Ατομική Διπλωματική Εργασία

Πρακτικός έλεγχος ταυτότητας χωρίς κωδικό πρόσβασης για κινητές συσκευές

Χαρίλαος Μικέλλη

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρακτικός έλεγχος ταυτότητας χωρίς κωδικό πρόσβασης για κινητές συσκευές

Χαρίλαος Μικέλλη

Επιβλέπων Καθηγητής
Ηλίας Αθανασόπουλος

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

Μάιος 2022

Ευχαριστίες

Δράττοντας την ευκαιρία, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή διπλωματικής μου εργασίας, Δρ Ηλία Αθανασόπουλο, του οποίου η τεχνογνωσία ήταν πολύτιμη στην καθοδήγηση μου καθ' όλη τη διάρκεια της διατριβής μου.

Επίσης, θα ήθελα να ευχαριστήσω τους φίλους και την οικογένεια μου, για όλη την υποστήριξη που μου προσφέρανε και για την συνεχή ενθάρρυνση που μου έδιναν κατά τη διάρκεια της διατριβής μου.

Περίληψη

Οι κωδικοί πρόσβασης αποτελούν έναν απλό και παραδοσιακό τρόπο σύνδεσης σε διαδικτυακές υπηρεσίες και συστήματα. Τα προβλήματα ασφάλειας και χρηστικότητας που προκύπτουν χρησιμοποιώντας κωδικούς πρόσβασης καταστούν επιτακτική ανάγκη την εύρεση εναλλακτικών βιώσιμων, εύχρηστων και ασφαλέστερων λύσεων αυθεντικοποίησης. Τα φυσικά βιομετρικά στοιχεία των ανθρώπων θεωρούνται μια απλή και έμπιστη πηγή προσδιορισμού μοναδικότητας ενός ατόμου με αποτέλεσμα τα τελευταία χρόνια να τραβούν όλο και περισσότερο την προσοχή και να γίνεται έτσι πιο εντατική η χρήση τους ακόμη και η ενσωμάτωση τους στην διαδικασία ελέγχου ταυτότητας ενός χρήστη. Αυτή η εργασία επικεντρώνεται στο πως μπορεί μια διαδικτυακή υπηρεσία να εκτελέσει έλεγχο ταυτότητας των χρηστών της μέσω βιομετρικών στοιχείων, για να συνδέονται οι χρήστες με ευκολία και ασφάλεια σε ένα σύστημα ή υπηρεσία. Έπειτα θα αναδείξουμε πως μπορεί ο χρήστης να χρησιμοποιεί πολλαπλές συσκευές για την σύνδεση του στον διακομιστή μέσω του δακτυλικού του αποτυπώματος και τι επιπτώσεις επιφέρει αυτή η δυνατότητα στην ασφάλεια της όλης διαδικασίας αυθεντικοποίησης.

Περιεχόμενα

Κεφάλαιο 1

1 Εισαγωγή

Ανέκαθεν οι κωδικοί πρόσβασης ήταν η πιο δημοφιλής μορφή σύνδεσης στο διαδίκτυο και γενικότερα. Ωστόσο, οι κωδικοί πρόσβασης δεν θεωρούνται ασφαλείς εδώ και πολύ καιρό. Συνοδεύονται από σημαντικά προβλήματα ασφάλειας και χρηστικότητας [1, 2] ενώ παράλληλα οι χρήστες συχνά επαναχρησιμοποιούν έναν κωδικό πρόσβασης για πολλαπλούς λογαριασμούς ή χρησιμοποιούν απλούς και προβλέψιμους κωδικούς πρόσβασης.

Ακόμα και στην καλύτερη περίπτωση που οι χρήστες δημιουργούν δυνατούς και μη-προβλέψιμους κωδικούς πρόσβασης, πιθανώς να είναι δύσκολο να τους θυμούνται, να τους απομνημονεύουν αλλά και πολλές φορές να τους συγχύζουν μεταξύ τους. Επίσης θεωρούνται ο νούμερο ένα στόχος των εγκληματιών στον κυβερνοχώρο τέτοιοι ώστε που το 81% των παραβιάσεων αφορούν αδύναμους ή κλεμμένους κωδικούς πρόσβασης.

Έχουν προταθεί διάφορες μέθοδοι για τη βελτίωση των κωδικών πρόσβασης. Ο έλεγχος ταυτότητας δύο παραγόντων (2FA) είναι μία από τις πιο κοινές βελτιώσεις που παρέχονται από εμπορικές υπηρεσίες [3, 4, 5, 6]. Αυτή η τεχνολογία απαιτεί να χρησιμοποιήσουμε "κάτι που γνωρίζουμε" όπως ένα PIN, "κάτι που έχουμε" όπως μια συσκευή, ή ακόμη και "κάτι που είμαστε" όπως για παράδειγμα τα δακτυλικά μας αποτυπώματα.

Παρόλο που το 2FA έχει βελτιώσει την ασφάλεια των κωδικών πρόσβασης, απαιτεί τη χρήση πρόσθετων συσκευών, μηνυμάτων SMS ή διευθύνσεων ηλεκτρονικού ταχυδρομείου πέρα από την εισαγωγή του κωδικού πρόσβασης, καθιστώντας έτσι δυσκολότερη τη πρόσβαση από έναν απλό κωδικό πρόσβασης.

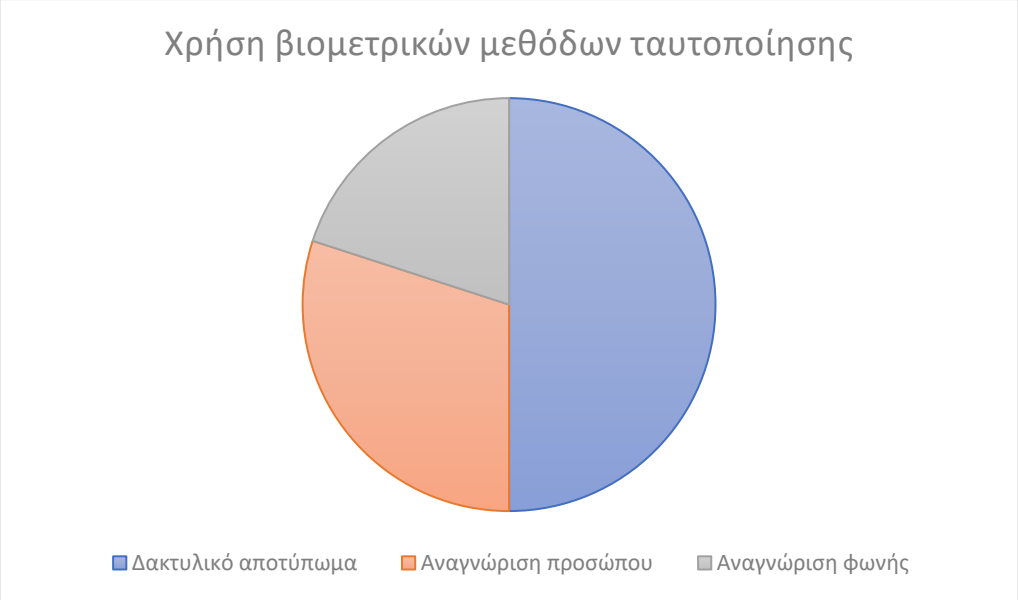
Η passwordless αυθεντικοποίηση είναι ένα μέσο επαλήθευσης της ταυτότητας ενός χρήστη, χωρίς τη χρήση κωδικού πρόσβασης. Αντ' αυτού, η passwordless αυθεντικοποίηση χρησιμοποιεί πιο ασφαλείς εναλλακτικές λύσεις, όπως είναι τα βιομετρικά στοιχεία (δακτυλικά αποτυπώματα, σαρώσεις αμφιβληστροειδούς) ή άλλοι παράγοντες κατοχής (κωδικοί μιας χρήσης (ΚΜΧ), ήδη καταχωρημένες συσκευές) για να ταυτοποιήσει ένα χρήστη.

Τρεις τρόποι με τους οποίους μπορούμε να επιτύχουμε έλεγχο ταυτότητας χωρίς κωδικό πρόσβασης είναι μέσω βιομετρικών στοιχείων (φυσικά χαρακτηριστικά που προσδιορίζουν μοναδικά ένα άτομο π.χ. δακτυλικό αποτύπωμα), παράγοντες κατοχής (αυθεντικοποίηση μέσω κάποιου αντικειμένου που κατέχει ο χρήστης π.χ. κινητό, ΚΜΧ) και μαγικούς συνδέσμους (αποστολή ενός συνδέσμου στο λογαριασμός ηλεκτρονικού ταχυδρομείου του χρήστη για απευθείας σύνδεση) [7]. Στην παρούσα εργασία θα επικεντρωθούμε στην αυθεντικοποίηση του χρήστη με βιομετρικά στοιχεία και συγκεκριμένα με δακτυλικό αποτύπωμα. Ένα βιομετρικό είναι ένα μοναδικό βιολογικό χαρακτηριστικό που μπορεί να χρησιμοποιηθεί για τον εντοπισμό και την επαλήθευση της ταυτότητας ενός ατόμου [8].

Από πολλαπλές έρευνες που έχουν γίνει κατά διάφορα χρονικά διαστήματα μπορούμε να δούμε πως οι περισσότεροι άνθρωποι έχουν κινητές συσκευές και συνδέονται καθημερινά σε διάφορες διαδικτυακές υπηρεσίες [9]. Είναι σημαντικό να τονιστεί ότι πολλοί χρήστες εξακολουθούν να συνδέονται σε διάφορες υπηρεσίες τουλάχιστο μια φορά την ημέρα χρησιμοποιώντας κωδικούς πρόσβασης. Επίσης, μπορούμε να παρατηρήσουμε ότι υπάρχει μια σταδιακή αύξηση στους χρήστες κινητών τηλεφώνων που συνδέονται με βιομετρικά στοιχεία (π.χ. δακτυλικό αποτύπωμα) έναντι κωδικών πρόσβασης. Ειδικότερα το 2019, 27% των ανθρώπων συνδέονταν σε υπηρεσίες χρησιμοποιώντας βιομετρικά τους στοιχεία, ενώ σε διάστημα ενός χρόνου αυξήθηκε στο 41% (δηλαδή περίπου οι μισοί χρήστες έξυπνων κινητών τηλεφώνων) και αναμένετε μέχρι το 2024 να γίνει ακόμη πιο εντατική η χρήση τους και να αυξηθεί σε ποσοστό 66% των χρηστών [10]. Πιο κάτω βλέπουμε ένα συγκεντρωτικό πίνακα με τις μετρήσεις διαφόρων ερευνών που αφορούν τα τελευταία έτη.

Άτομα που κατέχουν κινητό τηλέφωνο	91.54%
Άτομα που κατέχουν έξυπνο κινητό τηλέφωνο	83.72%
Χρήστες έξυπνων κινητών τηλεφώνων που χρησιμοποιούν βιομετρικά στοιχεία για έλεγχο ταυτότητας	41%
Συχνότητα χρήσης κωδικού πρόσβασης για σύνδεση σε μια υπηρεσία.	Τουλάχιστο μια φορά την μέρα

Παρόλο που η χρήση δακτυλικών αποτυπωμάτων εξακολουθεί να είναι η πιο δημοφιλής μέθοδος αναγνώρισης, ένας αυξανόμενος αριθμός χρηστών χρησιμοποιούν αναγνώριση προσώπου και φωνής [9].



Μπορούμε επίσης να δούμε ότι τα βιομετρικά στοιχεία, ιδιαίτερα εκείνα που σχετίζονται με δακτυλικά αποτυπώματα ή το πρόσωπο, είναι πλέον κοινά σε πολλές κινητές συσκευές και χρησιμοποιούνται από πολλούς χρήστες κυρίως για να ξεκλειδώνουν οθόνες. Η χρήση δακτυλικών αποτυπωμάτων έχει αυξήσει τον ρυθμό με τον οποίο οι χρήστες ξεκλειδώνουν τα κινητά τους [11] και η χρήση των φυσικών βιομετρικών πληροφοριών του χρήστη γίνεται πλέον αποδεκτή ως εναλλακτική λύση αντί της εισαγωγής κωδικών πρόσβασης, αριθμών PIN ή μοτίβων.

Παρόλο που αρχικά η χρήση των βιομετρικών στοιχείων ενός χρήστη θεωρούνταν επικίνδυνη λόγω της παγκόσμιας μοναδικότητας τους, [12] ο τρόπος που

χρησιμοποιούνται και αποθηκεύονται κάνει την χρήση τους αρκετά ασφαλή για τον χρήστη. Τα βιομετρικά στοιχεία θεωρούνται πλέον ως μια πολύ ασφαλής λύση, επειδή ο τρόπος αποθήκευσης βιομετρικών δεδομένων είναι διαφορετικός από τον τρόπο αποθήκευσης των PINs και των κωδικών πρόσβασης. Ενώ οι κωδικοί πρόσβασης αποθηκεύονται στο νέφος (cloud), τα δεδομένα από το δακτυλικό σας αποτύπωμα αποθηκεύονται με ασφάλεια αποκλειστικά στις συσκευές μας. Οι διακομιστές και οι εφαρμογές δεν έχουν ποτέ πρόσβαση στα δεδομένα των δακτυλικών αποτυπωμάτων μας, ούτε αποθηκεύονται στο νέφος [8].

Πλέον πολλές εφαρμογές και συστήματα υιοθετούν αυτή την εναλλακτική μορφή αυθεντικοποίησης για να ταυτοποιήσουν ένα χρήστη και έτσι να καθιστούν ευκολότερη και ασφαλέστερη την διαδικασία σύνδεσης σε μια εφαρμογή ή σύστημα που τρέχει στη συσκευή του χρήστη. Κατ' επέκταση των τοπικών εφαρμογών έχουν γίνει αρκετές προσπάθειες και αναπτύχθηκαν τεχνολογίες για την χρήση των βιομετρικών στοιχείων ακόμη και για διαδικτυακές εφαρμογές και συστήματα χωρίς να εκτίθενται τα βιομετρικά δεδομένα εκτός της κινητής συσκευής.

Η τεχνολογία που έχει προσελκύσει την προσοχή τα τελευταία έτη για έλεγχο ταυτότητας χωρίς κωδικό πρόσβασης είναι το Web Authentication API (WebAuthn) [13], που είναι μία προδιαγραφή του World Wide Web Consortium (W3C). Όταν οι χρήστες συνδέονται σε ένα ιστότοπο, το WebAuthn τους επιτρέπει να επιλέξουν από μια ποικιλία μεθόδων ελέγχου ταυτότητας. Μεταξύ αυτών των μεθόδων ελέγχου ταυτότητας, βιομετρικά στοιχεία όπως τα δακτυλικά αποτυπώματα θεωρούνται όλο και περισσότερο βιώσιμες εναλλακτικές λύσεις για τους κωδικούς πρόσβασης. Η τεχνολογία αυτή θα αναλυθεί περισσότερο σε επόμενη ενότητα.

Σε αυτή την εργασία, δείχνουμε πως μπορεί να γίνει αυθεντικοποίηση ενός χρήστη και η σύνδεση του σε ένα διακομιστή με ασφάλεια χρησιμοποιώντας βιομετρικά στοιχεία και συγκεκριμένα το δακτυλικό αποτύπωμα. Παράλληλα, θα δείξουμε πως μπορεί ένας χρήστης να συνδεθεί σε ένα διακομιστή μέσω του δακτυλικού αποτυπώματος του από διαφορετικές συσκευές και πως αυτή η δυνατότητα επηρεάζει την ασφάλεια της όλης διαδικασίας ελέγχου ταυτότητας του χρήστη.

Κεφάλαιο 2

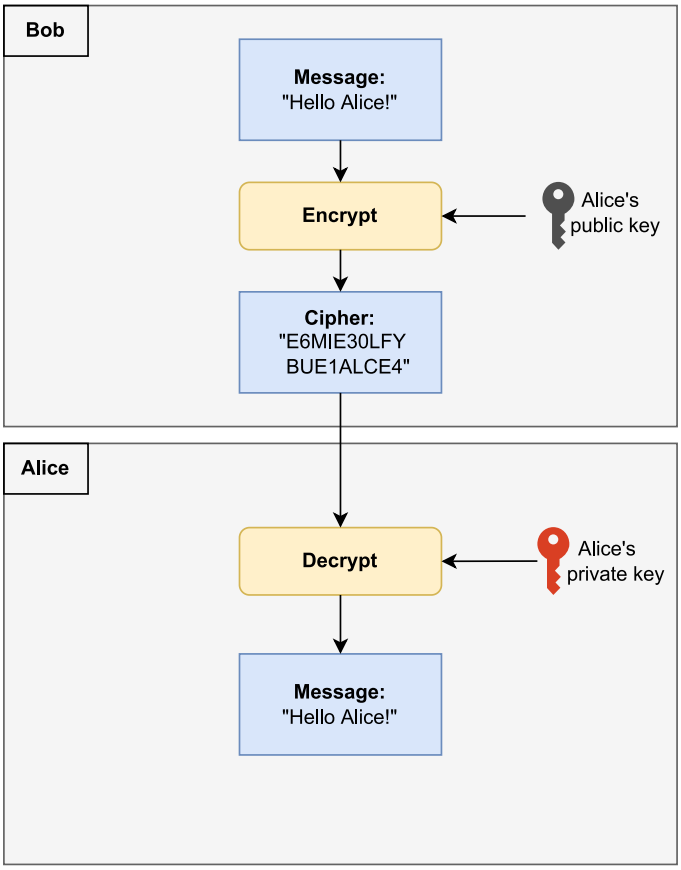
2 Υπόβαθρο

2.1 Κρυπτογράφηση δημοσίου κλειδιού

Για να επιτύχουμε την αυθεντικοποίηση του χρήστη θα χρησιμοποιήσουμε μια τεχνική η οποία ονομάζεται κρυπτογράφηση δημοσίου κλειδιού. Η κρυπτογράφηση δημοσίου κλειδιού βασίζεται στην ιδέα ενός ζεύγους κλειδιών το οποίο απαρτίζεται από ένα ιδιωτικό και ένα δημόσιο κλειδί. Τα δύο αυτά κλειδιά (ιδιωτικό και δημόσιο) έχουν μαθηματική σχέση μεταξύ τους [14]. Εάν το ένα χρησιμοποιηθεί για την κρυπτογράφηση κάποιου μηνύματος, τότε το άλλο χρησιμοποιείται για την αποκρυπτογράφηση αυτού.

Η επιτυχία αυτού του είδους κρυπτογραφικών αλγορίθμων βασίζεται στο γεγονός ότι η γνώση του δημόσιου κλειδιού κρυπτογράφησης δεν επιτρέπει με κανέναν τρόπο τον υπολογισμό του ιδιωτικού κλειδιού κρυπτογράφησης. Τα δεδομένα που έχουν κρυπτογραφηθεί με δημόσιο κλειδί μπορούν να αποκρυπτογραφηθούν μόνο με το αντίστοιχο ιδιωτικό κλειδί. Αντίστοιχα, τα δεδομένα που έχουν κρυπτογραφηθεί με ιδιωτικό κλειδί μπορούν να αποκρυπτογραφηθούν μόνο με το αντίστοιχο δημόσιο κλειδί. Ο ιδιοκτήτης του ζεύγους κλειδιών καθιστά το δημόσιο κλειδί διαθέσιμο σε όλους, ενώ

συνάμα κρατά το ιδιωτικό κλειδί κρυφό. Στην εικόνα 2-1 φαίνεται ένα παράδειγμα κρυπτογράφησης ενός μηνύματος χρησιμοποιώντας ένα ζεύγος κλειδιών.

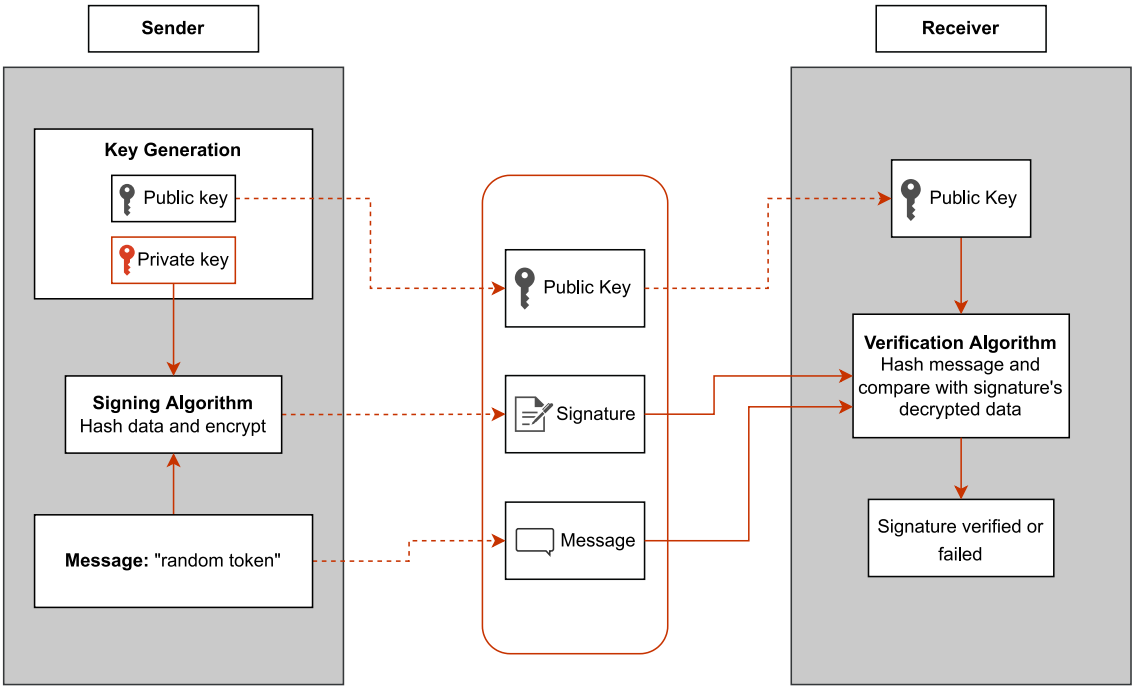


Εικόνα 2-1

Σε μια διαδικασία αυθεντικοποίησης, και γενικότερα όταν χρησιμοποιούμε ζεύγη κλειδιών και κατανέμουμε το δημόσιο κλειδί, χρειάζεται ένα πιστοποιητικό. Ένα πιστοποιητικό επαληθεύει ότι μια οντότητα είναι ο κάτοχος ενός συγκεκριμένου δημόσιου κλειδιού. Το πιστοποιητικό που χρησιμοποιήσουμε, το οποίο απαιτεί να συνοδεύει το δημόσιο κλειδί που δημιουργούμε, ακολουθεί το X.509 πρότυπο. Τα πιστοποιητικά που ακολουθούν το πρότυπο X.509 περιέχουν μια ενότητα δεδομένων και μια ενότητα υπογραφής. Η ενότητα δεδομένων περιλαμβάνει πληροφορίες όπως όνομα οντότητας που κατέχει το κλειδί, όνομα της οντότητας που εξέδωσε το πιστοποιητικό, χρονικό διάστημα κατά το οποίο ισχύει το πιστοποιητικό και το ίδιο το δημόσιο κλειδί [15].

Μπορούμε να αποκτήσουμε ένα πιστοποιητικό από μία Αρχή έκδοσης πιστοποιητικών (Certificate Authority – CA) όπως είναι η VeriSign. Εναλλακτικά, μπορούμε να δημιουργήσουμε ένα αυτό-υπογεγραμμένο πιστοποιητικό, στο οποίο ο κάτοχος και ο εκδότης είμαστε εμείς.

Στην εργασία αυτή αντί για κρυπτογράφηση και αποκρυπτογράφηση ενός μηνύματος θα εκτελέσουμε υπογραφή και επαλήθευση κάποιων δεδομένων. Η διαδικασία είναι παρόμοια με αυτή της κρυπτογράφηση και αποκρυπτογράφηση όπως αναφέραμε πιο πάνω με την διαφορά ότι δεν κρύβουμε το μήνυμα που στέλνουμε αλλά αντιθέτως το μήνυμα είναι γνωστό τόσο στον αποστολέα όσο και στον παραλήπτη. Στόχος της διαδικασίας αυτής είναι να επαληθεύσουμε ότι τα δεδομένα που στέλνονται υπογράφηκαν από τον αποστολέα και ότι παραμείναν απαράλλακτα, δηλαδή η υπογραφή που δημιουργήθηκε από τον αποστολέα να μπορεί να επαληθευτεί επιτυχώς από τον παραλήπτη. Η διαδικασία αυτή φαίνεται διαγραμματικά στο παρακάτω διάγραμμα (εικόνα 2-2).



Εικόνα 2-2

2.2 Ασύμμετρος αλγόριθμος

Καθώς χρησιμοποιούμε την κρυπτογράφηση δημοσίου κλειδιού, θα χρειαστούμε ένα ασύμμετρο αλγόριθμο με τον οποίο θα δημιουργήσουμε ένα ζεύγος κλειδιών (ιδιωτικό και δημόσιο κλειδί). Με αυτό το ζεύγος κλειδιών θα επιτύχουμε την σύνδεση στον διακομιστή. Ο αλγόριθμος που χρησιμοποιούμε είναι ο Elliptic Curves (EC). Αυτός ο αλγόριθμος χρησιμοποιεί ένα τυχαίο αριθμό από ένα εύρος αριθμών ως το ιδιωτικό κλειδί και επομένως είναι αρκετά γρήγορος. Το δημόσιο κλειδί είναι τα σημεία, δηλαδή ζεύγη ακέραιων συντεταγμένων (x, y), που ανήκουν πάνω στην ελλειπτική καμπύλη την οποία χρησιμοποιεί ο αλγόριθμος [16]. Χρησιμοποιούμε μια τυπική NIST ελλειπτική καμπύλη, την “secp256r1”.

Η επιλογή για τον αλγόριθμο EC έγκειται στο γεγονός ότι με τον αλγόριθμο RSA είναι δυσκολότερο να αναπαράγουμε ίδια ζεύγη κλειδιών. Αυτό οφείλεται στο γεγονός ότι είναι σχεδόν αδύνατο να διασφαλιστεί η παραγωγή δύο ίδιων αριθμών που πρέπει να είναι πρώτοι αριθμοί όπως απαιτεί η αρχικοποίηση του αλγορίθμου RSA. Αντίθετά, με τον EC χρειαζόμαστε μόνο ένα αριθμό ο οποίος θα επιλέγεται ντετερμινιστικά και στη συνέχεια επιλέγετε το ιδιωτικό κλειδί βάσει αυτού. Σημαντικό να αναφέρουμε πως ο αλγόριθμος EC παρέχει ισοδύναμη ισχύ κρυπτογράφησης όσο και ο RSA αλλά με μικρότερο μέγεθος κλειδιού με αποτέλεσμα να προσφέρει υψηλότερη ταχύτητα και μεγαλύτερη ασφάλεια.

2.3 Android Keystore

Για τον σκοπό μας πρέπει να αποθηκεύουμε ευαίσθητα δεδομένα, όπως το ιδιωτικό κλειδί που δημιουργούμε, σε ένα ασφαλές μέρος. Πρέπει να διασφαλίσουμε ότι το ιδιωτικό κλειδί αποθηκεύεται με ασφάλεια και ότι τα δεδομένα του δεν μπορούν να χρησιμοποιηθούν από κάποιον άλλο πέρα από τον πραγματικό χρήστη. Για το λόγο αυτό, το Android προσφέρει ένα Keystore, δηλαδή ένα χώρο όπου μπορούμε να αποθηκεύσουμε κρυπτογραφικά κλειδιά καθιστώντας δυσκολότερη την εξαγωγή τους από τη συσκευή. Τα κλειδιά που βρίσκονται στο Keystore, μπορούν να χρησιμοποιηθούν για κρυπτογραφικές λειτουργίες με το υλικό τους (key material) να παραμένει μη εξαγώγιμο. [17] Επιπλέον, προσφέρει δυνατότητες για τον περιορισμό του πότε και πώς

μπορούν να χρησιμοποιηθούν τα κλειδιά, όπως για παράδειγμα είναι η απαίτηση αυθεντικοποίησης του χρήστη πριν την χρήση τους [18] ή η χρήση των κλειδιών μόνο σε ορισμένες λειτουργίες κρυπτογράφησης (π.χ. για κρυπτογράφηση ή υπογραφή).

Το σύστημα Android Keystore προστατεύει το υλικό των κλειδιών από μη εξουσιοδοτημένη χρήση με δύο τρόπους. Πρώτον, το Android Keystore μετριάζει τη μη εξουσιοδοτημένη χρήση των κλειδιών εκτός της συσκευής Android, εμποδίζοντας την εξαγωγή του υλικού τους από τις διαδικασίες εφαρμογής και από τη συσκευή στο σύνολο της. Δεύτερον, το Android Keystore μετριάζει τη μη εξουσιοδοτημένη χρήση του υλικού των κλειδιών στη συσκευή Android, απαιτώντας από τις εφαρμογές να καθορίζουν εξουσιοδοτημένες χρήσεις των κλειδιών τους και στη συνέχεια επιβάλλοντας αυτούς τους περιορισμούς εκτός των διαδικασιών των εφαρμογών.

Η πρόληψη εξαγωγής γίνεται ακολουθώντας δύο μέτρα ασφαλείας στο υλικό των κλειδιών που αποθηκεύονται στο Android Keystore. Το πρώτο είναι ότι το υλικό του κλειδιού δεν εισέρχεται ποτέ στη διαδικασία της εφαρμογής. Δηλαδή, όταν μια εφαρμογή εκτελεί κρυπτογραφικές λειτουργίες χρησιμοποιώντας ένα κλειδί που βρίσκεται στο Android Keystore, στο παρασκήνιο το απλό κείμενο, το κρυπτογράφημα (κρυπτογραφημένο κείμενο) και τα μηνύματα που υπογράφονται ή επαληθεύονται δίνονται σε μια διαδικασία συστήματος που εκτελεί τις κρυπτογραφικές λειτουργίες. Εάν η διαδικασία της εφαρμογής έχει παραβιαστεί, ο εισβολέας ενδέχεται να μπορεί να χρησιμοποιήσει τα κλειδιά της εφαρμογής, αλλά δεν μπορεί να εξάγει το υλικό του κλειδιού του (για παράδειγμα, να χρησιμοποιηθεί εκτός της συσκευής Android). Το δεύτερο μέτρο είναι ότι το υλικό του κλειδιού μπορεί να συνδέεται με το ασφαλές υλικό της συσκευής Android (π.χ. Trusted Execution Environment (TEE), Secure Element (SE)). Όταν αυτή η δυνατότητα είναι ενεργοποιημένη για ένα κλειδί, το υλικό του δεν εκτίθεται ποτέ εκτός του ασφαλούς υλικού. Εάν το λειτουργικό σύστημα Android έχει παραβιαστεί ή ένας εισβολέας έχει πρόσβαση στον εσωτερικό χώρο αποθήκευσης της συσκευής, ο εισβολέας μπορεί να μπορεί να χρησιμοποιήσει τα κλειδιά μίας εφαρμογής που είναι αποθηκευμένα στο Android Keystore, αλλά δεν μπορεί να τα εξάγει από τη συσκευή. Επίσης, επισημαίνουμε ότι όταν ένα κλειδί αποθηκεύεται στο Android Keystore, επιστρέφεται ένα ψευδώνυμο στην εφαρμογή και μόνο το Keystore ξέρει πώς να αντιστοιχήσει κάποιο ψευδώνυμο με το κλειδί του.

Ο μετριασμός της μη εξουσιοδοτημένης χρήσης των κλειδιών γίνεται κατά την παραγωγή ή την εισαγωγή των κλειδιών στο Android Keystore. Μόλις δημιουργηθεί ή εισαχθεί ένα κλειδί, δεν είναι δυνατή η αλλαγή των εξουσιοδοτήσεων του. Στη συνέχεια, οι εξουσιοδοτήσεις επιβάλλονται από το Android Keystore κάθε φορά που χρησιμοποιείται το κλειδί. Οι εξουσιοδοτήσεις χρήσης κλειδιών μπορούν να εμπίπτουν σε τρεις κατηγορίες:

- Κρυπτογραφία: εξουσιοδοτημένος βασικός αλγόριθμος, λειτουργίες ή σκοποί (κρυπτογράφηση, αποκρυπτογράφηση, υπογραφή, επαλήθευση), padding schemes, block modes και digests με τα οποία μπορεί να χρησιμοποιηθεί το κλειδί.
- Χρονικό διάστημα εγκυρότητας: χρονικό διάστημα κατά το οποίο το κλειδί μπορεί να χρησιμοποιηθεί.
- Έλεγχος ταυτότητας χρήστη: το κλειδί μπορεί να χρησιμοποιηθεί μόνο εάν ο χρήστης έχει υποβληθεί σε έλεγχο ταυτότητας αρκετά πρόσφατα (π.χ. μέσω βιομετρικών στοιχείων).

Σημειώνουμε ότι ο πάροχος Android Keystore είναι μια υλοποίηση του KeyChain API. Η διαφορά μεταξύ αυτών των δύο είναι ότι το KeyChain API αποθηκεύει κλειδιά (διαπιστευτήρια) τα οποία είναι διαθέσιμα σε όλο το σύστημα, επιτρέποντας έτσι στις ξεχωριστές εφαρμογές μιας συσκευής να χρησιμοποιούν το ίδιο σύνολο κλειδιών με τη συγκατάθεση του χρήστη, ενώ το Android Keystore επιτρέπει σε μια μεμονωμένη εφαρμογή να αποθηκεύει τα δικά της διαπιστευτήρια στα οποία μόνο η ίδια η εφαρμογή μπορεί να έχει πρόσβαση. Το Android Keystore επιτρέπει στις εφαρμογές να διαχειρίζονται κλειδιά που μπορούν να χρησιμοποιούνται μόνο από αυτές, διατηρώντας παράλληλα το ίδιο επίπεδο ασφάλειας με το KeyChain API για κλειδιά και διαπιστευτήρια σε όλο το σύστημα [17]. Σε αυτή την εργασία, χρησιμοποιούμε τον πάροχο Android Keystore.

2.4 Biometric prompt and Crypto object

Το Android Keystore μας επιτρέπει να αποθηκεύουμε με ασφάλεια τα κρυπτογραφικά κλειδιά. Ωστόσο, πρέπει να μας δίνετε η δυνατότητα να τα χρησιμοποιήσουμε για

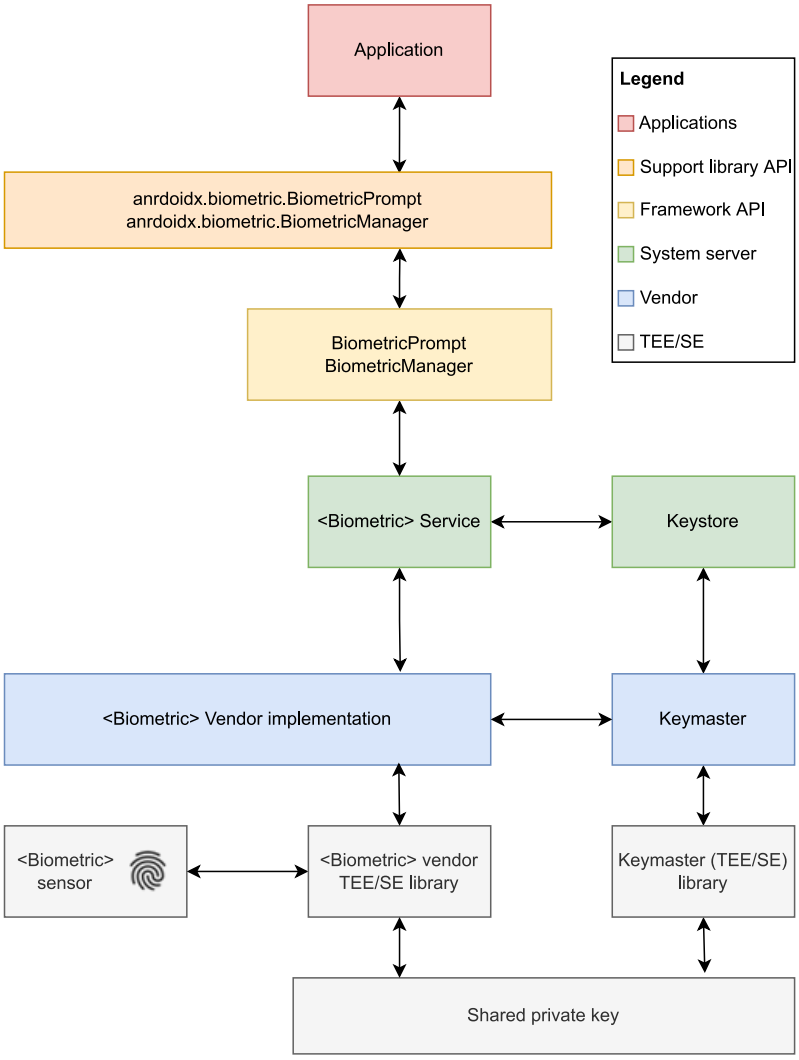
κρυπτογραφικές λειτουργίες. Για να γίνει αυτό, η Java μας παρέχει την κλάση CryptoObject η οποία μπορεί να συνδυαστεί με την BiometricPrompt για εκτέλεση κρυπτογραφικών λειτουργιών έπειτα αυθεντικοποίησης του χρήστη με βιομετρικά στοιχεία.

Η Java κλάση Biometric Prompt που μας παρέχεται από το Android, μας δίνει την δυνατότητα να δημιουργήσουμε ένα παράθυρο στο οποίο προτρέπουμε τον χρήστη να εισάγει το δακτυλικό του αποτύπωμα για έλεγχο ταυτότητας. Αυτό μας επιτρέπει να χρησιμοποιούμε βιομετρικά στοιχεία για να επαληθεύσουμε την ταυτότητα κάποιου ατόμου. Η δημιουργία ενός αντικειμένου αυτής της κλάσης απαιτεί ένα αντικείμενο callback το οποίο καθορίζει τι συμβαίνει σε κάθε περίπτωση αποτυχίας, επιτυχίας, ακύρωσης ή σφάλματος ελέγχου ταυτότητας [19].

Για να προστατεύσουμε περαιτέρω ευαίσθητες πληροφορίες μέσα σε μια εφαρμογή, ενδέχεται να ενσωματώσουμε την κρυπτογράφηση στη διαδικασία βιομετρικού ελέγχου ταυτότητας χρησιμοποιώντας αντικείμενα τύπου CryptoObject. Το CryptoObject είναι μια κλάση περιτύλιγμα (wrapper class) που μεταφέρει το κρυπτογράφημα οπουδήποτε μπορεί να χρειαστεί και γενικότερα λειτουργεί ως τομή μεταξύ βιομετρικών εργασιών και της κρυπτογράφησης. Τα αντικείμενα Signature, Cipher και Mac είναι μεταξύ των κρυπτογραφικών αντικειμένων που υποστηρίζονται από το αυτή την κλάση [20]. Αφού ο χρήστης πραγματοποιήσει επιτυχή έλεγχο ταυτότητας χρησιμοποιώντας βιομετρικά στοιχεία, η εφαρμογή μπορεί να εκτελέσει μια λειτουργία κρυπτογράφησης. Για παράδειγμα, εάν εκτελούμε έλεγχο ταυτότητας χρησιμοποιώντας ένα αντικείμενο Cipher, η εφαρμογή μας μπορεί στη συνέχεια να εκτελέσει κρυπτογράφηση και αποκρυπτογράφηση χρησιμοποιώντας ένα αντικείμενο SecretKey [19, 20]. Σε αυτή την εργασία, χρησιμοποιούμε το αντικείμενο Signature με το οποίο υπογράφουμε και επαληθεύουμε ορισμένα δεδομένα χρησιμοποιώντας ιδιωτικά και δημόσια κλειδιά μέσω του Android Keystore.

Δεδομένου ότι, τα κλειδιά που αποθηκεύονται στο Android Keystore ενδέχεται να απαιτούν έλεγχο ταυτότητας χρήστη πριν από τη χρήση. Επομένως, μπορούμε να ορίσουμε ένα callback το οποίο χρησιμοποιεί ένα αντικείμενο κρυπτογράφησης για την εκτέλεση λειτουργιών κρυπτογράφησης μετά από επιτυχή έλεγχο ταυτότητας.

Πιο κάτω βλέπουμε πως η κλάση Biometric prompt αλληλοεπιδρά με το Android Keystore καθώς και με τους διάφορους βιομετρικούς αισθητήρες και το TEE όπου εκτελούνται οι κρυπτογραφικές λειτουργίες [20]. Ουσιαστικά, μια εφαρμογή μπορεί να χρησιμοποιήσει τις διαδικασίες συστήματος, όπως είναι οι λειτουργίες που παρέχονται στο KeyStore, μέσω των αντιπροσωπειών που προσφέρονται από το Android π.χ. BiometricManager. Έπειτα οι διαδικασίες συστήματος χρησιμοποιούν το υλικό και κατ' επέκταση το ασφαλές υλικό για την εκτέλεση των διεργασιών τους.



Εικόνα 2-3

Κεφάλαιο 3

3 Τεχνικό μέρος

3.1 Αρχιτεκτονική

Η παρούσα ενότητα επικεντρώνεται στο πως τα διάφορα συστατικά στοιχεία επικοινωνούν μεταξύ τους για να ολοκληρώσουν την διαδικασία ελέγχου ταυτότητας χρήστη με βιομετρικά στοιχεία.

3.1.1 Βασικά συστατικά

Η εφαρμογή που δημιουργούμε ακολουθεί μια τυπική αρχιτεκτονική πελάτη-διακομιστή με διάφορα σημαντικά συστατικά να παίζουν καθοριστικό ρόλο και να υποβοηθούν την διαδικασία αυθεντικοποίησης. Επομένως τα κύρια συστατικά στοιχεία είναι:

- Relying party: Είναι μια αξιόπιστη υπηρεσία που εκτελείται από την πλευρά του διακομιστή. RP είναι η υπηρεσία που θα εκτελεί τον έλεγχο ταυτότητας του χρήστη και θα αποφασίζει αν είναι επιτυχείς ή όχι.
- Συσκευή πελάτη: μια συσκευή που ο χρήστης θα χρησιμοποιήσει για να καταχωρήσει τα βιομετρικά του στοιχεία στη συσκευή και να συνδεθεί στο

διακομιστή του διαδικτυακού συστήματος. Αυτή η συσκευή πρέπει να παρέχει ένα Keystore (όπως το Android Keystore) έτσι ώστε τα κρυπτογραφικά κλειδιά να μπορούν να αποθηκεύονται και να ανακτώνται με ασφάλεια και επίσης θα πρέπει να διαθέτει αισθητήρα δακτυλικών αποτυπωμάτων.

Όπως είδαμε και στην εικόνα 2-3, όλα τα απαραίτητα στοιχεία αλληλοεπιδρούν μεταξύ τους για να ολοκληρώσουν την εργασία αυθεντικοποίησης του χρήστη.

3.1.2 Διαδικασία

Επί της ουσίας, θέλουμε οι χρήστες να μπορούν να συνδεθούν σε κάποιο διακομιστή χρησιμοποιώντας το δακτυλικό τους αποτύπωμα από οποιαδήποτε συσκευή. Για το σκοπό αυτό δημιουργούμε μια εφαρμογή η οποία αποθηκεύει με ασφάλεια τις πληροφορίες που απαιτούνται για την συσχέτιση των βιομετρικών στοιχείων του χρήστη με κάποιο λογαριασμό στον διακομιστή. Η εφαρμογή επικοινωνεί με τον διακομιστή έτσι ώστε να εκτελεστεί η αυθεντικοποίηση του χρήστη. Η διαδικασία αυθεντικοποίησης γίνεται μέσω της κρυπτογραφικής υπογραφής και επαλήθευσης ενός μηνύματος.

Συγκεκριμένα, δημιουργούμε ένα ζεύγος κλειδιών για κάθε λογαριασμό χρήστη, αποθηκεύουμε με ασφάλεια στην συσκευή το ιδιωτικό κλειδί και στέλνουμε το δημόσιο στον διακομιστή για αποθήκευση. Έπειτα όποτε χρειάζεται να εκτελεστεί η διαδικασία αυθεντικοποίησης του χρήστη, ζητείται από την συσκευή να υπογράψει ένα μήνυμα και τα υπογεγραμμένα δεδομένα αποστέλλονται στον διακομιστή ο οποίος με τη σειρά του θα επαληθεύσει την υπογραφή του μηνύματος. Αν η επαλήθευση είναι επιτυχές τότε ο χρήστης μπορεί να συνδεθεί στον λογαριασμό του.

Για να μπορεί ο χρήστης να συνδεθεί από διάφορες συσκευές με το δακτυλικό του αποτύπωμα θα πρέπει σε κάθε συσκευή να αποθηκεύονται ή να αναπαράγονται οι απαραίτητες πληροφορίες, δηλαδή το αντίστοιχο ζεύγος κλειδιών. Για να μπορεί να γίνει αυτό χρησιμοποιούμε ένα αλγόριθμο που μας επιτρέπει την αναπαραγωγή ίδιου ζεύγους κλειδιών κάτω από συγκεκριμένες περιπτώσεις. Ειδικότερα, χρησιμοποιούμε τον κωδικό πρόσβασης του χρήστη για να αρχικοποιήσουμε τον αλγόριθμο και να μας επιστρέφει ντετερμινιστικά ένα ζεύγος κλειδιών βάσει αυτού. Έτσι σε κάθε συσκευή θα μπορούμε

να αναπαράγουμε το ζεύγος κλειδιών και να συνδεόμαστε στον διακομιστή μέσω βιομετρικών στοιχείων.

Ο χρήστης μπορεί να επιλέξει ένα ισχυρό ή τυχαίο κωδικό πρόσβασης τον οποίο δεν χρειάζεται να απομνημονεύσει και έπειτα να συνδέεται άμεσα με τα βιομετρικά του στοιχεία. Ακόμη και σε περίπτωση που ο χρήστης έχει ξεχάσει ή χάσει τον αρχικό κωδικό πρόσβασης θα μπορεί να συνδεθεί μέσω του δακτυλικού αποτυπώματος του και έπειτα επιτυχούς αυθεντικοποίησης να επιλέξει ένα καινούργιο. Φυσικά αυτό ισχύει μόνο σε περίπτωση που η συσκευή παραμένει σταθερή και μπορεί να συνδεθεί μέσω του δακτυλικού αποτυπώματος του. Σε αντίθετη περίπτωση, πιθανώς να ακολουθούνται άλλες διαδικασίες όπως επαναφορά κωδικού πρόσβασης ή εναλλακτικοί τρόποι ελέγχου ταυτότητας χωρίς κωδικό πρόσβασης (π.χ. OTP, magic links).

3.1.3 Ασφάλεια

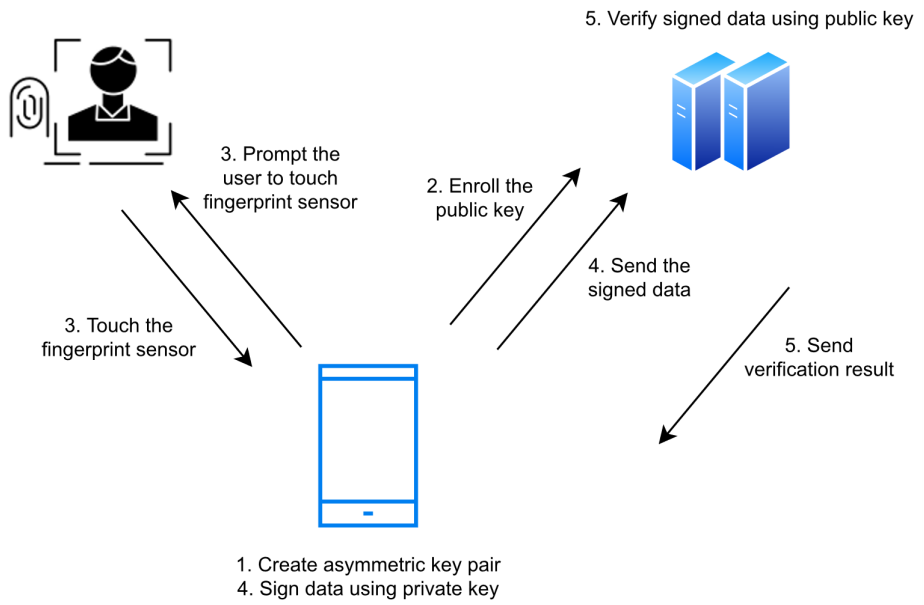
Χρησιμοποιώντας την κρυπτογράφηση δημοσίου κλειδιού σε συνδυασμό με την βιομετρική αυθεντικοποίηση του χρήστη αναμένουμε πως η όλη διαδικασία αυθεντικοποίησης του χρήστη καθώς και ο τρόπος που παράγουμε τα κλειδιά να είναι ασφαλές. Ωστόσο, ο αλγόριθμος που παρουσιάζουμε για να αυθεντικοποιήσουμε ένα χρήστη είναι ευάλωτος σε αρκετές επιθέσεις. Αυτό συμβαίνει επειδή θέλουμε ο χρήστης να μπορεί να αναπαράγει το ίδιο ζεύγος κλειδιών ακόμα και σε διαφορετικές συσκευές με αποτέλεσμα η όλη διαδικασία να βασίζεται σε ένα κωδικό ο οποίος αρχικοποιεί τον αλγόριθμο.

Αρχικά, η ασφάλεια του βασίζεται στο ότι ο χρήστης θα διαλέξει ένα αρκετά ισχυρό κωδικό πρόσβασης τέτοιο ώστε να είναι μοναδικός και μη προβλέψιμος. Όταν ο χρήστης επιθυμεί να συσχετίσει τα βιομετρικά του στοιχεία, ο αλγόριθμος απαιτεί την επιλογή ενός κωδικού πρόσβασης με τον οποίο θα αρχικοποιειθεί η διαδικασία και βάσει αυτού θα δημιουργηθεί ένα ζεύγος κλειδιών. Επομένως, δύο ζεύγη κλειδιών που δημιουργούνται με τον ίδιο κωδικό πρόσβασης θα είναι ίδια. Δεδομένου ότι δύο ζεύγη κλειδιών δημιουργήθηκαν με τον ίδιο κωδικό, αυτό φέρει ως αποτέλεσμα ένας χρήστης να μπορεί να συνδεθεί σε λογαριασμό άλλου χρησιμοποιώντας το δακτυλικό του αποτύπωμα (και το όνομα χρήστη του).

Η χρήση της κρυπτογράφησης δημοσίου κλειδιού μας προσφέρει την δυνατότητα να μπορούμε να απαλλαχτούμε από την αποθήκευση μυστικών κωδικών πρόσβασης στις βάσεις δεδομένων (ΒΔ). Ένας διακομιστής χρειάζεται να φυλάει στον αποθηκευτικό του χώρο μόνο το δημόσιο κλειδί του κάθε χρήστη. Θυμίζουμε ότι το δημόσιο κλειδί είναι άχρηστο χωρίς την παρουσία του αντίστοιχου ιδιωτικού κλειδιού και επομένως δεν έχει ουσιαστική σημασία να χρησιμοποιηθεί από μόνο του. Επομένως, το γεγονός ότι ένας διακομιστής δεν αποθηκεύει μυστικά κλειδιά αλλά μόνο δημόσια κλειδιά οι βάσεις δεδομένων γίνονται λιγότερο ελκυστικές προς τους χάκερς. Χρησιμοποιώντας ένα κωδικό πρόσβασης για την ντετερμινιστική παραγωγή αυτών των κλειδιών, χάνουμε την ασφάλεια που θα μας πρόσφερε η κρυπτογράφηση δημοσίου κλειδιού και έτσι οι ΒΔ γίνονται και πάλι ελκυστικές προς τους επιτιθέμενους.

3.2 Υλοποίηση

Στόχος μας είναι να συνδεθούμε σε έναν διακομιστή χρησιμοποιώντας βιομετρικά στοιχεία και συγκεκριμένα χρησιμοποιώντας δακτυλικά αποτυπώματα. Για να γίνει αυτό, χρειαζόμαστε ένα ζεύγος κλειδιών που θα χρησιμοποιηθεί για τη διαδικασία αυθεντικοποίησης. Συγκεκριμένα, το ιδιωτικό κλειδί θα αποθηκευτεί με ασφάλεια στη συσκευή Android (συγκεκριμένα στο Android Keystore) και το δημόσιο κλειδί θα σταλεί στο διακομιστή. Για το σκοπό αυτό, χρειαζόμαστε μια γεννήτρια κλειδιών που θα χρησιμοποιηθεί για τη δημιουργία ενός ζεύγους κλειδιών. Στη συνέχεια, κάθε φορά που ένας χρήστης θέλει να συνδεθεί, ο διακομιστής θα στείλει ένα μήνυμα που πρέπει να υπογραφεί χρησιμοποιώντας το ιδιωτικό κλειδί από το χρήστη και να σταλεί πίσω στο διακομιστή για επαλήθευση χρησιμοποιώντας το δημόσιο κλειδί. Το παρακάτω σχήμα απεικονίζει την παραπάνω διαδικασία.



Εικόνα 3-1

Όλη αυτή η διαδικασία απλοποιείται με την υλοποίηση μιας επέκτασης προγράμματος περιήγησης και τη χρήση ενός API που ονομάζεται WebAuthn. Αυτό το API δημιουργεί ένα τυχαίο ζεύγος κλειδιών RSA, αποθηκεύει με ασφάλεια το ιδιωτικό κλειδί στη συσκευή και στέλνει το δημόσιο κλειδί στο διακομιστή. Ωστόσο, αυτό το ζεύγος κλειδιών είναι μοναδικό κάθε φορά και δεν μπορεί να αναπαραχθεί εάν χρειαστεί, όπως εάν ο χρήστης αλλάξει συσκευή. Φυσικά, ένας ο διακομιστής μπορεί να αποθηκεύσει πολλά δημόσια κλειδιά για έναν συγκεκριμένο χρήστη.

Το WebAuthn χρησιμοποιεί τον RSA αλγόριθμο για την παραγωγή των κλειδιών καθιστώντας έτσι αδύνατη την αναπαραγωγή τους καθώς η ασφάλεια του RSA βασίζεται στην υπολογιστική δυσκολία της παραγοντοποίησης μεγάλων ακεραίων αριθμών. Σε αυτή τη μελέτη, προσπαθούμε να δημιουργήσουμε το ίδιο ζεύγος κλειδιών χρησιμοποιώντας έναν ισχυρό κωδικό πρόσβασης, ώστε να μπορεί να αναπαραχθεί όποτε επιθυμούμε. Για αυτό το σκοπό χρησιμοποιούμε τον αλγόριθμο EC αντί για RSA.

Η υλοποίηση μας χωρίζεται σε 2 φάσεις: εγγραφή ενός νέου χρήστη και σύνδεση του με βιομετρικά στοιχεία. Επιπλέον παρέχουμε την δυνατότητα σύνδεσης του χρήστη με κωδικό πρόσβασης χωρίς να αποθηκεύουμε αυτόν στον διακομιστή.

3.2.1 Έλεγχος για υποστήριξη βιομετρικών στοιχείων

Απαραίτητες προϋποθέσεις για ολοκληρωμένη και ορθή λειτουργία:

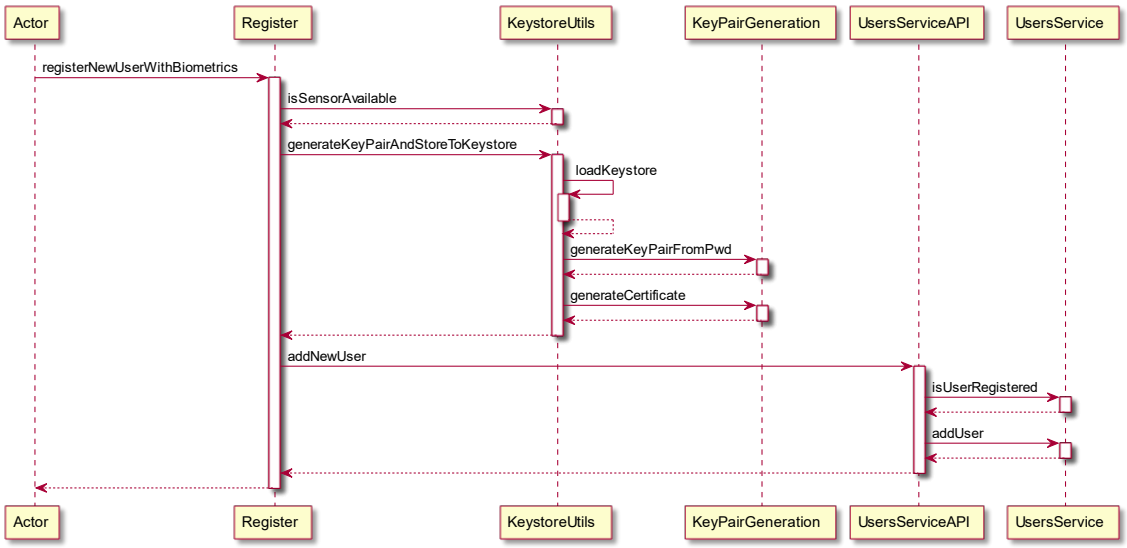
- Η συσκευή να παρέχει σαρωτή δακτυλικών αποτυπωμάτων
- Η ελάχιστη υποστηριζόμενη έκδοση Android της συσκευής να είναι το Android 10, δηλαδή API 29.
- Ο χρήστης πρέπει να έχει τουλάχιστο ένα δακτυλικό αποτύπωμα εγγεγραμμένο στην συσκευή του. (Σημειώνουμε ότι οποιοδήποτε από τα δακτυλικά αποτυπώματα που είναι εγγεγραμμένα στη συσκευή μπορούν να χρησιμοποιηθούν για τον ίδιο λογαριασμό χρήστη)

Δημιουργούμε μια συνάρτηση (isSensorAvailable) η οποία εκτελεί τους απαραίτητους ελέγχους και ενημερώνει τον χρήστη σε περίπτωση που δεν τηρείται κάποια προϋπόθεση. Η συνάρτηση αυτή καλείται πριν από διενέργειες που απαιτούν αυθεντικοποίηση του χρήστη, όπως κατά την φάση σύνδεσης ή εγγραφής του χρήστη.

```
1.  public static Response isSensorAvailable(Context context) {
2.     try {
3.        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
4.          BiometricManager biometricManager = BiometricManager.from(context);
5.          switch         (biometricManager.canAuthenticate(BIOMETRIC_STRONG         |
    DEVICE_CREDENTIAL)) {
6.            case BiometricManager.BIOMETRIC_SUCCESS:
7.               return new Response(true, "App can authenticate using biometrics.");
8.            case BiometricManager.BIOMETRIC_ERROR_NO_HARDWARE:
9.               return new Response(false, "No biometric features available on this
    device.");
10.           case BiometricManager.BIOMETRIC_ERROR_HW_UNAVAILABLE:
11.              return new Response( false, "Biometric features are currently
    unavailable.");
12.           case BiometricManager.BIOMETRIC_ERROR_NONE_ENROLLED:
13.              return new Response( false, "No fingerprints detected. Enroll one.");
14.           }
15.       } else
16.          return new Response(false,"Unsupported android version" );
17.    } catch (Exception e) {
18.          System.out.println("Error detecting biometrics availability: ");
19.    }
20.  return new Response(false,"Error detecting biometrics availability" );}
```

3.2.2 Εγγραφή νέου χρήστη

Για να προσθέσουμε ένα νέο χρήστη θα χρειαστεί να δημιουργήσουμε ένα ζεύγος κλειδιών και να το συσχετίσουμε με ασφαλές τρόπο με τα βιομετρικά στοιχεία του χρήστη. Έτσι δημιουργούμε ένα ζεύγος κλειδιών, αποθηκεύουμε το ιδιωτικό κλειδί στο Android Keystore το οποίο μπορεί να χρησιμοποιηθεί μόνο έπειτα αυθεντικοποίησης με βιομετρικά στοιχεία (δακτυλικό αποτύπωμα) και στέλνουμε στον διακομιστή το δημόσιο κλειδί μαζί με το όνομα χρήστη. Για τη δημιουργία του ζεύγους κλειδιού χρησιμοποιούμε μια γεννήτρια κλειδιών και ένα κωδικό πρόσβασης τον οποίο ζητάμε από το χρήστη. Πιο κάτω φαίνεται διαγραμματικά η διαδικασία εγγραφής ενός νέου χρήστη.



Εικόνα 3-2

3.2.2.1 Παραγωγή ζεύγους κλειδιών

Για την παραγωγή του ζεύγους κλειδιών χρησιμοποιούμε την κλάση KeyPairGenerator που προσφέρεται από την Java. Δημιουργούμε και αρχικοποιούμε ένα αντικείμενο αυτής της κλάσης έτσι ώστε να μας παράγει κλειδιά βάσει του αλγορίθμου EC. Η διαδικασία παραγωγής κλειδιών με τον αλγόριθμο EC χρησιμοποιεί ένα κρυπτογραφικά ασφαλής τυχαίο αριθμό για την επιλογή του ιδιωτικού κλειδιού. Έτσι, το KeyPairGenerator χρησιμοποιεί μια πραγματική πηγή τυχαιότητας για την παραγωγή νέων κλειδιών με αποτέλεσμα να έχουμε διαφορετικό ζεύγος κλειδιών σε κάθε εκτέλεση.

Η υλοποίηση του KeyPairGenerator μας προσφέρει την δυνατότητα να χρησιμοποιήσουμε τη δική μας ασφαλή γεννήτρια τυχαίων αριθμών (ΑΓΤΑ), και να υπερκαλύψουμε την προκαθορισμένη. Επομένως, χρειαζόμαστε μία ΑΓΤΑ με την οποία θα αρχικοποιήσουμε την γεννήτρια κλειδιών. Πρώτα όμως βεβαιώνουμε ότι η ΑΓΤΑ μας επιστρέφει πάντα τον ίδιο αριθμό εφόσον έχει ανατεθεί με μια τιμή αρχικοποίησης (seed).

Η κλάση KeyPairGenerator μας προσφέρει την επιλογή να ορίσουμε δική μας πηγή τυχαιότητας χρησιμοποιώντας την κλάση SecureRandom της Java και προαιρετικά ορίζοντας μία τιμή αρχικοποίησης. Η κλάση SecureRandom δημιουργεί έναν πραγματικό τυχαίο αριθμό ακόμα και όταν ορίζουμε μια σταθερή τιμή αρχικοποίησης. Έτσι, δεν θα μπορούσε να χρησιμοποιηθεί για την αρχικοποίηση του KeyPairGenerator, καθώς χρειαζόμαστε τη γεννήτρια να παράγει το ίδιο ζεύγος κλειδιών με τον ίδιο κωδικό πρόσβασης.

Η Bouncy Castle είναι μια συλλογή από δωρεάν ανοικτά API που κυρίως χρησιμοποιούνται στην κρυπτογραφία. Περιλαμβάνει API τόσο για τις γλώσσες προγραμματισμού Java όσο και για τις γλώσσες προγραμματισμού C#. Στην παρούσα υλοποίηση χρησιμοποιούμε αρκετές βιβλιοθήκες από αυτή τη συλλογή έναντι αυτών προσφέρονται απευθείας από την Java καθώς προσφέρουν διαφορετικές και συχνά πιο ευέλικτες υλοποιήσεις.

Η Bouncy Castle παρέχει μια κλάση, FixedSecureRandom , που επεκτείνει την SecureRandom κλάση και δημιουργεί τον ίδιο τυχαίο ισχυρό αριθμό, αν το seed που δίνεται παραμένει σταθερό. Δημιουργούμε ένα αντικείμενο FixedSecureRandom και το αρχικοποιούμε με μία σταθερή τιμή αρχικοποίησης. Βλέπουμε πως η ΑΓΤΑ που δημιουργήσαμε παράγει τον ίδιο αριθμό όταν αρχικοποιείται με την ίδια τιμή αρχικοποίησης κάθε φορά. Έτσι την χρησιμοποιούμε για να αρχικοποιήσουμε τη γεννήτρια κλειδιών.

Ένα αντικείμενο KeyPairGenerator έχει ως προεπιλεγμένη υπηρεσία παροχής υλοποίησης της γεννήτριας ζεύγους κλειδιών το AndroidOpenSSL. Ωστόσο, μας προσφέρει την δυνατότητα να ορίσουμε κάποιο άλλο παροχέα που θα δημιουργήσει τα

δύο κλειδιά. Η ασφαλέστερη επιλογή για την παραγωγή των κλειδιών είναι να χρησιμοποιήσουμε το Android Keystore ως παροχέα για την γεννήτρια κλειδιών έτσι ώστε η διαδικασία παραγωγής να γίνεται εσωτερικά στο Keystore (από μία διαδικασία συστήματος).

Χρησιμοποιούμε το KeyPairGenerator που παρέχεται από το Android Keystore για να δημιουργήσουμε ένα νέο ζεύγος κλειδιών που θα χρησιμοποιηθεί στη διαδικασία αυθεντικοποίησης. Το ζεύγος κλειδιών δημιουργείται απευθείας από το Android Keystore και επομένως η πρόσβαση από το επίπεδο της εφαρμογής είναι περιορισμένη.

Για παράδειγμα, το ιδιωτικό κλειδί δεν φεύγει ποτέ το Android Keystore, καθώς είναι συνδεδεμένο με το ασφαλές υλικό της συσκευής (π.χ. TEE). Όταν ζητάμε να λάβουμε το ιδιωτικό κλειδί από ένα ζεύγος κλειδιών, σε αντίθεση με το δημόσιο κλειδί στο οποίο επιστρέφονται τα δεδομένα του, τα δεδομένα του ιδιωτικού κλειδιού δεν εκτίθενται ποτέ εκτός του Keystore, αλλά αντ 'αυτού επιστρέφεται ένα αντικείμενο ιδιωτικού κλειδιού για να χρησιμοποιηθεί για άλλες λειτουργίες στην ίδια συσκευή.

Παρόλο που έχουμε ορίσει την δική μας πηγή τυχαιότητας η οποία παράγει ίδιους αριθμούς, η γεννήτρια κλειδιών εξακολουθεί να παράγει διαφορετικά ζεύγη κλειδιών σε κάθε εκτέλεση. Αυτό συμβαίνει επειδή η γεννήτρια κλειδιών που παρέχεται από το Android Keystore προσθέτει επίσης την δική της τυχαιότητα, οπότε είναι αδύνατο να δημιουργηθούν δύο πανομοιότυπα ζεύγη κλειδιών χρησιμοποιώντας την υλοποίηση γεννήτριας ζεύγους κλειδιών που παρέχεται από το Android Keystore.

Η πιο πάνω διαδικασία επαναλήφθηκε με ίδια αποτελέσματα και με τον προεπιλεγμένο παροχέα του KeyPairGenerator ο οποίος είναι ο AndroidOpenSSL. Ελέγχοντας άλλους παρόχους που υπάρχουν, βρίσκουμε πως η Bouncy Castle παρέχει επίσης υλοποίηση για τη γεννήτρια κλειδιών. Έτσι, μπορούμε να χρησιμοποιήσουμε το KeyPairGenerator με τον πάροχο BouncyCastle. Εκτελούμε ξανά την ίδια διαδικασία και βλέπουμε πως με αυτόν τον τρόπο επιτυγχάνουμε να δημιουργούμε δύο πανομοιότυπα ζεύγη κλειδιών χρησιμοποιώντας την πηγή τυχαιότητας που έχουμε ορίσει και αρχικοποιήσει με μια σταθερή τιμή.

Εάν αυτή η επιλογή αποτύγχανε επίσης, θα έπρεπε να κάνουμε τη δική μας υλοποίηση γεννήτριας ζεύγους κλειδιών, το οποίο δεν συνίσταται.

Για τον ορισμό της τιμή με την οποία αρχικοποιείται το SRNG χρησιμοποιούμε τον κωδικό πρόσβασης που δίνεται από τον χρήστη. Δηλαδή, περνάμε τον κωδικό του χρήστη μέσα σε μία συνάρτηση κατακερματισμού (SHA256) και έπειτα χρησιμοποιούμε αυτά τα bits για την αρχικοποίηση της ΑΓΤΑ. Εφόσον χρησιμοποιούμε την κλάση FixedSecureRandom ο ίδιος κωδικός χρήστη παράγει το την ίδια τιμή αρχικοποίησης κάθε φορά.

```
1.  public static KeyPair generateKeyPairFromPwd(String password) throws
    NoSuchAlgorithmException, NoSuchProviderException,
    InvalidAlgorithmParameterException {
2.
3.          //initialize trusted security
4.          Security.removeProvider("BC");
5.          Security.addProvider(new BouncyCastleProvider());
6.
7.          //hash the password and initialize the SRNG
8.          MessageDigest digest = MessageDigest.getInstance("SHA-256"); //min:128 bits
9.          byte[] seed = digest.digest(
10.                 password.getBytes(StandardCharsets.UTF_8));
11.         FixedSecureRandom randomnessSource = new FixedSecureRandom(seed);
12.
13.         //initialize EC key pair generator and generate key pair
14.         KeyPairGenerator keyGen = KeyPairGenerator.getInstance("EC", "BC");
15.         ECGenParameterSpec ecGenParameterSpec = new ECGenParameterSpec("secp256r1");
16.         keyGen.initialize(ecGenParameterSpec, randomnessSource);
17.         KeyPair keypair = keyGen.generateKeyPair();
18.
19.         return keypair;
20.     }
21.
```

Έτσι με την πιο πάνω διαδικασία, κάθε φορά που ο χρήστης πρέπει να αναπαράγει το ζεύγος κλειδιών του και να τα συσχετίζει με τα βιομετρικά του στοιχεία, αυτό μπορεί να γίνει χρησιμοποιώντας τον ίδιο κωδικό πρόσβασης που χρησιμοποιήθηκε κατά τη φάση εγγραφής.

3.2.2.2 Ασφαλής αποθήκευση του ζεύγους κλειδιού στο Android Keystore

Το Android Keystore είναι ένας ασφαλές τόπος για αποθήκευση του ζεύγους κλειδιών μας καθώς χρησιμοποιεί TEE και μας δίνει την δυνατότητα να περιορίσουμε την πρόσβαση του σε συγκεκριμένες κρυπτογραφικές λειτουργίες (π.χ. υπογραφή, επαλήθευση, κρυπτογράφηση, κλπ.). Επίσης, μπορούμε να εξασφαλίσουμε ότι αυτό το ζεύγος κλειδιών και ειδικότερα το ιδιωτικό κλειδί πως χρησιμοποιείται από τον ίδιο τον χρήστη ξεκλειδώνοντας το κλειδί μόνο έπειτα αυθεντικοποίησης του χρήστη (μέσω βιομετρικών στοιχείων). Δηλαδή κατά την δημιουργία του κλειδιού ορίζουμε τον σκοπό του κλειδιού (π.χ. *Purpose.SIGN | Purpose.VERIFY*) και την απαίτηση ταυτοποίησης του χρήστη πριν τη χρήση του (*setUserAuthneticationRequired(true)*).

Καθώς το ζεύγος κλειδιών που δημιουργούμε παράγεται εξωτερικά από το Android Keystore, θα πρέπει να το εισάγουμε ως μια νέα εγγραφή. Για την καταχώρηση του ζεύγους κλειδιών στο Android Keystore απαιτείται το δημόσιο κλειδί να συνοδεύεται με ένα X.509 πιστοποιητικό που πιστοποιεί το κλειδί. Εφόσον το ζεύγος κλειδιών δημιουργήθηκε με μια εξωτερική βιβλιοθήκη, κατασκευάζουμε και το πιστοποιητικό αυτό εξωτερικά από το Android Keystore.

Συνήθως τα δημόσια κλειδιά υπογράφονται από κάποιο αναγνωρισμένο οργανισμό. Στην περίπτωση μας, χρησιμοποιούμε και πάλι την Bouncy Castle που μας παρέχει την δυνατότητα να δημιουργήσουμε ένα αυτο-υπογεγραμμένο πιστοποιητικό X.509. Όπως βλέπουμε και στην συνάρτηση πιο κάτω χρησιμοποιούμε δικές μας πληροφορίες για να προσδιορίσουμε την οντότητα που υπογράφει το πιστοποιητικό. Επίσης, κατασκευάζουμε το πιστοποιητικό αυτό χρησιμοποιώντας το ιδιωτικό κλειδί μέσω του αλγορίθμου ECDSA με SHA-1. Στο τέλος εκτελούμε κάποιες δοκιμές για να επιβεβαιώσουμε ότι το πιστοποιητικό είναι έγκυρο και μπορεί να χρησιμοποιηθεί.

Συγκεκριμένα ελέγχουμε ότι μπορεί να χρησιμοποιηθεί την στιγμή που δημιουργείται και επαληθεύουμε ότι αυτό το πιστοποιητικό υπογράφηκε χρησιμοποιώντας το ιδιωτικό κλειδί που αντιστοιχεί στο καθορισμένο δημόσιο κλειδί.

```
1.  public static X509Certificate generateCertificate(KeyPair keyPair) {
2.   // subjects name - the same as we are self signed.
3.    String issuerString = "CN=bio_auth, OU=CS, O=UCY, L=Nicosia, C=CY";
4.    String subjectString = "CN=bio_auth, OU=CS, O=UCY, L=Nicosia, C=CY";
5.    X500Name issuer = new X500Name(issuerString);
6.    BigInteger serial = BigInteger.ONE;
7.    Date notBefore = new Date();
8.    Date notAfter = new Date(System.currentTimeMillis() + (2 * 365 * 24 * 60 * 60 *
    1000l));    //expires after 2 years
9.    X500Name subject = new X500Name(subjectString);
10.   PublicKey publicKey = keyPair.getPublic();
11.   JcaX509v3CertificateBuilder v3Bldr = new JcaX509v3CertificateBuilder(issuer,
    serial, notBefore, notAfter, subject, publicKey);
12.
13.   X509CertificateHolder certHldr = v3Bldr
14.      .build(new
    JcaContentSignerBuilder("SHA1WITHECDSA").setProvider("BC").build(keyPair.getPrivate
    ()));
15.
16.   X509Certificate cert = new
    JcaX509CertificateConverter().setProvider("BC").getCertificate(certHldr);
17.
18.   // testing our certificate that:
19.   cert.checkValidity(new Date()); //can be used now
20.   cert.verify(keyPair.getPublic());   //can verify using pubkey
21.   return cert;
22. }
23.
```

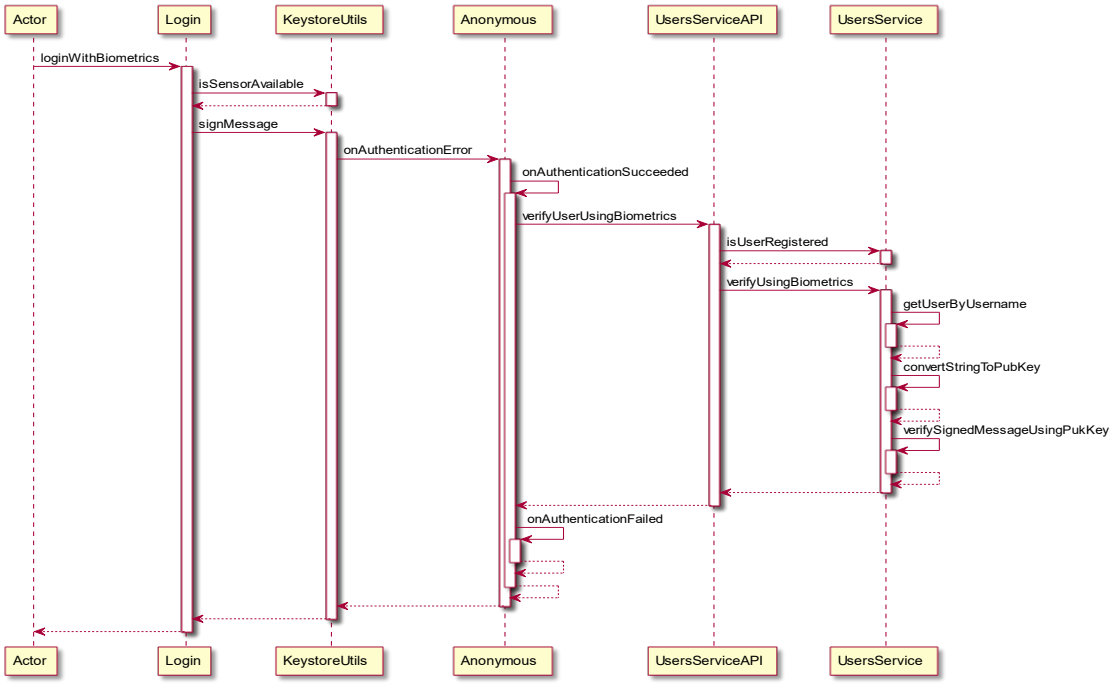
Αφού δημιουργήσουμε το πιστοποιητικό το χρησιμοποιούμε μαζί με το ζεύγος κλειδιών και ένα δικό μας ψευδώνυμο για να καταχωρήσουμε μια νέα εγγραφή στο Android Keystore. Το ψευδώνυμο χρησιμοποιείται ως αναγνωριστικό για την συγκεκριμένη εγγραφή έτσι ώστε να έχουμε πρόσβαση στο ζεύγος κλειδιών όταν χρειαστεί. Το ψευδώνυμο που χρησιμοποιούμε είναι μοναδικό ανά εφαρμογή και ανά συσκευή.

```
1.  KeyStore ks = loadKeystore(null);
2.  KeyPair keyPair = generateKeyPairFromPwd(password);
3.  X509Certificate certificate = generateCertificate(keyPair);
4.  ks.setKeyEntry(alias,  keyPair.getPrivate(),  null,  new X509Certificate[]{
5.      certificate
6.  });
```

3.2.3 Σύνδεση χρήστη με βιομετρικά στοιχεία

Όταν ο χρήστης επιθυμεί να συνδεθεί στο λογαριασμό του μπορεί να χρησιμοποιήσει τα βιομετρικά του στοιχεία, δηλαδή το δακτυλικό του αποτύπωμα, για να συνδεθεί απευθείας χωρίς να πρέπει να εισάγει κωδικό πρόσβασης.

Η όλη διαδικασία σύνδεσης γίνεται με τον χρήστη να υπογράφει ένα μήνυμα χρησιμοποιώντας το ιδιωτικό κλειδί που έχουμε δημιουργήσει και αποθηκεύσει με ασφάλεια στο Android Keystore. Έπειτα, αποστέλλουμε το υπογεγραμμένο μήνυμα στον διακομιστή για επαλήθευση της υπογραφής. Εάν το υπογεγραμμένο μήνυμα επαληθευτεί με επιτυχία χρησιμοποιώντας το αντίστοιχο δημόσιο κλειδί του χρήστη που είναι αποθηκευμένο στη βάση δεδομένων, τότε ο χρήστης μπορεί να συνδεθεί στο σύστημα. Στο πιο κάτω διάγραμμα παρουσιάζεται η εκτέλεση που ακολουθείται.



Εικόνα 3-3

Η παραπάνω διαδικασία εκτελείται μόνο μετά την επιτυχή επαλήθευση του χρήστη χρησιμοποιώντας τα βιομετρικά του στοιχεία (δακτυλικό αποτύπωμα).

3.2.3.1 Υπογραφή και αποστολή μηνύματος έπειτα ταυτοποίησης

Για να υπογράψουμε ένα μήνυμα πρέπει να δημιουργήσουμε ένα αντικείμενο τύπου Signature και να το αρχικοποιήσουμε με το ιδιωτικό κλειδί που θα χρησιμοποιήσει για να εκτελέσει την κρυπτογραφική λειτουργία, δηλαδή την διαδικασία υπογραφής. Σημειώνουμε ότι το ιδιωτικό κλειδί δεν φεύγει ποτέ από το Android Keystore, επομένως δεν μπορούμε να δούμε το τα πραγματικά δεδομένα του. Ωστόσο, μπορούμε να έχουμε πρόσβαση σε αυτό και να ζητήσουμε από το Android Keystore να εκτελέσει κρυπτογραφικές λειτουργίες, οι οποίες με ασφάλεια θα λαμβάνουν τόπο στο επίπεδο συστήματος και θα μας επιστρέψει το αποτέλεσμα. Έτσι, όταν αρχικοποιούμε το αντικείμενο Signature με ένα ιδιωτικό κλειδί, του δίνουμε ως παράμετρο ένα αντικείμενο ιδιωτικού κλειδιού με το οποίο μπορεί να έχει πρόσβαση στα πραγματικά δεδομένατου μόνο στο επίπεδο του Keystore.

Δημιουργούμε ένα αντικείμενο Signature (Crypto Object) το οποίο αρχικοποιούμε με τον αλγόριθμο ECDSA με SHA-256 και με το ιδιωτικό κλειδί. Χρησιμοποιούμε τη συνάρτηση authenticate του BiometricPrompt ζητώντας από τη συσκευή να εκτελέσει τη διαδικασία αυθεντικοποίησης του χρήστη μέσω των βιομετρικών του στοιχείων. Επιπλέον δίνουμε ως όρισμα στην συνάρτηση αυτή, το αντικείμενο Signature που έχουμε κατασκευάσει.

```
1.  Signature signature = Signature.getInstance("SHA256withECDSA");
2.  signature.initSign(getPrivateKey());
3.  BiometricPrompt.PromptInfo promptInfo = new BiometricPrompt.PromptInfo.Builder()
4.          .setDeviceCredentialAllowed(false)
5.          .setNegativeButtonText("Cancel?")
6.          .setTitle("Signing")
7.          .build();
8.  BiometricPrompt bp = new BiometricPrompt((FragmentActivity) view.getContext(),
    ContextCompat.getMainExecutor(view.getContext()), authCallBack);
9.  bp.authenticate(promptInfo, new BiometricPrompt.CryptoObject(signature));
```

Για τη διαδικασία αυθεντικοποίησης του χρήστη μέσω βιομετρικών στοιχείων (δακτυλικό αποτύπωμα) χρησιμοποιούμε τη βιβλιοθήκη Biometrics του Android X. Αρχικά δημιουργούμε ένα αντικείμενο BiometricPrompt πάνω στο οποίο θα καλέσουμε την συνάρτηση authenticate. Το BiometricPrompt χρειάζεται ένα Authentication

Callback αντικείμενο με το οποίο καθορίζουμε τι εκτελείτε σε κάθε περίπτωση που προκύπτει με τη διαδικασία της αυθεντικοποίησης. Δηλαδή, ορίζουμε τι θα εκτελέσει σε περίπτωση αποτυχίας, επιτυχίας ή ακύρωσης της διαδικασίας αυθεντικοποίησης.

Σε περίπτωση επιτυχίας, δίνεται πρόσβαση στο Crypto Object, δηλαδή στο Signature, και τότε μπορούμε να καλέσουμε τη συνάρτηση update του αντικειμένου Signature με την οποία θέτουμε το μήνυμα που θέλουμε να υπογραφεί. Στη συνέχεια, καλούμε τη συνάρτηση sign η οποία υπογράφει αυτό το μήνυμα βάσει των παραμέτρων που έχουμε ορίσει (αλγόριθμο για υπογραφή, ιδιωτικό κλειδί, κτλ.). Θυμίζουμε πως οι κρυπτογραφικές λειτουργίες πραγματοποιούνται από διαδικασίες συστήματος του Keystore καθώς εκεί υπάρχει η πρόσβαση για το ιδιωτικό κλειδί. Τέλος, στέλνουμε στον διακομιστή το υπογεγραμμένο μήνυμα και αναμένουμε για την επικύρωση του.

```
1.  Signature signature = Signature.getInstance("SHA256withECDSA");
2.  signature.initSign(getPrivateKey());
3.  BiometricPrompt.PromptInfo promptInfo = new BiometricPrompt.PromptInfo.Builder()
4.          .setDeviceCredentialAllowed(false)
5.          .setNegativeButtonText("Cancel?")
6.          .setTitle("Signing")
7.          .build();
8.  BiometricPrompt bp = new BiometricPrompt((FragmentActivity) view.getContext(),
    ContextCompat.getMainExecutor(view.getContext()), authCallBack);
9.  bp.authenticate(promptInfo, new BiometricPrompt.CryptoObject(signature));
```

Σε περίπτωση ακύρωσης ή αποτυχίας αυθεντικοποίησης δεν εκτελούμε κάποια ενέργεια πέρα από την εμφάνιση κατάλληλων μηνυμάτων.

3.2.3.2 Επικύρωση υπογεγραμμένου μηνύματος από διακομιστή

Αρχικά ελέγχουμε πως το όνομα χρήστη που δόθηκε υπάρχει στην βάση δεδομένων μας και έπειτα καλούμε μια συνάρτηση από τον διακομιστή για να επικυρώσει το υπογεγραμμένο μήνυμα. Όπως και πριν, αρχικοποιούμε ένα αντικείμενο Signature με το δημόσιο κλειδί, του αντίστοιχου χρήστη, που είναι αποθηκευμένο στην ΒΔ και με τον ίδιο αλγόριθμο (ECDSA με SHA-256). Καλούμε τη συνάρτηση update με το μήνυμα που αναμένουμε, δηλαδή το ίδιο μήνυμα με το οποίο εκτελέσαμε την υπογραφή. Έπειτα εκτελούμε την συνάρτηση verify στα δεδομένα του υπογεγραμμένου μηνύματος που

στάλθηκε από τον χρήστη. Η συνάρτηση verify επιστρέφει μια τιμή αληθείας ανάλογα με το αν το αποτέλεσμα της διαδικασίας επικύρωσης του υπογεγραμμένου μηνύματος είναι ίδιο με το αναμενόμενο μήνυμα.

```
1.  private boolean verifySignedMessageUsingPukKey(PublicKey pubKey, byte[] signedData)
    {
2.    try {
3.      Signature sig = Signature.getInstance("SHA256withECDSA");
4.      sig.initVerify(pubKey);
5.      sig.update("<random token>".getBytes(StandardCharsets.UTF_8));
6.      boolean authenticated = sig.verify(signedData);
7.      return authenticated;
8.    } catch (NoSuchAlgorithmException | InvalidKeyException | SignatureException e) {
9.        e.printStackTrace();
10.   }
11.   return false;
12. }
```

3.2.4 Αναπαραγωγή ζεύγους κλειδιών και συσχέτιση του με βιομετρικά στοιχεία

Η εφαρμογή μας δίνει την δυνατότητα να συνδεθούμε από διαφορετικές συσκευές Android, χρησιμοποιώντας τα βιομετρικά μας στοιχεία παρόλο που μπορεί να διερμηνεύονται διαφορετικά σε κάθε συσκευή. Αυτό επιτυγχάνεται αναπαράγοντας το ίδιο ζεύγος κλειδιών χρησιμοποιώντας τον αλγόριθμο EC και ένα ισχυρό κωδικό πρόσβασης, όπως περιγράψαμε στο μέρος 3.1.2.1.. Σημειώνουμε πως για να παραχθεί το ίδιο ζεύγος κλειδιών χρειαζόμαστε τον ίδιο κωδικό πρόσβασης τον οποίο χρησιμοποιήσαμε για να αρχικοποιήσουμε την γεννήτρια κλειδιών στην φάση εγγραφής του χρήστη.

Παράγουμε ένα ζεύγος κλειδιών χρησιμοποιώντας τον κωδικό πρόσβασης του χρήστη για να αρχικοποιήσουμε την γεννήτρια τυχαίων αριθμών και κατ' επέκταση την γεννήτρια κλειδιών. Αυτό το ζεύγος κλειδιών το αποθηκεύομαι με ασφάλεια στο Android Keystore και το ορίζουμε έτσι ώστε η χρήση του να απαιτεί αυθεντικοποίηση του χρήστη (με βιομετρικά στοιχεία). Στη συνέχεια, τροποποιούμε την εγγραφή στην ΒΔ που αφορά τον εκάστοτε χρήστη με το νέο δημόσιο κλειδί που παραγάγαμε. Με αυτό τον τρόπο,

έχουμε αποθηκεύσει με ασφάλεια στην συσκευή το ζεύγος κλειδιών που θα χρησιμοποιεί ο χρήστης για να συνδέεται στον διακομιστή.

```
1.  public static void alterUserWithBiometrics(String username, String password, View
    view) {

2.    Response rsp = isSensorAvailable(view.getContext());
3.    if(!rsp.isSuccess()) {
4.      Toast.makeText(view.getContext(), rsp.getMessage(), Toast.LENGTH_SHORT).show();
5.      return;
6.    }

7.    generateKeyPairAndStoreToKeystore(password);

8.    String pubKey = getPubKeyFromKeystore();

9.    if (pubKey == null) {
10.     Toast.makeText(view.getContext(), "Couldn't retrieve the public key",
    Toast.LENGTH_SHORT).show();
11.     return;
12.   }
13.
14.   Response response = UsersServiceAPI.addBiometricsToUser(username, pubKey);

15.   Toast.makeText(view.getContext(), response.getMessage(),
    Toast.LENGTH_SHORT).show();
16.
17. }
```

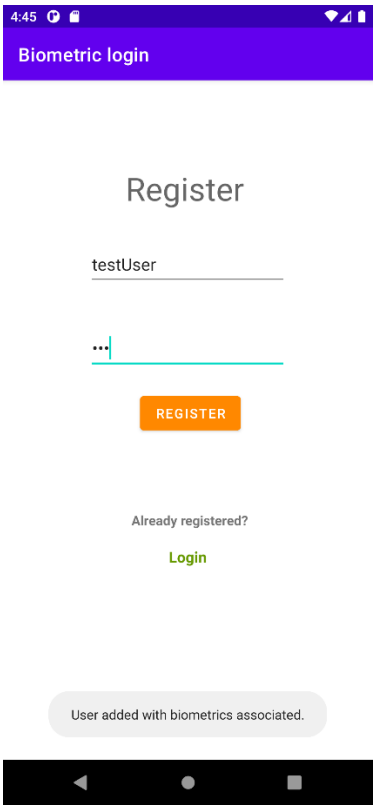
Επισημαίνουμε πως αν ο κωδικός πρόσβασης που χρησιμοποιούμε για αυτή τη διαδικασία είναι ίδιος με τον αρχικό κωδικό που χρησιμοποιήθηκε στην φάση εγγραφής τότε όλες οι συσκευές έχουν συσχετίσει τα βιομετρικά στοιχεία που έχουν εγγεγραμμένα με το ίδιο ζεύγος κλειδιών με αποτέλεσμα να έχουν πρόσβαση στον λογαριασμό του χρήστη μέσω του δακτυλικού αποτυπώματος. Αντιθέτως αν ο νέος κωδικός είναι διαφορετικός από τον αρχικό, οι υπόλοιπες συσκευές θα πρέπει να συσχετίσουν τα βιομετρικά τους στοιχεία επαναλαμβάνοντας αυτή τη διαδικασία χρησιμοποιώντας τον νέο κωδικό πρόσβασης.

3.3 Αξιολόγηση

Για να αξιολογήσουμε και να επικυρώσουμε την πιο πάνω διαδικασία δημιουργούμε μία εφαρμογή που υλοποιεί τις λειτουργίες που περιγράψαμε στο προηγούμενο τμήμα. Χρησιμοποιούμε την γλώσσα προγραμματισμού Java και υλοποιούμε μια απλή διεπιφάνεια χρήστη όπως θα δούμε πιο κάτω. Η εφαρμογή αυτή προσφέρει τη δυνατότητα εγγραφής νέου χρήστη και της σύνδεσης του στον διακομιστή μέσω δακτυλικού αποτυπώματος. Αποφεύγοντας την χρήση ενός πραγματικού διακομιστή, προσημειώνουμε ένα Java πακέτο με τις λειτουργίες που εκτελεί και διαθέτει ο διακομιστής. Επίσης, για την αποθήκευση των δεδομένων, δηλαδή των εγγεγραμμένων χρηστών, χρησιμοποιούμε μια δωρεάν βάση δεδομένων μέσω του Google Firebase.

3.3.1 Επικύρωση διαδικασίας εγγραφής

Εκτελούμε τις διεργασίες που περιεγράφηκαν πιο πάνω για να επιβεβαιώσουμε ότι λειτουργεί ορθά η διαδικασία εγγραφής και σύνδεσης. Αρχικά εγγράφουμε ένα νέο χρήστη με όνομα χρήστη «testUser» και κωδικό «123». Εμφανίζεται σχετικό μήνυμα που επιβεβαιώνει ότι ο χρήστης έχει εγγραφεί με επιτυχία και πως έχουν συσχετιστεί τα ήδη αποθηκευμένα βιομετρικά του στοιχεία με την εφαρμογή που δημιουργήσαμε στο Android Keystore (εικόνα 3-4). Επίσης, μπορούμε να δούμε στην ΒΔ που έχουμε δημιουργήσει ότι προστίθεται μια νέα εγγραφή με ονομασία το όνομα χρήστη ακολουθούμενο με το δημόσιο κλειδί που έχει δημιουργηθεί (εικόνα 3-5).



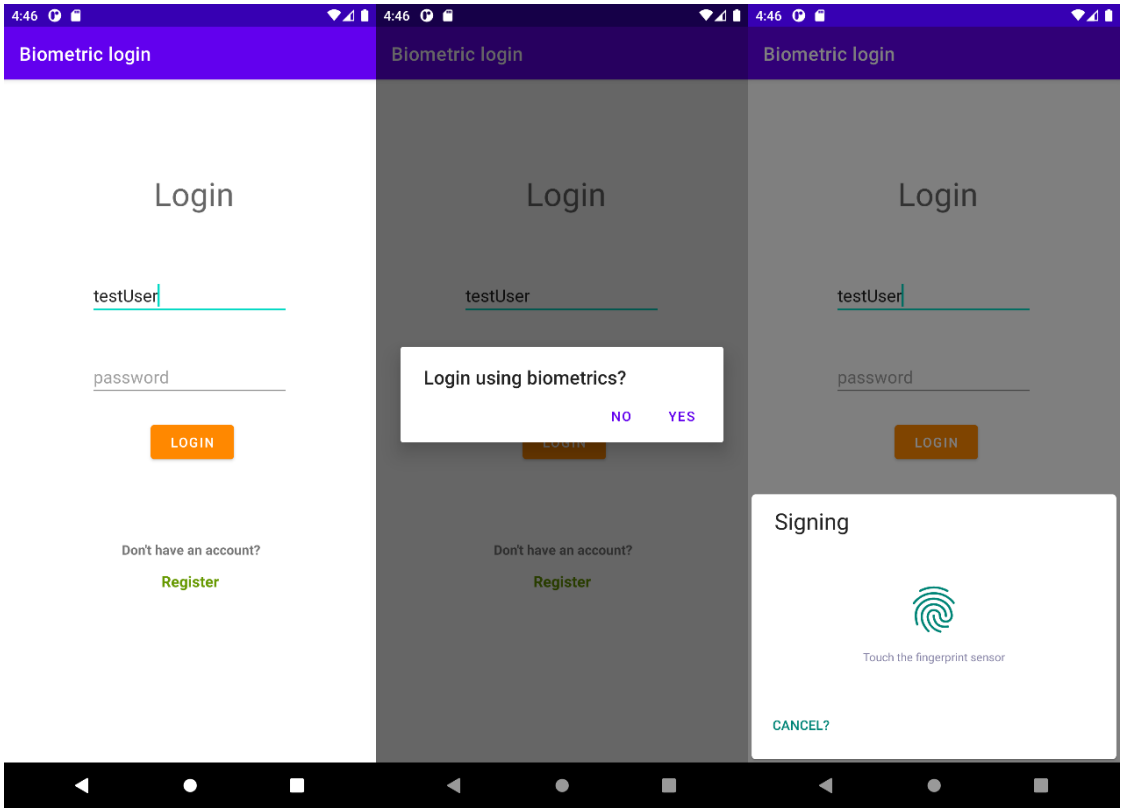
Εικόνα 3-4

Εικόνα 3-5

3.3.2 Επαλήθευση διαδικασίας σύνδεσης

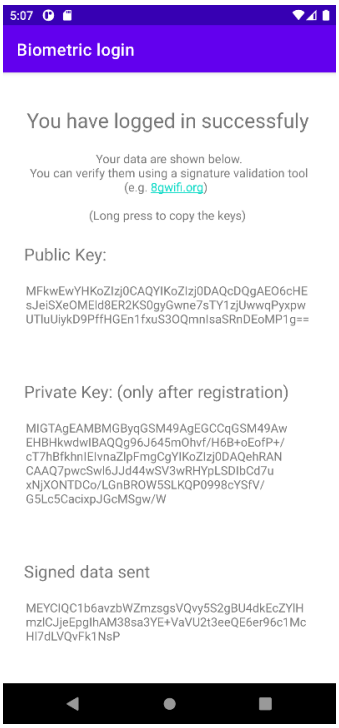
Η σύνδεση μπορεί να γίνει απευθείας μέσω βιομετρικών στοιχείων ή μέσω του κωδικού πρόσβασης. Εισάγουμε το όνομα χρήστη και πατώντας το κουμπί login εμφανίζεται ένα αναδυόμενο παράθυρο στο οποίο ερωτώμαστε αν θέλουμε να συνδεθούμε με βιομετρικά στοιχεία. Με την επιλογή «Yes» το λογισμικό μας προτρέπει για την σάρωση του δακτυλικού μας απωτυπώματος. Αν η αυθεντικοποίηση που γίνεται από το σύστημα είναι

επιτυχές, συνδεόμαστε στο σύστημα/υπηρεσία μέσω της διαδικασίας υπογραφής – επαλήθευσης ενός μηνύματος που περιεγράφηκε στην ενότητα 3.1.3.



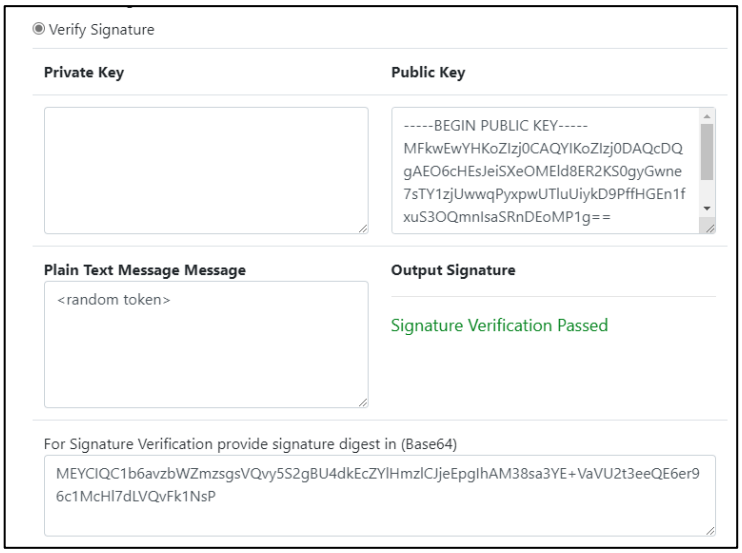
Εικόνα 3-6 *Εικόνα 3-7* *Εικόνα 3-8*

Έπειτα της επιτυχούς σύνδεσης στον διακομιστή παραπεμπόμαστε στην αρχική σελίδα (εικόνα 3-9) η οποία μας εμφανίζει διάφορες πληροφορίες που σχετίζονται με την διαδικασία σύνδεσης. Σε κάθε περίπτωση, εφόσον ο χρήστης συνδεθεί, εμφανίζεται το δημόσιο κλειδί. Αντίθετα, το ιδιωτικό κλειδί εμφανίζεται μόνο την πρώτη φορά σύνδεσης αμέσως μετά την εγγραφή του χρήστη καθώς πρόσβαση σε αυτό έχουμε μόνο κατά τη φάση δημιουργίας του κλειδιού. Φυσικά, το ιδιωτικό κλειδί δεν θα το εμφανίζαμε σε καμία περίπτωση αλλά μόνο για να εκτελέσουμε εξωτερική επαλήθευση της υπογραφής που δημιουργούμε. Τέλος εμφανίζουμε το υπογεγραμμένο μήνυμα το οποίο υπογράφεται με το ιδιωτικό κλειδί και επαληθεύεται με το δημόσιο.



Εικόνα 3-9

Χρησιμοποιούμε ένα εξωτερικό εργαλείο το οποίο προσφέρει δυνατότητες υπογραφής – επαλήθευσης για να επαληθεύσουμε τα αποτελέσματα μας. Συγκεκριμένα χρησιμοποιούμε το εργαλείο που προσφέρεται από τον οργανισμό 8gwifi (www.8gwifi.org/ecsignverify.jsp) μαζί με το δημόσιο κλειδί και το υπογεγραμμένο μήνυμα. Όπως βλέπουμε στην εικόνα 3-10, το υπογεγραμμένο μήνυμα επαληθεύεται με επιτυχία χρησιμοποιώντας το αρχικό μήνυμα που χρησιμοποιήσαμε για την υπογραφή του, δηλαδή το «<random token>».

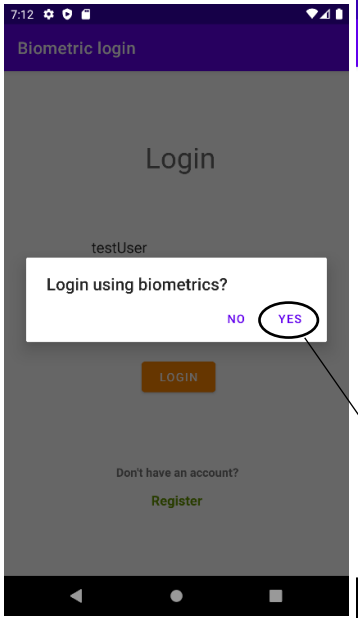


Εικόνα 3-10

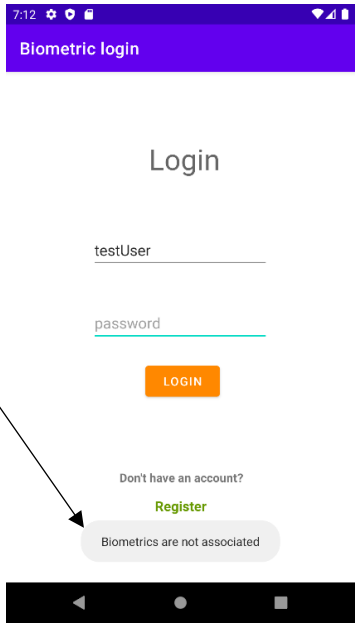
Όπως αναφέραμε προ ολίγου, η σύνδεση ενός χρήστη μπορεί να πραγματοποιηθεί με την χρήση του ονόματος χρήστη και κωδικού πρόσβασης επιλέγοντας «No» στο αναδυόμενο παράθυρο που εμφανίζεται μετά από το πάτημα κουμπιού “login” (εικόνα 3-7). Χρησιμοποιούμε το ίδιο όνομα χρήστη όπως προηγουμένως και τον κωδικό που θέσαμε, «123». Ο χρήστης συνδέεται με επιτυχία και παραπέμπεται στην ίδια οθόνη όπως φαίνεται στην εικόνα 3-9 με την διαφορά ότι πλέον δεν έχουμε πρόσβαση στα δεδομένα του ιδιωτικού κλειδιού και έτσι δεν εμφανίζονται στην οθόνη .

3.3.3 Αναπαραγωγή κλειδιών και συσχέτιση βιομετρικών στοιχείων μεταξύ διαφορετικών συσκευών

Για να επικυρώσουμε την διαδικασία αναπαραγωγής του ζεύγους κλειδιών χρησιμοποιούμε μια νέα συσκευή η οποία δεν έχει ξαναχρησιμοποιήσει την εφαρμογή μας. Αρχικά δοκιμάζουμε να συνδεθούμε μέσω δακτυλικού αποτυπώματος στον λογαριασμό που έχουμε ήδη δημιουργήσει με όνομα χρήστη testUser και βλέπουμε πως αποτυγχάνει. Αυτό είναι αναμενόμενο καθώς δεν έχουμε συσχετίσει ή αποθηκεύσει ακόμη κάποιο ζεύγος κλειδιών στο Android Keystore με τον χρήστη. Για να γίνει αυτό πρέπει ο χρήστης να δημιουργήσει νέο λογαριασμό στον διακομιστή ή να συνδεθεί σε υπάρχον λογαριασμό χρησιμοποιώντας τον κωδικό πρόσβασης.

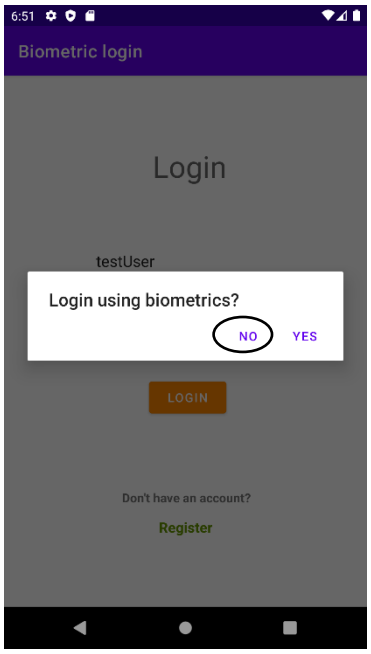


Εικόνα 3-11

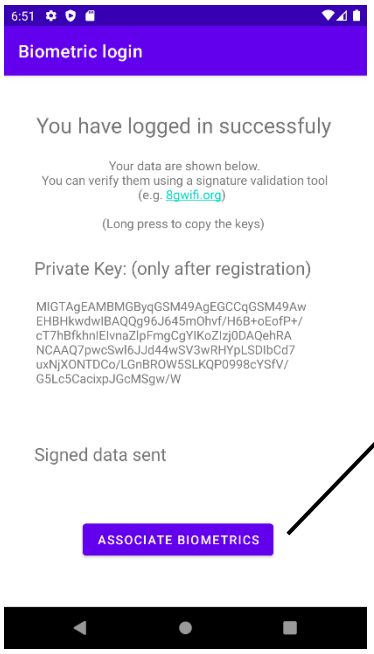


Εικόνα 3-12

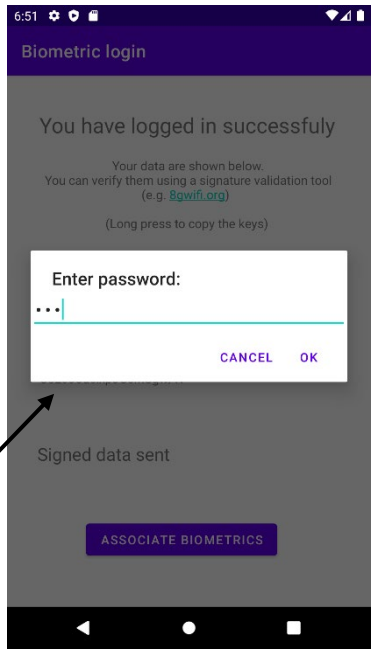
Εισάγουμε το όνομα χρήστη «testUser» και τον κωδικό πρόσβασης «123» και επιλέγουμε να συνδεθούμε χρησιμοποιώντας κωδικό πρόσβαση αντί βιομετρικών στοιχείων. Η σύνδεση εκτελείτε με επιτυχία και μεταφερόμαστε στην κύρια σελίδα όπου φαίνονται οι πληροφορίες που χρησιμοποιήθηκαν στην διαδικασία σύνδεσης. Για να δημιουργήσουμε και να αποθηκεύσουμε με ασφάλεια στο Android Keystore το ζεύγος κλειδιών μεταφερόμαστε στο κάτω τέλος της οθόνης και πατάμε το κουμπί «Associate Biometrics». Αμέσως ζητείται ένας κωδικός ο οποίος θα χρησιμοποιηθεί για την αρχικοποίηση της διαδικασίας ως επίσης και για επικύρωση της ταυτότητας του χρήστη. Ο κωδικός αυτός μπορεί είτε να παραμείνει ο ίδιος με τον κωδικό σύνδεσης είτε να γίνει εισαγωγή νέου κωδικού πρόσβασης.



Εικόνα 3-13

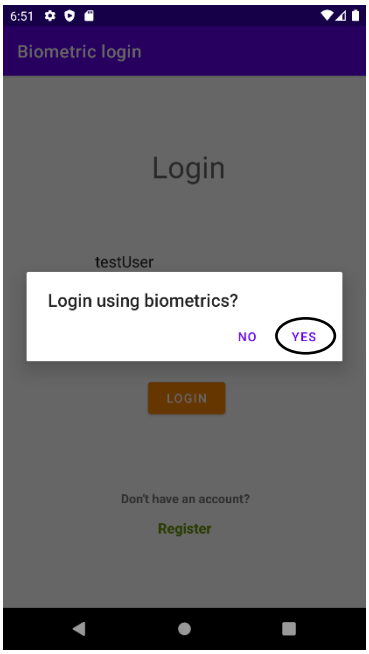


Εικόνα 3-14

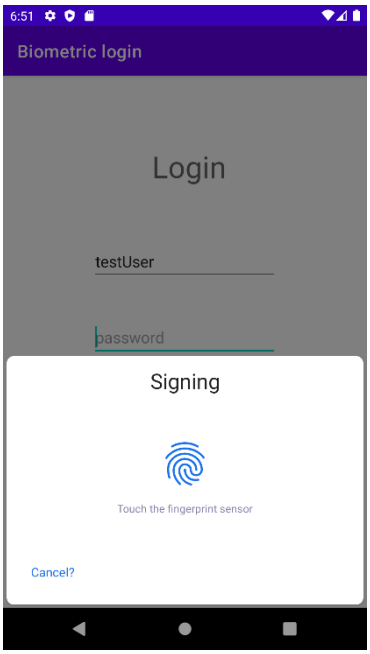


Εικόνα 3-15

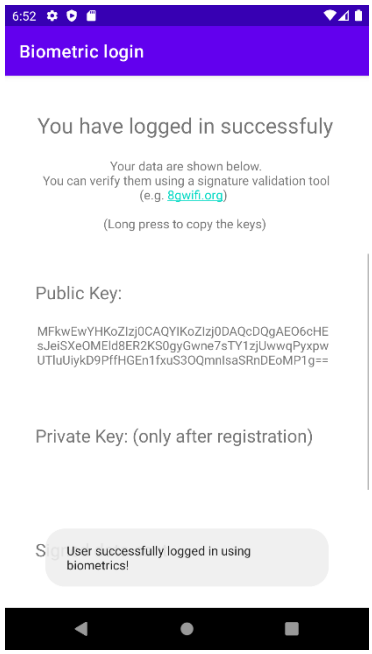
Έπειτα της εισαγωγής του κωδικού, εκτελείτε η διεργασία όπως περιεγράφηκε στο μέρος 3.1.4 και τώρα μπορούμε να συνδεθούμε μέσω δακτυλικού αποτυπώματος και από αυτή την Android συσκευή.



Εικόνα 3-16



Εικόνα 3-17



Εικόνα 3-18

Σημειώνουμε πως σε περίπτωση εισαγωγής διαφορετικού κωδικού πρόσβασης δημιουργείται διαφορετικό ζεύγος κλειδιών και επομένως οι χρήστες που έχουν ήδη αποθηκεύσει ζεύγος κλειδιών στο Android Keystore θα πρέπει να επαναλάβουν αυτήν την διαδικασία για να μπορούν να συνθέσουν με βιομετρικά στοιχεία.

Κεφάλαιο 4

4 Σχετικές εργασίες

Κατά καιρούς έχουν υπάρξει αρκετοί ερευνητές οι οποίοι ασχολήθηκαν με την αυθεντικοποίηση ενός χρήστη χωρίς κωδικό πρόσβασης αλλά με βιομετρικά στοιχεία. Παράλληλα, υπήρξαν και αρκετές υλοποιήσεις και προτάσεις για αυτού του είδους ελέγχου ταυτότητας. Επικρατέστερη επιλογή έχει γίνει το WebAuthn που αναφέραμε προηγουμένως και χρησιμοποιείται ευρέως από πολλά διαδικτυακά συστήματα. Ωστόσο το WebAuthn για λόγους ασφαλείας αλλά και για να διατηρήσει τα θετικά που προσφέρει η αυθεντικοποίηση χωρίς κωδικούς πρόσβασης και συγκεκριμένα η κρυπτογράφηση του δημοσίου κλειδιού, δεν επιτρέπει την αναπαραγωγή των κλειδιών γενικότερα. Δίνει όμως την επιλογή ένας χρήστης να μπορεί να έχει πολλαπλά μοναδικά ζεύγη κλειδιών και με αυτό τον τρόπο να μπορεί να συνδέεται σε ένα διακομιστή από διάφορες συσκευές.

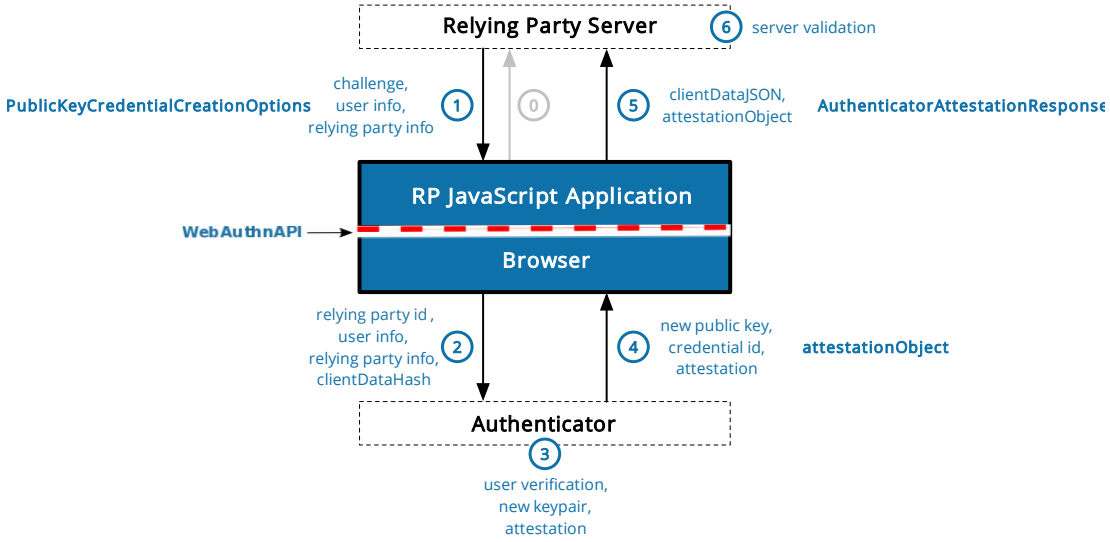
4.1 WebAuthn

Το Web Authentication (WebAuthn) είναι μια προδιαγραφή που δημιουργήθηκε από το W3C και το FIDO, με συνεισφορές από την Google, τη Mozilla, τη Microsoft, τη Yubico και άλλους. Το API επιτρέπει στους διακομιστές να καταχωρούν και να ελέγχουν την ταυτότητα των χρηστών χωρίς τη χρήση κωδικού πρόσβασης χρησιμοποιώντας μια ποικιλία μεθόδων αυθεντικοποίησης. Συγκεκριμένα χρησιμοποιεί την κρυπτογράφηση

δημόσιου κλειδιού. Συνοπτικά, αντί για έναν κωδικό πρόσβασης, δημιουργείται ένα ιδιωτικό και δημόσιο κλειδί (ζεύγος κλειδιών) για έναν ιστότοπο. Το ιδιωτικό κλειδί αποθηκεύεται με ασφάλεια στη συσκευή του χρήστη και το αντίστοιχο δημόσιο κλειδί αποστέλλεται στο διακομιστή για αποθήκευση. Ο διακομιστής τότε μπορεί να χρησιμοποιήσει το δημόσιο κλειδί για να επιβεβαιώσει την ταυτότητα του χρήστη. Το δημόσιο κλειδί δεν είναι μυστικό επειδή ουσιαστικά δεν μπορεί να χρησιμοποιηθεί χωρίς το αντίστοιχο ιδιωτικό κλειδί.

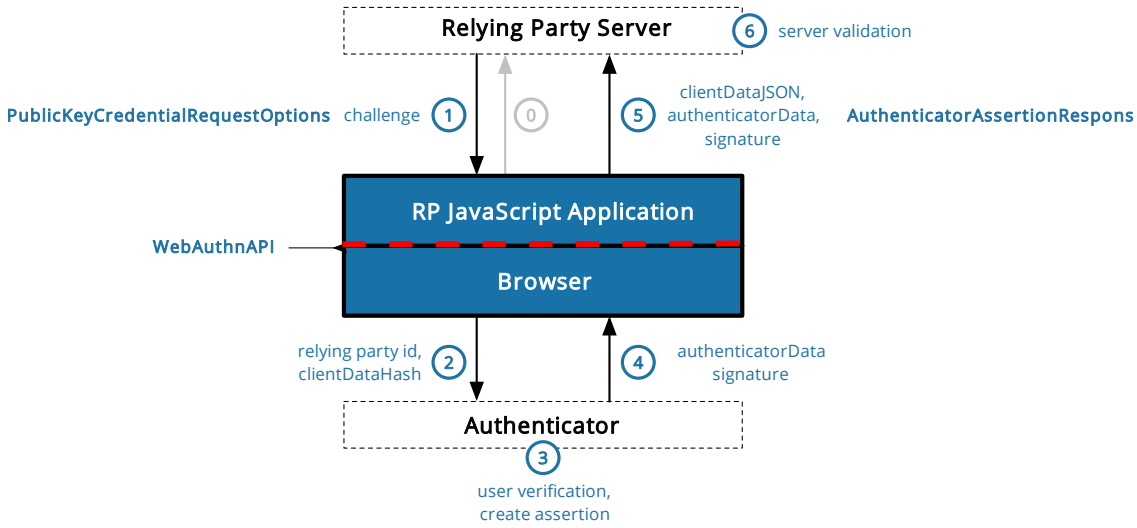
Η αυθεντικοποίηση υποστηρίζεται ιδανικά από μια λειτουργική μονάδα ασφαλές υλικού, η οποία μπορεί να αποθηκεύσει με ασφάλεια ιδιωτικά κλειδιά και να πραγματοποιήσει τις κρυπτογραφικές λειτουργίες που απαιτούνται για το WebAuthn. Επίσης, ένα ζεύγος κλειδιών είναι χρήσιμο μόνο για μία συγκεκριμένη προέλευση (origin). Δηλαδή, ένα ζεύγος κλειδιών που δημιουργείται για μία εφαρμογή δεν μπορεί να χρησιμοποιηθεί για κάποια άλλη, μετριάζοντας τις απειλές phishing. Χρησιμοποιείται επίσης ένα πιστοποιητικό που βοηθά τους διακομιστές να επαληθεύσουν ότι το δημόσιο κλειδί δεν προήλθε από μια δόλια πηγή, αλλά από έναν ελεγκτή ταυτότητας (authenticator) που εμπιστεύονται [21].

Το API WebAuthn έχει δύο φάσεις: τη φάση εγγραφής και ελέγχου ταυτότητας. Όταν ο χρήστης θέλει να δημιουργήσει ένα νέο λογαριασμό, ο διακομιστής (relying party) ζητά από τον υπολογιστή-πελάτη να του στείλει ένα δημόσιο κλειδί. Η συσκευή δημιουργεί με ασφάλεια ένα ζεύγος κλειδιών, αποθηκεύει το ιδιωτικό στη συσκευή και στέλνει το δημόσιο κλειδί πίσω στο διακομιστή. Ανταλλάσσουν επίσης επιπρόσθετες πληροφορίες μεταξύ τους όπως μία κρυπτογραφικά τυχαία συμβολοσειρά για να αποτρέψει επιθέσεις επανάληψης και άλλα. Παρακάτω, μπορούμε να δούμε τη ροή της φάσης εγγραφής.



Εικόνα 4-1 πηγή: www.w3.org/TR/webauthn-2

Μετά τη φάση εγγραφής, κάθε φορά που ο χρήστης θέλει να συνδεθεί, ξεκινά η φάση ελέγχου ταυτότητας. Ο διακομιστής στέλνει ορισμένα δεδομένα στον υπολογιστή-πελάτη και ζητά να τα υπογράψει χρησιμοποιώντας το ιδιωτικό κλειδί. Στη συνέχεια, η συσκευή δημιουργεί με ασφάλεια μια υπογραφή αυτών των δεδομένων χρησιμοποιώντας το ιδιωτικό κλειδί που είναι αποθηκευμένο στη συσκευή και το στέλνει πίσω στο διακομιστή. Ο διακομιστής χρησιμοποιεί το δημόσιο κλειδί που ανακτήθηκε κατά την εγγραφή για να επαληθεύσει τα υπογεγραμμένα δεδομένα. Η ροή της φάσης ελέγχου ταυτότητας φαίνεται παρακάτω.



Εικόνα 4-2 πηγή: www.w3.org/TR/webauthn-2

Όλες αυτές οι λειτουργίες εκτελούνται στον ελεγκτή ταυτότητας (authenticator) και διαμεσολαβούνται από την μεριά του πελάτη. Σε κανένα σημείο ο κώδικας δεν έχει πρόσβαση στα ίδια τα διαπιστευτήρια (credentials), ωστόσο λαμβάνει πληροφορίες για αυτά με τη μορφή αντικειμένων. Επίσης, ο ελεγκτής ταυτότητας, ο οποίος κατέχει και διαχειρίζεται τα διαπιστευτήρια, διασφαλίζει ότι όλες οι λειτουργίες εκτελούνται σε μια συγκεκριμένη προέλευση και δεν μπορούν να επαναληφθούν έναντι διαφορετικής προέλευσης, ενσωματώνοντας την προέλευση στις απαντήσεις του [22].

Σε αυτό το έργο εφαρμόζουμε τη βάση για κάτι παρόμοιο με αυτό το API με τη διαφορά ότι τα κλειδιά μπορούν να αναπαραχθούν. Στην περίπτωσή μας ο ελεγκτής ταυτότητας είναι το Android Keystore.

Κεφάλαιο 5

5 Μελλοντική δουλειά και εφαρμογές

5.1 Εναλλακτική λύση για σύνδεση από πολλαπλές συσκευές

Μια σημαντική και ασφαλές εναλλακτική λύση αποτελεί η δημιουργία και αποθήκευση πολλαπλών ζευγών κλειδιών για ένα λογαριασμό χρήστη από διαφορετικές συσκευές, χωρίς δυνατότητα αναπαραγωγής σε άλλη συσκευή. Δηλαδή, ο χρήστης θα μπορεί να δημιουργήσει ένα ζεύγος κλειδιών από κάθε συσκευή που κατέχει, και ο διακομιστής θα πρέπει να προσφέρει την δυνατότητα αποθήκευσης πολλαπλών δημοσίων κλειδιών για ένα λογαριασμό χρήστη. Με αυτό τον τρόπο μπορούμε να συνδεόμαστε με βιομετρικά στοιχεία από πολλαπλές συσκευές χρησιμοποιώντας την κρυπτογραφία δημοσίου κλειδιού χωρίς να χάνουμε τα κύρια οφέλη της.

5.2 Χρήση βιομετρικών στοιχείων στην υγειονομική περίθαλψη

Η ταχεία και ασφαλής πρόσβαση σε ιατρικές πληροφορίες ήταν πάντα κρίσιμη, αλλά η πανδημία COVID-19 έχει καταστήσει ακόμη περισσότερο εντατική την ανάγκη της. Τα

ιατρικά αρχεία και πληροφορίες γίνονται πιο πολύτιμα ενώ συνάμα το έγκλημα στον κυβερνοχώρο που περιλαμβάνει την κλοπή τους γίνεται όλο και πιο συχνό και σοβαρό. Επειδή ούτε οι ασθενείς ούτε οι φροντιστές μπορούν να αντέξουν οικονομικά τη διαταραχή ή τους κινδύνους των αδύναμων ή συμβιβασμών κωδικών πρόσβασης, η βιομετρική σάρωση και αυθεντικοποίηση του χρήστη εξαλείφει την ευκαιρία για εξωτερικούς παράγοντες να έχουν πρόσβαση σε αυτές τις προσωπικές πληροφορίες [23].

5.3 Χρήση βιομετρικών στοιχείων στο χώρο εργασίας

Κυρίαρχη άποψη που επικρατεί είναι πως η τεχνολογία σάρωσης προσώπου θα είναι το επόμενο κύμα βιομετρικών στοιχείων που θα αναπτυχθεί στο χώρο εργασίας. Η τεχνολογία σάρωσης προσώπου γίνεται ήδη μέρος της διαδικασίας ελέγχου ιστορικού κάποιου ατόμου με τη μορφή επαλήθευσης ταυτότητας. Η επαλήθευση ταυτότητας επιτρέπει στους εργοδότες να διασφαλίζουν ότι οι υποψήφιοι είναι αυτοί που λένε ότι είναι πριν ξεκινήσουν έναν έλεγχο για το ιστορικό τους.

Επιπρόσθετα, η βιομετρική τεχνολογία που χρησιμοποιείται σε έξυπνες κλειδαριές ή βιομετρικές κάρτες πρόσβασης μπορεί να απλοποιήσει τον φυσικό έλεγχο πρόσβασης και για ευέλικτη εργασία, καθώς οι παραδοσιακές ώρες εργασίας γίνονται πιο ευέλικτες. Με μόνο εξουσιοδοτημένους χρήστες να έχουν πρόσβαση, οι επιχειρήσεις έχουν την ηρεμία ότι μόνο οι υπάλληλοί τους είναι παρόν στον χώρο εργασίας, ανεξάρτητα από το χρόνο που βρίσκονται εκεί για να εργάζονται. Επίσης, τα βιομετρικά στοιχεία μπορούν να ενσωματωθούν στα συστήματα ταυτοποίησης, χρόνου και παρακολούθησης, ώστε οι υπεύθυνοι λήψης αποφάσεων ανθρώπινου δυναμικού να μπορούν να έχουν πιο αξιόπιστες γνώσεις σχετικά με την εργασία των εργαζομένων τους [24].

5.4 Χρήση βιομετρικών στοιχείων στις τράπεζες

Οι πελάτες τραπεζών απαιτούν ολοένα και μεγαλύτερη ευκολία, σε σημείο που να θέλουν ταχεία πρόσβαση στα χρηματικά τους κεφάλαια, ανεξάρτητα από το πού βρίσκονται ή τι είδους συσκευή χρησιμοποιούν.

Η σάρωση δακτυλικών αποτυπωμάτων και η αναγνώριση προσώπου στα έξυπνα κινητά βοηθούν ήδη σε αυτό, αλλά τα τραπεζικά ιδρύματα πρέπει να προχωρήσουν περαιτέρω

για να επιτρέψουν αξιόπιστη και αλάνθαστη αυθεντικοποίηση του χρήστη η οποία να γίνεται σε πραγματικό χρόνο. Για να διασφαλιστεί ότι ένας χρήστης που θέλει να αποκτήσει πρόσβαση σε έναν λογαριασμό είναι αυτός που λέει ότι είναι και προσπαθεί να επαληθεύσει την ταυτότητα του εκείνη τη στιγμή, ο έλεγχος ταυτότητας χρήστη σε πραγματικό χρόνο είναι απαραίτητος [23].

5.5 Άλλες εφαρμογές

Μια ενδιαφέρουσα εφαρμογή των βιομετρικών στοιχείων η οποία θα ευκολύνει αρκετά τον πελάτη στην πληρωμή των αγορών του είναι η πληρωμή χωρίς ταμείο. Η Amazon και άλλες μεγάλες επιχειρήσεις δοκιμάζουν όλο και περισσότερο μαγαζιά όπου οι πελάτες απλά παραλαμβάνουν τα αντικείμενα που χρειάζονται και η πληρωμή λαμβάνεται άμεσα. Η είσοδος σε αυτά τα μαγαζιά απαιτεί την χρήση μιας εφαρμογής για κινητές συσκευές για την διαδικασία αυθεντικοποίησης και πληρωμής. Ωστόσο, η βιομετρική σάρωση μέσω καμερών αναγνώρισης προσώπου ή δακτυλικού αποτυπώματος μπορεί να αντικαταστήσει την ανάγκη ελέγχου στα καταστήματα χρησιμοποιώντας μια εφαρμογή για κινητά και να κάνει την εμπειρία του ψωνίσματος ακόμα πιο απρόσκοπτη [23].

Κεφάλαιο 6

6 Συμπεράσματα

Εν κατακλείδι, η διαδικασία αυθεντικοποίησης ενός χρήστη είναι ιδιαίτερα σημαντική και χρειάζεται αρκετή μελέτη έτσι ώστε να γίνεται όσο πιο απλή και αθόρυβη ενώ παράλληλα να διατηρείται ισχυρή ασφάλεια. Τα βιομετρικά στοιχειά του χρήστη, αποτελούν ένα ασφαλές μέσο ταυτοποίησης ενός χρήστη καθώς προσδιορίζουν μοναδικά έναν άνθρωπο. Απαραίτητη προϋπόθεση έτσι ώστε η βασική διαδικασία ελέγχου ταυτότητας να θεωρείται ασφαλές είναι τα βιομετρικά δεδομένα του χρήστη να αποθηκεύονται αποκλειστικά στη συσκευή και συγκεκριμένα σε ένα ασφαλές και μη εξαγώγιμο χώρο. Σε καμία περίπτωση δεν πρέπει αυτές οι ευαίσθητες πληροφορίες να εκτίθενται έξω από τη συσκευή. Επίσης, η δυνατότητα αναπαραγωγής των πληροφοριών αυτών σε άλλες συσκευές είναι μια αντισταθμιζόμενη σχέση ασφαλείας μεταξύ των πλεονεκτημάτων που προσφέρει η κρυπτογραφία δημοσίου κλειδιού και των συνεπειών που επιφέρονται με τη χρήση κωδικών πρόσβασης. Ωστόσο, άλλες τεχνολογίες που αναπτύσσονται προσπαθούν να λύσουν αυτή τη διαφορά χρησιμοποιώντας διαφορετικές τεχνικές υλοποίησης απαλλάσσοντας έτσι εντελώς από την χρήση κωδικών πρόσβασης.

Βιβλιογραφία

[1] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," in *2012 IEEE Symposium on Security and Privacy*, pp. 553–567. doi: 10.1109/SP.2012.44.

[2] S. Pearman *et al.*, "Let's go in for a closer look: Observing passwords in their natural habitat," 2017, pp. 295–310. doi: 10.1145/3133956.3133973.

[3] J. Colnago *et al.*, "'It's not actually that horrible': Exploring adoption of two-factor authentication at a university," 2018, pp. 1–11. doi: 10.1145/3173574.3174030.

[4] J. Dutson, D. Allen, D. Eggett, and K. Seamons, "Don't Punish all of us: Measuring User Attitudes about TwoFactor Authentication," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 119–128. doi: 10.1109/EuroSPW.2019.00020.

[5] D. Han, Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Proximity-proof: Secure and usable mobile two-factor authentication," 2018, pp. 401–415. doi: 10.1145/3241539.3241574.

[6] E. M. Redmiles, M. L. Mazurek, and E. Liu, "You want me to do what? A design study of Two-Factor authentication messages," Jul. 2017. [Online]. Available: https://www.usenix.org/conference/soups2017/workshop-program/way2017/redmiles

[7] OneLogin, "Passwordless Authentication," *OneLogin*, 2020. https://www.onelogin.com/learn/passwordless-authentication

[8] N. A. G. Arachchilage, "Fingerprint login should be a secure defence for our data, but most of us don't use it properly," *The Conversation*. https://theconversation.com/fingerprint-login-should-be-a-secure-defence-for-our-data-but-most-of-us-dont-use-it-properly-127442

[9] A. Turner, "How Many People Have Smartphones Worldwide (Feb 2022)," *bankmycell*, 2018. https://www.bankmycell.com/blog/how-many-phones-are-in-the-world#:~:text=In%20total%2C%20the%20number%20of

[10] T. Sloane, "By 2024, How Many Smartphone Owners Will Use Biometrics?," *PaymentsJournal*, Jun. 04, 2020. https://www.paymentsjournal.com/by-2024-how-many-smartphone-owners-will-use-biometrics/#:~:text=Currently%2C%20Mercator%20estimates%20that%2041

[11] Android Developers, "What's new in Android security (M and N Version) - Google I/O 2016," May 19, 2016. https://youtu.be/XZzLjllizYs (accessed May 02, 2022).

[12] L. Carmona de Souza, "Is logging into your smartphone, websites, or apps with a fingerprint secure?," *blog.avast.com*, Feb. 18, 2015. https://blog.avast.com/2015/02/18/is-logging-into-your-smartphone-websites-or-apps-with-a-fingerprint-secure/ (accessed May 02, 2022).

[13] D. Balfanz *et al.*, Eds., "Web Authentication: An API for accessing Public Key Credentials - Level 1," *www.w3.org*, Mar. 04, 2019. https://www.w3.org/TR/2019/REC-webauthn-1-20190304/

[14] A. J. Menezes, P. C, and S. A. Vanstone, *Handbook of applied cryptography*. Boca Raton: Crc Press, 2001.

[15] Oracle Corporation and/or its affiliates, “Public Keys, Private Keys, and Certificates (Configuring Java CAPS for SSL Support),” *docs.oracle.com*, 2010. https://docs.oracle.com/cd/E19509-01/820-3503/ggbgc/index.html#:~:text=The%20owner%20of%20the%20key (accessed May 02, 2022).

[16] S. Nakov, “Elliptic Curve Cryptography (ECC),” *cryptobook.nakov.com*, Nov. 2018. https://cryptobook.nakov.com/asymmetric-key-ciphers/elliptic-curve-cryptography-ecc

[17] Android Developers, “Android keystore system,” *Android Developers*. https://developer.android.com/training/articles/keystore

[18] University of Southampton, “The Android Keystore,” *FutureLearn*, 2017. https://www.futurelearn.com/info/courses/secure-android-app-development/0/steps/21602 (accessed May 02, 2022).

[19] Android Developers, “Show a biometric authentication dialog,” *Android Developers*. https://developer.android.com/training/sign-in/biometric-auth

[20] I. Damier, “Using BiometricPrompt with CryptoObject: how and why,” *Android Developers*, Aug. 20, 2020. https://medium.com/androiddevelopers/using-biometricprompt-with-cryptoobject-how-and-why-aace500ccdb7 (accessed May 02, 2022).

[21] S. Raman, “Guide to Web Authentication,” *Guide to Web Authentication*. https://webauthn.guide/ (accessed May 02, 2022).

[22] J. Hodges, J. C. Jones, M. B. Jones, A. Kumar, and E. Lundberg, “Web Authentication: An API for accessing Public Key Credentials - Level 2,” *www.w3.org*, Apr. 08, 2021. https://www.w3.org/TR/webauthn-2/

[23] Veriff, “The Future of Biometric Technology — What’s Next?,” *Veriff*, Mar. 14, 2022. https://www.veriff.com/blog/the-future-of-biometric-technology

[24] D. Robb, “The Future of Biometrics in the Workplace,” *SHRM*, Feb. 22, 2022. https://www.shrm.org/resourcesandtools/hr-topics/technology/pages/the-future-biometrics-workplace.aspx

Παράρτημα Α

Ο κώδικας για την εφαρμογή βρίσκεται διαθέσιμος στο GitHub chrls5/biometric_authentication_thesis (github.com)