ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΧΩΡΙΣ ΣΥΝΘΗΜΑΤΙΚΑ

ΑΙΚΑΤΕΡΙΝΗ ΕΛΕΥΘΕΡΙΟΥ
713242017009

ΙΩΑΝΝΑ ΚΑΝΤΖΑΒΕΛΟΥ, ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΧΩΡΙΣΣΥΝΘΗΜΑΤΙΚΑ

ΑΙΚΑΤΕΡΙΝΗ ΕΛΕΥΘΕΡΙΟΥ
Α.Μ. 713242017009

Εισηγητής:

ΙΩΑΝΝΑ ΚΑΝΤΖΑΒΕΛΟΥ, ΕΠΙΚΟΥΡΗ ΚΑΘΗΓΗΤΡΙΑ

Ioanna Kantzavelou
Digitally signed by Ioanna Kantzavelou
Date: 2024.04.14 22:10:15 +03'00'

Εξεταστική Επιτροπή:

ΠΑΝΑΓΙΩΤΗΣ ΚΑΡΚΑΖΗΣ, ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΗΣ

Panagiotis Karkazis
Digitally signed by Panagiotis Karkazis
Date: 2024.04.15 13:15:26 +03'00'

ΧΡΙΣΤΙΝΑ ΓΕΩΡΓΟΥΛΑΚΗ, ΕΔΙΠ

Christina Georgoulaki
Digitally signed by Christina Georgoulaki
Date: 2024.04.15 19:30:45 +03'00'

Ημερομηνία εξέτασης 21/3/2024

ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΧΩΡΙΣ ΣΥΝΘΗΜΑΤΙΚΑ

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Η κάτωθι υπογεγραμμένη Ελευθερίου Αικατερίνη του Χρήστου, με αριθμό μητρώου 713242017009 φοιτήτρια του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της επεξεργασίας κειμένου. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.

Ακόμα θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση κατά τη διάρκεια των σπουδών μου.

ΠΕΡΙΛΗΨΗ

Στη σημερινή εποχή της ψηφιοποίησης , οι άνθρωποι έρχονται αντιμέτωποι με την πρόκληση της διαχείρισης μιας μεγάλης πληθώρας λογαριασμών σε διαφορετικές πλατφόρμες- υπηρεσίες. Κάθε μία από αυτές τις υπηρεσίες απαιτεί για τη δημιουργία λογαριασμού έναν κωδικό πρόσβασης ώστε να πραγματοποιηθεί η ταυτοποίηση. Οι κωδικοί ταυτοποίησης είναι ένας ευρέως διαδεδομένος τρόπος αυθεντικοποίησης. Πλέον οι διάφοροι χρήστες έχουν αποκτήσει μια οικειότητα τέτοιου βεληνεκούς με τα συνθηματικά, η οποία τους αποτρέπει από την αναζήτηση μιας νέας μεθόδου ταυτοποίησης, που μπορεί να τους προσδίδει μάλιστα και περισσότερα πλεονεκτήματα.

Ωστόσο, αν και τα συνθηματικά έχουν κατακτήσει την πρώτη θέση ανάμεσα στις ποικίλες μεθόδους αυθεντικοποίησης, δεν είναι άκρως ασφαλή και παρουσιάζουν κάποιες ευπάθειες. Ως εκ τούτου έχει σημειωθεί μια αυξανόμενη ζήτηση για νέες μεθόδους οι οποίες με τη σειρά τους θα δίνουν τη δυνατότητα ταυτοποίησης χωρίς να χρειάζεται να καταχωρηθεί κάποιος κωδικός πρόσβασης. Κατ' αυτόν τον τρόπο η πρόσβαση στις υπηρεσίες θα είναι πιο ασφαλής.

Στην διπλωματική αυτή εργασία θα παρουσιαστούν εν συντομία οι περιορισμοί που έχουν οι κωδικοί ως προς την ασφάλεια και θα διερευνηθεί το κίνητρο πίσω από την αναζήτηση ταυτοποίησης χωρίς τη χρήση κωδικών πρόσβασης. Θα παρουσιαστούν τρόποι εναλλακτικών μεθόδων που θα ενισχύουν τον έλεγχο της ταυτοποίησης αλλά και της προστασίας του χρήστη, ενώ παράλληλα θα απλοποιήσουν την εμπειρία των χρηστών με τις υπηρεσίες.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Κυβερνοασφάλεια
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: συνθηματικά, αυθεντικοποίηση, βιομετρικά χαρακτηριστικά, disCodeV2.

ABSTRACT

In today's age of digitization, people are faced with the challenge of managing several accounts on different platforms- services. Each of these services requires a password for account creation and subsequent authentication. Passwords are a widely used method of authentication. So now countless users have acquired such a familiarity with passwords and as a result it prevents them from looking for a new method of identification, which may even give them more advantages.

However, even if passwords have conquered the first place among various methods of authentication, they are not highly secure and have numerous vulnerabilities. Therefore, there is a growing demand for new methods which will enable authentication without having to enter a password (password- less authentication). In this way access to the services will be more secure.

This citation will briefly present the security limitations of passwords and explore the motivation behind seeking authentication without the use of passwords. Ways of alternative authentication methods will be presented, that will strengthen authentication and user's protection and simplify user's experience with the services.

SCIENTIFIC AREA: Cybersecurity
KEYWORDS: passwords, authentication, biometric characteristics, disCodeV2.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΣΥΝΤΟΜΕΥΣΗ	ΣΗΜΑΣΙΑ
2FA	TWO FACTOR AUTHENTICATION
FRS	FIRNGERPRINT RECOGNITION SYSTEM
OTP	ONE TIME PASSWORD
VPN	VIRTUAL PRIVATE NETWORKING
ECDH	ELLIPTIC CURVE DIFFIE – HELLMAN
TOTP	TIME – BASED ONE - TIME PASSWORD
HOTP	HASH – BASED ONE - TIME PASSWORD
MFA	MULTI - FACTOR AUTHENTICATION
MD5	MESSAGE DIGEST ALGORITHM
BSV	BIOMETRIC SIGNATURE VERIFICATION
FPGA	FIELD PROGRAMMABLE GATE ARRAY
VFPU	VECTOR FLOATING POINT UNIT
HMM	HIDDEN MARKOV MODEL
AMD	ADVANCED MICRO DEVICES
ERD	ENTITY RELATIONSHIP DIAGRAM
MPI	MESSAGE PASSING INTERFACE

ΚΕΦΑΛΑΙΟ 1: ΕΥΠΑΘΕΙΕΣ ΣΤΑ ΣΥΝΘΗΜΑΤΙΚΑ

1.1 Εισαγωγή

Στο πρώτο κεφάλαιο της διπλωματικής εργασίας αναλύονται οι ευπάθειες των συνθηματικών, αναφέρονται οι ποικίλες επιθέσεις που εκμεταλλεύονται τα τρωτά σημεία των συνθηματικών καθώς επίσης και γνωστά περιστατικά τέτοιων επιθέσεων.

1.2 Συνθηματικά και ισχυρά συνθηματικά

Τα συνθηματικά αποτελούν παραδοσιακό τρόπο επαλήθευσης της ταυτότητας των χρηστών σε πάρα πολλές υπηρεσίες. Λόγω της ευκολίας που προσδίδουν στη διαδικασία αυθεντικοποίησης οι άνθρωποι έχουν επαναπαυτεί με τη χρήση των συνθηματικών ως μέσο πρόσβασης σε διάφορες πλατφόρμες και υπηρεσίες. Οι κωδικοί πρόσβασης αποτελούν μια ακολουθία συμβόλων , τα σύμβολα των οποίων μπορεί να είναι χαρακτήρες είτε κεφαλαίοι είτε πεζοί , μπορεί να είναι αριθμοί αλλά και σύμβολα σύμφωνα πάντα με την εκάστοτε υπηρεσία και τις προϋποθέσεις που θέτει στον χρήστη κατά τον σχηματισμό του συνθηματικού. Προφανώς η ακολουθία αυτή θεωρείται μυστική.

Τα συνθηματικά έχουν ως στόχο να προστατεύσουν τους λογαριασμούς και τα δεδομένα ενός ατόμου από διάφορους εισβολείς που επιδιώκουν τη μη εξουσιοδοτημένη πρόσβαση. Τα δεδομένα αυτά ποικίλουν ανάλογα με το είδος της υπηρεσίας στην οποία προσπαθεί να συνδεθεί ο χρήστης. Κάθε φορά που ένα άτομο επιθυμεί να εγγραφτεί σε κάποια νέα υπηρεσία η οποία χρησιμοποιεί ως μέθοδο ταυτοποίησης τα συνθηματικά, εμφανίζεται μια φόρμα στην οποία ο νέος αυτός χρήστης καλείται να εισάγει κάποιο ψευδώνυμο και να συμπληρώσει έναν κωδικό.

Τα στοιχεία αυτά αποτελούν ουσιαστικά και το μέσο επαλήθευσης της ταυτότητας του χρήστη. Με αυτόν τον τρόπο την επόμενη φορά αλλά και σε μεταγενέστερες προσπάθειες το άτομο θα πρέπει να συμπληρώσει εκ νέου αυτές τις πληροφορίες ώστε να πραγματοποιηθεί σύγκριση στα εκάστοτε δεδομένα που εισάγει αλλά και στα πρωταρχικά που ο ίδιος όρισε ως διαπιστευτήρια. Στη περίπτωση που αυτά τα 2 συμπίπτουν τότε ο χρήστης ταυτοποιείται και έχει πλέον τη δυνατότητα να μεταβεί στην εκάστοτε υπηρεσία.

Οι κωδικοί πρόσβασης όμως θα πρέπει να πληρούν κάποιες προϋποθέσεις για να γίνουν αποδεκτοί από το εκάστοτε σύστημα. Το πεδίο ορισμού ενός συνθηματικού μπορεί να διαφέρει από υπηρεσία σε υπηρεσία αναλόγως με τις ανάγκες της κάθε μίας. Ωστόσο, κρίνεται αδιαμφισβήτητα σημαντικό να δημιουργηθεί ένας ισχυρός κωδικός ταυτοποίησης, ο οποίος θα προσφέρει τόσο αξιοπιστία όσο και ασφάλεια στον κάτοχο του.

Υπάρχουν μερικές βασικές προδιαγραφές τις οποίες ο χρήστης θα ήταν καλό να λάβει υπόψη του κατά τον σχηματισμό συνθηματικού. Οι βασικότερες από αυτές

είναι πλήθος των συμβόλων που μπορούν να χρησιμοποιηθούν , το μήκος της ακολουθίας που επιτρέπεται να συμπληρωθεί , το περιεχόμενο του κωδικού να μην αποτελεί κάποια λέξη που ανήκει στο λεξικό, να μην περιλαμβάνει γνωστά προσωπικά δεδομένα του χρήστη όπως για παράδειγμα το όνομα ή την ημερομηνία γέννησης, να μην γίνεται επαναχρησιμοποίηση του ίδιου συνθηματικού είτε στην ίδια υπηρεσία είτε σε άλλες και τέλος να είναι μια ακολουθία εύκολη μεν για τον ίδιο τον χρήστη αλλά δύσκολη ως προς τη σύλληψη από τον εισβολέα [5].

Το πλήθος των διαθέσιμων συμβόλων που έχει ορίσει η υπηρεσία είναι ένας βασικός παράγοντας για τη δημιουργία ενός ισχυρού κωδικού επαλήθευσης. Όσο πιο μεγάλο είναι το πλήθος των συμβόλων που μπορούν να αξιοποιηθούν τόσο περισσότερο αυξάνονται και οι συνδυασμοί που μπορούν να προκύψουν αποδίδοντας πολυπλοκότητα στο τελικό αποτέλεσμα. Τα σύμβολα μπορεί να περιλαμβάνουν χαρακτήρες σε διάφορες γλώσσες είτε αυτοί είναι πεζοί είτε κεφαλαίοι , αριθμούς είτε διάφορα σύμβολα αρκεί να είναι επιτρεπτά από την υπηρεσία. Φυσικά αυτό δε σημαίνει ότι κάποιος χρήστης θα προσπαθήσει να δημιουργήσει έναν κωδικό πρόσβασης χρησιμοποιώντας όλα τα διαθέσιμα σύμβολα. Με αυτόν τον τρόπο μειώνονται και οι πιθανότητες των επιθέσεων εξαντλητική αναζήτησης (επιθέσεις brute-force).

Ακριβώς το ίδιο ισχύει και για το μήκος της ακολουθίας. Όσο πιο μεγάλο είναι το μήκος της ακολουθίας που συμπληρώνεται και χρησιμοποιεί μια πληθώρα από τα διαθέσιμα σύμβολα τόσο πιο ισχυρό είναι το συνθηματικό που θα σχηματιστεί. Αυτό εν γένει σημαίνει πως κατά την προσπάθεια κάποιας εισβολής από κάποιο κακόβουλο άτομο ο χρόνος που θα χρειαστεί προκειμένου να σπάσει τον κωδικό θα έχει εκτιναχθεί πιθανότατα σε μερικά εκατομμύρια χρόνια. Κάποιες υπηρεσίες έχουν προβλέψει και έχουν ορίσει το ελάχιστο μήκος του κωδικού ώστε να γίνει αποδεκτός ενώ κάποιες έχουν ορίσει και το μέγιστο μήκος που μπορεί να καταχωρηθεί. Για παράδειγμα ένα συνθηματικό τουλάχιστον 12 χαρακτήρων θεωρείται αρκετά ισχυρό. Βέβαια μόνο το μήκος της ακολουθίας δεν προσφέρει την απαραίτητη ασφάλεια.

Αντίστοιχα θα πρέπει οι χρήστες να αποφύγουν να χρησιμοποιήσουν αυτούσιες λέξεις ως συνθηματικά και να προσπαθήσουν να αντικαταστήσουν κάποιους χαρακτήρες με άλλα σύμβολα όπως το «ε» στη θέση του οποίου θα μπορούσε να καταχωρηθεί το «3» κλπ. Επίσης ολόκληρες λέξεις θα μπορούσαν να αντικατασταθούν με σύμβολα. Ένα τέτοιο απλό παράδειγμα αποτελεί η λέξη «happy» η οποία θα μπορούσε να αποτυπωθεί και ως «:) » (βάσει των γνωστών ιδεογραμμάτων που χρησιμοποιούνται κυρίως στα μέσα κοινωνικής δικτύωσης ή άλλα μέσα γραπτής επικοινωνίας). Επίσης δε θα πρέπει να γίνεται χρήση των προσωπικών δεδομένων καθώς είναι πιο εύκολο να προβλεφθούν.

Σημαντικός παράγοντας επίσης είναι και η επαναχρησιμοποίηση των κωδικών. Μια καλή τακτική που πρέπει να υιοθετήσουν οι χρήστες είναι η δημιουργία κάθε φορά ενός νέου κωδικού διαφορετικού από τους άλλους που είτε έχουν ήδη χρησιμοποιήσει στην ίδια υπηρεσία είτε από εκείνους που χρησιμοποιούν στις υπόλοιπες, ενώ παράλληλα η τακτική ανανέωση αυτών προσθέτει μια επιπλέον ασφάλεια. Με αυτόν τον τρόπο σε περίπτωση που για τον οποιανδήποτε λόγο

κάποιος αποκτήσει μη εξουσιοδοτημένη πρόσβαση στον έναν λογαριασμό του να μην κινδυνεύσουν και οι υπόλοιποι.

Σε αυτό το σημείο θα πρέπει να τονιστεί ένας βασικός παράγοντας που θα πρέπει να ακολουθούν όλοι οι χρήστες προκειμένου να κρατήσουν ασφαλείς τους κωδικούς τους και κατ΄ επέκταση τους λογαριασμούς τους. Ένας χρήστης μπορεί να σχηματίσει ένα ισχυρό συνθηματικό ακολουθώντας τις παραπάνω συμβουλές. Ωστόσο, ένας ισχυρός κωδικός χάνει την ασφάλεια που μπορεί να προσφέρει εάν δημοσιοποιηθεί. Είναι άκρως σημαντικό τέτοιου είδους πληροφορίες να διατηρούνται υπό άκρα μυστικότητα.

Έστω ένα σενάριο για το οποίο ένας χρήστης επιθυμεί να εγγραφεί και να συνδεθεί στην υπηρεσία Υ. Για τις δικές της ανάγκες η υπηρεσία έχει ορίσει κάποιες προϋποθέσεις ώστε ο κωδικός που θα δοθεί να γίνει αποδεκτός. Υποθέτοντας πως διαθέσιμοι χαρακτήρες είναι χαρακτήρες του λατινικού αλφαβήτου (περιλαμβάνει τόσο πεζούς όσο και κεφαλαίους χαρακτήρες), αριθμούς αλλά και τα σύμβολα: «!,@,#,$,%,^,&,* » [Πίνακας 1].

ΣΥΝΟΛΙΚΑ ΣΥΜΒΟΛΑ	ΠΛΗΘΟΣ
Κεφαλαίοι χαρακτήρες (A - Z)	26
Πεζοί χαρακτήρες (a - z)	26
Ειδικά Σύμβολα (!,@,#,$,%,^,&,*)	8
Αριθμοί (0 – 9)	10

Πίνακας 1. Πιθανά σύμβολα δημιουργίας συνθηματικού

Από τον παραπάνω πίνακα προκύπτει ότι οι διαθέσιμοι χαρακτήρες είναι : 26 + 26 + 8 + 10 = 70. Αν τώρα γίνει η υπόθεση ότι ο κωδικός μπορεί να έχει μήκος n = 16 τότε προκύπτει ότι οι διαφορετικοί πιθανοί συνδυασμοί είναι $70^{n} = 70^{16} =$ 3.32329306E+29.

Πολύ συχνά και σε πολλές υπηρεσίες ακόμη και σήμερα παρατηρείται πως οι προϋποθέσεις που τίθενται σχετικά με τα συνθηματικά αφήνουν το περιθώριο στους χρήστες να σχηματίσουν κωδικούς που δεν πληρούν τα βασικά στάδια που απαιτούνται για τη διαμόρφωση ενός ισχυρού κωδικού ταυτοποίησης. Επομένως αν αυτό ισχύει για την υπηρεσία Υ, τότε είναι πολύ πιθανό οι χρήστες να επιλέξουν κάτι εύκολο και γρήγορο προκειμένου να επαληθεύσουν την ταυτότητά τους.

Αν λοιπόν στην παραπάνω υπόθεση τα σύμβολα και το μήκος μειωθούν παρατηρείται τεράστια πτώση στους πιθανούς συνδυασμούς. Στην περίπτωση δηλαδή που το μήκος είναι n = 6 τότε προκύπτουν 70^6 = 117.649.000.000 διαφορετικοί συνδυασμοί και αν τα διαθέσιμα σύμβολα είναι μόνο οι χαρακτήρες και οι αριθμοί τότε έχουμε συνδυασμούς της τάξης των 62^6 = 56.800.235.584. Γίνεται επομένως αντιληπτό πως όσο πιο μικρός σε μήκος και πολυπλοκότητα είναι ένας κωδικός τόσο πιο γρήγορα θα σπάσει για παράδειγμα σε μια επίθεση εξαντλητικής αναζήτησης (brute-force).

Η μαθηματική σχέση που συνδέει τις πιθανότητες που έχει κάθε χαρακτήρας να συμπεριλαμβάνεται στον κωδικό , το μήκος του συνθηματικού και τις υποθέσεις ακολουθεί παρακάτω:

C = (m^n)/2 ,όπου C οι υποθέσεις (δηλαδή οι πιθανοί συνδυασμοί των συνθηματικών), n το μήκος του κωδικού και m ο αριθμός των διαφορετικών χαρακτήρων που είναι αποδεκτοί (δηλαδή αν μπορούν να χρησιμοποιηθούν γράμματα ή για παράδειγμα ειδικοί χαρακτήρες κλπ).

Οι επιθέσεις εξαντλητική επίθεσης έχουν ως στόχο όπως είναι φανερό τις εξαντλητικές αναζητήσεις των συνδυασμών μέχρις ότου να βρεθεί ο σωστός κωδικός που θα δώσει μη εξουσιοδοτημένη πρόσβαση σε κάποιον λογαριασμό. Βάσει του παραπάνω σεναρίου λοιπόν, αν μια υπηρεσία επιτρέπει στον χρήστη να ορίσει ως κωδικό αυθεντικοποίησης μια συμβολοσειρά από 7 έως και 14 χαρακτήρες και αφήνει το περιθώριο επιλογής στον χρήστη να χρησιμοποιήσει όποια από τα διαθέσιμα σύμβολα εκείνος επιθυμεί υπάρχει μεγάλος κίνδυνος να σχηματιστεί αδύναμο συνθηματικό.

Στη συνέχεια θα παρουσιαστούν 3 πιθανά σενάρια με διαφορετικούς κωδικούς ταυτοποίησης και θα χρησιμοποιηθεί ένα διαδικτυακό εργαλείο [14] για τον υπολογισμό των ωρών που χρειάζονται προκειμένου να «σπάσει» το συνθηματικό. Στο εργαλείο αυτό έχει επιλεχθεί ως κρυπτογραφικός αλγόριθμος ο MD5 (Message Digest Algorithm 5) με ταχύτητα κατακερματισμού ορισμένη σε 2071,5 MH/s (δηλαδή 2.071.500.000 κατακερματισμούς το δευτερόλεπτο). Ο πίνακας [Πίνακας1] περιγράφει τους αποδεκτούς χαρακτήρες που μπορούν να χρησιμοποιηθούν.

ΣΕΝΑΡΙΟ 1^{Ο} :

Αν ένας χρήστης Α καταχωρίσει τον κωδικό «password » τότε προκύπτουν τα εξής δεδομένα:

1. Το συνθηματικό έχει μήκος 8 χαρακτήρες.
2. Χρησιμοποιούνται μόνο πεζά γράμματα => Άρα οι πιθανοί χαρακτήρες είναι 26

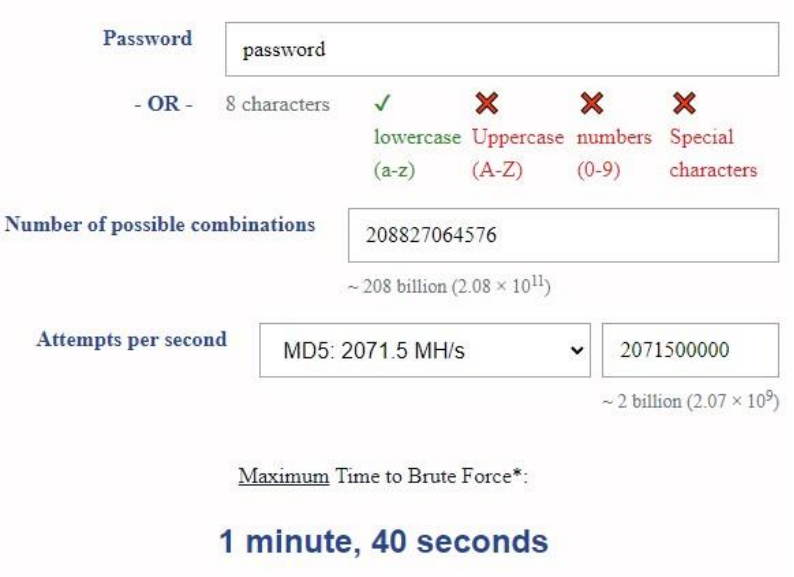
Άρα οι πιθανοί συνδυασμοί είναι 26^8 = 208.827.064.576.

Ο υπολογισμός τους χρονικού διαστήματος που θα διαρκέσει η επίθεση δίνεται από τον τύπο : t = C / u όπου t είναι ο χρόνος που διαρκεί η εξαντλητική αναζήτηση, C οι πιθανοί συνδυασμοί και u η ταχύτητα του κατακερματισμού.

Επομένως βάσει του τύπου το αποτέλεσμα που προκύπτει είναι :

t = C / u = 208.827.064.576 / 2.071.500.000 = 100,77 sec (δηλαδή περίπου 1 min και 40,77 sec).

Προφανώς γίνεται αντιληπτό πως ένα τέτοιου είδους συνθηματικό προκειμένου είναι πολύ εύκολο να σπάσει και κατ' επέκταση ο επιτιθέμενος να αποκτήσει μη εξουσιοδοτημένη πρόσβαση. Ακολουθεί εικόνα με τα αποτελέσματα που έδωσε και το εργαλείο



Εικόνα 1. Επίθεση εξαντλητικής αναζήτησης του συνθηματικού "password"

ΣΕΝΑΡΙΟ 2^{Ο} :

Σε αυτό το σενάριο θα δοθεί ένα περίπλοκο συνθηματικό. Έστω λοιπόν πως το συνθηματικό αυτή τη φορά είναι το «As1D^_^@#84e0$». Τα δεδομένα που είναι τα εξής :

1. Το συνθηματικό έχει μήκος 14 χαρακτήρες.
2. Χρησιμοποιούνται :
    - πεζά γράμματα => Άρα οι πιθανοί χαρακτήρες για τα πεζά είναι 26
    - κεφαλαία γράμματα => Άρα οι πιθανοί χαρακτήρες για τα κεφαλαία είναι 26
    - αριθμοί => Άρα το πλήθος των πιθανών αριθμών είναι 10 (0-9)
    - ειδικοί χαρακτήρες/σύμβολα => Έχει οριστεί ένα πλήθος 10 συμβόλων

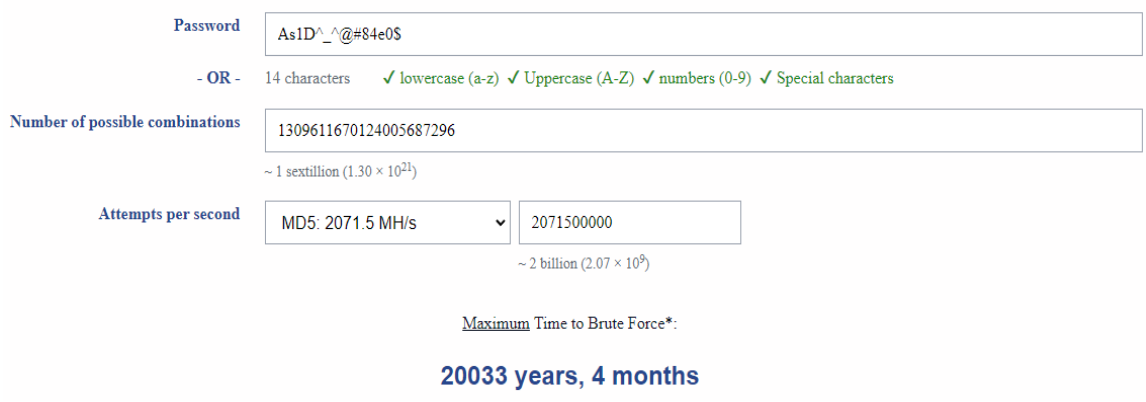
Στην περίπτωση αυτή ο υπολογισμός των πιθανών συνδυασμός προκύπτει από το άθροισμα κάθε διαφορετικού τύπου (πεζά γράμματα, αριθμοί κλπ) υψωμένο σε δύναμη ίση με το μήκος της συμβολοσειράς. Άρα οι πιθανοί συνδυασμοί είναι $26^{14}+26^{14}+10^{14}+10^{14}$= 1.309.611.670.124.005.687.296.

Ο υπολογισμός τους χρονικού διαστήματος που θα διαρκέσει η επίθεση δίνεται από τον τύπο : t = C / u όπου t είναι ο χρόνος που διαρκεί η εξαντλητική αναζήτηση, C οι πιθανοί συνδυασμοί και u η ταχύτητα του κατακερματισμού.

Επομένως βάσει του τύπου το αποτέλεσμα που προκύπτει είναι :

t = C / u = 1.309.611.670.124.005.687.296/ 2.071.500.000 = 632.204.523.352,163015832005792903 69sec (δηλαδή περίπου 20.033 χρόνια).

Ακολουθεί εικόνα με τα αποτελέσματα.



Εικόνα 2. Επίθεση εξαντλητικής αναζήτησης του συνθηματικού «As1D^_^@#84e0$»

Είναι ξεκάθαρο πως το συνθηματικό αυτό λόγω της πολυπλοκότητας του θα χρειαστεί εκατομμύρια χρόνια για να σπάσει αντίθετα με το συνθηματικό που παρουσιάστηκε στο προηγούμενο σενάριο. Ωστόσο η μεγάλη ποικιλία των χαρακτήρων που θα επιλεχθούν δεν είναι ο μοναδικός τρόπος επίτευξης ενός ισχυρού συνθηματικού.

ΣΕΝΑΡΙΟ 3^{Ο} :

Στο τρίτο και τελευταίο σενάριο ο χρήστης έστω ότι χρησιμοποιεί ένα συνθηματικό μεγάλου μήκους αλλά μόνο με χαρακτήρες (δε θα συμπεριληφθούν αριθμοί ή ειδικά σύμβολα). Ο κωδικός είναι ο «ZkPeVssLkwsOfx». Τα δεδομένα που είναι τα εξής :

3. Το συνθηματικό έχει επίσης μήκος 14 χαρακτήρες (όπως και του σεναρίου 2).
4. Χρησιμοποιούνται :
    - πεζά γράμματα => Άρα οι πιθανοί χαρακτήρες για τα πεζά είναι 26
    - κεφαλαία γράμματα => Άρα οι πιθανοί χαρακτήρες για τα κεφαλαία είναι 26

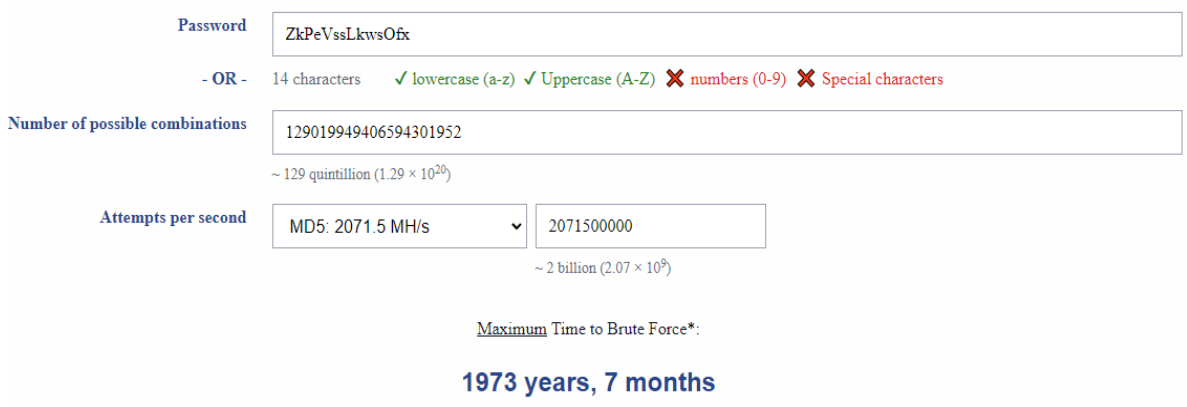
Άρα οι πιθανοί συνδυασμοί είναι $26^{14}+26^{14}$ =129.019.949.406.594.301.952.

Ο υπολογισμός τους χρονικού διαστήματος που θα διαρκέσει η επίθεση δίνεται από τον τύπο : t = C / u όπου t είναι ο χρόνος που διαρκεί η εξαντλητική αναζήτηση, C οι πιθανοί συνδυασμοί και u η ταχύτητα του κατακερματισμού.

Επομένως βάσει του τύπου το αποτέλεσμα που προκύπτει είναι :

t = C / u = 129.019.949.406.594.301.952/ 2.071.500.000 το οποίο είναι περίπου 62.260.271.758,57358sec (δηλαδή περίπου 1974 χρόνια).

Μπορεί το συνθηματικό αυτό να μην είναι όσο ισχυρό από εκείνο του δεύτερου σεναρίου, ωστόσο προσφέρει μια ασφάλεια 1974 χρόνων. Επομένως και σε αυτή την περίπτωση το συνθηματικό θεωρείται ισχυρό παρόλο που δεν έχει χρησιμοποιηθεί όλο το εύρος των αποδεκτών τύπων. Ακολουθεί εικόνα με τα αποτελέσματα.



Εικόνα 3. Επίθεση εξαντλητικής αναζήτησης του συνθηματικού «ZkPeVssLkwsOfx»

1.3 Προβλήματα και ευπάθειες

Στη σημερινή εποχή η προσωπική ζωή των ανθρώπων ανεξαρτήτου ηλικίας είναι άρρηκτα συνδεδεμένη με τα συνθηματικά ταυτοποίησης (passwords). Οι κωδικοί αυθεντικοποίησης έχουν γίνει συνώνυμα της καθημερινότητας, καθώς ένα φυσικό πρόσωπο χρειάζεται να διαχειριστεί μια τεράστια πληθώρα ψηφιακών λογαριασμών, κάθε ένας από τους οποίους έχει και ένα συνθηματικό. Οι άνθρωποι προκειμένου να έχουν πρόσβαση σε ποικίλες υπηρεσίες είναι απαραίτητο να δημιουργήσουν ψηφιακούς χρήστες.

Μερικά απλά παραδείγματα τέτοιων υπηρεσιών που απαιτούν ταυτοποίηση είναι το ηλεκτρονικό ταχυδρομείο, όπου κάποιος μπορεί να διαχειρίζεται αρκετές ηλεκτρονικές διευθύνσεις είτε προσωπικές είτε εταιρικές , τα μέσα κοινωνικής δικτύωσης όπου επίσης κάποιος είναι δυνατό να έχει περισσότερους από έναν λογαριασμούς , οι εφαρμογές που σχετίζονται με τις τράπεζες , οι πλατφόρμες διαδικτυακών αγορών και ψυχαγωγίας , οι πλατφόρμες που έχουν σχέση με την εκπαίδευση και πολλές άλλες.

Τα συνθηματικά χρησιμοποιούνται ως δικλείδα ασφαλείας και προστασίας ευαίσθητων δεδομένων. Αποτελούν έναν πρωταρχικό ευρέως διαδεδομένο τρόπο ταυτοποίησης εδώ και καιρό σε ό,τι σχετίζεται τόσο με τον ψηφιακό τομέα αλλά και πέρα από αυτόν. Η χρήση κωδικών είναι ένας αρκετά οικείος τρόπος αφού οι άνθρωποι είναι ιδιαίτερα εξοικειωμένοι με τη χρήση τους. Είναι εύκολοι ως προς τη

δημιουργία καθώς επίσης και τη διαχείρισή τους και οι χρήστες έχουν άμεσο έλεγχο των λογαριασμών τους. Η διαδικασία της αυθεντικοποίησης μέσω αυτών δεν είναι ακριβή σαν μέθοδος και πλέον οι περισσότερες πλατφόρμες-υπηρεσίες την έχουν υιοθετήσει. Βέβαια η χρήση ενός password δε σημαίνει αυτομάτως ότι ο εκάστοτε λογαριασμός είναι προστατευμένος στο έπακρο.

Ολοένα και περισσότερο αμφισβητείται η αποτελεσματικότητά τους λόγω των τρωτών τους σημείων καθώς εμφανίζονται συνεχώς νέες και πιο εξελιγμένες απειλές στον κυβερνοχώρο. Παρακάτω θα αναλυθούν πιο λεπτομερώς τα τρωτά σημεία των κωδικών πρόσβασης, καθώς επίσης θα αποτυπωθούν σημαντικά σφάλματα και επιθέσεις οι οποίες με τη σειρά τους εκμεταλλεύτηκαν αυτές τις αδυναμίες προξενώντας σε πολλές περιπτώσεις σοβαρές επιπτώσεις.

Ένα βασικό ευάλωτο σημείο των κωδικών πρόσβασης είναι η χρήση αυτών με αδύναμη μορφή. Κάποιες υπηρεσίες αναφέρουν τον αριθμό των χαρακτήρων που χρειάζονται, αν μπορούν να χρησιμοποιηθούν μόνο αλφαριθμητικοί χαρακτήρες ή όχι, ώστε να δημιουργηθεί ένα ισχυρό συνθηματικό ενώ κάποιες άλλες όχι. Αυτό σημαίνει αυτόματα ότι ο σχηματισμός του συνθηματικού είναι εν μέρει στο χέρι του χρήστη. Ακριβώς για αυτόν τον λόγο υπάρχει πρόβλημα ως προς το πόσο ισχυρός θα είναι ένας κωδικός.

Κατά τη διαδικασία σχηματισμού ενός συνθηματικού οι χρήστες επιλέγουν κάτι σχετικό με την προσωπική τους ζωή όπως η ημερομηνία γέννησης, ώστε να το θυμούνται εύκολα, είτε χρησιμοποιούν το ελάχιστο των χαρακτήρων που προτείνεται, είτε οι χαρακτήρες είναι μια απλή κοινή λέξη ή ένας απλός συνδυασμός από αριθμούς και γενικώς είναι κάτι που εύκολα κάποιος θα μπορούσε να μαντέψει. Επιπλέον είναι πολύ φυσικό λόγω του πλήθους των λογαριασμών κάποιος να επιλέξει να καταχωρήσει το ίδιο password αν όχι σε όλους τους λογαριασμούς, σε ένα μεγάλο μέρος αυτών. Αυτομάτως εάν κάποιος καταφέρει να βρει τον κωδικό για τον λογαριασμό ενός άλλου ατόμου θα μπορεί να έχει πρόσβαση και σε άλλους λογαριασμούς του τελευταίου.

1.3.1 Δημοσιοποίηση συνθηματικών

Πέραν του προβλήματος που σχετίζεται όμως με τα αδύναμα συνθηματικά και την επανειλημμένη χρήση αυτών υπάρχει και η πιθανότητα του ανθρώπινου λάθους. Πιο συγκεκριμένα είναι πιθανό άθελά του κάποιος να δώσει πληροφορίες για τους κωδικούς του σε διάφορες υπηρεσίες, είτε να δώσει για δικούς του λόγους τον κωδικό σε άτομα που θεωρεί έμπιστα ή ακόμη και να τους καταγράψει σε κάποιες τοποθεσίες που δεν είναι ασφαλείς. Κάτι τέτοιο εν συνεχεία μπορεί να οδηγήσει σε μη εξουσιοδοτημένη πρόσβαση στους λογαριασμούς του.

Κάπως έτσι η εταιρεία SolarWinds δέχτηκε επίθεση. Η SolarWinds είναι μια αμερικανική εταιρεία η οποία αναπτύσσει λογισμικό το οποίο στη συνέχεια το χρησιμοποιούν επιχειρήσεις. Το λογισμικό που διατίθεται στην αγορά από αυτή την εταιρεία επικεντρώνεται σε συστήματα και εργαλεία διαχείρισης αλλά και παρακολούθησης των δικτύων. Στα τέλη του 2020 η εταιρεία αυτή δέχθηκε επίθεση

και το περιστατικό αυτό θεωρείται μία από τις μεγαλύτερες, αν όχι η μεγαλύτερη, παραβιάσεις της κυβερνοασφάλειας του 21ου αιώνα.

Η επίθεση αυτή επηρέασε χιλιάδες οργανισμούς και μάλιστα και της ίδιας της κυβέρνησης των Η.Π.Α. Η εταιρεία αυτή έχει συσχετιστεί με τον όρο «supply chain breach» (παραβίαση εφοδιαστικής αλυσίδας). Στις επιθέσεις αυτές οι επιτιθέμενοι βρίσκοντας τα τρωτά σημεία του εκάστοτε λογισμικού (για παράδειγμα μη προστατευμένες υποδομές στους διακομιστές ή μη ασφαλή πρωτόκολλα δικτύου) μεταβάλλουν τον πηγαίο κώδικα και στη συνέχεια αποκρύπτουν το κακόβουλο αυτό λογισμικό στις διαδικασίες της κατασκευής και της ενημέρωσης.

Το λογισμικό που δημιουργείται και κυκλοφορεί από αξιόπιστους προμηθευτές είτε είναι καινούριο είτε αποτελεί μια ενημέρωση κάποιου ήδη υπάρχοντος λογισμικού φέρει υπογραφές αλλά και πιστοποίηση. Φυσικά στις περιπτώσεις που έχουν γίνει επιθέσεις στην αλυσίδα του εφοδιασμού οι ίδιοι οι πωλητές (πιθανώς) δε γνωρίζουν την ύπαρξη του κακόβουλου λογισμικού στο προϊόν τους. Έτσι το λογισμικό αυτό κυκλοφορεί στο κοινό – πελάτες της εταιρείας. Στη συνέχεια και όταν το λογισμικό εγκαθίσταται στους πελάτες ξεκινά και υποκλέβει στοιχεία ανάλογα με τη λειτουργία για την οποία είναι κατασκευασμένο.

Οι επιθέσεις αυτές μπορούν να γίνουν εξίσου και σε λογισμικό αλλά και σε υλικό. Οι παραβιάσεις υλικού αφορούν επιθέσεις που σχετίζονται με τις φυσικές συσκευές όπως για παράδειγμα μονάδες USB, τηλέφωνα, tablets ή ακόμη και πληκτρολόγια. Οι συσκευές αυτές μολύνονται στο στάδιο κατασκευής τους και στην πορεία θα λειτουργήσουν ως μέσο ώστε να αποκτηθεί μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς ή στα διάφορα συστήματα που χρησιμοποιούν οι κάτοχοί τους.

Η επίθεση που σχετίζεται με την SolarWinds είναι παραβίαση λογισμικού. Η παραβίαση έγινε στο εργαλείο Orion υπεύθυνο για τη διαχείριση των δρομολογητών και των μεταγωγών εντός εταιρικών δικτύων. Μια ομάδα Ρώσων (ονομασία Nobelium) κατάφερε να διεισδύσει στο σύστημα Orion και να εισάγει κακόβουλο λογισμικό το οποίο στη συνέχεια εγκαταστάθηκε σε πολλές εταιρείες. Όπως απέδειξαν οι έρευνες 100 εταιρείες αλλά και κυβερνητικές υπηρεσίες όπως το Κέντρο Ελέγχου και Πρόληψης Ασθενειών των Η.Π.Α (CDC), το Υπουργείο Δικαιοσύνης των Η.Π.Α και πολλές άλλες.

Πως όμως ξεκίνησε όλη αυτή η επίθεση; Η απάντηση είναι ένας δημόσιος κωδικός πρόσβασης [33]. Το 2019 ένας ερευνητής ασφαλείας (Vinoth Kumar) ενημέρωσε πως στο δημόσιο αποθετήριο GitHub υπήρχε ο κωδικός πρόσβασης « solarwinds123 » με τον οποίο ο οποιοσδήποτε θα μπορούσε να αποκτήσει πρόσβαση στον διακομιστή της SolarWinds και να φορτώσει κάποιο κακόβουλο πρόγραμμα. Ωστόσο, ο διευθύνων σύμβουλος της εταιρείας υποστήριξε ότι ο κωδικός αυτός ήταν σε ισχύ μέχρι και το 2017 και επομένως δεν είχε καμία σχέση με την επίθεση. Μπορεί όντως οι επιτιθέμενοι να μην απόκτησαν πρόσβαση μέσω αυτού του συνθηματικού αλλά και μόνο η πρακτική της διάδοσής του, και γενικά η διάδοση του οποιουδήποτε κωδικού πρόσβασης από αμέλεια ή εκ προμελέτης εγκυμονεί μεγάλο ρίσκο [10].

Φυσικά υπάρχει η περίπτωση ένα άτομο να διαρρεύσει συγκεκριμένους κωδικούς πρόσβασης ηθελημένα για διάφορους λόγους που προφανώς θα τον επωφελήσουν. Ένα τέτοιο παράδειγμα είναι η μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς τρίτων από την εταιρεία Ticketmaster, μια εταιρεία πωλήσεων αλλά και διανομής εισιτηρίων με έδρα το Beverly Hills που ιδρύθηκε στις 02 Οκτωβρίου του 1976.

Η ίδια η εταιρεία παραδέχτηκε ότι είχε μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς οι οποίοι ανήκαν σε αντίπαλη εταιρεία [33]. Ένας πρώην υπάλληλος της αντίπαλης εταιρείας και νυν της Ticketmaster παρέδωσε στα στελέχη της τελευταίας κάποια εμπιστευτικά εσωτερικά έγγραφα τα οποία είχε κρατήσει ο ίδιος από τον προηγούμενο εργοδότη του καθώς επίσης και ποικίλα διαπιστευτήρια για τη σύνδεσή του σε πολλούς εταιρικούς λογαριασμούς.

Ο υπάλληλος μάλιστα τους έδειξε πως να χρησιμοποιήσουν αυτούς τους κλεμμένους κωδικούς προκειμένου να αποκτήσουν πρόσβαση στους λογαριασμούς που διαχειριζόταν η αντίπαλη εταιρεία για τις προπωλήσεις εισιτηρίων. Με αυτόν τον τρόπο η Ticketmaster αντλούσε δεδομένα παράνομα αλλά και επανειλημμένα με στόχο τη συλλογή των επιχειρηματικών πληροφοριών. Προφανώς η εταιρεία δε χρειάστηκε να προβεί σε καμία διαδικασία σχηματισμού κάποιας επίθεσης για να κλέψει αυτά τα δεδομένα. Χρειάστηκε μόνο ένα άτομο για να διαρρεύσει τους κωδικούς.

Γίνεται λοιπόν κατανοητό πως ακόμα και ένας ισχυρός κωδικός πρόσβασης από τη στιγμή που θα διαρρεύσει παύει να προσφέρει ασφάλεια στον κάτοχό του. Φυσικά κάτι τέτοιο θα μπορούσε να αποφευχθεί εάν οι κωδικοί στους οποίους είχε πρόσβαση ή γνώριζε ο υπάλληλος είχαν αλλάξει μόλις ο τελευταίος έφυγε από την εταιρεία. Επομένως οι παλιοί χρήστες με διαπιστευτήρια δε θα είναι σε θέση να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση.

Η εταιρεία δήλωσε ένοχη, στις αρχές του 2021, αφού κατηγορήθηκε για παραβιάσεις του νόμου που συμπεριλάμβαναν την απάτη και την κατάχρηση των υπολογιστών, εισβολή σε υπολογιστές για εμπορικά πλεονεκτήματα ή ιδιωτικό οικονομικό όφελος όπως επίσης και πολλές άλλες κατηγορίες που σχετίζονταν με την παράνομη αλλά και επανειλημμένη πρόσβαση στους λογαριασμούς της αντίπαλης εταιρείας. Έτσι η Ticketmaster κλήθηκε να πληρώσει ένα πρόστιμο των 10.000.000 δολαρίων ως πρόστιμο [34].

Ωστόσο ένα password μπορεί να κλαπεί χωρίς να έχει δοθεί κάποια πληροφορία από τον κάτοχο του λογαριασμού είτε αυτό είναι άμεσο είτε έμμεσο. Ένα απλό παράδειγμα όπου ο χρήστης δεν έχει δώσει ο ίδιος πληροφορίες είναι στις περιπτώσεις επιθέσεων phishing. Οι επιθέσεις phishing ουσιαστικά εξαπατούν τους χρήστες ώστε να αποκαλύψουν τα συνθηματικά τους χρησιμοποιώντας παραπλανητικά μηνύματα, emails ή ακόμη και ιστότοποι που μιμούνται τους νόμιμους. Εκμεταλλευόμενες τα τρωτά σημεία των ανθρώπων όπως η εμπιστοσύνη ή η επείγουσα ανάγκη αποκτούν πρόσβαση στα συνθήματα σύνδεσής τους σε λογαριασμούς.

Εκτός όμως από τις επιθέσεις που γίνονται στον ψηφιακό κόσμο , μπορούν να γίνουν και στον φυσικό κόσμο. Είναι πολύ πιθανό κάποιος να πληκτρολογήσει σε

κάποιο μηχάνημα- συσκευή αφής έναν κωδικό του και ένα δεύτερο άτομο να δει τα δακτυλικά αποτυπώματα , με τη βοήθεια του φωτός, που ο προηγούμενος άφησε (υπάρχουν βέβαια εφαρμογές που σε κάθε χρήση οι αριθμοί τοποθετούνται τυχαία μπροστά του επομένως δεν είναι εφικτό κάποιος κακόβουλος να βρει τον κωδικό). Το ίδιο ισχύει και στην περίπτωση που ο χρήστης για τον οποιονδήποτε λόγο δεν έχει καθαρά και στεγνά χέρια. Εάν επιδιώξει να εισάγει έναν κωδικό σε μια δημόσια συσκευή με οθόνη αφής πάνω στην οποία αποτυπώνονται οι αριθμοί τότε είναι πολύ πιθανό ο επόμενος που θα χρησιμοποιήσει τη συσκευή να μπορεί να διακρίνει τους αριθμούς που χρησιμοποιήθηκαν. Αυτό ακριβώς μπορεί να συμβεί και στην οθόνη του κινητού. Κάποιος θα μπορούσε να δει είτε το μοτίβο που έχει σχηματιστεί είτε τους αριθμούς που επιλέχθηκαν.

1.3.2 Κλοπή συνθηματικών με παρακολούθηση (Shoulder surfing)

Αυτή η επίθεση αναφέρεται κυρίως στην προσπάθεια που καταβάλει ο επιτιθέμενος προκειμένου να μάθει το συνθηματικό που χρησιμοποιεί το θύμα του παρακολουθώντας απλώς τις κινήσεις του. Είναι μια τακτική η οποία δεν προϋποθέτει κάποια εξεζητημένη γνώση [28]. Ο επιτιθέμενος επιδιώκει να υποκλέψει το προσωπικό συνθηματικό αναγνώρισης του θύματός του ώστε να λάβει μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς και διάφορα ευαίσθητα δεδομένα.

Ουσιαστικά ο ίδιος ο χρήστης έμμεσα αποκαλύπτει το συνθηματικό του εν αγνοία του φυσικά καθώς δεν έχει αντιληφθεί πως κάποιος τον παρακολουθεί. Πιο συγκεκριμένα ο επιτιθέμενος επιλέγει μια κοντινή απόσταση από το θύμα του (και κάπως έτσι έχει προκύψει και η ονομασία της επίθεσης- shoulder surfing με το σκεπτικό- ιδέα ότι ο θύτης στέκεται κοντά στον ώμο του θύματος του) σε κάποιον δημόσιο χώρο με πολύ κόσμο ώστε να μην γίνει αντιληπτός. Η βασική στρατηγική είναι η παρατήρηση των κινήσεων του θύματος όταν εκείνο εισάγει το συνθηματικό του σε ένα πληκτρολόγιο είτε αυτό είναι σε υπολογιστή είτε σε οθόνη αφής.

1.3.3 Επίθεση εξαντλητικής αναζήτησης

Βασικός στόχος αυτού του τύπου επιθέσεων είναι οι εξαντλητικές δοκιμές προκειμένου ο επιτιθέμενος να αποκτήσει πρόσβαση σε λογαριασμούς των θυμάτων του. Η μέθοδος αυτή διακρίνεται σε 3 κατηγορίες [28]:

1. Επίθεση εξαντλητικής αναζήτησης (Brute force attack)
2. Αντίστροφη επίθεση εξαντλητικής αναζήτησης (reverse brute force attack)
3. Επίθεση λεξικού (Dictionary attack)

Η επίθεση εξαντλητικής αναζήτησης αναφέρθηκε και στην προηγούμενη ενότητα [1.2]. Η μέθοδος αυτή ουσιαστικά αποτελεί μια εξαντλητική αναζήτηση συνθηματικών, όπως υποδηλώνει και η ονομασία της, που ταιριάζουν σε ένα όνομα χρήστη. Πιο συγκεκριμένα πραγματοποιούνται συστηματικά πάρα πολλές δοκιμές πιθανών συνδυασμών μέχρι να βρεθεί ο σωστός. Είναι ένα είδος επίθεσης το οποίο είναι σαφώς πιο πολύπλοκο από την επίθεση παρακολούθησης από κοντινή απόσταση. Ο επιτιθέμενος επιδιώκει να σπάσει τους κωδικούς πρόσβασης χρησιμοποιώντας όλους τους πιθανούς συνδυασμούς. Βασικό εργαλείο για την επίθεση αυτή αποτελεί μία λίστα η οποία περιλαμβάνει αυτούς τους συνδυασμούς.

Ωστόσο, μόνο η λίστα δεν είναι αρκετή για να δώσει τα επιθυμητά αποτελέσματα. Τα συνθηματικά πλέον δεν αποθηκεύονται στις διάφορες υπηρεσίες ως απλά κείμενα (plaintext). Αυτό γίνεται για λόγους ασφαλείας. Άρα οι επιτιθέμενοι θα πρέπει να έχουν αναγνωρίσει ποιος αλγόριθμος κατακερματισμού (hash algorithm) χρησιμοποιείται στην εκάστοτε περίπτωση και κατ' επέκταση να κάνουν τις απαραίτητες μετατροπές στο περιεχόμενο της παραγμένης λίστας. Ουσιαστικά θα πρέπει να κατακερματίσουν αυτά τα δεδομένα με τον ίδιο αλγόριθμο ώστε τα νέα δεδομένα που θα προκύψουν να ταιριάζουν ως προς τη μορφή με εκείνα του συστήματος.

Έχοντας ολοκληρώσει τα παραπάνω βήματα ξεκινάει η διαδικασία συνεχών δοκιμών και απορρίψεων-σφαλμάτων ώστε να εντοπιστεί το σωστό συνθηματικό. Κατά τις εξαντλητικές αυτές δοκιμές το ζητούμενο είναι να ταιριάξει ένα συνθηματικό της λίστας με εκείνο του χρήστη. Η διαδικασία προφανώς θα σταματήσει μόλις εντοπιστεί το σωστό συνθηματικό αλλιώς θα συνεχίσει να ψάχνει προχωρώντας στον επόμενο πιθανό συνδυασμό.

Παράδειγμα αποτελεί η επίθεση του συστήματος «WordPress». Η επίθεση έλαβε χώρα τον Απρίλιο του 2013. Χρησιμοποιήθηκαν περισσότεροι από 90.000 εξυπηρετητές για αυτόν τον σκοπό και κύριος στόχος της επίθεσης ήταν οι λογαριασμού με όνομα χρήστη «admin» [29]. Παρομοίως σημειώθηκε το 2013 άλλη μία επίθεση εξαντλητικής αναζήτησης, αυτή τη φορά με στόχο το GitHub. Κύριος στόχος της επίθεσης ήταν λογαριασμοί με αδύναμα συνθηματικά ή συνθηματικά τα οποία ήταν κοινά και με άλλες υπηρεσίες (δηλαδή ο χρήστης είχε χρησιμοποιήσει το ίδιο συνθηματικό και σε άλλη υπηρεσία «επαναχρησιμοποίηση συνθηματικού») [29].

Η δεύτερη κατηγορία, εκείνη της αντίστροφης επίθεσης, περικλείει προφανώς τις εξαντλητικές δοκιμές. Παρόλα αυτά διαφέρει με την προηγούμενη κατηγορία καθώς η βασική ιδέα σε αυτήν την περίπτωση είναι το ταίριασμα των ονομάτων των χρηστών με το συνθηματικό «αφετηρία» της επίθεσης. Αυτό σημαίνει ότι ο επιτιθέμενος ξεκινάει ορίζοντας ένα συνθηματικό και αρχίζει να ψάχνει σε ποια ονόματα χρηστών αντιστοιχεί προκειμένου να αποκτήσει την μη εξουσιοδοτημένη πρόσβαση. Ωστόσο, είναι απαραίτητο να τονιστεί ότι αυτός ο τύπος δεν αποτελεί επίθεση σε συνθηματικό αλλά σε όνομα χρήστη [27].

Αντίθετα η επίθεση λεξικού συμπεριλαμβάνει ως ένα ακόμα είδος επίθεσης εξαντλητικής αναζήτησης όσα ειπώθηκαν παραπάνω αλλά θα μπορούσε να πει κάποιος ότι αποτελεί μια πιο ειδική περίπτωση (χρησιμοποιείται κυρίως σε συνδυασμό με άλλες μεθόδους για μια πιο αποτελεσματική επίθεση). Αυτό συμβαίνει γιατί η πληθώρα των πιθανών συνδυασμών των συνθηματικών πλέον δεν είναι ένα πλήθος χαρακτήρων γενικά (κεφαλαία , πεζά γράμματα, αριθμοί ή ειδικά σύμβολα). Το σύνολο των συνδυασμών σε αυτήν την περίπτωση περιλαμβάνει αυστηρά μια προκαθορισμένη λίστα λέξεων ή ακόμη και φράσεων. Οι λέξεις ή φράσεις που μπορούν να χρησιμοποιηθούν είναι μόνο εκείνες οι οποίες θεωρούνται αποδεκτές από ένα λεξικό.

Με τον όρο λεξικό δεν εννοείται το λεξικό με την κυριολεκτική του έννοια, αλλά σε λίστες οι οποίες περιλαμβάνουν λέξεις-φράσεις που συνδέονται με τον ίδιο τον χρήστη και είναι πιο προσωπικοί (για παράδειγμα το όνομα του, η ημερομηνία

γενεθλίων κλπ.), ή έχουν προκύψει από προηγούμενες επιτυχείς επιθέσεις όπου και καταγράφηκαν τα συνθηματικά. Επομένως γίνεται κατανοητό πως το πλήθος των συνδυασμών είναι αυτομάτως μικρότερο από εκείνο των προηγούμενων 2 υποκατηγοριών. Ένα παράδειγμα τέτοιου συνθηματικού είναι το «password» ή το «1234», τα οποία χρησιμοποιούνται από αρκετούς ακόμα και σήμερα γιατί προσφέρει μια εύκολη λύση που μπορεί κάποιος να τη θυμάται εύκολα.

1.3.4 Καταγραφείς πληκτρολογίου (KeyLoggers)

Οι καταγραφείς πληκτρολογίου είναι άλλος ένας τύπος επίθεσης στα συνθηματικά. Η ιδέα πίσω από την επίθεση αυτή είναι η υποκλοπή του συνθηματικού κατά την πληκτρολόγησή του από τον ίδιο τον χρήστη. Ο επιτιθέμενος λοιπόν θα πρέπει, προκειμένου να αποκτήσει το συνθηματικό, είτε να εγκαταστήσει στο σύστημα-συσκευή του θύματος του ένα λογισμικό καταγραφής (keylogger), είτε ο ίδιος ο χρήστης να εγκαταστήσει εν αγνοία του το συγκεκριμένο αρχείο στη συσκευή του. Με αυτό το λογισμικό παρακολουθεί τις ενέργειες του χρήστη καταγράφοντας κάθε ένα από τα πλήκτρα που πατάει.

Είναι μια μέθοδος που δρα στο «παρασκήνιο» και καταγράφει οποιεσδήποτε κινήσεις του χρήστη σε όλες τις πιθανές υπηρεσίες- ιστοσελίδες. Τα δεδομένα αυτά αποθηκεύονται σε ένα αρχείο και αποστέλλονται στον επιτιθέμενο, ο οποίος πλέον έχει στην κατοχή του διαπιστευτήρια που μπορούν να χρησιμοποιηθούν για μη εξουσιοδοτημένη πρόσβαση σε υπηρεσίες ή ακόμα και σε εμπλοκή σκανδάλων.

Παράδειγμα αποτελεί η επίθεση στην υπηρεσία LastPass [30]. Αρκετά πρόσφατα, το 2023 η υπηρεσία διαχείρισης κωδικών πρόσβασης LastPass δέχθηκε επίθεση παρόλο που είχαν παρθεί αυστηρά μέτρα προφύλαξης σχετικά με το σύστημα. Ο επιτιθέμενος φαίνεται πως κατάφερε να εγκαταστήσει στον οικιακό υπολογιστή ενός από τους 4 μηχανικούς (DevOps engineers), λογισμικό καταγραφής πληκτρολογίου. Οι μηχανικοί ήταν οι μόνοι 4 που είχαν στη διάθεσή τους τα κλειδιά αποκρυπτογράφησης που απαιτούνταν για την πρόσβαση στο σύστημα για τη δημιουργίας των αντιγράφων ασφαλείας της LastPass [31].

1.3.5 Επίθεση με πίνακα «Ουράνιου τόξου» (Rainbow table attack)

Είναι πολύ σημαντικό τα συνθηματικά να μην αποθηκεύονται ως απλά κείμενα (plaintext) όπως έχει ήδη ειπωθεί. Για αυτόν τον λόγο πολλές υπηρεσίες έχουν υιοθετήσει τη μέθοδο κατακερματισμού των κωδικών πρόσβασης με τη βοήθεια κρυπτογραφικών συναρτήσεων. Βέβαια αν η συνάρτηση αυτή δεν χρησιμοποιηθεί κατάλληλα τότε όλα τα όμοια συνθηματικά θα αναπαρίστανται με την ίδια κατακερματισμένη τιμή. Επομένως σε μια δυνητική επίθεση ο χρόνος εύρεσης των κωδικών θα μειωθεί πάρα πολύ, διευκολύνοντας έτσι τους επιτιθέμενους.

Η επίθεση με πίνακα «Ουράνιου τόξου» έχει ως βασικό στόχο να σπάσει τα συνθηματικά των χρηστών χρησιμοποιώντας όμως την κατακερματισμένη τιμή του [32]. Ο επιτιθέμενος παράγει ένα λεξικό με πληθώρα πιθανών συνθηματικών που χρησιμοποιούνται συχνά και για κάθε έναν από αυτούς θα πρέπει να υπολογίσει την τιμή κατακερματισμού (αλυσίδες κατακερματισμού). Το αποτέλεσμα με τα δεδομένα που προκύπτει είναι ένα τεράστιος πίνακας ο οποίος ονομάζεται πίνακας «Ουράνιου τόξου» και περιλαμβάνει τόσο τις κατακερματισμένες τιμές όσο και το συνθηματικό ως απλό κείμενο.

Στη συνέχεια ξεκινάει η σύγκριση αυτών των τιμών (εγγραφές του πίνακα) με εκείνες που έχουν αποθηκευτεί στο σύστημα της υπηρεσίας. Αν κατά τις συγκρίσεις βρεθεί κάποιο ζεύγος που ταιριάζει τότε ο επιτιθέμενος μπορεί γρήγορα να βρει την αντιστοιχία του συνθηματικού σε μορφή απλού κειμένου χωρίς να χρειαστεί να κάνουν κάποια επιπλέον ενέργεια στους αποθηκευμένους κατακερματισμένους κωδικούς της υπηρεσίας.

Ένα παράδειγμα τέτοιας μορφής επίθεσης σημειώθηκε το 2012. Το Linkedin είναι μία από τις μεγαλύτερες ιστοσελίδες τόσο επαγγελματικής δικτύωσης όσο και κοινωνικής δικτύωσης στον κόσμο. Τα μέλη που είναι εγγεγραμμένα ξεπερνάνε κατά πολύ τα 200 εκατομμύρια και έχουν καταχωρήσει στη σελίδα δεδομένα που αφορούν την επαγγελματική κατάρτισή τους αλλά και πληροφορίες προσωπικές. Έχοντας τόσες πολλές πληροφορίες για μια τόσο μεγάλη πληθώρα ανθρώπων είναι απόλυτα φυσικό η σελίδα να έχει θέσει ως πρωταρχικό της μέλημα την προφύλαξη των προσωπικών και επαγγελματικών δεδομένων των ατόμων που τη χρησιμοποιούν.

Ωστόσο, το 2012 οι διακομιστές του Linkedin παραβιάστηκαν αποκαλύπτοντας τους κατακερματισμένους κωδικούς πρόσβασης μιας τάξης των 117 εκατομμυρίων [4]. Οι κωδικοί αυτοί αποκωδικοποιήθηκαν γρήγορα και στη συνέχεια πωλήθηκαν. Υπεύθυνος για αυτό το γεγονός θεωρείται ο Ρώσσος Yevgeniy Alexandrovich Nikulin, ο οποίος καταδικάστηκε χρόνια αργότερα (Οκτώβριος του 2016) [22]. Η πώληση αυτών των στοιχείων ωστόσο δε συνδέεται εντελώς με τον Nikulin αλλά με ένα άλλο άτομο που χρησιμοποιούσε το ψευδώνυμο «Peace».

Τρωτό σημείο της υπηρεσίας ήταν η ακατάλληλη χρήση της κρυπτογραφικής συνάρτησης. Το Linkedin χρησιμοποίησε με τη σειρά του ως συνάρτηση κατακερματισμού το SHA-1. Πιο συγκεκριμένα η μέθοδος του κατακερματισμού των συνθηματικών πρόσβασης αφού λάβει τον κωδικό x μήκους εξάγει έξοδο η οποία έχει μεν σταθερό μήκος αλλά μοιάζει σαν τυχαία συμβολοσειρά. Το SHA-1 που δημοσιεύτηκε το 1990 έχει τη δυνατότητα να κωδικοποιήσει μια είσοδο έως και 264 bit σε μια έξοδο 160 bit. Το SHA-1 όμως αποθήκευε απλά τους κατακερματισμένους κωδικούς πρόσβασης απευθείας στον διακομιστή αντί να συμπεριλάβει salts δηλαδή τυχαίους αριθμούς μοναδικούς για κάθε έναν χρήστη.

Όπως έγινε γνωστό από την εταιρεία ασφαλείας KoreLogic πάνω από ένα εκατομμύριο χρήστες χρησιμοποίησαν τη φράση «123456» ως κωδικό πρόσβασης. Αυτό το γεγονός μειώνει δραστικότατα το υπολογιστικό κόστος αφού ο συγκεκριμένος κωδικός κατακερματίζεται με την ίδια τιμή πάντα. Εξετάζοντας την παραπάνω περίπτωση προκύπτουν τα ακόλουθα. Έστω λοιπόν ο κωδικός «123456». Με τον αλγόριθμο SHA-1 προκύπτει μια έξοδος της μορφής 7c4a8d09ca3762af61e59520943dc26494f8941b. Επομένως κάθε φορά που ένας χρήστης καταχωρεί αυτόν τον κωδικό στη σελίδα ο αλγόριθμος θα παράγει κάθε φορά ξανά και ξανά την ίδια έξοδο δηλαδή το 7c4a8d09ca3762af61e59520943dc26494f8941b. Η έξοδος αυτή προκύπτει αν μετατραπεί κάθε ένας χαρακτήρας σε μορφή ASCII.

ΧΑΡΑΚΤΗΡΑΣ	ΧΑΡΑΚΤΗΡΑΣ ΣΕ ASCII
1	49
2	50
3	51
4	52
5	53
6	54

Πίνακας 2. Αντιστοίχιση αριθμών με χαρακτήρες ASCII

Από αυτή την αντιστοίχιση προκύπτει μια ακολουθία της μορφής: 495051525354. Αυτή την ακολουθία στη συνέχεια θα την μετατρέψει ο αλγόριθμος μέσω διάφορων λειτουργιών στην τελική μορφή που αναφέρθηκε παραπάνω.

Η αλυσίδα κατακερματισμού που προκύπτει είναι της μορφής:

```
( ‘7c4a8d09ca3762af61e59520943dc26494f8941b’ , ‘123456’)
```

Εάν τώρα προστεθούν και τα salts, για παράδειγμα 3 bits, αυτομάτως οι συνδυασμοί που παράγονται από $2^0 = 1$ γίνονται $2^3 = 9$. Το συγκεκριμένο σενάριο φυσικά αυξάνει το κόστος ενός «rainbow table» αφού για κάθε έναν κωδικό θα πρέπει να υπολογίζει και να αποθηκεύει 9 διαφορετικούς συνδυασμούς. Οι συνδυασμοί που προκύπτουν είναι οι εξής:

1. (‘7c4a8d09ca3762af61e59520943dc26494f8941b’ , ‘123456’) ο κωδικός χωρίς salt
2. (‘b2f8c9d466d6f7fe8a4c0c1b62adb9c90283dc3b’ , ‘000123456’) με salt 000
3. (‘fc4d728d66e38263617d2c29506a6c29b44c9cc4’ , ‘001123456’) με salt 001
4. (‘a28abc8ecc77d17cb8476c78ded184ce6415aa2a’ , ‘010123456’) με salt 010
5. (‘d37f933a71f037446aaf327d1c07e94c56835243’ , ‘011123456’) με salt 011
6. (‘0971afa21ac7d91e9840b16138ff8fb015210071’ , ‘100123456’) με salt 100
7. (‘3e567892ddeda72ec5d94128bd5b257889792790’ , ‘101123456’) με salt 101
8. (‘febeca0dc3ed1b49521bdd723fb79dff5a721df2’ , ‘110123456’) με salt 110
9. (‘be0d5470ef1c9413fb436711508094b306d35ca0’ , ‘111123456’) με salt 111

Φυσικά όσο αυξάνονται τα bits των salts τόσο περισσότεροι συνδυασμοί παράγονται και άρα δυσκολεύουν περισσότερο τους επιτιθέμενους.

Οι έξοδοι αυτοί μπορούν να υπολογιστούν από κάποιο διαδικτυακό γεννήτορα για τον αλγόριθμο SHA-1 ή χρησιμοποιώντας τη βιβλιοθήκη hashlib της python. Παρακάτω ακολουθεί κώδικας σε python που εξάγει το ίδιο αποτέλεσμα (δηλαδή input + salt)

```
import hashlib
```

```
password = "000123456"
hashed_password = hashlib.sha1(password.encode()).hexdigest()

print(hashed_password)
```

Η βιβλιοθήκη hashlib περιλαμβάνει τη συνάρτηση ***sha1()*** η οποία δημιουργεί αντικείμενα κατακερματισμού τύπου sha-1. Η συνάρτηση αυτή δέχεται ως είσοδο μόνο bytes. Άρα με τη μέθοδο ***encode()*** επιτυγχάνεται η μετατροπή της εκάστοτε συμβολοσειράς εισόδου σε bytes. Στη συνέχεια με τη μέθοδο ***hexdigest()*** το αποτέλεσμα του κατακερματισμού δίνεται σε δεκαεξαδική μορφή. Τέλος όπως φαίνεται και στον κώδικα τυπώνεται το αποτέλεσμα. Περισσότερες πληροφορίες μπορεί κάποιος να βρει στην επίσημη σελίδα της python [2].

Άρα για να αποφευχθεί αυτός ο κίνδυνος χρησιμοποιούνται οι τιμές των αλάτων (salts) στις συναρτήσεις κατακερματισμού. Έτσι οι πίνακες «Ουράνιου τόξου» δεν είναι πλέον αποτελεσματικοί αφού έχει αυξηθεί σημαντικά η πολυπλοκότητα των υπολογισμών. Ουσιαστικά με την προσθήκη αλάτων ο επιτιθέμενος θα πρέπει να δημιουργήσει όσους πίνακες χρειάζεται για να καλυφθούν όλοι οι πιθανοί κατακερματισμοί του ίδιου συνθηματικού.

Και κάπου εδώ κρίνεται σκόπιμο να αναφερθεί άλλη μία περίπτωση επίθεσης κατά την οποία ναι μεν τα αδύναμα συνθηματικά δεν ήταν ρίζα του προβλήματος αλλά είχαν ως αποτέλεσμα την διόγκωση του προβλήματος. Σημαντική περίπτωση παραβίασης δεδομένων είναι εκείνη της Equifax. Το 2017 μία από τις μεγαλύτερες εταιρείες παροχής πιστωτικών αναφορών , η Equifax , υπέστη μαζική παραβίαση ασφαλείας, η οποία έθεσε σε κίνδυνο ευαίσθητες πληροφορίες στις οποίες συμπεριλαμβάνονταν και οι κωδικοί πρόσβασης.

Τον Ιούλιο του 2017 η εταιρεία διαπίστωσε μια κυβερνοεπίθεση στο αυτόματο σύστημα συνεντεύξεων των καταναλωτών της (automated consumer interview system - ACIS). Η επίθεση αυτή εντοπίστηκε κατά την αναβάθμιση μιας πιστοποίησης του SSL (secure socket layer) η λειτουργία της οποίας συνδεόταν με την παρακολούθηση της εξερχόμενης αλλά και της εισερχόμενης διαδικτυακής κίνησης μεταξύ των συστημάτων της Equifax.

Μείζον ρόλο στην καθυστέρηση εντοπισμού της επίθεσης έπαιξε η λήξη των πιστοποιητικών SSL. Ουσιαστικά από τον Μάϊο όπου και θεωρείται πως ξεκίνησε η επίθεση [7] και για 2 μήνες κανείς δεν είχε υποπτευθεί κάτι. Μόλις λοιπόν ολοκληρώθηκε η αναβάθμιση τον Ιούλιο, παρατηρήθηκε περίεργη κίνηση σε σύστημα της εταιρείας και όπως λένε οι πληροφορίες περιλάμβανε αρχεία με φωτογραφίες που σχετίζονταν με έρευνες πιστωτικής φύσεως για τους εκάστοτε καταναλωτές. Οι επιτιθέμενοι κατόπιν έρευνας φαίνεται να ήταν μια ομάδα από την Κίνα (βάσει των IPs).

Η επίθεση αυτή εν γένει θα μπορούσε να έχει αντίκτυπο σε πάνω από 143 δισεκατομμύρια καταναλωτές στις Ηνωμένες Πολιτείες και αυτό λόγο τρομερής αμέλειας. Η ομάδα έκτακτης ανάγκης των Ηνωμένων Πολιτειών για τους υπολογιστές (United States Computer Emergency Readiness Team, US-CERT) όπως έχει γίνει γνωστό είχε ενημερώσει από τις 8 Μαρτίου, προτού λάβει χώρα η

συγκεκριμένη επίθεση, για την ευπάθεια που εμφάνισε το ανοιχτού κώδικα πλαίσιο εφαρμογών ιστού Apache Struts 2 (Open source Apache Struts 2 web application framework), ένα framework που χρησιμοποιούσε και το σύστημα της εταιρείας με αποτέλεσμα την μη εξουσιοδοτημένη πρόσβαση των επιτιθέμενων στο δίκτυο της Equifax. Είναι επίσης σημαντικό να τονιστεί πως στο GitHub στις 11 Μαρτίου τέθηκαν στη διάθεση του κοινού τρόποι με τους οποίους θα μπορούσαν να εκμεταλλευτούν αυτή την ευπάθεια.

Πως σχετίζεται αυτό όμως με τα αδύναμα συνθηματικά; Σε αρκετά συστήματα της Equifax παρατηρήθηκε πως οι κωδικοί πρόσβασης σε προνομιακούς λογαριασμούς ήταν αδύναμοι. Πιο συγκεκριμένα μία από τις βάσεις δεδομένων, στις οποίες είχαν πλέον πρόσβαση οι επιτιθέμενοι, ο κωδικός πρόσβασης αποτελούνταν από 4 χαρακτήρες μόνο , πεζούς οι οποίοι μάλιστα ταίριαζαν και με το όνομα της βάσης. Επομένως μη έχοντας αποδώσει τη δέουσα προσοχή στην επίλυση της ευπάθειας επέτρεψαν την μη εξουσιοδοτημένη πρόσβαση στα συστήματά τους και λόγω του ακατάλληλου ελέγχου ταυτότητάς τους διευκόλυναν ώστε να λάβουν στην κατοχή τους ακόμη περισσότερα στοιχεία.

Είναι σημαντικό όλες οι υπηρεσίες και ειδικά εκείνες που διαχειρίζονται τόσο κρίσιμα δεδομένα για τους καταναλωτές όπως οι τραπεζικοί λογαριασμοί τους και άλλα προσωπικά δεδομένα να φροντίζουν η δομή των συστημάτων τους να προσφέρει την απαιτούμενη ασφάλεια από όλες τις πλευρές και να μην επιτρέπει σε δευτερεύοντα πρόσωπα να προσφύγουν σε διάφορες δόλιες δραστηριότητες χρησιμοποιώντας τα στοιχεία που έχουν βρει σε βάρος των κατόχων τους.

ΚΕΦΑΛΑΙΟ 2: ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΧΩΡΙΣ ΣΥΝΘΗΜΑΤΙΚΑ

2.1 Εισαγωγή

Στην προηγούμενη ενότητα έγινε μια αναφορά στις ευπάθειες και τα τρωτά σημεία των συνθηματικών πρόσβασης. Πραγματοποιήθηκε επίσης μια συνοπτική παρουσίαση κάποιων γνωστών περιστατικών στα οποία η χρήση ανίσχυρων κωδικών πρόσβασης ήταν είτε η αιτία είτε αποτέλεσαν καθοριστικό ρόλο στην μετέπειτα διόγκωση του προβλήματος. Παρουσιάστηκαν και περιστατικά τα οποία είχαν ως αποτέλεσμα τη διαρροή συνθηματικών και μάλιστα σε απλό κείμενο και επομένως κάποιος θα μπορούσε να τα χρησιμοποιήσει απευθείας για να συνδεθεί σε λογαριασμούς. Τα περιστατικά αυτά συντάραξαν τη διεθνή κοινότητα και θεωρούνται ένα καλό μάθημα ως προς την ασφάλεια τόσο από την πλευρά των υπηρεσιών όσο και από την πλευρά των χρηστών.

Το βασικότερο και πιο κοινό πρόβλημα είναι εκείνο του αδύναμου συνθηματικού (μικρού μήκους, συνηθισμένα χρησιμοποιούμενη λέξη ή συμβολοσειρά, εύκολο να το υποθέσει κάποιος κλπ), στη συνεχή επαναχρησιμοποίηση του ίδιου συνθηματικού σε διαφορετικούς λογαριασμούς ή ακόμη και η ανακύκλωση κωδικών (η χρήση συγκεκριμένων κωδικών από μια λίστα) στον ίδιο λογαριασμό, η μη ασφαλής αποθήκευση των κωδικών αυτών ή η αποκάλυψη αυτών από τον ίδιο τον χρήστη από λάθος. Οι επιθέσεις στα συνθηματικά μπορούν να θέσουν σε κίνδυνο τους κωδικούς πρόσβασης και κατ' επέκταση τους λογαριασμούς των χρηστών.

Προκειμένου οι χρήστες να απαλλαχθούν από τα συνθηματικά και τα όσο προβλήματα φέρουν είναι σημαντικό να χρησιμοποιηθούν άλλοι πιο ισχυροί τρόποι ταυτοποίησης πέρα από ισχυρότερους κωδικούς. Με το σκεπτικό ότι κάποιος θα είναι ιδιαίτερα προσεκτικός δε λύνεται το πρόβλημα αφού οι επιθέσεις και οι απόπειρες είναι πάρα πολλές. Επομένως θα ήταν πολύ χρήσιμη τακτική η ταυτοποίηση χωρίς να απαιτείται η εισαγωγή κάποιου κωδικού (passwordless authentication).

Κρίνεται απαραίτητο να σημειωθεί πως μια τέτοιου τύπου ταυτοποίηση εξαρτάται από πολλούς διαφορετικούς παράγοντες , οι οποίοι συμπεριλαμβάνουν τις ειδικές απαιτήσεις του εκάστοτε συστήματος , της βάσης που θα περιλαμβάνει τα στοιχεία των χρηστών και φυσικά το επιθυμητό επίπεδο ασφαλείας. Η προσεκτική εξέταση των παραγόντων αυτών θα αποφέρει μια πιο αποτελεσματική λύση όπως επίσης και η προσαρμογή της εκάστοτε μεθόδου σε αυτούς τους παράγοντες θα διασφαλίσει μια απρόσκοπτη αλλά και ασφαλή εμπειρία στον ίδιο τον χρήστη. Στο κεφάλαιο αυτό θα αναλυθεί λεπτομερώς ο τρόπος της βιομετρικής ταυτοποίησης (βάσει φυσικών χαρακτηριστικών) ως μέσο αυθεντικοποίησης και ποια είναι η διαδικασία που ακολουθείται προκειμένου να επιτευχθεί αυτό.

2.2 Βιομετρικοί έλεγχοι ταυτοποίησης φυσικών χαρακτηριστικών

Ο βιομετρικός έλεγχος αυθεντικοποίησης (Biometric authentication) αξιοποιεί τα βιομετρικά χαρακτηριστικά των ανθρώπων προκειμένου να γίνει η ταυτοποίηση του χρήστη. Τα βιομετρικά χαρακτηριστικά διακρίνονται σε 2 κατηγορίες. Η πρώτη κατηγορία αφορά τα φυσικά (physiological) χαρακτηριστικά όπως για παράδειγμα τα δάκτυλα, η ίριδα, το πρόσωπο κλπ. Η δεύτερη κατηγορία περιλαμβάνει χαρακτηριστικά που σχετίζονται με τη συμπεριφορά (behavioral) και τον τρόπο που ένας χρήστης κάνει κάτι όπως η βάδιση , κάποιες κινήσεις του κεφαλιού, η πληκτρολόγηση κλπ.

Στην προσπάθεια για ταυτοποίηση μπορεί να συνδυαστούν και οι δύο κατηγορίες των βιομετρικών χαρακτηριστικών ή η μέθοδος να επικεντρωθεί μόνο σε μια ολόκληρη κατηγορία είτε απλώς και μόνο σε ένα χαρακτηριστικό. Από όλα αυτά τα βιομετρικά, τα πιο διαδεδομένα και εκείνα που έχουν κυριαρχήσει στις εφαρμογές, που δε χρησιμοποιούν συνθηματικά για την ταυτοποίηση του χρήστη, είναι κατά κύριο λόγο τα φυσικά , δηλαδή τα δακτυλικά αποτυπώματα , το πρόσωπο και η ίριδα.

Αυτό συμβαίνει γιατί τόσο τα δακτυλικά αποτυπώματα , η σάρωση της ίριδας όσο και η αναγνώριση προσώπου είναι βολικά ως προς τη χρήση ενώ παράλληλα παρέχουν αξιοπιστία αλλά και ακρίβεια στις επιδόσεις τους. Τα χαρακτηριστικά αυτά είναι μοναδικά για κάθε ένα άτομο (για παράδειγμα ακόμη και τα μονοζυγωτικά δίδυμα έχουν διαφορετικά δακτυλικά αποτυπώματα) [6] και είναι αρκετά δύσκολο να τα παραποιήσει κάποιος. Ωστόσο, υπάρχουν και άλλα φυσικά χαρακτηριστικά που προσελκύουν το ενδιαφέρον.

2.2.1 Βήματα διαδικασίας βιομετρικού ελέγχου ταυτοποίησης με φυσικά χαρακτηριστικά

Σε αυτή την υπό ενότητα θα παρουσιαστούν τα βασικά βήματα που ακολουθεί μια μέθοδος βιομετρικού ελέγχου ταυτοποίησης (οποιοδήποτε βιομετρικό χαρακτηριστικό και να χρησιμοποιεί).

Κατά τη διαδικασία της εγγραφής, το εκάστοτε βιομετρικό δείγμα ενός ατόμου είτε αυτό είναι δακτυλικό αποτύπωμα, είτε το πρόσωπο, είτε η ίριδα κατόπιν σάρωσης, λαμβάνεται χρησιμοποιώντας εξειδικευμένους βιομετρικούς αισθητήρες. Αφού έχει ληφθεί αυτό το δείγμα υποβάλλεται εν συνεχεία σε επεξεργασία προκειμένου να εξαχθεί η μοναδική πληροφορία που σχετίζεται φυσικά με την επιλεγμένη βιομετρική μέθοδο. Η εξαγόμενη πληροφορία μετατρέπεται αργότερα σε ένα τυποποιημένο πρότυπο , το οποίο με τη σειρά του αντιπροσωπεύει τα βιομετρικά δεδομένα του ατόμου, σε μαθηματική μορφή. Για παράδειγμα, κατά τη σάρωση της ίριδας το σύστημα δε θα χρησιμοποιήσει ως πρότυπο τη φωτογραφία του χαρακτηριστικού αυτού όπως θα ήταν αναμενόμενο.

Τα πρότυπα βεβαίως φυλάσσονται σε ένα καλά προστατευμένο περιβάλλον - βάση δεδομένων. Κατά τη διαδικασία της αναγνώρισης και προκειμένου να

επιτευχθεί αποτελεσματικότερη ανάκτηση αλλά και γρηγορότερη αναζήτηση γίνεται ευρετηρίαση και έχει προηγηθεί προηγουμένως σωστή οργάνωση αυτών. Η βάση δεδομένων είναι σε θέση να διαχειρίζεται έναν τεράστιο όγκο προτύπων, τα οποία μπορεί να κυμαίνονται από χιλιάδες έως και πιθανότατα δισεκατομμύρια.

Κάθε φορά που ένα άτομο παρουσιάζει το βιομετρικό του δείγμα για να πραγματοποιηθεί η αναγνώριση , το σύστημα συλλαμβάνει αυτό το νέο δείγμα και ακολουθεί την ίδια διαδικασία εξαγωγής της πληροφορίας όπως συνέβη και κατά τη διαδικασία της εγγραφής. Αφού ολοκληρωθεί αυτό το στάδιο, τα δεδομένα που έχουν εξαχθεί συγκρίνονται με το πρότυπα που είναι ήδη αποθηκευμένο στη βάση δεδομένων.

Το σύστημα χρησιμοποιώντας τον αλγόριθμο αντιστοίχισης που του έχουν καθορίσει οι προγραμματιστές υπολογίζει την ομοιότητα των δύο αυτών δειγμάτων. Ο αλγόριθμος του λογισμικού για τις αντιστοιχίσεις υπολογίζει ουσιαστικά το πόσο όμοια ή κατά πόσο διαφέρουν τα εγγεγραμμένα πρότυπα από εκείνα του νέου δείγματος παράγοντας μια βαθμολογία αντιστοίχισης ή αλλιώς βαθμολογία ομοιότητας.

Έχει τεθεί εξ’ αρχής ένα όριο το οποίο ονομάζεται κατώφλι (threshold) και βάσει αυτού κρίνεται εάν το ποσοστό ομοιότητάς τους είναι αποδεκτό ή όχι. Η ρύθμιση κατωφλίου χρησιμοποιείται με σκοπό να καθορίσει εάν η βαθμολογία ομοιότητας ξεπερνά το εκάστοτε προκαθορισμένο όριο αποδοχής ή το όριο απόρριψης. Η τιμή του κατωφλίου μπορεί να προσαρμοστεί ανάλογα με το επιθυμητό αποτέλεσμα ως προς το επίπεδο ασφαλείας και τις απαιτήσεις της ίδιας της εφαρμογής.

Προφανώς όσο πιο υψηλή είναι η τιμή που έχει αποδοθεί στο όριο αυτό τόσο πιο αυστηρό είναι το κριτήριο της αντιστοίχισης, αφού με αυτόν τον τρόπο μειώνονται σημαντικά οι πιθανότητες να γίνει αποδεκτό ένα ψευδές δεδομένο αλλά ταυτόχρονα υπάρχει και η περίπτωση δυνητικά να αυξηθούν οι πιθανότητες των ψευδών απορρίψεων (δηλαδή ενώ το δείγμα είναι όντως του χρήστη που θέλει να ταυτοποιηθεί, για λόγους θορύβου ή κάποιας αδυναμίας στους αισθητήρες , να θεωρηθεί ψευδές και άρα να μην μπορέσει ο χρήστης να αποκτήσει πρόσβαση στους λογαριασμούς του).

Είναι σημαντικό να αναγνωρίσει κάποιος ότι η επίτευξη μιας τέλειας ισορροπίας μεταξύ των ποσοστών ψευδούς αποδοχής και απόρριψης μπορεί να αποτελέσει μια πρόκληση. Τα αυστηρότερα όρια, όπως έχει αναφερθεί, που ορίζονται για τη μείωση των ποσοστών της ψευδούς αποδοχής ενδέχεται να αυξήσουν ακούσια τα ποσοστά ψευδούς απόρριψης, οδηγώντας σε μια κατάσταση απογοήτευσης τους χρήστες αφού θα απαγορεύεται συνεχώς η πρόσβασή τους. Αντίθετα αν τα όρια που τίθενται είναι πιο χαλαρά , τότε εν μέρει ελαχιστοποιούνται τα ποσοστά ψευδούς απόρριψης αλλά παράλληλα ενδέχεται αυτή η προσέγγιση να θέσει σε κίνδυνο την ασφάλεια του συστήματος. Η προσέγγιση αυτή θα επιτρέψει σε μη εξουσιοδοτημένα άτομα να αποκτήσουν πρόσβαση στις εκάστοτε υπηρεσίες. Αυτός είναι και ο βασικός λόγος για την εύρεση της βέλτιστης ισορροπίας.

Η εύρεση της βέλτιστης ισορροπίας απαιτεί προσεκτική ρύθμιση και συνεχή αξιολόγηση της απόδοσης του συστήματος. Για τον μετριασμό των ποσοστών των

ψευδών αποτελεσμάτων μπορούν να χρησιμοποιηθούν διάφορες τεχνικές. Φυσικά ο ορισμός σωστής τιμής κατωφλίου για την εξισορρόπηση της αντιστάθμισης μεταξύ ευκολίας και ασφάλειας είναι επίσης μια λύση στο πρόβλημα.

Καθώς οι συγκρίσεις που λαμβάνουν χώρα είναι το σημαντικότερο κομμάτι της όλης διαδικασίας είναι συνετό να χρησιμοποιηθούν εξελιγμένοι αλγόριθμοι αντιστοίχισης για τις συγκρίσεις μεταξύ των αποθηκευμένων προτύπων και των εξαγόμενων πληροφοριών του νέου δείγματος. Οι προηγμένοι αλγόριθμοι και τα μοντέλα μηχανικής μάθησης χρησιμοποιούνται για τη βελτίωση της ακρίβειας στην αντιστοίχιση των βιομετρικών δεδομένων με τα εγγεγραμμένα πρότυπα (templates). Αυτοί οι αλγόριθμοι χρησιμοποιούν μαθηματικές τεχνικές για τη μέτρηση της ομοιότητας μεταξύ των συνόλων των χαρακτηριστικών και τη δημιουργία των σχετικών βαθμολογιών. Το κριτήριο επιλογής ενός αλγορίθμου αντιστοίχισης ποικίλλει αναλόγως της βιομετρικής μεθόδου και πάντα τις απαιτήσεις της εφαρμογής.

Με βάση αυτή τη βαθμολογία ομοιότητας που προκύπτει καθώς επίσης και το σύνολο των ορίων που έχουν τεθεί, το σύστημα είναι πλέον σε θέση να αποφασίσει εάν θα αποδεχτεί ή τελικώς αν θα πρέπει να απορρίψει το εκάστοτε δείγμα που έχει παρουσιαστεί. Στην περίπτωση που το αποτέλεσμα έχει κατορθώσει να υπερβεί το όριο που έχει τεθεί , τότε η αντιστοίχιση είναι επιτυχής και πλέον η ταυτότητα του χρήστη μπορεί να επιβεβαιωθεί. Από την άλλη πλευρά στην περίπτωση που η βαθμολογία είναι μικρότερη από εκείνη του ορίου, το σύστημα απορρίπτει άμεσα το δείγμα υποδεικνύοντας ότι υπήρξε αδυναμία αντιστοίχισης.

Όλες αυτές οι λειτουργίες που παρέχονται από το εκάστοτε λογισμικό ως διαδικασίες αναγνώρισης και επαλήθευσης, έχουν σχεδιαστεί ώστε να λειτουργούν σε πραγματικό χρόνο και να παράγουν αποτελέσματα πάρα πολύ γρήγορα. Τόσο η ταχύτητα όσο και η ακρίβεια του συστήματος αυτού εξαρτώνται από πολλούς διαφορετικούς παράγοντες. Η ποιότητα του ίδιου του βιομετρικού δείγματος που θα ληφθεί ως είσοδος για το σύστημα , το μέγεθος της βάσης δεδομένων , η υπολογιστική ισχύς της υποδομής αλλά και η αποτελεσματικότητα των αλγορίθμων αντιστοίχισης είναι μερικοί από τους παράγοντες που επηρεάζουν άμεσα το σύστημα [9].

Γενικά τα βασικά βήματα που θα πρέπει να ακολουθούνται προκειμένου να γίνει μια βιομετρική αυθεντικοποίηση παρουσιάζονται στο παρακάτω γράφημα [Διάγραμμα 1]:

Διάγραμμα 1. Βασικά βήματα βιομετρικής αυθεντικοποίησης

Άρα προκύπτουν τα εξής:

- Το πρώτο στάδιο της διαδικασίας ταυτοποίησης αφορά τη συλλογή του βιομετρικού δείγματος **(Sample Capturing – Συλλογή Δεδομένων)**. Ουσιαστικά ένας χρήστης την πρώτη φορά που θα δημιουργήσει έναν λογαριασμό θα χρειαστεί να καταχωρήσει το εκάστοτε βιομετρικό του χαρακτηριστικό. Το χαρακτηριστικό αυτό θα χρησιμοποιηθεί ως μέσω σύγκρισης για την αξιολόγηση του δείγματος που θα εισάγει ο χρήστης τις επόμενες φορές που θα προσπαθήσει να συνδεθεί στο λογαριασμό του. Τα βιομετρικά χαρακτηριστικά που δέχεται ένα σύστημα μπορεί να ποικίλουν αλλά η ανίχνευσή τους μπορεί να πραγματοποιηθεί με τη χρήση αισθητήρων ή κάποιου άλλου εργαλείου όπως για παράδειγμα μικρόφωνο. Φυσικά για αυτού του είδους τις περιπτώσεις θα πρέπει να επιλέγονται άριστης ποιότητας εργαλεία προκειμένου το αποτέλεσμα να είναι όσο το δυνατόν πιο αξιόπιστο.
- Στη συνέχεια ακολουθεί η προεπεξεργασία του δείγματος **(Data Preprocessing - Προεπεξεργασία δεδομένων)**. Το δείγμα που θα συλλεχθεί θα πρέπει να απαλλαχτεί όσον το δυνατόν περισσότερο από κάθε είδους θόρυβο ή άλλους παράγοντες που το αλλοιώνουν. Το δείγμα αυτό θα χρησιμοποιηθεί ως είσοδος στο σύστημα επομένως αυτή η προεργασία θα πρέπει να προηγηθεί προτού ξεκινήσει η μετατροπή του.
- Κατά το τρίτο στάδιο της διαδικασίας έχουν αντληθεί κάποιες πληροφορίες σε ένα απειροελάχιστο χρονικό διάστημα και έχει προκύψει μια έξοδος βασιζόμενη σε κάποια κρίσιμα **(Minutiate Extraction)** χαρακτηριστικά. Τα στοιχεία που έχουν εξαχθεί από το προηγούμενο στάδιο μετατρέπονται σε μια μαθηματική σχέση και δημιουργείται ένα πρότυπο **(Mathematical Depiction - Μαθηματική Αποτύπωση)**. Το πρότυπο αυτό αποτελεί πλέον το αποτυπωμένο βιομετρικό δείγμα που λήφθηκε αρχικά ως είσοδος στο σύστημα και είναι σε θέση να αποθηκευτεί και στη συνέχεια να γίνουν συγκρίσεις με αυτό όπως αναφέρθηκε και στο πρώτο στάδιο.
- Το τέταρτο στάδιο είναι αυτό της σύγκρισης **(Comparison - Σύγκριση)**. Αφού λοιπόν έχει προηγηθεί όλη η προηγούμενη επεξεργασία το πρότυπο αυτό που έχει αποθηκευτεί σε μια βάση δεδομένων μπορεί πλέον να χρησιμοποιηθεί για την ταυτοποίηση του χρήστη κατά τις επόμενες συνδέσεις του στον εκάστοτε λογαριασμό. Την επόμενη φορά δηλαδή που θα επιχειρήσει ο χρήστης να ταυτοποιηθεί θα εισάγει εκ νέου ένα νέο δείγμα (σύμφωνα με τα βήματα που αναφέρθηκαν στα πρώτα 3 στάδια) το οποίο θα συγκριθεί με το πρότυπο προκειμένου να δοθεί ένα τελικό αποτέλεσμα.
- Τέλος το πέμπτο στάδιο το οποίο αποτελεί και τελικό στάδιο της διαδικασίας είναι η ταυτοποίηση **(Identification - Ταυτοποίηση)**. Ουσιαστικά βάσει του αποτελέσματος που προκύπτει από τη σύγκριση του προηγούμενο σταδίου ο χρήστης ταυτοποιείται ή η προσπάθεια απορρίπτεται.

2.2.2 Δακτυλικά Αποτυπώματα

Ένας παραδοσιακός τρόπος ταυτοποίησης χωρίς συνθηματικά είναι τα δακτυλικά αποτυπώματα. Πολλές εφαρμογές έχουν υιοθετήσει αυτή τη μορφή αυθεντικοποίησης λαμβάνοντας υπόψη την αξιοπιστία στα δακτυλικά μοτίβα με το πέρας του χρόνου. Τα δακτυλικά αποτυπώματα είναι μοναδικά για κάθε άνθρωπο. Οι πολύπλοκες και μοναδικές περιελίξεις του δέρματος στην άκρη των δαχτύλων δημιουργούν μια τραχιά επιφάνεια η οποία συμβάλλει στην αίσθηση της αφής. Πέρα από τα γονίδια το τελικό ανάγλυφο των δακτυλικών αποτυπωμάτων εξαρτάται σε μεγάλο βαθμό και από την ανατομία του άκρου.

Ένα κλασσικό παράδειγμα ελέγχου ταυτότητας με τη χρήση δακτυλικών αποτυπωμάτων είναι στην πλατφόρμα Android. Αποτελεί μια βιομετρική δυνατότητα που επιτρέπει στους χρήστες να ξεκλειδώσουν με ασφάλεια τις κινητές συσκευές τους ή να ελέγχουν κάποιες ενέργειες. Ο έλεγχος αυτός απαιτεί φυσικά συσκευές οι οποίες έχουν ενσωματωμένο αισθητήρα δακτυλικών αποτυπωμάτων. Τα περισσότερα σύγχρονα smartphones και tablets είναι εξοπλισμένα με τέτοιους αισθητήρες στο κουμπί της αρχικής οθόνης ή στο κουμπί λειτουργίας. Οι αισθητήρες καταγράφουν τα μοναδικά μοτίβα και τις ραβδώσεις του δαχτύλου του χρήστη. Η διαδικασία αυτή είναι άμεση.

Η πλατφόρμα Android παρέχει ένα fingerprint API (Application programming interface) το οποίο δίνει τη δυνατότητα στους προγραμματιστές να ενσωματώσουν τον έλεγχο αυτό στις εφαρμογές τους. Το API διαχειρίζεται την επικοινωνία μεταξύ εφαρμογής και αισθητήρα διασφαλίζοντας έτσι την ασφαλή και τυποποιημένη πρόσβαση στα δεδομένα των δακτυλικών αποτυπωμάτων. Φυσικά για να επιτευχθεί η ταυτοποίηση ο χρήστης θα πρέπει σε προγενέστερο διάστημα να έχει εγγράψει τα δακτυλικά του αποτυπώματά στη συσκευή του.

Η εγγραφή των δακτυλικών αποτυπωμάτων μπορεί να περιλαμβάνει την καταχώρηση είτε ενός δακτυλικού αποτυπώματος είτε περισσότερων. Συνήθως αυτή η δυνατότητα συμπεριλαμβάνεται στο σύστημα. Παρέχονται οδηγίες που ενημερώνουν το χρήστη για τα βήματα που θα πρέπει να ακολουθήσουν αφού τοποθετήσουν το δάκτυλό τους πάνω στον αισθητήρα. Η καταχώρηση πολλών αποτυπωμάτων διευκολύνει τον έλεγχο της ταυτότητας καθώς επιτρέπει την ταυτοποίηση μέσω των διαφορετικών δακτύλων. Αφού ολοκληρωθεί αυτή η διαδικασία οι χρήστες μπορούν να ρυθμίσουν τη συσκευή τους ώστε να χρησιμοποιεί ως μέθοδο ξεκλειδώματος της οθόνης (οθόνη κλειδώματος) την αναγνώριση των αποτυπωμάτων τους. Με αυτόν τον τρόπο αντικαθίσταται ο παραδοσιακός τρόπος με τη χρήση συνθηματικών ή μοτίβων.

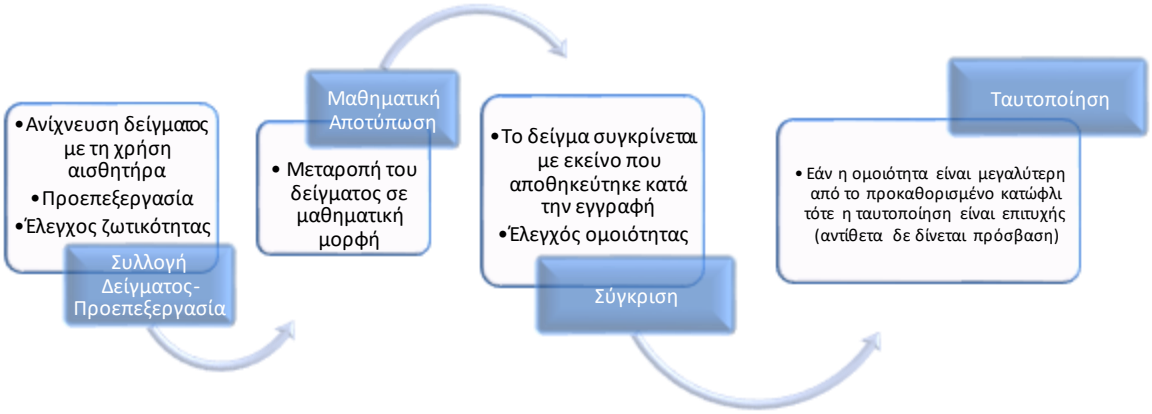
Δίνοντας προτεραιότητα στην ασφάλεια, τα δεδομένα αυτά αποθηκεύονται μέσα στη συσκευή και είναι απρόσιτα σε άλλες εφαρμογές ή διαδικασίες. Θα πρέπει να τονιστεί ότι τα δεδομένα που αποθηκεύονται δεν έχουν τη μορφή εικόνας. Για την ακρίβεια δεν είναι οι πραγματικές εικόνες των δακτυλικών αποτυπωμάτων που αποθηκεύονται αλλά μια μαθηματική αναπαράσταση αυτών (μαθηματική αναπαράσταση βασικών χαρακτηριστικών των αποτυπωμάτων) ενισχύοντας με

αυτόν τον τρόπο το απόρρητο και την προστασία από τη μη εξουσιοδοτημένη πρόσβαση.

Όπως αναφέρθηκε και παραπάνω η διαδικασία είναι ταχύτατη. Προσφέρει έναν γρήγορο αλλά και βολικό τρόπο ξεκλειδώματος συσκευών ή ελέγχου ταυτότητας ενεργειών χωρίς να είναι αναγκαία η εισαγωγή κάποιου κωδικού ή μοτίβου που είναι πολύ πιθανό για τον οποιονδήποτε λόγο ο χρήστης να το έχει ξεχάσει την εκάστοτε στιγμή. Απλώς με την τοποθέτηση του δακτύλου, που έχουν καταχωρήσει, στον αισθητήρα μπορεί να ολοκληρωθεί η ταυτοποίηση. Πιο συγκεκριμένα γίνεται αντιστοίχιση του καταχωρημένου δεδομένου (πρότυπου) με το αποτύπωμα που σαρώνει εκείνη τη στιγμή ο αισθητήρας και εάν αυτά τα δύο ταιριάζουν τότε επιτρέπεται η πρόσβαση σε ένα ελάχιστο χρονικό διάστημα. Σε αντίθετη περίπτωση η πρόσβαση απορρίπτεται άμεσα. Φυσικά στη διαδικασία περιλαμβάνεται και ο έλεγχος προκειμένου να διαλευκανθεί εάν το δακτυλικό αποτύπωμα εκείνη τη στιγμή ανήκει πραγματικά σε έναν εν ζωή οργανισμό.

Για αυτόν τον σκοπό έχει φτιαχτεί και το Σύστημα Αναγνώρισης Δακτυλικών Αποτυπωμάτων (Fingerprint Recognition System FRS). Το σύστημα αυτό περιλαμβάνει τα ατομικά δακτυλικά μοτίβα για λόγους που σχετίζονται με την ταυτοποίηση. Όπως έχει ήδη αναφερθεί αυτό το βιομετρικό χαρακτηριστικό είναι μοναδικό για κάθε άνθρωπο αφού αποτελεί γεγονός πως διαθέτουν ξεχωριστά μοτίβα κορυφογραμμών ridge patterns).

Ακολουθεί διάγραμμα με τα βήματα που ακολουθούνται [Διάγραμμα 2]



Διάγραμμα 2. Βήματα αυθεντικοποίησης με δακτυλικά αποτυπώματα

2.2.3 Αναγνώριση ίριδας

Άλλο ένα βιομετρικό χαρακτηριστικό το οποίο εδώ και χρόνια έχει χρησιμοποιηθεί για την ταυτοποίηση των χρηστών είναι και η ίριδα. Αποτελεί μία από τις παλαιότερες μεθόδους αυθεντικοποίησης όπως και εκείνη με τα δακτυλικά αποτυπώματα. Η ίριδα είναι το έγχρωμο τμήμα του ματιού το οποίο περιβάλλει την κόρη. Τα μοτίβα της ίριδας είναι διακριτά και παραμένουν αμετάβλητα καθ' όλη τη διάρκεια της ζωής του

ανθρώπου επομένως το συγκεκριμένο χαρακτηριστικό είναι κατάλληλο για την ταυτοποίηση των χρηστών. Ακόμα και τα μονοζυγωτικά δίδυμα έχουν διαφορετικά μοτίβα.

Για τη λήψη των δεδομένων χρειάζονται σαρωτές, συνήθως με υπέρυθρο φως ώστε να φωτίζουν την ίριδα επιτρέποντας τη λήψη δεδομένων υψηλής ανάλυσης. Στη συνέχεια με τη βοήθεια αλγορίθμων εξάγονται οι απαραίτητες πληροφορίες οι οποίες αποτυπώνονται με μια μαθηματική σχέση και δημιουργείται ένα πρότυπο. Το πρότυπο αυτό θα αποθηκευτεί σε μια βάση και όταν ο χρήστης επιδιώξει να συνδεθεί σε μια υπηρεσία θα χρησιμοποιήσει ξανά τον σαρωτή προκειμένου να συλλεχθούν τα δεδομένα της ίριδας του και να γίνει σύγκριση μεταξύ αυτών και του προτύπου.

Πιο συγκεκριμένα όταν ο χρήστης θέλει να συνδεθεί σε μια υπηρεσία ή γενικά να ταυτοποιηθεί (για παράδειγμα να εισέλθει σε κάποιον χώρο) πλησιάζει κοντά στον σαρωτή (εκμεταλλευόμενος το υπέρυθρο φως). Μόλις ολοκληρωθεί η συλλογή των πληροφοριών από την ίριδα (για παράδειγμα τα αυλάκια, η υφή, η ζωτικότητα κλπ) ξεκινάει η προεπεξεργασία αυτών ώστε το δείγμα να είναι όσο το δυνατόν πιο ακριβές (απαλλαγή από παράγοντες που τα αλλοιώνουν). Στη συνέχεια αυτό το δείγμα θα αποτυπωθεί σε μια μαθηματική σχέση η οποία θα συγκριθεί με το πρότυπο που αποθηκεύτηκε στη βάση, με τον ίδιο τρόπο κατά τη διάρκεια της εγγραφής. Επόμενο και τελικό βήμα είναι η σύγκριση αυτών των 2. Αν κατά τη σύγκριση το δείγμα και το πρότυπο κρίνονται όμοια τότε ο χρήστης ταυτοποιείται. Σε αντίθετη περίπτωση η προσπάθεια του χρήστη θα απορριφθεί. Η διαδικασία αυτή παρόλο που είναι πολύπλοκη είναι ταχύτατη.

2.2.4 Αναγνώριση φωνής

Τα βιομετρικά χαρακτηριστικά όπως έχει αποδειχτεί είναι πιο δύσκολο να πλαστογραφηθούν ενώ παράλληλα είναι πολύ πιο βολικά για τους χρήστες, αφού δε χρειάζεται να θυμούνται πληθώρα κωδικών πρόσβασης ή να φέρουν κάποιο φυσικό διακριτικό (token） το οποίο πολύ εύκολα θα μπορούσε να κλαπεί είτε να χαθεί. Είναι μέρος του ατόμου και δεν υπάρχει πιθανότητα να αποσπαστούν από αυτόν τουλάχιστον οικειοθελώς. Ένα βιομετρικό χαρακτηριστικό που κεντρίζει ιδιαίτερα το ενδιαφέρον είναι η αναγνώριση φωνής (voice recognition) το οποίο δε μπορεί να προσδιοριστεί επακριβώς, εάν ανήκει στα φυσικά βιομετρικά χαρακτηριστικά ή σε εκείνα που σχετίζονται με τη συμπεριφορά.

Η φωνή παράγεται μέσω φυσικών μηχανισμών του ανθρώπινου σώματος. Περιλαμβάνει τον συντονισμό διαφορετικών φυσιολογικών συστατικών όπως είναι οι πνεύμονες, οι φωνητικές χορδές, ο λάρυγγας , το στόμα και οι ρινικές κοιλότητες. Η μοναδική δομή και τα χαρακτηριστικά αυτών των φυσικών στοιχείων συμβάλλουν στην ποιότητα, τον τόνο , τον συντονισμό όπως και άλλες ακουστικές ιδιότητες της φωνής. Ωστόσο, η φωνή επηρεάζεται από διάφορους εξωτερικούς παράγοντες όπως τα μοτίβα ομιλίας, η προφορά, ο τονισμός και η ένταση. Τα στοιχεία αυτά διαμορφώνονται βάσει των περιβαλλοντικών, πολιτιστικών αλλά και προσωπικών

παραγόντων. Η φωνή μπορεί να αντικατοπτρίζει την εκπαίδευση, την ανατροφή, τη συναισθηματική κατάσταση αλλά και το προσωπικό ύφος επικοινωνίας κάποιου. Για αυτό τον λόγο δεν μπορεί να καταταχθεί ξεκάθαρα σε μια εκ των δύο βιομετρικών κατηγοριών.

Άλλες ονομασίες με τις οποίες μπορεί να συναντηθεί αυτή η τεχνική είναι η αναγνώριση ηχείου (speaker recognition) ή έλεγχος ταυτότητας φωνής (voice authentication). Η έννοια της αναγνώρισης της φωνής δεν πρέπει να ταυτιστεί με εκείνη της αναγνώρισης της ομιλίας (speech recognition). Η δεύτερη τεχνική χρησιμοποιείται σε εφαρμογές ομιλίας σε κείμενα ή εικονικούς βοηθούς. Βασικός στόχος της τεχνολογίας αυτής είναι η αναγνώριση του νοήματος της λέξης ή φράσης που ειπώνεται, δηλαδή επικεντρώνεται κυρίως στη λεκτική γλώσσα. Για παράδειγμα σε εφαρμογές ή κατά τη διάρκεια μιας τηλεφωνικής κλήσης μπορεί ο πελάτης να κληθεί να αξιολογήσει την ίδια την υπηρεσία δίνοντας κάποιο βαθμό σε μια προκαθορισμένη κλίμακα.

Φυσικά η παραπάνω υπόθεση έχει νόημα όταν στην άλλη πλευρά του τηλεφώνου δεν είναι κάποιο φυσικό άτομο της υπηρεσίας. Σε αυτό ακριβώς το σημείο θα γίνει χρήση της τεχνολογίας της αναγνώρισης ομιλίας ώστε να γίνει η αντιστοίχιση του βαθμού που θα αναφέρει ο ομιλητής με εκείνον που υπάρχει στην ορισμένη από τους προγραμματιστές λίστα. Επομένως δε θα μπορούσε να χρησιμοποιηθεί ως τεχνική για την επαλήθευση ενός ατόμου βάση των φωνητικών του χαρακτηριστικών.

Τα συστήματα που χρησιμοποιούν ως βιομετρικό χαρακτηριστικό τη φωνή εκμεταλλεύονται κάποιες μεθόδους παραμετροποίησης της ανθρώπινης ομιλίας ή την αντιστοίχιση/βαθμολόγηση προτύπων (όπως άλλωστε αναφέρθηκε και σε προηγούμενο σημείο) με σκοπό τη δημιουργία μιας μοναδικής «υπογραφής» θα μπορούσε να πει κάποιος για την αναγνώριση του ατόμου ή αλλιώς ένα μοναδικό «φωνητικό αποτύπωμα». Η φωνή και η παραγωγή ανθρώπινης ομιλίας είναι μια διαδικασία που περιλαμβάνει τους πνεύμονες , τις φωνητικές χορδές και τις φωνητικές οδούς.

Μόλις ένα άτομο προσπαθήσει να μιλήσει , ο αέρας που βρίσκεται μέσα στους πνεύμονες αποβάλλεται και περνάει μέσα από τις φωνητικές χορδές οι οποίες με τη σειρά τους διαστέλλονται (dilate) ή επεκτείνονται (expand) επιτρέποντας κατ' αυτόν τον τρόπο στη ροή του αέρα να παράγει ήχους. Έπειτα ο αέρας συντονίζεται και στην πορεία αναδιαμορφώνεται από τη φωνητική οδό που αποτελείται από πολλαπλά όργανα (λαιμός , στόμα, γλώσσα μύτη, χείλη, δόντια, γνάθος). Η διαμόρφωση αυτών (των οργάνων) , η αλληλεπίδραση μεταξύ τους και τέλος οι κινήσεις τους, είναι σε θέση να παράγουν μοναδικούς ήχους αλλά και να αλλάξουν τα φωνητικά κύματα για κάθε ένα άτομο.

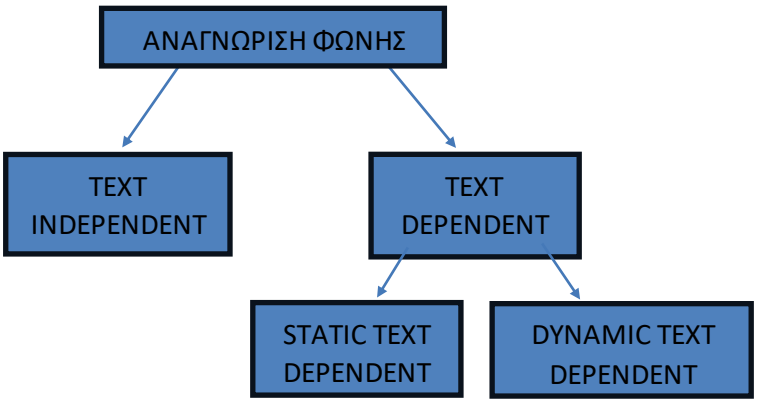
Υπάρχουν δύο ειδών τρόποι προσέγγισης για την αναγνώριση φωνής. Τα δύο είδη διακρίνονται σε :

- Αναγνώριση ανεξάρτητη κειμένου (Text Independent)
- Αναγνώριση εξαρτώμενη κειμένου (Text Dependent

Η αναγνώριση η οποία είναι ανεξάρτητη κειμένου (Text Independent) εκτελεί την επαλήθευση της ταυτότητας του χρήστη χρησιμοποιώντας μια προφορική φράση ως

μέσο πρόσβασης. Δηλαδή ο ομιλητής θα έχει τη δυνατότητα να αρθρώσει οποιαδήποτε φράση θέλει. Από την άλλη πλευρά η αναγνώριση η οποία είναι εξαρτημένη από κάποιο κείμενο χρησιμοποιεί φράσεις πρόσβασης που έχουν χρησιμοποιηθεί κατά την εγγραφή του χρήστη ώστε να γίνει στη συνέχεια η επαλήθευση. Άρα ο ομιλητής σε αυτήν την περίπτωση δεν έχει τη δυνατότητα να πει κάτι που εκείνος θα ήθελε. Η φράση που θα του ζητηθεί είναι προκαθορισμένη.

Η αναγνώριση φωνής εξαρτώμενη από κάποιο κείμενο (Text Dependent) χωρίζεται επίσης σε 2 υποκατηγορίες. Οι υποκατηγορίες αυτές είναι του στατικού κειμένου και του δυναμικού. Στην αναγνώριση φωνής που εξαρτάται από στατικό κείμενο (Static text-dependent) χρησιμοποιείται η ίδια φράση πρόσβασης συνεχώς για την επαλήθευση της ταυτότητας του ατόμου. Αντιθέτως στον δυναμικό φωνητικό έλεγχο (Dynamic text-dependent) που εξαρτάται επίσης από κείμενο η φράση πρόσβασης δεν είναι προκαθορισμένη αλλά κάθε φορά παράγεται μια νέα ακολουθία η οποία θα μπορούσε να αποτελείται από αριθμούς. Βέβαια είναι απαραίτητο αυτή η φράση πρόσβασης να υπάρχει καταχωρημένη στη βάση δεδομένων (να έχει καταγραφεί).



Διάγραμμα 3. Κατηγορίες μεθόδων αναγνώρισης φωνής

Η διαδικασία αναγνώρισης της φωνής απαιτεί ως βασικό εργαλείο τη χρήση μικροφώνου. Με το μικρόφωνο ηχογραφείται η φωνή και αυτό το δείγμα θα αποθηκευτεί και στη συνέχεια θα χρησιμοποιηθεί ως ένα πρότυπο αναφοράς σε μελλοντικές προσπάθειες ταυτοποίησης. Οι μοναδικές φωνητικές ιδιότητες που αναλύονται είναι η διάρκεια , η ένταση , η δυναμική και τέλος η αντιληπτή συχνότητα των ηχητικών δονήσεων που παράγονται από τις φωνητικές χορδές (pitch). Αυτού του είδους τεχνικής ταυτοποίησης είναι χρήσιμη για όλες τις υπηρεσίες που χρησιμοποιούν κάποια μέθοδο ταυτοποίησης ενώ επίσης μπορεί να χρησιμοποιηθεί και κατά τη διάρκεια κλήσεων μεταξύ υπηρεσίας και πελάτη με στόχο την υποστήριξη του δεύτερου. Ένα παράδειγμα υπηρεσίας που χρησιμοποιεί την αναγνώριση φωνής ως μέσο επαλήθευσης του πελάτη είναι η τράπεζα Barclays.

Η τράπεζα Barclays έχει εντάξει στις δυνατότητες που παρέχει στους πελάτες της και τη χρήση αναγνώρισης φωνής [3]. Ένας πελάτης μπορεί να κάνει εγγραφή και να έχει πρόσβαση σε αυτήν την υπηρεσία αφού πρώτα επικοινωνήσει με τα σχετικά άτομα της τράπεζας. Πριν ξεκινήσει να χρησιμοποιεί αυτήν την τεχνική θα πρέπει να έχει προηγηθεί μια πληθώρα επικοινωνιών (συνεδρίες) προκειμένου η τράπεζα να

συλλέξει το απαιτητό υλικό (μια ποσότητα που έχει οριστεί σύμφωνα με τις ανάγκες της εφαρμογής). Αυτό το υλικό θα αποθηκευτεί σε μια βάση δεδομένων και στη συνέχεια θα είναι διαθέσιμο ώστε κάθε φορά που θα ζητάει να ταυτοποιηθεί ο χρήστης να υπάρχει το κατάλληλο υπόβαθρο για μια αξιόπιστη επαλήθευση.

2.2.5 Αναγνώριση μοτίβου φλεβών στις παλάμες

Πέρα από τα δακτυλικά αποτυπώματα η ταυτοποίηση μπορεί να επιτευχθεί και μέσω των φλεβών της παλάμης. Η εταιρεία Fujitsu έχει κατασκευάσει το Palmsecure μια βιομετρική τεχνολογία ελέγχου ταυτοποίησης η οποία χρησιμοποιεί ως μέσο επαλήθευσης της ταυτότητας των ατόμων την αναγνώριση φλέβας της παλάμης. Οι φλέβες που βρίσκονται στις παλάμες των ανθρώπων δημιουργούν ένα εξίσου μοναδικό μοτίβο όπως και τα δακτυλικά αποτυπώματα. Για να συλληφθεί το συγκεκριμένο μοτίβο θα πρέπει η παλάμη να εκτεθεί σε υπέρυθρο φως σε κοντινή απόσταση.

Οι φλέβες στις παλάμες έχουν προσελκύσει προσφάτως το παγκόσμιο ενδιαφέρον καθώς το μοτίβο τους διαφοροποιείται από άνθρωπο σε άνθρωπο (ακόμη και τα δίδυμα έχουν διαφορετικό μοτίβο) καθιστώντας αυτές έναν ακόμη τρόπο ταυτοποίησης ενός χρήστη (βιομετρική ταυτοποίηση). Το μοτίβο αυτό δεν αλλάζει κατά το πέρασμα του χρόνου, παραμένοντας αναλλοίωτο καθ' όλη τη διάρκεια της ζωής του ανθρώπου. Η αναγνώριση φλέβας της παλάμης λειτουργεί καταγράφοντας, όπως αναφέρθηκε, τα μοτίβα των φλεβών που υπάρχουν στην παλάμη του χεριού ενός ατόμου. Το μοτίβο αυτό είναι δύσκολο να αναπαραχθεί από ένα άλλο άτομο με αποτέλεσμα να αποτελεί μια αξιόπιστη βιομετρική μέθοδο.

Ωστόσο δεν είναι μόνο η μοναδικότητα του μοτίβου των φλεβών που τις καθιστά αξιόπιστες ως έναν ακόμα βιομετρικό τρόπο ταυτοποίησης ενός χρήστη. Τα σχέδια των φλεβών δεν είναι ορατά ή προσβάσιμα καθώς βρίσκονται εσωτερικά του δέρματος γεγονός που ενισχύει την ασφάλεια που προσδίδουν. Αντίθετα με άλλες βιομετρικές μεθόδους ταυτοποίησης τα μοτίβα των φλεβών όπως τα δακτυλικά αποτυπώματα, τα χαρακτηριστικά του προσώπου, η φωνή ή η βάδιση είναι λιγότερο πιθανό να επηρεαστούν από εξωτερικούς παράγοντες όπως η βρωμιά, η υγρασία, η γήρανση του δέρματος, η θερμοκρασία του σώματος, η υγεία (π.χ με ένα κρυολόγημα η φωνή ενός ατόμου είναι πιθανό να αλλάξει) [11].

Μια ακόμη σημαντική λεπτομέρεια που θα πρέπει να σημειωθεί είναι ο ανέπαφος χαρακτήρας της τεχνολογίας αυτής. Η αναγνώριση φλέβας είναι μια τεχνολογία ανέπαφων που σημαίνει ότι η φυσική επαφή μεταξύ της ίδιας της παλάμης και του αισθητήρα να μην είναι απαιτητή. Έχοντας περάσει από την πανδημία του Covid – 19 , οι άνθρωποι αποζητούν την καθαριότητα και την υγιεινή ολοένα και περισσότερο στις εφαρμογές και τις υπηρεσίας που χρησιμοποιούν , κάτι που αναδεικνύει τον ανέπαφο χαρακτήρα της ταυτοποίησης αυτής ως μια αρκετά καλή επιλογή.

Το σύστημα PalmSecure καταγράφει την εικόνα της φλέβας της παλάμης σαρώνοντας το χέρι του χρήστη πάνω από έναν αισθητήρα. Ο αισθητήρας αυτός εκπέμπει εγγύς (κοντινό) υπέρυθρο φως, το οποίο απορροφάται από την

αιμοσφαιρίνη [1] στις φλέβες με αποτέλεσμα να εμφανίζονται ως σκοτεινές γραμμές στην εικόνα που λήφθηκε. Πιο συγκεκριμένα οι φλέβες αποτυπώνονται ως ένα μαύρο μοτίβο [12]. Η τεχνολογία της απεικόνισης κοντά στο υπέρυθρο φως είναι που εξασφαλίζει ακριβή και αξιόπιστη ανίχνευση των μοτίβων των φλεβών.

Μόλις η εικόνα των φλεβών της παλάμης ληφθεί , η εικόνα υποβάλλεται σε μια διαδικασία προ επεξεργασίας για τη βελτίωση της ποιότητάς της και την απομάκρυνση τυχόν θορύβου. Αυτό το στάδιο της προ επεξεργασίας πιο συγκεκριμένα περιλαμβάνει τεχνικές όπως η βελτίωση της αντίθεσης και η μείωση του. Η επακόλουθη ανάλυση μπορεί να πραγματοποιηθεί πιο αποτελεσματικά εάν έχει βελτιωθεί προηγουμένως η ποιότητα της εικόνας.

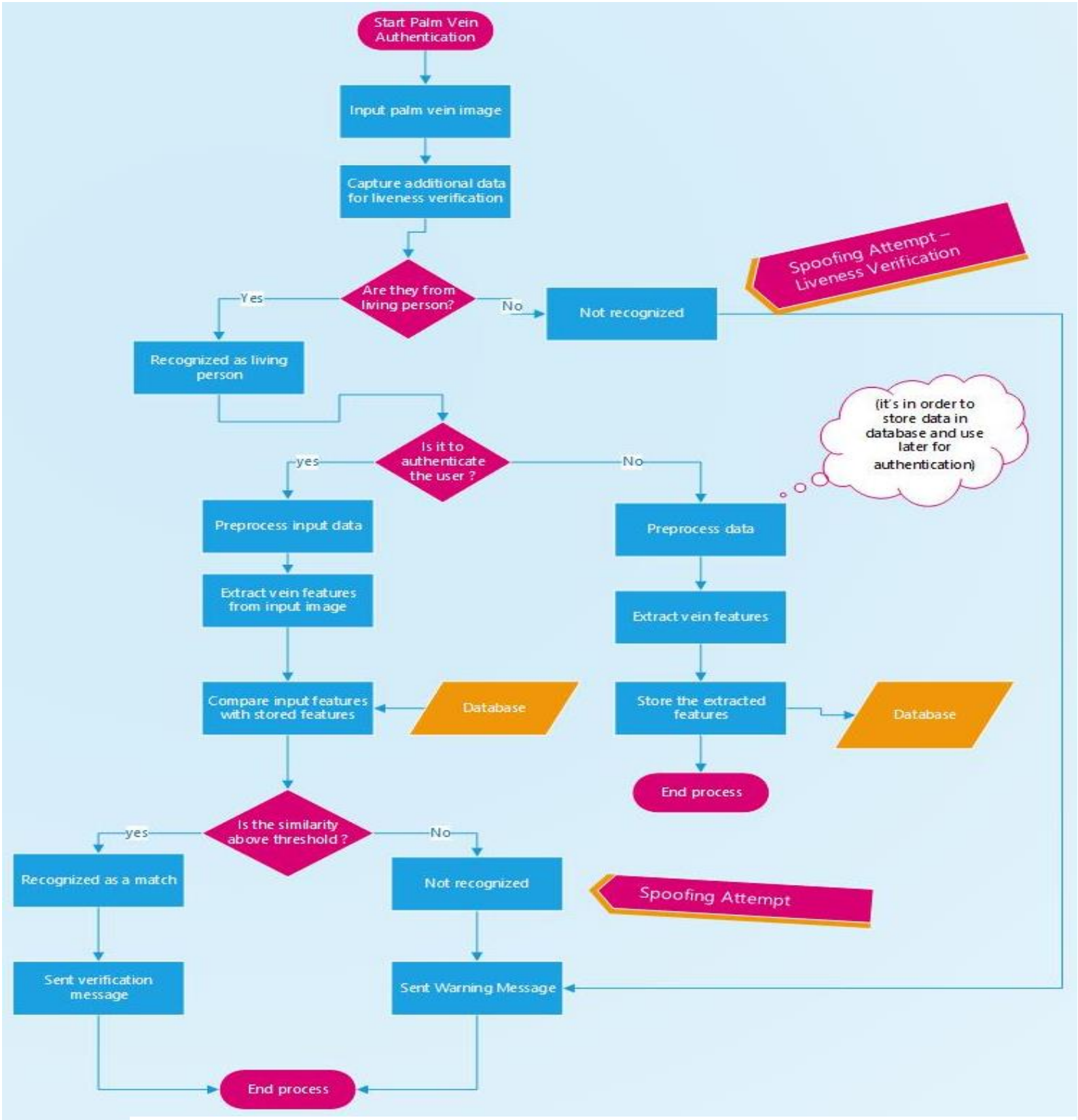
Αμέσως μετά ακολουθεί η εξαγωγή των χαρακτηριστικών από το δείγμα. Εφαρμόζονται διάφοροι αλγόριθμοι και τεχνικές στο προ επεξεργασμένη δείγμα της φλέβας της παλάμης για τον εντοπισμό του μοναδικού μοτίβου των φλεβών. Οι αλγόριθμοι αυτοί, αναλύουν τη θέση, το σχήμα και τον προσανατολισμό των φλεβών εξάγοντας στη συνέχεια τα διακριτικά χαρακτηριστικά που διαφοροποιούν το σχέδιο των φλεβών στην παλάμη ενός ανθρώπου από εκείνο ενός άλλου.

Κατόπιν τα εξαγόμενα χαρακτηριστικά χρησιμοποιούνται για τη δημιουργία ενός προτύπου ή μιας μαθηματικής αναπαράστασης του σχεδίου της φλέβας της παλάμης. Αυτό το πρότυπο θα χρησιμοποιηθεί μελλοντικά ως αναφορά για τις επερχόμενες συγκρίσεις που θα πραγματοποιηθούν κατά τη διαδικασία ελέγχου της ταυτότητας του χρήστη. Το πρότυπο αυτό περιλαμβάνει τις σχετικές πληροφορίες που απαιτούνται για την επαλήθευση του χρήστη βασισμένο στις φλέβες. Τα δεδομένα αυτά αποθηκεύονται σε μια βάση δεδομένων.

Κατά τη διαδικασία επαλήθευσης, δηλαδή όταν ο χρήστης θα προσπαθήσει να ταυτοποιήσει τον εαυτό του, λαμβάνεται άλλο ένα δείγμα των φλεβών της παλάμης που υποβάλλεται ακολούθως στα βήματα προ επεξεργασίας και εξαγωγής χαρακτηριστικών που αποτυπώθηκαν πιο πάνω. Τα νέα εξαγόμενα χαρακτηριστικά, δηλαδή από την καινούρια εικόνα, συγκρίνονται με το αποθηκευμένο πρότυπο. Ο βαθμός ομοιότητας ή διαφοράς μεταξύ αυτών υπολογίζεται ώστε να προσδιοριστεί εάν το άτομο έχει ταυτοποιηθεί ή όχι. Η διαδικασία αυτή περιλαμβάνει διάφορους εξελιγμένους αλγόριθμους που λαμβάνουν υπόψιν τους παράγοντες για να προκύψει ένα όσο το δυνατόν πιο ακριβές αποτέλεσμα.

[1] Η αιμοσφαιρίνη (Hgb) είναι η χρωστική ουσία των ερυθροκυττάρων που μεταφέρει το οξυγόνο.

Ακολουθεί ένα διάγραμμα ροής το οποίο δημιουργήθηκε αποτυπώνοντας βάσει των βημάτων που ακολουθούνται καθ' όλη τη διαδικασία [Διάγραμμα 4].



Διάγραμμα 4. Διάγραμμα ροής με τα βήματα για την αναγνώριση φλεβών παλάμης

Οι ακριβείς λειτουργίες και οι αλγόριθμοι που χρησιμοποιεί η PalmSecure είναι αποκλειστικές και δεν έχουν αποκαλυφθεί δημοσίως. Ωστόσο, θα μπορούσε κάποιος να υποθέσει ότι για την επίτευξη του σκοπού του συστήματος έχουν υιοθετηθεί γνώσεις από τον τομέα της μηχανικής μάθησης (machine learning) και της αναγνώρισης προτύπων. Η μηχανική μάθηση αποτελεί ένα εργαλείο , σύνολο τεχνικών της αναγνώρισης προτύπων. Παρέχει τα μέσα για την αυτοματοποίηση της διαδικασίας της αναγνώρισης εκπαιδεύοντας τα μοντέλα σε δεδομένα και χρησιμοποιώντας τα για την αναγνώριση των προτύπων στα νέα δεδομένα. Οι αλγόριθμοι εποπτευόμενης μάθησης (supervised learning), μάθησης χωρίς επίβλεψη (unsupervised learning), και βαθιάς μάθησης (deep learning) χρησιμοποιούνται όταν επιδιώκεται η αναγνώριση προτύπων. Στις βιομετρικές μεθόδους σε εφαρμογές χρησιμοποιούνται ευρέως τα νευρωνικά δίκτυα για τη βελτίωση της ακρίβειας και της ευρωστίας.

Για να παράξει τα επιθυμητά αποτελέσματα μια βιομετρική μέθοδος χρειάζονται φυσικά και τα κατάλληλα εργαλεία. Τα εργαλεία ή αλλιώς συσκευές που έχουν προσαρτημένο έναν αισθητήρα πάνω τους θα πρέπει να είναι αξιόπιστο. Αυτό σημαίνει ότι η τιμή του θα είναι ακριβή, ενώ υπάρχει και η πιθανότητα να κυκλοφορήσουν συσκευές με μια πάρα πολύ φθηνή τιμή οι οποίες βέβαια δε θα αποδίδουν στο 100% των απαιτήσεων. Ωστόσο, το χαμηλό κόστος σε μία λογική κλίμακα φυσικά, δε θα πρέπει να αποτρέψει τον χρήστη από το να το αγοράσει και να το χρησιμοποιήσει. Η ενημέρωση και η έρευνα είναι το «Α και το Ω» όταν οι ενέργειες κάποιου συμπεριλαμβάνουν μια τέτοια διαδικασία.

Η PalmSecure έχει καταφέρει να μειώσει ως ένα βαθμό το κόστος της αγοράς (οι συσκευές παρόλα αυτά δε μπορούν να θεωρηθούν οικονομικές) ενώ παράλληλα διαφυλάσσουν την αξιοπιστία και την αποτελεσματικότητά τους [11]. Εδώ [13] μπορεί κάποιος να βρει περισσότερες πληροφορίες σχετικά με τις συσκευές που θα χρειαστεί αν επιδιώξει να κάνει ο ίδιος δοκιμές και να παρατηρήσει τον τρόπο που λειτουργεί η εφαρμογή και να πειραματιστεί.

2.2.6 Αναγνώριση προσώπου ή χαμόγελου

Ένας ακόμη τρόπος ταυτοποίησης είναι και η βιομετρική μέθοδος της αναγνώρισης του προσώπου ή του χαμόγελου. Ουσιαστικά ο τρόπος αυτός χρησιμοποιεί τα χαρακτηριστικά του προσώπου προκειμένου να ταυτοποιήσει το εκάστοτε άτομο και στη συνέχεια να πραγματοποιηθεί η ενέργεια που θέλει ο χρήστης. Πιο συγκεκριμένα η τεχνολογία αυτή βασίζεται στην τρισδιάστατη οπτική αναγνώριση- απεικόνιση η οποία συλλέγει πληροφορίες σχετικά με τα στοιχεία του προσώπου όπως για παράδειγμα η θέση της μύτης, η υφή του προσώπου, το μέγεθος του στόματος, οι αποστάσεις κάποιων χαρακτηριστικών σε σχέση με άλλα ή ακόμη και τα μάτια.

Αναγνώριση προσώπου:

Η αναγνώριση προσώπου είναι και αυτή άλλη μια μέθοδος που χρησιμοποιείται ήδη αρκετά χρόνια. Όπως αναφέρθηκε και προηγουμένως τα στοιχεία που

συλλέγονται ποικίλουν (υφή, αποστάσεις από χαρακτηριστικά, σημάδια, οι γωνίες των ματιών, τα ακραία σημεία των χειλιών κλπ). Τα στοιχεία αυτά μαζί προσδίδουν μια μοναδική ταυτότητα στον άνθρωπο. Η τεχνική βασίζεται στη συλλογή και ανάλυση του προσώπου δημιουργώντας ένα πρότυπο (template , faceprint).

Εν μέσω της πανδημίας του COVID-19, η τεχνική αυτή εξελίχθηκε ακόμη περισσότερο, ώστε να προσαρμοστεί στις συνθήκες της καθημερινότητας και στις ανάγκες των ανθρώπων. Πιο συγκεκριμένα, η χρήση της μάσκας ήταν υποχρεωτική παντού. Επομένως σε όλους σχεδόν τους χώρους είτε εσωτερικούς είτε εξωτερικούς, ο χρήστης δεν είχε τη δυνατότητα να χρησιμοποιήσει τις υπηρεσίας στις οποίες χρειαζόταν αναγνώριση του προσώπου τους προκειμένου να ταυτοποιηθούν.

Ακριβώς για αυτόν τον λόγο αναβαθμίστηκαν οι ήδη εξελιγμένοι αλγόριθμοι προκειμένου να δοθεί η δυνατότητα στους χρήστες να επιλέξουν αν επιθυμούν να γίνεται η ταυτοποίηση τους τόσο με τη χρήση ή όχι της μάσκας. Ωστόσο, στην περίπτωση που ένας χρήστης δεν επιλέξει την ταυτοποίηση με μάσκα τότε όταν την φοράει το πρόσωπο του δεν είναι διακριτό και άρα δεν αναγνωρίζεται.

Αναγνώριση χαμόγελου:

Την τεχνική αυτή φαίνεται να έχουν προσεγγίσει μερικές εταιρείες οι οποίες προσφέρουν χρηματοοικονομικές υπηρεσίες. Ήδη από το 2017 έχει υιοθετηθεί ως μέσο πληρωμής από τον κινεζικό όμιλο Alibaba. Ο όμιλος αυτός παρέχει τις υπηρεσίες του διεθνώς αν και φανερά η κύρια αγορά του είναι στην Κίνα. Μια από τις κύριες πλατφόρμες πληρωμών του για κινητές συσκευές είναι Alipay, το οποίο αποτελεί πλέον μια ολοκληρωμένη πλατφόρμα χρηματοοικονομικών υπηρεσιών. Οι χρήστες μπορούν να πραγματοποιήσουν μέσω του κινητού τηλεφώνου τους διάφορες ενέργειες όπως μεταφορά χρημάτων ή πληρωμές λογαριασμών. Πως όμως η αναγνώριση προσώπου σχετίζεται με την Alipay;

Η πλατφόρμα Alipay έχει αναπτύξει το «Smile to pay» μια τεχνική που βασίζεται στην τεχνολογία του FRP. Το FRP (facial recognition payment), δηλαδή η πληρωμή με τη χρήση αναγνώρισης προσώπου είναι ένα πολλά υποσχόμενο εργαλείο το οποίο αναπτύχθηκε κυρίως επί της πανδημίας του COVID-19, καθώς η χρήση ανέπαφων μεθόδων κλήθηκε σχεδόν απαραίτητη. Το FRP έγινε αποδεκτό κάτω από αυτές τις συνθήκες και φυσικά συμβαδίζει με τη ραγδαία εξέλιξη της τεχνολογίας και την καθημερινότητα των ανθρώπων. Πάντα μιλώντας για τον τρόπο που δρα το FRP στην Κίνα, κατόπιν έρευνας ήδη το 2021 πάνω από 495 εκατομμύρια άνθρωποι έκαναν εγγραφή ώστε να μπορούν να χρησιμοποιούν την αναγνώριση προσώπου. Μάλιστα είναι τόσο ευρέως διαδεδομένο ώστε συσκευές FRP έχουν τοποθετηθεί σε υπεραγορές (supermarkets) , λιανοπωλεία ακόμη και στους αυτόματους πωλητές [23].

Τα βήματα που ακολουθούνται προκειμένου να γίνει η χρήση αυτής της τεχνολογίας δεν είναι δυσνόητα. Αρχικά το άτομο την πρώτη φορά που θα επιδιώξει να χρησιμοποιήσει την τεχνική αυτή θα πρέπει να ανεβάσει μια «φωτογραφία» με το πρόσωπό του (selfie), η οποία έχει επαληθευτεί από τον λογαριασμό του σε μια

πλατφόρμα πληρωμών όπως η Alipay. Λέγοντας φωτογραφία εννοείται μια συλλογή από λεπτομέρειες του προσώπου του χρήστη και όχι η στατική εικόνα, όπως είναι γνωστή. Φυσικά θα πρέπει να αποδεχτούν τους όρους ώστε οι πληροφορίες του προσώπου τους να αποθηκευτούν σε μια βάση δεδομένων. Έπειτα το μόνο που χρειάζεται είναι μπροστά σε μια τέτοια συσκευή να επιλεχθεί το κουμπί «Face recognition pay» και να κοιτάξουν την οθόνη (ο χρήστης μπορεί ακόμη και να φοράει προστατευτική μάσκα).

Με αυτόν τον τρόπο η συσκευή σαρώνει το πρόσωπο και ταυτοποιεί τον χρήστη που χαμογελάει ώστε να ολοκληρωθεί η συναλλαγή. Ένα δεύτερο είδος ταυτοποίησης εφαρμόζεται σε περιπτώσεις που η εικόνα δεν είναι ευδιάκριτη και περιλαμβάνει την πληκτρολόγηση ψηφίων από το κινητό τηλέφωνο που έχει καταχωρηθεί στην πλατφόρμα. Βέβαια, το θέμα της διπλωματικής δεν είναι η γρήγορη και εύκολη πληρωμή σε διάφορα καταστήματα αλλά η ταυτοποίηση δίχως τη χρήση συνθηματικών. Η μέθοδος όμως που χρησιμοποιεί η Alipay, εμπεριέχει ακριβώς αυτή την προσέγγιση.

Ουσιαστικά το «Smile to pay» με αλγόριθμους αναγνώρισης προσώπου αναλύει τα μοναδικά χαρακτηριστικά στο πρόσωπο του χρήστη, ο οποίος θα χρειαστεί να χαμογελάσει με σκοπό την αυθεντικοποίησή του [24]. Κατ' αυτόν τον τρόπο προστίθεται ένας επιπλέον έλεγχος-κανόνας που αυξάνει την ασφάλεια. Η επιπλέον ασφάλεια επιτυγχάνεται καθώς το χαμόγελο συμπεριλαμβάνει δυναμικές εκφράσεις και κινήσεις των μυών, στοιχεία τα οποία είναι δύσκολο να αντιγραφούν ή να βρεθούν πανομοιότυπα σε μια στατική εικόνα. Επιπλέον το σύστημα επιδιώκει να διασφαλίσει πως ο χρήστης είναι παρών και αλληλοεπιδρά με αυτό. Το επίπεδο πολυπλοκότητας που προσδίδει το χαμόγελο μειώνει τις πιθανότητες κάποιος κακόβουλος να προσπαθήσει να χρησιμοποιήσει στατικές εικόνες για να προσπελάσει το σύστημα.

Η τελική «εικόνα» είναι αποτέλεσμα σάρωσης 3D που αφορά το περίγραμμα του προσώπου και του σχήματος μέσω υπέρυθρων κουκκίδων στο πρόσωπο του χρήστη. Ακριβώς με την ίδια διαδικασία λαμβάνεται η εικόνα τόσο στην πλατφόρμα Alipay όσο και στο σύστημα στο οποίο ο χρήστης θα πληρώσει. Οι δύο αυτές εικόνες συγκρίνονται και αν συμπίπτουν τότε η ταυτοποίηση είναι επιτυχής και η διαδικασία πληρωμής ολοκληρώνεται. Σε αντίθετη περίπτωση , η πληρωμή δεν πραγματοποιείται.

Μια αντίστοιχη προσέγγιση επιδίωξε και η Mastercard το 2022 η οποία θα δίνει τη δυνατότητα στους χρήστες να χαμογελάνε ή να κάνουν ένα νεύμα στο σύστημα προκειμένου να γίνει η ταυτοποίησή τους και κατ΄ επέκταση η πληρωμή [26]. Σύμφωνα με δεδομένα από το 2022 το πρόγραμμα αυτό εκείνη την περίοδο ήταν σε ένα πρώιμο στάδιο και οι δοκιμές προγραμματίστηκαν να γίνουν στη Βραζιλία σε 5 συγκεκριμένες υπεραγορές [25].

2.2.7 Σύγκριση των βιομετρικών μεθόδων με φυσικά χαρακτηριστικά

Σε αυτή την υπό ενότητα θα γίνει μια σύντομη σύγκριση μεταξύ των βιομετρικών φυσικών χαρακτηριστικών και των αντίστοιχων μεθόδων τους. Θα παρουσιαστεί πίνακας που συμπεριλαμβάνει τα φυσικά χαρακτηριστικά και μερικές πληροφορίες σχετικά με τις μεθόδους αυθεντικοποίησης που έχουν ήδη αναφερθεί.

Ακολουθεί ο πίνακας [Πίνακας 3]

Χαρακτηριστικό	Αισθητήρας	Μοναδικότητα	Αμετάβλητα	Κόστος
Δακτυλικά αποτυπώματα	Σαρωτής δακτυλικών αποτυπωμάτων	Πολύ υψηλή	Όχι	Μέτριο
Ίριδα	Σαρωτής ίριδας	Πολύ υψηλή	Ναι	Υψηλό
Φωνή	Μικρόφωνο	Υψηλή	Όχι	Χαμηλό
Μοτίβο φλεβών παλάμης	Σαρωτής παλάμης	Πολύ υψηλή	Ναι	Υψηλό
Πρόσωπο	Κάμερα	Μέτρια	Όχι	Χαμηλό
Χαμόγελο	Κάμερα – Ανιχνευτής χαμόγελου	Μέτρια	Όχι	Μέτριο

Πίνακας 3. Σύγκριση βιομετρικών μεθόδων με φυσικά χαρακτηριστικά

- Αισθητήρας: Τρόπος με τον οποίο πραγματοποιείται η συλλογή των πληροφοριών για το φυσικό χαρακτηριστικό.
- Μοναδικότητα: Αναφέρεται στη μοναδικότητα που προσδίδουν αυτά τα χαρακτηριστικά (για παράδειγμα μεταξύ διδύμων)
- Αμετάβλητα: Πόσο ακατάβλητα παραμένουν τα χαρακτηριστικά κατά την πάροδο του χρόνο
- Κόστος: Αναφέρετε στο κόστος εξοπλισμού

Αξίζει να γίνει μια σύντομη ανάλυση στις στήλες Μοναδικότητα και Αμετάβλητα.

Δακτυλικά αποτυπώματα:

Τα δακτυλικά αποτυπώματα είναι μοναδικά και μάλιστα μεταξύ μονοζυγωτικών διδύμων. Ωστόσο, δε μπορούν α θεωρηθούν εντελώς αμετάβλητα καθώς διάφοροι εξωτερικοί παράγοντες μπορούν να τα επηρεάσουν (κάψιμο, λάσπη)

Ίριδα:

Η ίριδα είναι μοναδική για κάθε άτομο και παραμένει αμετάβλητη κατά την πάροδο του χρόνου και δεν επηρεάζεται από ασθένειες.

Φωνή:

Η φωνή δεν είναι ένα απόλυτα μοναδικό χαρακτηριστικό καθώς υπάρχουν περιπτώσεις όπου ένα άτομο μπορεί να μιμηθεί τη φωνή κάποιου άλλου σε τεράστιο

βαθμό. Επίσης αποτελεί χαρακτηριστικό που επηρεάζεται από την υγεία του ατόμου (τραχειοτομία, απλό κρυολόγημα) και την ηλικία του.

Μοτίβο φλεβών παλάμης:

Το μοτίβο των φλεβών της παλάμης είναι μοναδικό για κάθε άτομο και παραμένει αμετάβλητο με την πάροδο του χρόνου.

Πρόσωπο:

Το πρόσωπο δεν είναι ένα απόλυτα μοναδικό φυσικό χαρακτηριστικό, Οι περιπτώσεις των διδύμων αποτελούν μια συνεχή πρόκληση για τη μέθοδο αυτή. Επιπλέον επηρεάζεται από εξωτερικούς παράγοντες (σκοτάδι, ατύχημα που αλλοιώνει τα χαρακτηριστικά του προσώπου) και την ηλικία του αλλά.

Χαμόγελο:

Για τους ίδιους λόγους όπως και το πρόσωπο, το χαμόγελο δε μπορεί να θεωρηθεί ένα απόλυτα μοναδικό χαρακτηριστικό ή εντελώς αμετάβλητο.

2.3 Αμφισβήτηση και αρνητικές πτυχές των βιομετρικών μεθόδων βάσει φυσικών χαρακτηριστικών

Ο βιομετρικός έλεγχος ταυτοποίησης αναφέρεται στη διαδικασία επαλήθευσης της ταυτότητας ενός ατόμου βασιζόμενη στα μοναδικά χαρακτηριστικά του είτε αυτά είναι φυσιολογικά (δακτυλικά αποτυπώματα, φλέβες παλάμης, φωνή κλπ) είτε είναι χαρακτηριστικά συμπεριφοράς (βάδιση , κινήσεις χεριών κλπ). Οι μέθοδοι που σχετίζονται με τον βιομετρικό έλεγχο αποτελούν μια νέα τάση τα τελευταία χρόνια τόσο λόγω της ασφάλειας που παρέχουν όσο και της βολικότητας.

Τα τεχνολογικά μέσα που έχουν υιοθετήσει τις βιομετρικές μεθόδους ως εργαλείο για την αυθεντικοποίηση των χρηστών είναι σε θέση να προσφέρουν ένα εξαιρετικά ακριβές αποτέλεσμα και ένα ασφαλές περιβάλλον, όπου η ισχυρή επαλήθευση της ταυτοποίησης του χρήστη είναι απαραίτητη. Όπως τονίστηκε και σε προηγούμενο σημείο της ενότητας τα βιομετρικά δεδομένα είναι δύσκολο να παραποιηθούν και κατ΄ επέκταση να γίνουν εκμεταλλεύσιμα από κάποιο άτομο πέραν του ίδιου του χρήστη – κατόχου. Η μοναδικότητα αυτών τα καθιστά ως έναν ισχυρό αποτρεπτικό παράγοντα κατά της κλοπής της ταυτότητας αλλά και της μη εξουσιοδοτημένης πρόσβασης.

Επιπλέον οι χρήστες με αυτόν τον τρόπο επαλήθευσης της ταυτότητάς τους δε χρειάζεται να θυμούνται σύνθετους κωδικούς πρόσβασης ή να φέρουν κάποια φυσικά διακριτικά. Είναι σε θέση να χρησιμοποιήσουν απευθείας τα φυσικά τους βιομετρικά χαρακτηριστικά τα οποία είναι πάντα άμεσα διαθέσιμα και να κατορθώσουν να συνδεθούν στις διάφορες υπηρεσίες, συστήματα ή συσκευές που

εκείνοι επιθυμούν. Ο βαθμός ευκολίας της χρήσης τους είναι ικανή να προτρέψει τους χρήστες να υιοθετήσουν ολοένα και περισσότερο τέτοιου είδους τεχνολογίες όπως επίσης να σηματοδοτήσει μια εποχή βελτιωμένης συνολικής εμπειρίας για τον ίδιο τον χρήστη.

Τα υπάρχοντα συστήματα και οι υπάρχουσες τεχνολογίες μπορούν να ενσωματώσουν απρόσκοπτα τον βιομετρικό έλεγχο επιτρέποντας κατ΄ αυτόν τον τρόπο σε διάφορους οργανισμούς να ενισχύσουν σε μεγάλο βαθμό την ασφάλειά τους δίχως να κρίνονται απαραίτητες σημαντικές αλλαγές στην υποδομή. Τα βιομετρικά συστήματα είναι επεκτάσιμα και είναι σε θέση να χειριστούν μεγάλους όγκους χρηστών καθιστώντας τα κατάλληλα για τη χρήση τους σε ένα ευρύ φάσμα εφαρμογών στις οποίες συμπεριλαμβάνονται εκείνες σε επίπεδο επιχείρησης αλλά φυσικά και οι προσωπικές συσκευές.

Ωστόσο καμία μέθοδος δεν είναι απόλυτα ασφαλής. Όσο περισσότερο λαμβάνονται μέτρα για περαιτέρω προστασία τόσο πιο πιθανό είναι κάποιοι κακόβουλοι να στρέψουν την προσοχή τους σε αυτόν τον τομέα. Επομένως δεν είναι ακατόρθωτο να παρατηρηθούν πετυχημένες επιθέσεις οι οποίες στοχεύουν ακριβώς στις αδυναμίες της εκάστοτε μεθόδου. Όσο πιο ευρέως διαδεδομένη και ασφαλής είναι μια μέθοδος ταυτοποίησης τόσο πιο πιθανό είναι να αυξηθεί το ενδιαφέρον και οι προσπάθειες εύρεσης των τρωτών σημείων της.

Μπορεί ο βιομετρικός έλεγχος να προσφέρει πολλά πλεονεκτήματα αλλά πρέπει να τονιστεί ότι κρύβει και κινδύνους. Παρακάτω αποτυπώνονται οι κίνδυνοι που προκύπτουν.

- Προβλήματα απορρήτου
- Αμετάβλητα βιομετρικά
- Κόστος και η υλοποίησή τους
- Τα ποσοστά ψεύδους αποδοχής και ψεύδους απόρριψης
- Προβλήματα υγείας και σωματικά
- Η φύση του ανθρώπου

Προβλήματα απορρήτου

Ο βιομετρικός έλεγχος ταυτότητας εγείρει ανησυχίες που σχετίζονται με το απόρρητο και την ασφάλεια των προσωπικών δεδομένων. Η συλλογή αυτών των δεδομένων καθώς επίσης και η αποθήκευση των βιομετρικών πληροφοριών μπορεί να θεωρηθεί ως ένα είδος « εισβολής » στο απόρρητο ενός ατόμου. Παράλληλα υπάρχει ο κίνδυνος της παραβίασης αλλά και της κακής χρήσης των δεδομένων.

Ως δεδομένα προσωπικού χαρακτήρα θεωρούνται οποιεσδήποτε πληροφορίες που αναφέρονται στο υποκείμενο των δεδομένων (δηλαδή στον κάτοχό τους). Σε αυτά τα δεδομένα δε συγκαταλέγονται τα στατιστικής φύσεως συγκεντρωτικά στοιχεία από τα οποία δε μπορούν πλέον να προσδιοριστούν τα υποκείμενα των δεδομένων. Τα δεδομένα προσωπικού χαρακτήρα δε θα πρέπει να ταυτίζονται με τα ευαίσθητα δεδομένα που αφορούν στη φυλετική ή εθνική προέλευση, στα πολιτικά φρονήματα, στις θρησκευτικές ή φιλοσοφικές πεποιθήσεις , στη συμμετοχή

σε συνδικαλιστική οργάνωση, στην υγεία, στην κοινωνική πρόνοια , την ερωτική ζωή, στα σχετικά με ποινικές διώξεις ή καταδίκες, καθώς και στη συμμετοχή σε συναφείς με τα ανωτέρω ενώσεις προσώπων.

Τα βιομετρικά δεδομένα ορίζονται από το άρθρο 4, παράγραφος 14, ως δεδομένα προσωπικού χαρακτήρα. Τα χαρακτηριστικά αυτά προκύπτουν από την ειδική τεχνική επεξεργασία συνδεόμενη με βιολογικά, χαρακτηριστικά συμπεριφοράς ή φυσικά χαρακτηριστικά φυσικού προσώπου και τα οποία επιτρέπουν ή επιβεβαιώνουν την αδιαμφισβήτητη ταυτότητα του εν λόγω φυσικού προσώπου όπως είναι τα δακτυλοσκοπικά δεδομένα (δακτυλικά αποτυπώματα), οι εικόνες του προσώπου κλπ.

Επομένως οι οργανισμοί που θα ορίσουν ως μέσο ταυτοποίησης τους βιομετρικούς ελέγχους θα πρέπει να έχουν λάβει υπόψιν τις εκάστοτε κυρώσεις που προκύπτουν με την παραβίαση αυτών των βιομετρικών πληροφοριών. Ακριβώς για αυτόν τον λόγο οι οργανισμοί θα πρέπει να εφαρμόζουν αυστηρά μέτρα ασφαλείας και να συμμορφώνονται αντίστοιχα με τους αυστηρούς κανονισμούς προστασίας των βιομετρικών δεδομένων.

Αμετάβλητα βιομετρικά

Έγινε πολλάκις αναφορά στη μοναδικότητα των βιομετρικών χαρακτηριστικών και τα οφέλη που προσφέρουν. Ωστόσο το ότι είναι αμετάβλητα προξενεί κινδύνους. Αντίθετα με τους παραδοσιακούς κωδικούς πρόσβασης, τα βιομετρικά χαρακτηριστικά δεν έχουν τη δυνατότητα να αλλάξουν αφού συνοδεύουν τον ίδιο τον άνθρωπο καθ' όλη τη διάρκεια της ζωής του. Άρα στην περίπτωση που παραβιαστούν ή κλαπούν αυτές οι πληροφορίες δεν είναι εύκολο να ενημερωθούν ξανά. Τίθεται συνεπώς μια πρόκληση, ζωτικής σημασίας καθώς τα παραβιασμένα πλέον δεδομένα δε μπορούν να «επαναφερθούν» όπως γίνεται στην περίπτωση των συνθηματικών αφήνοντας έτσι τα άτομα ευάλωτα σε μη εξουσιοδοτημένη πρόσβαση.

Κόστος και υλοποίηση

Αντίθετα με τον πλέον διαδεδομένο τρόπο επαλήθευσης της ταυτότητας , μια εφαρμογή βιομετρικού συστήματος μπορεί να είναι ιδιαίτερα δαπανηρή. Ειδικότερα αν ένας οργανισμός έχει ήδη κάποια υπάρχουσα υποδομή θα πρέπει να αναβαθμιστεί ώστε να συμπεριληφθούν οι νέες δυνατότητες ή ακόμη και να αντικατασταθεί. Πέρα από αυτό το πρόβλημα το κόστος περιλαμβάνει επίσης και την απόκτηση των κατάλληλων εργαλείων για την απόκτηση των συσκευών που θα συλλέγουν τα βιομετρικά δεδομένα.

Άλλο ένα βασικό πρόβλημα του βιομετρικού ελέγχου που αυξάνει το κόστος είναι η απαίτηση εξειδικευμένου λογισμικού ή υλικού υποστήριξης. Γίνεται αντιληπτό ότι για μια τέτοια υποδομή χρειάζεται μια ομάδα από εκπαιδευμένους developers που θα κληθούν να δημιουργήσουν αυτή την εφαρμογή αλλά κυρίως τη διατήρηση της ασφάλειας και της απόδοσης του συστήματος. Σίγουρα αυτή η ομάδα θα πρέπει να είναι σε θέση να υποστηρίζει άμεσα την υποδομή όταν προκύπτουν διάφορα

προβλήματα. Επιπλέον είναι αρκετά δύσκολο να αντέξει οικονομικά τις απαιτήσεις αυτές ένας μικρός οργανισμός.

Ποσοστά ψεύδους αποδοχής και ψευδούς απόρριψης

Τα ποσοστά ψευδούς αποδοχής και ψευδούς απόρριψης είναι σημαντικές μετρήσεις οι οποίες υπολογίζουν την ακρίβεια και την αξιοπιστία των βιομετρικών συστημάτων που ελέγχουν την ταυτοποίηση. Το ποσοστό ψευδούς αποδοχής αφορά την εσφαλμένη αποδοχή και αντίστοιχα το ποσοστό ψευδούς απόρριψης την εσφαλμένη αδυναμία αναγνώρισης. Πιο συγκεκριμένα η εσφαλμένη αποδοχή παρουσιάζεται όταν το ίδιο το σύστημα προσδιορίζει το δείγμα που λαμβάνει ως εξουσιοδοτημένο χρήστη ενώ στην πραγματικότητα ανήκει σε έναν μη εξουσιοδοτημένο χρήστη.

Ακριβώς το αντίθετο συμβαίνει στην περίπτωση της ψευδούς απόρριψης. Η ψευδής απόρριψη παρατηρείται όταν το σύστημα αυτή τη φορά αποτυγχάνει να αναγνωρίσει κάποιον εξουσιοδοτημένο χρήστη βάσει του δείγματος που έλαβε και ουσιαστικά του απαγορεύει την πρόσβαση στην εκάστοτε υπηρεσία. Τόσο η εσφαλμένη αποδοχή όσο και η ψευδής απόρριψη είναι δύο σενάρια πολύ πιθανά τα οποία είναι ικανά να επιφέρουν επιπτώσεις στην ασφάλεια και την εμπειρία του χρήστη.

Οι λόγοι για τους οποίους μπορεί να συμβεί κάποιο από τα δύο σενάρια ποικίλλουν. Για παράδειγμα οι διάφορες περιβαλλοντικές συνθήκες που μπορεί να επικρατούν όπως είναι ο φωτισμός, ο θόρυβος ή η ποιότητα του ίδιου του βιομετρικού δείγματος μπορούν να επηρεάσουν την ακρίβεια του συστήματος. Έστω ότι οι συνθήκες φωτισμού δεν είναι καλές και η βιομετρική μέθοδος απαιτεί τις καλύτερες συνθήκες για την ταυτοποίηση του χρήστη όπως συμβαίνει στην αναγνώριση προσώπου , κατά την σάρωση του προσώπου , το αποτέλεσμα που θα παραχθεί μπορεί να οδηγήσει σε υψηλότερα ποσοστά ψευδούς αποδοχής. Το σύστημα δηλαδή μπορεί να ταιριάξει εσφαλμένα το πρόσωπο κάποιου με κάποιου άλλου.

Η πιθανότητα η ποιότητα του δείγματος να μην είναι η κατάλληλη είναι επίσης αυξημένη και κατ΄ επέκταση πάλι επηρεάζει την ακρίβεια του συστήματος. Αυτό μπορεί να συμβεί λόγω της υγιεινής αλλά και λόγω της κατάστασης των εργαλείων που χρησιμοποιούνται για τη συλλογή του δείγματος. Για παράδειγμα στην περίπτωση των δακτυλικών αποτυπωμάτων, η κατάσταση των δαχτύλων παίζει σημαντικό ρόλο. Αν τα δάχτυλα είναι βρώμικα είναι πολύ πιθανό να αλλοιωθεί το δείγμα. Αντίστοιχα όταν ο αισθητήρας (το βασικό εργαλείο για τη συλλογή των αποτυπωμάτων) έχει βρωμιά ή ακόμη και υγρασία τα ποσοστά ψευδούς απόρριψης αυξάνονται, αφού το σύστημα προσπαθεί να ταιριάξει με απόλυτη ακρίβεια το δακτυλικό αποτύπωμα.

Προβλήματα υγείας και σωματικά

Γενικά τα προβλήματα υγείας όσο και τα σωματικά προβλήματα επηρεάζουν άμεσα το δείγμα που συλλέγεται από κάθε είδους βιομετρικό σύστημα. Ορισμένες σωματικές καταστάσεις ή κάποιοι τραυματισμοί έγκειται να δημιουργήσουν προκλήσεις στα άτομα που χρησιμοποιούν ως μέσο ταυτοποίησης κάποια βιομετρική μέθοδο. Θα ακολουθήσουν πιο συγκεκριμένα παραδείγματα που αναφέρονται στις τεχνικές που αναλύθηκαν στο κεφάλαιο αυτό (αναγνώριση φωνής, αναγνώριση δακτυλικών αποτυπωμάτων και αναγνώριση φλεβών παλάμης)

Τα συστήματα που αναγνωρίζουν τη φωνή αναλύουν τα φωνητικά χαρακτηριστικά του εκάστοτε ατόμου για να επιτευχθεί η επαλήθευση της ταυτότητάς του. Κάποιες φυσικές συνθήκες ή εξωτερικοί παράγοντες επηρεάζουν άρρηκτα την ακρίβεια του φωνητικού δείγματος και φυσικά την αναγνώριση. Αν δηλαδή ένα άτομο αντιμετωπίζει δυσκολίες ή κάποια διαταραχή στην ομιλία ή τις φωνητικές χορδές είναι σίγουρο ότι θα βρει εμπόδια κατά την αναπαραγωγή των φωνητικών μοτίβων που έχει επιλέξει.

Αν μάλιστα το άτομο παρείχε ένα φυσιολογικό δείγμα ως το πρότυπο (το εγγεγραμμένο στη βάση δείγμα που θα χρησιμοποιηθεί ως πρότυπο για την αντιστοίχισή του με το δείγμα που συλλέγεται εκείνη την ώρα) και μετά λόγω ποικίλλων αιτιών διαγνωστεί με κάποια διαταραχή είναι δεδομένο ότι η πρόσβαση στις υπηρεσίες του θα απαγορεύεται. Το ίδιο ακριβώς θα συμβεί και στις περιπτώσεις που το άτομο έχει ένα απλό κοινό κρυολόγημα με αποτέλεσμα την αλλοίωση της χροιάς και της ποιότητας της φωνής του.

Άτομα που έχουν υποβληθεί σε χειρουργικές επεμβάσεις ή κάποιου είδους ιατρικές θεραπείες μπορεί να αντιμετωπίσουν το ίδιο πρόβλημα καθώς οι πιθανότητες της μη αποτελεσματικής αναγνώρισης της φωνής αυξάνονται. Η μη αποτελεσματική αναγνώριση της φωνής προκύπτει από το ενδεχόμενο κατά το χειρουργείο ή τη θεραπεία να έχουν επηρεαστεί τα φωνητικά χαρακτηριστικά του ατόμου που επιδιώκει να επαληθεύσει την ταυτότητά του με σκοπό την πρόσβαση σε κάποια εφαρμογή ή υπηρεσία.

Αντιστοίχως υπάρχουν και αυτές οι περιπτώσεις και κατά τη διαδικασία αναγνώρισης των δακτυλικών αποτυπωμάτων, μιας τεχνικής πιο διαδεδομένης από τις άλλες βιομετρικές. Και σε αυτή την μέθοδο η σωματική κατάσταση και οι τραυματισμοί δημιουργούν προβλήματα σε όσους επιχειρούν να ταυτοποιηθούν. Στην περίπτωση που τα άτομα που χρησιμοποιούν αυτή την τεχνική φέρουν φθαρμένα ή κατεστραμμένα δακτυλικά αποτυπώματα (μπορεί να συμπεριληφθεί και ο ακρωτηριασμός) είτε λόγω ατυχήματος ή λόγω του επαγγέλματος (για παράδειγμα η χειρωνακτική εργασία ή οι μουσικοί) είναι πιθανό οι σαρώσεις των δακτυλικών αποτυπωμάτων τους να μην είναι ακριβείς.

Ακόμη και οι δερματικές παθήσεις όπως είναι η ψωρίαση ή το έκζεμα μπορεί να εμφανίσουν ασυνέπειες ως προς το αποτέλεσμα της σάρωσης λόγω των αλλαγών που έχει υποστεί η υφή του δέρματος στα σημεία αυτά. Επίσης το δέρμα των ανθρώπων όσο μεγαλώνουν αλλοιώνεται. Άρα και οι ηλικιωμένοι θα είναι δυσκολότερο να λάβουν υψηλής ποιότητας σαρώσεις των δακτυλικών τους

αποτυπωμάτων. Το ίδιο συμβαίνει και στις περιπτώσεις του ξηρού δέρματος, με αποτέλεσμα τα υψηλότερα ποσοστά ψευδούς απόρριψης.

Όσον αφορά την τεχνική του ελέγχου των φλεβών της παλάμης τα προβλήματα που προκύπτουν είναι ποικίλα. Παρόλο που θεωρείται βιομετρική μέθοδος υψηλής ακριβείας κάποιοι φυσικοί περιορισμοί είναι ικανοί να μειώσουν την αποτελεσματικότητά της αισθητά. Η τεχνική αυτή χρησιμοποιεί υπέρυθρο φως για να συλλάβει και στη συνέχεια να αναλύσει τα μοναδικά σχέδια που σχηματίζουν οι φλέβες της παλάμης. Όταν όμως το άτομο που επιδιώκει την αυθεντικοποίησή του έχει κακή κυκλοφορία του αίματος ή αγγειακές παθήσεις όπως το φαινόμενο Raynaud[2] είναι επίσης πιθανό και πάλι να προκύψουν δυσκολίες κατά τη λήψη του δείγματος από τις σαρώσεις των φλεβών λόγω της ανεπαρκούς ροής του αίματος.

Τα άτομα επίσης με σοβαρούς τραυματισμούς στα χέρια , ή με εγκαύματα ή και ακρωτηριασμούς που επηρεάζουν προφανώς την περιοχή της παλάμης καθιστούν τον έλεγχο ταυτοποίησης μέσω των φλεβών ανούσιο. Οι φυσικοί αυτοί περιορισμοί οδηγούν και πάλι, όπως αναφέρθηκε και στις άλλες βιομετρικές μεθόδους προγενέστερα, σε υψηλότερα ποσοστά ψευδούς απόρριψης αφού η διαδικασία της συλλογής και της αντιστοιχίας των μοτίβων είναι είτε δύσκολη είτε αδύνατη.

Η φύση του ανθρώπου (έλλειψη παιδείας, καχυποψία)

Πέρα από τους κινδύνους που αναφέρθηκαν προηγουμένως η ίδια η φύση του ανθρώπου αποτελεί ένα πρόβλημα έναντι των βιομετρικών μεθόδων γενικά. Από τη φύση του ο άνθρωπος είναι επιφυλακτικός, δύσπιστος και καχύποπτος απέναντι στις καινούριες τεχνικές είτε αυτές έχουν να κάνουν με τον τομέα της ασφάλειας και την ταυτοποίηση των χρηστών είτε σχετίζονται με τον τομέα της υγείας είτε με τον τομέα της εκπαίδευσης κλπ. Οι άνθρωποι είναι ιδιαίτερα επιφυλακτικοί με τις τεχνικές αυτές καθώς μια επιτυχημένη επίθεση σημαίνει αυτομάτως ότι τα προσωπικά τους δεδομένα θα διαρρεύσουν.

Οι βιομετρικές μέθοδοι δεν είναι τόσο διαδεδομένες όσο τα συνθηματικά πρόσβασης. Επομένως πολλοί είναι εκείνοι που θα βασιστούν στα σχόλια και τις κριτικές των συνανθρώπων τους που έχουν τη δυνατότητα να αλληλεπιδρούν με τέτοιου είδους υπηρεσίες προκειμένου να σχηματίσουν τη δική τους άποψη. Αυτό συμβαίνει κυρίως στις μεγαλύτερες ηλικίες που δεν έχουν αρκετή τριβή με τις ταχύτατες εξελίξεις της τεχνολογίας. Σύμφωνα με έρευνα θεωρείται ότι η φήμη μιας μεθόδου ταυτοποίησης παίζει καθοριστικό ρόλο στην επιλογή κάποιου μέσου επαλήθευσης.

Επίσης σημαντικό παράγοντα για να ξεκινήσουν να χρησιμοποιούν έστω τις ήδη υπάρχουσες εφαρμογές που προσφέρουν βιομετρικό έλεγχο (όπως για παράδειγμα το δακτυλικό αποτύπωμα για το κλείδωμα του κινητού) είναι και η κατανόηση της λειτουργίας της όλης διαδικασίας. Όσο πιο δυσνόητη και δύσκολη είναι στο χειρισμό,

[2] Το σύνδρομο Raynaud είναι ένα φαινόμενο της επεισοδιακής αναστρέψιμης ισχαιμίας και μεταβολής του χρώματος των δαχτύλων και των άνω (μύτη , πτερύγια αυτιών) ή και κάτω άκρων. Η κυκλοφορία του αίματος ουσιαστικά διακόπτεται ξαφνικά. Το φαινόμενα οφείλεται σε σπάσιμο των μικρών αρτηριών στα άκρα αυτά.

τόσο περισσότερο θα επιδεινώσει τη σχέση του χρήστη με αυτήν. Προφανώς ένα άτομο δε θα επιλέξει να υιοθετήσει μια τέτοια προσέγγιση εάν η συνολική του εμπειρία είναι υποδεέστερη από εκείνη των συνθηματικών, που ουσιαστικά τόσα χρόνια γνωρίζει και έχει συνδέσει με την καθημερινότητά του.

Επιπροσθέτως υπάρχει η πιθανότητα να μην είναι αρεστή σε πολλά άτομα η αποχή της ενεργής τους συμμετοχής στην ίδια την τεχνική επαλήθευσης της ταυτότητας του χρήστη. Για παράδειγμα στην μέθοδο ταυτοποίησης ενός ατόμου με τη χρήση συνθηματικών, ο χρήστης είναι αυτός που την πρώτη φορά θα καθορίσει ποιο είναι αυτό το συνθηματικό, σύμφωνα πάντα με τις προδιαγραφές που έχει ορίσει προηγουμένως η υπηρεσία, και θα έχει τη δυνατότητα να τον αλλάζει όπως εκείνος επιθυμεί. Αντίθετα στις βιομετρικές μεθόδους ο χρήστης βασίζεται σε αισθητήρες οι οποίοι θεωρείται από ένα μέρος του πληθυσμού ότι δεν είναι απολύτως αξιόπιστοι καθ' όλη τη διάρκεια αλληλεπίδρασης ενός ατόμου με την υπηρεσία.

ΚΕΦΑΛΑΙΟ 3: ΠΡΟΣΘΕΤΟΙ ΜΗΧΑΝΙΣΜΟΙ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ ΧΩΡΙΣ ΣΥΝΘΗΜΑΤΙΚΑ

3.1 Εισαγωγή

Στο προηγούμενο κεφάλαιο αναλύθηκαν οι βιομετρικές μέθοδοι που μπορούν να χρησιμοποιηθούν για μια ταυτοποίηση χρήστη χωρίς συνθηματικά. Η μελέτη επικεντρώθηκε κυρίως σε 6 βιομετρικά χαρακτηριστικά. Τα χαρακτηριστικά αυτά περιλαμβάνουν τα δακτυλικά αποτυπώματα , την αναγνώριση φωνής, την αναγνώριση μοτίβου φλέβας παλάμης, την αναγνώριση ίριδας και την αναγνώριση προσώπου - χαμόγελου. Οι περισσότερες μέθοδοι αποτελούν τρόπους πιο οικείους στον άνθρωπο σε σχέση με την τεχνική του μοτίβου των φλεβών της παλάμης, καθώς υπάρχουν περισσότερες εφαρμογές που τις χρησιμοποιούν ή άτομα από τον κοινωνικό τους περίγυρο τους έχουν πληροφορήσει για αυτά.

Ωστόσο οι βιομετρικές μέθοδοι δεν είναι ο μοναδικός τρόπος για την ταυτοποίηση ενός χρήστη χωρίς την αναγκαιότητα εισαγωγής συνθηματικών. Η αυξανόμενη ζήτηση εφαρμογών – υπηρεσιών που δεν χρειάζονται κάποιον κωδικό για τον έλεγχο ταυτότητας έχει εδραιώσει κάποιες άλλες τεχνικές οι οποίες δεν εμπεριέχουν πληροφορίες άμεσα συνδεδεμένες με τα προσωπικά ή ευαίσθητα δεδομένα. Τέτοιου είδους τεχνικές, για τον έλεγχο επαλήθευσης του ατόμου, είναι οι token – based, τα μοτίβα πληκτρολόγησης και τα γραφικά συνθήματα καθώς επίσης και τεχνικές που εκμεταλλεύονται τις δυνατότητες των κατανεμημένων συστημάτων. Φυσικά υπάρχουν και τα blockchain για την αυθεντικοποίηση χρήστη και χρησιμοποιούνται σε ιδιαίτερα χρήσιμα συστήματα όπως είναι για παράδειγμα εκείνα των τραπεζών, ή των εκλογικών συστημάτων [39].

Φυσικά η διαδικασία της ταυτοποίησης ενός χρήστη δεν απαγορεύει τον συνδυασμό 2 ή περισσότερων τεχνικών επαλήθευσης ταυτότητας. Στο κεφάλαιο αυτό θα αναλυθούν ως ένα βαθμό οι token – based μέθοδοι, τα otp, οι βιομετρικές μέθοδοι βάσει χαρακτηριστικών συμπεριφοράς, τα γραφικά συνθηματικά και το OpenID ενώ παράλληλα θα αποτυπωθούν τα πλεονεκτήματα και τα μειονεκτήματα αυτών, καθώς επίσης και διάφορες εφαρμογές που ήδη κάνουν χρήση αυτών. Τέλος θα παρουσιαστούν τρόποι επίλυσης των προβλημάτων αλλά και των κινδύνων που κρύβει μια τέτοιου είδους προσέγγιση.

3.2 Usb token based

Ένα είδος ταυτοποίησης με τη χρήση διακριτικού (token) είναι ο έλεγχος βασιζόμενος σε μια συσκευή USB. Ό έλεγχος ταυτοποίησης που βασίζεται σε USB αποτελεί μια τεχνική ελέγχου για την οποία δε χρειάζεται η εισαγωγή κάποιου προσωπικού συνθηματικού. Η τεχνική αξιοποιεί τις δυνατότητες που προσδίδει μια συσκευή USB και η διαδικασία είναι ασφαλής καθώς επίσης και βολική. Ο έλεγχος περιλαμβάνει τη χρήση φυσικών κλειδιών ασφαλείας ή διακριτικών που συνδέονται

είτε με τον υπολογιστή είτε με κάποια φορητή συσκευή μέσω των θυρών USB. Το κλειδί ασφαλείας ή αλλιώς διακριτικό αναφέρεται σε μια φυσική συσκευή η οποία βρίσκεται στην κατοχή του χρήστη και την χρησιμοποιεί με σκοπό να επαληθεύσει την ταυτότητά του. Είναι σχεδιασμένα να διατηρούν την ασφαλή αποθήκευση κρυπτογραφικών κλειδιών ή άλλων ασφαλών στοιχείων τα οποία είναι απαραίτητα για τη διαδικασία του ελέγχου της ταυτότητας του χρήστη.

Τα κρυπτογραφικά κλειδιά αποτελούν θεμελιώδη στοιχεία των συστημάτων κρυπτογράφησης και ασφάλειας. Είναι μεγάλες σειρές από δεδομένα τα οποία χρησιμοποιούνται για ποικίλες κρυπτογραφικές λειτουργίες όπως για παράδειγμα η κρυπτογράφηση , η αποκρυπτογράφηση , η δημιουργία των ψηφιακών υπογραφών καθώς επίσης και η επαλήθευση της γνησιότητας των μηνυμάτων. Οι συσκευές USB διαθέτουν ενσωματωμένο υλικό και λογισμικό που είναι σε θέση να δημιουργήσει, να αποθηκεύσει και να εκτελέσει τις κρυπτογραφικές λειτουργίες χρησιμοποιώντας τα κλειδιά αυτά με ασφάλεια. Τα κλειδιά που είναι ήδη αποθηκευμένα στη συσκευή είναι μοναδικά για τον κάθε χρήστη και φυσικά διατηρούνται με ασφάλεια στη συσκευή, συνήθως σε έναν χώρο αποθήκευσης που βασίζεται στο υλικό ή σε ένα ασφαλές στοιχείο.

Το ασφαλές στοιχείο είναι στην πραγματικότητα ένα αποκλειστικό chip που περικλείεται μέσα στη συσκευή USB και παρέχει υψηλό επίπεδο ασφαλείας. Είναι κατασκευασμένο με τέτοιο τρόπο ώστε να είναι ανθεκτικό σε παραβιάσεις , μη εξουσιοδοτημένη πρόσβαση ή σε προσπάθειες εξαγωγής των κλειδιών. Ακόμη και στην περίπτωση που χαθεί ή κλαπεί η συσκευή είναι διασφαλισμένο ότι τα κλειδιά που έχουν αποθηκευτεί δε θα παραβιάζονται εύκολα.

Η διαδικασία της ταυτοποίησης βάσει ενός USB περιλαμβάνει κάποια βασικά βήματα. Αρχικά ο χρήστης χρειάζεται να έχει στην κατοχή του μια συσκευή USB. Στη συνέχεια θα χρειαστεί να καταχωρηθεί αυτή η συσκευή, να γίνει το αίτημα της ταυτοποίησης από τον χρήστη, να ανιχνευθεί η συσκευή, να ξεκινήσει η διαδικασία της επαλήθευσης και το τελευταίο στάδιο αφορά την χορήγηση της πρόσβασης στην υπηρεσία που είναι επιθυμητή.

Πιο συγκεκριμένα ο χρήστης θα πρέπει να έχει στην κατοχή του ένα κλειδί ασφαλείας USB ή διακριτικό που περιλαμβάνει τα κρυπτογραφικά κλειδιά που αναφέρθηκαν και προηγουμένως. Η συσκευή θα πρέπει να καταχωρηθεί στο σύστημα ή την υπηρεσία που ελέγχει την ταυτότητά του. Η διαδικασία της εγγραφής απαιτεί τη σύνδεση της συσκευής με τον λογαριασμό του χρήστη και τα απαραίτητα κρυπτογραφικά κλειδιά ή οι πληροφορίες αναγνώρισης αποθηκεύονται με ασφάλεια στη συσκευή.

Στη συνέχεια και όταν ο χρήστης επιδιώξει να επαληθεύσει την ταυτότητά του, εισάγει την καταχωρημένη πλέον συσκευή USB στην αντίστοιχη θύρα (θύρα USB) στον υπολογιστή του ή σε όποια άλλη συσκευή χρησιμοποιεί. Το σύστημα ελέγχου με τη σειρά του ανιχνεύει την παρουσία της συσκευής αυτής και ξεκινά η διαδικασία για την ταυτοποίηση του χρήστη. Το σύστημα επικοινωνεί με τη συσκευή , επαληθεύει την αυθεντικότητα της και εκτελεί κάποιες κρυπτογραφικές λειτουργίες για την επικύρωση της ταυτότητας του χρήστη. Ως κρυπτογραφικές λειτουργίες θεωρούνται μια πληθώρα από μαθηματικούς υπολογισμούς και αλγορίθμους που

εκτελούνται με τη χρήση των κρυπτογραφικών κλειδιών. Εάν η συσκευή επαληθευτεί επιτυχώς τότε ο χρήστης θεωρείται πλέον « ταυτοποιημένος » και έχει πρόσβαση στο σύστημα ή την εφαρμογή.

Μερικά βασικά παραδείγματα κρυπτογραφικών λειτουργιών που χρησιμοποιούνται στον έλεγχο που βασίζεται στο USB είναι η δημιουργία και η επαλήθευση της ψηφιακής ταυτότητας, ο έλεγχος ταυτότητας πρόκλησης – απόκρισης , οι λειτουργίες κατακερματισμού κρυπτογράφησης, η αλλαγή των κλειδιών κλπ. Στην πρώτη περίπτωση της ψηφιακής ταυτότητας γίνεται αναφορά στην ψηφιακή υπογραφή. Ουσιαστικά τα αποθηκευμένα στο USB κρυπτογραφικά κλειδιά χρησιμοποιούνται για τη δημιουργία των ψηφιακών υπογραφών.

Η ψηφιακή υπογραφή αποτελεί ένα μοναδικό αναγνωριστικό καθώς χρησιμοποιεί το ιδιωτικό κλειδί του χρήστη και επαληθεύεται χρησιμοποιώντας το αντίστοιχο δημόσιο κλειδί του. Στη συνέχεια η συσκευή USB δημιουργεί μια ψηφιακή υπογραφή που βασίζεται στο αίτημα ή την πρόκληση ελέγχου ταυτότητας. Αφού ολοκληρωθεί αυτή η διαδικασία το σύστημα ελέγχου ταυτότητας χρησιμοποιεί το δημόσιο κλειδί του χρήστη (το οποίο είναι αποθηκευμένο στην πλευρά του διακομιστή) για να επαληθεύσει την ψηφιακή υπογραφή και να διασφαλίσει την αυθεντικότητα του χρήστη.

Στον έλεγχο ταυτότητας πρόκλησης- απόκρισης το σύστημα παρουσιάζει μια πρόκληση ή μια τυχαία τιμή στη συσκευή USB. Με τη σειρά της η συσκευή αξιοποιεί το κρυπτογραφικό κλειδί προκειμένου να εκτελέσει κάποιους υπολογισμούς σχετικά με την πρόκληση και να παράξει μια απάντηση. Αυτή η απάντηση αποστέλλεται αμέσως μετά πίσω στο σύστημα ελέγχου, το οποίο επαληθεύει την ορθότητα της απάντησης βάσει του κρυπτογραφικού κλειδιού που είναι αποθηκευμένο όπως αναφέρθηκε στην πλευρά του διακομιστή. Με αυτή τη διαδικασία διασφαλίζεται ότι η συσκευή USB διαθέτει το σωστό κρυπτογραφικό κλειδί και συνακόλουθα συνδέεται με τον εξουσιοδοτημένο χρήστη.

Τα κρυπτογραφικά κλειδιά τα οποία είναι αποθηκευμένα στη συσκευή USB μπορούν να χρησιμοποιηθούν για ασφαλή πρωτόκολλα ανταλλαγής κλειδιών. Αυτά τα πρωτόκολλα επιτρέπουν στη συσκευή και στο ίδιο το σύστημα ελέγχου ταυτότητας να διαπραγματεύονται με ασφάλεια και ως εκ τούτου να δημιουργούν ένα κοινό μυστικό κλειδί για επακόλουθη ασφαλή επικοινωνία. Παράδειγμα τέτοιων πρωτοκόλλων ανταλλαγής περιλαμβάνουν τα Diffie – Hellman και Elliptic Curve Diffie – Hellman (ECDH).

Επίσης οι συσκευές USB δύνανται να εκτελούν κρυπτογραφικές λειτουργίες κατακερματισμού. Μια συνάρτηση κατακερματισμού λαμβάνει ως είσοδο ένα μήνυμα ή κάποια δεδομένα και παράγει μια έξοδο σταθερού μεγέθους που ονομάζεται τιμή κατακερματισμού. Η συσκευή είναι σε θέση να υπολογίσει έναν κατακερματισμό του αιτήματος ή της πρόκλησης ελέγχου ταυτότητας και το σύστημα στη συνέχεια μπορεί να το συγκρίνει με την αναμενόμενη τιμή κατακερματισμού με σκοπό να διασφαλίσει την ακεραιότητα των δεδομένων που ανταλλάσσονται κατά τη διαδικασία ελέγχου ταυτότητας.

Τα παραδείγματα που αναφέρθηκαν είναι κάποιες κρυπτογραφικές λειτουργίες που μπορούν να εκτελεστούν κατά τον έλεγχο ταυτοποίησης χωρίς τη χρήση κωδικού πρόσβασης που βασίζεται σε USB. Οι συγκεκριμένες λειτουργίες εξαρτώνται από τους κρυπτογραφικούς αλγόριθμους, τα πρωτόκολλα καθώς επίσης και τα πρότυπα που εφαρμόζονται από το σύστημα και τη συσκευή. Αυτές οι λειτουργίες συμβάλλουν συλλογικά στην ασφάλεια και την ακεραιότητα της διαδικασίας ελέγχου ταυτότητας.

Αυτή η μέθοδος ταυτοποίησης προσφέρει πολλά πλεονεκτήματα.

- Ισχυρή ασφάλεια: Σε σύγκριση με τα παραδοσιακά συνθηματικά που χρησιμοποιούνται για την ταυτοποίηση ενός χρήστη οι συσκευές USB παρέχουν ένα σαφώς υψηλότερο επίπεδο ασφαλείας. Οι κρυπτογραφικοί έλεγχοι και τα ασφαλή στοιχεία καθιστούν δύσκολη την παραβίαση της διαδικασίας ελέγχου της ταυτότητας από εισβολείς.
- Αντίσταση σε phishing: Ο έλεγχος ταυτοποίησης είναι ανθεκτικός σε επιθέσεις phishing καθώς η διαδικασία βασίζεται στη φυσική κατοχή της συσκευής USB. Ακόμη και στις περιπτώσεις που ο εκάστοτε χρήστης επισκεφτεί εν αγνοία του έναν κακόβουλο ιστότοπο ή λάβει κάποιο μήνυμα ηλεκτρονικού «ψαρέματος» (phishing), ο εισβολέας δε μπορεί να πραγματοποιήσει έλεγχο ταυτότητας χωρίς να κατέχει φυσικά τη συσκευή USB.
- Ευκολία από την πλευρά του χρήστη: Ο έλεγχος αυτός προσφέρει ευκολία εξαλείφοντας την ανάγκη ο χρήστης να θυμάται και να διαχειρίζεται πολύπλοκους κωδικούς πρόσβασης. Οι χρήστες το μόνο που χρειάζεται να κάνουν είναι απλώς να εισάγουν τη συσκευή USB στην αντίστοιχη θύρα για τον απρόσκοπτο έλεγχο ταυτότητας.

Ωστόσο όπως όλες οι μέθοδοι ταυτοποίησης έτσι και αυτή παρουσιάζει κάποια πιθανά μειονεκτήματα.

- Εξάρτηση υλικού (hardware): Για τον έλεγχο ταυτότητας χρησιμοποιώντας συσκευή USB είναι απαραίτητη η φυσική συσκευή υλικού (δηλαδή το διακριτικό USB). Σε περίπτωση που το διακριτικό αυτό χαθεί, δεν τοποθετηθεί σωστά ή κλαπεί ή ακόμη καταστραφεί και γενικά ή απώλεια αυτού ισοδυναμεί με δυσκολίες πρόσβασης σε συστήματα έως ότου αποκτηθεί το νέο διακριτικό αντικατάστασης. Η εξάρτηση αυτή από το υλικό αποτελεί μια υλικοτεχνική πρόκληση και μπορεί να απαιτεί εν γένει πρόσθετη υποστήριξη και διαχείριση.
- Κόστος και πολυπλοκότητα ανάπτυξης: Τα διακριτικά θα πρέπει σαφώς να αγοραστούν ενώ υπάρχει η πιθανότητα να χρειάζεται να δημιουργηθεί η απαραίτητη υποδομή και τα συστήματα για την υποστήριξη των USB. Το κόστος αυτό θα πλήξει κυρίως τους μικρότερους οργανισμούς με περιορισμένους πόρους. Πέρα από την ανάπτυξη των συστημάτων σημαντική είναι και η διαχείριση αυτών, κάτι που απαιτεί επίσης εξειδικευμένη τεχνογνωσία αυξάνοντας κατά αυτόν τον τρόπο την πολυπλοκότητα αλλά και το κόστος της υλοποίησης.
- Ευκολία και προσβασιμότητα: Στα πλεονεκτήματα της ταυτοποίησης των χρηστών με τη χρήση USB συσκευής αναφέρθηκε ότι προσφέρει ευκολία στο

χρήστη για τον τρόπο που ο ίδιος τη διαχειρίζεται. Ωστόσο, θα πρέπει να τονιστεί ότι η φυσική κατοχή και η εισαγωγή του διακριτικού στην αντίστοιχη θύρα μπορεί να προξενήσει εν γένει ταλαιπωρία. Αυτό συμβαίνει γιατί ο χρήστης μπορεί να έχει πρόσβαση σε διάφορα συστήματα από πολλές διαφορετικές συσκευές ή να είναι εν κινήσει. Προφανώς η συσκευή USB απαιτεί την κατάλληλη θύρα USB η οποία είναι διαθέσιμη σε συγκεκριμένες συσκευές.

- Συμβατότητα και μονοπώλιο προμηθευτή : Αυτή η μέθοδος ελέγχου ενδέχεται να απαιτεί συγκεκριμένους drivers ή στοιχεία λογισμικού για να λειτουργήσουν αποτελεσματικά. Μπορεί να προκύψουν προβλήματα συμβατότητας στην περίπτωση που το USB δεν υποστηρίζεται από όλα τα λειτουργικά συστήματα ή στην περίπτωση που οι drivers δεν είναι διαθέσιμοι σε ορισμένες συσκευές. Για τη λύση του διακριτικού USB οι οργανισμοί μπορεί να εξαρτώνται από έναν συγκεκριμένο προμηθευτή κάτι που ενδεχομένως μπορεί να οδηγήσει σε μονοπώλιο κάποιου προμηθευτή και κατ' επέκταση στην περιορισμένη ευελιξία στην υιοθέτηση εναλλακτικών λύσεων αλλά και τεχνολογιών.
- Μεμονωμένο «σημείο αποτυχίας »: Εάν το διακριτικό USB για κάποιον λόγο δυσλειτουργεί ή δεν είναι προσβάσιμο λόγω προβλημάτων του υλικού ο χρήστης πιθανότατα θα αποκλειστεί από τα συστήματα που επιδιώκει να επαληθεύσει την ταυτότητά του μέχρις ότου το διακριτικό επισκευαστεί ή αντικατασταθεί. Αυτό το μεμονωμένο « στοιχείο αποτυχίας » μπορεί να προξενήσει σημαντικές επιπτώσεις ειδικά εάν δεν είναι διαθέσιμες κάποιες άλλες εφεδρικές μέθοδοι ταυτοποίησης.

3.3 OTP

Οι κωδικοί μιας χρήσης OTP (One - time password) ενισχύουν την ασφάλεια και τον έλεγχο της ταυτοποίησης ενός ατόμου στον ψηφιακό κόσμο. Το OTP αποτελεί έναν μηχανισμό που χρησιμοποιείται για τον έλεγχο της ταυτότητας των χρηστών σε διάφορα συστήματα ή εφαρμογές χωρίς να βασίζεται στον παραδοσιακό τρόπο ταυτοποίησης χρήστη. Αυτή η τεχνική εμπεριέχει την ιδέα της παραγωγής ενός προσωρινού αλλά και μοναδικού κωδικού πρόσβασης ο οποίος ισχύει για μια και μόνο χρήση. Είναι μια κοινή και ευρέως διαδεδομένη μέθοδος και μπορούν να θεωρηθούν ένα είδος ταυτοποίησης χωρίς την εισαγωγή ενός προσωπικού συνθηματικού. Αποτελεί μέθοδο 2 επιπέδων (2FA- 2 factor authentication) υποκατηγορία των MFA (multi factor authentication) και λειτουργεί ως επιπλέον ασφάλεια [38].

Ο μοναδικός αυτός κωδικός πρόσβασης που δημιουργείται όπως αναφέρθηκε μπορεί να χρησιμοποιηθεί μόνο μια φορά εξ ου και το όνομα της μεθόδου «one - time». Κάθε φορά που ο χρήστης επιδιώκει να ταυτοποιηθεί δημιουργείται ένας καινούριος κωδικός πρόσβασης. Μόλις ο χρήστης ξεκινήσει τη διαδικασία επαλήθευσης για τη σύνδεση σε ένα σύστημα, το σύστημα αυτό δημιουργεί ένα OTP βάσει χρόνου ή συμβάντων και το αποστέλλει στον χρήστη μέσω ενός επιλεγμένου καναλιού επικοινωνίας όπως για παράδειγμα SMS , email ή κάποιας άλλης

αποκλειστικής εφαρμογής ελέγχου ταυτότητας (ο χρήστης έχει δηλώσει τα απαραίτητα στοιχεία προκειμένου η επικοινωνία να είναι επιτυχής). Στη συνέχεια ο χρήστης εισάγει το OTP που έχει λάβει στη διεπαφή σύνδεσης για να αποδείξει την ταυτότητά του. Μετά την επαλήθευση, παρέχεται η πρόσβαση στον χρήστη [18].

Οι περίπλοκες επιθέσεις που σχετίζονται με τα συνθηματικά συχνά στοχεύουν τα πιο αδύναμα διαπιστευτήρια από το πλήθος των διαθέσιμων καθώς είναι ευκολότερο να ανακτηθούν αυτά τα δεδομένα. Οι κωδικοί πρόσβασης μιας χρήσης αποτελεί μια φυσική βελτίωση σε σχέση με το παραδοσιακό σχήμα ταυτοποίησης : όνομα χρήστη/κωδικός ταυτοποίησης (username / password) που παρατηρείται σε διάφορες υπηρεσίες. Η εξωτερική ανάθεση (outsource) στο cloud των αναπτύξεων στα OTP από τους παρόχους διευκολύνει τους πελάτες στο cloud να ενεργοποιήσουν τους λογαριασμούς τους στον πάροχο OTP σε ποικίλες διαδικτυακές υπηρεσίες όταν φυσικά ο επαληθευτής OTP φιλοξενείται από αυτήν την cloud υπηρεσία.

Η χρήση του OTP αποτελεί μια συχνή επιλογή σε διάφορες πλατφόρμες είτε αυτές αφορούν τα μέσα κοινωνικής δικτύωσης είτε κάποιο προσωπικό εικονικό δίκτυο (VPN). Οι κωδικοί OTP είναι μια αλληλουχία αριθμών ενός συγκεκριμένου μήκους. Η αριθμητική αυτή αλληλουχία παράγεται κάθε φορά από έναν γεννήτορα και παύει να ισχύει όταν δεν πληρούνται κάποιες προϋποθέσεις. Τα OTP χρησιμοποιούνται ευρέως σε διαφορετικά περιβάλλοντα και έχουν τη δυνατότητα να δημιουργούνται χρησιμοποιώντας διάφορες μεθόδους. Υπάρχουν ποικίλοι τύποι OTP κάθε ένας από τους οποίους έχει τη δική του μέθοδο παραγωγής και τα δικά του χαρακτηριστικά. Οι πιο συνηθισμένοι είναι οι κωδικοί μιας χρήσης βάσει χρόνου, βάσει συμβάντων , βάσει SMS και κωδικοί μιας χρήσης βάσει email.

Ο κωδικός μιας χρήσης βάσει χρόνου (TOTP) δημιουργείται με βάση την τρέχουσα ώρα και ένα κοινό μυστικό κλειδί. Το κλειδί είναι γνωστό τόσο στον διακομιστή ελέγχου ταυτότητας όσο και στη συσκευή του χρήστη. Χρησιμοποιώντας έναν αλγόριθμο όπως (HMAC-SHA1 ή HMAC-SHA256), το σύστημα συνδυάζει το μυστικό κλειδί με την τρέχουσα ώρα και δημιουργεί έναν εξαψήφιο ή οκταψήφιο OTP. Το OTP αλλάζει βάσει ενός προκαθορισμένου χρονικού διαστήματος το οποίο μπορεί να είναι είτε 30 είτε 60 δευτερόλεπτα αναλόγως της υλοποίησης. Ένα παράδειγμα εφαρμογής που χρησιμοποιεί TOTP είναι το δημοφιλές « Google Authenticator». Η εφαρμογή αυτή επαληθεύει την ταυτότητα του χρήστη βασιζόμενη σε TOTP. Χρησιμοποιείται ευρέως από διαδικτυακές υπηρεσίες όπως η Google, το DropBox και το GitHub.

Ο κωδικός πρόσβασης μιας χρήσης βάσει συμβάντων (HOTP) αντίθετα με το TOTP δεν εξαρτάται από το χρόνο αλλά από ένα γεγονός. Κάθε φορά που κάποιος χρήστης συνδέεται, το σύστημα αυξάνει έναν μετρητή και τον συνδυάζει με το κοινό μυστικό κλειδί για να δημιουργήσει ένα OTP. Η τιμή του μετρητή συγχρονίζεται μεταξύ του διακομιστή και της συσκευής του χρήστη. Αυτός ο τύπος OTP είναι χρήσιμος σε σενάρια όπου ο χρήστης μπορεί να μην έχει αξιόπιστο ρολόι ή σύνδεση για συγχρονισμό ώρας. Ένα παράδειγμα ευρέως χρησιμοποιούμενης εφαρμογής που χρησιμοποιεί HOTP είναι το «Microsoft Authenticator». Η εφαρμογή αυτή υποστηρίζει τόσο τη μέθοδο TOTP αλλά και OTP που βασίζεται σε HOTP.

Χρησιμοποιείται από διάφορες υπηρεσίες όπως της Microsoft συμπεριλαμβανομένων των Office 365 και Azure.

Η μέθοδος OTP βάσει ενός SMS συμπεριλαμβάνει την αποστολή του OTP στην κινητή συσκευή του χρήστη μέσω SMS. Το σύστημα δημιουργεί ένα τυχαίο OTP και το παραδίδει στον καταχωρημένο αριθμό τηλεφώνου του χρήστη. Ο χρήστης στη συνέχεια εισάγει το OTP που έχει λάβει προηγουμένως στη διεπαφή σύνδεσης για έλεγχο ταυτότητας. Ωστόσο , αν και αυτή η μέθοδος είναι ευρέως χρησιμοποιούμενη θεωρείται λιγότερο ασφαλής από άλλες μεθόδους λόγω των πιθανών τρωτών σημείων στην υποδομή SMS.

Η μέθοδος που βασίζεται σε email είναι παρόμοια με εκείνη με το SMS. Το OTP και σε αυτή την περίπτωση αποστέλλεται στην καταχωρημένη διεύθυνση email του χρήστη. Το σύστημα δημιουργεί το OTP και το στέλνει μέσω email. Ο χρήστης ανακτά το OTP από τα εισερχόμενα email του και το χρησιμοποιεί στη συνέχεια για τον έλεγχο ταυτότητας. Βέβαια και αυτή η μέθοδος αν και βολική βασίζεται στην ασφάλεια του λογαριασμού του email του χρήστη το οποίο αποτελεί επίσης ένα τρωτό σημείο.

Τα OTP εφαρμόζονται και σε συστήματα ηλεκτρονικών τραπεζών αλλά και σε εικονικά ιδιωτικά δίκτυα (VPN) και συστήματα απομακρυσμένης πρόσβασης. Στα ηλεκτρονικά τραπεζικά συστήματα κατά τη σύνδεση ή κατά τη διάρκεια μιας συναλλαγής (θεωρείται ευαίσθητη συναλλαγή) από έναν τραπεζικό λογαριασμό σε έναν άλλον απαιτείται επιβεβαίωση ότι ο εντολέας είναι όντως ο δικαιούχος – κάτοχος του εκάστοτε λογαριασμού προκειμένου να ολοκληρωθεί η τραπεζική συναλλαγή. Μόλις ο χρήστης επιδιώξει την ταυτοποίησή του τότε παράγεται ένα OTP η λήψη του οποίου γίνεται μέσα από ένα κανάλι επικοινωνίας SMS ή από την αποκλειστική συσκευή ελέγχου ταυτότητας. Στη συνέχεια θα πρέπει να εισάγει αυτόν τον προσωρινό κωδικό στην περίπτωση που δεν αναγράφεται ήδη αυτόματα στην εφαρμογή και να τον αποστείλει προκειμένου να ολοκληρωθεί η διαδικασία της ταυτοποίησης. Στην περίπτωση που το OTP είναι ακόμη έγκυρο η ταυτοποίηση επιτυγχάνεται. Στα VPN και στα συστήματα απομακρυσμένης πρόσβασης συχνά χρησιμοποιούνται OTP για τη διασφάλιση μιας ασφαλούς σύνδεσης. Οι χρήστες θα πρέπει να παρέχουν κάθε φορά και με κάθε προσπάθεια σύνδεσής τους το έγκυρο OTP ώστε να ταυτοποιηθούν και να αποκτήσουν την πρόσβαση στο ιδιωτικό δίκτυο.

Συνοψίζοντας τα OTP προσφέρουν αρκετά πλεονεκτήματα καθώς η αλληλουχία τους είναι μοναδική και ισχύει για μια χρήση ή για περιορισμένο χρονικό διάστημα μειώνοντας έτσι τις πιθανότητες κλοπής του κωδικού πρόσβασης ή της μη εξουσιοδοτημένης πρόσβασης. Επιπλέον λόγω του σύντομου χρονικού διαστήματος που ένας από αυτούς τους κωδικούς ισχύει ένας εισβολέας ακόμη και αν καταφέρει να αναχαιτίσει το OTP θα του είναι άχρηστο μετά από το προκαθορισμένο αυτό χρονικό διάστημα. Οι χρήστες παράλληλα δε χρειάζεται να θυμούνται πολλούς κωδικούς πρόσβασης κάθε φορά που πραγματοποιείται έλεγχος ταυτότητας εξαλείφοντας με αυτόν τον τρόπο τον κίνδυνο επαναχρησιμοποίησης του κωδικού πρόσβασης σε όλες τις πλατφόρμες.

Η μέθοδος αυτή είναι εύκολη καθώς επίσης και βολική για τον εκάστοτε χρήστη. Είναι σημαντικό να τονιστεί πως περιπτώσεις όπου η σύνδεση στο διαδίκτυο δεν

είναι διαθέσιμη ή αξιόπιστη μπορούν να χρησιμοποιηθούν OTP που βασίζονται σε HOTP. Οι χρήστες έχουν τη δυνατότητα να χρησιμοποιήσουν OTP εκτός σύνδεσης αφού αυτός ο τύπος βασίζεται σε συμβάντα και όχι στον συγχρονισμό σε πραγματικό χρόνο με κάποιον διακομιστή. Αυτή η ευελιξία καθιστά το OTP κατάλληλο για διάφορα περιβάλλοντα.

Παρόλα τα πλεονεκτήματα η μέθοδος των OTP έχει και μειονεκτήματα. Η εξάρτηση από τις φορητές συσκευές γίνεται αντιληπτή. Πολλές από τις εφαρμογές OTP απαιτούν από τους χρήστες να διαθέτουν τουλάχιστον μία φορητή συσκευή ή κάποιο smartphone για να λαμβάνουν τα OTP. Αυτού του είδους η εξάρτηση μπορεί να αποτελέσει πρόκληση σε άτομα που είτε δεν έχουν πρόσβαση σε κινητές συσκευές είτε προτιμούν να μην τις χρησιμοποιούν για έλεγχο ταυτότητας. Επιπλέον τα OTP αποστέλλονται μέσω κάποιων καναλιών επικοινωνίας όπως SMS, email τα οποία είναι ευάλωτα σε υποκλοπές ή πλαστογράφηση. Αν οι εισβολείς αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε αυτά τα κανάλια επικοινωνίας ενδέχεται να υποκλέψουν το OTP θέτοντας σε κίνδυνο την ασφάλεια της διαδικασίας.

Άλλο ένα μειονέκτημα που αναφέρθηκε και στην ταυτοποίηση χρήστη βάσει διακριτικού USB είναι το μεμονωμένο «σημείο αποτυχίας». Αν η συσκευή που παράγει το OTP για τον οποιονδήποτε λόγο δεν είναι διαθέσιμη ο χρήστης θα αντιμετωπίσει δυσκολίες ως προς την πρόσβασή του στους λογαριασμούς του. Πολλοί χρήστες επίσης είναι πιθανό να βρουν την διαδικασία εγκατάστασης και διαμόρφωσης της εφαρμογής άβολη λόγω ελλιπούς εξοικείωσης με την τεχνολογία επηρεάζοντας τη συνολική εμπειρία του χρήστη. Τέλος η εξάρτηση από τους παρόχους αποτελεί ένα κρίσιμο κομμάτι στις μεθόδους OTP. Οι χρήστες βασίζονται στην αξιοπιστία και την ασφάλεια των παρόχων αυτών και επομένως σε σενάρια με παραβιάσεις στην υποδομή του παρόχου των υπηρεσιών ή σε τυχόν ευπάθειες τίθεται κίνδυνος της ασφάλειας του ελέγχου ταυτότητας OTP.

3.4 Έξυπνες κάρτες– smartcards

Οι έξυπνες κάρτες ακριβώς όπως και τα usb αποτελούν υλικά μέσα τα οποία μπορούν να χρησιμοποιηθούν για την αυθεντικοποίηση ενός χρήστη. Τι είναι οι έξυπνες κάρτες όμως και πως λειτουργούν; Οι έξυπνες κάρτες είναι στην ουσία μια πλαστική κάρτα στην οποία έχει ενσωματωθεί ένας μικροεπεξεργαστής. Είναι κατασκευασμένες με τέτοιο τρόπο ώστε να μπορούν να αποθηκεύσουν πληροφορίες και σε επόμενο στάδιο να τις επεξεργαστούν. Η δυνατότητα αυτή οφείλεται στο λειτουργικό σύστημα καθώς επίσης και στη μνήμη που διαθέτουν.

Βέβαια για να μπορέσει κάποιος να χρησιμοποιήσει μια τέτοια κάρτα θα πρέπει να διαθέτει κάποιο εργαλείο – συσκευή η οποία να είναι σε θέση να αναγνωρίσει την έξυπνη κάρτα. Η αναγνώριση μπορεί να γίνει με δύο τρόπους, είτε εξ επαφής με τη συσκευή είτε χωρίς επαφή (εκμεταλλευόμενη τις ραδιοσυχνότητες). Έστω λοιπόν ένας χρήστης που θέλει να ταυτοποιηθεί χρησιμοποιώντας την εξ επαφής δυνατότητα.

Για να χρησιμοποιήσει ένας χρήστης την έξυπνη κάρτα θα πρέπει πρώτα να την εισάγει στη μονάδα ανάγνωσης καρτών της συσκευής του ώστε η μονάδα να εγκαθιδρύσει επικοινωνία με την κάρτα. Την πρώτη φορά που ο χρήστης θα επιδιώξει να χρησιμοποιήσει την έξυπνη κάρτα (διαδικασία εγγραφής), ένας αξιόπιστος οργανισμός θα πρέπει να έχει εγκαταστήσει την έξυπνη κάρτα με τα απαραίτητα κρυπτογραφικά κλειδιά και πιστοποιητικά. Αφού λοιπόν έχει ολοκληρωθεί η παραπάνω διαδικασία ο χρήστης θα κληθεί να εισάγει έναν προσωπικό κωδικό (φυσικά όπως και στην περίπτωση των συνθηματικών ταυτοποίησης ο κωδικός θα πρέπει να είναι ισχυρός) που θα συνδέεται με την έξυπνη κάρτα.

Έχοντας πραγματοποιήσει και αυτό το βήμα, κάθε φορά που ο χρήστης θα επιδιώκει να συνδέεται στην υπηρεσία που επιθυμεί θα πληκτρολογεί αυτόν τον κωδικό. Ακολούθως ο μικροεπεξεργαστής της κάρτας δημιουργεί μια ψηφιακή υπογραφή βασιζόμενη στο ιδιωτικό κλειδί και στα δεδομένα που αφορούν την προσπάθεια της σύνδεσης (τα δεδομένα αυτά δημιουργούνται ώστε να αποδώσουν μοναδικότητα στην ψηφιακή υπογραφή, τα δεδομένα μπορεί να περιλαμβάνουν για παράδειγμα την ώρα που συμβαίνει η προσπάθεια, τον χρήστη κλπ).

Έπειτα αποστέλλονται τόσο η ψηφιακή υπογραφή, όσο και το δημόσιο κλειδί στον εξυπηρετητή, που σχετίζεται με την αυθεντικοποίηση των χρηστών, της υπηρεσίας. Ο εξυπηρετητής με τη σειρά του διεξάγει έλεγχο που σχετίζεται με την υπογραφή και το δημόσιο κλειδί. Εάν αυτές οι πληροφορίες επαληθευτούν τότε η υπηρεσία ταυτοποιεί τον χρήστη και επιτρέπει σε αυτόν την αλληλεπίδραση με την υπηρεσία χωρίς να χρειάζεται ξανά η επαλήθευσή του για ένα χρονικό περιθώριο που έχει καθοριστεί. Σε αντίθετη περίπτωση, δηλαδή εάν δεν επαληθευτούν τα στοιχεία, η προσπάθεια για μη εξουσιοδοτημένη πρόσβαση απορρίπτεται [19].

Τέτοιες κάρτες δεν είναι δύσκολο να τις προμηθευτεί κάποιος. Επί καθημερινής βάσης πολλοί είναι εκείνοι που χρησιμοποιούν έξυπνες κάρτες αλλά πολλές φορές όχι σαν μέσο ταυτοποίησης αυτό καθ΄ αυτό. Για παράδειγμα οι τραπεζικές κάρτες αποτελούν μια μορφή έξυπνης κάρτας. Πολλές εταιρείες έχουν υιοθετήσει τη μέθοδο των καρτών πρόσβασης προκειμένου να ελέγχονται τα άτομα που εισέρχονται στις εγκαταστάσεις τους. Οι έξυπνες κάρτες καθιστούν την επαλήθευση της ταυτότητας μια ιδιαίτερα φιλική διαδικασία ως προς τον χρήστη ενώ παράλληλα μπορούν να χρησιμοποιηθούν μαζί με άλλες μεθόδους και τεχνικές ταυτοποίησης.

Βασικά πλεονεκτήματα της μεθόδου αυτής είναι η ασφάλεια που παρέχεται, η φορητότητα, η κωδικοποίηση , η ταυτοποίηση δίχως την αναγκαιότητα του δικτύου αλλά και η συμβατότητα. Αναλυτικότερα, μια έξυπνη κάρτα έχει τη δυνατότητα να προσφέρει κάποια ισχυρά χαρακτηριστικά τα οποία είναι δύσκολο είτε να παραποιηθούν είτε να υποκλαπούν. Αυτό οφείλεται στη χρήση αλγορίθμων κωδικοποίησης για την προστασία των δεδομένων που έχουν αποθηκευτεί. , στον κρυπτογραφικό της χαρακτήρα (αποθήκευση κρυπτογραφικών κλειδιών, ψηφιακές υπογραφές κλπ), χρειάζεται πολλές φορές και ένας ακόμη βαθμός ταυτοποίησης.

Οι κάρτες είναι εύκολο να μεταφερθούν και παράλληλα επιτρέπουν και την αυθεντικοποίηση ενός χρήστη δίχως να βασίζονται σε μια διαρκή σύνδεση στο διαδίκτυο. Επίσης η ποικιλομορφία των τύπων τους μπορούν να χρησιμοποιηθούν

σε διάφορες εφαρμογές και για διάφορες ενέργειες. Επομένως γίνεται αντιληπτό πως διευκολύνουν ιδιαίτερα τους χρήστες. Ωστόσο, ούτε και οι έξυπνες κάρτες είναι μια αλάνθαστη μέθοδος. Όπως και όλες οι προηγούμενες τεχνικές είχαν τα πλεονεκτήματα και τα μειονεκτήματά τους έτσι και αυτή.

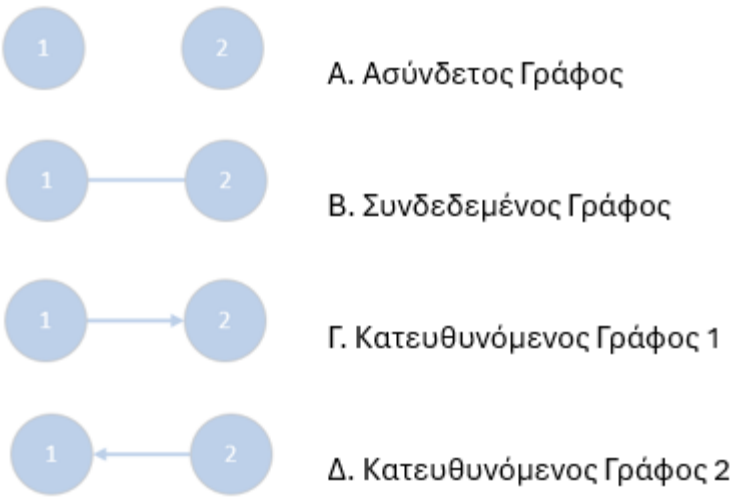
Ως ένα φυσικό μέσο και εργαλείο ταυτοποίησης είναι πολύ πιθανό να κλαπεί ή να χαθεί. Κατ΄ επέκταση αυτό μπορεί να οδηγήσει σε μη εξουσιοδοτημένη πρόσβαση. Επίσης είναι πιθανό ένας επιτιθέμενος να επιδιώξει να δημιουργήσει ακριβώς μια ίδια έξυπνη κάρτα με αυτή του χρήστη (είναι μια μέθοδος «κλωνοποίησης» της κάρτας). Επιπλέον και όπως αναφέρθηκε και στο «ΚΕΦΑΛΑΙΟ 1: PASSWORD ΚΑΙ ΕΥΠΑΘΕΙΕΣ» ένα τέτοιο εργαλείο θα πρέπει να ελέγχεται πριν τη χρήση του ή την πώληση του σε κάποιον ώστε να αποφευχθούν τυχόν κακόβουλες ενέργειες που σχετίζονται με τον τρόπο που έχει κατασκευαστεί η κάρτα.

3.5 Γραφικά συνθήματα - μοτίβα

Τα γραφικά συνθήματα αποτελούν ουσιαστικά έναν γράφο δηλαδή ένα γράφημα στο οποίο αποτυπώνονται διάφορα σημεία, που ονομάζομαι κόμβοι ή αλλιώς κορυφές, τα οποία μπορεί να είναι συνδεδεμένα μεταξύ τους ή και όχι με ακμές []. Ως γνωστόν έχοντας ένα πλήθος σημείων μπορούν να σχηματιστούν πολλοί διαφορετικοί γράφοι. Τα συνθήματα αυτά είναι μια μέθοδος που βρίσκει έφορο έδαφος κυρίως στις συσκευές που απαρτίζονται από οθόνες αφής. Στη σημερινή εποχή η βασικότερη συσκευή που κατά κόρον χρησιμοποιεί σε μεγάλο βαθμό ο άνθρωπος καθημερινά, και έχει οθόνη αφής, είναι τα κινητά τηλέφωνα.

Ακριβώς επειδή από ένα πλήθος σημείων μπορεί να σχεδιαστεί μια αντίστοιχη πληθώρα γράφων είναι μια πολύ καλή τεχνική που μπορεί να χρησιμοποιηθεί για να αντικαταστήσει τα παραδοσιακά συνθηματικά πρόσβασης. Έχοντας για παράδειγμα 2 κόμβους (σημεία) προκύπτουν 2 γράφοι (εάν εξαιρεθούν οι βρόχοι, δηλαδή οι περιπτώσεις που από τον έναν κόμβο η ακμή – γραμμή καταλήγει πάλι στον ίδιο κόμβο χωρίς να μεσολαβήσει άλλος) [Διάγραμμα 5, (Α,Β)].

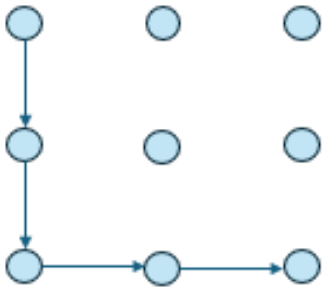
Φυσικά στη συγκεκριμένη περίπτωση όπου επιδιώκεται η σχεδίαση ενός μοτίβου ως μέσο επαλήθευσης της ταυτότητας του χρήστη, η πρώτη περίπτωση δεν είναι χρήσιμη. Επίσης θα πρέπει να τονιστεί ότι κατά το σχηματισμό ενός γραφικού συνθήματος στο κινητό ο γράφος που προκύπτει είναι κατευθυνόμενος και ακολουθείται μια συγκεκριμένη διαδρομή (ακολουθία κορυφών που συνδέονται διαδοχικά μεταξύ τους με ακμές και δεν απαγορεύεται φυσικά μια κορυφή να συναντάται παραπάνω από μία φορές) [20]. Επομένως για τη δεύτερη περίπτωση που αναφέρθηκε προκύπτουν 2 γράφοι και όχι ένας [Διάγραμμα 5, (Γ,Δ)].



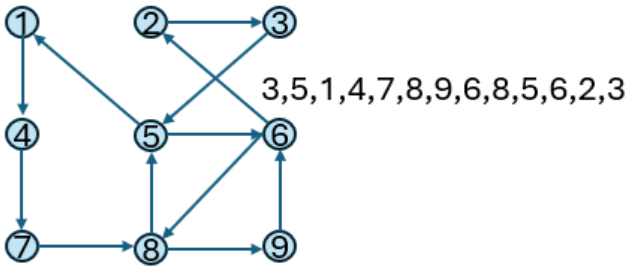
Διάγραμμα 5. Τύποι γράφων

Επομένως όταν ένας χρήστης προσπαθεί να συνδεθεί στο λογαριασμό του καλείται να σχεδιάσει το μοτίβο που έχει προεπιλέξει (κατά τη διαδικασία εγγραφής ή κάποια άλλη στιγμή που άλλαξε τον τρόπο με τον οποίο επιθυμεί να συνδέεται σε γραφικό συνθηματικό) στην οθόνη αφής. Στη συνέχεια αυτό το γράφημα που χαράχθηκε θα αξιολογηθεί κατά πόσον είναι όμοιο με εκείνο που έχει προεπιλεγεί. Στη περίπτωση που το γράφημα έχει ξεπεράσει το προκαθορισμένο όριο ομοιότητας τότε ο χρήστης ταυτοποιείται. Σε αντίθετη περίπτωση η πρόσβαση δεν επιτρέπεται και εμφανίζεται μήνυμα λανθασμένης προσπάθειας.

Παρόλο που ο σχεδιασμός μοτίβων αποτελεί μια τεχνική αυθεντικοποίησης, η οποία χρησιμοποιείται κυρίως στα κινητά τηλέφωνα (και άρα σχηματίζεται μόνο ένας γράφος κατά την είσοδο στη συσκευή) , δεν ενδείκνυται για την ευρεία χρήση τους ως μέσο ταυτοποίησης σε υπηρεσίες. Αυτό συμβαίνει γιατί όπως και με τα συνθηματικά πρόσβασης παρατηρούνται ανίσχυροι γράφοι, που συμπεριλαμβάνουν μόνο 2 κόμβους, είναι πολύ πιθανό να επαναχρησιμοποιηθούν και φυσικά απαιτούν την ύπαρξη οθόνης αφής.



Διάγραμμα 6. Αδύναμος γράφος ταυτοποίησης



Διάγραμμα 7. Ισχυρός γράφος ταυτοποίησης

Μπορεί τα μοτίβα αυτά να προσφέρουν μια φιλική αλληλεπίδραση μεταξύ της τεχνικής και του χρήστη αλλά ο περιορισμός που σχετίζεται με την οθόνη αφής παρεμποδίζει την ευκολία ταυτοποίησης σε υπηρεσίες όταν αυτή δεν είναι διαθέσιμη. Για παράδειγμα όταν κάποιος χρήστης επιδιώξει να αποκτήσει εξουσιοδοτημένη πρόσβαση σε μια υπηρεσία χρησιμοποιώντας τον υπολογιστή του είτε αυτός είναι φορητός είτε σταθερός και δεν έχει κάποιο εργαλείο για να συνδέσει τα σημεία τότε η όλη προσέγγιση παύει να είναι χρηστική.

Όπως έγινε αντιληπτό η παραπάνω μέθοδος αφορά το σχηματισμό μοτίβων με την έννοια που χρησιμοποιείται και κατά την είσοδο στα κινητά τηλέφωνα (για όσους έχουν επιλέξει αυτόν τον τρόπο εισόδου), δηλαδή χρησιμοποιώντας τα διαθέσιμα 9 σημεία που παρέχονται να σχηματιστεί ένας κατευθυνόμενος γράφος με τουλάχιστον 2 κόμβους συνδεδεμένους με 1 ακμή. Υπάρχει όμως και η περίπτωση ο γράφος να σχηματιστεί εξ ολοκλήρου από τον ίδιο τον χρήστη σαν ένα αποτέλεσμα «ζωγραφικής» κατά το οποίο δεν έχει προκαθορισμένα σημεία τα οποία μπορεί να χρησιμοποιήσει.

Πλέον γίνεται αναφορά στην αυθεντικοποίηση βάση ενός σχεδιασμένου από τον χρήστη μοτίβου το οποίο έχει σχεδιάσει με τη χρήση κάποιου εργαλείου όπως είναι για παράδειγμα η γραφίδα. Η τεχνική αφορά μια συμπεριφορική βιομετρική τεχνική η οποία επίσης παρόλο που μπορεί να θεωρηθεί φιλική ως προς τον χρήστη δεν ενδείκνυται για την επαλήθευση της ταυτότητας του χρήστη. Ο χρήστης αρχικά επιλέγει τι θέλει να σχεδιάσει και αφού δημιουργήσει το μοτίβο του το καταχωρεί. Παρόλο που το τελικό σχήμα μπορεί να είναι ένας απλώς κύκλος, το μοτίβο είναι μοναδικό αφού έχει επηρεαστεί άρρηκτα από την ψυχολογία του χρήστη την εκάστοτε στιγμή αλλά και από τις ίδιες του τις ικανότητες.

Άρα προκύπτει ένα μοναδικό μοτίβο σαν πρότυπο για τις μετέπειτα ταυτοποιήσεις. Ωστόσο, σε αντίθεση με τα βιομετρικά χαρακτηριστικά σε αυτήν την περίπτωση αξιολογείται ως είσοδος ναι μεν ένα μοτίβο που είναι μοναδικό αλλά όπως αναφέρθηκε και προηγουμένως είναι συνδεδεμένο και με την ψυχολογική κατάσταση του εκάστοτε χρήστη. Επομένως την επόμενη φορά που θα προσπαθήσει να σχεδιάσει το ίδιο ακριβώς μοτίβο είναι πολύ πιθανό να μην σχεδιάσει κάτι που να έχει την απαιτούμενη ομοιότητα με το πρότυπο και έτσι να μην αποκτήσει πρόσβαση στην επιθυμητή υπηρεσία. Φυσικά αυτή η προσέγγιση που καθορίζεται σε μεγάλο

βαθμό από την ψυχολογία του χρήστη την εκάστοτε στιγμή δε θα μπορούσε να υιοθετηθεί ως ένα μέσο ταυτοποίησης.

Επιπλέον σε αυτού του είδους την τεχνική παίζει ρόλο και ο παράγοντας περιβάλλον. Για παράδειγμα αν ένας χρήστης βρίσκεται μέσα σε ένα κινούμενο όχημα και προσπαθήσει να αποκτήσει πρόσβαση σε μια υπηρεσία σχεδιάζοντας το μοτίβο του, είναι δεδομένο ότι λόγω των κινήσεων του οχήματος το τελικό αποτέλεσμα να μην είναι το αναμενόμενο. Καθώς το σύνολο των γραμμών που έχουν χρησιμοποιηθεί θα πρέπει να είναι όσο το δυνατών πιο κοντά στο αρχικό σχεδιασμένο μοτίβο η οποιαδήποτε απόκλιση (που θα έχει ξεπεράσει βέβαια το καθορισμένο όριο) θα πρέπει να οδηγεί σε απόρριψη. Πολλές καθημερινές δραστηριότητες και συνήθειες του χρήστη δυσχεραίνουν ιδιαίτερα την όλη διαδικασία.

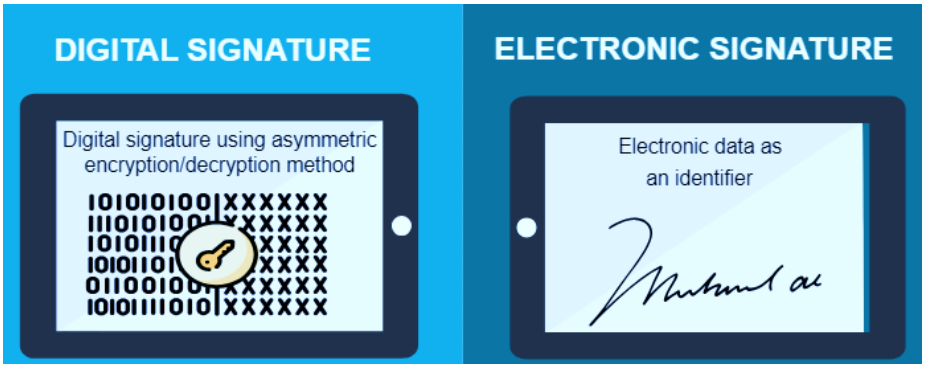
Η διαδικασία χρήσης μοτίβου όπως στη λειτουργία ξεκλειδώματος της οθόνης είτε η διαδικασία σχεδιασμού ενός μοτίβου πέραν από τα μειονεκτήματα που καταγράφηκαν πιο πάνω παρουσιάζουν επίσης και ένα ακόμη βασικό μειονέκτημα. Και στις δύο περιπτώσεις το τελικό αποτέλεσμα μπορεί εύκολα να καταγραφεί, να παραβιαστεί και τέλος να χρησιμοποιηθεί από τρίτους για την απόκτηση μη εξουσιοδοτημένης πρόσβασης. Άρα γίνεται σαφές ότι δεν αποτελούν μια προσέγγιση που θα επιλύσει τα προβλήματα που παρουσιάζει η παραδοσιακή μέθοδος των κωδικών πρόσβασης.

3.6 Χαρακτηριστικά συμπεριφοράς

Τα συμπεριφορικά χαρακτηριστικά που αφορούν τη συμπεριφορά που έχει ένα άτομο και μπορούν να μετρηθούν – υπολογιστούν ονομάζονται συμπεριφορικά βιομετρικά χαρακτηριστικά. Στο κεφάλαιο 2 παρουσιάστηκαν εκτενώς κάποια φυσικά βιομετρικά χαρακτηριστικά και έγινε μια πολύ σύντομη αναφορά στα συμπεριφορικά βιομετρικά χαρακτηριστικά. Στην παρούσα ενότητα θα μελετηθούν ο ρυθμός πληκτρολόγησης, η αλληλεπίδραση με το ποντίκι και ο γραφικός χαρακτήρας με τη μορφή υπογραφής. Ο σχεδιασμός ενός μοτίβου αναλύθηκε στην προηγούμενη ενότητα επομένως δε θα αναλυθεί και σε αυτή την ενότητα παρόλο που είναι συμπεριφορικό βιομετρικό χαρακτηριστικό.

Η μέθοδος που αφορά τον γραφικό χαρακτήρα με τη μορφή υπογραφής (signature dynamics) ουσιαστικά ελέγχει και αναλύει τον γραφικό χαρακτήρα ενός ατόμου όπως είναι δηλαδή η υπογραφή και η ταχύτητα με την οποία ολοκληρώνεται ο σχεδιασμός αυτός. Σε αυτό το σημείο θα πρέπει να διευκρινίσουμε πως με τον όρο υπογραφή δεν εννοείται η υπογραφή όπως είναι γνωστή και χρησιμοποιείται. Πιο συγκεκριμένα ως υπογραφή μπορεί να θεωρηθεί και μια απλή φράση. Επίσης η έννοια δε θα πρέπει να συγχέεται με αυτή της ηλεκτρονικής υπογραφής.

Ακολουθεί εικόνα αναπαράστασης μιας ψηφιακής και μιας ηλεκτρονικής υπογραφής [37].



Εικόνα 4. Απεικόνιση ψηφιακής και ηλεκτρονικής υπογραφής

Η βιομετρική επαλήθευση υπογραφής (BSV) μπορεί να χρησιμοποιηθεί τότε σε συστήματα στα οποία η σύνδεση στο διαδίκτυο είναι αναγκαία όσο και σε συστήματα τα οποία λειτουργούν και χωρίς σύνδεση (online, offline systems). Αυτή η δυνατότητα προσφέρει σαφώς ευκολία στους χρήστες καθώς μπορούν να βασιστούν σε αυτή την τεχνική άσχετα από την περιοχή που βρίσκονται (υπάρχουν ακόμη και σήμερα περιοχές στις οποίες το διαδύκτιο είναι περιορισμένο, αν όχι όλη τη μέρα, τις περισσότερες ώρες αυτής).

Για κάθε μία περίπτωση χρησιμοποιείται και μια διαφορετική προσέγγιση. Πιο συγκεκριμένα για τις online περιπτώσεις γίνεται χρήση προγραμματιζόμενων συστοιχιών πύλης (FPGAs – Field programmable gate arrays). Στα FPGAs έχουν επιλεγεί αλγόριθμοι αναγνώρισης ώστε να εντοπίσουν και να ανιχνεύσουν τα απαραίτητα μοτίβα ή διάφορες πληροφορίες των ψηφιακών υπογραφών. Καταγράφονται οι κινήσεις , ο θόρυβος και τα μοτίβα στα οποία οφείλεται η καθυστέρηση της διαδικασίας της επαλήθευσης της υπογραφής και κατ΄ επέκταση της της επαλήθευσης της ταυτότητας του χρήστη.

Στις online περιπτώσεις πέρα από τη μέθοδο των FPGAs μπορεί να χρησιμοποιηθεί επίσης και η μέθοδος VFPU (vector floating – pointing units). Αυτή η μέθοδος της κινητής υποδιαστολής έχει τη δυνατότητα να προσδιορίζει την τιμή των κινητών υποδιαστολών στις υπογραφές. Αυτή η διαδικασία επιτυγχάνεται κατόπιν υπολογισμού κάποιων δεδομένων αξιολόγησης που περιλαμβάνονται σε μια βάση και περιέχουν την ακριβή τιμή της κάθε υπογραφής. Αντίθετα στις περιπτώσεις εκτός σύνδεσης χρησιμοποιείται κυρίως το μοντέλο HMM (Hidden Markov Model)

3.7 OpenID

Το OpenID είναι ένα πρωτόκολλο αυθεντικοποίησης το οποίο επιτρέπει στους χρήστες να συνδέονται σε διάφορες υπηρεσίες ή ιστότοπους χρησιμοποιώντας ένα σύνολο διαπιστευτηρίων. Με αυτόν τον τρόπο ο χρήστης δεν χρειάζεται να διαχειρίζεται πληθώρα συνθηματικών και usernames. Παρακάτω ακολουθεί ο τρόπος με τον οποίο λειτουργεί το πρωτόκολλο αυτό.

Αρχικά ο χρήστης θα πρέπει να εγγραφεί σε έναν πάροχο OpenID όπως για παράδειγμα στα μέσα κοινωνικής δικτύωσης. Όταν λοιπόν ο χρήστης θα προσπαθήσει να συνδεθεί σε κάποια υπηρεσία – ιστοσελίδα που φυσικά υποστηρίζει την λειτουργία το OpenID τότε η υπηρεσία στέλνει αίτημα ταυτοποίησης στον επιλεγμένο πάροχο του χρήστη. Στη συνέχεια ο πάροχος ταυτοποιεί τον χρήστη και ζητά τη συγκατάθεσή του προκειμένου να μοιραστεί πληροφορίες όπως η ηλεκτρονική διεύθυνση ή το όνομά του με την υπηρεσία.

Αφού ολοκληρωθεί η παραπάνω διαδικασία και ο χρήστης έχει δώσει τη συγκατάθεσή του ο πάροχος σχηματίζει μια «απάντηση» που ταυτοποιεί τον χρήστη και περιλαμβάνει τις απαραίτητες πληροφορίες. Η «απάντηση» αυτή αποστέλλεται πίσω στην υπηρεσία και αν επαληθευτεί με επιτυχία ο χρήστης πλέον αποκτά πρόσβαση στην ιστοσελίδα. Η διαδικασία της επαλήθευσης περιλαμβάνει κάποια μέσα ασφαλείας όπως για παράδειγμα η κρυπτογραφημένη υπογραφή.

Κατά τη διάρκεια της επαλήθευσης της «απάντησης» είναι απαραίτητο να αποδειχθεί η αυθεντικότητά της. Πως γίνεται αυτό; Μέσω διάφορων μέσων ασφαλείας όπως είναι για παράδειγμα η κρυπτογραφημένη υπογραφή επιτυγχάνεται η απόδειξη γνησιότητας της «απάντησης». Πιο συγκεκριμένα, ο πάροχος OpenID θα υπογράψει την «απάντηση» που θα αποστείλει ώστε η υπηρεσία να έχει τη δυνατότητα να αναγνωρίσει την εγκυρότητα του παρόχου. Αν η επαλήθευση γίνει με επιτυχία τότε ο χρήστης αποκτά εξουσιοδοτημένη πρόσβαση στην υπηρεσία. Σε αντίθετη περίπτωση αποτρέπεται η είσοδος.

Τα πλεονεκτήματα αυτής της μεθόδου είναι πολλά καθώς μειώνει σε σημαντικό βαθμό την ανάγκη μεγάλου αριθμού συνθηματικών καιusernames. Με ένα συγκεκριμένο σύνολο διαπιστευτηρίων ένας χρήστης μπορεί να ταυτοποιείται και να συνδέεται σε ποικίλες υπηρεσίες- ιστοσελίδες. Φυσικά η μέθοδος είναι πιο ασφαλής αφού η απουσία συνθηματικών μειώνει σε τεράστιο βαθμό τους κινδύνους σε ζητήματα ασφαλείας απέναντι σε επιθέσεις (που σχετίζονται με αυτά).

Άλλο ένα σημαντικό προτέρημα είναι η δυνατότητα του χρήστη να αποφασίσει ο ίδιος ποιες και πόσες πληροφορίες θα χρησιμοποιηθούν κατά τη διάρκεια της ταυτοποίησης αλλά και η συμβατότητα - δια λειτουργικότητα. Με αυτόν τον τρόπο οι χρήστες έχουν τον απόλυτο έλεγχο της ψηφιακής τους ταυτότητας και μπορούν να το χρησιμοποιούν τόσο σε διαφορετικά λειτουργικά συστήματα όσο και σε πάρα πολλές εφαρμογές ή υπηρεσίες. Δεν είναι τυχαίο που ολοένα και περισσότερες ιστοσελίδες υιοθετούν αυτή την προσέγγιση για την αυθεντικοποίηση του χρήστη.

Εκτός όμως από τα πλεονεκτήματα υπάρχουν και μειονεκτήματα. Βασικό πρόβλημα είναι η εξάρτηση από το πάροχο OpenID. Αν η υπηρεσία που θέλει ο

χρήστης να συνδεθεί δεν υποστηρίζει αυτή τη μέθοδο προφανώς και δε θα μπορέσει να συνδεθεί. Αν υπάρξει κάποιο πρόβλημα με τον ίδιο τον πάροχο, όπως για παράδειγμα διακοπή λειτουργίας, η διαδικασία σαφώς και θα επηρεαστεί. Κίνδυνο αποτελεί φυσικά και η έκθεση των πληροφοριών – προσωπικών δεδομένων. Παρόλο που οι πληροφορίες που χρησιμοποιούνται για την ταυτοποίηση επιλέγονται μέχρι ένα βαθμό από τον χρήστη, δεν παύει αυτά να είναι προσωπικά δεδομένα.

Σημαντικό είναι να τονιστεί πως η συγκεκριμένη προσέγγιση απαιτεί επιπλέον εξειδικευμένες γνώσεις και ενέργειες για την ανάπτυξη του ελέγχου ταυτοποίησης. Συγκριτικά με την παραδοσιακή μέθοδο ταυτοποίησης, δηλαδή τα συνθηματικά, ο βαθμός πολυπλοκότητας είναι ιδιαίτερα αυξημένος. Αυτό το μειονέκτημα σε συνδυασμό με τις διαφορές που μπορεί να συναντήσει ένας προγραμματιστής κατά την προσαρμογή της διαδικασίας ελέγχου σε διάφορες πλατφόρμες που είναι πιθανό να χρησιμοποιηθούν δυσχεραίνει την κατάσταση.

Επειδή ακριβώς η μέθοδος έχει υιοθετηθεί από πολλές υπηρεσίες υπάρχει η πιθανότητα η να διαφέρει η διαδικασία από πλατφόρμα σε πλατφόρμα ή από υπηρεσία σε υπηρεσία και επομένως να παρατηρηθούν ασυνέπειες στην εμπειρία των χρηστών. Είναι ιδιαίτερα σημαντικό ο χρήστης καθ’ όλη τη διάρκεια που χρησιμοποιεί τη μέθοδο αλλά και κάθε φορά που θα τη χρησιμοποιεί η εμπειρία του χρήστη να είναι ίδια αν όχι καλύτερη.

ΚΕΦΑΛΑΙΟ 4: ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΜΗΧΑΝΙΣΜΟΥ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ ΜΕ OTP ΚΑΙ MPI

4.1 Εισαγωγή

Στην ενότητα αυτή θα γίνει η ανάλυση και θα παρουσιαστεί ο σχεδιασμός του disCodeV2 ως μέθοδος ταυτοποίησης. Στο στάδιο της ανάλυσης θα γίνει αναφορά στο disCode, τη βασική του ιδέα και τις διαφορές με το disCodeV2. Κατά το στάδιο του σχεδιασμού θα παρουσιαστούν διαγράμματα και πληροφορίες που σχετίζονται με τον τρόπο με τον οποίο θα κατασκευαστεί η βάση δεδομένων και οι διαδικασίες της εγγραφής και της σύνδεσης.

4.2 Ανάλυση του disCodeV2

Όπως έχει γίνει ήδη αντιληπτό χρησιμοποιώντας 2 μεθόδους αυθεντικοποίησης το επίπεδο ασφαλείας αυξάνεται εκθετικά. Σε αυτή την ενότητα θα μελετηθεί το σενάριο της χρήσης OTP και του MPI με το οποίο η συνολική διαδικασία χωρίζεται σε μικρότερες διεργασίες.

Βασικό σκεπτικό αυτής της μεθόδου είναι η ύπαρξη και η εκμετάλλευση κάποιου μοναδικού στοιχείου – χαρακτηριστικού που έχει ένας χρήστης ώστε να γίνει η επαλήθευσή του κατά τη διαδικασία της ταυτοποίησης. Όταν ένας χρήστης συνδέεται στο διαδίκτυο αποκτά κάποια στοιχεία τα οποία είναι μοναδικά. Για παράδειγμα η IP είναι μια μοναδική διεύθυνση που ανήκει σε ένα μόνο άτομο. Ωστόσο, η IP που χρησιμοποιεί ένας απλός άνθρωπος για προσωπική του περιήγηση στο διαδίκτυο δεν είναι στατική αλλά δυναμική.

Οι IPs μπορεί να είναι είτε δυναμικές είτε στατικές. Οι στατικές διευθύνσεις έχουν αποδοθεί σε συγκεκριμένα άτομα και κάθε φορά που συνδέονται στο διαδίκτυο έχουν την ίδια διεύθυνση. Αντίθετα οι δυναμικές διευθύνσεις δεν έχουν συγκεκριμένο κάτοχο. Αυτό σημαίνει ότι ένας χρήστης με δυναμική IP κάθε φορά που συνδέεται στο διαδίκτυο λαμβάνει μια διαφορετική IP. Επομένως οι διευθύνσεις αυτές δε μπορούν να χρησιμοποιηθούν ως το μοναδικό χαρακτηριστικό πάνω στο οποίο θα βασιστεί η μέθοδος ταυτοποίησης.

Προς το παρόν σε ένα άτομο, όταν συνδέεται στο διαδίκτυο, δεν αποδίδεται κάποιο μοναδικό χαρακτηριστικό το οποίο είναι ικανό να επαληθεύσει την ταυτότητα του.

Το disCodeV2 είναι μια μέθοδος αυθεντικοποίησης χρήστη η οποία βασίστηκε στο disCode [21]. Ωστόσο, έχουν γίνει μερικές σημαντικές προσθήκες οι οποίες στοχεύουν στη δημιουργία μιας μοναδικής ταυτότητας για τους χρήστες, στην προστασία των δεδομένων τους αλλά και στη δημιουργία διεπαφής για την καλύτερη εμπειρία τους κατά την αλληλεπίδραση με την υπηρεσία.

4.2.1 disCode και disCodeV2

Κρίνεται σημαντικό να γίνει αρχικά μια αναφορά στο disCode και τη λειτουργία του προκειμένου να γίνει κατανοητός ο τρόπος με τον οποίο λειτουργεί και το disCodeV2 και στη συνέχεια να παρουσιαστούν και οι διαφορές τους.

disCode

Το disCode όπως αναφέρθηκε έχει παρουσιαστεί ήδη ως μια εφαρμογή σε γλώσσα προγραμματισμού python χωρίς γραφικό περιβάλλον χρήστη αφού εκτελείται σε τερματικό παράθυρο (terminal). Τα 2 βασικότερα εργαλεία για την επίτευξη της ταυτοποίησης του χρήστη είναι η χρήση του MPI και το OTP.

Το MPI (Message Passing Interface) αποτελεί ένα πρότυπο προγραμματισμού που εκμεταλλεύεται τη δυνατότητα ανταλλαγής μηνυμάτων μεταξύ διάφορων διεργασιών σε εφαρμογές. Η ανταλλαγή των μηνυμάτων αυτών επιτρέπει την παράλληλη εκτέλεση εργασιών σε διαφορετικούς επεξεργαστές ή υπολογιστικούς κόμβους. Άρα οι διεργασίες μπορούν και γίνονται ταυτόχρονα. Φυσικά η επικοινωνία μεταξύ των διεργασιών είναι ασφαλής αλλά και αποτελεσματική. Για αυτό το λόγο το MPI θεωρείται ένα ιδιαίτερα ισχυρό εργαλείο για την ανάπτυξη παράλληλων και κατανεμημένων εφαρμογών.

Ως δεδομένα για την ταυτοποίηση του χρήστη χρησιμοποιήθηκε το usernameκαι οι πληροφορίες της μηχανής. Πιο συγκεκριμένα οι πληροφορίες της μηχανής αποτελούνται από τα εξής χαρακτηριστικά:

1. **Τύπος Μηχανής ή Αρχιτεκτονική:** Ουσιαστικά γίνεται αναφορά στην αρχιτεκτονική της μηχανής. Για παράδειγμα μια πιθανή τιμή είναι το «AMD64» το οποίο αποτελεί αρχιτεκτονική επεξεργαστή ανεπτυγμένη από την AMD64 (Advanced Micro Devices) η οποία έχει ως στόχο την προσθήκη δυνατοτήτων 64 bit σε αρχιτεκτονική τύπου x68.
2. **Πλατφόρμα:** Το χαρακτηριστικό αυτό σχετίζεται με το λειτουργικό σύστημα της μηχανής, την έκδοση αλλά και τον τύπο του υλικού. Για παράδειγμα «Windows-11-11.1.11111-XXX»
3. **Σύστημα:** Το σύστημα είναι αποκλειστικά και μόνο η ονομασία του λειτουργικού συστήματος, δηλαδή Linux, Windows, macOs κλπ.
4. **Πληροφορίες Συστήματος**: Οι πληροφορίες συστήματος αφορούν το όνομα του δικτύου της μηχανής ,την έκδοση του συστήματος, τον επεξεργαστή και συμπεριλαμβάνουν ξανά το λειτουργικό σύστημα καθώς επίσης και την έκδοσή του , όπως και τον τύπο μηχανής. Άρα ο τύπος που θα αποδοθεί στο συγκεκριμένο χαρακτηριστικό κατά τη συγγραφή του κώδικα είναι αυτός της πλειάδας (tuple).
5. **Επεξεργαστής:** Το χαρακτηριστικό αυτό είναι φανερό πως αφορά τον επεξεργαστή (για παράδειγμα DESKTOP-XXXXXXX, x86_64 κλπ).

Το disCode προσφέρει στον χρήστη 2 επιλογές, την εγγραφή και τη σύνδεση. Για απλούστευση έχουν χρησιμοποιεί 3 διεργασίες οι οποίες είναι υπεύθυνοι για διαφορετικές ενέργειες (rank0: χρήστης, rank1: γεννήτορας του OTP, rank2: υπεύθυνο για τη σύγκριση των δεδομένων της εγγραφής με εκείνα που εισάγει ο

χρήστης κατά τη σύνδεση). Η επικοινωνία αυτών των εξυπηρετητών γίνεται μέσω της ανταλλαγής μηνυμάτων (MPI messages), τα οποία μεταφέρουν σχετικές πληροφορίες.

Κατά τη εγγραφή ο χρήστης εισάγει ένα username και στη συνέχεια ξεκινάει η ανταλλαγή μηνυμάτων μεταξύ των τριών διεργασιών. Αρχικά αποστέλλεται στα rank1, rank2 η επιλογή του χρήστη η οποία έχει οριστεί ως «1». Αμέσως μετά αντλούνται οι πληροφορίες μηχανής και στη συνέχεια αποστέλλεται τόσο στο rank0 όσο και στο rank2 το OTP. Αυτό το OTP ο χρήστης θα πρέπει να το πληκτρολογήσει και να το στείλει με μήνυμα στο rank2, ο οποίος θα κάνε τη σύγκριση μεταξύ αυτού που έλαβε από τον χρήστη και εκείνου από το rank1. Αν τα 2 αυτά είναι ίδια τότε η διαδικασία της εγγραφής είναι επιτυχής.

Αν ο χρήστης επιθυμεί να συνδεθεί τότε η διαδικασία που ακολουθείται είναι παρόμοια. Η επιλογή της σύνδεσης έχει οριστεί ως «2». Ο χρήστης πληκτρολογεί το usemame του και αντλούνται οι πληροφορίες της μηχανής. Αν ο χρήστης είναι ήδη καταχωρημένος τότε εμφανίζεται μήνυμα στην οθόνη για την εισαγωγή του OTP το οποίο έχει σταλεί από το rank1. Το OTP στη συνέχεια αποστέλλεται στο rank2 το οποίο έχει λάβει επίσης ένα OTP και πραγματοποιεί τη σύγκριση αυτών των 2. Αν είναι όμοια η σύνδεση είναι επιτυχής.

disCodeV2

Ωστόσο, στόχος είναι ο εμπλουτισμός του disCode και τελικά η δημιουργία μιας διαδικτυακής εφαρμογής. Έχοντας σχηματίσει μια μικρή ιδέα για τα βασικά εργαλεία στα οποία θα στηριχτεί η εφαρμογή, θα δοθούν περισσότερες λεπτομέρειες για τα δεδομένα που θα χρησιμοποιηθούν ως μέσα ταυτοποίησης (Username και Machine data). Πρέπει να τονιστεί σε αυτό το σημείο πως στις πληροφορίες της μηχανής θα συμπεριληφθεί και ο σειριακός αριθμός της μητρικής. Ο σειριακός αριθμού της μητρικής προσδίδει έναν επιπλέον επίπεδο ασφάλειας καθώς αποτελεί μοναδικό αριθμό τουλάχιστον για τον εκάστοτε κατασκευαστή. Μαζί όμως με τις υπόλοιπες πληροφορίες δημιουργείται μια μοναδική ταυτότητα για κάθε μηχανή η οποία την καθιστά διακριτή.

Πέρα από την πληροφορία του σειριακού αριθμού στα δεδομένα της μηχανής έχει προστεθεί μια βάση δεδομένων (με κατακερματισμένα τα δεδομένα της μηχανής) καθώς επίσης και η διεπαφή που θα μπορεί να χρησιμοποιήσει ο χρήστης κατά τη διάρκεια της διαδικασίας της ταυτοποίησης του. Οι κινήσεις που έχει τη δυνατότητα να κάνει είναι η εγγραφή και η σύνδεση στην εφαρμογή προκειμένου να λάβει πρόσβαση στην κεντρική σελίδα.

Η προηγούμενη έκδοση του disCode χρησιμοποιούσε 3 διεργασίες μια αποδοχή που θα παραμείνει ίδια και σε αυτή την έκδοση (disCodeV2) με τις προσθήκες για να μην αυξηθεί η πολυπλοκότητα της εφαρμογής και να παραμείνει κατανοητή. Αναλυτικότερα η διεπαφή ορίζεται ως το κομμάτι με το αναγνωριστικό rank0 , δηλαδή θα αποτελεί τον χρήστη και κατ' επέκταση τις επιλογές του. Τη βάση θα τη διαχειρίζεται ένα διαφορετικό τμήμα του κώδικα το οποίο ορίζεται ως rank2 και τέλος το αναγνωριστικό rank1 που αφορά την παραγωγή του OTP και την αποστολή email

στον χρήστη (κατά τη διάρκεια της σύνδεσης του χρήστη , το OTP αποστέλλεται μέσω emailστον χρήστη).

Δεν είναι όμως μόνο αυτές οι προσθήκες και οι δυνατότητες που πλαισιώνουν τη νέα έκδοση. Όπως αναφέρθηκε το OTP αποστέλλεται στην ηλεκτρονική διεύθυνση του χρήστη, επομένως είναι απαραίτητο να γίνει η καταχώρηση μιας ηλεκτρονικής διεύθυνσης κατά τη διαδικασία της εγγραφής. Επιπλέον στη διαδικασία της εγγραφής έχει γίνει η προσθήκη της επιλογής ότι ο χρήστης δεν είναι ρομπότ (reCaptchaV2).

Άρα ποιες είναι οι βασικές διαφορές του disCode και disCodeV2; Ακολουθεί ένας συγκεντρωτικός πίνακας.

Μέθοδος	reCaptcha	Βάση δεδομένων	Ασφάλεια	Μοναδικότητα ταυτότητας	Εξυπηρετητές	Διεπαφή	Αποστολή OTP
disCode	Όχι	Όχι	Υψηλή	Όχι	3	Console app	Στην οθόνη του τερματικού
discodeV2	Ναι	Ναι	*Πολύ υψηλή	**Ναι	3	Web app	Μέσω email

Πίνακας 4. Συγκριτικός πίνακας μεταξύ disCode και disCodeV2

*Η ασφάλεια αφορά τον τρόπο που αποθηκεύονται τα δεδομένα του χρήστη. Όπως ειπώθηκε το disCodeV2 αποθηκεύει τα δεδομένα μηχανής σε βάση όχι ως απλά κείμενα αλλά κατακερματισμένα.

** Η μοναδικότητα ταυτότητας αφορά τα δεδομένα της μηχανής. Το disCode χρησιμοποιεί ως δεδομένα μηχανής ένα πρότυπο που περιλαμβάνει τον τύπο της μηχανής, την πλατφόρμα, το σύστημα, διάφορες πληροφορίες συστήματος και τον επεξεργαστή. Αυτά τα δεδομένα όμως δεν είναι μοναδικά ιδίως αν αφορούν τον ίδιο κατασκευαστή ενώ το disCodeV2 περιλαμβάνει και το σειριακό αριθμό της μητρικής προσδίδοντας την απαραίτητη μοναδική ταυτότητα στον χρήστη.

4.3 Σχεδιασμός του disCodeV2

Στην ενότητα αυτή θα παρουσιαστεί η δομή του προγράμματος και του τρόπου με τον οποίο πραγματοποιείται η αυθεντικοποίηση χρήστη. Επίσης θα παρατεθούν διαγράμματα που θα περιγράφουν τις διαδικασίες της εγγραφής και της σύνδεσης , τον τρόπο που δουλεύουν τα μηνύματα MPI καθώς επίσης και της βάσης.

Βασικά στοιχεία του disCodeV2 είναι το MPI, η διεπαφή χρήστη (web app) καθώς επίσης και η βάση δεδομένων. Το MPI όπως έχει ήδη αναφερθεί είναι πρωτόκολλο ανταλλαγής μηνυμάτων

MPI

Το MPI όπως έχει ήδη αναφερθεί είναι πρωτόκολλο ανταλλαγής μηνυμάτων το οποίο επιτρέπει σε πολλές διαφορετικές διεργασίες να τρέχουν παράλληλα. Σε κάθε μία διεργασία αποδόθηκαν συγκεκριμένες καθήκοντα. Ειπώθηκε πως το disCodeV2 θα περιλαμβάνει 3 διεργασίες (rank0,rank1,rank2). Η κάθε μία είναι υπεύθυνη για διαφορετικές ενέργειες.

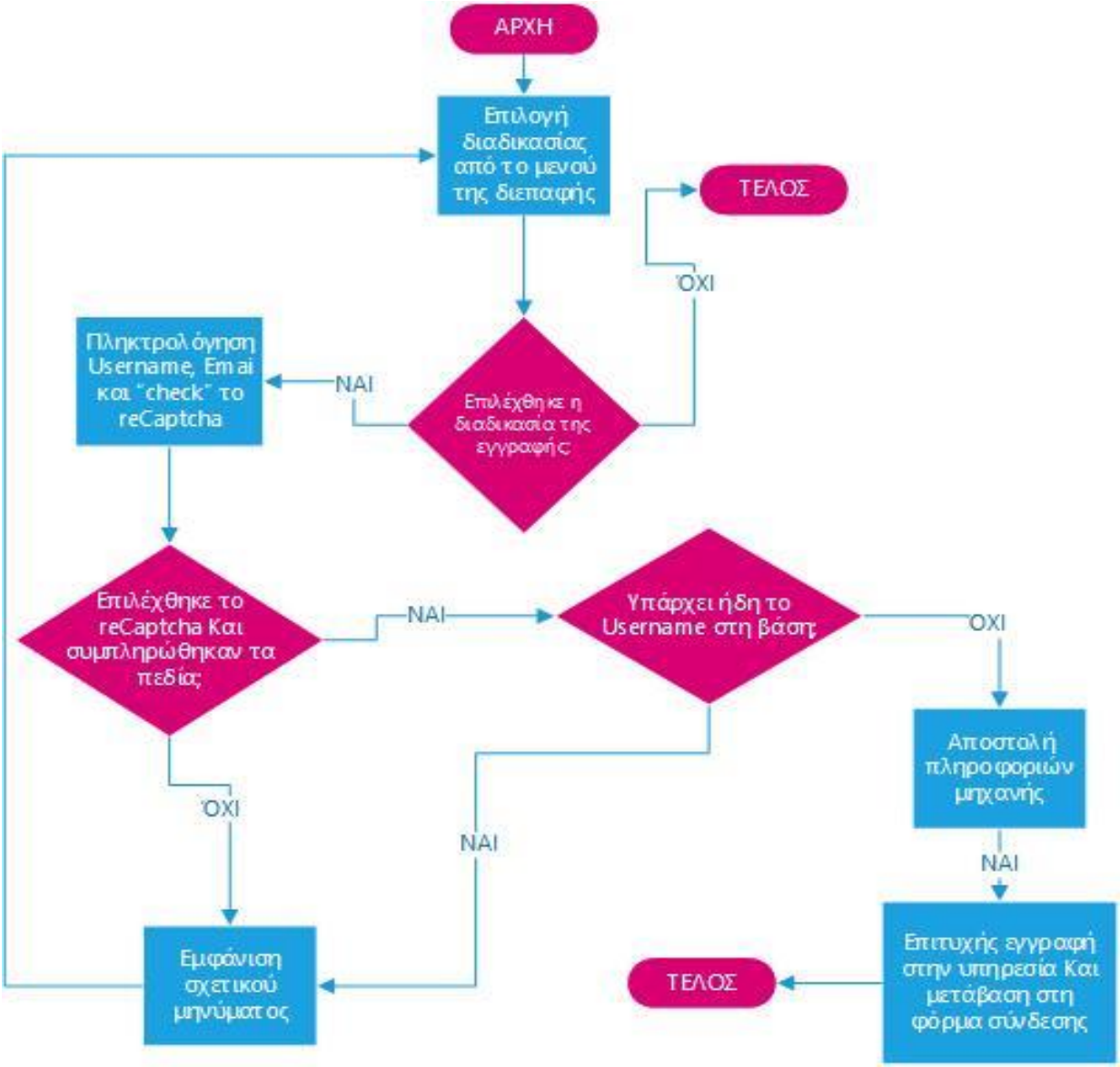
Για τις 3 διεργασίες οι οποίες αποτελούν πυρήνα του προγράμματος και εκτελούν ορισμένες ενέργειες, έχουν αποδοθεί οι ονομασίες rank0,rank1,rank2. Το rank0 είναι ο χρήστης, το rank1 ο γεννήτορας του OTP και το rank2 είναι ουσιαστικά η διεργασία που ασχολείται με τη βάση και την ταυτοποίηση του χρήστη. Το rank1 και η παραγωγή του OTP θα χρησιμοποιηθεί μόνο κατά τη διαδικασία της σύνδεσης. Επομένως βασικό ρόλο κατά τη διαδικασία της εγγραφής παίζουν τα rank0 και rank2, ενώ στην διαδικασία της σύνδεσης χρησιμοποιούνται και τα 3. Ωστόσο, και στις 2 περιπτώσεις τρέχουν παράλληλα και οι 3 διεργασίες απλώς στην πρώτη περίπτωση το rank1 δεν εκτελεί καμία ενέργεια.

Οι 3 διεργασίες ανταλλάσσουν μηνύματα μεταξύ τους μέσω του MPI (send, recv) τα οποία περιλαμβάνουν διαφορετικά δεδομένα ανάλογα με την περίσταση. Αυτά τα μηνύματα αποστέλλονται τόσο κατά τη διάρκεια της εγγραφής όσο και κατά τη διάρκεια της σύνδεσης. Τα δεδομένα που μπορεί να περιέχονται στα μηνύματα είναι η επιλογή του χρήστη (1: Εγγραφή , 2: Σύνδεση), το username του , το email , τα δεδομένα μηχανής και οι απαντήσεις που στέλνει το rank2 στο rank0 με το αποτέλεσμα των συγκρίσεων.

Κατά την εγγραφή ο χρήστης θα εισάγει το username και το email (απαραίτητο για την αποστολή του OTP κατά το στάδιο της σύνδεσης, θα αναλυθεί μετέπειτα). Στη συνέχεια αν πληρούνται οι προδιαγραφές τότε αντλούνται και τα δεδομένα μηχανής και αποθηκεύονται στη βάση δεδομένων (τα δεδομένα μηχανής κατακερματισμένα). Αν δεν υπάρξει κάποιο πρόβλημα κατά την αποθήκευση των δεδομένων στη βάση η διαδικασία της εγγραφής ολοκληρώνεται με επιτυχία.

Κατασκευάστηκε διάγραμμα ροής (flowchart) για την επιλογή της εγγραφής.

- Διαδικασία Εγγραφής SIGN UP [Διάγραμμα 8]

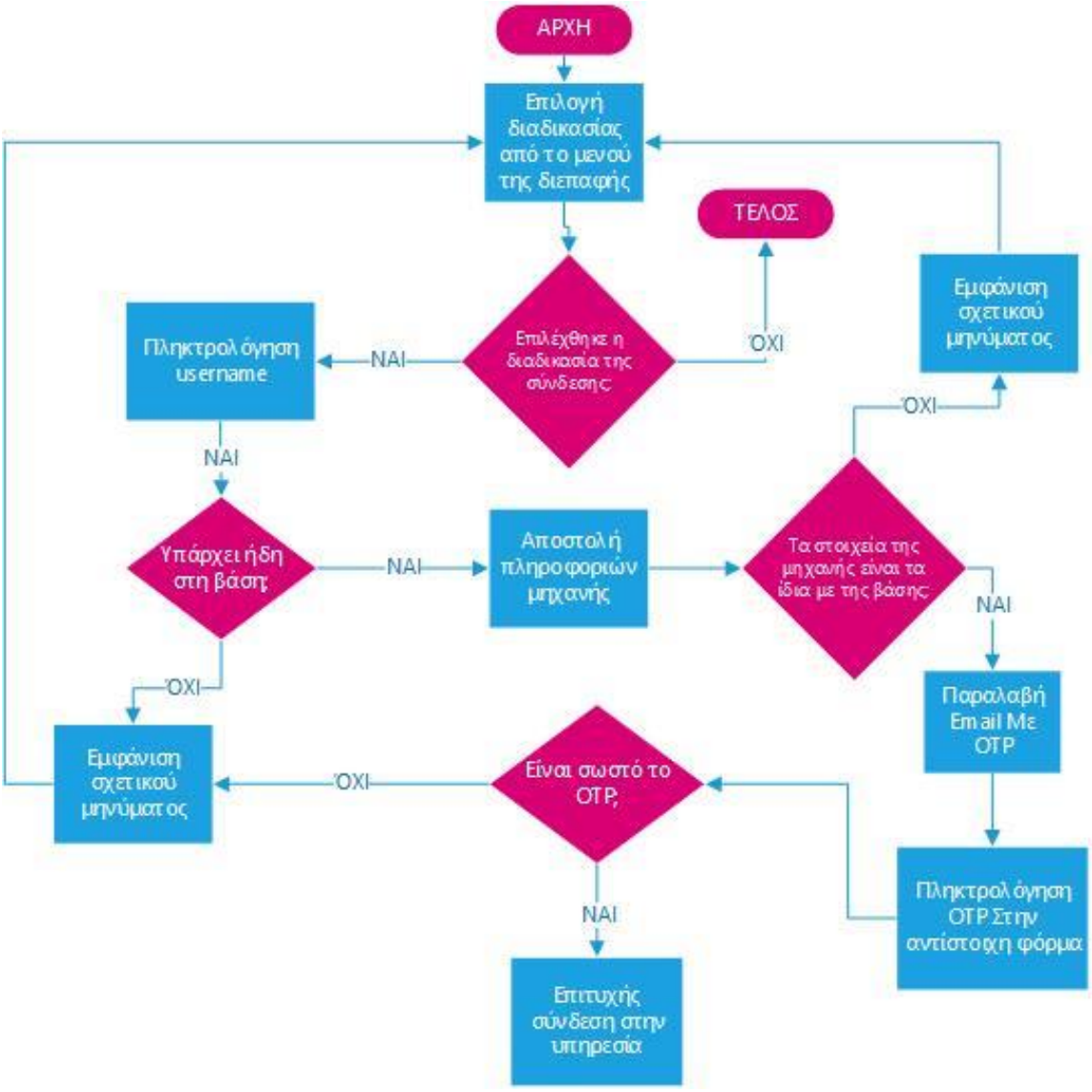


Διάγραμμα 8. Διάγραμμα ροής διαδικασίας εγγραφής με το disCodeV2

Η διαδικασία της σύνδεσης ακολουθεί την ίδια λογική. Κατά τη σύνδεση ο χρήστης θα εισάγει το username του. Στη συνέχεια αν το username είναι ήδη καταχωρημένο στη βάση τότε αντλούνται και τα δεδομένα μηχανής. Γίνεται η σύγκριση των δεδομένων αυτών με εκείνα που έχουν αποθηκευτεί στη βάση, δηλαδή το πρότυπο, και αντιστοιχούν στο συγκεκριμένο username. Αν τα δεδομένα είναι ίδια με το πρότυπο τότε ο χρήστης έχει ταυτοποιηθεί επιτυχώς.

Κατασκευάστηκε διάγραμμα ροής (flowchart) για την επιλογή της σύνδεσης.

- Διαδικασία Σύνδεσης LOGIN [Διάγραμμα 9]



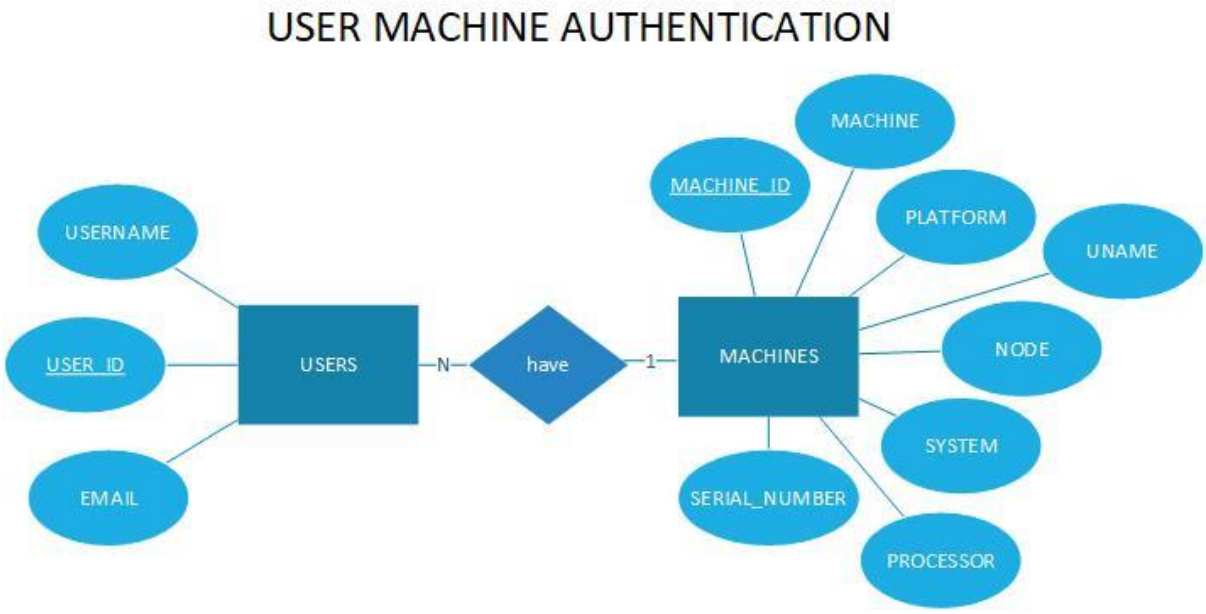
Διάγραμμα 9. Διάγραμμα ροής διαδικασίας σύνδεσης με το disCodeV2

Βάση δεδομένων

Σε αυτό το σημείο θα πρέπει να περιγραφεί και η βάση δεδομένων. Τα δεδομένα που θα πρέπει να αποθηκευτούν είναι τα:

- Username
- Email
- Machine data (πληροφορίες μηχανής και σειριακός αριθμός μητρικής)

Με την παραδοχή ότι κάθε username είναι μοναδικό και αντιστοιχίζεται με μια μόνο μηχανή κατασκευάστηκε ένα διάγραμμα οντοτήτων συσχετίσεων (erd diagram–entity relation diagram) με σημειογραφία Chen.



Διάγραμμα 10. Διάγραμμα ERD σημειογραφίας CHEN για εννοιολογική απεικόνιση της βάσης

Από το παραπάνω διάγραμμα [Διάγραμμα 10] προκύπτουν τα εξής:

Για τις 2 οντότητες USERS και MACHINES θα πρέπει να αποδοθούν σχετικοί πίνακες οι οποίοι θα περιλαμβάνουν τα αντίστοιχα πεδία.

MACHINES							
MACHINE_ID (PK)	MACHINE	PLATFORM	UNAME	NODE	SYSTEM	PROCESSOR	SERIAL_NUMBER

Πίνακας 5.Σχεσιακό μοντέλο του πίνακα MACHINES

Οι αντιστοιχίες του πίνακα ακολουθούν:

1. MACHINE_ID: πρωτεύον κλειδί
2. MACHINE: αποτελεί τον τύπο μηχανής που αποτυπώθηκε σε προηγούμενο σημείο της ενότητας
3. PLATFORM: πλατφόρμα
4. UNAME: πληροφορίες συστήματος
5. NODE: όνομα μηχανής
6. SYSTEM: σύστημα
7. PROCESSOR: επεξεργαστής
8. SERIAL_NUMBER: σειριακός αριθμός μητρικής

Ο πίνακας USERS αποτελείται από 3 πεδία:

1. USER_ID: πρωτεύον κλειδί
2. USERNAME: το ψευδώνυμο που θα χρησιμοποιήσει ο χρήστης
3. MACHINE_ID: δευτερεύον κλειδί, με αναφορά στον πίνακα MACHINES και συνδέει ουσιαστικά τους δύο πίνακες μεταξύ τους

USERS			
USER_ID (PK)	USERNAME	EMAIL	MACHINE_ID (FK)

Πίνακας 6. Σχεσιακό μοντέλο του πίνακα USERS

Διεπαφή χρήστη

Ο χρήστης θα έχει τη δυνατότητα να αλληλεπιδρά με το πρόγραμμα χρησιμοποιώντας μια διεπαφή, στην προκειμένη περίπτωση ένα γραφικό περιβάλλον το οποίο θα είναι φιλικότερο προς το χρήστη έναντι του disCode (περιβάλλον με τερματικό). Άρα ο χρήστης θα ανακατευθύνεται στις διάφορες σελίδες (αρχική σελίδα , σελίδα για εγγραφή , σύνδεση, σελίδα σχετική με τοOTP) της διαδικτυακής εφαρμογής στις οποίες θα κληθεί να συμπληρώσει διαφορετικά στοιχεία και στις οποίες θα εμφανίζονται μηνύματα λάθους (αν χρειαστεί), ώστε να ενημερώνεται ο χρήστης σχετικά με τυχόν αλλαγές που θα πρέπει να κάνει. Αυτές οι φόρμες θα παρουσιαστούν στο επόμενο κεφάλαιο που θα περιλαμβάνει την υλοποίηση της μεθόδου.

ΚΕΦΑΛΑΙΟ 5: ΥΛΟΠΟΙΗΣΗ ΜΗΧΑΝΙΣΜΟΥ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ ΜΕ OTP ΚΑΙ MPI

5.1 Εισαγωγή

Στο κεφάλαιο 4 έγινε αναφορά στο disCodeV2 και τις διαφορές του από το disCode. Στη συνέχεια έγινε η ανάλυση disCodeV2, και παρουσιάστηκε η σχεδίαση του. Στο κεφάλαιο αυτό θα παρουσιαστεί η υλοποίηση του. Θα αναλυθεί ο κώδικας που περιλαμβάνει τις διαθέσιμες λειτουργίες του προγράμματος, οι σελίδες καθώς επίσης και οι φόρμες και τα δεδομένα που θα πρέπει να συμπληρώσει ο χρήστης.

5.2 Υλοποίηση τεχνικού μέρους

Σε αυτό το σημείο και πριν παρουσιαστεί ο κώδικας, θα πρέπει να παραταθούν οι προαπαιτούμενες προϋποθέσεις προκειμένου να μπορεί να τρέξει ο κώδικας. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η python (έκδοση 3.10.7) και το περιβάλλον είναι το PyCharm. Προκειμένου να επιτευχθούν οι παράλληλες λειτουργίες θα χρειαστεί να εγκατασταθεί το MPI. Τα 2 αρχεία msmpisdk.msi και msmpisetup.exe που είναι απαραίτητα ώστε να γίνει η επιτυχής εγκατάσταση του MPI είναι εμπορικά διαθέσιμα [1]. Αφού γίνει η εγκατάσταση αυτών μπορεί να ξεκινήσει η διαδικασία υλοποίησης του κώδικα που θα επιτρέπει στον χρήστη να ταυτοποιηθεί δίχως να χρειάζεται η εισαγωγή κάποιου συνθηματικού.

Σκελετό της εφαρμογής αποτελούν 6 αρχεία (mainSerialNumber.py, app.py, myDatabase.py, helpers.py, app.yaml και configuratorStrings.py) τα οποία διαχειρίζονται συγκεκριμένες διεργασίες καθώς επίσης και ο φάκελος templates ο οποίος περιέχει 5 αρχεία html τα οποία σχετίζονται με τη διεπαφή και την οπτική εικόνα που έχει ο χρήστης σχετικά με την εφαρμογή.

Η εφαρμογή τρέχει με την εντολή **mpiexec -n 3 python mainSerialNumber.py**

Παρακάτω θα παρουσιαστούν, ως ένα βαθμό, τμηματικά τα παραπάνω αρχεία.

app.yaml:

Το αρχείο αυτό αποτελεί αρχείο ρυθμίσεων (configuration file) ώστε να αποθηκεύονται εκεί πληροφορίες σχετικές με το όνομα της βάσης αλλά και του email (email , κωδικός). Ο κωδικός αυτός αποκτήθηκε με τον εξής τρόπο. Στις ρυθμίσεις του λογαριασμού Google >Ασφάλεια > Επαλήθευση σε 2 βήματα > Κωδικοί πρόσβασης εφαρμογής και σε αυτό το σημείο πρέπει να οριστεί ένα όνομα για την εφαρμογή. Επιλέγοντας το «Δημιουργία» δημιουργείται ένας κωδικός πρόσβασης 19 χαρακτήρων έχοντας συμπεριλάβει και τα κενά. Άρα είναι της μορφής:

XXXX XXXX XXX XXXX

Σημαντικό είναι να αναφερθεί πως αποκτήθηκε και ο κωδικός reCapsecret_key. Στη σελίδα https://www.google.com/recaptcha αφού έχει γίνει σύνδεση σε λογαριασμό Google έγινε επιλογή του reCAPTCHA v2(«I'm not a robot»)και έτσι προέκυψε ο κωδικός.

```
database:
  connection_string: όνομα_βασης.db

email:
  sender: xxx@gmail.com
  password: xxxx xxxx xxxx xxxx

key:
  secret_key: yyyyyyyyyyyy
  reCapsecret_key: xyxyxyxyxyyxyxyxyxyxyxyxyxyxy
```

Επομένως όλα τα δεδομένα που θα πρέπει να μείνουν κρυφά θα αποθηκευτούν σε αυτό το αρχείο.

configuratorStrings.py:

Το συγκεκριμένο αρχείο διαβάζει τις πληροφορίες από το .yaml. Περιλαμβάνει 4 συναρτήσεις.

Η συνάρτηση load_config() ανοίγει και διαβάζει το αρχείο yaml που δέχεται ως παράμετρο.

```
# open and read yaml file
def load_config(config_file_path):
    with open(config_file_path, 'r') as yaml_file:
        return yaml.safe_load(yaml_file)
```

Η συνάρτηση databaseConnectionStrings() ουσιαστικά διαβάζει και επιστρέφει την τιμή που έχουν οριστεί για το «database» στο yaml.

```
def databaseConnectionString(config_file_path):
    config = load_config(config_file_path)
    connection_string = config['database']['connection_string']
    # You can set other variables here based on your configuration
    return connection_string
```

Η συνάρτηση emailStrings διαβάζει και επιστρέφει τις τιμές που έχουν οριστεί για το «email»στο yaml , δηλαδή τοemailτου αποστολέα και τον κωδικό.

```
def emailStrings(config_file_path):
    config = load_config(config_file_path)
    email_sender = config['email']['sender']
    email_password = config['email']['password']
    # You can set other variables here based on your configuration
    return email_sender, email_password
```

Η συνάρτηση getSecretKey() διαβάζει και επιστρέφει το ιδιωτικό κλειδί που θα χρησιμοποιηθεί κατά τη διαδικασία της σύνδεσης και της εγγραφής προκειμένου τα δεδομένα της μηχανής (πληροφορίες και σειριακός αριθμός) να αποθηκευτούν στη βάση κατακερματισμένα (hashed).

```
def getSecretKey(config_file_path):
    config = load_config(config_file_path)
    secret_key = config['key']['secret_key']
    return secret_key
```

Τέλος η συνάρτηση reCaptchaSecretKey() διαβάζει και επιστρέφει το ιδιωτικό κλειδί που θα χρησιμοποιηθεί κατά τη διαδικασία της εγγραφής προκειμένου να αποδειχτεί ότι ο χρήστης δεν είναι κάποιο μηχάνημα (robot).

```
def reCaptchaSecretKey(config_file_path):
    config = load_config(config_file_path)
    secret_key = config['key']['reCapsecret_key']
    return secret_key
```

myDatabase.py:

Στο αρχείο αυτό δημιουργείται η βάση , οι 2 πίνακες της (USERS, MACHINES) και έχουν κατασκευαστεί συναρτήσεις οι οποίες εισάγουν δεδομένα στους πίνακες ή εξάγουν δεδομένα από αυτούς. Οι συγκεκριμένες συναρτήσεις θα χρησιμοποιηθούν αποκλειστικά από τον επεξεργαστή τάξης 2. Για να είναι δυνατή η χρήση βάσης δεδομένων θα χρειαστεί η εισαγωγή της βιβλιοθήκης sqlite3.

```
import sqlite3
```

Παρακάτω ακολουθούν οι συναρτήσεις του αρχείου που αφορούν ενέργειες σχετικές με τη βάση και τις διαχειρίζεται το rank2.

```
def create_tables_if_not_exist ():

    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    #con.execute('''DROP TABLE USERS''')
    #con.execute('''DROP TABLE MACHINES''')
    con.execute('''
        CREATE TABLE IF NOT EXISTS MACHINES (
            MACHINE_ID INTEGER PRIMARY KEY AUTOINCREMENT,
            MACHINE TEXT,
            PLATFORM TEXT,
            UNAME TEXT,
            NODE TEXT,
            SYSTEM TEXT,
            PROCESSOR TEXT,
            SERIAL_NUMBER TEXT
        )
```

```
''')

con.execute('''
    CREATE TABLE IF NOT EXISTS USERS (
        USER_ID INTEGER PRIMARY KEY AUTOINCREMENT,
        USERNAME TEXT,
        EMAIL TEXT,
        MACHINE_ID INTEGER,
        FOREIGN KEY (MACHINE_ID) REFERENCES MACHINES(MACHINE_ID)
    )
''')
connection.commit()
```

Η συνάρτηση **create_tables_if_not_exist()** δημιουργεί τους πίνακες USERS και MACHINES, εάν φυσικά δεν υπάρχουν ήδη, αφού προηγουμένως έχει πραγματοποιηθεί η σύνδεση με τη βάση 'myclients.db'. Ο πίνακας MACHINES αποτελείται από 8 πεδία που έχουν ήδη παρουσιαστεί και τώρα δίνονται οι τύποι τους:

1. MACHINE_ID τύπου long (πρωτεύον κλειδί το οποίο με κάθε προσθήκη νέας μηχανής στον πίνακα αυξάνεται κατά 1 [AUTOINCREMENT])
2. MACHINE τύπου text
3. PLATFORM τύπου text
4. UNAME τύπου text
5. NODE τύπου text
6. SYSTEM τύπου text
7. PROCESSOR τύπου text
8. SERIAL_NUMBER τύπου text

MACHINES							
MACHINE_ID (PK) Long	MACHINE text	PLATFORM text	UNAME text	NODE text	SYSTEM text	PROCESSOR Text	SERIAL_NUMBER Text

Πίνακας 7. Σχεσιακό μοντέλο του πίνακα MACHINES με τύπους

Ο πίνακας USERS αποτελείται από 3 πεδία:

1. USER_ID τύπου long (πρωτεύον κλειδί το οποίο με κάθε προσθήκη νέου χρήστη στον πίνακα αυξάνεται κατά 1 [AUTOINCREMENT])
2. USERNAME τύπου text (το ψευδώνυμο που θα χρησιμοποιήσει ο χρήστης)
3. MACHINE_ID τύπου long (δευτερεύον κλειδί, με αναφορά στον πίνακα MACHINES και συνδέει ουσιαστικά τους δύο πίνακες μεταξύ τους)

Ακολουθεί το σχεσιακό μοντέλο του πίνακα [Πίνακας 8]

USERS			
USER_ID (PK) long	USERNAME text	EMAIL text	MACHINE_ID (FK)

Πίνακας 8. Σχεσιακό μοντέλο του πίνακα USERS με τύπους

Μετά την ολοκλήρωση των εντολών για τη δημιουργία των 2 πινάκων εκτελείται η εντολή connection.commit() ώστε να αποθηκευτούν οι οποιεσδήποτε αλλαγές στη βάση δεδομένων.

Η συνάρτηση **close_connection** κλείνει ουσιαστικά την αρχική σύνδεση με τη βάση δεδομένων.

```
def close_connection (con):
   con.close()
```

Η συνάρτηση drop_tables() διαγράφει τους 2 πίνακες.

```
def drop_tables(con):
   connection=sqlite3.connect(connection_string)
   con=connection.cursor()
   con.execute('''DROP TABLE USERS''')
   con.execute('''DROP TABLE MACHINES''')
   connection.commit()
```

Η συνάρτηση insert_machine_info() εισάγει τα δεδομένα της μηχανής στον πίνακα MACHINES και δέχεται ως παράμετρο εκτός από τη σύνδεση με τη βάση τις πληροφορίες της μηχανής και το σειριακό αριθμό.

```
def insert_machine_info(con,machine_info,serial_number):
   connection=sqlite3.connect(connection_string)
   con=connection.cursor()
   machine_id=con.execute(
"INSERT INTO MACHINES (MACHINE, PLATFORM, UNAME, NODE, SYSTEM,
PROCESSOR, SERIAL_NUMBER) VALUES (?, ?, ?, ?, ?, ?, ?)",
 (machine_info['machine'],machine_info['platform'],machine_info['uname'],
machine_info['node'],machine_info['system'],machine_info['processor'],serial_numbe
r)
).lastrowid
   connection.commit()
   return machine_id
```

Η συνάρτηση insert_user() εισάγει τα δεδομένα του χρήστη στον πίνακα USERS και δέχεται ως παράμετρο εκτός από τη σύνδεση με το username και το email.

```
def insert_user(con,username,user_email,machine_id):
```

```
    connection=sqlite3.connect(connection_string)
    con=connection.cursor()
    con.execute("INSERT INTO USERS (USERNAME, EMAIL, MACHINE_ID)
VALUES (?, ?, ?)",
     (username,user_email,machine_id))
    connection.commit()
```

Η συνάρτηση fetch_user_email() επιστρέφει το email του χρήστη που ορίζεται στην παράμετρο.

```
def fetch_user_email(con,username):
    connection=sqlite3.connect(connection_string)
    con=connection.cursor()
    user_email=con.execute(
"SELECT EMAIL FROM USERS WHERE USERNAME=?",(username,))
    connection.commit()
    return user_email.fetchone()
```

Η συνάρτηση check_username_existence() ελέγχει εάν το username υπάρχει ήδη στη βάση.

```
def check_username_existence(con,username):
    connection=sqlite3.connect(connection_string)
    con=connection.cursor()
    con.execute("SELECT COUNT(USERNAME) FROM USERS WHERE
USERNAME=?",(username,))
    username_check=con.fetchone()
    connection.commit()

    return username_check[0]!=0
```

Η συνάρτηση fetch_registered_machine_info() επιστρέφει τις πληροφορίες της μηχανής βάσει του username που λαμβάνει ως παράμετρο.

```
def fetch_registered_machine_info(con,username):
    connection=sqlite3.connect(connection_string)
    con=connection.cursor()
    cursor=con.execute(
"SELECT MACHINES.MACHINE, MACHINES.PLATFORM, MACHINES.UNAME,
MACHINES.NODE, MACHINES.SYSTEM, MACHINES.PROCESSOR"
          " FROM MACHINES, USERS WHERE MACHINES.MACHINE_ID =
          USERS.MACHINE_ID AND USERS.USERNAME=?",
     (username,))
    connection.commit()
    returncursor.fetchone()
```

Η συνάρτηση fetch_serialNumber() επιστρέφει τον σειριακό αριθμό του χρήστη βάσει του username που λαμβάνει ως παράμετρο.

```
def fetch_serialNumber(con,username):
   connection=sqlite3.connect(connection_string)
   con=connection.cursor()
   cursor=con.execute(
"SELECT SERIAL_NUMBER FROM MACHINES, USERS WHERE
MACHINES.MACHINE_ID = USERS.MACHINE_ID AND USERS.USERNAME=?",
   (username,))
   connection.commit()
   returncursor.fetchone()
```

Τέλος η συνάρτηση print_table_rows() τυπώνει όλες τις σειρές των 2 πινάκων.

```
def print_table_raws():
   connection=sqlite3.connect(connection_string)
   con=connection.cursor()
   cursor=con.execute("SELECT * FROM USERS , MACHINES WHERE
MACHINES.MACHINE_ID = USERS.MACHINE_ID")
  connection.commit()

  forrowincon.fetchall():
    print(row)
```

helpers.py:

Το αρχείο αυτό περιέχει χρήσιμες συναρτήσεις οι οποίες θα χρησιμοποιηθούν από τα mainSerialNumber.py και app.py.

Η συνάρτηση write_to_file() έχει καθαρά ρόλο επαλήθευσης των δεδομένων. Ανάλογα με τον τύπο της παραμέτρου που δέχεται ως όρισμα τυπώνει αυτή την πληροφορία με συγκεκριμένη δομή σε αρχείο. Κατά τη συγγραφή του κώδικα και τις ανταλλαγές των μηνυμάτων ήταν δύσκολο να βρεθεί που κολλούσε το πρόγραμμα και περίμενε για κάποια ενέργεια. Επίσης χρησιμοποιήθηκε γιατί μόνο η εντολή print() δε λειτουργεί αν το τμήμα του κώδικα δεν ανήκει στο rank0 (πλευρά του χρήστη).

```
def write_to_file(filename, content):

   with open(filename, 'a') as file:
      if isinstance(content, str):
         file.write(content + '\n')
      elif isinstance(content, tuple):
         for item in content:
            file.write(str(item) + ' |  ')
         file.write('\n')
      elif isinstance(content, dict):
         for key, value in content.items():
            file.write(f"{value} |  ")
```

```
            file.write('\n')
        else:
            file.write(str(content) + '\n')
```

Η συνάρτηση get_motherboard_serial_number() αντλεί το σειριακό αριθμό της μητρικής από το εκάστοτε μηχάνημα χρησιμοποιώντας τις αντίστοιχες εντολές για κάθε μία από τις παρακάτω πλατφόρμες. Χρησιμοποιεί την κλάση platform η οποία νόμιμα μπορεί να συλλέγει δεδομένα μιας συσκευής (έχοντας φυσικά τη συγκατάθεση του χρήστη).

```
def get_motherboard_serial_number():
    system = platform.system()
    if system == "Windows":
        try:
            import wmi
            c = wmi.WMI()
            for board in c.Win32_BaseBoard():
                return board.SerialNumber
        except Exception as e:
            print("Error:", e)
            return None
    elif system == "Linux":
        try:
            output = subprocess.check_output(['sudo', 'dmidecode', '-s', 'baseboard-
serial-number']).strip()
            return output.decode("utf-8")
        except subprocess.CalledProcessError:
            return None
    else:
        print("Unsupported platform:", system)
        return None
```

Η συνάρτηση generateOTP() δημιουργεί ένα 6ψήφιο OTP το οποίο θα χρησιμοποιηθεί κατά τη διαδικασία σύνδεσης αφού ο χρήστης έχει πληκτρολογήσει το username του και έχουν ταυτοποιηθεί τα δεδομένα της μηχανής.

```
def generateOTP ():
    digits = "0123456789"
    OTP = ""
    for i in range(6):
        OTP += digits [random.randint(0, 9)]
    return OTP
```

Η συνάρτηση getEntryInfo() αντλεί τις πληροφορίες της μηχανής (όχι τον σειριακό αριθμό). Οι πληροφορίες αυτές επιστρέφονται ως μια συλλογή από στοιχεία χωρισμένα με «,» (δηλαδή πλειάδα). Χρησιμοποιήθηκε η δομή της πλειάδας και όχι της λίστας καθώς η τελευταία δίνει τη δυνατότητα επεξεργασίας, προσθήκης ή αφαίρεσης στοιχείων. Αντίθετα οι πλειάδες είναι αμετάβλητες και για μικρές συλλογές

στοιχείων είναι πιο αποδοτικές σε θέματα μνήμης και προσπέλασης. Για να αντληθούν οι πληροφορίες χρησιμοποιήθηκε η κλάση platform η οποία εγγυάται πως μπορεί να συλλέξει τα δεδομένα από τη συσκευή στην οποία τρέχει η εφαρμογή που της καλεί.

```
def getEntryInfo ():

    try:
        machine_info = platform.machine()
        platform_info = platform.platform()
        uname_info = platform.uname()
        system_info = platform.system()
        processor_info = platform.processor()
        uname_string = f"system={uname_info.system}, node={uname_info.node},
release={uname_info.release}, version={uname_info.version},
machine={uname_info.machine}"
        return machine_info, platform_info, uname_string, uname_info.node,
system_info, processor_info

    except Exception as e:
        print("Error while getting entry information:", e)
        return None, None, None, None, None, None
```

Η συνάρτηση send_otp_viaEmail() δέχεται ως παραμέτρους το email του αποδέκτη (το οποίο έχει καταχωρηθεί στη βάση κατά τη διαδικασία εγγραφής στην εφαρμογή) και το OTP

```
def send_otp_viaEmail(email_receiver, otp):

    email_sender, email_password = emailStrings(config_file_path)
    subject = 'Otp'
    body = f"The OTP you received from MyDrama App is: {otp}"

    em = EmailMessage()
    em['From'] = email_sender
    em['To'] = email_receiver
    em['Subject'] = subject
    em.set_content(body)

    # layer of security
    context = ssl.create_default_context()

    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
        smtp.login(email_sender, email_password)
        smtp.sendmail(email_sender, email_receiver, em.as_string())
```

Φάκελος templates:

Ο φάκελος αυτός περιλαμβάνει 5 αρχεία html (welcome.html, register.html, login.html, otp_form.html και main.html). Τα αρχεία αυτά διαχειρίζονται διαφορετικό στάδιο της διαδικασίας της ταυτοποίησης και κατ΄ επέκταση της διαδικτυακής εφαρμογής.

- welcome.html:
  Το αρχείο αυτό είναι η σελίδα που βλέπει ο χρήστης κατά την εκκίνηση του προγράμματος. Περιλαμβάνονται 2 επιλογές (σύνδεση και εγγραφή). Ανάλογα με την επιλογή ο χρήστης ανακατευθύνεται στη σωστή σελίδα.

```
<!DOCTYPE html>
<html>
<head>
   <title>Welcome</title>
</head>
<body>
   <h1>Welcome to myPassLess Web App Authentication Form</h1>
   <p>Please select an option:</p>
   <ul>
      <li><a href="/register">Register</a></li>
      <li><a href="/login">Login</a></li>
   </ul>
</body>
</html>
```

- register.html
  Αν ο χρήστης επιλέξει την εγγραφή τότε ανακατευθύνεται στη συγκεκριμένη σελίδα όπου καλείται να πληκτρολογήσει ένα username και το email του (πεδία υποχρεωτικά) καθώς επίσης να επιλέξει και την επιλογή ότι δεν είναι μηχανή (απαραίτητο πεδίο).

```
<htmllang="en">
<head>
<metacharset="UTF-8">
<title>Registration</title>
<scriptsrc="https://www.google.com/recaptcha/api.js"asyncdefer></script>
<!-- Include Google reCAPTCHA JavaScript -->
</head>
<body>
<h1>Registration</h1>
<p>Enter your details below:</p>

   {% if message %}
<pstyle="color: red;">{{ message }}</p>
   {% endif %}
```

```
<formaction="/register"method="post">
<labelfor="username">Username:</label>
<inputtype="text"id="username"name="username"required><br><br>
<labelfor="username">Email:</label>
<inputtype="text"id="Email"name="email"required><br><br>
<!-- Google reCAPTCHA -->
<divclass="g-recaptcha" data-
sitekey="6LfN43YpAAAAAJlGuEfpQbYLGZVA_uEXqp5COcAk"></div>
<br>
<inputtype="submit"value="Sign Up">
</form>
</body>
</html>
```

- login.html

  Αν ο χρήστης επιλέξει τη σύνδεση από τη σελίδα welcome.html ή μετά την επιτυχή εγγραφή (μετά την εγγραφή ανακατευθύνεται αυτόματα στη σελίδα σύνδεσης) χρειάζεται να πληκτρολογήσει το username του (υποχρεωτικό πεδίο).

```
<!DOCTYPE html>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>Login</title>
</head>
<body>
<h1>Login</h1>

   {% if message %}
<pstyle="color: red;">{{ message }}</p>
   {% endif %}

<formaction="/login"method="post">
<labelfor="username">Username:</label><br>
<inputtype="text"id="username"name="username"required><br><br>

<buttontype="submit">Login</button>
</form>
</body>
</html>
```

- otp_form.html

  Στη σελίδα αυτή ο χρήστης αφού έχει εισάγει το username του στη σελίδα login.html χρειάζεται να συμπληρώσει το 6ψήφιο OTP που έλαβε μέσω email.

```
<!DOCTYPE html>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>OTP</title>
<style>
   body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
   }

  .otp-form {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
   }

  .otp-input {
      width: 40px;
      height: 40px;
      text-align: center;
      font-size: 18px;
      margin: 0 5px;
      border: 1px solid #ccc;
      border-radius: 4px;
      outline: none;
   }

  .otp-input:focus {
      border-color: #007bff;
   }

  .submit-btn {
      background-color: #007bff;
      color: #fff;
      border: none;
```

```
        padding: 10px 20px;
        font-size: 16px;
        border-radius: 4px;
        cursor: pointer;
    }
</style>
</head>
<body>
<divclass="otp-form">
<h2>Enter OTP</h2>
<formaction="/verify_otp"method="post">
<inputtype="text"class="otp-
input"name="digit1"maxlength="1"requiredautofocus>
<inputtype="text"class="otp-input"name="digit2"maxlength="1"required>
<inputtype="text"class="otp-input"name="digit3"maxlength="1"required>
<inputtype="text"class="otp-input"name="digit4"maxlength="1"required>
<inputtype="text"class="otp-input"name="digit5"maxlength="1"required>
<inputtype="text"class="otp-input"name="digit6"maxlength="1"required>
<br><br>
<buttontype="submit"class="submit-btn">Submit OTP</button>
</form>
</div>
</body>
</html>
```

- main.html

  Τέλος η σελίδα αυτή είναι το τελικό στάδιο της διαδικασίας ταυτοποίησης και ο χρήστης πλέον έχει πρόσβαση στις δυνατότητες της υπηρεσίας.

```
<!DOCTYPE html>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>myPassLess</title>
<style>
        /* CSS styles go here */
        body {
            font-family: Arial, sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
        }

        header {
            background-color: #333;
            color: #fff;
                    padding: 10px;
            text-align: center;
```

```
    }

    nav {
      background-color: #444;
      color: #fff;
      padding: 10px;
      text-align: center;
    }

    nav ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
    }

    nav ul li {
      display: inline;
      margin-right: 10px;
    }

    nav ul li a {
      color: #fff;
      text-decoration: none;
      padding: 5px 10px;
    }

    nav ul li a:hover {
      background-color: #ddd; /* Light grey background on hover */
      color: #444; /* Dark grey text on hover */
    }

    section {
      padding: 20px;
      text-align: center;
    }

    footer {
      background-color: #333;
      color: #fff;
      padding: 10px;
      text-align: center;
      position: fixed;
      bottom: 0;
      width: 100%;
    }
</style>
</head>
<body>
<header>
<h1>myPassLess</h1>
```

```
    </header>

    <nav>
    <ul>
    <li><a href="#">Help</a></li>
    </ul>
    </nav>

    <section>
    <h2>Welcome to myPassLess</h2>
    <p>This is my sample webpage.</p>
    </section>

    </body>
    </html>
```

app.py:

Το αρχείο αυτό αφορά την πλευρά του χρήστη καθώς σχετίζεται με την διεπαφή. Πιο συγκεκριμένα περιλαμβάνει τις ενέργειες για τη σύνδεση και την εγγραφή από την πλευρά του χρήστη ο οποίος βλέπει τις φόρμες που θα πρέπει να συμπληρώσει. Ξεκινώντας με τη σελίδα υποδοχής χρήστη όπου ανακατευθύνεται στη σελίδα welcome.html.

```
def welcome():
    return render_template('welcome.html')
```

Στη συνέχεια αν επιλέξει την εγγραφή τότε το τμήμα του κώδικα που διαχειρίζεται αυτή την ενέργεια είναι το παρακάτω

```
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method=='GET':
        # Handle GET request
        return render_template('register.html')
    elif request.method=='POST':
        # Handle POST request
        msg1['option'] = 1
        username = request.form['username']
        user_email = request.form['email']
        recaptcha_response = request.form.get('g-recaptcha-response')
        # Verify the reCAPTCHA response with Google's reCAPTCHA verification API
        response = requests.post('https://www.google.com/recaptcha/api/siteverify',
                                 data={'secret': recaptcha_secret_key, 'response':
recaptcha_response})
        challengeResult = response.json()
        if challengeResult['success']:
            msg1['username'] = username
            comm.send(msg1, dest=2, tag=1)
```

```
        # Receive response from rank 2 if username already exists in database or is
available
        message = comm.recv(source=2, tag=2)
        if message['flag'] == 0 : # username not available
            write_to_file('results.txt', message['message'])
            return render_template('register.html', message=message['message'])
        else: # when username is available , then fetching device info

            write_to_file('results.txt',message['message'])
            entry['username'] = username
            entry['email'] = user_email
            entry['machineInfo'] = getEntryInfo()
            entry['serialNumber'] = get_motherboard_serial_number()
            write_to_file('results.txt', entry['machineInfo'])
            write_to_file('results.txt', entry['serialNumber'])
            # Username is available, proceed with registration or redirect to another
page
            comm.send(entry, dest=2, tag=3)
            message = comm.recv(source=2, tag=4)
            if message ['flag']==1:
                return redirect(url_for('login'))
            else:
                return render_template('register.html', message="something went
wrong")
        else:
            #reCAPTCHA verification failed, handle accordingly
            return render_template('register.html', message="reCAPTCHA verification
failed. Please try again.")
```

Τη στιγμή που ο χρήστης επιλέγει να εγγραφεί αυτόματα ανακατευθύνεται στη σελίδα που σχετίζεται με το rergister.html. Ορίζεται το msg1 τύπου dictionary. Τα dictionaries ουσιαστικά χρησιμοποιούνται για να αποθηκεύσουν ζεύγη πληροφοριών που σχηματίζουν ένα αντικείμενο. Η δομή τους είναι τύπου :

Dict[‘key’] = value(όπου key είναι το είδος της πληροφορίας καιvalue η πληροφορία)

Το msg1 προς το παρόν περιλαμβάνει την επιλογή του χρήστη για το msg1[‘option’] (έχει οριστεί πως η εγγραφή είναι η επιλογή με αναγνωριστικό 1). Ο χρήστης εισάγει τα δεδομένα στη φόρμα. Αν δεν επιλέξει το reCAPTCHAV2 εμφανίζεται αντίστοιχο μήνυμα στη σελίδα. Τα πεδία username και email είναι υποχρεωτικά οπότε αν δεν έχουν συμπληρωθεί εμφανίζεται και πάλι αντίστοιχο μήνυμα. Αν τώρα ο χρήστης τα πραγματοποιήσει όλα ορθά τότε στο msg1 προστίθεται και το username (msg1[‘username’]) και αποστέλλεται μήνυμα στο rank2, δηλαδή τη διεργασία που σχετίζεται με τη βάση. Το μήνυμα αυτό περιέχει την πληροφορία του msg1.

Στη συνέχεια αναμένει να λάβει απάντηση από το rank2 το οποίο περιλαμβάνει πληροφορία σχετικά με τη διαθεσιμότητα του username στη βάση. Αν λοιπόν το usemame δεν είναι διαθέσιμο τότε εμφανίζεται σχετικό μήνυμα αλλιώς οι πληροφορίες για το email , το username και τα δεδομένα της μηχανής (πληροφορίες

μηχανής και σειριακός αριθμός) αποστέλλονται με μήνυμα πάλι στο rank2 μέσω του dictionary entry με τα αντίστοιχα ζεύγη. Στη συνέχεια λαμβάνει απάντηση από το rank2 που σχετίζεται με την καταχώρηση των πληροφοριών του χρήστη στη βάση. Αν η απάντηση είναι θετική (1) τότε ο χρήστης ανακατευθύνεται στη σελίδα της σύνδεσης (login). Σε αντίθετη περίπτωση εμφανίζεται σχετικό μήνυμα.

Με την ίδια λογική ανταλλαγής μηνυμάτων έχει χτιστεί και ο κώδικας για την σύνδεση του χρήστη στην υπηρεσία. Κατά τη διαδικασία της σύνδεσης ο χρήστης καλείται να συμπληρώσει δύο διαφορετικές φόρμες (εκείνη της σύνδεσης και μετέπειτα του OTP). Παρακάτω ακολουθεί το αντίστοιχο τμήμα του κώδικα και μια σύντομη ανάλυση του.

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        # Handle GET request
        return render_template('login.html')
    elif request.method =='POST':
        # Handle POST request
        msg1['option'] = 2
        username = request.form['username']
        msg1['username'] = username
        comm.send(msg1, dest=2, tag=1)
        # Receive response from rank 2 if username already exists in database or is
available
        message = comm.recv(source=2, tag=2)
        if message['flag'] == 0 : # username not registered
            return render_template('login.html', message=message['message'])
        else: # send machine info
            entry['username'] = username
            entry['machineInfo'] = getEntryInfo()
            entry['serialNumber'] = get_motherboard_serial_number()
            #write_to_file('results.txt', "blabla2")
            comm.send(entry, dest=2, tag=3)
        message = comm.recv(source=2, tag=4)
        if message['flag'] == 0:  # unsuccessful login
            return render_template('login.html', message=message['message'])
        else:  # successfully login
            return render_template('otp_form.html')
```

Για τη διαδικασία της σύνδεσης έχει οριστεί το αναγνωριστικό 2. Άρα το dictionary msg1 περιλαμβάνει την επιλογή του χρήστη για το msg1[‘option’] = 2. Στη συνέχεια και το msg1[‘username’] λαμβάνει την τιμή του username και αποστέλλεται σχετικό μήνυμα στο με το msg1 στο rank2. Αναμένει την απάντηση του rank2 σχετικά με την ύπαρξη ή όχι το συγκεκριμένου username στη βάση. Αν το μήνυμα είναι αρνητικό (0) τότε εμφανίζεται σχετικό μήνυμα στον χρήστη αλλιώς η διαδικασία συνεχίζει και αποστέλλεται το dictionary entry που συμπεριλαμβάνει το username, τις πληροφορίες της μηχανής και τον σειριακό αριθμό και πάλι στο rank2. Αναμένει την

απάντηση του η οποία σχετίζεται με την ταύτιση – ομοιότητα των δεδομένων της μηχανής που αποστέλλονται και εκείνων που έχουν αποθηκευτεί κατά τη διαδικασία της εγγραφής. Αν λοιπόν τα δεδομένα δεν ταυτίζονται εμφανίζεται σχετικό μήνυμα αλλιώς ο χρήστης ανακατευθύνεται στη φόρμα του OTP.

```
@app.route('/verify_otp', methods=['GET', 'POST'])

def verify_otp():
    if request.method == 'GET':
        # Handle GET request
        return render_template('otp_form.html')
    elif request.method == 'POST':
        # Retrieve OTP entered by the user from the form
        otp = request.form['digit1'] + request.form['digit2'] + request.form['digit3'] +
request.form['digit4'] + request.form['digit5'] + request.form['digit6']
        #writeStr_to_file('results.txt', otp)
        comm.send(otp, dest=2, tag=5)
        message = comm.recv(source=2, tag=6)
        if message['flag'] == 0:  # wrong otp
            return render_template('login.html')
        else:
            return redirect(url_for('main'))
        # Send entered OTP to Rank 2 using MPI
```

Σε αυτό το σημείο ο χρήστης καλείται να συμπληρώσει το 6ψήφιο OTP που έλαβε μέσω email. Μόλις το συμπληρώσει αποστέλλεται μήνυμα στο rank2 που περιλαμβάνει αυτή την πληροφορία και αναμένει την απάντηση του. Όταν στάλθηκε το email στον χρήστη στάλθηκε μήνυμα με το ίδιο OTP και στο rank2.Επομένως αν η απάντηση του rank2 είναι θετική (1), άρα έχουν ταυτιστεί το OTP του χρήστη με εκείνο που rank2, ο χρήστης ταυτοποιείται πλήρως και αποκτά πρόσβαση στην υπηρεσία – σελίδα. Σε αντίθετη περίπτωση ανακατευθύνεται πάλι στη φόρμα της σύνδεσης.

mainSerialNumber.py

Έχοντας παρουσιάσει ήδη την πλευρά του χρήστη με τη διεπαφή θα γίνει μια σύντομη ανάλυση και στην πλευρά των διεργασιών rank1 και rank2 αλλά και μια μικρή συμπλήρωση στο κομμάτι του χρήστη. Όπως έγινε αντιληπτό η διεργασία που αφορά τον χρήστη rank0 στέλνει μηνύματα μόνο στο rank2. Ωστόσο, εδώ έγινε αναφορά και στο rank1. Τι είναι λοιπόν το rank1 και ποια η λειτουργία του;

Η διεργασία rank1 χρησιμοποιείται μόνο στην περίπτωση της σύνδεσης ώστε να παραχθεί το OTP που θα αποσταλεί και στον χρήστη και στο rank2. Πιο λεπτομερής περιγραφή θα παρατεθεί και στη συνέχεια μαζί με τον αντίστοιχο κώδικα.

Το αρχείο mainSerialNumber.py είναι σημαντικό να αναφερθεί πως περιλαμβάνει τη main, επομένως αυτό αποτελεί και το εναρκτήριο σημείο του προγράμματος. Εδώ γίνεται και ο διαχωρισμός των ranks και των λειτουργιών που εκτελεί το κάθε ένα.

Επομένως αν η διεργασία είναι το rank0 τότε καλείται η συνάρτηση openWebApi() που περιγράφηκε στο app.py

```
if rank==0:# client
  whileTrue:
    openWenApi()
```

Στο κομμάτι του κώδικα για το rank2 και τη διαδικασία της εγγραφής ο κώδικας παρατίθεται παρακάτω

```
 elif msg1 ['option']==1:  # register

            if check_username_existence(con, msg1 ['username']):
                retval ['flag'] = 0
                retval ['message'] = "Username already in use. Please choose a
different username."
                comm.send(retval, dest=0, tag=2)
            else:  # the username can be used
                retval ['flag'] = 1
                retval ['message'] = "Available username. We will now fetch some
device information"
                comm.send(retval, dest=0, tag=2)
                # fetch machine info and serial number of motherboard
                entry = comm.recv(source=0, tag=3)
                # before insert plaintext data into tables it's a good addition to hash
them
                # Serialize the tuple to a string
                # Insert user information into the database
                machine_info_tuple = entry ['machineInfo']
                serial_number = entry ['serialNumber']

                if (machine_info_tuple is None):
                    retval ['flag'] = 0
                    retval ['message'] = "Something went wrong with the Machine Info"
                    comm.send(retval, dest=0, tag=4)
                if (serial_number is None):
                    retval ['flag'] = 0
                    retval ['message'] = "Something went wrong whit the serial number"
                    comm.send(retval, dest=0, tag=4)

                else:
                    hashed_serialNumber = hmac.new(secret_key,
str(serial_number).encode('utf-8'),
                                                   hashlib.sha256).hexdigest()
                    uname_result = platform.uname()
                    uname_string = f"system={uname_result.system},
node={uname_result.node}, release={uname_result.release},
version={uname_result.version}, machine={uname_result.machine}"
                    hashed_machine_info = {
                        'machine': hmac.new(secret_key, str(machine_info_tuple
[0]).encode('utf-8'),
```

```
                                    hashlib.sha256).hexdigest(),
                    'platform': hmac.new(secret_key, str(machine_info_tuple
[1]).encode('utf-8'),
                                    hashlib.sha256).hexdigest(),
                    'uname': hmac.new(secret_key, uname_string.encode('utf-8'),
hashlib.sha256).hexdigest(),
                    'node': hmac.new(secret_key, uname_result.node.encode('utf-8'),
hashlib.sha256).hexdigest(),
                    'system': hmac.new(secret_key, str(machine_info_tuple
[4]).encode('utf-8'),
                                    hashlib.sha256).hexdigest(),
                    'processor': hmac.new(secret_key, str(machine_info_tuple
[5]).encode('utf-8'),
                                    hashlib.sha256).hexdigest(),
                }

                write_to_file('results.txt', hashed_machine_info)
                # Insert hashed_machine_info into the database
                machine_id = insert_machine_info(con, hashed_machine_info,
hashed_serialNumber)

                insert_user(con, entry ['username'], entry['email'], machine_id)
                retval ['flag'] = 1
                retval ['message'] = "You have successfully signed up"
                comm.send(retval, dest=0, tag=4)
```

Αν λοιπόν η επιλογή που λαμβάνει το rank2 από το rank1 είναι 1 (δηλαδή εγγραφή) τότε γίνεται έλεγχος εάν το username είναι ήδη καταχωρημένο στη βάση. Αν δεν είναι διαθέσιμο τότε στέλνει στο rank1 αρνητική απάντηση (0) με αντίστοιχο μήνυμα που ενημερώνει ποιο είναι το πρόβλημα. Αντίθετα αν είναι διαθέσιμο τότε η απάντηση που αποστέλλεται είναι θετική (1) και το μήνυμα επιβεβαίωσης.

Στη συνέχεια αναμένει τις πληροφορίες του χρήστη (email , username , πληροφορίες μηχανής και σειριακό αριθμό). Μόλις τις λάβει γίνεται έλεγχος αν όντως έλαβε σωστά αυτές τις πληροφορίες. Αν δηλαδή τα δεδομένα της μηχανής λήφθηκαν κενά τότε στέλνει αντίστοιχο μήνυμα στο χρήστη ώστε να τον πληροφορήσει κατάλληλα. Αλλιώς ξεκινάει η διαδικασία κατακερματισμού των δεδομένων της μηχανής ώστε να αποθηκευτούν στη βάση.

Ο αλγόριθμος κατακερματισμού που χρησιμοποιήθηκε είναι ο hmac. Κάνοντας κατάλληλες μετατροπές στην πλειάδα με τις πληροφορίες της μηχανής και στο σειριακό αριθμό, ξεκινάει η διαδικασία εισαγωγής τους στη βάση. Πρώτα γίνεται η εισαγωγή στον πίνακα machines ώστε να σχηματιστεί και το machine_id το οποίο θα χρησιμοποιηθεί ως δευτερεύον κλειδί στον πίνακα users αμέσως μετά. Για την εισαγωγή του νέου χρήστη χρησιμοποιούνται το username, email και το machine_id. Αν αυτές οι ενέργειες ολοκληρωθούν τότε από το rank2 αποστέλλεται κατάλληλο μήνυμα επιτυχούς εγγραφής στο rank1.

Για τη διαδικασία της σύνδεσης ο αντίστοιχος κώδικας ακολουθεί.

```
elif msg1 ['option']==2:  # the user chose login option
            # check if username exists
            if not check_username_existence(con, msg1 ['username']):
                retval ['flag'] = 0
                retval ['message'] = "Wrong username"
                comm.send(retval, dest=0, tag=2)
            else:
                retval ['flag'] = 1
                retval ['message'] = "Username is included"
                comm.send(retval, dest=0, tag=2)
                entry = comm.recv(source=0, tag=3)
                expected_machine_info = fetch_registered_machine_info(con,
entry['username'])
                expected_serialNumber = fetch_serialNumber(con, entry['username'])
                #writeTuple_to_file('results.txt', expected_machine_info)
                if expected_machine_info is None:
                    retval ['flag'] = 0
                    retval['message'] = "Error when fetching the machine info"
                elif expected_serialNumber is None:
                    retval ['flag'] = 0
                    retval ['message'] = "Error when fetching the stored serial number "
                else:  # hash the data we get from user in order to check if they are the
same as in database
                    machine_info_tuple = entry['machineInfo']
                    serial_number = entry['serialNumber']
                    hashed_serialNumber = hmac.new(secret_key,
str(serial_number).encode('utf-8'),
                                                   hashlib.sha256).hexdigest()
                    #writeTuple_to_file('results.txt', machine_info_tuple)
                    uname_result = platform.uname()
                    uname_string = f"system={uname_result.system},
node={uname_result.node}, release={uname_result.release},
version={uname_result.version}, machine={uname_result.machine}"
                    hashed_machine_info = {
                        'machine': hmac.new(secret_key, str(machine_info_tuple
[0]).encode('utf-8'),
                                            hashlib.sha256).hexdigest(),
                        'platform': hmac.new(secret_key, str(machine_info_tuple
[1]).encode('utf-8'),
                                             hashlib.sha256).hexdigest(),
                        'uname': hmac.new(secret_key, uname_string.encode('utf-8'),
hashlib.sha256).hexdigest(),
                        'node': hmac.new(secret_key, uname_result.node.encode('utf-8'),
hashlib.sha256).hexdigest(),
                        'system': hmac.new(secret_key, str(machine_info_tuple
[4]).encode('utf-8'),
                                           hashlib.sha256).hexdigest(),
                        'processor': hmac.new(secret_key, str(machine_info_tuple
[5]).encode('utf-8'),
                                              hashlib.sha256).hexdigest(),
```

```
                }
                #writeDict_to_file('results.txt', hashed_machine_info)
                hashed_machine_info_tuple = (
                    hashed_machine_info ['machine'],
                    hashed_machine_info ['platform'],
                    hashed_machine_info ['uname'],
                    hashed_machine_info ['node'],
                    hashed_machine_info ['system'],
                    hashed_machine_info ['processor']
                )

                #write_to_file('results.txt', str(type(hashed_serialNumber)))
                #write_to_file('results.txt', str(type(expected_serialNumber)))

                #expected_machine_info_set = set(expected_machine_info.items())
                if hashed_machine_info_tuple == expected_machine_info and
hashed_serialNumber == str( expected_serialNumber[0] ):
                    retval['flag'] = 1   # machine info have been matched
                    msg1['email'] = fetch_user_email(con,entry['username'])
                    comm.send(msg1, dest=1, tag=11)
                    local_OTP = comm.recv(source=1 , tag=12)
                    write_to_file('results.txt','local_otp: '+local_OTP)
                    retval['message'] = "Email with otp has been send"
                    comm.send(retval, dest=0, tag=4)
                    user_OTP = comm.recv(source=0, tag=5)
                    write_to_file('results.txt', 'user_otp: ' + user_OTP)
                    if local_OTP == user_OTP:
                        retval ['flag'] = 1
                        retval ['message'] = "Correct Otp "
                    else:
                        retval ['flag'] = 0
                        retval ['message'] = "Wrong Otp "
                    write_to_file('results.txt', retval['message'])
                    comm.send(retval, dest=0, tag=6)

                else:
                    retval ['flag'] = 0
                    retval ['message'] = "Machine info were not matched with those in
system... "
                    write_to_file('results.txt', retval['message'] )
                    comm.send(retval, dest=0, tag=4)
```

Αν η επιλογή που λαμβάνει το rank2 από το rank0 είναι 2 (δηλαδή σύνδεση) τότε γίνεται έλεγχος εάν το username είναι ήδη καταχωρημένο στη βάση. Αν δεν είναι διαθέσιμο τότε στέλνει στο rank1 αρνητική απάντηση (0) με αντίστοιχο μήνυμα που ενημερώνει ποιο είναι το πρόβλημα. Αντίθετα αν είναι διαθέσιμο τότε η απάντηση που αποστέλλεται είναι θετική (1) και το μήνυμα επιβεβαίωσης.

Στη συνέχεια αναμένει τις πληροφορίες του χρήστη (username , πληροφορίες μηχανής και σειριακό αριθμό). Μόλις τις λάβει γίνεται έλεγχος αν όντως έλαβε σωστά

αυτές τις πληροφορίες. Αν δηλαδή τα δεδομένα της μηχανής λήφθηκαν κενά τότε στέλνει αντίστοιχο μήνυμα στο χρήστη ώστε να τον πληροφορήσει κατάλληλα. Αλλιώς ξεκινάει η διαδικασία ελέγχου ομοιότητας μεταξύ των δεδομένων που μόλις έλαβε από τον χρήστη (αφού γίνει η επεξεργασία τους και πάλι με τον αλγόριθμο κατακερματισμού hmac) και των δεδομένων που είναι καταχωρημένα στη βάση. Αν όλα αυτά ταυτιστούν τότε για τον εκάστοτε χρήστη αναζητείται το καταχωρημένο email. Η εύρεση του email πραγματοποιείται προκειμένου να σταλθεί στη σωστή ηλεκτρονική διεύθυνση το OTP.

Μόλις βρεθεί το σωστό email από τη βάση αποστέλλεται σχετικό μήνυμα στο rank1. Στη συνέχεια αναμένει να λάβει το OTP που θα του στείλει το rank1. Αμέσως μετά στέλνει ενημερωτικό μήνυμα στο rank0 και περιμένει να λάβει το μήνυμα με το OTP,που συμπλήρωσε ο χρήστης, από το rank0. Γίνεται η σύγκριση του OTP που έχει το rank2 με εκείνο που στάλθηκε από το rank0. Αν είναι όμοια τότε το μήνυμα ταυτοποίησης είναι θετικό (1) και αποστέλλεται στο rank0 αλλιώς αποστέλλεται αρνητικό μήνυμα (0).

Τέλος το κομμάτι που αφορά το rank1 παρατίθεται παρακάτω.

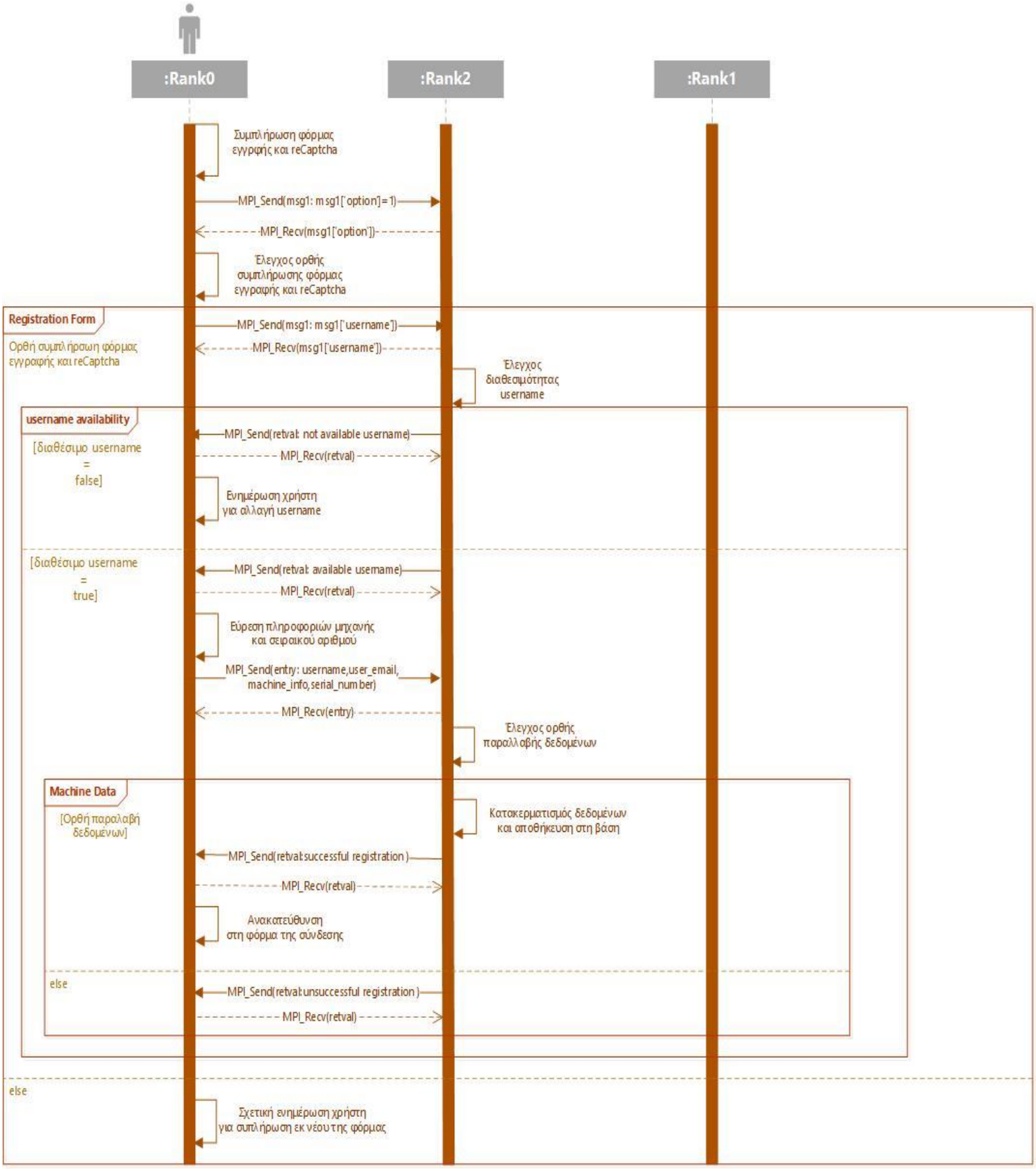
```
elif msg1['option']==2:
        msg2['OTP'] = generateOTP()
        msg2['len'] = len(msg2['OTP'])
        send_otp_viaEmail(msg1['email'], msg2['OTP'])
        comm.send(msg2['OTP'], dest=2, tag=12)
```

Όπως έχει γίνει ήδη αναφορά η διεργασία rank1 είναι υπεύθυνη για τη δημιουργία του 6ψήφιου OTP και της αποστολής του τόσο στο rank2 μέσω μηνύματος όσο και στον χρήστη μέσω email κατά τη διάρκεια της σύνδεσης του χρήστη. Η λειτουργία του είναι περιορισμένη αλλά παραμένει εξίσου σημαντική.

Αφού παρουσιάστηκε ο τρόπος με τον οποίο λειτουργεί το disCodeV2 θα παρατεθούν 2 διαγράμματα. Θα δοθεί έμφαση στον τρόπο που αλληλεπιδρούν οι διαφορετικές διεργασίες μεταξύ τους αποτυπώνοντας έτσι τον τρόπο που δουλεύει το MPI. Για τα διαγράμματα θα χρησιμοποιηθεί UML (Unified Modeling Language– Ενιαία γλώσσα προγραμματισμού) η οποία αποτελεί μια γραφική γλώσσα για την ανάλυση και τη σχεδίαση πληροφοριακών συστημάτων.

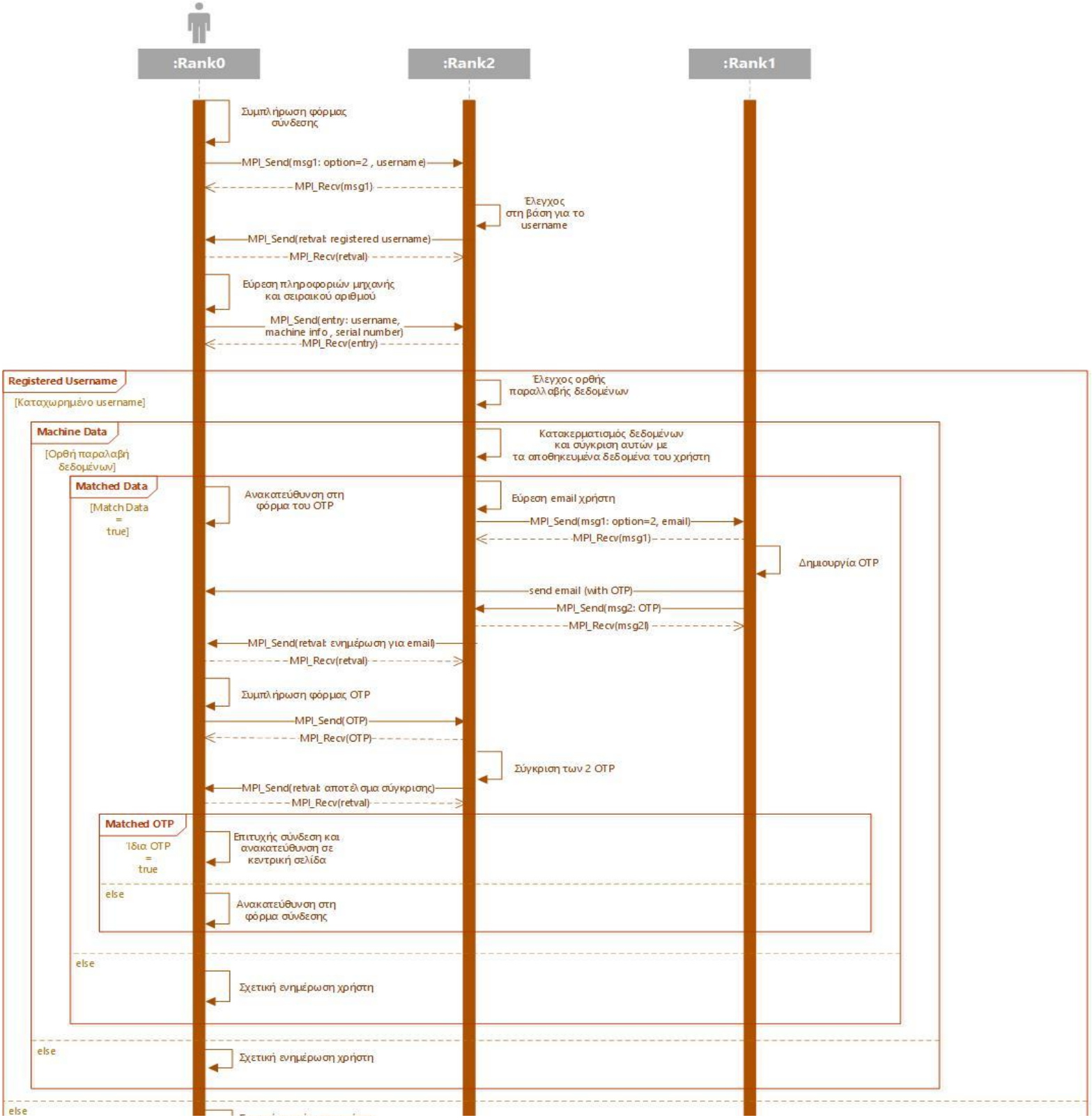
Από τους 12 διαφορετικούς τύπους διαγραμμάτων που ορίζονται θα χρησιμοποιηθεί ο τύπος των διαγραμμάτων ακολουθίας (Sequence diagrams)που ανήκουν στην κατηγορία των δυναμικών διαγραμμάτων συμπεριφοράς και πιο συγκεκριμένα στα διαγράμματα αλληλεπίδρασης (διαγράμματα ακολουθίας και διαγράμματα συνεργασίας) [36]. Η δομή που θα χρησιμοποιηθεί είναι η βασική (προτιμάται για την απεικόνιση του τρόπου με τον οποίο οι διεργασίες αλληλεπιδρούν μεταξύ τους)

Για τη διαδικασία της εγγραφής κατασκευάστηκε το παρακάτω διάγραμμα που αποτυπώνει τη λογική της εφαρμογής [Διάγραμμα 11].



Διάγραμμα 11. Διάγραμμα ακολουθίας της διαδικασίας εγγραφής

Για τη διαδικασία της σύνδεσης ακολουθεί το διάγραμμα το οποίο κατασκευάστηκε προκειμένου να αποτυπώσει συνοπτικά τη λογική της προσέγγισης που χρησιμοποιήθηκε [Διάγραμμα 12].



Διάγραμμα 12. Διάγραμμα ακολουθίας για τη διαδικασία της σύνδεσης

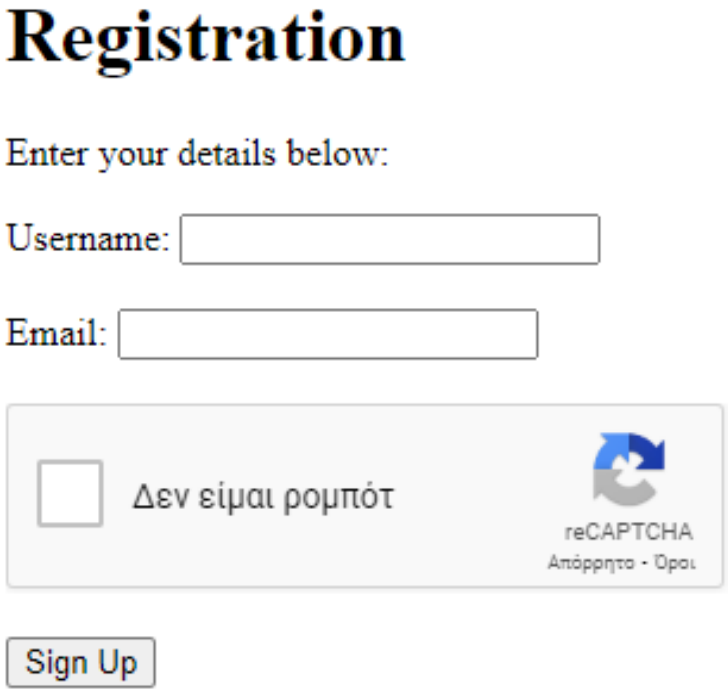
5.3 Οδηγός Χρήστη

Η υπό ενότητα αυτή θα επικεντρωθεί στον χρήστη, δηλαδή στις φόρμες που πρέπει συμπληρώσει ώστε να ταυτοποιηθεί αλλά και τα μηνύματα λάθους που εμφανίζονται στην οθόνη.

Η πρώτη σελίδα που βλέπει ο χρήστης είναι η σελίδα που τον καλωσορίζει και του δίνει τη δυνατότητα να επιλέξει μεταξύ της εγγραφής και της σύνδεσης. Αν

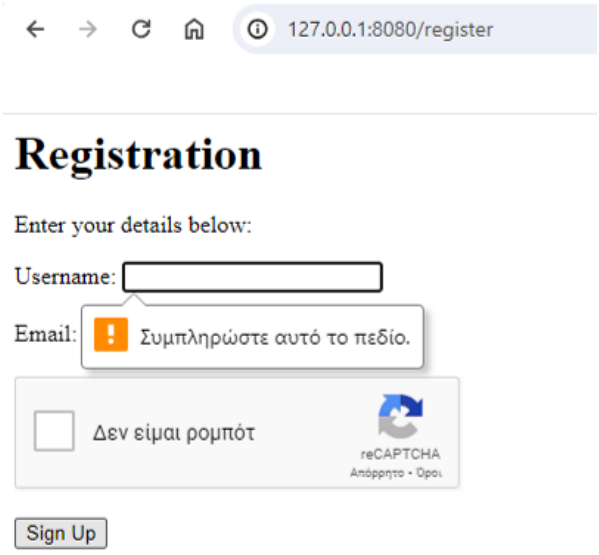
Εικόνα 5. Welcome page

Αν ο χρήστης επιλέξει την εγγραφή τότε ανακατευθύνεται στην αντίστοιχη σελίδα:



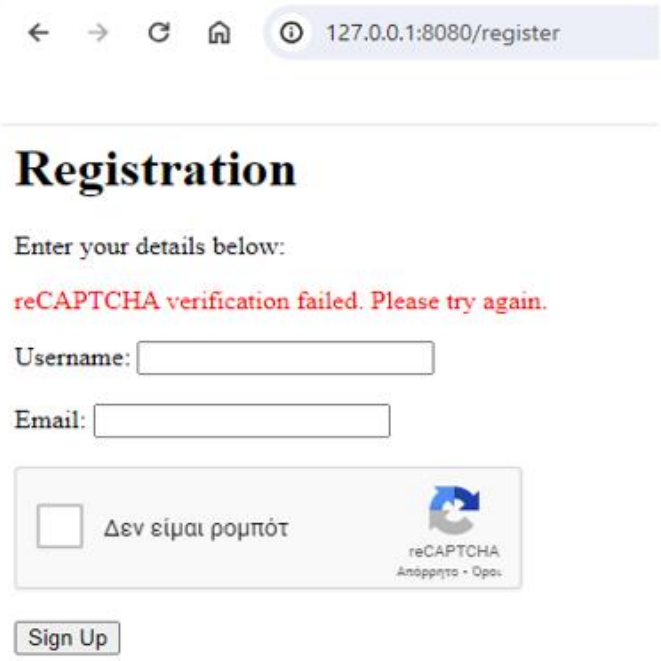
Εικόνα 6. Registration form

Ο χρήστης θα πρέπει να πληκτρολογήσει το username και το email του. Τα πεδία αυτά είναι υποχρεωτικά και επομένως αν δε συμπληρώσει κάποιο από τα δύο εμφανίζεται το παρακάτω μήνυμα.



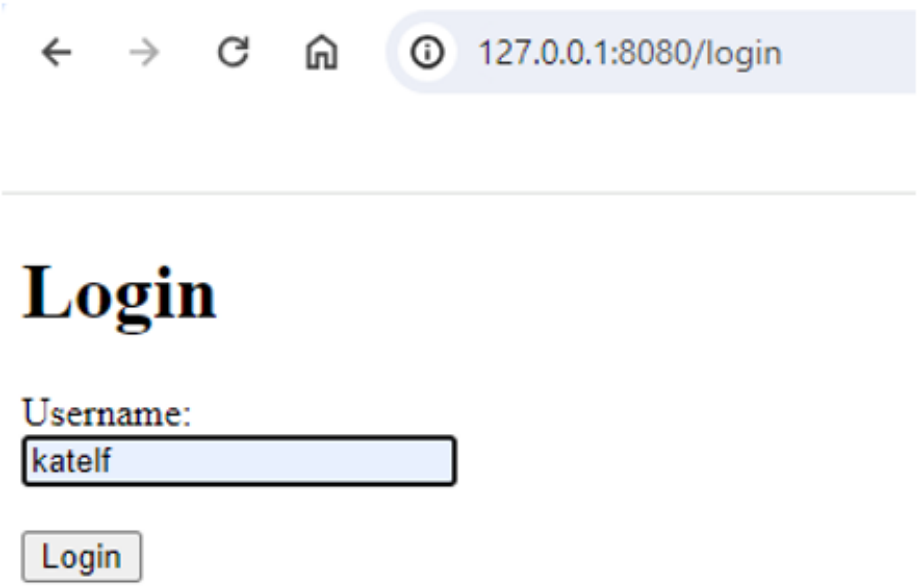
Εικόνα 7. Ελλειπή στοιχεία στη φόρμα εγγραφής

Με την ίδια λογική αν δεν επιλέξει το «Δεν είμαι Ρομπότ» και πατήσει το signup δεν ολοκληρώνεται η εγγραφή και εμφανίζεται αντίστοιχο μήνυμα.



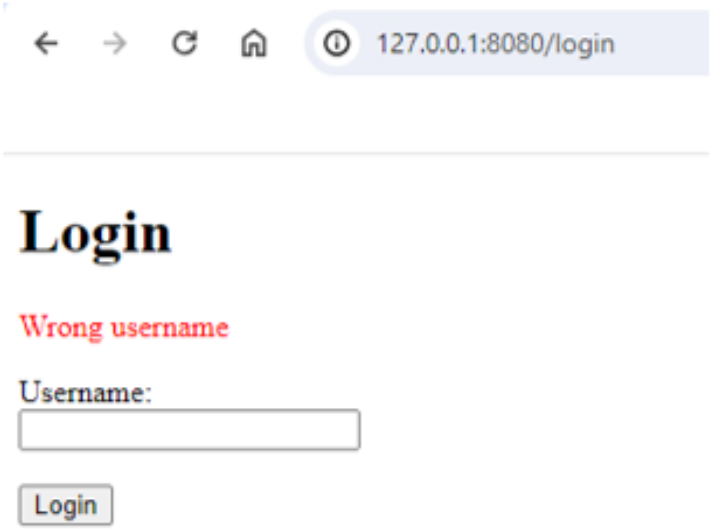
Εικόνα 8. Φόρμα εγγραφής και "I'm not a robot"

Αν ο χρήστης συμπληρώσει τη φόρμα ορθά τότε ανακατευθύνεται στη σελίδα της σύνδεσης.



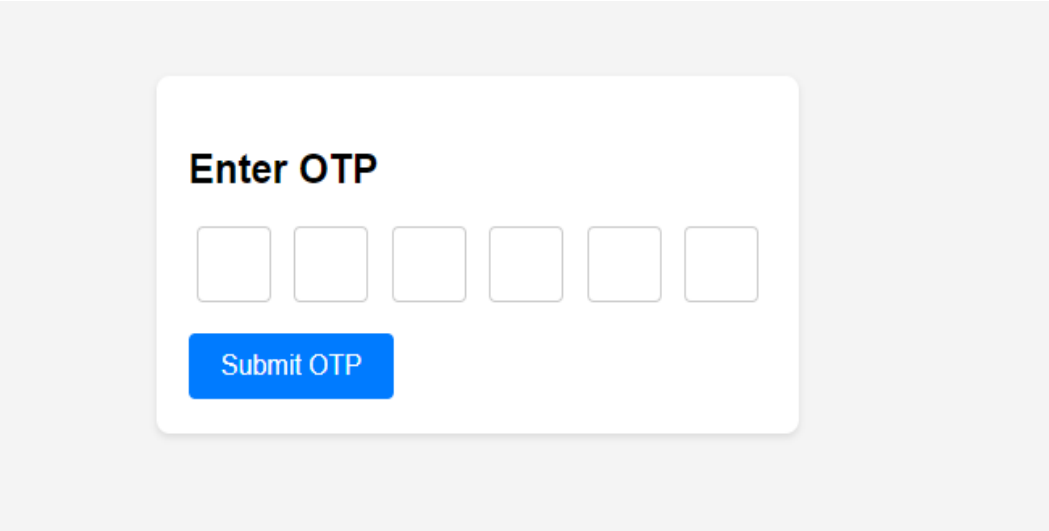
Εικόνα 9.Φόρμα σύνδεσης

Αν ο χρήστης πληκτρολογήσει λάθος username τότε εμφανίζεται σχετικό μήνυμα:



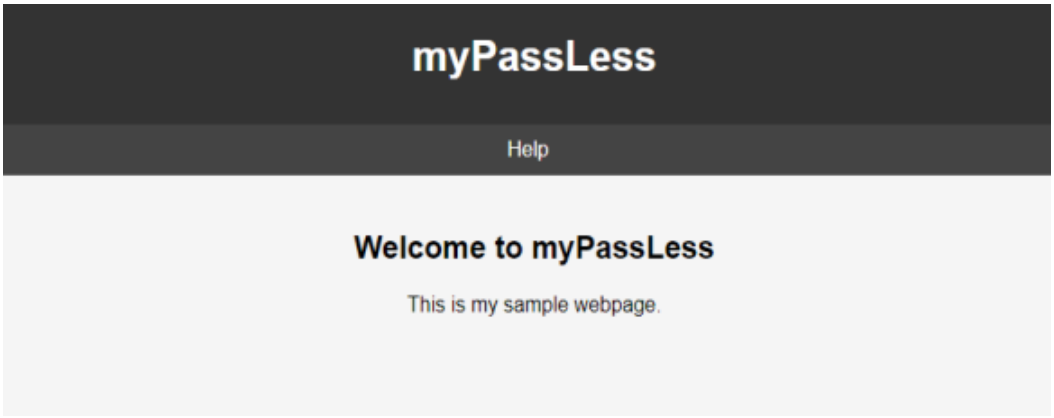
Εικόνα 10.Μη εγγεγραμμένο username

Αλλιώς ο χρήστης ανακατευθύνεται στη φόρμα συμπλήρωσης του OTP



Εικόνα 11.OTP form page

Αν το OTP είναι λανθασμένο τότε ο χρήστης ανακατευθύνεται στη σελίδα σύνδεσης και προσπαθεί εκ νέου να συνδεθεί. Σε αντίθετη περίπτωση ο χρήστης έχει ταυτοποιηθεί επιτυχώς και ανακατευθύνεται στην κύρια σελίδα.



Εικόνα 12. Main page – successful authentication

Τώρα αν ο χρήστης επιλέξει κατευθείαν από την αρχική σελίδα την επιλογή της σύνδεσης η διαδικασία που ακολουθείται είναι η ίδια (φόρμα σύνδεσης -> φόρμα OTP -> Κύρια σελίδα MyPassLess)

ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

5.1 Προοπτικές disCodeV2

Στην ενότητα αυτή θα παρουσιαστούν κάποιες παρατηρήσεις και βελτιώσεις σχετικά με το disCodeV2 οι οποίες ουσιαστικά αποτελούν και τις προοπτικές της εφαρμογής. Όπως παρουσιάστηκε ήδη το disCodeV2 αποτελεί μια μέθοδο αυθεντικοποίησης χρήστη χωρίς τη χρήση συνθηματικών. Βασική σκέψη είναι η χρήση κάποιων πληροφοριών μηχανής καθώς επίσης και του σειριακού αριθμού της μητρικής, δηλαδή δεδομένα που στο σύνολό τους αποδίδουν μια μοναδική ταυτότητα στο μηχάνημα του χρήστη. Φυσικά η μέθοδος αυτή έχει και θετικές πτυχές αλλά και αρνητικές.

Με αυτή τη μέθοδο ταυτοποίησης σαφώς η διαδικασία είναι ευκολότερη από την πλευρά του χρήστη καθώς δε χρειάζεται να αναλάβει να σχηματίσει κάποιο ισχυρό συνθηματικό και στην πορεία να το θυμάται. Η επιλογή αδύναμων συνθηματικών έχει αποδειχτεί ότι δεν είναι ασφαλής για πάρα πολλούς λόγους που έχουν αναλυθεί σε προηγούμενα κεφάλαια.

Πέρα από τα παραδοσιακά συνθηματικά η μέθοδος αυτή σαφώς αυξάνει το επίπεδο ασφαλείας καθώς είναι δύσκολο κάποιος να τα αναπαράγει και να αποκτήσει μη εξουσιοδοτημένη πρόσβαση. Φυσικά αυτό υποδηλώνει πως η δεν είναι επιρρεπείς σε διάφορες τεχνικές παραπλάνησης ή πλαστοπροσωπίας συγκριτικά με τη μέθοδο των παραδοσιακών συνθηματικών.

Βέβαια σε αυτού του είδους τη μέθοδο είναι απαραίτητο το επίπεδο της ασφάλειας να είναι υψηλό. Για αυτό το λόγο προστέθηκε η τεχνική αποστολής OTP αλλά και ο κατακερματισμός των δεδομένων που εισάγονται στον πίνακα MACHINES. Μία σκέψη ήταν όλα τα δεδομένα να κρυπτογραφηθούν και να περαστούν σε μια στήλη του πίνακα (έτσι αντί για 8 πεδία , ο πίνακας θα είχε 2 πεδία). Κατά αυτόν τον τρόπο ένας πιθανός επιτιθέμενος με πρόσβαση στη βάση θα έπρεπε να βρει με ποια σειρά αποθηκεύονται τα δεδομένα. Ωστόσο, η ιδέα αυτή απορρίφθηκε γιατί η προσέγγιση αυτή θα δυσκολέψει τους προγραμματιστές ως ένα βαθμό καθώς θα πρέπει να βρουν τα αρχικά δεδομένα αλλά και να προσδιορίσουν ποιο τμήμα του πεδίου αυτού αντιστοιχεί στον εκάστοτε τύπο (σειριακός αριθμός, πλατφόρμα, επεξεργαστής κλπ).

Ωστόσο, αυτές οι πληροφορίες στην πραγματικότητα είναι απόρρητες και η χρήση αυτών μπορεί να εγείρει ανησυχίες στους χρήστες αλλά και κινδύνους. Πρωταρχική ανησυχία είναι η πρόσβαση σε ευαίσθητα δεδομένα του συστήματος και η αποθήκευση αυτών. Αν διαρρεύσουν αυτά τα δεδομένα ο αντίκτυπος θα είναι πολύ σοβαρότερος εν γένει από τη διαρροή ενός συνθηματικού καθώς στη δεύτερη περίπτωση το πρόβλημα περιορίζεται μόνο στον εκάστοτε λογαριασμό ο οποίος μπορεί και να μην εκθέσει ευαίσθητες πληροφορίες για τον χρήστη. Στην πρώτη περίπτωση όμως εκτίθενται, άσχετα με τις πληροφορίες που θα βρει ο επιτιθέμενος αποκτώντας μη εξουσιοδοτημένη πρόσβαση σε λογαριασμό του χρήστη, ήδη από το αρχικό σημείο σημαντικά δεδομένα για τη μηχανή.

Ένας επιτιθέμενος μπορεί να υποκλέψει με ποικίλους τρόπους τον σειριακό αριθμό της μητρικής αλλά και άλλων στοιχείων της μηχανής.

1. Ένα απλό παράδειγμα είναι η χρήση κακόβουλου λογισμικού: Ο εισβολές ενδέχεται να χρησιμοποιήσει κακόβουλο λογισμικό ώστε να συλλέξουν πληροφορίες του συστήματος.
2. Ευπάθειες λογισμικού: Προφανώς και ένας εισβολέας θα χρησιμοποιήσει προς όφελος του τυχόν τρωτά σημεία ενός λογισμικού ή λειτουργικού συστήματος προκειμένου να αποκτήσει πληροφορίες για τη μηχανή όπως είναι και ο σειριακός αριθμός.
3. Αποκάλυψη από τους ίδιους τους χρήστες: Είναι πολύ πιθανό να χρησιμοποιηθούν τεχνικές οι οποίες θα εξαπατήσουν τους χρήστες ώστε να αποκαλύψουν οι ίδιοι δεδομένα του συστήματός του όπως είναι και ο σειριακός αριθμός μηχανής.
4. Φυσική πρόσβαση: Φυσικά αν ο εισβολέας έχει φυσική πρόσβαση στη μηχανή δε χρειάζεται να γνωρίζει καθόλου τον αριθμό του σειριακού αριθμού. Έχοντας ανακαλύψει το username του χρήστη απλώς θα συνδεθεί στην υπηρεσία χωρίς ιδιαίτερη δυσκολία.
5. Πρόσβαση στην ίδια τη βάση: Σε αυτή την περίπτωση η πρόσβαση στη βάση δίνει τη δυνατότητα στον εισβολέα να αποκτήσει όχι μόνο πληροφορίες για το σύστημα ενός χρήστη αλλά για όλους τους εγγεγραμμένους.

Πως μπορούν όμως όλα αυτά να αποφευχθούν;

1. Πρόληψη κακόβουλου λογισμικού: Η εγκατάσταση αξιόπιστου λογισμικού για την προστασία από ιούς αλλά και κακόβουλα λογισμικά σε όλες τις συσκευές θα πρέπει να αποτελέσει βασική προϋπόθεση για τους χρήστες. Είναι απαραίτητη η ενημέρωση και όχι ο εφησυχασμός. Εργαλεία όπως η σάρωση των συσκευών για κακόβουλα λογισμικά ή ύποπτες δραστηριότητες είναι επίσης μια λύση.
2. Ισχυροί έλεγχοι για ευπάθειες σε λογισμικό: Πρέπει να ακολουθούνται όλα τα πρωτόκολλα ασφαλείας. Είναι σημαντικό οι προγραμματιστές να εντοπίζουν και να επιδιορθώνουν άμεσα τυχόν ευπάθειες που έχουν προκύψει στο λογισμικό πριν αυτό χρησιμοποιηθεί από τους χρήστες. Συστηματικοί και πολλοί έλεγχοι ώστε οι ενημερώσεις να εφαρμόζονται έγκαιρα στο λογισμικό.
3. Αφύπνιση χρηστών: Οι χρήστες θα πρέπει να είναι πιο καχύποπτοι απέναντι σε ενέργειες που τους ζητούν να αποκαλύψουν τέτοιες πληροφορίες. Η αποφυγή των μηνυμάτων ηλεκτρονικού ψαρέματος, κλήσεις ή προσπάθειες πρόσβασης σε ψευδείς ιστότοπους ή απομίμησης των πραγματικών οι οποίες με λίγη προσοχή γίνονται αντιληπτές (είναι κακοφτιαγμένες σε σχέση με τις πρωτότυπες - πραγματικές).
4. Αποφυγή φυσικής πρόσβασης από εισβολείς: Βασική προϋπόθεση είναι ο χρήστης να μην επιτρέπει σε άτομα στα οποία δεν έχει καθόλου εμπιστοσύνη να χρησιμοποιήσουν τη μηχανή τους. Αν παρόλα αυτά κάποιος όντως χρησιμοποιήσει τη συσκευή αυτή καθ' αυτή καλό είναι να υπάρχει τουλάχιστον έλεγχος πολλαπλών παραγόντων ώστε να ενημερωθεί ο χρήστης για την προσπάθεια απόκτησης μη εξουσιοδοτημένης πρόσβασης

και να αποτρέψει την είσοδο κάνοντας κάποια αναφορά πως δεν είναι ο ίδιος που δοκιμάζει να συνδεθεί στην εκάστοτε υπηρεσία.

5. Πρόληψη παραβίασης δεδομένων σε βάσεις: Η εφαρμογή ισχυρών μέτρων ασφάλειας είναι άκρως απαραίτητη. Τείχη προστασίας , κάποιο σύστημα ανίχνευσης και αποτροπής εισβολών στο σύστημα είναι μερικοί τρόποι για να προστατευτεί η βάση. Βέβαια και τα δεδομένα που είναι αποθηκευμένα στη βάση θα πρέπει να είναι κρυπτογραφημένα , κατακερματισμένα για επιπλέον ασφάλεια. Οι ύποπτες δραστηριότητες και η μη εξουσιοδοτημένη πρόσβαση στο σύστημα θα πρέπει να γίνονται αντιληπτές γρήγορα και να αποτρέπονται.

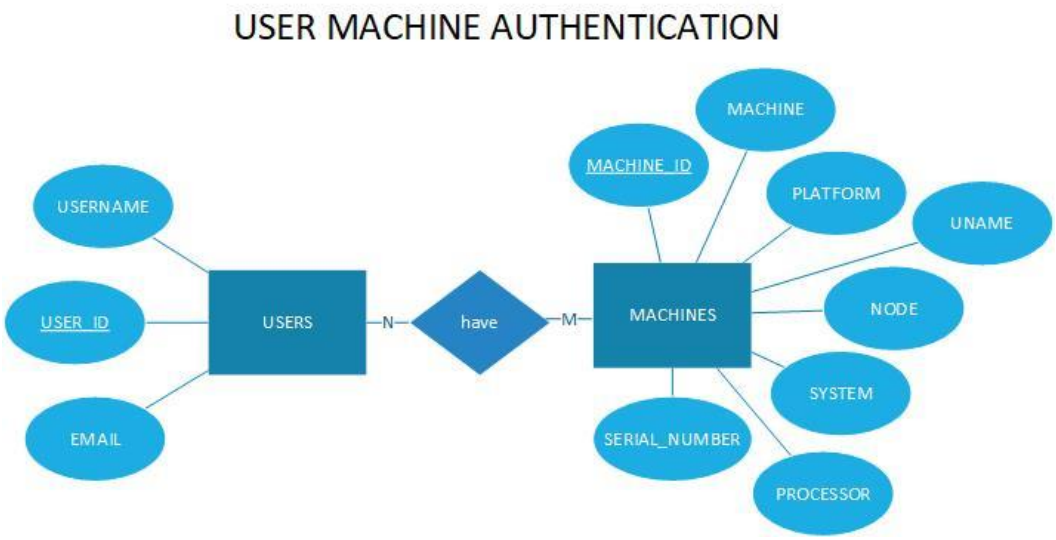
Τουλάχιστον οι ισχυροί έλεγχοι για την εύρεση ευπαθειών, η χρήση ελέγχουν 2 ή πολλαπλών παραγόντων αλλά και η αποφυγή παραβίασης της ίδιας της βάσης δεδομένων είναι κάτι που μπορούν να προβλέψουν και να αποτρέψουν οι κατασκευαστές προγραμματιστές της εφαρμογής. Ως ένα βαθμό το disCodeV2 έχει προβλέψει για όλα αυτά χρησιμοποιώντας έλεγχο 2 παραγόντων (OTP) αλλά και αποθήκευση δεδομένων χρησιμοποιώντας κρυπτογραφικό αλγόριθμο.

Εκτός όμως από τον κίνδυνο έκθεσης αυτών των πληροφοριών για τον οποίο δόθηκαν λύσεις υπάρχουν και άλλα βασικά προβλήματα. Η ενσωματωμένη βάση στο projectαλλά και η προσθήκη συσκευών κάτω από τον ίδιο χρήστη ή η αλλαγή της μηχανής μετά από κάποια ενημέρωση του συστήματος. Η βάση προς το παρόν και λόγο ευκολίας έχει ενσωματωθεί μέσα στο περιβάλλον του PyCharm. Αυτό προφανώς δεν είναι λειτουργικό για μια εφαρμογή. Η καλύτερη προσέγγιση θα ήταν να βρίσκεται κάπου στο cloud, όπου θα είναι πιο ασφαλής και προστατευμένη. Από την άλλη πλευρά ο χρήστης μπορεί να χρειαστεί να κάνει κάποια ενημέρωση στη συσκευή του και επομένως να αλλάξει κάποια πληροφορία της μηχανής. Φυσικά θα μπορούσε να προβλεφθεί σε αυτές τις περιπτώσεις να ενημερώνεται και το disCodeV2. Σημαντικό είναι και το ζήτημα προσθήκης επιπλέον μηχανής στον εκάστοτε χρήστη. Ουσιαστικά κατά τη διαδικασία της εγγραφής στην εφαρμογή χρησιμοποιώντας μια μηχανή αυτόματα προκαλείται μια εξάρτηση από το ίδιο το μηχάνημα. Ένας χρήστης έχοντας εγγραφεί από το μηχάνημα X δε θα μπορεί να συνδεθεί χρησιμοποιώντας το μηχάνημα Y.

Αυτό το πρόβλημα όμως μπορεί να επιλυθεί αν η εφαρμογή επιτρέπει στον χρήστη να προσθέσει και άλλη μηχανή (άρα για το ίδιο username να υπάρχουν παραπάνω από μία μηχανές). Αυτή τη στιγμή η εφαρμογή λειτουργεί με την αποδοχή πως usernameμπορεί να αντιστοιχηθεί μόνο με μία μηχανή, αλλά μία μηχανή μπορεί να αντιστοιχηθεί με περισσότερα username. Σύμφωνα με αυτή την αποδοχή θα πρέπει να αλλάξει η προσέγγιση της εφαρμογής και για παράδειγμα στην αρχική σελίδα να προστεθεί άλλη μια επιλογή, εκείνη της «προσθήκης μηχανής».

Η επιλογή αυτή θα επιτρέπει στον χρήστη να χρησιμοποιήσει το ήδη εγγεγραμμένο usernametου και να σταλεί στο αποθηκευμένο του email ένα μήνυμα στο οποίο ζητά να επιβεβαιωθεί η προσθήκη της μηχανής που προσπαθεί να εγγραφεί. Αν η απάντηση είναι θετική τότε η διαδικασία θα είναι παρόμοια με της εγγραφής απλώς δε θα χρειαστεί να δοθεί εκ νέου το email ξανά.

Η βάση πλέον θα με αναπαράσταση erd κατά Chen θα είναι πλέον όπως στο παρακάτω διάγραμμα [Διάγραμμα 13]

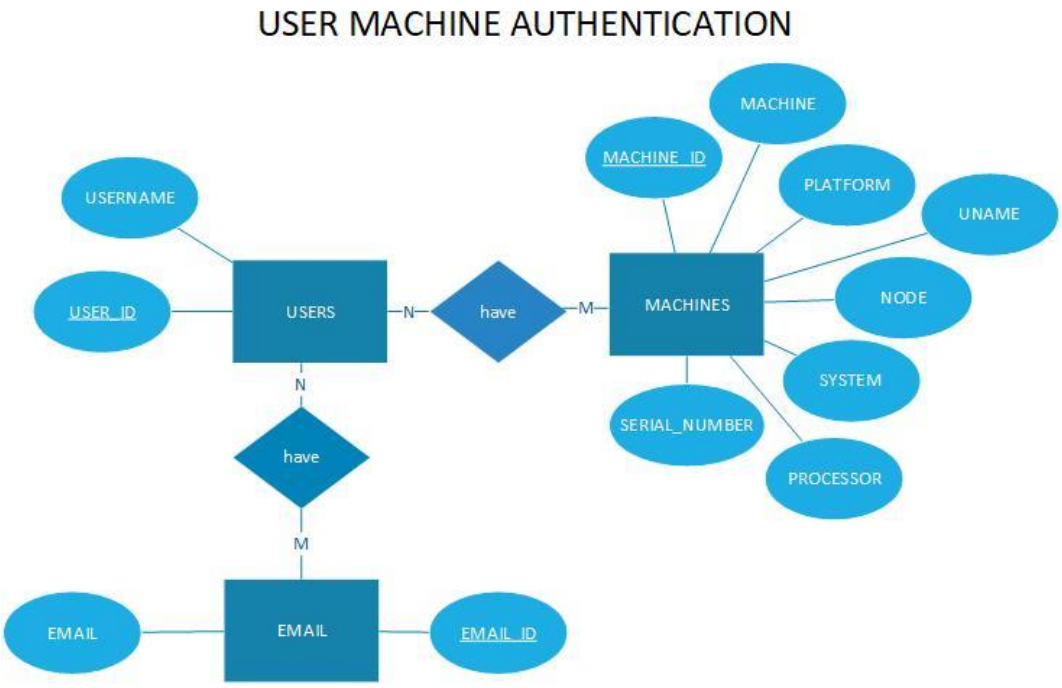


Διάγραμμα 13.ERD όταν η σχέση χρήστη μηχανής είναι N-M

Αυτομάτως όταν σχέση μεταξύ 2 πινάκων είναι πολλά προς πολλά θα πρέπει να δημιουργηθεί ένας ενδιάμεσος πίνακας ο οποίος θα περιλαμβάνει αντιστοιχισμένα τα πρωτεύοντα κλειδιά.

Ωστόσο, χωρίς να αλλάξει η δομή της βάσης, δηλαδή σε έναν χρήστη να αντιστοιχεί 1 μηχάνημα, το disCodeV2 θα μπορούσε να χρησιμοποιηθεί εντός εταιρειών (για παράδειγμα τράπεζα) όπου οι υπάλληλοι έχουν ένα συγκεκριμένο μηχάνημα το οποίο διαχειρίζονται και έτσι δε χρειάζεται να γίνει κάποια προσθήκη επιπλέον συσκευής.

Άλλη μία λειτουργία που θα μπορούσε να προστεθεί είναι η επεξεργασία της διεύθυνσης ηλεκτρονικού ταχυδρομείου. Δηλαδή να είναι σε θέση να την αλλάξει εντελώς ή να τη διορθώσει. Ακόμα θα μπορούσε να προχωρήσει ένα βήμα παρακάτω και να έχει τη δυνατότητα να προσθέσει μια λίστα με emails μια αλλαγή που , όπως και στην «προσθήκη μηχανής» θα μεταβάλει τη δομή της βάσης. Επομένως αν πραγματοποιηθεί αυτή η αλλαγή, το erd κατά Chen θα μεταβληθεί όπως στο παρακάτω διάγραμμα [Διάγραμμα 14].



Διάγραμμα 14. ERD όταν ο χρήστης έχει περισσότερα από 1 email

Ένα ακόμα βασικό πρόβλημα είναι η περιορισμένη ευελιξία της μεθόδου καθώς οι εντολές που αντλούν δεδομένα από τη μηχανή δεν είναι ίδιες. Επομένως αναλόγως την πλατφόρμα που χρησιμοποιεί κάθε μηχάνημα θα πρέπει να έχει αποδοθεί στην κατάλληλη συνάρτηση η σωστή εντολή ώστε να «τραβήξει» τα δεδομένα. Αυτό σημαίνει ότι θα πρέπει να έχουν προβλεφθεί όλα τα πιθανά λειτουργικά συστήματα που θα μπορούσε να χρησιμοποιήσει ο χρήστης και να γίνουν δοκιμές σε όλες. Το disCodeV2 έχει δοκιμαστεί μόνο σε περιβάλλον Windows , άρα είναι πολύ πιθανό σε άλλο περιβάλλον να υπάρχουν ζητήματα ή νέες προκλήσεις τα οποία θα πρέπει φυσικά να λυθούν. Ωστόσο η κλάση platformμπορεί να λειτουργεί σε διαφορετικά λειτουργικά συστήματα, άρα δε θα υπάρξει πρόβλημα ως προς τη συλλογή των δεδομένων από τη μηχανή.

Σε αυτό το σημείο και καθώς έχουν αναλυθεί πιθανές βελτιώσεις του disCodeV2 κρίνεται σημαντικό να αναφερθεί πως το MPI δίνει τη δυνατότητα διαμόρφωσης ενός κατανεμημένου συστήματος. Αυτό συμβαίνει γιατί η συνολική διαδικασία μπορεί να διαχωριστεί σε μικρότερες διεργασίες κάθε μία από τις οποίες μπορεί να λειτουργεί σε διαφορετική συσκευή ή εξυπηρετητή και να επιτευχθεί επικοινωνία μεταξύ τους. Κατ' επέκταση η διαδικασία εγγραφής και της σύνδεσης θα μπορούσε να τρέχει σε διαφορετικό εξυπηρετητή.

5.2 Συμπεράσματα

Όπως έχει γίνει ήδη αντιληπτό και από τον τίτλο της διπλωματικής εργασίας, η παραδοσιακή μέθοδος αυθεντικοποίησης χρήστη χρησιμοποιώντας συνθηματικά ενέχει πολλούς κινδύνους εάν το συνθηματικό δεν είναι ισχυρό. Έγιναν αναφορές σε πιθανές επιθέσεις όπως εξαντλητικής αναζήτησης, καταγραφείς πληκτρολογίου ή

ακόμα και επιθέσεις με πίνακα «ουράνιου τόξου». Αναφέρθηκαν επίσης και οι πιθανότητες όπου η κλοπή συνθηματικών πραγματοποιήθηκε από αμέλεια του χρήστη (αποκάλυψε ο ίδιος το συνθηματικό είτε λεκτικά είτε πληκτρολογώντας το συνθηματικό σε ψευδείς ιστότοπους είτε emails ή αποκαλύφθηκε κατόπιν φυσικής παρακολούθησης του χρήστη).

Φυσικά η μέθοδος ταυτοποίησης με τη χρήση συνθηματικών έχει ξεκινήσει και αντικαθίσταται από άλλες μεθόδους όχι μόνο λόγω της ευπάθειας των αδύναμων κωδικών απέναντι σε διάφορες επιθέσεις. Δεν είναι όλοι οι χρήστες που χρησιμοποιούν αδύναμους κωδικούς, κάποιοι επιλέγουν να σχηματίσουν ισχυρούς κωδικούς γιατί θέλουν να αποτρέψουν τη μη εξουσιοδοτημένη πρόσβαση σε λογαριασμούς τους. Άρα χρησιμοποιούν πολύπλοκά και διαφορετικά συνθηματικά στους λογαριασμούς του.

Που είναι το πρόβλημα όμως σε αυτό; Ακριβώς επειδή σε καθημερινή βάση ένας χρήστης είτε θα συνδεθεί σε μια πληθώρα λογαριασμών είτε θα εγγραφεί σε νέες υπηρεσίες ο όγκος των διαφορετικών συνθηματικών αποτελεί ένα βάρος στον ίδιο τον χρήστη. Δεν είναι εύκολο να απομνημονευτούν τόσα πολλά συνθηματικά και κάπως έτσι ξεκινάνε τα λάθη (είτε διατηρούνται σημειώσεις με αυτά είτε επαναχρησιμοποιούνται ξανά και ξανά τα ίδια συνθηματικά είτε επιλέγονται πολύ απλά συνθηματικά τα οποία είναι και ευάλωτα σε επιθέσεις).

Επομένως οι μέθοδοι που δε χρησιμοποιούν συνθηματικά ως μέσο αυθεντικοποίησης και έχουν υιοθετηθεί από ένα τεράστιο κομμάτι του πληθυσμού έχουν ως βασικό σκοπό την ασφάλεια των δεδομένων του χρήστη καθώς επίσης προσφέρουν ευκολία στον χρήστη αφού δεν καλείται να θυμάται τίποτε άλλο εκτός από το username του για τον εκάστοτε λογαριασμό. Οι πιο διαδεδομένοι τρόποι αυθεντικοποίησης χρήστη που αφήνουν στην άκρη την παραδοσιακή τεχνική είναι οι βιομετρικές μέθοδοι και οι έλεγχοι παραγόντων (είτε είναι δύο είτε είναι πολλαπλών). Τόσο οι βιομετρικές μέθοδοι όσο και οι έλεγχοι παραγόντων προσφέρουν σαφώς υψηλότερη ασφάλεια σε σχέση με τα συνθηματικά. Η κάθε μία μέθοδος με τα δικά της πλεονεκτήματα και μειονεκτήματα.

Η πρώτη από τη μία πλευρά χρησιμοποιεί τα βιομετρικά χαρακτηριστικά είτε αυτά είναι χαρακτηριστικά συμπεριφοράς (βάδιση, γραφικός χαρακτήρας κλπ) είτε είναι φυσικά χαρακτηριστικά (ίριδα, δακτυλικά αποτυπώματα κλπ) με τα τελευταία να είναι πιο δημοφιλή καθώς δεν επηρεάζονται από την ψυχολογική κατάσταση του χρήστη. Ενώ η δεύτερη μέθοδος με τους ελέγχους πολλαπλών παραγόντων είναι πιο συμβατή με τον παραδοσιακό τρόπο ταυτοποίησης, με τη χρήση συνθηματικών, αφού σε πολλές περιπτώσεις δρα επικουρικά και προσθέτει επίπεδο ασφαλείας προκειμένου να καλύψει τα τρωτά σημεία τους.

Ωστόσο, και άλλες μέθοδοι όπως η ταυτοποίηση μέσω κάποια υλικής συσκευής όπως τα USBχρησιμοποιούνται συχνά, ή οι έξυπνες κάρτες. Και στις δύο περιπτώσεις όμως υπάρχει το πρόβλημα της φορητότητας καθώς είναι κατασκευασμένα κατά κύριο λόγο για χρήση σε φορητούς ή σταθερούς υπολογιστές χωρίς αυτό βέβαια να σημαίνει ότι δε μπορούν να χρησιμοποιηθούν για παράδειγμα σε κινητά. Απλώς στις κινητές συσκευές πρέπει να υπάρχει η απαραίτητη υποστήριξη.

Σε αυτό το σημείο δεν πρέπει να παραληφθεί και το disCodeV2 το οποίο έχει αρκετές προοπτικές και με τις ανάλογες βελτιώσεις θα μπορούσε να χρησιμοποιηθεί ως μια νέα τεχνική. Σε σχέση με τις πιο διαδεδομένες μεθόδους ήδη χρησιμοποιεί τη μία (OTP) ενώ δεν απαιτεί τα επιπλέον εργαλεία (αισθητήρες) που είναι απαραίτητοι για τις βιομετρικές μεθόδους. Είναι βασικό ένας χρήστης να μπορεί χωρίς να χρειαστεί να αγοράσει νέο εξοπλισμό ή συσκευή να μπορεί να χρησιμοποιήσει άμεσα τη μέθοδο αυθεντικοποίησης.

Τι ζητάνε όμως οι χρήστες;

1. Ασφάλεια
2. Αξιοπιστία
3. Ακριβή αποτελέσματα
4. Χαμηλό κόστος
5. Ταχύτητα
6. Ευκολία στη χρήση
7. Ευκολία στη μεταφορά

Παρακάτω ακολουθεί ένας πίνακας στον οποίο γίνονται συγκρίσεις μεταξύ των μεθόδων που έχουν παρουσιαστεί σε προηγούμενα κεφάλαια σε σχέση με τα 7 χαρακτηριστικά που είναι επιθυμητά σε μια μέθοδο αυθεντικοποίησης [Πίνακας 6].

Μέθοδος Αυθεντικοποίησης	Ασφάλεια	Αξιοπιστία	Ακρίβεια	Κόστος	Ταχύτητα	Ευκολία στη Χρήση	Ευκολία στη μεταφορά
Συνθηματικά	Χαμηλή	Χαμηλή	Υψηλή	Χαμηλό	Υψηλή	Υψηλή	Υψηλή
Έξυπνες κάρτες	Υψηλή	Υψηλή	Υψηλή	Υψηλό	Υψηλή	Μέτρια	Χαμηλή
USB token	Υψηλή	Υψηλή	Υψηλή	Υψηλό	Υψηλή	Μέτρια	Χαμηλή
OTP	Υψηλή	Μέτρια	Υψηλή	Χαμηλό	Υψηλή	Υψηλή	Υψηλή
OpenID	Υψηλή	Υψηλή	Υψηλή	Χαμηλό	Υψηλή	Υψηλή	Υψηλή
Βιομετρική μέθοδος	Υψηλή	Υψηλή	Υψηλή	Μέτριο	Μέτρια	Μέτρια	Μέτρια
disCodeV2	Υψηλή	Υψηλή	Υψηλή	Χαμηλό	Υψηλή	Υψηλή	Μέτρια

Πίνακας 9. Κριτήρια αξιολόγησης μεθόδων ταυτοποίησης

Όπως φαίνεται και στον [Πίνακα 6] 3 επικρατέστερες μέθοδοι αυθεντικοποίησης χρήστη είναι το OpenID, οι βιομετρικές μέθοδοι και το disCodeV2. Σχετικά με τις βιομετρικές μεθόδους έχει ειπωθεί ότι εκείνες που βασίζονται σε φυσικά χαρακτηριστικά είναι πιο ακριβείς έναντι εκείνων που βασίζονται σε χαρακτηριστικά συμπεριφοράς (behavioral characteristics). Το OpenID φαίνεται να έχει τις καλύτερες επιδόσεις ενώ δεύτερο έρχεται το disCodeV2.

Γιατί όμως οι βιομετρικές μέθοδοι δεν έχουν επίσης άριστες επιδόσεις σχετικά με τα συγκεκριμένα κριτήρια; Στις μεθόδους αυτές χρειάζεται απαραίτητος εξοπλισμός. Πιο συγκεκριμένα αισθητήρες. Οι κινητές συσκευές έχουν διάφορους αισθητήρες όπως αναγνώρισης προσώπου ή αναγνώρισης δακτυλικών αποτυπωμάτων. Ωστόσο, για τους σταθερούς ή φορητούς υπολογιστές δεν ισχύει το ίδιο. Οι παλαιότεροι φορητοί υπολογιστές δε διαθέτουν κανέναν αισθητήρα, αντίθετα με νέα μοντέλα που έχουν εξωτερικά συνδεδεμένους αισθητήρες όπως για παράδειγμα αισθητήρας αναγνώρισης δακτυλικών αποτυπωμάτων.

Γενικά οι αισθητήρες των βιομετρικών χαρακτηριστικών διευκολύνουν στη φορτητότητα καθώς είναι ενσωματωμένοι στη συσκευή και διαθέτουν μεγαλύτερη ποικιλία συγκριτικά με τους υπολογιστές. Στους υπολογιστές όμως όπως έχει ήδη αναφερθεί κατά κύριο λόγο είναι εξωτερικοί και πολλές φορές όχι τοποθετημένοι πάνω στην επιφάνεια του μηχανήματος, με αποτέλεσμα να μην διευκολύνεται η μετακίνηση του χρήστη με τον φορητό υπολογιστή (καθώς θα πρέπει να τα μεταφέρουν και αυτά).

Οι βιομετρικοί αισθητήρες στους υπολογιστές δεν ποικίλουν καθώς εγκαθίστανται στο υλικό από τους κατασκευαστές (ανάλογα με τη λειτουργικότητα που θέλουν να αποκτήσει) σε αρχικό στάδιο και μετά οι χρήστες έχουν τη δυνατότητα να προσθέσουν οι ίδιοι κάποιους άλλους. Άρα οι πιθανότητες ένας χρήστης να χρειαστεί να αγοράσει εξοπλισμό προσπαθώντας να καλύψει την ανάγκη του είναι πολλές.

Οι αισθητήρες αναγνώρισης βιομετρικών χαρακτηριστικών όμως, στις κινητές συσκευές διαφέρουν και ως προς τον τρόπο χρήσης τους σε σχέση με εκείνους στους υπολογιστές (είτε είναι φορητοί είτε σταθεροί). Αυτό συμβαίνει γιατί οι αισθητήρες των βιομετρικών χρησιμοποιούν αλγόριθμους αναγνώρισης οι οποίοι είναι σχεδιασμένοι με τέτοιο τρόπο ώστε να αντιμετωπίζουν τους περιορισμένους πόρους αλλά και την ευαισθησία της ίδιας της συσκευής. Αντίθετα εκείνοι στους υπολογιστές είναι πιο εξελιγμένοι καθώς η ισχύς ενός υπολογιστή συγκριτικά με του κινητού είναι μεγαλύτερη όπως επίσης και ο αποθηκευτικός χώρος.

Οι επιδώσεις του OpenID σύμφωνα με τον πίνακα είναι άριστες. Ισχύει αυτό όμως στην πραγματικότητα; Όπως έχει γίνει σαφές, η ταυτοποίηση χρήστη χρησιμοποιώντας το πρωτόκολλο OpenID προϋποθέτει ο χρήστης να έχει εγγραφεί ήδη σε έναν πάροχο. Για να εγγραφεί σε αυτόν τον πάροχο έχει χρησιμοποιήσει κάποια μέθοδο αυθεντικοποίησης, ο οποίος μπορεί να συμπεριλαμβάνει για παράδειγμα έλεγχο πολλαπλών παραγόντων, βιομετρικούς ελέγχους ή ακόμα και τα μέσα κοινωνικής δικτύωσης.

Εστιάζοντας λοιπόν στη λογική ότι πολλοί είναι εκείνοι που επιλέγουν ως πάροχο είτε το λογαριασμό του περιηγητή τους είτε τους λογαριασμούς τους στα μέσα κοινωνικής δικτύωσης είναι σημαντικό να τονιστεί πως για την εγγραφή τους σε αυτά χρησιμοποιήθηκε ένα συνθηματικό. Αυτό σημαίνει πως οι ίδιοι οι πάροχοι είναι ευάλωτοι σε επιθέσεις που στοχεύουν τα συνθηματικά. Επομένως αυτή η εξάρτηση από τον πάροχο πιθανώς να ενέχει κινδύνους οι οποίοι δεν οφείλονται στο πρωτόκολλο αυτό καθ' αυτό αλλά στον τρόπο με τον οποίο δημιουργούνται οι λογαριασμοί στους παρόχους.

Αντίθετα με το OpenID, το disCodeV2 δεν βασίζεται σε παρόχους για την ταυτοποίηση του χρήστη. Άρα η όποια επίθεση ή κάποιο άλλο πρόβλημα που ενδέχεται να παρουσιαστεί σε αυτούς δεν επηρεάζει καθόλου τη μέθοδο. Είναι σημαντικό η οποιαδήποτε τεχνική ταυτοποίησης να μην επηρεάζεται από την πολιτική ασφαλείας των παρόχων (που αποτελούν εξωτερικό παράγοντα) ώστε οι όποιες αλλαγές και διορθώσεις σε διάφορα κομμάτια της τεχνικής να μπορούν να γίνουν άμεσα.

Το disCodeV2 (το οποίο χρησιμοποιεί και τη μέθοδο του OTP) φαίνεται να υστερεί μόνο ως προς το κομμάτι της εύκολης μεταφοράς βάσει του πίνακα. Προς το παρόν η εφαρμογή είναι καθαρά φτιαγμένη προκειμένου να λειτουργεί σε υπολογιστή. Αυτό σημαίνει πως στις κινητές συσκευές οι χρήστες θα αντιμετωπίσουν πρόβλημα. Ωστόσο, σύμφωνα με τις παρατηρήσεις και τις λύσεις που έχουν ήδη δοθεί, στην προηγούμενη ενότητα, αυτή η δυσκολία μπορεί να ξεπεραστεί δίνοντας τη δυνατότητα στους χρήστες να συνδέονται στους λογαριασμούς τους μέσω των κινητών συσκευών τους. Επομένως τείνει και αυτή η μέθοδος να αποκτήσει άριστες προδιαγραφές με κάποιες αλλαγές που δεν είναι ιδιαίτερα περίπλοκες.

Παρόλα αυτά όποιο μέσο ταυτοποίησης και να χρησιμοποιηθεί, οι κίνδυνοι δεν εξαλείφονται πλήρως. Υπάρχουν πάντα και πλεονεκτήματα και μειονεκτήματα. Σκοπός είναι η κάθε υπηρεσία - εφαρμογή να χρησιμοποιήσει μια μέθοδο που θα καλύπτει τις απαιτήσεις της και θα προσφέρει το καλύτερο δυνατό αποτέλεσμα βάσει κάποιον κριτηρίων. Φυσικά με την εξέλιξη της τεχνολογίας θα πρέπει να εξετάζονται οι πιθανοί κίνδυνοι ή πιθανά σενάρια βελτίωσης της μεθόδου.

ΠΑΡΑΡΤΗΜΑ Α’

Στο παράρτημα αυτό παρατίθεται ο συνολικός κώδικας ανάπτυξης της εφαρμογής.

mainSeralNumber.py

```
import app
import subprocess
import math
import random
import matplotlib.pyplot as plt
from mpi4py import MPI
import platform
import time
import sqlite3
import json
from myDatabase import *
from app import *
import sys
import hashlib
import hmac
from helpers import *
from configuratorStrings import getSecretKey

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

msg1 = {}
msg2 = {}
retval = {'flag': 0, 'message': ''}
users = []
entry = {'username': None, 'machineInfo': None, 'serialNumber': None, 'email':
None}

# Secret key for HMAC (keep this secure)
config_file_path = 'app.yaml'
# get secret key as bytes
secret_key = getSecretKey(config_file_path).encode('utf-8')

if __name__=="__main__":
    con = sqlite3.connect('myclients.db')  # Initialize the connection here
    create_tables_if_not_exist()
    print_table_raws()
    entry = {'username': None, 'info': None}
    flag = True
    motherboard_serial_number = get_motherboard_serial_number()
    if motherboard_serial_number:
```

```
        print("Motherboard Serial Number:", motherboard_serial_number)
    else:
        print("Failed to retrieve motherboard serial number.")
    if rank==0:  # client
        while True:
            # subprocess.Popen(['python', 'app.py'])
            openWenApi()

    if rank==2:
        while flag:
            msg1 = comm.recv(source=0, tag=1)
            if msg1 ['option']==0:
                close_connection(con)
                comm.send(msg1, dest=1, tag=11)
                flag = False
                exit()
            elif msg1 ['option']==1:  # register
                if check_username_existence(con, msg1 ['username']):
                    retval ['flag'] = 0
                    retval ['message'] = "Username already in use. Please choose a
different username."
                    comm.send(retval, dest=0, tag=2)
                else:  # the username can be used
                    retval ['flag'] = 1
                    retval ['message'] = "Available username. We will now fetch some
device information"
                    comm.send(retval, dest=0, tag=2)
                    # fetch machine info and serial number of motherboard
                    entry = comm.recv(source=0, tag=3)
                    # before insert plaintext data into tables it's a good addition to hash
them
                    # Serialize the tuple to a string
                    # Insert user information into the database
                    machine_info_tuple = entry ['machineInfo']
                    serial_number = entry ['serialNumber']

                    if (machine_info_tuple is None):
                        retval ['flag'] = 0
                        retval ['message'] = "Something went wrong with the Machine Info"
                        comm.send(retval, dest=0, tag=4)
                    if (serial_number is None):
                        retval ['flag'] = 0
                        retval ['message'] = "Something went wrong whit the serial number"
                        comm.send(retval, dest=0, tag=4)

                    else:
                        hashed_serialNumber = hmac.new(secret_key,
str(serial_number).encode('utf-8'),
                                                       hashlib.sha256).hexdigest()
                        uname_result = platform.uname()
```

```
                uname_string = f"system={uname_result.system},
node={uname_result.node}, release={uname_result.release},
version={uname_result.version}, machine={uname_result.machine}"
                hashed_machine_info = {
                    'machine': hmac.new(secret_key, str(machine_info_tuple
[0]).encode('utf-8'),
                                        hashlib.sha256).hexdigest(),
                    'platform': hmac.new(secret_key, str(machine_info_tuple
[1]).encode('utf-8'),
                                         hashlib.sha256).hexdigest(),
                    'uname': hmac.new(secret_key, uname_string.encode('utf-8'),
hashlib.sha256).hexdigest(),
                    'node': hmac.new(secret_key, uname_result.node.encode('utf-8'),
hashlib.sha256).hexdigest(),
                    'system': hmac.new(secret_key, str(machine_info_tuple
[4]).encode('utf-8'),
                                       hashlib.sha256).hexdigest(),
                    'processor': hmac.new(secret_key, str(machine_info_tuple
[5]).encode('utf-8'),
                                          hashlib.sha256).hexdigest(),
                }

                write_to_file('results.txt', hashed_machine_info)
                # Insert hashed_machine_info into the database
                machine_id = insert_machine_info(con, hashed_machine_info,
hashed_serialNumber)

                insert_user(con, entry ['username'], entry['email'], machine_id)
                retval ['flag'] = 1
                retval ['message'] = "You have successfully signed up"
                comm.send(retval, dest=0, tag=4)
        elif msg1 ['option']==2:  # the user chose login option
            # check if username exists
            if not check_username_existence(con, msg1 ['username']):
                retval ['flag'] = 0
                retval ['message'] = "Wrong username"
                comm.send(retval, dest=0, tag=2)
            else:
                retval ['flag'] = 1
                retval ['message'] = "Username is included"
                comm.send(retval, dest=0, tag=2)
                entry = comm.recv(source=0, tag=3)
                expected_machine_info = fetch_registered_machine_info(con,
entry['username'])
                expected_serialNumber = fetch_serialNumber(con, entry['username'])
                #writeTuple_to_file('results.txt', expected_machine_info)
                if expected_machine_info is None:
                    retval ['flag'] = 0
                    retval['message'] = "Error when fetching the machine info"
                elif expected_serialNumber is None:
```

```
                    retval ['flag'] = 0
                    retval ['message'] = "Error when fetching the stored serial number "
                else:  # hash the data we get from user in order to check if they are the
same as in database
                    machine_info_tuple = entry['machineInfo']
                    serial_number = entry['serialNumber']
                    hashed_serialNumber = hmac.new(secret_key,
str(serial_number).encode('utf-8'),
                                                   hashlib.sha256).hexdigest()
                    #writeTuple_to_file('results.txt', machine_info_tuple)
                    uname_result = platform.uname()
                    uname_string = f"system={uname_result.system},
node={uname_result.node}, release={uname_result.release},
version={uname_result.version}, machine={uname_result.machine}"
                    hashed_machine_info = {
                        'machine': hmac.new(secret_key, str(machine_info_tuple
[0]).encode('utf-8'),
                                            hashlib.sha256).hexdigest(),
                        'platform': hmac.new(secret_key, str(machine_info_tuple
[1]).encode('utf-8'),
                                             hashlib.sha256).hexdigest(),
                        'uname': hmac.new(secret_key, uname_string.encode('utf-8'),
hashlib.sha256).hexdigest(),
                        'node': hmac.new(secret_key, uname_result.node.encode('utf-8'),
hashlib.sha256).hexdigest(),
                        'system': hmac.new(secret_key, str(machine_info_tuple
[4]).encode('utf-8'),
                                           hashlib.sha256).hexdigest(),
                        'processor': hmac.new(secret_key, str(machine_info_tuple
[5]).encode('utf-8'),
                                              hashlib.sha256).hexdigest(),
                    }
                    hashed_machine_info_tuple = (
                        hashed_machine_info ['machine'],
                        hashed_machine_info ['platform'],
                        hashed_machine_info ['uname'],
                        hashed_machine_info ['node'],
                        hashed_machine_info ['system'],
                        hashed_machine_info ['processor']
                    )

                    if hashed_machine_info_tuple == expected_machine_info and
hashed_serialNumber == str( expected_serialNumber[0] ):
                        retval['flag'] = 1   # machine info have been matched
                        msg1['email'] = fetch_user_email(con,entry['username'])
                        comm.send(msg1, dest=1, tag=11)
                        local_OTP = comm.recv(source=1 , tag=12)
                        write_to_file('results.txt','local_otp: '+local_OTP)
                        retval['message'] = "Email with otp has been send"
                        comm.send(retval, dest=0, tag=4)
```

```
                    user_OTP = comm.recv(source=0, tag=5)
                    write_to_file('results.txt', 'user_otp: ' + user_OTP)
                    if local_OTP == user_OTP:
                        retval ['flag'] = 1
                        retval ['message'] = "Correct Otp "
                    else:
                        retval ['flag'] = 0
                        retval ['message'] = "Wrong Otp "
                    write_to_file('results.txt', retval['message'])
                    comm.send(retval, dest=0, tag=6)

                else:
                    retval ['flag'] = 0
                    retval ['message'] = "Machine info were not matched with those in
system... "
                    write_to_file('results.txt', retval['message'] )
                    comm.send(retval, dest=0, tag=4)

if rank==1:
    while True:
        msg1 = comm.recv(source=2, tag=11)
        if msg1 ['option']==0:
            exit()
        elif msg1['option']==2:
            msg2['OTP'] = generateOTP()
            msg2['len'] = len(msg2['OTP'])
            send_otp_viaEmail(msg1['email'], msg2['OTP'])
            comm.send(msg2['OTP'], dest=2, tag=12)
```

app.py

```
import webbrowser
from threading import Timer
from mpi4py import MPI
import requests
import platform
import subprocess
from  helpers import *

from flask import Flask, render_template,request, redirect, url_for
from configuratorStrings import reCaptchaSecretKey

config_file_path = 'app.yaml'
recaptcha_secret_key = reCaptchaSecretKey(config_file_path)

app = Flask(__name__)
comm = MPI.COMM_WORLD
msg1 = {}
```

```
msg2 = {}
retval = {'flag': 0, 'message': ''}
entry = {'username': None, 'machineInfo': None, 'serialNumber': None, 'email':
None}

@app.route("/")
def welcome():
    #return "Hello World!"
    return render_template('welcome.html')


@app.route('/main')
def main():
    return render_template('main.html')

#@app.route('/register')
#def register():
#    return render_template('register.html')
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method=='GET':
        # Handle GET request
        return render_template('register.html')
    elif request.method=='POST':
        # Handle POST request
        msg1['option'] = 1
        username = request.form['username']
        user_email = request.form['email']
        recaptcha_response = request.form.get('g-recaptcha-response')
        # Verify the reCAPTCHA response with Google's reCAPTCHA verification API
        response = requests.post('https://www.google.com/recaptcha/api/siteverify',
                                 data={'secret': recaptcha_secret_key, 'response':
recaptcha_response})
        challengeResult = response.json()
        if challengeResult['success']:
            msg1['username'] = username
            comm.send(msg1, dest=2, tag=1)
            # Receive response from rank 2 if username already exists in database or is
available
            message = comm.recv(source=2, tag=2)
            if message['flag'] == 0 : # username not available
                write_to_file('results.txt', message['message'])
                return render_template('register.html', message=message['message'])
            else: # when username is available , then fetching device info
                entry['username'] = username
                entry['email'] = user_email
                entry['machineInfo'] = getEntryInfo()
                entry['serialNumber'] = get_motherboard_serial_number()
```

```
            # Username is available, proceed with registration or redirect to another
page
            comm.send(entry, dest=2, tag=3)
            message = comm.recv(source=2, tag=4)
            if message ['flag']==1:
                #return render_template('register.html',  message=message['message'])
                return redirect(url_for('login'))
                #return redirect(url_for('welcome'))
            else:
                return render_template('register.html', message="something went
wrong")
        else:
            #reCAPTCHA verification failed, handle accordingly
            return render_template('register.html', message="reCAPTCHA verification
failed. Please try again.")



@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        # Handle GET request
        return render_template('login.html')
    elif request.method =='POST':
        # Handle POST request
        msg1['option'] = 2
        username = request.form['username']
        msg1['username'] = username
        comm.send(msg1, dest=2, tag=1)
        # Receive response from rank 2 if username already exists in database or is
available
        message = comm.recv(source=2, tag=2)
        if message['flag'] == 0 : # username not registered
            return render_template('login.html', message=message['message'])
        else: # send machine info
            entry['username'] = username
            entry['machineInfo'] = getEntryInfo()
            entry['serialNumber'] = get_motherboard_serial_number()
            comm.send(entry, dest=2, tag=3)
        message = comm.recv(source=2, tag=4)
        if message['flag'] == 0:  # unsuccessful login
            return render_template('login.html', message=message['message'])
        else:  # successfully login
            return render_template('otp_form.html')

@app.route('/verify_otp', methods=['GET', 'POST'])
def verify_otp():
    if request.method == 'GET':
        # Handle GET request
        return render_template('otp_form.html')
```

```
    elif request.method == 'POST':
        # Retrieve OTP entered by the user from the form
        otp = request.form['digit1'] + request.form['digit2'] + request.form['digit3'] +
request.form['digit4'] + request.form['digit5'] + request.form['digit6']
        #writeStr_to_file('results.txt', otp)
        comm.send(otp, dest=2, tag=5)
        message = comm.recv(source=2, tag=6)
        if message['flag'] == 0:  # wrong otp
            return render_template('login.html')
        else:
            return redirect(url_for('main'))

def open_browser():
      webbrowser.open_new("http://127.0.0.1:8080")

def openWenApi():
    Timer(1, open_browser).start()
    app.run(port=8080)
```

myDatabase.py

```
import sqlite3
import yaml

from configuratorStrings import databaseConnectionString

# Load configuration and get connection string
config_file_path='app.yaml'
connection_string=databaseConnectionString(config_file_path)

# everything that includes database info
defcreate_tables_if_not_exist():
connection=sqlite3.connect(connection_string)
con=connection.cursor()
#con.execute('''DROP TABLE USERS''')
#con.execute('''DROP TABLE MACHINES''')
con.execute('''
        CREATE TABLE IF NOT EXISTS MACHINES (
            MACHINE_ID INTEGER PRIMARY KEY AUTOINCREMENT,
            MACHINE TEXT,
            PLATFORM TEXT,
            UNAME TEXT,
            NODE TEXT,
            SYSTEM TEXT,
            PROCESSOR TEXT,
            SERIAL_NUMBER TEXT
        )
    ''')
```

```
con.execute('''
        CREATE TABLE IF NOT EXISTS USERS (
            USER_ID INTEGER PRIMARY KEY AUTOINCREMENT,
            USERNAME TEXT,
            EMAIL TEXT,
            MACHINE_ID INTEGER,
            FOREIGN KEY (MACHINE_ID) REFERENCES MACHINES(MACHINE_ID)
        )
    ''')
connection.commit()

def close_connection (con):
    con.close()


def insert_machine_info (con, machine_info):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    machine_id = con.execute(
        "INSERT INTO MACHINES (MACHINE, PLATFORM, UNAME, NODE,
SYSTEM, PROCESSOR) VALUES (?, ?, ?, ?, ?, ?)",
        (machine_info['machine'], machine_info['platform'], machine_info['uname'],
         machine_info['node'], machine_info['system'], machine_info['processor'])
    ).lastrowid
    connection.commit()
    return machine_id


def insert_machine_info(con, machine_info, serial_number):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    machine_id = con.execute(
        "INSERT INTO MACHINES (MACHINE, PLATFORM, UNAME, NODE,
SYSTEM, PROCESSOR, SERIAL_NUMBER) VALUES (?, ?, ?, ?, ?, ?, ?)",
        (machine_info['machine'], machine_info['platform'], machine_info['uname'],
         machine_info['node'], machine_info['system'], machine_info['processor'],
serial_number)
    ).lastrowid
    connection.commit()
    return machine_id


def insert_user(con, username, user_email, machine_id):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    con.execute("INSERT INTO USERS (USERNAME, EMAIL, MACHINE_ID)
VALUES (?, ?, ?)",
                (username, user_email, machine_id))
    connection.commit()
```

```
def fetch_user_email(con, username):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    user_email = con.execute(
        "SELECT EMAIL FROM USERS WHERE USERNAME=?",(username,))
    connection.commit()
    return user_email.fetchone()

def check_username_existence (con, username):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    con.execute("SELECT COUNT(USERNAME) FROM USERS WHERE
USERNAME=?", (username,))
    username_check = con.fetchone()
    connection.commit()

    return username_check[0] != 0


def fetch_registered_machine_info (con, username):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    cursor = con.execute(
        "SELECT MACHINES.MACHINE, MACHINES.PLATFORM,
MACHINES.UNAME, MACHINES.NODE, MACHINES.SYSTEM,
MACHINES.PROCESSOR"
        " FROM MACHINES, USERS WHERE MACHINES.MACHINE_ID =
USERS.MACHINE_ID AND USERS.USERNAME=?",
        (username,))
    connection.commit()

    return cursor.fetchone()

def fetch_serialNumber (con, username):
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    cursor = con.execute(
        "SELECT SERIAL_NUMBER FROM MACHINES, USERS WHERE
MACHINES.MACHINE_ID = USERS.MACHINE_ID AND USERS.USERNAME=?",
        (username,))
    connection.commit()
    return cursor.fetchone()


def print_table_raws ():
    connection = sqlite3.connect(connection_string)
    con = connection.cursor()
    cursor = con.execute("SELECT * FROM USERS , MACHINES WHERE
MACHINES.MACHINE_ID = USERS.MACHINE_ID")
    connection.commit()
```

```
    for row in con.fetchall():
        print(row)

def drop_tables(con):
    connection=sqlite3.connect(connection_string)
    con=connection.cursor()
    con.execute("'DROP TABLE USERS"')
    con.execute("'DROP TABLE MACHINES"')
    connection.commit()
```

configuratorStrings.py

```
importyaml


defload_config(config_file_path):
withopen(config_file_path,'r')asyaml_file:
returnyaml.safe_load(yaml_file)


defdatabaseConnectionString(config_file_path):
config=load_config(config_file_path)
connection_string=config['database']['connection_string']
returnconnection_string

defemailStrings(config_file_path):
config=load_config(config_file_path)
email_sender=config['email']['sender']
email_password=config['email']['password']
returnemail_sender,email_password

defgetSecretKey(config_file_path):
config=load_config(config_file_path)
secret_key=config['key']['secret_key']
returnsecret_key

defreCaptchaSecretKey(config_file_path):
config=load_config(config_file_path)
secret_key=config['key']['reCapsecret_key']
returnsecret_key
```

helpers.py

```
# this is for storing all the functions needed for the project to run
import platform
import subprocess
import random
```

```
import os
import ssl
from email.message import EmailMessage
import smtplib
from configuratorStrings import emailStrings

config_file_path = 'app.yaml'

def write_to_file(filename, content):
    with open(filename, 'a') as file:
        if isinstance(content, str):
            file.write(content + '\n')
        elif isinstance(content, tuple):
            for item in content:
                file.write(str(item) + ' |  ')
            file.write('\n')
        elif isinstance(content, dict):
            for key, value in content.items():
                file.write(f"{value} |  ")
            file.write('\n')
        else:
            file.write(str(content) + '\n')

def get_motherboard_serial_number():
    system = platform.system()
    if system == "Windows":
        try:
            import wmi
            c = wmi.WMI()
            for board in c.Win32_BaseBoard():
                return board.SerialNumber
        except Exception as e:
            print("Error:", e)
            return None
    elif system == "Linux":
        try:
            output = subprocess.check_output(['sudo', 'dmidecode', '-s', 'baseboard-
serial-number']).strip()
            return output.decode("utf-8")
        except subprocess.CalledProcessError:
            return None
    else:
        print("Unsupported platform:", system)
        return None

def generateOTP ():
    digits = "0123456789"
    OTP = ""
    for i in range(6):
        OTP += digits [random.randint(0, 9)]
```

```
    return OTP

def generateInputCaptcha():
    digits = "0123456789"
    inputcap = ""
    for i in range(6):
        inputcap += digits[random.randint(0, 9)]
    return inputcap


def getEntryInfo ():
    try:
        machine_info = platform.machine()
        platform_info = platform.platform()
        uname_info = platform.uname()
        system_info = platform.system()
        processor_info = platform.processor()
        uname_string = f"system={uname_info.system}, node={uname_info.node},
release={uname_info.release}, version={uname_info.version},
machine={uname_info.machine}"
        return machine_info, platform_info, uname_string, uname_info.node,
system_info, processor_info

    except Exception as e:
        print("Error while getting entry information:", e)
        return None, None, None, None, None, None

def send_otp_viaEmail(email_receiver, otp):

    email_sender, email_password = emailStrings(config_file_path)
    subject = 'Otp'
    body = f"The OTP you received from MyDrama App is: {otp}"

    em = EmailMessage()
    em['From'] = email_sender
    em['To'] = email_receiver
    em['Subject'] = subject
    em.set_content(body)

    # layer of security
    context = ssl.create_default_context()

    with smtplib.SMTP_SSL('smtp.gmail.com', 465, context=context) as smtp:
        smtp.login(email_sender, email_password)
        smtp.sendmail(email_sender, email_receiver, em.as_string())
```

welcome.html

```
<!DOCTYPE html>

<html>
<head>
   <title>Welcome</title>
</head>
<body>
   <h1>Welcome to myPassLess Web App Authentication Form</h1>
   <p>Please select an option:</p>
   <ul>
      <li><a href="/register">Register</a></li>
      <li><a href="/login">Login</a></li>
   </ul>
</body>
</html>
```

register.html

```
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Registration</title>
   <script src="https://www.google.com/recaptcha/api.js" async defer></script>
<!-- Include Google reCAPTCHA JavaScript -->
</head>
<body>
   <h1>Registration</h1>
   <p>Enter your details below:</p>

   {% if message %}
   <p style="color: red;">{{ message }}</p>
   {% endif %}

   <form action="/register" method="post">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username" required><br><br>
      <label for="username">Email:</label>
      <input type="text" id="Email" name="email" required><br><br>
      <!-- Google reCAPTCHA -->
      <div class="g-recaptcha" data-
sitekey="6LfN43YpAAAAAJlGuEfpQbYLGZVA_uEXqp5COcAk"></div>
      <br>
      <input type="submit" value="Sign Up">
   </form>
</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Login</title>
</head>
<body>
   <h1>Login</h1>

   {% if message %}
   <p style="color: red;">{{ message }}</p>
   {% endif %}

   <form action="/login" method="post">
      <label for="username">Username:</label><br>
      <input type="text" id="username" name="username" required><br><br>

      <button type="submit">Login</button>
   </form>
</body>
</html>
```

otp_form.html

```
<!DOCTYPE html>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>OTP</title>
<style>
   body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
   }

  .otp-form {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
```

```
  }

  .otp-input {
      width: 40px;
      height: 40px;
      text-align: center;
      font-size: 18px;
      margin: 0 5px;
      border: 1px solid #ccc;
      border-radius: 4px;
      outline: none;
  }

  .otp-input:focus {
      border-color: #007bff;
  }

  .submit-btn {
      background-color: #007bff;
      color: #fff;
      border: none;
      padding: 10px 20px;
      font-size: 16px;
      border-radius: 4px;
      cursor: pointer;
  }
</style>
</head>
<body>
<div class="otp-form">
   <h2>Enter OTP</h2>
   <form action="/verify_otp" method="post">
      <input type="text" class="otp-input" name="digit1" maxlength="1" required
autofocus>
      <input type="text" class="otp-input" name="digit2" maxlength="1" required>
      <input type="text" class="otp-input" name="digit3" maxlength="1" required>
      <input type="text" class="otp-input" name="digit4" maxlength="1" required>
      <input type="text" class="otp-input" name="digit5" maxlength="1" required>
      <input type="text" class="otp-input" name="digit6" maxlength="1" required>
      <br><br>
      <button type="submit" class="submit-btn">Submit OTP</button>
   </form>
</div>
</body>
</html>
```

main.html

```
<!DOCTYPE html>
<htmllang="en">
<head>
<metacharset="UTF-8">
<metaname="viewport"content="width=device-width, initial-scale=1.0">
<title>myPassLess</title>
<style>
      /* CSS styles go here */
      body {
         font-family: Arial, sans-serif;
         background-color: #f5f5f5;
         margin: 0;
         padding: 0;
      }

      header {
         background-color: #333;
         color: #fff;
         padding: 10px;
         text-align: center;
      }

      nav {
         background-color: #444;
         color: #fff;
         padding: 10px;
         text-align: center;
      }

      nav ul {
         list-style-type: none;
         margin: 0;
         padding: 0;
      }

      nav ul li {
         display: inline;
         margin-right: 10px;
      }

      nav ul li a {
         color: #fff;
         text-decoration: none;
         padding: 5px 10px;
      }

      nav ul li a:hover {
         background-color: #ddd; /* Light grey background on hover */
         color: #444; /* Dark grey text on hover */
```

```
      }

      section {
         padding: 20px;
         text-align: center;
}

footer {
         background-color: #333;
         color: #fff;
         padding: 10px;
         text-align: center;
         position: fixed;
         bottom: 0;
         width: 100%;
      }
</style>
</head>
<body>
<header>
<h1>myPassLess</h1>
</header>

<nav>
<ul>
<li><a href="#">Help</a></li>
</ul>
</nav>

<section>
<h2>Welcome to myPassLess</h2>
<p>This is my sample webpage.</p>
</section>

</body>
</html>
```

ΑΝΑΦΟΡΕΣ

[1] Center, M. S. -. D., n.d. Download Microsoft MPI v10.0 from Official Microsoft Download Center. [Online] Available at: https://www.microsoft.com/en-us/download/details.aspx?id=57467[Accessed 11 August 2023].

[2] docs.python.org, n.d. hashlib — Secure hashes and message digests — Python 3.8.4rc1 documentation.

[3] Dr. Xu, J., ed., (2022). The Future And Fintech: ABCDI And Beyond. Singapore: World Scientific Publishing Company.

[4] Gune, A., 2017. The Cryptographic Implications of the LinkedIn Data Breach.

[5] Kabanov, I. a. M. S., 2021. Applying the Lessons from the Equifax Cybersecurity Incident to Build a Better Defense. MIS Quarterly Executive, 20(2), p. Article 4.

[6] Pathak, R. K., 2022. Network Security Implementation Layer through Voice Biometric. Network Security Implementation Layer through Voice Biometric, p. 16.

[7] Sadat, S. E. L. H. &. A. N., 2023. Highly Secure and Easy to Remember Password-Based Authentication Approach,. Journal for Research in Applied Sciences and Biotechnology, 2(1), pp. 134-141.

[8] Steven A. Grosz and Anil K. Jain, L. F., 2015. Latent Fingerprint Recognition: Fusion of Local and. JOURNAL OF LATEX CLASS FILES, August, 14(8), p. 13 [Preprint] Available at: Available at: https://arxiv.org/pdf/2304.13800.pdf [Accessed 25 Aug. 2023].

[9] von Paris, S., (2013) Bayesian Evaluation of Forensic Fingerprint Evidence with Automatic Biometric Systems - Implementation and Statistical Performance Analysis).Master's Thesis. Staffordshire University. Available at: https://www.researchgate.net (Accessesd: 25 August 2023).

[10]Williams, L., 2021. The People Who Live in Glass Houses Are Happy the Stones Weren't Thrown at Them [From the Editors]. IEEE Security & Privacy, May-June, 19(3), pp. 4-7.

[11] www.fujitsu.com. (n.d.). How PalmSecure works - Fujitsu Malaysia. [online] Available at:https://www.fujitsu.com/my/solutions/business-technology/security/palmsecure/technology/ [Accessed 26 Aug. 2023].

[12] Rash, W.R.W.R. is a longtime technology journalist who has directed product testing centers H. is P. of W., Associates, Analysis, A. and Washington, editorial services firm located near (n.d.). Review: Fujitsu PalmSecure Sensor. [online] Technology Solutions That Drive Government. Available at: https://statetechmagazine.com/article/2011/10/review-fujitsu-palmsecure-sensor [Accessed 26 Aug. 2023].

[13] Fujitsu Frontech North America. (n.d.). PalmSecure SDK. [online] Available at: https://fujitsufrontechna.com/palmsecure/sdk/ [Accessed 26 Aug. 2023].

[14] ProxyNova. (n.d.). Brute Force Calculator - How long would it take to Brute Force your password? [online] Available at: https://www.proxynova.com/tools/brute-force-calculator/ [Accessed 20 Jul. 2023].

[15] Jones, C. (2021). The Most Significant Password Breaches Of 2020. [online] Expert Insights. Available at: https://expertinsights.com/insights/the-most-significant-password-breaches/.

[16] Southall, A., Weiser, B. and Rubinstein, D. (2021). This Agency's Computers Hold Secrets. Hackers Got In With One Password. The New York Times. [online] 18 Jun. Available at: https://www.nytimes.com/2021/06/18/nyregion/nyc-law-department-hack.html.

[17] Weiser, B. (2021). Fallout From Hack of City Law Department Could Linger for Months. The New York Times. [online] 9 Jul. Available at: https://www.nytimes.com/2021/07/09/nyregion/nyc-law-department-hacked.html.

[18] richards, kathleen and Wigmore, I. (2021). What is a one-time password (OTP)? Definition from Search Security. [online] Search Security. Available at: https://www.techtarget.com/searchsecurity/definition/one-time-password-OTP.

[19] paolo matarazzo (2022). Smart Card Architecture - Windows Security. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/windows/security/identity-protection/smart-cards/smart-card-architecture [Accessed 27 Aug. 2023].

[20] SIPSER, M. (2012). Εισαγωγή στη θεωρία υπολογισμού. 3rd ed. Translated by Χ. Καπούτσης. Ηράκλειο Κρήτης: Νικ. Πλαστήρα 100, Βασιλικά Βουτών, 700 13: Πανεπιστημιακές εκδόσεις Κρήτης, pp.11–14.

[21] Ιωαννίδου, Β. (2023) *Μηχανισμοί ελέγχου πρόσβασης με χρήση συνθηματικών*. Διπλωματική Εργασία. Πανεπιστήμιο Δυτικής Αττικής.

[22] www.justice.gov. (2020). Russian Hacker Sentenced to Over 7 Years in Prison for Hacking into Three Bay Area Tech Companies. [online] Available at: https://www.justice.gov/usao-ndca/pr/russian-hacker-sentenced-over-7-years-prison-hacking-three-bay-area-tech-companies.

[23] Lee, C.T. and Pan, L.-Y. (2023), "Smile to pay: predicting continuous usage intention toward contactless payment services in the post-COVID-19 era", International Journal of Bank Marketing, Vol. 41 No. 2, pp. 312-332. https://doi.org/10.1108/IJBM-03-2022-0130

[24] Masters of Media. (2020). No card, No Phone, No problem! Alipay: Pay with your face, A balance between convenience, security and privacy. [online] Available at: https://mastersofmedia.hum.uva.nl/blog/2020/09/27/no-card-no-phone-no-problem-alipay-pay-with-your-face-a-balance-between-convenience-security-and-privacy/ [Accessed 3 Dec. 2023].

[25] Makortoff, K. and correspondent, K.M.B. (2022). Mastercard launches 'smile to pay' system amid privacy concerns. The Guardian. [online] 17 May. Available at:

https://www.theguardian.com/technology/2022/may/17/mastercard-launches-smile-to-pay-amid-privacy-concerns.

[26] developer.mastercard.com. (n.d.). Mastercard Developers. [online] Available at: https://developer.mastercard.com/blog/biometric-checkout/ [Accessed 3 Dec. 2023].

[27] Farooq, U. (2020). Real Time Password Strength Analysis on a Web Application Using Multiple Machine Learning Approaches. International Journal of Engineering Research &Technology (IJERT), [online] 9(12). [Accessed 10 Feb. 2024].

[28] Wang, X. et al. (2021) 'Attacks and defenses in user authentication systems: A survey', Journal of Network and Computer Applications, 188, p. 103080. doi: 10.1016/j.jnca.2021.103080.

[29]Honda, S., Unno, Y., Maruhashi, K., Takenaka, M. and Torii, S. (2014). Detection of Novel-Type Brute Force Attacks Used Ephemeral Springboard IPs as Camouflage. *Journal of Advances in Computer Networks*, [online] 2(4). Available at: https://www.researchgate.net [Accessed 11 Feb. 2024].

[30] Boye Aziboledia Frederick, Onate Egerton Taylor. Analysis on Cybersecurity Control and Monitoring Techniques in Industrial IoT: Industrial Control Systems. Internet of Things and Cloud Computing. Vol. 11, No. 1, 2023, pp. 1-17. doi: 10.11648/j.iotcc.20231101.11

[31] PCMAG. (n.d.). *Hacker Breached LastPass by Installing Keylogger on Employee's Home Computer*. [online] Available at: https://www.pcmag.com/news/hacker-breached-lastpass-by-installing-keylogger-on-employees-home-computer.[Accessed 11 Feb. 2024].

[32] Stallings, W. andBrown, L. (2016). *ΑΣΦΑΛΕΙΑ ΥΠΟΛΟΓΙΣΤΩΝ Αρχές και πρακτικές*. 3rd ed. Translated by Γ. Στάμου. ΕκδόσειςΚλειδάριθμος, pp.112–113.

[33] Thapa, A., Dhapola, C. S. and Saini, H. (2022) 'Security Analysis of User Authentication and Methods', in *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing*. New York, NY, USA: Association for Computing Machinery (IC3-2022), pp. 564–572. doi: 10.1145/3549206.3549304.

[34] www.justice.gov. (2020). *Ticketmaster Pays $10 Million Criminal Fine for Intrusions into Competitor's Computer Systems*. [online] Available at: https://www.justice.gov/usao-edny/pr/ticketmaster-pays-10-million-criminal-fine-intrusions-competitor-s-computer-systems-0. [Accessed 12 Feb. 2024].

[35] Stoney, D. A. et al. (2020) 'Occurrence and associative value of non-identifiable fingermarks', Forensic Science International, 309, p. 110219. doi: 10.1016/j.forsciint.2020.110219.

[36]Avison, D. and Fitzgerald, G. (2017). *Ανάπτυξη Πληροφοριακών Συστημάτων-Μεθοδολογίες, Τεχνικές και Εργαλεία*. 3η ed. Translated by Νικ.Σπ. Βώρος. Translated by Γρ.Ν. Μπεληγιάννης. and Translated by Γ.Αθ. Τσιρογιάννης. Στουρνάρη 49Α, 106 82 Αθήνα: ΕΚΔΟΣΕΙΣΝΕΩΝΤΕΧΝΟΛΟΓΙΩΝ, pp.402–408.

[37]About SSL. (n.d.). *Digital Signature vs. Electronic Signature – What's the Difference?* [online] Available at: https://aboutssl.org/digital-signature-vs-electronic-signature/ [Accessed 2 Mar. 2024].

[38] Papaspirou, V.; Papathanasaki, M.; Maglaras, L.; Kantzavelou, I.; Douligeris, C.; Ferrag, M.A.; Janicke, H. A Novel Authentication Method That Combines Honeytokens and Google Authenticator. *Information* **2023**, *14*, 386. https://doi.org/10.3390/info14070386.

[39] Spanos, A. and Kantzavelou, I. An Ethereum based e-Voting system. EAI WiCON 2023 - 16th EAI International Conference on Wireless Internet, Athens, December 2023.