουσιάζει την αρχικοποίηση του συστήματος, κατά την οποία ο διαχειριστής δημιουργεί το συνέδριο όπως επίσης και τρία κλειδιά (συμμετρικό, ιδιωτικό, δημόσιο) για το συγκεκριμένο συνέδριο. Στη συνέχεια λέει στο νέφος ποιοι θα είναι οι κριτές για αυτό` το συνέδριο, και στέλνει με ασφαλή τρόπο (π.χ. ηλεκτρονικό ταχυδρομείο) το συμμετρικό κλειδί (Kconf) στους κριτές. Σε αυτό το σημείο όποιος θέλει να καταθέσει ένα έγγραφό του για κριτική, κρυπτογραφεί το έγγραφο με ένα δικό του προσωπικό κλειδί, και αυτό το κλειδί με το Pub(Conf) και τα στέλνει όλα στο νέφος. Το έγγραφο και το κλειδί που στέλνει ένας συγγραφέας παίρνουν έναν δείκτη λ και

αποθηκεύονται αντίστοιχα στις βάσεις δεδομένων για έγγραφα και κλειδιά, τις οποίες δημιουργεί το νέφος.



Σχήμα 3.3: Κριτική και αξιολόγηση εγγράφων από τους κριτές. [1]

Σε αυτό το σημείο, ο διαχειριστής παίρνει όλα τα κλειδιά από την βάση δεδομένων, τα αποκρυπτογραφεί με το Priv(Conf) (αφού όλα είναι κρυπτογραφημένα με το Pub(Conf)), και κρυπτογραφεί πίσω με το Kconf. Τέλος, δίνει σε κάθε κλειδί ένα νέο επιπρόσθετο δείκτη μ (το λ ακόμα υπάρχει) πριν τα στείλει πίσω στη βάση δεδομένων στο νέφος. Με αυτό τον τρόπο όλοι οι κριτές έχουν πρόσβαση σε όλα τα κλειδιά (λόγω του Kconf) και με τον δείκτη μ το νέφος δεν μπορεί πλέον να ξέρει ποιο κλειδί αντιστοιχεί σε ποιο έγγραφο (δεν μπορεί να συνδέσει μ και λ). Ακολούθως, ο διαχειριστής λέει στο νέφος ποιοι κριτές αφορούν κάθε κλειδί έτσι ώστε το νέφος να είναι σε θέση να ελέγξει ποιους θα επιτρέψει να ανακτήσουν κάθε κλειδί. Αφού οι κριτές ανακτήσουν τα κλειδιά που δικαιούνται, ανακτούν και όλα τα έγγραφα. Αποκρυπτογραφούν τα κλειδιά, βλέπουν από τον δείκτη λ σε ποια έγγραφα έχουν πρόσβαση και τα αποκρυπτογραφούν για να τα διαβάσουν. Αφού τελειώσουν, γράφουν την κριτική τους και την ανεβάζουν πάνω πίσω στο νέφος κρυπτογραφημένη με Kconf και με επικολλημένο τον δείκτη μ.



Σχήμα 3.4: Συζήτηση, τελική απόφαση, και ειδοποίηση συγγραφέα. [1]

Εδώ παρουσιάζεται το τι γίνεται αφού ολοκληρωθούν οι κριτικές από τους κριτές. Αρχικά προχωρούν σε ένα άλλο στάδιο για συζήτηση, στο οποίο έχουν την δυνατότητα να δουν κριτικές από άλλους κριτές, πάντα για τα έγγραφα που έχουν δικαίωμα, και είναι σε θέση να προσθέσουν σχόλια αν επιθυμούν. Ο τρόπος που στέλνονται τα σχόλια στο υπολογιστικό είναι πανομοιότυπος με τις κριτικές. Τέλος, ο διαχειριστής μαζεύει για κάθε έγγραφο όλες τις κριτικές και τα τυχόν σχόλια και τα ομαδοποιεί κρυπτογραφημένα με το κλειδί που έστειλε ο συγγραφέας στην αρχή όταν κατέθεσε το έγγραφο. Επιπλέον, εκτός από αυτή την κρυπτογράφηση, επαναφέρεται ο δείκτης λ και τοποθετείται σε αυτό το σύνολο για να μπορεί το νέφος να δώσει σε κάθε συγγραφέα αυτά που του αντιστοιχούν.

Περιληπτικά αυτή είναι η λειτουργία του ConfiChair, και σε καμία περίπτωση δεν θεωρείται επίσημη περιγραφή του πρωτοκόλλου. Για την πλήρη λειτουργική

περιγραφή, ή αν κάτι δεν ήταν ξεκάθαρο, εξηγούνται τα πάντα αναλυτικά στο [1]. Ωστόσο, για κατανόηση συγκεκριμένων ορισμών υποτίθεται γνώση κάποιων ορολογιών του πεδίου Ασφάλειας Πληροφοριών.

3.5 Προσέγγιση Επιλογής

Μετά από μελέτη όλων των πιο πάνω και κατανόηση πολλών νέων εννοιών, κατέληξα που θέλω να επικεντρωθώ. Σε αυτή την εργασία επέλεξα να μελετήσω συνθήκες κοινωνικής δικτύωσης και να υλοποιήσω ένα σύστημα που να προσφέρει ασφάλεια από το υπολογιστικό νέφος. Σε αυτό το σημείο θα μιλήσω πολύ περιληπτικά για το σύστημα, αφού θα εξηγηθεί αναλυτικά αργότερα.

Για το σύστημα αυτό, έχω εξετάσει τις προσεγγίσεις για ασφάλεια από το νέφος που περιεγράφηκαν προηγουμένως και διάλεξα να ακολουθήσω την φιλοσοφία της Μετάφρασης Κλειδιών στον Πλοηγητή. Υπάρχουν λόγοι που έκανα αυτή την επιλογή αντί οποιασδήποτε άλλης προσέγγισης, με κυριότερο το ότι θεωρείται η πιο ασφαλής. Επιπλέον, για τις λειτουργίες που είχα σκοπό να υλοποιήσω και τις ανάγκες ενός κοινωνικού δικτύου η προσέγγιση αυτή ήταν η ιδανική, αφού οι απαιτήσεις του συστήματος θα είναι αποθηκευτικές και «store-and-forward». Επίσης, όσο πιο πολύ δουλέψουμε με αυτή την προσέγγιση, τόσο πιο μεγάλη πιθανότητα υπάρχει να επεκταθεί αλλά και να εμφανιστούν τρόποι για βελτίωση ελαττωμάτων που συναντούμε τώρα.

Ένα πολύ καλό παράδειγμα παρόμοια προσέγγισης είναι το ConfiChair. Παρόλο που το σύστημα που υλοποιεί είναι ριζικά διαφορετικό από το δικό μου, ο απώτερος μας στόχος είναι κοινός: η ασφάλεια του συστήματος με το οποίο ασχολούμαστε. Όσον αφορά τις συγκεκριμένες προδιαγραφές του ConfiChair, υπάρχουν κάποια χαρακτηριστικά τους που έλαβα υπόψη και προσπάθησα να υιοθετήσω και να προσαρμόσω στο δικό μου σύστημα: η απόκρυψη των δεδομένων των χρηστών, η απόκρυψη των σχέσεων μεταξύ των χρηστών, και η αποθήκευση κλειδιών στο νέφος. Ωστόσο, υπάρχει μία αρκετά σημαντική διαφορά μεταξύ των δύο πρωτοκόλλων, η οποία και επιφέρει αυτήν την ανάγκη για προσαρμογή των διάφορων χαρακτηριστικών. Η διαφορά αυτή είναι το γεγονός ότι το ConfiChair ασχολείται με ομάδες χρηστών οι οποίοι

πετυχαίνουν την ασφάλεια έχοντας έναν διαχειριστή. Στο δικό μας σύστημα η σχέσεις μεταξύ χρηστών είναι ένα-προς-ένα.

Δουλεύοντας με τεχνικές που βασίζονται σε αυτή την φιλοσοφία προσφέρει εξάσκηση και εξοικείωση με σημαντικούς αλγόριθμους και άλλες τεχνικές οι οποίες βρίσκονται παντού σήμερα σε θέματα ασφάλειας. Εκτός αυτού, εγώ προσπάθησα να πειραματιστώ όσο το δυνατό περισσότερο σε αυτήν εργασία, με στόχο να κερδίσω όσα περισσότερα μπορώ από πολλές όψεις. Οι άλλες προσεγγίσεις που συζητήσαμε είναι και δύσκολες στην υλοποίηση (ή αδύνατες στην περίπτωση με το ειδικό υλικό), ειδικά για τέτοιο σκοπό, και δεν είναι οι ιδανικές σε αυτή την περίπτωση.

Κεφάλαιο 4

Τεχνικές και Εργαλεία

Σε αυτό το κεφάλαιο θα αναφερθούμε και θα εξηγήσουμε τα εργαλεία που θα χρησιμοποιηθούν για την επιτυχή υλοποίηση της ιδέας που θα αναπτύξουμε. Είναι απαραίτητο να αναλυθούν σε αρκετό βάθος για να μπορούμε να αναφερόμαστε στις έννοιες όταν θα περιγράφουμε το σύστημα και να γίνονται πλήρως κατανοητές. Αυτά τα εργαλεία είναι κυρίως αλγόριθμοι και τεχνικές του πεδίου Ασφάλειας Πληροφοριών της Πληροφορικής, με τους περισσότερους να προέρχονται από την πολύ εκτενή κατηγορία της Κρυπτογραφίας. Θα ξεκινήσουμε μιλώντας γενικά για την έννοια της Κρυπτογραφίας πριν προχωρήσουμε σε συγκεκριμένες υποκατηγορίες της από τις οποίες θα χρησιμοποιήσουμε συγκεκριμένες τεχνικές και αλγορίθμους. Όλες οι τεχνικές και αλγόριθμοι θα αναλυθούν ξεχωριστά, παρέχοντας για όλα πλήρη περιγραφή, πλεονεκτήματα, μειονεκτήματα και σενάρια χρήσης. Δεν θα ασχοληθούμε με τον τρόπο υλοποίησης των παραπάνω επειδή ανήκει σε μια άλλη πολύ πιο μεγάλη συζήτηση, η οποία δεν είναι καν σχετική εδώ αφού όλα παρέχονται υλοποιημένα από βιβλιοθήκες.

Στη συνέχεια, θα ακολουθήσει μία σύντομη συζήτηση γύρω από το λειτουργικό σύστημα Android, το οποίο και θα χρησιμοποιηθεί για το σύστημα που έπεται. Στη συζήτηση αυτή θα περιλαμβάνονται πολύ βασικές έννοιες που πιθανόν να είναι απαραίτητες για την περιγραφή του συστήματος.

4.1 Κρυπτογραφία (Cryptography)

Η Κρυπτογραφία είναι το επιστημονικό πεδίο που ασχολείται και μελετά τεχνικές για προστασία πληροφοριών από τρίτους ή γενικότερα από άτομα προς τα οποία δεν προορίζονταν. Οι πληροφορίες που ενδιαφερόμαστε περισσότερο να προστατεύσουμε προέρχονται από κάποιου είδους επικοινωνία η οποία υπάρχει ενδεχόμενο να παραβιαστεί. Με τον όρο τεχνικές, αναφερόμαστε σε διάφορους αλγόριθμους κρυπτογράφησης/αποκρυπτογράφησης ή άλλου σκοπού, όπως επίσης και διάφορα πρωτόκολλα για επιτυχή και σωστή χρήση αυτών των αλγορίθμων για να πετύχουμε την ασφάλεια που επιθυμούμε εκεί που επιθυμούμε. Πίσω από αυτούς τους αλγόριθμους βρίσκονται ένα σύνολο επιστημών με κυριότερες από αυτές τα Μαθηματικά και την Πληροφορική. Η Κρυπτογραφία βρίσκει εφαρμογές σε διάφορους τομείς και υπηρεσίες της σημερινής κοινωνίας, αφού με την τεχνολογική εξέλιξη που απολαμβάνουμε ταυτόχρονα εξελίσσονται και οι τυχόν απειλές από τις οποίες οφείλουμε να προσέχουμε.

Ο ορισμός κρυπτογράφηση αναφέρεται στη διαδικασία κωδικοποίησης δεδομένων έτσι ώστε να μην είναι κατανοητά από κανέναν παρά μόνο εξουσιοδοτημένα άτομα. Η κωδικοποίηση αυτή βασίζεται πάνω σε ένα «κλειδί» το οποίο είναι μια σειρά μεγάλου (ανάλογα του αλγορίθμου) μεγέθους από bytes, έτσι ώστε να είναι όσο το δυνατό πιο δύσκολο να υπολογιστεί από κακόβουλα άτομα (θα εξηγηθεί αργότερα). Το/α κλειδί/ιά πρέπει να είναι γνωστά από τα ορθά άτομα, και εδώ βασίζονται όλα. Αφού κρυπτογραφηθούν κάποια δεδομένα, μόνο με κατοχή του κλειδιού (όχι απαραίτητα του ίδιου) μπορούν να ανακτηθούν στην καθαρή τους μορφή, και η διαδικασία αυτή ονομάζεται αποκρυπτογράφηση. Αναφερόμαστε στα κρυπτογραφημένα δεδομένα ως κρυπτογραφημένο κείμενο (ciphertext) και στα αποκρυπτογραφημένα δεδομένα ως καθαρό κείμενο (plaintext).

Για να θεωρείται ένας αλγόριθμος κρυπτογράφησης ασφαλής, βάσει του Claude Shannon πρέπει να πληροί τις αρχές της σύγχυσης και διάχυσης (confusion and diffusion). Η σύγχυση αναφέρεται στο ότι κάθε χαρακτήρας του κρυπτογραφημένου κειμένου πρέπει να βασίζεται σε πολλαπλά μέρη του κλειδιού, δηλαδή η σχέση κρυπτογραφημένου κειμένου και κλειδιού να είναι όσο πολύπλοκη γίνεται. Η διάχυση τώρα, αναφέρεται στη σχέση κρυπτογραφημένου κειμένου και καθαρού κειμένου, τα οποία δεν πρέπει να προσφέρουν καμιά εγγύηση το ένα για το άλλο. Αν για παράδειγμα αλλάξει κάτι μικρό στο καθαρό κείμενο, τότε πολλά πρέπει να αλλάξουν στο κρυπτογραφημένο κείμενο, και το αντίστροφο.

Υπάρχουν πολλοί αλγόριθμοι κρυπτογράφησης/αποκρυπτογράφησης ή άλλου σκοπού και ο καθένας έχει τα δικά του χαρακτηριστικά. Για να πετύχουμε ένα ολοκληρωμένο και ασφαλές σύστημα χρησιμοποιούμε διάφορα είδη αλγορίθμων/πρωτοκόλλων οι οποίοι αλληλοσυμπληρώνονται· δηλαδή ο καθένας αντιμετωπίζει πιθανώς τα ελαττώματα του άλλου, τα οποία θα αναλυθούν στη συνέχεια. Σε αυτό το κεφάλαιο θα συζητήσουμε μόνο τους αλγορίθμους που θα χρησιμοποιήσουμε εμείς στο σύστημά μας, αφού υπάρχει μεγάλη πλειάδα αλγορίθμων στο σύνολο, και δεν μας ενδιαφέρουν όλοι.

Τέλος, το πεδίο της Κρυπτογραφίας δε θα υπήρχε χωρίς τα κακόβουλα άτομα και τις διάφορες απειλές. Πέραν της γενικής απειλής του υπολογιστικού νέφους, δεν μιλήσαμε για άλλες συγκεκριμένες απειλές, αλλά ούτε για το πώς ακριβώς το νέφος μπορεί να προσπαθήσει να «σπάσει» τα μέτρα ασφαλείας που θα εγκαταστήσουμε. Θα αναφερόμαστε στις απειλές που ασχολείται και αντιμετωπίζει το σύστημά μας κατά το σημείο που είναι σχετικό μαζί τους, έτσι ώστε να γίνονται κατανοητές εκεί και όταν πρέπει. Αυτά τα σημεία μπορεί να είναι οι περιγραφές των αλγορίθμων που ακολουθούν, ή η περιγραφή του συστήματος που θα γίνει στο επόμενο κεφάλαιο. Αρχικά όμως όσον αφορά το «σπάσιμο» κλειδιών, θα αναφέρουμε τον πιο απλό και ευθύ τρόπο υπολογισμού του κλειδιού που ψάχνουμε, την επίθεση ωμής βίας (brute-force attack). Αυτή η επίθεση περιλαμβάνει δημιουργία και δοκιμή όλων των πιθανών εισόδων, κλειδιών σε αυτή την περίπτωση, για το πρόβλημα που θέλουμε να σπάσουμε και συνήθως χρησιμοποιείται ως τελευταία επιλογή.

4.2 Συμμετρικά Κρυπτοσυστήματα (Symmetric Encryption)

Σε αυτή την κατηγορία περιλαμβάνονται οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούν συμμετρικά κλειδιά (symmetric keys). Με τον όρο συμμετρικό κλειδί αναφερόμαστε στη φιλοσοφία ότι το ίδιο κλειδί χρησιμοποιείται για την κρυπτογράφηση αλλά και για την αποκρυπτογράφηση. Επομένως, το κλειδί παριστάνει ένα «κοινό μυστικό» (shared secret) μεταξύ των δύο (ή περισσότερων) εμπλεκόμενων ατόμων, και είναι υποχρεωτικό να το γνωρίζουν όλοι για να γίνει μία σωστή επικοινωνία. Για τη δημιουργία κλειδιών υπάρχουν πολλοί τρόποι, με τους ασφαλέστερους να θεωρούνται οι «τυχαιοποιημένοι», εκεί όπου είναι εφικτό να χρησιμοποιηθούν.

Οι αλγόριθμοι αυτής της κατηγορίας μπορεί να είναι δύο ειδών, block ciphers, και stream ciphers. Περιληπτικά, η πρώτη κατηγορία όπως εξυπακούεται και από το όνομα, ενεργεί πάνω σε blocks, σειριακά block ανά block. Ένα block είναι ένας σταθερός αριθμός bytes ο οποίος διαφέρει από υλοποίηση σε υλοποίηση ή από αλγόριθμο σε αλγόριθμο. Υπάρχουν διάφορες υλοποιήσεις, πάντοτε βασισμένες στο κοινό κλειδί. Όσον αφορά τους stream ciphers, χρησιμοποιούν το κλειδί ως seed για να αρχικοποιήσουν ένα ψευδοτυχαίο δημιουργό αριθμών (Pseudorandom Number Generator). Στη συνέχεια, χρησιμοποιούν τα «τυχαία» bytes/bits και τα συνδυάζουν με κάποιο τρόπο αντίστοιχα με bytes/bits του καθαρού κειμένου για να παράξουν το κρυπτογραφημένο κείμενο. Η αποκρυπτογράφηση και των δύο ειδών γίνεται με μια αντίστροφη διαδικασία η οποία χρειάζεται απαραίτητα το κλειδί.

Τα πλεονεκτήματα των αλγορίθμων συμμετρικής κρυπτογράφησης είναι ότι υπάρχουν υλοποιήσεις που είναι υπολογιστικά ανέφικτο να «σπάσουν», δηλαδή να υπολογιστεί το κλειδί. Υπολογιστικά ανέφικτο υπονοεί ότι ο μέσος χρόνος που χρειάζεται για την επιτυχή ανακάλυψη του κλειδιού μέσω δοκιμής συνδυασμών είναι πάρα πολύς, μετρημένος πιθανότατα σε δεκάδες χρόνων. Επιπλέον, εφαρμόζουν πολύπλοκες πράξεις πάνω στα blocks, οι οποίες δεν λύνονται εύκολα, με στόχο να είναι πάρα πολύ δύσκολο μέχρι σχεδόν αδύνατο να σπάσει ο αλγόριθμος χωρίς το κλειδί. Τα γεγονότα αυτά κάνουν τους συγκεκριμένους αλγορίθμους πολύ ασφαλείς. Επίσης, είναι και πολύ αποδοτικοί σε ταχύτητα λόγω των υπολογισμών που κάνουν οι οποίοι είναι

σχετικά φθηνοί με σχετικά μικρά (υπολογιστικά) κλειδιά, κάποιοι από τους οποίους γίνονται στο υλικό.

Όσον αφορά μειονεκτήματα, ένα είναι το κύριο μειονέκτημα αυτών των υλοποιήσεων και αυτό είναι η μεταφορά του «κοινού μυστικού». Επειδή αυτό είναι το μόνο που χρειάζεται για την αποκρυπτογράφηση, είναι ένα θέμα να μπορεί να ανταλλάσσεται με ασφάλεια στην άλλη πλευρά. Εδώ είναι που έρχονται και τα άλλα είδη αλγορίθμων (τα οποία θα περιγραφούν στα επόμενα υποκεφάλαια) να συνεισφέρουν στο σύστημα, και θα εξηγήσουμε στη συνέχεια πώς.

4.2.1 Ο Αλγόριθμος AES

Ο Αλγόριθμος AES (Advanced Encryption Standard), είναι ένας block cipher αλγόριθμος συμμετρικής κρυπτογράφησης και συγκεκριμένα θεωρείται από τους πιο ασφαλείς αν όχι ο πιο ασφαλής. Σε γενικές γραμμές, δουλεύει με 128-bit blocks και περιλαμβάνει τέσσερα κύρια στάδια (SubBytes, ShiftRows, MixColumns, AddRoundKey) κατά τη διάρκεια της κρυπτογράφησης τα οποία αποτελούν και ένα γύρο (round) από τους πολλούς που χρειάζονται για να ολοκληρωθεί η κρυπτογράφηση/αποκρυπτογράφηση. Ο αριθμός των γύρων εξαρτάται από το μέγεθος του κλειδιού που χρησιμοποιείται. Ο AES έχει την δυνατότητα να δουλέψει με τρία μεγέθη κλειδιών: 128-bit, 192-bit, 256-bit, με 10, 12, 14 γύρους αντίστοιχα. Όσο μεγαλύτερο είναι το μέγεθος του κλειδιού τόσο περισσότεροι γύροι θα εκτελεστούν μέχρι να τελειώσει η κρυπτογράφηση, γεγονός που σίγουρα συμβάλλει στην ασφάλεια. Πέραν του αριθμού των γύρων, το μέγεθος του κλειδιού επηρεάζει την ασφάλεια και σε άλλα σημεία στα διάφορα στάδια του αλγορίθμου, αλλά και στο ότι είναι δυσκολότερο να σπάσει. Στο δικό μας σύστημα, ο αλγόριθμος που θα χρησιμοποιηθεί είναι ο AES-256.

Θα προχωρήσουμε αναλύοντας περισσότερο τον αλγόριθμο κάνοντας μια περιγραφική συζήτηση πάνω στα τέσσερα στάδιά που αποτελούν τους γύρους του. Αρχικά, να πούμε ξανά ότι ο AES ενεργεί πάνω σε blocks (128-bit), και συνεπώς, οι ενέργειες που θα ακολουθήσουν θα αναφέρονται πάντα σε ένα block. Θα

χρησιμοποιούμε ως παράδειγμα κλειδί 128-bit για χάρη απλότητας και είναι επίσης βολικό να τα αναπαριστάμε ως πίνακες 4x4 bytes:

- **SubBytes:** Σε αυτό το στάδιο αντικαθίστανται τα bytes με ένα στατικό τρόπο από bytes από ένα πίνακα αναζήτησης (lookup table) ο οποίος βασίζεται πάνω στο Rijndael S-box.
- **ShiftRows:** Στη συνέχεια, ανάλογα με το δείκτη της κάθε γραμμής του block (0, 1, 2, 3) εφαρμόζουμε τόσα bytes αριστερή μετατόπιση.
- **MixColumns:** Ακολούθως, κάθε στήλη του block αναπαρίσταται ως ένας 1x4 πίνακας και πολλαπλασιάζεται με ένα στατικό 4x4 πίνακα.
- **AddRoundKey:** Σε αυτό το στάδιο, συνδυάζεται το κλειδί του συγκεκριμένου γύρου με το block με ένα bitwise XOR. Για να υπολογιστεί το κλειδί του γύρου χρειάζεται το προηγούμενο κλειδί (αρχικά είναι το κανονικό κλειδί) και εφαρμόζεται το Rijndael's key schedule.

Αυτά είναι τα τέσσερα στάδια που εφαρμόζονται σε κάθε γύρο του αλγορίθμου, ωστόσο υπάρχουν δύο σημαντικά σημεία όσον αφορά αυτό:

- Πριν να ξεκινήσουν οι γύροι εφαρμόζεται ένα AddRoundKey στο κλειδί που παίρνουμε ως είσοδο.
- Στον τελευταίο γύρο (10^o, 12^o ή 14^o), ΔΕΝ εφαρμόζεται το στάδιο MixColumns.

4.3 Ασύμμετρα Κρυπτοσυστήματα (Asymmetric Encryption)

Σε αυτή την κατηγορία περιλαμβάνονται οι αλγόριθμοι κρυπτογράφησης που χρησιμοποιούν ασύμμετρα κλειδιά. Αυτό το είδος κρυπτογράφησης επίσης είναι γνωστό ως κρυπτογράφηση δημόσιου κλειδιού. Με τον όρο ασύμμετρα κλειδιά αναφερόμαστε στη φιλοσοφία ότι υπάρχουν δύο κλειδιά, τα οποία είναι μαθηματικά συσχετισμένα, ένα δημόσιο κλειδί (public key) και ένα ιδιωτικό κλειδί (private key). Ο τρόπος που δημιουργούνται τα κλειδιά είναι καθαρά μαθηματικός και βασίζεται σε μαθηματικά προβλήματα για τα οποία δεν υπάρχει (προς το παρόν) αποδοτική λύση. Συνήθως κάθε τέτοιος αλγόριθμος βασίζεται σε διαφορετικό πρόβλημα ή εφαρμόζει διαφορετική

προσέγγιση. Η δημιουργία των κλειδιών είναι φθηνή διαδικασία όμως. Ο τρόπος που είναι συνδεδεμένα αυτά τα κλειδιά είναι ότι το ένα πετυχαίνει την αντίστροφη ενέργεια του άλλου. Για παράδειγμα, αν κρυπτογραφήσουμε με οποιοδήποτε από τα δύο τότε μόνο με το άλλο μπορούμε να αποκρυπτογραφήσουμε, ούτε καν με το ίδιο που κρυπτογραφήσαμε.

Σε μια επικοινωνία που θέλουμε να κάνουμε ασφαλή, ο κάθε εμπλεκόμενος πρέπει να δημιουργήσει το δικό του ζεύγος κλειδιών, και να μοιραστεί με τους υπόλοιπους το δημόσιο κλειδί του ανοιχτά και καθαρά. Όταν οποιοσδήποτε θέλει να στείλει κάτι σε κάποιον άλλον, απλά κρυπτογραφεί το μήνυμα με το δημόσιο κλειδί του παραλήπτη και το στέλνει. Με αυτό τον τρόπο μόνο ο προορισμένος παραλήπτης (ο οποίος είναι ο μόνος που έχει το ιδιωτικό κλειδί) μπορεί να αποκρυπτογραφήσει το μήνυμα. Η ασφάλεια αυτού του αλγορίθμου στο ότι παρόλο που το ένα κλειδί είναι γνωστό ανοιχτά και μπορεί να το γνωρίζει ο καθένας, θεωρείται υπολογιστικά ανέφικτο για κάποιον να υπολογίσει το ένα κλειδί από το άλλο· σε αυτή την περίπτωση το ιδιωτικό από το δημόσιο.

Αυτό το είδος αλγορίθμων έχει και πλεονεκτήματα όπως και μειονεκτήματα. Το κύριο πλεονέκτημα είναι το γεγονός ότι το ιδιωτικό κλειδί δεν χρειάζεται ποτέ να μοιραστεί με κανέναν, άρα δεν υπάρχει κίνδυνος στο να στέλνουμε το δημόσιο κλειδί. Επίσης, υπάρχουν κι άλλοι τρόποι να χρησιμοποιηθούν αυτά τα κλειδιά, πέραν από το προηγούμενο σενάριο. Το μειονέκτημα της κρυπτογράφησης δημόσιου κλειδιού, είναι ότι για να διατηρεί τα ασφαλή της χαρακτηριστικά και να προστατεύεται από υπολογισμό κλειδιών, χρησιμοποιούνται πολύ μεγάλα κλειδιά. Επίσης, η διαδικασία κρυπτογράφησης και αποκρυπτογράφησης περιλαμβάνει τεχνικές που είναι αισθητά πιο αργές από τους γνωστούς συμμετρικούς αλγορίθμους. Όλα αυτά κάνουν αυτό το είδος κρυπτογράφησης να κοστίζει πολύ, για να εφαρμοστεί ανεξάρτητο σε ένα σύστημα πάνω σε πολλά καθαρά δεδομένα. Η κρυπτογράφηση/αποκρυπτογράφηση ενός κλειδιού όμως είναι μηδαμινή και εδώ είναι που έρχονται αυτοί οι αλγόριθμοι να προσφέρουν.

Συνεπώς, εκμεταλλευόμαστε την ταχύτητα και ασφάλεια των συμμετρικών αλγορίθμων με την ασφάλεια επικοινωνίας των ασύμμετρων για να αλληλοκαλύπτει ο ένας τα ελαττώματα του άλλου. Συγκεκριμένα, τυπικά οι ασύμμετροι αλγόριθμοι

χρησιμοποιούντα για ανταλλαγή συμμετρικών κλειδιών, και οι συμμετρικοί για κρυπτογράφηση δεδομένων. Αυτή είναι και η γενική κοινή λογική πίσω από ένα ολοκληρωμένο και ασφαλές σύστημα. Ο συμμετρικός αλγόριθμος κρυπτογράφησης που θα δούμε και θα χρησιμοποιήσουμε είναι ο RSA, όπως επίσης και η τεχνική της ψηφιακής υπογραφής.

4.3.1 Ο Αλγόριθμος RSA

Ο αλγόριθμος RSA, είναι από τους πιο γνωστούς αλγόριθμους ασύμμετρης κρυπτογράφησης και πιο χρησιμοποιήσιμους στην πράξη. Τα αρχικά της ονομασίας προέρχονται από τα επίθετα των τριών ατόμων που πρότειναν την τεχνική: Rivest, Shamir, Adleman. Το μαθηματικό πρόβλημα στο οποίο βασίζεται για την δημιουργία των κλειδιών ο συγκεκριμένος αλγόριθμος είναι το πρόβλημα της παραγοντοποίησης ενός αριθμού σε πρώτους αριθμούς. Για πολύ μεγάλους αριθμούς, δεν υπάρχει μέχρι σήμερα αποδοτική λύση στο πρόβλημα αυτό, με αποτέλεσμα να θεωρείται υπολογιστικά ανέφικτο να «σπάσουν» τα κλειδιά.

Σε αυτό το σημείο θα δείξουμε βήμα ανά βήμα την διαδικασία που ακολουθείται από κάθε εμπλεκόμενο για τη δημιουργία του ζεύγους των κλειδιών του:

- Επιλέγονται δύο τυχαίοι μεγάλοι πρώτοι αριθμοί p και q (απαιτούνται τουλάχιστον 2048 bits για τον καθένα για να θεωρείται ασφαλής η υλοποίηση).
- Υπολογίζεται το $n = p * q$.
- Υπολογίζεται το $\varphi(n) = \varphi(p) * \varphi(q) = (p - 1) * (q - 1) = n - (p + q - 1)$.
- Επιλέγεται περιττός ακέραιος e τέτοιος ώστε $1 < e < \varphi(n)$ και $\text{ΜΚΔ}(e, \varphi(n)) = 1$.
- Υπολογισμός του d as $d * e \bmod \varphi(n) = 1$.
- Το e μαζί με το n αποτελούν το δημόσιο κλειδί και το d μαζί με το n αποτελούν το ιδιωτικό κλειδί.

Για την κρυπτογράφηση/αποκρυπτογράφηση του μηνύματος M, το μετατρέπουμε σε m τέτοιο ώστε $0 <= m < n$, επομένως το μήνυμα μπορεί να χρειαστεί να διασπαστεί σε blocks (πολλά m).

Για κρυπτογράφηση (c = ciphertext) με το δημόσιο κλειδί e εφαρμόζουμε:

$$c \equiv m^e \pmod{n}$$

Για αποκρυπτογράφηση με το ιδιωτικό κλειδί d εφαρμόζουμε:

$$m \equiv c^d \pmod{n}$$

4.3.2 Η Ψηφιακή Υπογραφή (Digital Signature)

Πληροφοριακά και περιληπτικά, τρεις κύριες αρχές της Ασφάλειας Πληροφοριών που πρέπει να πληροί οποιοδήποτε σύστημα θεωρείται ασφαλές είναι:

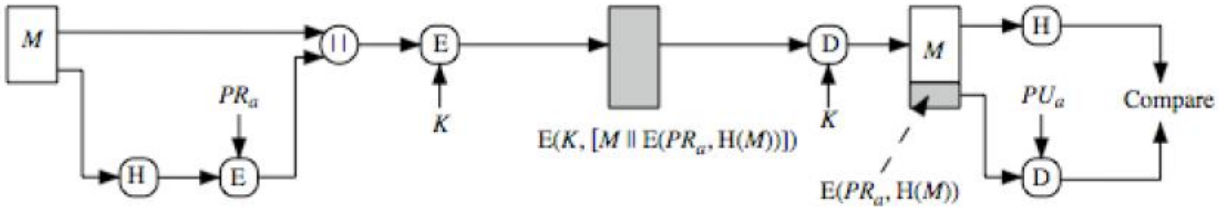
- Εμπιστευτικότητα (Confidentiality): Η έννοια της απόκρυψης δεδομένων από όλους εκτός προορισμένους παραλήπτες (αυτούς που έχουν το κλειδί) μέσω κρυπτογράφησης.
- Ταυτοποίηση (Authentication): Η έννοια του να αποδεικνύει κάποιος ότι είναι αυτός που ισχυρίζεται, δηλαδή ο εαυτός του, με τη χρήση μιας πληροφορίας που μόνο εκείνος θα μπορούσε να γνωρίζει (π.χ. το ιδιωτικό κλειδί του). Για να είναι ξεκάθαρο, η εμπιστευτικότητα δεν εξυπονοεί ταυτοποίηση αφού πολλά άτομα μπορεί να έχουν το κλειδί που χρησιμοποιήθηκε για την κρυπτογράφηση/αποστολή.
- Ακεραιότητα (Integrity): Η εγγύηση ότι το μήνυμα που λαμβάνει κάποιος είναι ακριβώς όπως στάλθηκε από τον αποστολέα και δεν αλλοιώθηκε με οποιονδήποτε τρόπο, είτε από κάποιο σφάλμα είτε από κάποιο κακόβουλο άτομο.

Ουσιαστικά μέχρι τώρα αναφερόμασταν στην εμπιστευτικότητα, και επίσης είναι το πιο ευκατανόητο από τους τρεις αυτούς ορισμούς, αφού είναι το πρώτο πράγμα που φαντάζεται κάποιος. Σε αυτό το σημείο που κατανοήσαμε και τις άλλες δύο έννοιες, θα προχωρήσουμε στην ψηφιακή υπογραφή (digital signature). Η ψηφιακή υπογραφή δημιουργήθηκε για να αποδεικνύει αυθεντικότητα (authenticity) των μηνυμάτων, μέσω ενός αριθμού από bytes τα οποία παράγονται με την εφαρμογή κάποιων τεχνικών πάνω στο καθαρό μήνυμα. Ο ορισμός αυθεντικότητα παριστάνει το σύνολο της ταυτοποίησης και της ακεραιότητα.

Η ψηφιακή υπογραφή τυπικά περιλαμβάνει έννοιες της ασύμμετρης κρυπτογράφησης και του κατατεμαχισμού (hashing). Ο κατατεμαχισμός θα εξηγηθεί αμέσως μετά αλλά θα αναφέρουμε τα απαραίτητα χαρακτηριστικά του για κατανόηση της ψηφιακής υπογραφής. Μια κρυπτογραφική συνάρτηση κατατεμαχισμού παίρνει ψηφιακά δεδομένα (αυθαίρετα bytes) απροσδιόριστου μεγέθους ως είσοδο και παράγει μία σύνοψη (digest), η οποία είναι επίσης ψηφιακά δεδομένα. Το κύριο σημείο μιας συνάρτησης κατατεμαχισμού είναι ότι για οποιοδήποτε μήνυμα παράγεται μοναδικό και διαφορετικό αποτέλεσμα, και η έννοια αυτή μας είναι απαραίτητη για έλεγχο ακεραιότητας. Αν έχουμε τη σύνοψη ενός μηνύματος που μόλις παραλάβαμε, μπορούμε να εφαρμόσουμε και εμείς την ίδια συνάρτηση κατατεμαχισμού στο μήνυμα για να ελέγξουμε αν ταιριάζουν οι δύο συνόψεις. Αν όχι, τότε το μήνυμα αλλοιώθηκε κατά την μετάδοση, είτε κακόβουλα είτε από κάποιο λάθος. Επομένως, η σύνοψη ενός μηνύματος συνοδεύει το μήνυμα στην αποστολή για βεβαίωση ότι το μήνυμα θα μείνει αναλλοίωτο. Η σύνοψη επικολλάται πάνω στο μήνυμα και κρυπτογραφείται μαζί του. Λόγω της σύγχυσης και διάχυσης δεν είναι εφικτό ο κακόβουλος να αλλοιώσει το κρυπτογραφημένο μήνυμα και ταυτόχρονα να διατηρηθεί ορθή σύνοψη.

Μέχρι τώρα πετύχαμε την ακεραιότητα, και στη συνέχεια θα χρησιμοποιήσουμε ασύμμετρα κλειδιά για να πετύχουμε και την ταυτοποίηση. Απ' ό,τι είδαμε μέχρι τώρα, τα ασύμμετρα κλειδιά μπορούν να χρησιμοποιηθούν για βεβαιωθούμε ότι μόνο ο επιθυμητός παραλήπτης μπορεί να διαβάσει ένα μήνυμα. Αυτό γίνεται με το να κρυπτογραφήσουμε πρώτα με το δημόσιο, και ο παραλήπτης να χρησιμοποιήσει το ιδιωτικό του κλειδί για να διαβάσει το μήνυμα. Σε αυτό το σημείο, θα χρησιμοποιήσουμε την αντίθετη λογική, κρυπτογράφηση με το ιδιωτικό κλειδί στην πλευρά του αποστολέα. Αυτό δεν προσφέρει κάποιο είδος εμπιστευτικότητας στα δεδομένα αφού όλοι μπορεί να έχουν το δημόσιο κλειδί. Ωστόσο, δεν θα το χρησιμοποιήσουμε στα καθαρά δεδομένα, αλλά στη σύνοψη που παράγεται από την συνάρτηση κατατεμαχισμού. Συνεπώς τα δεδομένα μας παραμένουν συμμετρικά κρυπτογραφημένα όπως πριν, και συνοδεύονται από την σύνοψη της συνάρτησης κατατεμαχισμού κρυπτογραφημένη με το ιδιωτικό κλειδί του αποστολέα. Τέλος, ο παραλήπτης απλά αποκρυπτογραφεί με το συμμετρικό κλειδί, παίρνει την σύνοψη και την αποκρυπτογραφεί με το δημόσιο για να πάρει την καθαρή σύνοψη, και υπολογίζει νέα σύνοψη από το μήνυμα για σύγκριση των δύο. Αν οτιδήποτε δεν πάει ακριβώς όπως πρέπει, τότε κάτι έχει παραβιαστεί.

Πιο κάτω θα δείξουμε μία τυπική αποστολή μηνύματος ακριβώς όπως αυτή που περιγράψαμε μόλις προηγουμένως, με συμμετρική κρυπτογράφηση, κατατεμαχισμό, ψηφιακή υπογραφή, και πώς ο παραλήπτης επαληθεύει:



Σχήμα 4.1: Αποστολή κρυπτογραφημένου μηνύματος με ψηφιακή υπογραφή.

M: Το καθαρό μήνυμα.

H: Η κρυπτογραφικά συνάρτηση κατατεμαχισμού.

E: Κρυπτογράφηση συμμετρική ή ασύμμετρη ανάλογα με το κλειδί που παρέχεται.

D: Αποκρυπτογράφηση συμμετρική ή ασύμμετρη ανάλογα με το κλειδί που παρέχεται.

PRa: Το ιδιωτικό κλειδί του a (αποστολέα).

PUa: Το δημόσιο κλειδί του a.

K: Το συμμετρικό κλειδί των δύο πλευρών.

| |: Συνένωση/Επικόλληση των (δύο) εισόδων.

4.4 Κατατεμαχισμός (Hashing)

Σε αυτό το σημείο θα μιλήσουμε λίγο πιο αναλυτικά για τις κρυπτογραφικές συναρτήσεις κατατεμαχισμού. Μια κρυπτογραφική συνάρτηση κατατεμαχισμού όπως αναφέρθηκε και πριν, παίρνει ψηφιακά δεδομένα απροσδιόριστου μεγέθους ως είσοδο και παράγει μία σύνοψη, άλλα ψηφιακά δεδομένα. Οι συνόψεις δεν θεωρούνται κατά κανένα λόγο κρυπτογραφήσεις, ούτε μπορούν να φερθούν με τέτοιο τρόπο, όπως θα φανεί και στη συνέχεια με τις ιδιότητες που πρέπει να πληροί μία κρυπτογραφική συνάρτηση κατατεμαχισμού. Επίσης, οι συνόψεις αυτές στους περισσότερους αλγόριθμους κατατεμαχισμού έχουν σταθερό μέγεθος. Όσο μεγαλύτερο είναι αυτό το μέγεθος τόσο περισσότερες συνόψεις υπάρχουν και συνεπώς τείνουν να είναι «πιο μοναδικές».

Προηγουμένως αναφέραμε μόνο τα σημεία της συνάρτησης κατατεμαχισμού που μας ενδιέφεραν, με κατανοητά λόγια για εκείνο το παράδειγμα. Τώρα θα τα συζητήσουμε πιο τυπικά. Μια τέτοια συνάρτηση πρέπει να έχει τέσσερις βασικές ιδιότητες:

- Να είναι one-way, δηλαδή να μην είναι εφικτό να υπολογιστεί το μήνυμα αν έχουμε στην κατοχή μας μόνο την σύνοψη.
- Να είναι γρήγορο και εύκολο στον υπολογισμό, εφόσον έχουμε το μήνυμα.
- Να μην είναι εφικτό να αλλοιώσουμε το μήνυμα και η σύνοψη να μην αλλάζει.
- Κάθε σύνοψη να είναι μοναδική, να είναι ανέφικτο να υπολογίσουμε δύο μηνύματα με την ίδια σύνοψη.

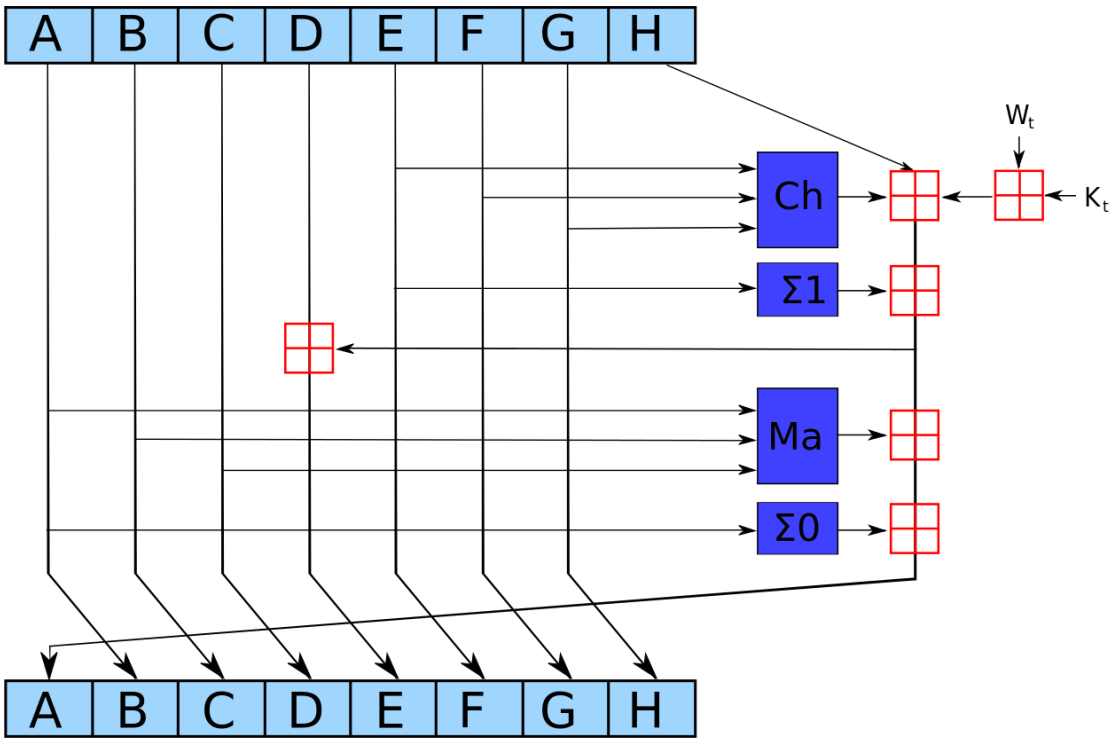
Όσο καλύτερα πληροί αυτές τις ιδιότητες μιας συνάρτηση κατατεμαχισμού, τόσο δυσκολότερο είναι να «σπάσει». Με τον χαρακτηρισμό «σπάσει» αναφερόμαστε στον οποιοδήποτε τρόπο συνεισφέρει σε ένα σύστημα μια τέτοια συνάρτηση, και πόσο κακό μπορεί να επιφέρει τυχόν παραβίαση των συνόψεων.

Μια κρυπτογραφική συνάρτηση κατατεμαχισμού έχει εφαρμογές σε πολλούς τομείς της Ασφάλειας Πληροφοριών. Ένας από αυτούς είναι η ψηφιακή υπογραφή που αναλύσαμε προηγουμένως, ή τα MAC (Message Authentication Code), αποθήκευση κωδικών, κ.ά.

4.4.1 Ο Αλγόριθμος SHA

Ο αλγόριθμος κατατεμαχισμού που θα χρησιμοποιήσουμε εμείς είναι ο SHA (Secure Hash Algorithm). Υπάρχουν διάφορες οικογένειες του αλγόριθμου SHA, SHA-0, SHA-1, SHA-2, SHA-3. Κάθε τέτοια οικογένεια προσθέτει νέους αλγόριθμους που αφορούν διαφορετικές υλοποιήσεις και διαφορετικά μεγέθη μηνυμάτων, ως επίσης και περισσότερη ασφάλεια. Εμείς χρησιμοποιούμε τον SHA-256, οποίος ανήκει στην οικογένεια SHA-2.

Οι αλγόριθμοι SHA εφαρμόζονται πάνω σε blocks των οποίων το μέγεθος μπορεί να διαφέρει από οικογένεια σε οικογένεια, και περιλαμβάνουν πολλούς γύρους (επαναλήψεις της ίδια πράξης), με κάθε γύρο να εξαρτάται από το αποτέλεσμα του προηγούμενου.



Σχήμα 4.2: Μία επανάληψη κατατεμαχισμού οικογένειας SHA-2.

Στο πιο πάνω σχήμα φαίνεται ένα γύρος αλγορίθμου οικογένειας SHA-2. Το κάθε γράμμα από τα A, B, C, …, H αποτελείται από 32 bits. Παρατηρούμε το πώς καταλήγουν τα γράμματα A, B, C, …, H στο τέλος του γύρου, και έτσι θα χρησιμοποιηθούν στον επόμενο. Το κόκκινο τετράγωνο αντιπροσωπεύει πρόσθεση των εισόδων ακολουθούμενη από modulo 2^{32}. Όλοι οι αριθμοί που αναφέρονται ρητά αφορούν μόνο το SHA-256. Τέλος, θα εξηγήσουμε τις πράξεις στα μπλε τετράγωνα:

- $\mathrm{Ch}(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$
- $\mathrm{Ma}(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
- $\Sigma_0(A) = (A >>> 2) \oplus (A >>> 13) \oplus (A >>> 22)$
- $\Sigma_1(E) = (E >>> 6) \oplus (E >>> 11) \oplus (E >>> 25)$

4.4.1.1 Παραγωγή Κλειδιού από Κωδικό (Password-Based Cryptography)

Στο σύστημα μας χρησιμοποιούμαι τον κωδικό (password) του κάθε χρήστη για να παράγουμε ένα συμμετρικό κλειδί το οποίο επηρεάζει κατά πολύ την ασφάλεια των πληροφοριών του χρήστη. Αυτή η προσέγγιση θεωρείτε ανασφαλής λόγω του ότι οι κωδικοί κάποτε μπορεί να είναι πολύ αδύναμοι και μικροί, πράγμα που δεν συνιστάται πουθενά. Εξαιτίας του πιο πάνω, είναι εύκολο με μια επίθεση ωμής βίας να ανακαλύψει κάποιος κακόβουλος τον κωδικό ενός χρήστη. Γι' αυτό το λόγο, κάνουμε χρήση της συνάρτησης PBKDF2 (Password-Based Key Derivation Function 2). Αυτή η μέθοδος περιλαμβάνει κρυπτογραφικά ψευδοτυχαίες συναρτήσεις (π.χ. αλγόριθμους κατατεμαχισμού όπως ο SHA), ενισχυμένες με ένα salt (επιπλέον bytes για ενίσχυση του καθαρού κωδικού), και πολλές (χιλιάδες) επαναλήψεις των συναρτήσεων αυτών έτσι ώστε να γίνει αρκετά αργό. Επίσης, μπορούμε να καθορίσουμε στη συνάρτηση το μέγεθος που θέλουμε να έχει το αποτέλεσμά της (το οποίο είναι ουσιαστικά μια ισχυρή σύνοψη), για να μπορούμε να το χρησιμοποιήσουμε ως κλειδί. Το «αρκετά» αργό συγκεκριμένα δεν υπονοεί να καθυστερεί του χρήστη από να το υπολογίζει. Αντιθέτως, είναι μέσα στις προδιαγραφές ότι πρέπει να είναι τόσο αργό όσο που δεν θα γίνεται αντιληπτό στον χρήστη σχεδόν καθόλου, αλλά θα κοστίζει πολύ σε όποιον θέλει να επιτεθεί στο σύστημα. Με μια επίθεση ωμής βίας ο χρόνος που παίρνει αυτός ο υπολογισμός θα αυξηθεί κατά πολύ και πιθανότατα δε θα αξίζει τον κόπο. Τέλος, τακτική αλλαγή κωδικού κρίνεται απαραίτητη.

4.5 Το Λειτουργικό Android

Το Android είναι ένα λειτουργικό σύστημα για κινητά έξυπνα τηλέφωνα, σχεδιασμένο από την Google. Το σύστημα που θα κτίσουμε σε αυτή την εργασία θα είναι υλοποιημένο σε λειτουργικό Android. Το λειτουργικό αυτό δεν υστερεί πουθενά όσον αφορά δυνατότητες και ευκολία χρήσης μπροστά στα άλλα λογισμικά διαθέσιμα για ιστοσελίδες κλπ. Επίσης, ο προγραμματιστής είναι σε θέση να πετύχει «οτιδήποτε» επιθυμεί, νοουμένου ότι ξέρει τον τρόπο, αλλά από την μεριά του λειτουργικού συστήματος όλα γίνονται με τα κατάλληλα δικαιώματα από τον χρήστη.

Η γλώσσα προγραμματισμού που χρησιμοποιείται στο Android είναι η Java για το λειτουργικό κομμάτι, αλλά αυτή είναι μόνο η βάση αφού υπάρχει ένας τεράστιος αριθμός βιβλιοθηκών για τις εκατοντάδες επιπλέον αποκλειστικές εντολές του λειτουργικού. Για το εμφανισιακό μέρος μίας εφαρμογής χρησιμοποιούνται αρχεία XML αλλά και Java, αναλόγως με την περίσταση. Επίσης, για σωστή χρήση του Android πρέπει να υπάρχει και κατανόηση της υποδομής και του αφαιρετικού τρόπου λειτουργίας του.

Κεφάλαιο 5

Το Σύστημα

Σε αυτό το κεφάλαιο θα αφοσιωθούμε πλήρως στο σύστημα που υλοποιήσαμε καλύπτοντας με λεπτομέρεια. Θα συμπεριλάβουμε λεκτική περιγραφή, διάγραμμα ροής, διαγράμματα για επεξήγηση συγκεκριμένων λειτουργιών, ενδεικτικές εικόνες και γενικά χαρακτηριστικά. Έχουμε ήδη περιγράψει διάφορους ορισμούς που θα χρησιμοποιούμε στη συνέχεια, γεγονός που θα επιτρέψει σε λιγότερο πολύπλοκη εξήγηση εφόσον έγιναν κατανοητά τα προηγούμενα.

5.1 Γενική Περιγραφή

Όπως αναφέραμε και νωρίτερα στο έγγραφο, η εφαρμογή θα έχει χαρακτηριστικά κοινωνικής δικτύωσης επιτρέποντας στους χρήστες να ανταλλάσσουν δεδομένα μεταξύ τους. Τα δεδομένα αυτά εξαρτούνται από τον κάθε χρήστη, εφόσον επιλέξει να έχει δεδομένα, η μπορεί να επιλέξει να μην τα μοιραστεί όλα. Επιπλέον, ο κάθε χρήστης είναι

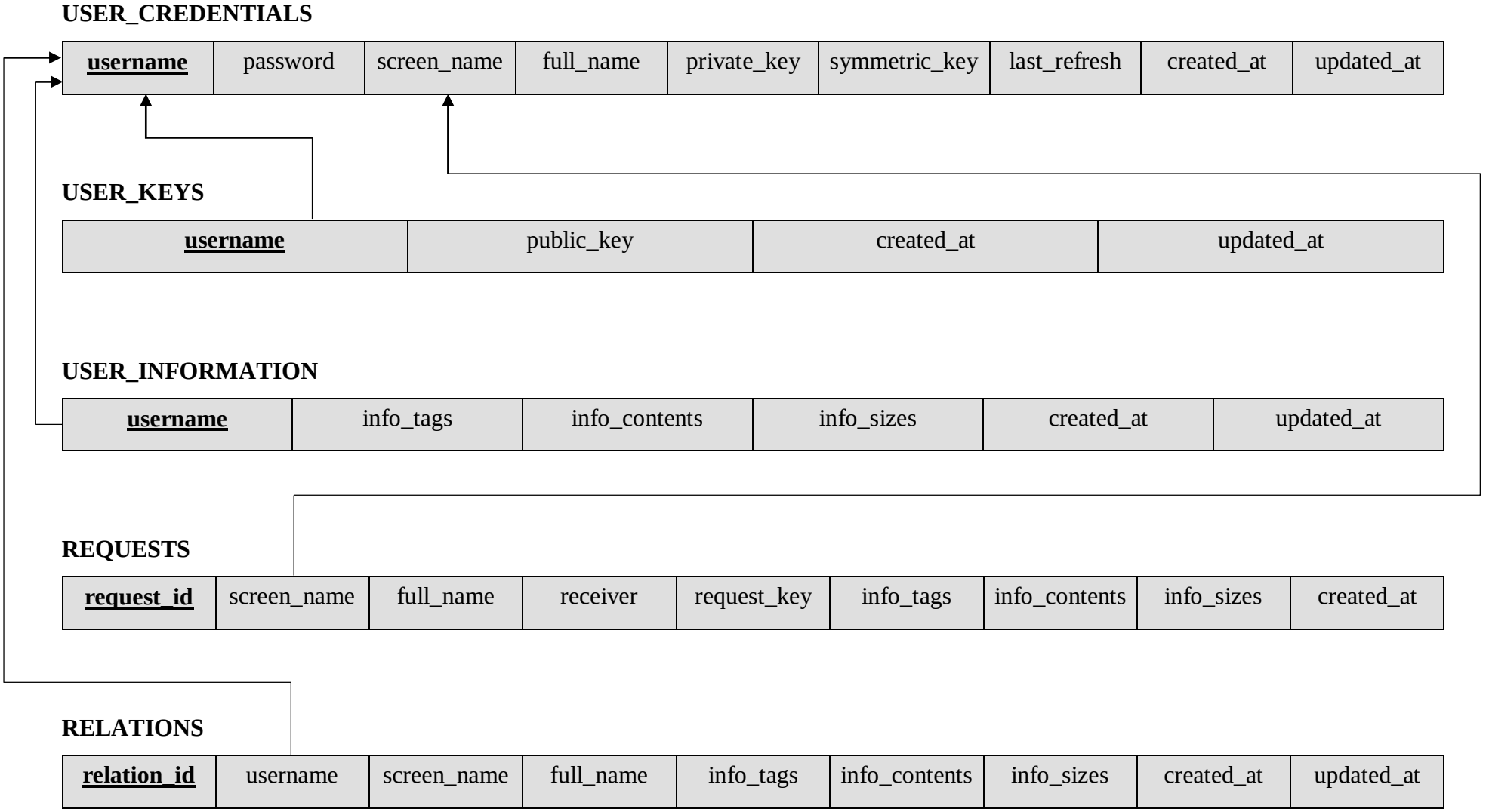
σε θέση να επιλέξει σε ποιους άλλους χρήστες θέλει να στείλει τις τυχόν πληροφορίες του. Τέλος, οι χρήστες έχουν την δυνατότητα να αναφερθούν όποτε θέλουν στα δεδομένα που έλαβαν από τον κάθε άλλο χρήστη/φίλο. Αυτή είναι η επιφανειακή λειτουργία του συστήματος σε υψηλό επίπεδο, εφόσον στην υποδομή του εφαρμόζονται ισχυρές τεχνικές ασφαλείας για την πλήρη ασφάλεια των χρηστών συνεχώς, χωρίς καν να γίνεται αντιληπτό από αυτούς. Η ασφάλεια αυτή περιλαμβάνει την απόκρυψη των δεδομένων που ανταλλάσσονται αλλά και της σχέσης των χρηστών που εμπλέκονται σε κάθε ανταλλαγή. Ακολούθως, θα αναφερθούμε λεπτομερώς στις λειτουργίες της εφαρμογής όπως επίσης και στο πώς επιτυγχάνεται η ασφάλεια γύρω από αυτές.

5.2 Λεπτομερής Λειτουργία

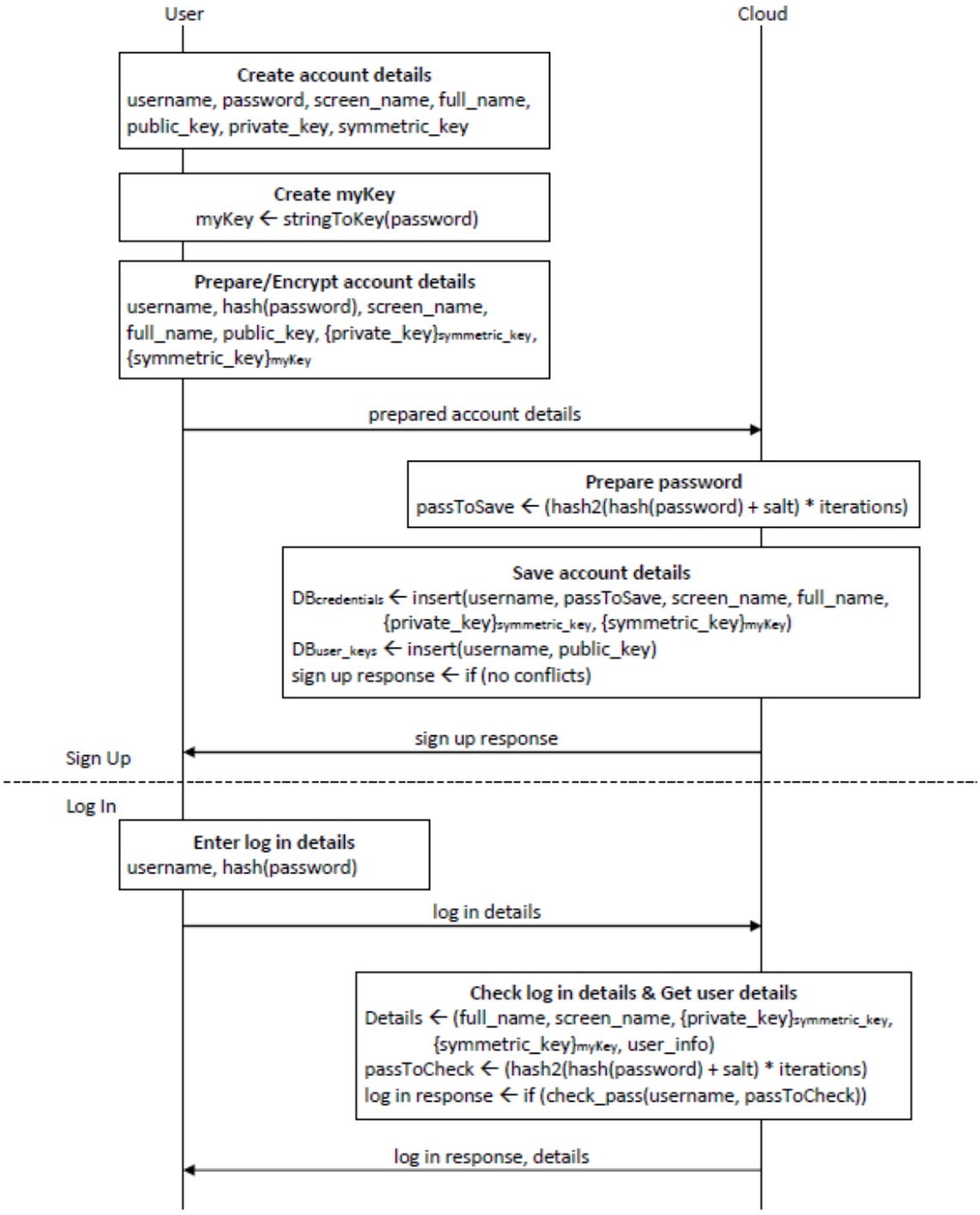
Σε αυτό το υποκεφάλαιο θα αναλύσουμε με χρήση πολύ λεπτομερών διαγραμμάτων κάθε είδους συναλλαγή δεδομένων που γίνεται στο σύστημά μας και κάθε υπολογισμού, ταυτόχρονα αναφερόμενοι στην ασφάλεια που υπάρχει σε κάθε σημείο και πώς επιτυγχάνεται. Κατά τη διάρκεια αυτής της περιγραφής θα γίνονται και αναφορές πιθανότατα στον ακριβή τρόπο που υλοποιήθηκε κάποια λειτουργία, εφόσον κρίνεται απαραίτητο ή σημαντικό. Ένα πολύ μεγάλο μέρος αυτών των λειτουργιών γίνονται αυτόματα από την εφαρμογή, και συνεπώς οι χρήστες δεν αντιλαμβάνονται πολλές από αυτές τις ενέργειες. Κύριος σκοπός αυτού του υποκεφαλαίου είναι να προβάλει κυρίως τις ενέργειες που γίνονται στο παρασκήνιο.

5.2.1 Βάση Δεδομένων

Εδώ θα παρουσιάσουμε την βάση δεδομένων που χρησιμοποιεί η εφαρμογή μας, και οποία βρίσκεται στο υπολογιστικό νέφος. Αναφορά στο παρακάτω μοντέλο συνιστάται κατά τη διάρκεια της μελέτης των διαγραμμάτων των λειτουργιών που θα ακολουθήσουν, για καλύτερη κατανόηση.


USER_CREDENTIALS
username
password
screen_name
full_name
private_key
symmetric_key
last_refresh
created_at
updated_at
USER_KEYS
username
public_key
created_at
updated_at
USER_INFORMATION
username
info_tags
info_contents
info_sizes
created_at
updated_at
REQUESTS
request_id
screen_name
full_name
receiver
request_key
info_tags
info_contents
info_sizes
created_at
RELATIONS
relation_id
username
screen_name
full_name
info_tags
info_contents
info_sizes
created_at
updated_at

5.2.2 Sign Up/Log In



Σχήμα 5.1: Δημιουργία λογαριασμού και σύνδεση χρήστη.

Στο πιο πάνω διάγραμμα φαίνεται η διαδικασία δημιουργίας λογαριασμού (sign up) και σύνδεσης (log in) ενός χρήστη με το νέφος. Η διαδικασία σύνδεσης θα

αναφέρεται και σε επόμενα υποκεφάλαια για σκοπούς ταυτοποίησης σε διάφορες ενέργειες, δηλαδή ότι ο χρήστης είναι αυτός που ισχυρίζεται.

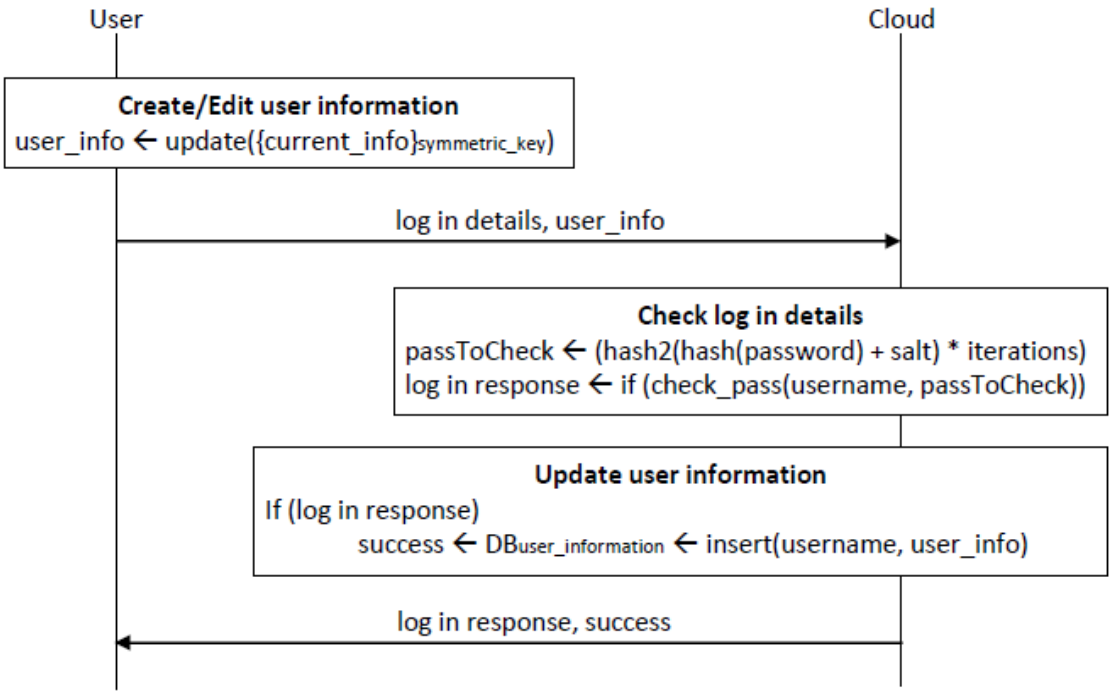
Για την δημιουργία λογαριασμού κάθε χρήστης δημιουργεί τον κωδικό του, ένα δημόσιο κλειδί, ένα ιδιωτικό κλειδί, και ένα προσωπικό συμμετρικό κλειδί, όπως και τα στοιχεία του (όνομα λογαριασμού, ονοματεπώνυμο, ψευδώνυμο). Έπειτα, χρησιμοποιεί το συμμετρικό κλειδί που δημιούργησε για να κρυπτογραφήσει με AES το ιδιωτικό κλειδί του έτσι ώστε να έχει μόνο αυτός πρόσβαση πάνω του. Στη συνέχεια, δημιουργεί το myKey του μέσω του κωδικού του με τον τρόπο που περιγράψαμε στο κεφάλαιο 4.4.1.1 με σκοπό να κρυπτογραφήσει το συμμετρικό του κλειδί. Αυτό γίνεται για να μπορεί μέσω του myKey (που είναι στην ουσία ο κωδικός) να έχει κάθε φορά που θα συνδέεται πρόσβαση στο συμμετρικό κλειδί. Επίσης, κατατεμαχίζει (hash) τον κωδικό του για να τον προστατεύσει κατά την διάδοση προς το νέφος και για προστασία του myKey του, συγκεκριμένα ξανά με την προσέγγιση του 4.4.1.1. Αφού η αντιστοιχία κωδικού-σύνοψης (digest) θεωρείται 1 προς 1, η ασφάλεια του κωδικού δεν αλλάζει. Το myKey θα μπορούσε να χρησιμοποιηθεί στη θέση του προσωπικού συμμετρικού κλειδιού, αλλά αφού το myKey είναι το πιο ευάλωτο (επειδή προέρχεται από τον κωδικό του χρήστη), είναι ασφαλέστερο να χρησιμοποιούμε τυχαία κλειδιά για συχνές κρυπτογραφήσεις. Τέλος, μετά από αυτή τη διαδικασία στέλνει όλα τα παραπάνω στο νέφος για αποθήκευση, με τα στοιχεία του να είναι σε καθαρό κείμενο για να δίνεται η δυνατότητα στους άλλους χρήστες να αναφέρονται σε έναν συγκεκριμένο χρήστη.

Το νέφος όταν παραλάβει τα δεδομένα αυτά, τα αποθηκεύει όπως φαίνεται και στο σχήμα, αφού εφαρμόσει κάποιο είδος δικού του κατατεμαχισμού για τον ήδη κατατεμαχισμένο κωδικό, τεχνική που είναι συνηθισμένη για αυτή τη χρήση. Εφαρμόζει ουσιαστικά την PBKDF2 συνάρτηση που αναφέραμε νωρίτερα στο έγγραφο για να το πετύχει αυτό, η οποία προσφέρει και το δικό της επίπεδο ασφαλείας. Ο λόγος που γίνεται αυτός ο κατατεμαχισμός στο νέφος είναι για να παρέχεται ασφάλεια σε περίπτωση που παραβιαστεί η βάση δεδομένων του νέφους από τρίτους, δηλαδή ο ουσιαστικός κωδικός να μη είναι αποθηκευμένος πουθενά, παρά μόνο η σύνοψή του (η οποία δεν αντιστρέφεται). Σε περίπτωση παραβίασης, ο επιτιθέμενος θα πρέπει να υπολογίζει συνόψεις κωδικών με τις δεδομένες επαναλήψεις και salt, μέχρι να βρει το σωστό που να

ταιριάζει στην σύνοψη που είναι αποθηκευμένη. Αυτό θεωρητικά είναι εξίσου δύσκολο με μια επίθεση ωμής βίας μέχρι να πετύχουν τον καθαρό κωδικό.

Κάθε φορά που θέλει να συνδεθεί ο χρήστης, υπολογίζεται η σύνοψη του κωδικού που εισήγαγε με τις ίδιες επαναλήψεις και salt, και συγκρίνεται με αυτήν που είναι αποθηκευμένη. Εφόσον όλα επιτύχουν, επιτρέπεται η πρόσβαση. Τώρα κατά την πρόσβαση του χρήστη στο σύστημα, παίρνει τα στοιχεία του, τα κλειδιά του, και όλα τα προσωπικά δεδομένα που έχει αποθηκεύσει για τον εαυτό του, έτσι ώστε να μπορεί να τα χρησιμοποιήσει σε συναλλαγές ή και να τα επεξεργαστεί αν επιθυμεί.

5.2.3 Εισαγωγή/Διαγραφή Δεδομένων

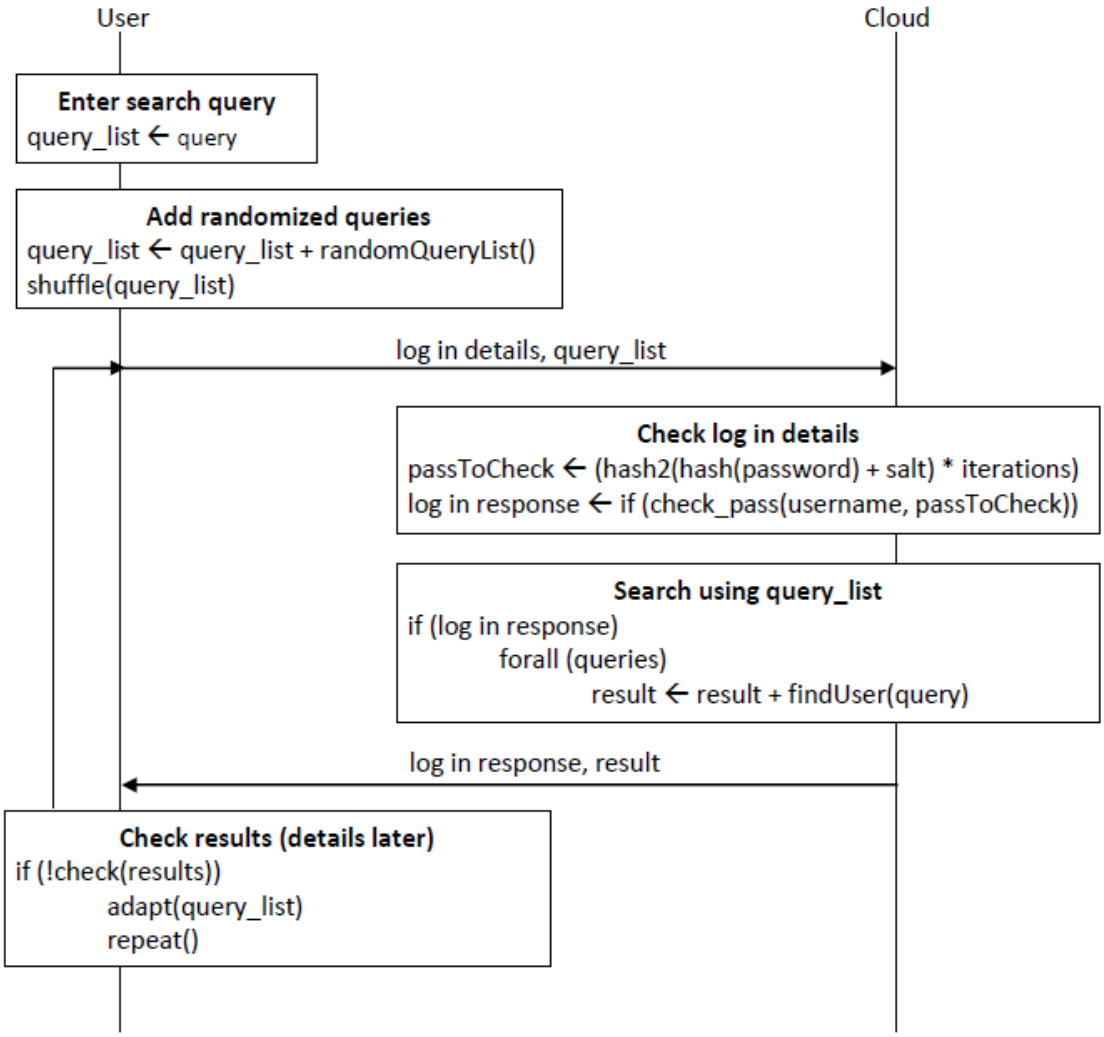


Σχήμα 5.2: Ενημέρωση προσωπικών δεδομένων του χρήστη μετά από κάποια αλλαγή.

Σε αυτό το στάδιο γίνεται η ιδιωτική ανανέωση των προσωπικών δεδομένων του χρήστη. Τα αυτά δεδομένα εισάγονται με τη μορφή [Είδος | Περιεχόμενα], μέσω της εφαρμογής στο σύστημα. Ένας χρήστης μπορεί να έχει όσα δεδομένα επιθυμεί, με την εγγύηση ότι είναι εντελώς ιδιωτικά εφόσον δεν επιλέξει να τα μοιραστεί με κάποιον. Η ιδιωτικότητα επιτυγχάνεται χρησιμοποιώντας το προσωπικό συμμετρικό κλειδί του

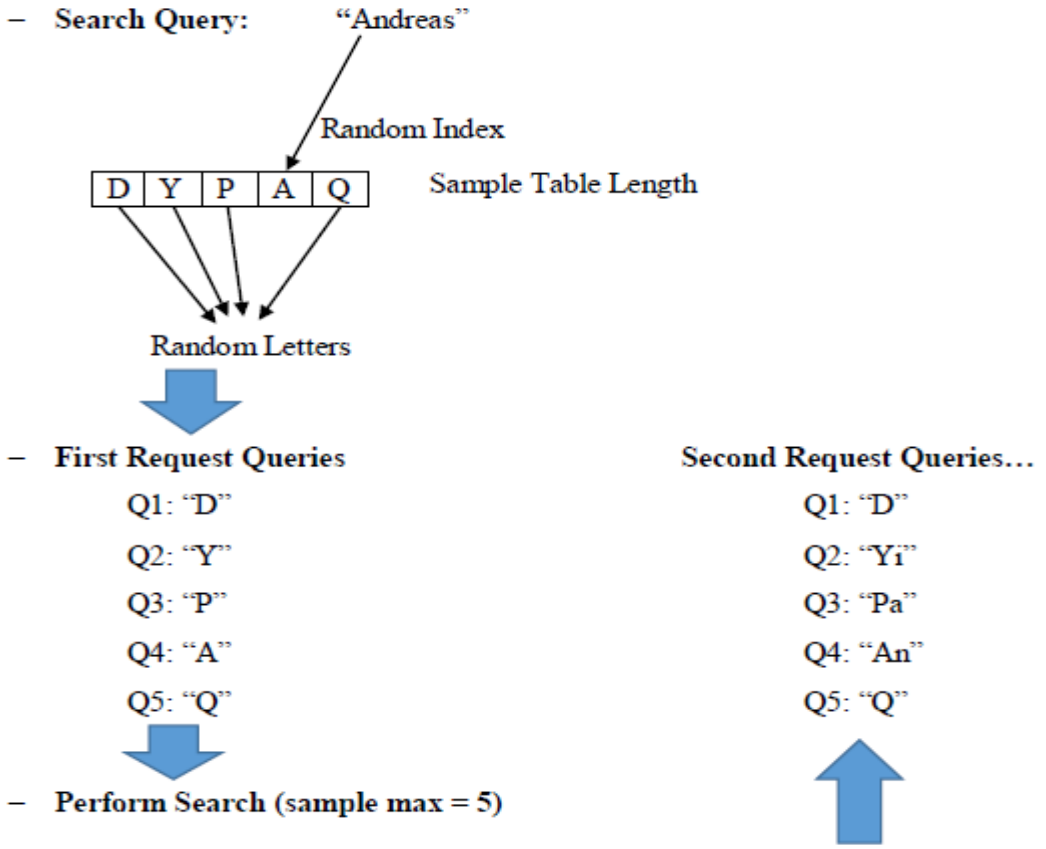
χρήστη για κρυπτογράφηση όλων των δεδομένων. Όπως βλέπουμε και στο σχήμα, πριν να δοθεί οποιαδήποτε πρόσβαση σε χρήστη πάνω στα δεδομένα, ακολουθείται η τυπική διαδικασία ταυτοποίησής του μέσω των διακριτικών του (username, password), και σε περίπτωση αποτυχίας η αίτηση απορρίπτεται. Τέλος, σε αυτό το στάδιο οποιαδήποτε δεδομένα εισαγάγει ο χρήστης δεν είναι ορατά σε κανέναν παρά τον ίδιο που κατέχει το myKey και συνεπώς το προσωπικό του συμμετρικό κλειδί που αποθηκεύεται κρυπτογραφημένο με το myKey.

5.2.4 Αναζήτηση Χρήστη



Σχήμα 5.3: Αναζήτηση χρήστη μέσω μίας ερώτησης, κρύβοντας την αρχική ερώτηση.

Προχωρώντας θα συζητήσουμε για την αναζήτηση, δηλαδή τον τρόπο που μπορεί ο κάθε χρήστης να ανακαλύψει άλλους χρήστες με πιθανό σκοπό να θέλει να στείλει τυχόν δεδομένα σε αυτούς. Στην εφαρμογή αυτή, υλοποιούμε μία τεχνική δική μας τεχνική αναζήτησης η οποία έχει ως απώτερο στόχο την απόκρυψη της ακριβής πρότασης που εισήγαγε ο χρήστης από το νέφος, δηλαδή των σχέσεων μεταξύ χρηστών. Αυτό το πετυχαίνουμε ζητώντας περισσότερα δεδομένα από το νέφος με ψευδοτυχαίο τρόπο. Ο ακριβής αλγόριθμος περιλαμβάνει επαναληπτικές διαδικασίες μέχρι να αποκτήσει το επιθυμητό αποτέλεσμα, ως επίσης και προσαρμοστικές τεχνικές σε κάποιες περιπτώσεις.



Q1	Q2	Q3	Q4	Q5
2 Results	**5 Results**	**5 Results**	**5 Results**	**1 Result**
Diana	Yiannis	Peter	Andreas	Quora
David	Yi	Paul	Anna	
	Yacks	Pamela	Achilleas	
	Yodi	Pola	Aristos	
	Yrick	Pity	Anthi	

Σχήμα 5.4: Παράδειγμα αναζήτησης συγκεκριμένου χρήστη.

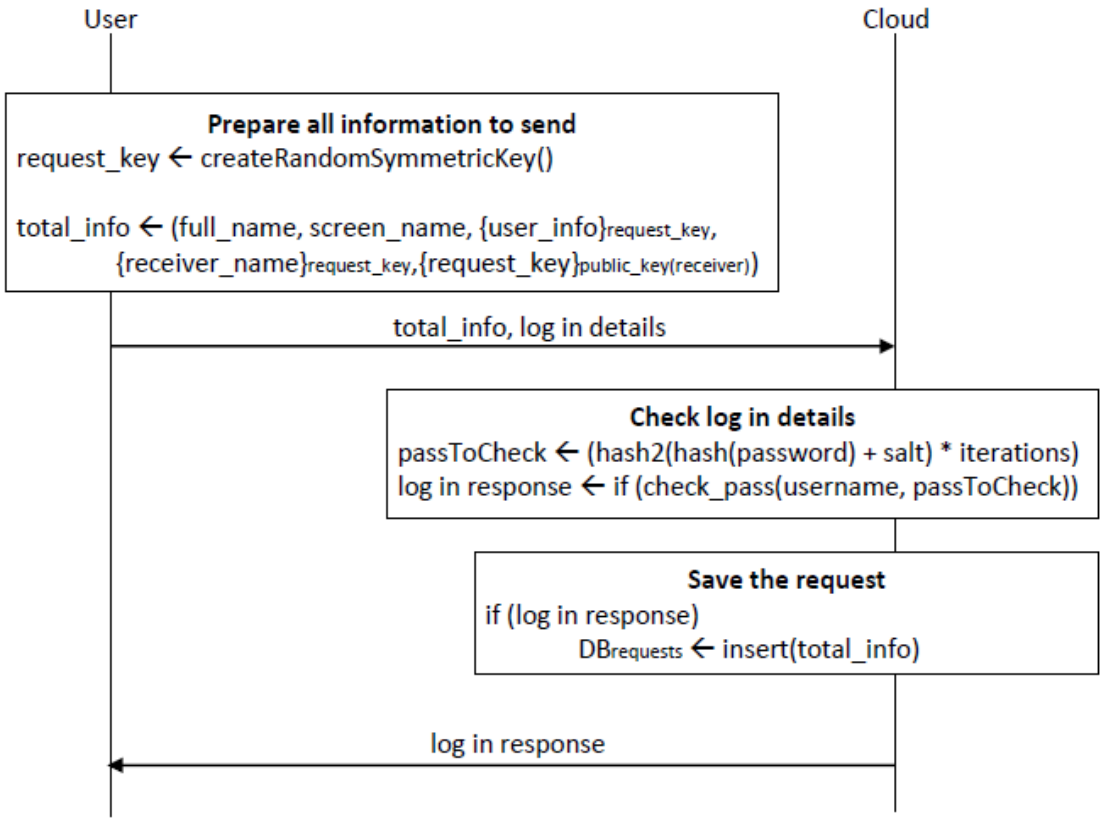
Για να προβούμε σε πιο λεπτομερή εξήγηση, παίρνουμε την πρόταση αναζήτησης του χρήστη και αποκόπτουμε το πρώτο της γράμμα. Στη συνέχεια, τυχαία επιλέγουμε ακόμη έναν προκαθορισμένο αριθμό (τον οποίο εμείς θέτουμε) από άλλα γράμματα και κάνουμε διαφορετικές αναζητήσεις βάσει όλων των γραμμάτων, δηλαδή είναι σαν να δημιουργούμε και άλλες ψεύτικες προτάσεις αναζήτησης. Οι αναζητήσεις που κάνουμε έχουν κάποιο όριο αποτελεσμάτων, για παράδειγμα για κάθε μία από τις προτάσεις που καταλήγουμε μπορούμε να πάρουμε μέχρι έναν άλλο προκαθορισμένο αριθμό αποτελεσμάτων (τον οποίο επίσης εμείς θέτουμε).

Για όσες προτάσεις πάρουμε το όριο αποτελεσμάτων που θέσαμε, τις περιορίζουμε προσθέτοντας περισσότερα γράμματα για να πάρουμε λιγότερα αποτελέσματα. Αν είναι τυχαία πρόταση, τότε επιλέγουμε τυχαίο όνομα από τα αποτελέσματά της και προσθέτουμε γράμματα από εκείνο. Αν ωστόσο η πρόταση αυτή είναι η επιθυμητή μας, προσθέτουμε γράμματα συγκεκριμένα από εκείνη. Όποια περίπτωση και να ισχύει, επαναλαμβάνουμε την αναζήτηση ξανά και ξανά με αυτό τον τρόπο μέχρι να έχουμε για όλες τις προτάσεις (τυχαίες και την αληθινή) αποτελέσματα λιγότερα από το όριο. Με αυτό τον τρόπο, επιτυγχάνουμε να πάρουμε με σιγουριά τα αποτελέσματα της επιθυμητής πρότασης, αφού αν δεν τα πάρουμε λόγω του μέγιστου ορίου, απλά ενισχύουμε συνεχώς την αναζήτηση για να μειώνονται τα αποτελέσματα. Το γεγονός που μας προσφέρει την απόκρυψη της κανονικής αναζήτησης είναι το ότι πράττουμε ακριβώς τις ίδιες ενέργειες και στις ψεύτικες προτάσεις.

Ένα άλλο σενάριο είναι η ύπαρξη πολλών αποτελεσμάτων που ικανοποιούν την επιθυμητή πρόταση, πιθανότατα επειδή είναι πολύ γενική και μικρή. Για παράδειγμα μπορεί να είναι μερικά γράμματα μόνο ή και ένα όνομα πολύ κοινό. Συνεπώς, όταν ανιχνευθεί τέτοια περίπτωση η εφαρμογή αυτόματα απορρίπτει όλες τις τυχαίες προτάσεις που περιγράψαμε προηγουμένως και απλά αιτείται από το νέφος αποτελέσματα αναζήτησης που ικανοποιούν αυτή την γενική πρόταση. Υπάρχει και σε αυτό το σημείο μέγιστος αριθμός αποτελεσμάτων που μπορούμε να ανακτήσουμε, άρα πολύ πιθανόν ο χρήστης που κάνει την αναζήτηση να πρέπει να γίνει πιο συγκεκριμένος αν δεν βρει αυτό που ψάχνει.

Κατά την ανάκτηση αποτελεσμάτων, παίρνουμε για όλους φροντίζουμε να παίρνουμε για όλους ονοματεπώνυμο, ψευδώνυμο και δημόσιο κλειδί για χρήση εάν επιθυμήσουμε να στείλουμε δεδομένα κάποιου από τα αποτελέσματά μας. Είναι σημαντικό να αναφέρουμε ότι για όλα τα αποτελέσματα (ακόμη και τα τυχαία) ανακτούμε τα δημόσια κλειδιά στην προσπάθειά μας να αποκρύψουμε από το νέφος ποιους χρήστες όντως στείλαμε δεδομένα. Το μόνο που χρειαζόμαστε για να στείλουμε κάτι ενός χρήστη είναι το ψευδώνυμό του και το δημόσιο κλειδί του, επομένως με τα αποτελέσματα της αναζήτησης είμαι σε θέση να πραγματοποιήσουμε αιτήματα φιλίας τα οποία θα περιγραφούν στη συνέχεια.

5.2.5 Αποστολή Δεδομένων

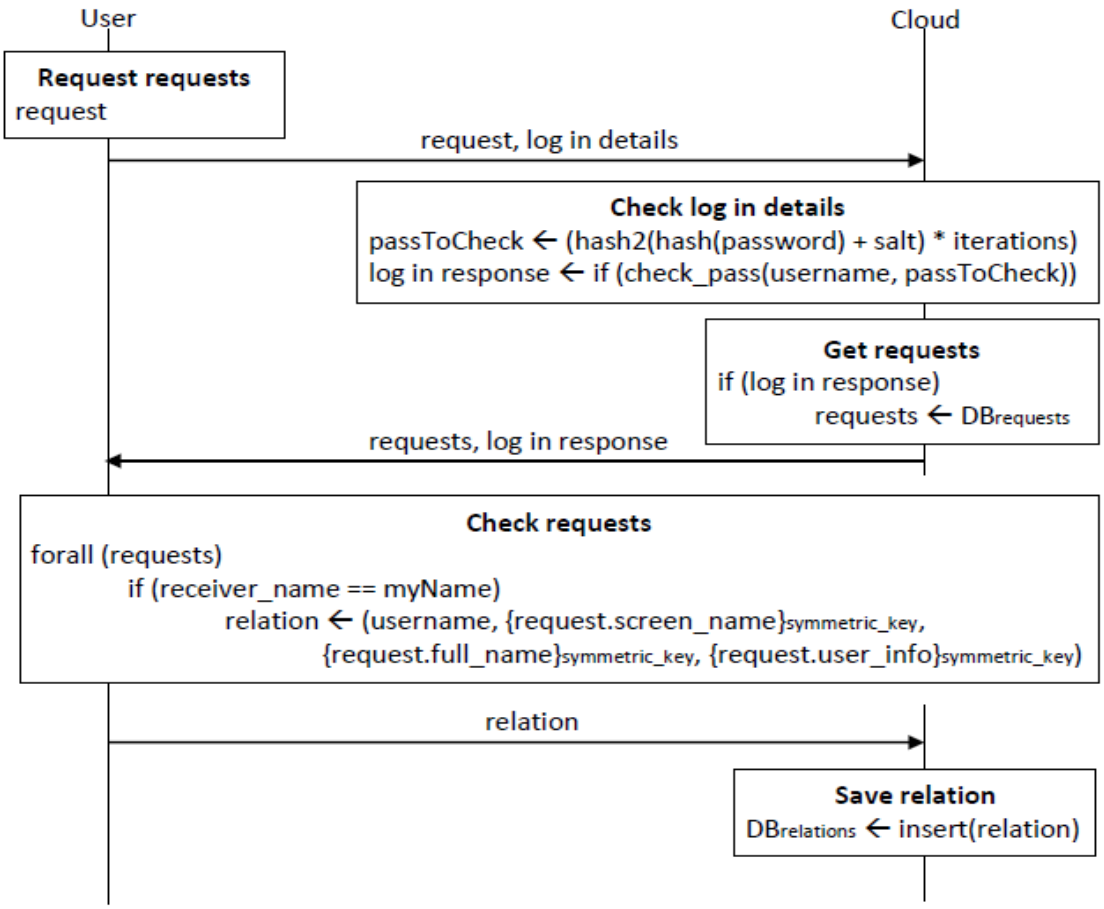


Σχήμα 5.5: Αποστολή δεδομένων προς άλλο συγκεκριμένο χρήστη σαν αίτημα φιλίας.

Αυτό το διάγραμμα παριστάνει την διαδικασία που χρειάζεται να ακολουθηθεί ούτως ώστε να πραγματοποιηθεί ένα «αίτημα φιλίας» από έναν χρήστη προς έναν άλλο. Πιο ουσιαστικά μιλώντας, να στείλει ο χρήστης Α τα προσωπικά δεδομένα που επιθυμεί

στον χρήστη Β. Για να επιτευχθεί αυτό, ο αποστολέας πρέπει να ακολουθήσει κάποια συγκεκριμένη σειρά ενεργειών έτσι ώστε να εξασφαλίσει τόσο την ασφάλεια που απαιτείται, όσο και το επιθυμητό αποτέλεσμα. Αρχικά, για κάθε αίτημα από οποιονδήποτε προς οποιονδήποτε δημιουργείται ένα τυχαίο καινούργιο συμμετρικό κλειδί request_key. Το κλειδί αυτό χρησιμοποιείται για την κρυπτογράφηση όλων των δεδομένων του αποστολέα, όπως και το όνομα του παραλήπτη. Τα στοιχεία (ονοματεπώνυμο, ψευδώνυμο) του αποστολέα στέλνονται σε καθαρό κείμενο. Ακολούθως, αφού τελειώσουμε με τα προηγούμενα βήματα, κρυπτογραφούμε και το ίδιο το request_key με το δημόσιο κλειδί του παραλήπτη. Με αυτόν τον τρόπο μόνο ο χρήστης που έχει πρόσβαση στο σωστό ιδιωτικό κλειδί μπορεί να διαβάσει τα δεδομένα που στάλθηκαν, ούτε το νέφος ούτε κανείς άλλος. Επιπλέον, αποκρύπτουμε και την σχέση μεταξύ χρηστών αφού πάντα μόνο ο αποστολέας είναι φανερός προς το νέφος.

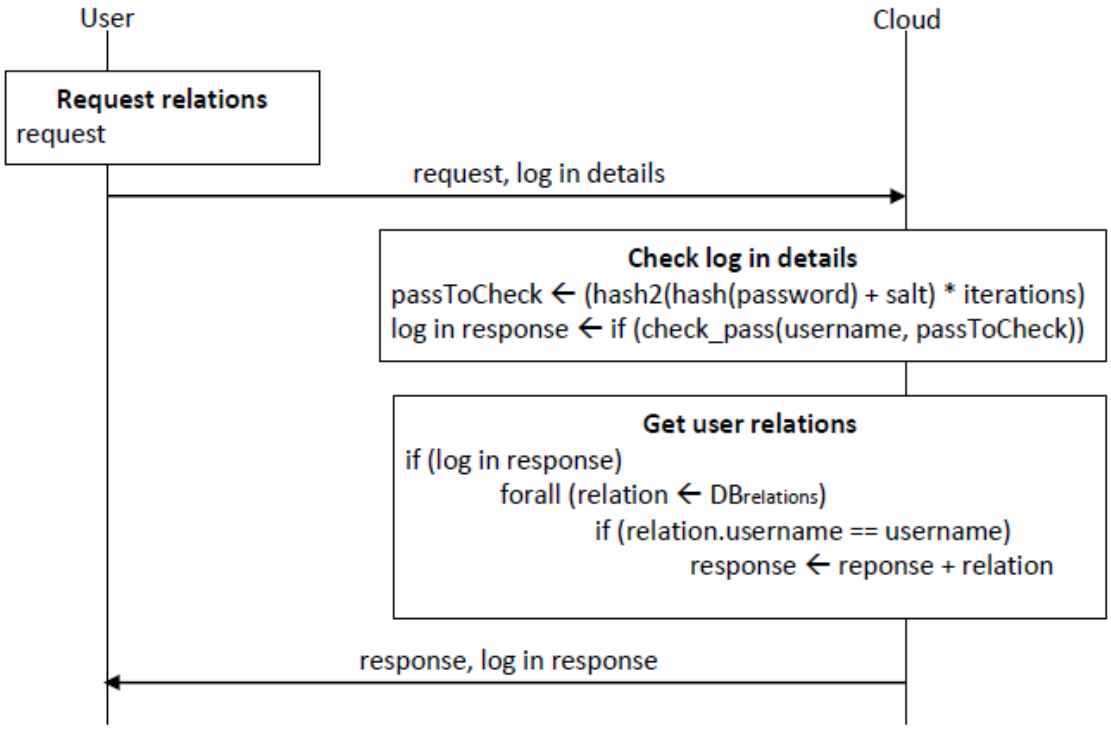
5.2.6 Ανανέωση Φίλων



Σχήμα 5.6: Μέσω των προηγούμενων αιτημάτων ανανεώνονται/προστίθενται «φίλοι».

Ως αποτέλεσμα από την προηγούμενη διαδικασία, στον πίνακα requests της βάσης δεδομένων βρίσκονται αποθηκευμένα όλα τα αιτήματα από χρήστες προς άλλους. Σε αυτό το σημείο θα παρουσιάσουμε τη σειρά ενεργειών που ακολουθεί η εφαρμογή για κάθε χρήστη ούτως ώστε να ανανεώσει την λίστα φίλων του, βάσει των τυχόν καινούργιων αιτήσεων. Σε πρώτο στάδιο ενημερώνει το νέφος ότι έχει αυτόν τον σκοπό, και το νέφος αφού ταυτοποιήσει τον χρήστη του στέλνει όλα τα αιτήματα φιλτράροντάς τα για να είναι σίγουρο ότι δεν πήρε τα ίδια στο παρελθόν. Το νέφος γνωρίζει τα αιτήματα που δεν εξέτασε ο χρήστης στο παρελθόν επειδή αποθηκεύει την χρονική στιγμή που έγινε η τελευταία ανανέωση του κάθε χρήστη· αυτή η τεχνική είναι για καλύτερη απόδοση. Για κάθε αίτημα ο χρήστης αποκρυπτογραφεί το request_key με το ιδιωτικό του κλειδί και στη συνέχεια τον παραλήπτη με το request_key· και αν το αποτέλεσμα είναι ο εαυτός του τότε σημαίνει αυτός είναι ο παραλήπτης αφού το ιδιωτικό του κλειδί είναι το σωστό. Τότε, απλά αποθηκεύει τον αποστολέα ως φίλο μαζί με τα προσωπικά δεδομένα που έλαβε όπως φαίνεται στο σχήμα, τα οποία ήταν επίσης κρυπτογραφημένα με το request_key. Η αποθήκευση αυτή ουσιαστικά δημιουργεί μία νέα εγγραφή στον πίνακα «relations» με ορατό μόνο το όνομα λογαριασμού του χρήστη που την δημιουργεί. Τα υπόλοιπα δεδομένα (αποστολέας αιτήματος και τα προσωπικά του δεδομένα) κρυπτογραφούνται με το προσωπικό συμμετρικό κλειδί του χρήστη αυτού, το οποίο περιγράψαμε νωρίτερα στο έγγραφο. Αυτές οι σχέσεις είναι μονής κατεύθυνσης, δηλαδή κάθε ένας από τους δύο εμπλεκόμενους χρήστες θα πρέπει να καταχωρήσει την δική του εγγραφή για αυτή την «φιλία».

5.2.7 Λήψη Φίλων



Σχήμα 5.7: Λήψη υπαρχόντων φίλων για πρόσβαση στα δεδομένα τους.

Με την ολοκλήρωση της διαδικασίας ανανέωσης των φίλων, ο κάθε χρήστης τώρα μπορεί ανά πάσα στιγμή να αποτίθεται στον πίνακα relations της βάσης δεδομένων και να ανακτά τα δεδομένα των χρηστών που έχει ως φίλους. Όπως φαίνεται στο σχήμα, ο χρήστης απλά λέει στο νέφος να του ανακτήσει τους φίλους του και το νέφος ψάχνει μέσα στον πίνακα αυτό για να βρει όλες τις σχέσεις που περιλαμβάνουν το όνομα λογαριασμού του αιτούμενου χρήστη ως πρώτο μέλος, αφού το δεύτερο μέλος είναι κρυπτογραφημένο.

5.3 Υποδομή Ασφαλείας

Ο κύριος στόχος του συστήματος/εφαρμογής μας είναι η ασφάλεια στο υπολογιστικό νέφος, η οποία αναλύθηκε στο προηγούμενο κεφάλαιο. Το υπολογιστικό νέφος δεν έχει πρόσβαση στα δεδομένα των χρηστών. Ωστόσο, το σύστημά μας επεκτείνεται και στην αντιμετώπιση άλλων επιθέσεων ασφαλείας, είτε έμμεσα με τις ήδη

υπάρχοντες λειτουργίες του είτε άμεσα. Θα αναφέρουμε τις πιο συχνές και κυριότερες από αυτές περιληπτικά, με μικρή περιγραφή και πώς το σύστημά μας τις προσεγγίζει:

- **Αποθήκευση Κρυπτογραφημένου Κειμένου:** Το κρυπτογραφημένο κείμενο αποτελείται από οποιοδήποτε συνδυασμό bits/bytes, γεγονός που το κάνει δύσκολο στην επεξεργασία, αφού αυτά τα bits/bytes ενδέχεται να είναι χαρακτήρες που να μπορούν να βλάψουν το σύστημα σε συγκεκριμένα σημεία. Γι' αυτό το λόγο, η εφαρμογή μας μεταφράζει κάθε κρυπτογραφημένη πληροφορία στην Base64 κωδικοποίηση για ευκολία χρήστης. Αυτή η κωδικοποίηση χρησιμοποιεί ανά 4 τα bits και τα μεταφράζει σε συγκεκριμένους χαρακτήρες, κυρίως γράμματα και αριθμούς.
- **Παραβίαση Βάσης Δεδομένων:** Η βάση δεδομένων της εφαρμογής μας βρίσκεται αποθηκευμένη στο νέφος, τυχόν παραβίασή της από κακόβουλα άτομα είναι απίθανο να οδηγήσει σε έκθεση ή καταστροφή των δεδομένων μας. Αυτό είναι για διάφορους λόγους, όπως του γεγονότος ότι τα πάντα είναι κρυπτογραφημένα. Επίσης, οι κωδικοί των χρηστών αποθηκεύονται με την ισχυρή τεχνική που περιγράψαμε προηγουμένως που τους καθιστά πρακτικά άχρηστους σε όποιον αποκτήσει πρόσβαση σε αυτούς.
- **Επιθέσεις Ωμής Βίας:** Θεωρητικά αυτές οι επιθέσεις δεδομένου άπειρου χρόνου είναι σίγουρες, αλλά στις πλείστες περιπτώσεις ο απαιτούμενος χρόνος τις καθιστά άχρηστες. Επιπλέον, η εφαρμογή μας χρησιμοποιεί κορυφαίους αλγόριθμους σε κάθε περίπτωση, γεγονός που συμβάλλει κατά πολύ στην άμυνά μας.
- **Man-in-the-Middle:** Αυτές οι επιθέσεις έχουν να κάνουν με τους κακόβουλους να αποκτούν πρόσβαση σε δεδομένα που στέλνονται μέσα στο δίκτυο και να τα χρησιμοποιούν είτε ενεργητικά είτε σε μεταγενέστερο στάδιο. Τα δεδομένα που ανταλλάσσονται με την εφαρμογή μας είναι κρυπτογραφημένα, ωστόσο ο κακόβουλος αποκτά πρόσβαση σε κρυπτογραφημένα μόνο. Μπορεί να προκαλέσει βλάβη με το να καθυστερήσει, διαγράψει τα μηνύματα, αλλά ποτέ δεν θα αποκτήσει πρόσβαση σε δεδομένα. Συνήθως τα πιο ευάλωτα σημεία ενός συστήματος είναι κατά την αρχικοποίηση, σε αυτή την περίπτωση το Sign Up. Ο κωδικός όμως δεν μεταδίδεται ποτέ σε καθαρή μορφή, άρα χωρίς αυτόν δεν

υπάρχει το myKey και χωρίς το myKey δεν υπάρχει εύκολη πρόσβαση πουθενά. Το συγκεκριμένο είδος επιθέσεων θεωρείται εντελώς λυμένο με χρήση HTTPS.

- **Buffer Overflow:** Αυτή είναι μία κατηγορία επιθέσεων και όχι μία μοναδική επίθεση, αφού υπάρχουν πολλοί πιο συγκεκριμένοι κληρονόμοι της. Ωστόσο, σε γενικές γραμμές ο στόχος τέτοιων επιθέσεων είναι η κακόβουλη εισαγωγή κώδικα μέσω των μηχανισμών που δύναται ο χρήστης να εισαγάγει δεδομένα στο σύστημα. Δηλαδή, βασίζονται στην υπερχείλιση των buffers αυτών με στόχο τα υπερχειλισμένα δεδομένα τους να εισχωρήσουν μέσα στην περιοχή μνήμης του προγράμματος με αποτέλεσμα να υπάρχει ενδεχόμενο να εκτελεστούν ως κανονικός κώδικας. Εμείς αντιμετωπίζουμε με δύο τρόπους τέτοιες επιθέσεις. Αρχικά, ελέγχεται το μέγεθος των δεδομένων που μπορεί να εισαχθούν σε κάθε σημείο της εφαρμογής, και η ίδια η Java εφαρμόζει δικούς της μηχανισμούς για αυτό το σκοπό.
- **SQL Injection:** Αυτό το είδος επιθέσεων επίσης έχει να κάνει με κακόβουλη είσοδο δεδομένων μέσω της εφαρμογής, στοχεύοντας τα SQL statements που χρησιμοποιούνται. Για παράδειγμα, σε πεδίο που είναι προορισμένο για όνομα λογαριασμού να εισαχθεί κώδικας SQL και όταν χρησιμοποιηθεί το string αυτό αναλλοίωτο με concatenation στο SQL statement θα επηρεάσει το αποτέλεσμα αν είναι σωστός κώδικας. Για αυτό το πρόβλημα υπάρχουν διάφορα μέτρα που μπορούν να ληφθούν, αρχικά μπορεί να γίνει χρήση prepared statements, ή φιλτράρισμα των σημείων εισόδου δεδομένων. Στην περίπτωσή μας εφαρμόζεται φιλτράρισμα στο όνομα λογαριασμού και στο ψευδώνυμο των χρηστών. Σε όλα τα άλλα δεδομένα δεν είναι απαραίτητο αφού δεν γίνεται χρήση του καθαρού κειμένου, άρα κατά την κρυπτογράφηση όλα αλλάζουν.

Κεφάλαιο 6

Συμπεράσματα

Μετά από την ολοκλήρωση της περιγραφής του συστήματος, θα προχωρήσουμε σε γενική συζήτηση γύρω από αυτό και γενικότερα γύρω από αυτή την εργασία.

6.1 Αποτέλεσμα και Δοκιμή

Σε αυτό το σημείο έχουμε καταλήξει σε μία εφαρμογή η οποία πραγματοποιεί ασφαλή ανταλλαγή δεδομένων μεταξύ των χρηστών μέσα στο υπολογιστικό νέφος με τον τρόπο που είδαμε νωρίτερα, αποκρύπτοντας τα δεδομένα όπως επίσης και τους εμπλεκόμενους χρήστες. Εδώ θα συζητήσουμε κάποια συμπεράσματα μετά από τη δοκιμή που πραγματοποιήθηκε.

Η δοκιμή που εφαρμόσαμε εμείς σε αυτό το σύστημα περιλάμβανε μικρό αριθμό από χρήστες. Ο καθένας τους ανέλαβε να δημιουργήσει τον λογαριασμό του και να προσθέσει προσωπικά δεδομένα. Στη συνέχεια, τυχαία έστειλαν τα δεδομένα τους σε άλλους χρήστες. Τέλος, όλοι οι παραλήπτες έλαβαν με επιτυχία τα δεδομένα και είχαν πρόσβαση πάνω τους όπως θα έπρεπε, και επίσης όλες οι λειτουργίες ολοκληρώθηκαν σε μηδαμινό χρόνο. Επιπλέον, ελέγχθηκε η βάση δεδομένων και επιβεβαιώθηκε ότι τα πάντα είναι κρυπτογραφημένα, ως επίσης και τα δεδομένα τυχόν αναζητήσεων ήταν τυχαία όπως θα έπρεπε.

6.2 Αξιολόγηση

Ακολούθως, θα αναφερθούμε σε κάποια χαρακτηριστικά της εφαρμογής όσον αφορά το πώς πετυχαίνει το στόχο της, αλλά και την πολυπλοκότητα των λειτουργιών της. Κατ' αρχήν, το σύστημά μας επιτυγχάνει ασφάλεια στο υπολογιστικό νέφος, με την προσέγγιση μετάφρασης κλειδιών. Αυτή η προσέγγιση όπως είπαμε και πριν, περιορίζει τις ενέργειες που μπορεί να κάνει το υπολογιστικό νέφος για τον χρήστη. Πιο συγκεκριμένα, το νέφος δεν είναι σε θέση να πράξει οποιονδήποτε υπολογισμό πάνω στα δεδομένα επειδή είναι κρυπτογραφημένα. Επομένως, τα συστήματα αυτής της προσέγγισης περιορίζονται στο να χρησιμοποιούν το νέφος για να βοηθά στην ορθή ροή των δεδομένων. Στην περίπτωσή μας, το νέφος πραγματοποιεί έλεγχο της ροής δεδομένων μέσω του ονόματος λογαριασμού και του κωδικού του κάθε χρήστη, ούτως ώστε λάθος δεδομένα να μην είναι προσβάσιμα από μη-αρμόδιους χρήστες. Επίσης, δεν επιτρέπει σε χρήστες να πράξουν ως άλλοι χρήστες με τον έλεγχο των στοιχείων σε κάθε συναλλαγή.

Ωστόσο, το επίπεδο ασφάλειας που πετυχαίνει το σύστημά μας σε γενικές γραμμές έχει κάποια δικά του θέματα. Συνήθως, τέτοιες προσεγγίσεις συνοδεύονται με κόστη πάνω στην απόδοση λόγω περιττών ενεργειών. Συγκεκριμένα, η εφαρμογή μας αποθηκεύει πολλές φορές τα ίδια δεδομένα στην βάση δεδομένων, μέσω του τρόπου που πραγματοποιεί τα αιτήματα φιλίας. Επίσης, κάνει παραπάνω αιτήσεις προς την βάση δεδομένων κατά τη διάρκεια των λειτουργιών αναζήτησης χρήστη και ανανέωσης φίλων. Τέλος, κάνει και επιπλέον υπολογισμούς, μέσω των πολλαπλών αποκρυπτογραφήσεων, ειδικά με ιδιωτικά κλειδιά κατά την ανανέωση φίλων. Όλα αυτά αναμένεται να έχουν ως αποτέλεσμα επιπτώσεις στην απόδοση, τουλάχιστον όταν έχουμε να κάνουμε με μεγάλο αριθμό χρηστών.

Πέραν αυτών, στις υπόλοιπες λειτουργίες της, η εφαρμογή είναι γρήγορη αφού χειρίζεται σχετικά μικρό αριθμό δεδομένων στο παρόν στάδιο. Φυσικά, αυτό θα μπορούσε να αλλάξει δεδομένης νέας δυνατότητας για χειρισμό πολλαπλών ειδών αρχείων.

6.3 Αποκομίσεις

Κατά τη διάρκεια αυτής της διπλωματικής εργασίας αποκόμισα διάφορα πράγματα και απέκτησα ένα είδος εμπειρίας που είναι κατά κάποιον τρόπο μοναδικό στη διάρκεια της πορείας μου στο Πανεπιστήμιο Κύπρου. Αυτό είναι εξαιτίας του γεγονότος ότι η εργασία αυτή είναι ατομική, απαιτεί πολλή δουλειά, και τέλος, συνήθως είναι πιο ολοκληρωμένη από άλλες προηγούμενες εργασίες. Επομένως, στη δική μου περίπτωση εξοικειώθηκα και με ερευνητικό κομμάτι, και με δημιουργία δικής μου ιδέας, και με υλοποίησή της. Ακόμα, κατανόησα σε πολύ βάθος το υπολογιστικό νέφος και πολλά από τα θέματα που το αφορούν, με κυριότερο την ασφάλεια. Το γεγονός ότι τα πάντα εξαρτούνται από δική μου προσπάθεια και οργάνωση με καθιστά πιο έτοιμο για να ανταπεξέλθω σε μελλοντικές προκλήσεις.

Συγκεκριμένες δεξιότητες που εξάσκησα, είναι το επίπεδό μου όσον αφορά το λειτουργικό Android και τη γλώσσα προγραμματισμού Java, αφού όσο περισσότερο χρόνο αφιερώνει κάποιος χρησιμοποιώντας ένα εργαλείο, τόσο καλύτερος γίνεται με αυτό. Άλλο εργαλείο ήταν η server-side scripting γλώσσα PHP, μέσω της οποίας έμαθα να χειρίζομαι δεδομένα από την πλευρά του εξυπηρετητή και να επικοινωνώ με τον χρήστη μέσω του διαδικτύου, διαδικασία με την οποία ήμουν εντελώς άπειρος.

Επίσης, απέκτησα ακόμη περισσότερη εμπειρία πάνω σε θέματα ασφαλείας και αλγόριθμους κρυπτογράφησης. Τέτοιες ικανότητες είναι πολύ σημαντικές στις μέρες μας, με το διαδίκτυο να είναι σε τέτοια επίπεδα σε όλο τον κόσμο. Δυστυχώς, οι ρυθμοί με τους οποίους αναπτύσσεται η τεχνολογία δεν βγαίνουν πάντα σε καλό.

6.4 Μελλοντική Εργασία και Προοπτικές

Η δυνατότητα για ανταλλαγή/αποθήκευση δεδομένων που παρέχει η εφαρμογή μου στους χρήστες με απόκρυψη των σχέσεων μεταξύ τους είναι μια λειτουργία που θα μπορούσε να χρησιμοποιηθεί από πολλά είδη εφαρμογών που θέλουν να πετύχουν αυτό

το είδος ασφάλειας στο υπολογιστικό νέφος. Εγώ προσωπικά την θεωρώ ως ένα σκελετό ικανό να φανεί χρήσιμος σε διάφορα σενάρια.

Πιθανή επέκταση αυτής της ιδέας θα ήταν η δημιουργία μίας ιστοσελίδας η οποία να συνάπτει με την Android εφαρμογή και οι χρήστες να δύνανται να αλληλοεπιδρούν ανεξαρτήτως της πλατφόρμας στην οποία έχουν εγκατεστημένο το σύστημα. Επίσης, θα μπορούσε να δοθεί έμφαση στην αποδοτικότητα της εφαρμογής για όταν έχει να κάνει με μεγάλο αριθμό χρηστών. Με αυτό εννοώ τους τρόπους που αποθηκεύονται δεδομένα στη βάση δεδομένων στο υπολογιστικό νέφος, και αλγόριθμους που θα μπορούσαν να επιταχύνουν κάποιες διαδικασίες όπως για παράδειγμα, η ανανέωση φίλων.

Βιβλιογραφία

[1] M. Arapinis et al, “Privacy supporting cloud computing: Confichair, a case study,” in *Principles of Security and Trust: Proceedings of the First International Conference, POST 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012*, P. Degano, J. D. Guttman, Eds. Berlin: Springer, 2012. pp. 89-108.

[2] M. Armbrust et al, “A view of cloud computing,” *Communications of the ACM*, vol 53, no. 4, pp. 50-58, April 2010.

[3] S. Carlin, and K. Curran, “Cloud computing security,” *International Journal of Ambient Computing and Intelligence*, vol 3, no. 1, pp. 14-19, January 2011.

[4] Cloud Security Alliance, “Top threats to cloud computing v1. 0,” *Cloud Security Alliance*, March 2010. [Online]. Available: https://cloudsecurityalliance.org. [Accessed: May 20, 2015].

[5] P. M. Mell and T. Grance, “The NIST definition of cloud computing,” National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep. SP 800-145, September 28, 2011.

[6] M. D. Ryan, “Cloud computing security: The scientific challenge, and a survey of solutions,” *Journal of Systems and Software*, vol 86, no. 9, pp. 2263-2268, September 2013.

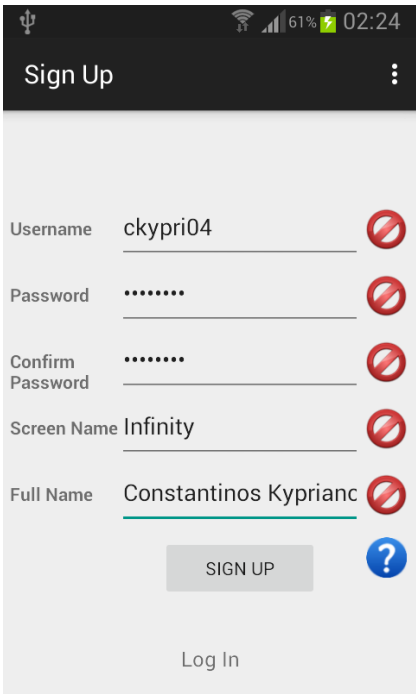
[7] Top Threats Working Group, "The notorious nine: Cloud computing top threats in 2013," *Cloud Security Alliance*, February 2013. [Online]. Available: https://cloudsecurityalliance.org. [Accessed: May 20, 2015].

[8] M. van Dijk et al, "Fully homomorphic encryption over the integers," in *Advances in cryptology–EUROCRYPT 2010: Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010*, H. Gilbert, Eds. Berlin: Springer, 2010. pp. 24-43.

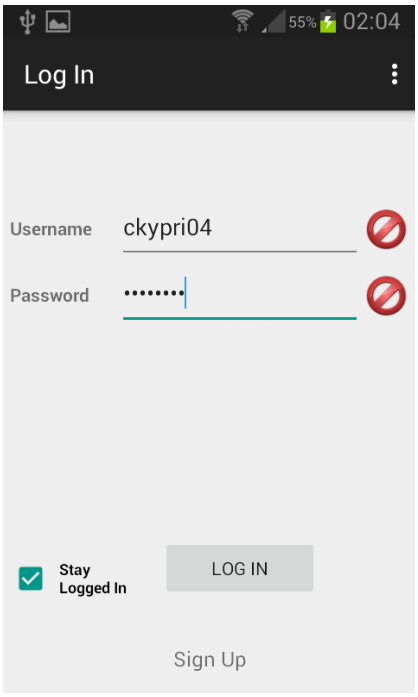
[9] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol 28, no. 3, pp. 583-592, March 2012.

Παράρτημα Α

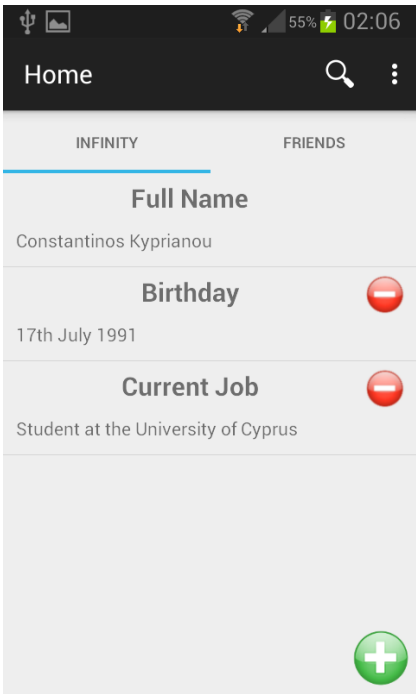
Δημιουργία Λογαριασμού



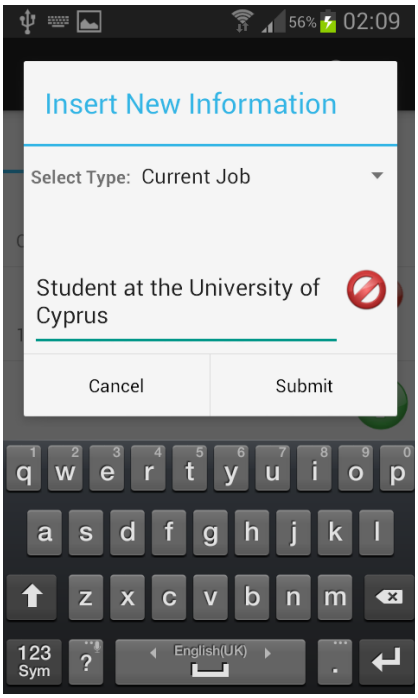
Σύνδεση Χρήστη



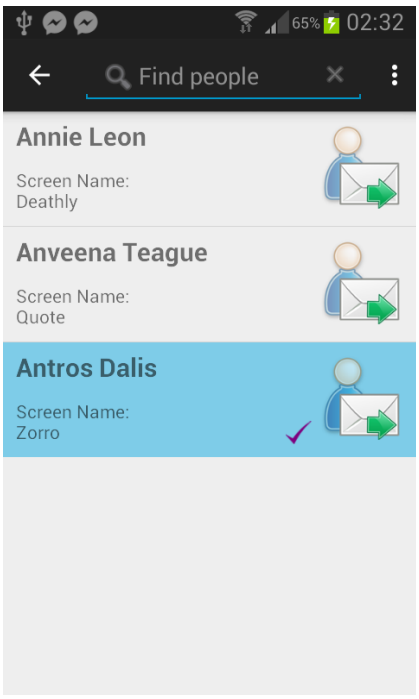
Προσωπική Σελίδα



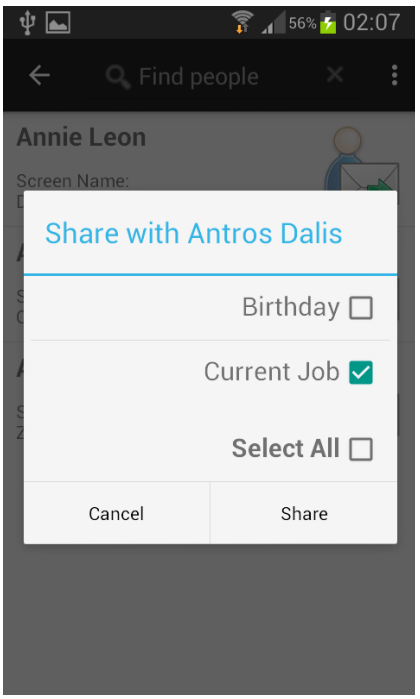
Πρόσθεση Δεδομένων



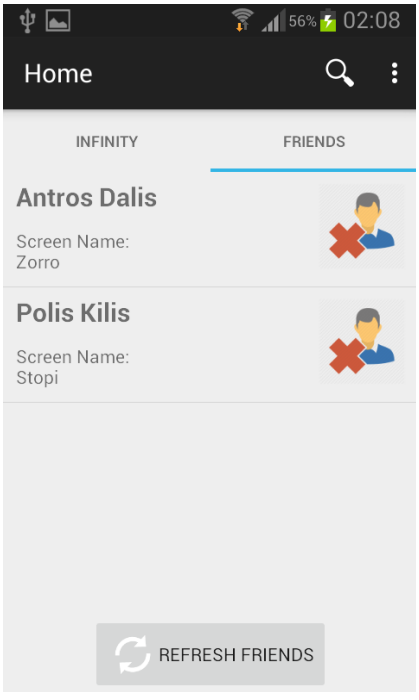
Αναζήτηση Χρηστών



Αποστολή Δεδομένων



Φίλοι



Δεδομένα Φίλου