



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Συστήματος Smart Lock για Ελεγχόμενη
Πρόσβαση σε Εργαστηριακούς Χώρους**

**Του φοιτητή
Άγγελου Δούκα
Αρ. Μητρώου: 517030**

**Επιβλέπουσα
Μαρία Παπαδοπούλου
Επίκουρη Καθηγήτρια**

*Τίτλος Δ.Ε.: Ανάπτυξη Συστήματος Smart Lock για Ελεγχόμενη Πρόσβαση σε
Εργαστηριακούς Χώρους*

Κωδικός Δ.Ε.: 25377

Ονοματεπώνυμο φοιτητή: Δούκας Άγγελος

Ονοματεπώνυμο εισηγητή: Παπαδοπούλου Μαρία

Ημερομηνία ανάληψης Δ.Ε. 29-11-2025

Ημερομηνία περάτωσης Δ.Ε.

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Δούκα Άγγελου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Προλογος

Η ασφάλεια και ο έλεγχος πρόσβασης σε κλειστούς χώρους αποτελούν κρίσιμους παράγοντες για τη σωστή λειτουργία οργανισμών, εκπαιδευτικών ιδρυμάτων και ερευνητικών υποδομών. Ιδιαίτερα στους εργαστηριακούς χώρους, όπου συχνά φιλοξενούνται εξειδικευμένος εξοπλισμός, ευαίσθητα δεδομένα και δυνητικά επικίνδυνες ουσίες, η ανάγκη για αξιόπιστα και ευέλικτα συστήματα ελέγχου πρόσβασης είναι επιτακτική. Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη ενός συστήματος smart lock για ελεγχόμενη πρόσβαση σε εργαστηριακούς χώρους, το οποίο αξιοποιεί σύγχρονες τεχνολογίες ενσωματωμένων συστημάτων και αυτοματισμού. Το προτεινόμενο σύστημα στοχεύει στη βελτίωση του επιπέδου ασφαλείας, στη διευκόλυνση της διαχείρισης χρηστών

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία αφορά την ανάπτυξη ενός έξυπνου συστήματος κλειδώματος (Smart Lock) βασισμένου σε Raspberry Pi, με στόχο την ασφαλή και ευέλικτη πρόσβαση σε εργαστηριακούς χώρους.

Το σύστημα υλοποιήθηκε ως ολοκληρωμένη λύση ελεγχόμενης πρόσβασης και περιλαμβάνει τέσσερα βασικά υποσυστήματα: α) οθόνη προβολής πληροφοριών για πρόγραμμα αίθουσας και ανακοινώσεις, β) τοπικό web service βασισμένο σε Flask για απομακρυσμένο έλεγχο κλειδώματος/ξεκλειδώματος μέσω ασφαλούς σύνδεσης, προσβάσιμο αποκλειστικά από το εσωτερικό δίκτυο, γ) εφεδρική λειτουργία NFC με module RC522 για άνοιγμα με κάρτα σε περίπτωση απουσίας δικτύου και δ) ηλεκτροπύρο 12V ως φυσικό μηχανισμό κλειδώματος.

Για την οδήγηση του ηλεκτροπύρου σχεδιάστηκε κύκλωμα Low-Side Switch με Logic-Level MOSFET IRLZ44N, με αντιστάσεις προστασίας πύλης 220Ω και pull-down 10kΩ και δίοδο flyback 1N4007 για προστασία από επαγωγικές υπερτάσεις. Το λογισμικό αναπτύχθηκε σε Python με αρχιτεκτονική multi-threading, ώστε ο Flask server και το NFC polling loop να λειτουργούν ταυτόχρονα με κοινό σημείο ελέγχου.

Τα αποτελέσματα των δοκιμών έδειξαν αξιόπιστη λειτουργία τόσο μέσω web όσο και μέσω NFC, με χαμηλή κατανάλωση και θερμική καταπόνηση. Το σύστημα προσφέρει ασφάλεια, ευχρηστία και ανεξαρτησία από εμπορικούς προμηθευτές, αποτελώντας μια οικονομική και επεκτάσιμη λύση για ελεγχόμενη πρόσβαση σε χώρους.

ABSTRACT

«Development of a Smart Lock System for Controlled Access to Laboratory Areas»

«Angelos Doukas»

This thesis presents the development of a smart lock system based on Raspberry Pi, aimed at providing secure and flexible access control for laboratory spaces.

The system was implemented as an integrated access control solution comprising four main subsystems: a) an information display for room scheduling and announcements, b) a local web service based on Flask for remote lock/unlock control via a secure connection, accessible only from the internal network, c) a backup NFC functionality using an RC522 module for card-based access in case of network unavailability, and d) a 12V electric strike as the physical locking mechanism.

For driving the electric strike, a Low-Side Switch circuit was designed using a Logic-Level MOSFET IRLZ44N, with a 220Ω gate protection resistor, a $10k\Omega$ pull-down resistor, and a 1N4007 flyback diode for protection against inductive voltage spikes. The software was developed in Python using a multi-threading architecture, allowing the Flask server and the NFC polling loop to operate concurrently with a common control point.

Test results demonstrated reliable operation both via the web interface and NFC, with low power consumption and thermal stress. The system offers security, usability, and independence from commercial vendors, providing a cost-effective and scalable solution for controlled access to facilities

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στην οικογένειά μου και στους φίλους μου, οι οποίοι στάθηκαν αδιάκοπα δίπλα μου, προσφέροντάς μου ηθική, πνευματική και οικονομική στήριξη καθ' όλη τη διάρκεια της προσπάθειάς μου. Ιδιαίτερη ευγνωμοσύνη οφείλω στην επιβλέπουσα καθηγήτριά μου, κα. Μαρία Παπαδοπούλου, για την άριστη συνεργασία και την πολύτιμη καθοδήγησή της.

Περιεχομενα

Προλογος.....	3
Περίληψη.....	4
Abstract.....	5
Ευχαριστιες.....	6
Περιεχομενα.....	7
Εισαγωγή.....	9
Κεφάλαιο 1 : Συστήματα Ασφαλείας και Κλειδαριές.....	10
1.1 Ιστορικο.....	10
1.1.1 Πρώιμα Μέσα Ασφαλείας.....	10
1.1.2 Αρχαία Μεσοποταμία και Αίγυπτος.....	10
1.1.3 Αρχαία Ελλάδα και Ρώμη.....	10
1.1.4 Αναγέννηση και Νεότεροι Χρόνοι.....	11
1.1.5 Βιομηχανική Επανάσταση (18ος–19ος αιώνας).....	11
1.1.6 20ός Αιώνας.....	11
1.1.7 21ος Αιώνας – Σύγχρονα Συστήματα Ασφάλειας.....	11
1.2 Εξυπνες Κλειδαριες.....	12
1.2.1 Η έννοια της «έξυπνης» κλειδαριάς (Smart Lock).....	12
Κεφαλαιο 2: Δικτυα Υπολογιστων.....	14
2.1 Κατηγοριες Δικτυων.....	14
2.1.1 Local Area Network (LAN).....	15
2.1.2 Wireless Local Area Network (WLAN).....	18
2.1.3 Wide Area Network (WAN).....	18
2.2 Μοντελο OSI.....	19
2.3 Internet Protocol Version 4.....	21
2.4 Θύρες δικτύου.....	22
2.5 Zigbee.....	25
2.6 Ραδιοσυχνότητα.....	28
2.7 Bluetooth.....	30
2.8 VPN (Virtual Private Network).....	33
Κεφαλαιο 3: Raspberry Pi.....	38
3.1 Ο Ρολος του Raspberry Pi σε ενα συστημα Smart Lock.....	39
3.1.1 Αρχιτεκτονική συστήματος.....	40
3.2 Συστημα και αισθητηρες RFID.....	41
3.3 Ηλεκτροπυρος.....	43
Κεφαλαιο 4: Υλοποίηση Συστήματος Smart Lock για Ελεγχόμενη Πρόσβαση σε Εργαστηριακούς Χώρους.....	44
4.1 Λειτουργια MOSFET ως Διακοπτης.....	45
4.1.1 Σχεδιασμος κυκλωματος MOSFET.....	46
4.2 Δημιουργία Web Service ιστοσελίδας.....	48
4.2.1 Κωδικες.....	48
4.2.2 Αναλυση Κωδικα - Backend (app.py).....	50

4.2.3 Ανάλυση Κώδικα - Frontend (index.html).....	51
4.3 Υλοποίηση Συστήματος Προβολής με Οθόνη 7”.....	53
4.3.1. Τεχνικά Χαρακτηριστικά Οθόνης.....	53
4.3.2 Συνδεσμολογία.....	54
4.3.3 Ανάλυση Κώδικα Εφαρμογής.....	54
4.4. Υποσύστημα Αναγνώρισης NFC.....	56
4.4.1. Εισαγωγή στην τεχνολογία NFC.....	56
4.4.2. Το module RC522.....	56
4.4.3. Λογισμικό ανάγνωσης και διαχείριση UID.....	57
4.4.4. Θέματα ασφαλείας.....	58
4.5 Τελική Υλοποίηση Λογισμικού Και Ενοποίηση Υποσυστημάτων.....	58
4.5.1. Αρχιτεκτονική λογισμικού.....	58
4.5.2 Μοντέλο νημάτων και ανάγκη για threading.....	58
4.5.3. Ανάλυση τελικού κώδικα app.py	59
4.5.4. Δοκιμές και επαλήθευση.....	60
Κεφάλαιο 5: Συμπεράσματα.....	61
5.1 Σύνοψη της εργασίας.....	61
5.2 Αποτελέσματα και επίτευξη στόχων.....	61
5.3 Περιορισμοί.....	62
5.4 Μελλοντικές επεκτάσεις.....	62
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	63

Εισαγωγή

Η ραγδαία εξέλιξη των τεχνολογιών του Διαδικτύου των Πραγμάτων (Internet of Things – IoT), των ενσωματωμένων συστημάτων και των έξυπνων αισθητήρων έχει οδηγήσει στην ανάπτυξη «έξυπνων» συστημάτων ασφάλειας, τα οποία επιτρέπουν τον δυναμικό και απομακρυσμένο έλεγχο πρόσβασης. Στο πλαίσιο αυτό, τα συστήματα smart lock συνδυάζουν υλικό και λογισμικό, προσφέροντας δυνατότητες όπως η αυθεντικοποίηση χρηστών, η καταγραφή συμβάντων, η διαχείριση δικαιωμάτων και η ενσωμάτωση με άλλες ψηφιακές υποδομές. Η εργασία επικεντρώνεται τόσο στον σχεδιασμό και την υλοποίηση του υλικού του συστήματος όσο και στην ανάπτυξη του αντίστοιχου λογισμικού, εξετάζοντας ζητήματα αξιοπιστίας, ασφάλειας και χρηστικότητας. Μέσα από την ανάλυση και την υλοποίηση του συστήματος, επιδιώκεται η ανάδειξη της σημασίας των έξυπνων συστημάτων κλειδώματος ως σύγχρονης λύσης ελέγχου πρόσβασης σε απαιτητικά περιβάλλοντα, όπως οι εργαστηριακοί χώροι.

Κεφάλαιο 1 : Συστήματα Ασφαλείας και Κλειδαριές

Η ανάγκη του ανθρώπου να προστατεύσει τον εαυτό του, την οικογένειά του και την περιουσία του αποτελεί θεμελιώδες κοινωνικό φαινόμενο, το οποίο εμφανίζεται ήδη από τις πρώτες οργανωμένες ανθρώπινες κοινότητες και εξελίσσεται παράλληλα με την ανάπτυξη του πολιτισμού. Από τις πρώτες οργανωμένες κοινότητες μέχρι τις σύγχρονες «έξυπνες πόλεις», τα συστήματα ασφαλείας ενός χώρου εξελίχθηκαν συνεχώς, ακολουθώντας την τεχνολογική, κοινωνική και πολιτισμική πρόοδο. Κεντρικό ρόλο σε αυτή την εξέλιξη κατέχουν οι κλειδαριές, ως το βασικό μέσο ελέγχου πρόσβασης.[4][5]

1.1 Ιστορικό

1.1.1 Πρώιμα Μέσα Ασφαλείας

Στις πρωτόγονες κοινωνίες δεν υπήρχαν κλειδαριές με τη σύγχρονη έννοια, αλλά απλά εμπόδια που καθιστούσαν δύσκολη την είσοδο. Η ασφάλεια βασιζόταν κυρίως στην φυσική απομόνωση η οποία επιτυγχάνοταν με τις σπηλιές ή με δημιουργία καταλυμάτων σε υπερυψωμένα σημεία, επίσης ένας ακόμα τρόπος ήταν η συλλογική φύλαξη από κάποια μέλη της ομάδας.

Τέλος ένα ακόμα μέσο ασφαλείας ήταν η δημιουργία εμποδίων με φυσικά μέσα όπως πέτρες ξύλα και πάσσαλοι.[4][5]

1.1.2 Αρχαία Μεσοποταμία και Αίγυπτος

Οι πρώτες γνωστές κλειδαριές εμφανίζονται περίπου το 2000 π.Χ. στη Μεσοποταμία και στην Αίγυπτο. Οι κλειδαριες αυτές ήταν ξύλινες με πείρους (pin locks) τα κλειδιά τους ήταν φυσικά και αυτά φτιαγμένα από ξύλο ενώ η κύρια χρήση τους ήταν σε αποθήκες παλάτια και ναούς. Αυτές οι κλειδαριές θεωρούνται και οι πρόγονοι των σημερινών κλειδαριών ασφαλείας. [1][2][4]

1.1.3 Αρχαία Ελλάδα και Ρώμη

Στην Αρχαία Ελλάδα, τα σπίτια είχαν απλές θύρες με μάνδαλα. Οι κλειδαριές ήταν μεταλλικές (σιδηρές ή μπρούντζινες) και η ασφάλεια βασιζόταν κυρίως στην κοινωνική συνοχή, παρά στην τεχνολογία. Αργότερα, κατά την εποχή της Ρωμαϊκής Αυτοκρατορίας, οι Ρωμαίοι έκαναν σημαντικά άλματα, καθιερώνοντας τις μεταλλικές κλειδαριές μικρού μεγέθους, με κλειδιά που φοριούνταν σαν δαχτυλίδια, χρησιμοποιώντας πολυπλοκότερους μηχανισμούς για την εποχή.[2][3][5]

1.1.4 Αναγέννηση και Νεότεροι Χρόνοι

Κατά την Αναγέννηση και με τη βελτίωση της μεταλλουργίας, οι κλειδαριές γίνονται πιο ακριβείς και περίπλοκες, ενώ συγχρόνως εμφανίζονται κρυφοί μηχανισμοί και διακοσμητικά στοιχεία.

Αργότερα, κατά τον 17ο–18ο αιώνα, παρατηρείται ανάπτυξη πιο εξελιγμένων μηχανικών κλειδαριών, ενώ και τα σπίτια των αστών αποκτούν βασική ιδιωτική ασφάλεια.[2][3]

1.1.5 Βιομηχανική Επανάσταση (18ος–19ος αιώνας)

Η Βιομηχανική Επανάσταση άλλαξε ριζικά τα συστήματα ασφάλειας. Συγκεκριμένα, παρατηρείται η εμφάνιση των κλειδαριών ακριβείας. Η πρώτη τέτοια κλειδαριά εμφανίστηκε το 1778 και ήταν η κλειδαριά Bramah (μία από τις πρώτες κλειδαριές υψηλής ασφάλειας).

Το 1861 έκανε την εμφάνισή της και η κλειδαριά Yale με κυλινδρικό μηχανισμό πείρων. Οι κλειδαριές αυτές αποτελούν τη βάση των σύγχρονων μηχανικών κλειδαριών.[2][3][6][7]

1.1.6 20ός Αιώνας

Κατά τον 20ό αιώνα δημιουργούνται ηλεκτρομηχανικά συστήματα ασφάλειας, με ηλεκτρικές κλειδαριές, κλειδιά με κωδικούς και συστήματα συναγερμών με αισθητήρες. Παράλληλα, καθιερώνεται η χρήση συστημάτων επιτήρησης μέσω καμερών και το κλειστό κύκλωμα τηλεόρασης (CCTV).[5][11]

1.1.7 21ος Αιώνας – Σύγχρονα Συστήματα Ασφάλειας

Στον 21ο αιώνα, και στην εποχή που ζούμε, μέσα από τα άλματα που έχουν επιτευχθεί στην τεχνολογία, παρατηρείται η δημιουργία ψηφιακών και «έξυπνων» κλειδαριών. Οι σύγχρονες κλειδαριές μπορούν να ανοίγουν με PIN, κάρτες RFID ή βιομετρικά στοιχεία (δακτυλικό αποτύπωμα, αναγνώριση προσώπου), ενώ είναι δυνατός και ο έλεγχος μέσω κινητού τηλεφώνου.

Στα πιο ολοκληρωμένα συστήματα ασφάλειας περιλαμβάνονται συναγερμοί συνδεδεμένοι στο διαδίκτυο, κάμερες με τεχνητή νοημοσύνη, καθώς και συστήματα καταγραφής και ανάλυσης δεδομένων.[5][9][10]

1.2 Εξυπνες Κλειδαριες

1.2.1 Η έννοια της «έξυπνης» κλειδαριάς (Smart Lock)

Οι εξυπνες κλειδαριες είναι η φυσική συνέχεια της ιστορικής πορείας των κλειδαριών. Από τους ξυλινούς πυλούς της αρχαιότητας στους μηχανικούς κυλινδρους του 19ου αιώνα και τελικά στα ψηφιακά συστήματα πρόσβασης. Η βασική αρχή παραμένει ίδια και είναι ο έλεγχος της πρόσβασης αυτό που αλλάζει είναι το τεχνολογικό μέσο.

Τα έξυπνα συστήματα κλειδώματος (smart locks) αποτελούν μία από τις σημαντικότερες σύγχρονες εξελίξεις στον τομέα των συστημάτων ασφάλειας και ελεγχόμενης πρόσβασης. Πρόκειται για ηλεκτρονικές ή ηλεκτρομηχανικές κλειδαριές που αντικαθιστούν ή συμπληρώνουν τις παραδοσιακές μηχανικές κλειδαριές, προσφέροντας αυξημένη λειτουργικότητα, ευελιξία και δυνατότητες αυτοματοποίησης. Η βασική τους καινοτομία έγκειται στο γεγονός ότι η πρόσβαση σε έναν χώρο δεν βασίζεται αποκλειστικά σε ένα φυσικό κλειδί, αλλά σε ψηφιακά μέσα ταυτοποίησης και ελέγχου.

Τα smart locks αξιοποιούν τεχνολογίες όπως ασύρματη επικοινωνία (Wi-Fi, Bluetooth, Zigbee, NFC), μικροελεγκτές ή μικροϋπολογιστές, καθώς και λογισμικό για τη διαχείριση χρηστών και δικαιωμάτων πρόσβασης. Η διαδικασία ανοίγματος και κλειδώματος μπορεί να πραγματοποιείται μέσω κινητών συσκευών, καρτών RFID, κωδικών PIN, βιομετρικών δεδομένων (όπως δακτυλικό αποτύπωμα) ή απομακρυσμένα μέσω διαδικτύου. Με τον τρόπο αυτό, ο έλεγχος πρόσβασης γίνεται πιο ευέλικτος και προσαρμόσιμος στις ανάγκες κάθε χώρου.

Ένα από τα σημαντικότερα πλεονεκτήματα των smart locks είναι η δυνατότητα δυναμικής διαχείρισης χρηστών. Σε αντίθεση με τις παραδοσιακές κλειδαριές, όπου η απώλεια ενός κλειδιού συνεπάγεται αλλαγή κυλίνδρου ή αυξημένο κίνδυνο παραβίασης, στα έξυπνα συστήματα πρόσβασης ένα ψηφιακό «κλειδί» μπορεί να απενεργοποιηθεί άμεσα. Επιπλέον, είναι δυνατή η δημιουργία διαφορετικών επιπέδων πρόσβασης, όπως προσωρινή πρόσβαση για επισκέπτες, περιορισμένη πρόσβαση σε συγκεκριμένες ώρες ή πλήρης πρόσβαση για διαχειριστές. Αυτή η λειτουργία είναι ιδιαίτερα χρήσιμη σε εργαστηριακούς, επαγγελματικούς και εκπαιδευτικούς χώρους.

Σημαντικό ρόλο στη λειτουργία των smart locks παίζει το λογισμικό. Μέσω εφαρμογών ή web interfaces, οι διαχειριστές μπορούν να παρακολουθούν την κατάσταση της κλειδαριάς, να λαμβάνουν ειδοποιήσεις σε πραγματικό χρόνο και να τροποποιούν τις ρυθμίσεις του συστήματος. Η ασφάλεια του λογισμικού αποτελεί κρίσιμο παράγοντα, καθώς τα smart locks βασίζονται σε δικτυακές επικοινωνίες. Για τον λόγο αυτό, εφαρμόζονται τεχνικές όπως κρυπτογράφηση δεδομένων, έλεγχος ταυτότητας χρηστών και χρήση ασφαλών πρωτοκόλλων επικοινωνίας.

Παρά τα πλεονεκτήματά τους, τα smart locks παρουσιάζουν και προκλήσεις. Η εξάρτηση από ηλεκτρική ενέργεια ή μπαταρίες απαιτεί σωστό σχεδιασμό για την αποφυγή δυσλειτουργιών, ενώ η σύνδεση στο διαδίκτυο εισάγει πιθανούς κινδύνους κυβερνοασφάλειας. Ωστόσο, με σωστή υλοποίηση και τακτική συντήρηση, τα

πλεονεκτήματα υπερτερούν σημαντικά, ειδικά σε περιβάλλοντα όπου απαιτείται αυξημένος έλεγχος πρόσβασης.

Στο πλαίσιο ακαδημαϊκών και εργαστηριακών εφαρμογών, τα smart locks αποτελούν ιδανική λύση, καθώς συνδυάζουν τεχνολογία υλικού (hardware) και λογισμικού (software). Η χρήση πλατφορμών όπως το Raspberry Pi επιτρέπει την ανάπτυξη προσαρμοσμένων συστημάτων έξυπνης κλειδαριάς, τα οποία μπορούν να επεκταθούν με πρόσθετες λειτουργίες, όπως καταγραφή συμβάντων, οθόνες πληροφόρησης και απομακρυσμένη διαχείριση. Με αυτόν τον τρόπο, τα smart locks δεν λειτουργούν μόνο ως μέσο ασφάλειας, αλλά και ως πεδίο εφαρμογής σύγχρονων τεχνολογιών πληροφορικής και αυτοματισμού.[9][10]

Κεφάλαιο 2: Δίκτυα Υπολογιστών

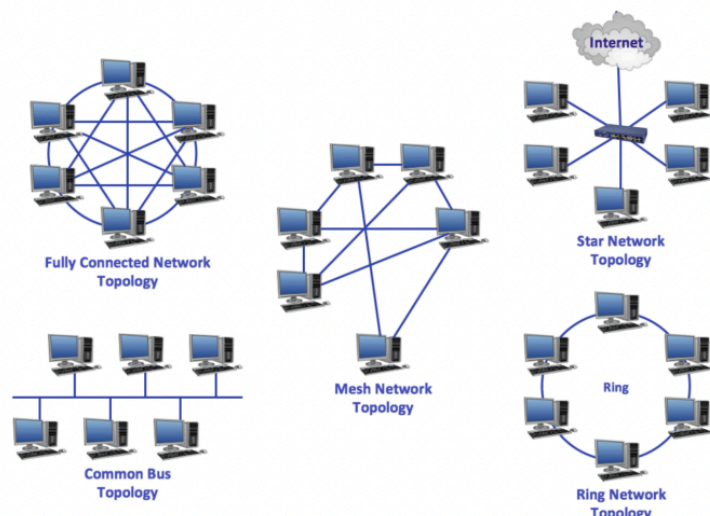
2.1 Κατηγορίες Δικτύων:

Οι κατηγορίες των δικτύων καθορίζονται κυρίως βάσει της γεωγραφικής έκτασης που καλύπτουν, του τρόπου σύνδεσης των συσκευών και της υποδομής που απαιτούν. Από τις πιο περιορισμένες τοπικά, μέχρι εκείνες που επεκτείνονται σε παγκόσμια κλίμακα, οι κατηγορίες αυτές εξυπηρετούν διαφορετικούς σκοπούς και ανάγκες επικοινωνίας. Στο πλαίσιο ενός έξυπνου σπιτιού, ιδιαίτερη σημασία έχουν τα δίκτυα PAN, LAN, WLAN και WAN, καθώς αλληλεπιδρούν άμεσα με τις συσκευές αυτοματισμού, τον τρόπο σύνδεσης στο διαδίκτυο και την απομακρυσμένη πρόσβαση.

2.1.1 Local Area Network (LAN):

Το LAN (Local Area Network) είναι η πλέον διαδεδομένη μορφή ενσύρματου (αλλά και ασύρματου δικτύου, βλ. WLAN) τοπικού δικτύου, και αφορά την καλωδιακή σύνδεση συσκευών μέσα σε έναν φυσικά περιορισμένο χώρο, όπως ένα σπίτι, ένα διαμέρισμα ή ένα γραφείο. Το LAN λειτουργεί συνήθως με τη χρήση τεχνολογιών Ethernet και απαιτεί τη χρήση επί το πλείστον switches, routers ή modems. Στα έξυπνα σπίτια, το LAN επιτρέπει τη σταθερή, αξιόπιστη και ταχύτατη σύνδεση συσκευών όπως το Raspberry Pi, οι NAS servers, οι κάμερες ασφαλείας IP, και οι έξυπνοι ελεγκτές. Το LAN διακρίνεται για τη σταθερότητα, την υψηλή ταχύτητα μετάδοσης δεδομένων και την ασφάλεια, αφού περιορίζεται εντός φυσικών συνδέσεων και δεν είναι άμεσα εκτεθειμένο σε εξωτερικές παρεμβάσεις.

Η χρήση LAN είναι ιδανική για συσκευές που απαιτούν συνεχή λειτουργία, μεγάλο εύρος μετάδοσης δεδομένων ή κριτική σημασία για την ασφάλεια, όπως οι αυτοματισμοί θέρμανσης ή οι συσκευές παρακολούθησης.[1]



Σχήμα 2.1: Τοπολογίες Δικτύων [1.α]

2.1.2 Wireless Local Area Network (WLAN)

Το WLAN (Wireless Local Area Network) αποτελεί την ασύρματη εκδοχή του LAN, δηλαδή ένα τοπικό δίκτυο που επιτρέπει τη σύνδεση συσκευών μέσω Wi-Fi χωρίς καλώδια. Η πιο κοινή μορφή WLAN είναι το οικιακό δίκτυο Wi-Fi, που βασίζεται σε τεχνολογίες του προτύπου IEEE 802.11. Στο έξυπνο σπίτι, το WLAN χρησιμοποιείται για τη σύνδεση κινητών τηλεφώνων, tablet, έξυπνων τηλεοράσεων, ασύρματων καμερών, θερμοστατών, διακοπών και πλήθους άλλων IoT συσκευών. Παρόλο που προσφέρει μεγάλη ευελιξία και ευκολία εγκατάστασης, το WLAN υστερεί έναντι του ενσύρματου LAN ως προς την ασφάλεια, τη σταθερότητα και την αντοχή σε παρεμβολές, ιδίως όταν υπάρχουν πολλές ταυτόχρονες συνδέσεις ή εμπόδια στον χώρο. Εξαιτίας της φύσης του, το WLAN απαιτεί καλή διαχείριση καναλιών, χρήση ισχυρού κρυπτογραφικού πρωτοκόλλου (όπως WPA3) και συχνή επιτήρηση για εντοπισμό ευπαθειών. Το Wi-Fi, ακρωνύμιο του “Wireless Fidelity”, άρχισε να αναπτύσσεται στις αρχές της δεκαετίας του 1990, ενώ το πρώτο εμπορικό πρότυπο, το IEEE 802.11, εγκρίθηκε το 1997. Η αναγκαιότητά της αναδείχθηκε από την αυξανόμενη ανάγκη για κινητικότητα και την επιθυμία για ευκολότερη πρόσβαση σε δικτυακούς πόρους χωρίς τους περιορισμούς που επιβάλλει η ενσύρματη σύνδεση. Η αποδοχή της τεχνολογίας Wi-Fi εκτοξεύθηκε στις αρχές του 21ου αιώνα, καθιερώνοντάς την ως το πρότυπο για οικιακά και επαγγελματικά τοπικά ασύρματα δίκτυα. [1]

Το Wi-Fi λειτουργεί σε δύο κύριες ζώνες συχνοτήτων: Στα 2.4 GHz και στα 5 GHz, ενώ πιο πρόσφατα, με την εισαγωγή του Wi-Fi 6E και Wi-Fi 7, αξιοποιείται και το φάσμα των 6 GHz. Η ζώνη των 2.4 GHz προσφέρει μεγαλύτερη εμβέλεια, αλλά μικρότερες ταχύτητες και μεγαλύτερη ευαισθησία σε παρεμβολές, λόγω του κορεσμού της. Αντίθετα, η ζώνη των 5 GHz παρέχει υψηλότερες ταχύτητες και λιγότερες παρεμβολές, αλλά με μικρότερη εμβέλεια και δυσκολία διείσδυσης σε τοίχους και εμπόδια. Η επιλογή και χρήση καναλιών εντός αυτών των φασματικών ζωνών είναι κρίσιμη για τη σωστή λειτουργία ενός Wi-Fi δικτύου. Στο φάσμα των 2.4 GHz υπάρχουν 13 επικαλυπτόμενα κανάλια (στην Ευρώπη), εκ των οποίων μόνο τα 1, 6 και 11 είναι πλήρως μη επικαλυπτόμενα, κάτι που είναι απαραίτητο για την αποφυγή παρεμβολών. Στα 5 GHz, τα διαθέσιμα κανάλια είναι περισσότερα και κατανομημένα πιο αραιά, κάτι που διευκολύνει την αποδοτικότερη χρήση τους, ιδίως σε πυκνοκατοικημένα αστικά περιβάλλοντα.

Η διαδικασία σύνδεσης μίας συσκευής σε ένα Wi-Fi δίκτυο περιλαμβάνει ένα αρχικό handshake μέσω του λεγόμενου 4-way handshake, το οποίο διασφαλίζει την αυθεντικοποίηση του χρήστη και την κρυπτογράφηση της επικοινωνίας. Πιο συγκεκριμένα, το access point και η συσκευή-πελάτης ανταλλάσσουν πληροφορίες σχετικές με τα κλειδιά κρυπτογράφησης μέσω του πρωτοκόλλου WPA (Wi-Fi Protected Access), το οποίο στις σύγχρονες εκδόχές του (WPA2 & WPA3) χρησιμοποιεί εξελιγμένες μεθόδους όπως το AES (Advanced Encryption Standard) και τον μηχανισμό SAE (Simultaneous Authentication of Equals) για την πρόληψη

επιθέσεων τύπου dictionary ή man-in-the-middle. Η συνολική ασφάλεια του Wi-Fi βασίζεται όχι μόνο στα πρωτόκολλα αυθεντικοποίησης και κρυπτογράφησης αλλά και στην κατάλληλη ρύθμιση του εξοπλισμού. Εσφαλμένες ρυθμίσεις ή η χρήση παρωχημένων μηχανισμών όπως το WEP (Wired Equivalent Privacy), που έχει αποδειχθεί ευάλωτο, μπορούν να εκθέσουν το δίκτυο σε σοβαρούς κινδύνους.

Στο τεχνικό επίπεδο, το Wi-Fi βασίζεται στο μοντέλο CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), προκειμένου να διαχειριστεί τη χρήση του ασύρματου μέσου μετάδοσης. Πριν μία συσκευή εκπέμψει δεδομένα, "ακούει" το κανάλι για πιθανή δραστηριότητα και, εφόσον αυτό είναι ελεύθερο, ξεκινά τη μετάδοση. Αν διαπιστωθεί συμφόρηση, καθυστερεί με τυχαίο τρόπο τη μετάδοση ώστε να μειωθεί η πιθανότητα σύγκρουσης πακέτων δεδομενογραμμάτων. Τα πρότυπα του Wi-Fi έχουν εξελιχθεί σημαντικά με το πέρασμα του χρόνου. Ξεκινώντας από το 802.11b με μέγιστες ταχύτητες 11 Mbps, ακολούθησαν το 802.11a (54 Mbps, στα 5 GHz), το 802.11g (54 Mbps, στα 2.4 GHz), το 802.11n (Wi-Fi 4, έως 600 Mbps), το 802.11ac (Wi-Fi 5, έως και 3.5 Gbps) και το 802.11ax (Wi-Fi 6/6E), που εισήγαγε τεχνικές όπως το OFDMA (Orthogonal Frequency Division Multiple Access) και το MU-MIMO (Multi User - Multiple Input Multiple Output) για ταυτόχρονη εξυπηρέτηση πολλαπλών χρηστών. Το Wi-Fi 7 (802.11be), το οποίο βρίσκεται σε φάση υιοθέτησης, αναμένεται να φέρει ταχύτητες άνω των 30 Gbps, χαμηλό latency και προηγμένες δυνατότητες διαχείρισης φάσματος και ταυτοχρόνα να αξιοποιήσει την τεχνολογία MLO (Multi Link Operation) όπου η συγκεκριμένη, επιτρέπει την ταυτόχρονη χρήση των τριών διαφορετικών συχνοτήτων παράλληλα, μεγιστοποιώντας το διαθέσιμο εύρος ζώνης.

Όπως αναφέραμε παραπάνω, η τεχνολογία Wi-Fi έχει εξελιχθεί μέσα από μια σειρά προτύπων του IEEE 802.11, τα οποία καθορίζουν τις τεχνικές παραμέτρους και τις δυνατότητες κάθε γενιάς ασύρματης δικτύωσης. Κάθε πρότυπο εισάγει βελτιώσεις ως προς τη χωρητικότητα, τη σταθερότητα, την εμβέλεια, την πολυπλεξία χρηστών και, φυσικά, την ασφάλεια. Το 802.11b, που κυκλοφόρησε το 1999, αποτέλεσε το πρώτο πρότυπο μαζικής αποδοχής, προσφέροντας ταχύτητες έως 11 Mbps στη ζώνη των 2.4 GHz. Την ίδια χρονιά εμφανίστηκε και το 802.11a, το οποίο λειτουργούσε στα 5 GHz και προσέφερε υψηλότερες ταχύτητες (έως 54 Mbps), αλλά με μικρότερη διείσδυση. Ακολούθησε το 802.11g (2003), το οποίο επανέφερε την ταχύτητα των 54 Mbps στη ζώνη των 2.4 GHz με καλύτερη συμβατότητα και σταθερότητα. Η σημαντική αναβάθμιση ήρθε με το 802.11n (γνωστό και ως Wi-Fi 4, 2009), που εισήγαγε τη χρήση πολλαπλών κεραιών (MIMO – Multiple Input Multiple Output), φτάνοντας θεωρητικά τα 600 Mbps, και παρέχει υποστήριξη τόσο για τα 2.4 GHz όσο και για τα 5 GHz. Το 802.11ac (Wi-Fi 5, 2013) εστίασε αποκλειστικά στην μπάντα των 5 GHz και εισήγαγε τεχνολογίες όπως το beamforming και το MU-MIMO (Multi-User MIMO), επιτρέποντας σε routers να εξυπηρετούν ταυτόχρονα πολλαπλές συσκευές χωρίς σημαντική απώλεια ταχύτητας. Οι ταχύτητες εκτοξεύθηκαν έως και 3.5 Gbps, σε ιδανικές συνθήκες. Η πιο δραστική και μοντέρνα αλλαγή ήρθε με το 802.11ax (Wi-Fi 6, 2019), που υποστηρίζει και τις δύο ζώνες (2.4 και 5 GHz), αλλά

και επιπλέον δυνατότητες όπως το OFDMA (Orthogonal Frequency Division Multiple Access), το Target Wake Time (για εξοικονόμηση ενέργειας σε IoT συσκευές) και εξελιγμένο MU-MIMO για ταυτόχρονη εξυπηρέτηση πολλών συσκευών. Το Wi-Fi 6E είναι επέκταση του Wi-Fi 6 στο απελευθερωμένο φάσμα των 6 GHz, που προσφέρει ευρύτερα κανάλια και μηδενικές παρεμβολές από παλιές συσκευές, γεγονός που το καθιστά ιδανικό για εφαρμογές με υψηλές απαιτήσεις. Το πιο πρόσφατο πρότυπο, το 802.11be (Wi-Fi 7), που βρίσκεται στη διαδικασία υλοποίησης, προσφέρει εντυπωσιακές ταχύτητες άνω των 30 Gbps, υποστηρίζει 320 MHz κανάλια, 4K-QAM για μεγαλύτερη πυκνότητα πληροφορίας και τεχνολογίες multi-link operation (MLO), οι οποίες επιτρέπουν σε μια συσκευή να συνδέεται ταυτόχρονα σε διαφορετικές συχνότητες για αυξημένη αξιοπιστία και απόδοση. Η προσέγγιση του Wi-Fi 7 δείχνει ξεκάθαρα τη στροφή της τεχνολογίας προς εφαρμογές με υψηλό throughput, ελάχιστο latency και υψηλή σταθερότητα

Όσον αφορά την ασφάλεια, το πρωτόκολλο WEP (Wired Equivalent Privacy) υπήρξε η αρχική μορφή προστασίας, αλλά γρήγορα κρίθηκε ανεπαρκές λόγω σοβαρών αδυναμιών. Το WPA (Wi-Fi Protected Access) ήρθε ως ενδιάμεση λύση, μέχρι να θεσπιστεί το WPA2, το οποίο για χρόνια αποτελούσε το βασικό πρότυπο ασφάλειας για τα περισσότερα Wi-Fi δίκτυα. Το WPA2 εισήγαγε το AES (Advanced Encryption Standard) για ισχυρή συμμετρική κρυπτογράφηση, αλλά διατηρούσε κάποιες δομικές ευπάθειες, κυρίως στον τρόπο αυθεντικοποίησης. Το WPA3, που ανακοινώθηκε το 2018, αποτελεί τη νεότερη και πιο ασφαλή μορφή προστασίας. Εισάγει το SAE (Simultaneous Authentication of Equals) ως μηχανισμό handshake, ο οποίος είναι ανθεκτικός σε offline επιθέσεις λεξικού (dictionary attacks), καθώς και forward secrecy, που εξασφαλίζει ότι ακόμη και αν κάποιος αποκτήσει ένα session key, δεν μπορεί να αποκρυπτογραφήσει προηγούμενη κίνηση. Επιπλέον, το WPA3 υποστηρίζει το Enhanced Open, που προσφέρει κρυπτογράφηση ακόμα και σε ανοιχτά δίκτυα μέσω του Opportunistic Wireless Encryption (OWE), και προσθέτει μέτρα για την αποτροπή επιθέσεων μέσω IoT συσκευών.

IEEE Standard	WiFi Gen	Year	Frequency	Max PHY Data Rate	Max Range
802.11	-	1997	2.4 GHz	2 Mbps	20m (indoor) 100m (outdoor)
802.11 a	-	1999	5 GHz	54 Mbps	35m (indoor) 120m (outdoor)
802.11 b	-	1999	2.4 GHz	11 Mbps	35m (indoor) 140m (outdoor)
802.11 g	-	2003	2.4 GHz	54 Mbps	38m (indoor) 140m (outdoor)
802.11 n	WiFi 4	2009	2.4/5 GHz	600 Mbps	70m (indoor) 250m (outdoor)
802.11 ac	WiFi 5	2013	5 GHz	6.9 Gbps	35m (indoor)
802.11 ad	-	2012	60 GHz	8.1 Mbps	3.3m (indoor)
802.11 ah	-	2017	Sub 1 GHz	347 Mbps	1km
802.11 ax	WiFi 6	2021	2.4/5/6 GHz	9.6 Gbps	30m (indoor) 120m (outdoor)
802.11 ay	-	2021	60 GHz	303 Gbps	10m (indoor) 100m (outdoor)
802.11 be	WiFi 7	2024	2.4/5/6 GHz	46.1 Gbps	30m (indoor) 120m (outdoor)

Εικόνα 2.2: Πρωτόκολλα Επικοινωνίας Wi-fi [2.α]

2.1.3.: Wide Area Network (WAN):

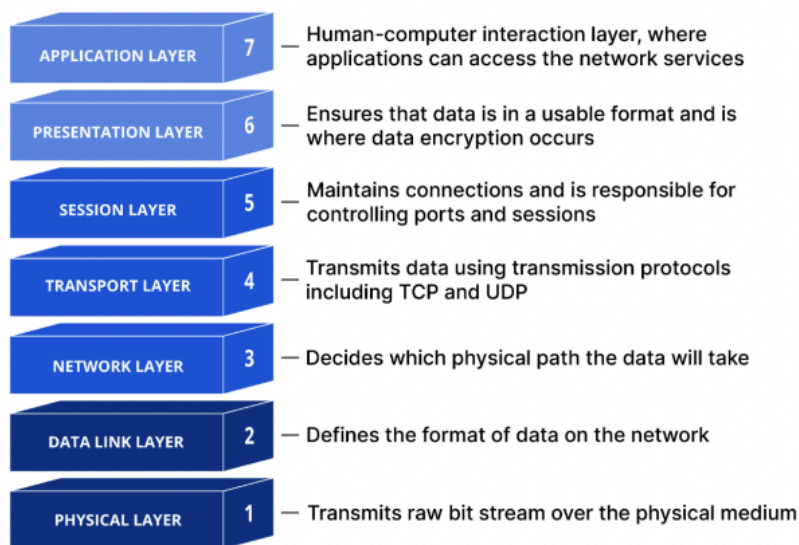
Το WAN (Wide Area Network) είναι ένα δίκτυο μεγάλης εμβέλειας, το οποίο διασυνδέει επιμέρους LAN και WLAN δίκτυα σε γεωγραφικά κατανομημένα σημεία, και το πλέον χαρακτηριστικό παράδειγμα του είναι το διαδίκτυο (Internet). Το WAN επιτρέπει την απομακρυσμένη πρόσβαση στο οικιακό δίκτυο από οποιοδήποτε σημείο του πλανήτη, υπό την προϋπόθεση ύπαρξης κατάλληλης δικτυακής υποδομής. Για μια έξυπνη κλειδαριά, το WAN είναι ζωτικής σημασίας για τη διάγνωση προβλημάτων εξ αποστάσεως, την ειδοποίηση μέσω cloud και γενικότερα για οποιαδήποτε μορφή έξυπνης απομακρυσμένης διαχείρισης. Η πρόκληση στη χρήση WAN για οικιακά δίκτυα είναι η ασφάλεια, καθώς η έκθεση συστημάτων στο διαδίκτυο δημιουργεί κινδύνους υποκλοπής ή επιθέσεων. Επομένως, η χρήση τεχνολογιών που θα αναφέρουμε παρακάτω είναι κρίσιμη για την αποφυγή αυτών των κινδύνων.

2.2 Μοντέλο OSI

Το μοντέλο OSI (Open Systems Interconnection) είναι ένα θεωρητικό πλαίσιο που δημιουργήθηκε από τον ISO για να περιγράψει και να τυποποιήσει τον τρόπο με τον οποίο επικοινωνούν τα δίκτυα υπολογιστών, χωρίζοντας τη διαδικασία σε επτά διακριτά επίπεδα. Στο πρώτο επίπεδο, το Physical Layer (Φυσικό Επίπεδο), μεταδίδονται τα πραγματικά bits μέσω φυσικών μέσων όπως καλώδια Ethernet, οπτικές ίνες ή ασύρματα σήματα, με παραδείγματα προτύπων όπως το Ethernet 100BASE-T ή το Wi-Fi 802.11. Στο δεύτερο επίπεδο, το Data Link Layer (Επίπεδο Σύνδεσης Δεδομένων), γίνεται η οργάνωση των δεδομένων σε frames και η διαχείριση της φυσικής διεύθυνσης (MAC address), με χαρακτηριστικά πρωτόκολλα όπως το Ethernet και το PPP (Point-to-Point Protocol). Το τρίτο επίπεδο, το Network Layer (Δικτυακό Επίπεδο), ασχολείται με τη δρομολόγηση των πακέτων μεταξύ διαφορετικών δικτύων, χρησιμοποιώντας πρωτόκολλα όπως το IPv4 και το IPv6. Στο τέταρτο επίπεδο, το Transport Layer (Επίπεδο Μεταφοράς), εξασφαλίζεται η αξιόπιστη ή μη αξιόπιστη μεταφορά των δεδομένων μεταξύ εφαρμογών, με πρωτόκολλα όπως το TCP (Transmission Control Protocol) για αξιόπιστη μεταφορά και το UDP (User Datagram Protocol) για ταχύτερη αλλά μη εγγυημένη μεταφορά. Το πέμπτο επίπεδο, το Session Layer (Επίπεδο Συνεδρίας), διαχειρίζεται τη δημιουργία, διατήρηση και τερματισμό συνεδριών επικοινωνίας, με παραδείγματα όπως το NetBIOS ή το RPC (Remote Procedure Call). Το έκτο επίπεδο, το Presentation Layer (Επίπεδο Παρουσίασης), φροντίζει για τη μορφοποίηση, τη συμπίεση και την κρυπτογράφηση των δεδομένων, με πρωτόκολλα και τεχνολογίες όπως το SSL/TLS για ασφάλεια ή το MPEG για συμπίεση πολυμέσων. Τέλος, το έβδομο επίπεδο, το Application Layer (Επίπεδο Εφαρμογής), είναι το πλησιέστερο στον χρήστη και περιλαμβάνει τα πρωτόκολλα που επιτρέπουν την άμεση αλληλεπίδραση, όπως το HTTP/HTTPS για ιστό, το SMTP για email και το FTP για μεταφορά αρχείων.

Η λογική του μοντέλου OSI είναι ότι κάθε επίπεδο προσφέρει συγκεκριμένες υπηρεσίες στο επίπεδο από πάνω του και χρησιμοποιεί τις λειτουργίες του επιπέδου από κάτω, επιτρέποντας έτσι τη modular ανάπτυξη και την ευκολότερη διάγνωση προβλημάτων. Για παράδειγμα, ένα πρόβλημα στο φυσικό επίπεδο (π.χ. κομμένο καλώδιο) μπορεί να απομονωθεί χωρίς να επηρεάσει την κατανόηση της λειτουργίας στα υψηλότερα επίπεδα, ενώ ένα πρόβλημα στο επίπεδο εφαρμογής (π.χ. σφάλμα HTTP) μπορεί να εντοπιστεί ανεξάρτητα από την υποδομή μεταφοράς.

Το μοντέλο OSI, πέρα από το ότι είναι θεωρητικό και δεν εφαρμόζεται αυτούσιο στην πράξη, λειτουργεί ως θεμελιώδης οδηγός κατανόησης της αρχιτεκτονικής των δικτύων και της διαστρωμάτωσης των λειτουργιών. Κάθε επίπεδο είναι αυστηρά καθορισμένο ώστε να επικοινωνεί μόνο με το ακριβώς πάνω ή κάτω από αυτό, κάτι που επιτρέπει την ανεξαρτησία της τεχνολογίας. Για παράδειγμα, μπορείς να αλλάξεις το φυσικό μέσο (από χαλκό σε οπτική ίνα) χωρίς να αλλάξεις τίποτα στα υψηλότερα επίπεδα, ή να αλλάξεις ένα πρωτόκολλο εφαρμογής (π.χ. από FTP σε HTTP) χωρίς να επηρεάσεις το πώς γίνεται η μετάδοση στο φυσικό επίπεδο.



Σχήμα 2.3: Επίπεδα του μοντέλου OSI [3.α]

Στην πράξη, το μοντέλο OSI συνδέεται στενά με το μοντέλο TCP/IP, το οποίο χρησιμοποιείται ευρέως στο διαδίκτυο. Στο TCP/IP, τα επτά επίπεδα του OSI συμπύσσονται σε τέσσερα: Network Access (που περιλαμβάνει τα επίπεδα Physical και Data Link), Internet (που αντιστοιχεί στο Network Layer), Transport (ίδιο με του OSI) και Application (που περιλαμβάνει τα τρία υψηλότερα επίπεδα του OSI: Session, Presentation, Application). Η αντιστοίχιση αυτή βοηθά στην κατανόηση του γιατί πολλά πρωτόκολλα αναφέρονται ως “Layer 3” ή “Layer 4” — για παράδειγμα, το IPv4 είναι καθαρά πρωτόκολλο Layer 3, ενώ το TCP είναι Layer 4.

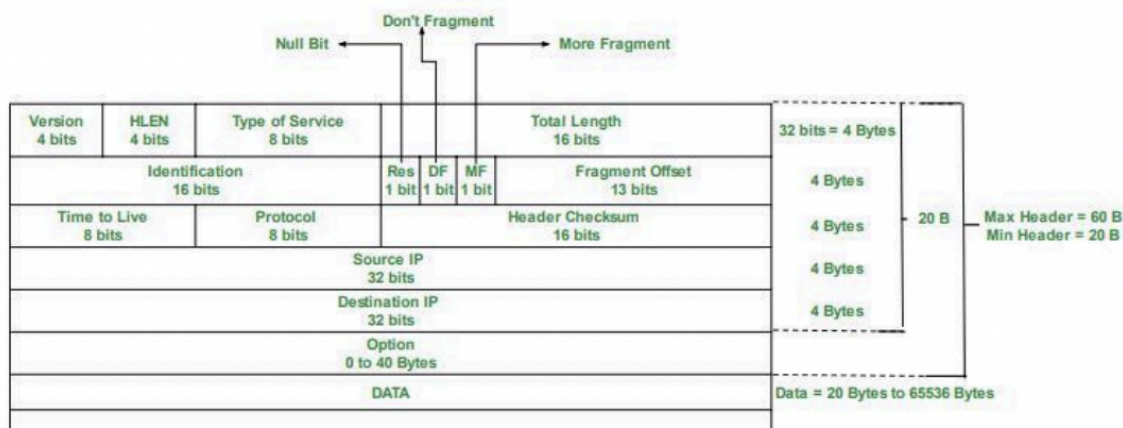
Ένα από τα σημαντικά πλεονεκτήματα της ύπαρξης του OSI είναι η τυποποίηση και η δυνατότητα troubleshooting. Στην ανάλυση προβλημάτων, οι τεχνικοί συχνά ακολουθούν την προσέγγιση “From Layer 1 to Layer 7”, δηλαδή ελέγχουν πρώτα το φυσικό μέσο (καλώδια, σήμα), μετά τη συνδεσιμότητα (MAC addresses, switches), στη συνέχεια την IP δρομολόγηση, και προχωρούν μέχρι το επίπεδο εφαρμογής. Για παράδειγμα, αν ένας χρήστης δεν μπορεί να φορτώσει μια ιστοσελίδα, η ανάλυση μπορεί να ξεκινήσει από το Layer 1 (είναι συνδεδεμένο το καλώδιο;) και να φτάσει μέχρι το Layer 7 (είναι σωστά ρυθμισμένος ο web server;).

Επιπλέον, το OSI προσφέρει σαφή διαχωρισμό ρόλων σε συσκευές δικτύου. Τα switches λειτουργούν κυρίως στο Layer 2, οι routers στο Layer 3, τα firewalls μπορεί να λειτουργούν από Layer 3 μέχρι Layer 7 (ανάλογα με την πολυπλοκότητα), ενώ οι load balancers δουλεύουν κυρίως στα ανώτερα επίπεδα. Ορισμένες συσκευές, όπως τα Layer 3 switches, γεφυρώνουν λειτουργίες δύο επιπέδων

2.3: Internet Protocol Version 4

Το IPv4 (Internet Protocol version 4) αποτελεί την τέταρτη έκδοση του πρωτοκόλλου διαδικτύου και είναι μέχρι σήμερα η πλέον διαδεδομένη μορφή διευθυνσιοδότησης σε παγκόσμιο επίπεδο. Σχεδιάστηκε αρχικά τη δεκαετία του 1980 στο πλαίσιο του ARPANET και τέθηκε σε πλήρη λειτουργία κατά τη διάρκεια της δεκαετίας του 1990, αποτελώντας τη βάση για τη δημιουργία του σύγχρονου Διαδικτύου. Το IPv4 είναι ένα πρωτόκολλο δικτύου του επιπέδου δικτύου του μοντέλου TCP/IP, και βασική του λειτουργία είναι η παροχή μοναδικών διευθύνσεων σε κάθε συσκευή που συμμετέχει σε ένα δίκτυο, έτσι ώστε να είναι δυνατή η μετάδοση πακέτων μεταξύ αποστολέα και παραλήπτη.

Μια διεύθυνση IPv4 είναι 32-bit και συνήθως αναπαρίσταται με τέσσερις δεκαδικούς αριθμούς (οκτάδες) διαχωρισμένους με τελείες, όπως για παράδειγμα: 192.168.1.1. Το γεγονός ότι υπάρχουν 32 δυαδικά ψηφία συνεπάγεται θεωρητικά τη δυνατότητα ύπαρξης περίπου 4,3 δισεκατομμυρίων μοναδικών διευθύνσεων. Ωστόσο, στην πράξη, εξαιτίας της ανάγκης για ομαδοποιήσεις, broadcast διευθύνσεις και κρατημένες διευθύνσεις για ειδικούς σκοπούς, ο αριθμός των πραγματικά διαθέσιμων διευθύνσεων είναι σημαντικά μικρότερος. Αυτό το πρόβλημα της εξάντλησης των διαθέσιμων διευθύνσεων υπήρξε η αφορμή για την ανάπτυξη του IPv6, το οποίο χρησιμοποιεί 128-bit διευθύνσεις. Για να αντιμετωπιστεί το ζήτημα της έλλειψης διαθέσιμων IPv4 διευθύνσεων χωρίς άμεση μετάβαση στο IPv6, εφαρμόστηκε η τεχνική του NAT (Network Address Translation). Το NAT επιτρέπει σε πολλές συσκευές μέσα σε ένα τοπικό δίκτυο (LAN) να χρησιμοποιούν ιδιωτικές, μη δρομολογήσιμες διευθύνσεις (όπως 192.168.x.x, 10.x.x.x, 172.16.x.x έως 172.31.x.x) και να μοιράζονται μία μόνο δημόσια διεύθυνση IP για να επικοινωνούν με το Διαδίκτυο. Αυτό επιτυγχάνεται με τη μετάφραση των ιδιωτικών διευθύνσεων και των αντίστοιχων θυρών (ports) σε μία δημόσια διεύθυνση μέσω ενός NAT router ή firewall. Υπάρχουν διάφοροι τύποι NAT. Το πιο διαδεδομένο είναι το PAT (Port Address Translation), γνωστό και ως NAT Overload, όπου πολλές εσωτερικές συσκευές μοιράζονται μία δημόσια IP και διαχωρίζονται μεταξύ τους μέσω διαφορετικών αριθμών θύρας. Υπάρχουν επίσης το Static NAT, όπου μια εσωτερική διεύθυνση αντιστοιχίζεται μόνιμα σε μια δημόσια, και το Dynamic NAT, όπου οι εσωτερικές διευθύνσεις αντιστοιχίζονται προσωρινά από ένα pool δημόσιων διευθύνσεων. Η ύπαρξη του NAT επιφέρει σημαντικές επιπτώσεις στην αρχιτεκτονική του Διαδικτύου. Από τη μία πλευρά, επιτρέπει την εξοικονόμηση δημόσιων διευθύνσεων και προσφέρει ένα επίπεδο ασφάλειας, καθώς οι εσωτερικές συσκευές δεν είναι άμεσα προσβάσιμες από το εξωτερικό. Από την άλλη, καθιστά πιο περίπλοκη τη δημιουργία συνδέσεων προς τα μέσα (inbound connections), απαιτώντας τεχνικές όπως port forwarding ή tunneling για να καταστεί δυνατή η πρόσβαση από το Διαδίκτυο σε εσωτερικές υπηρεσίες. Επίσης, δυσκολεύει την υλοποίηση end-to-end συνδέσεων, κάτι που ήταν θεμελιώδης στόχος στο αρχικό όραμα του Διαδικτύου.



Σχήμα 2.4: Ένα Δεδομενογράμμο ενός πακέτου IPv4 [4.α]

2.4: Θύρες δικτύου

Οι θύρες δικτύου, γνωστές ως network ports, αποτελούν αναπόσπαστο στοιχείο της αρχιτεκτονικής επικοινωνίας στο Διαδίκτυο και στηρίζονται στο μοντέλο TCP/IP. Είναι αριθμητικοί αναγνωριστές μεγέθους 16-bit, που κυμαίνονται από το 0 έως το 65535, και χρησιμοποιούνται για να κατευθύνουν 18 την κυκλοφορία δεδομένων προς συγκεκριμένες εφαρμογές ή υπηρεσίες που εκτελούνται σε μια συσκευή δικτύου. Κάθε θύρα αντιστοιχεί σε ένα συγκεκριμένο πρωτόκολλο ή υπηρεσία, επιτρέποντας σε έναν υπολογιστή ή server να εξυπηρετεί πολλές εφαρμογές ταυτόχρονα, μέσω της ίδιας IP διεύθυνσης. Η σύνταξη που χρησιμοποιείται για να καθοριστεί μια συγκεκριμένη σύνδεση είναι της μορφής IP:θύρα, όπως για παράδειγμα 192.168.1.100:8123, όπου η διεύθυνση παραπέμπει στο web interface του Home Assistant, μιας πλατφόρμας έξυπνου αυτοματισμού.

Οι θύρες διακρίνονται σε τρεις βασικές κατηγορίες: Οι well-known ports, οι οποίες καταλαμβάνουν το εύρος από 0 έως 1023 και προορίζονται για θεμελιώδεις υπηρεσίες όπως το HTTP (θύρα 80), το HTTPS (443), το SSH (22), και το DNS (53), οι registered ports, που βρίσκονται στο εύρος 1024 έως 49151 και προορίζονται για εφαρμογές τρίτων κατασκευαστών ή προσωποποιημένες υπηρεσίες και οι dynamic ή ephemeral ports, από 49152 έως 65535, που χρησιμοποιούνται προσωρινά για την εξυπηρέτηση εξερχόμενων συνδέσεων. Στο πλαίσιο του οικιακού αυτοματισμού, μια ιδιαίτερα χαρακτηριστική θύρα είναι η 8123, η οποία αποτελεί την προεπιλεγμένη πόρτα για την πρόσβαση στη διεπαφή του Home Assistant. Μέσω αυτής της θύρας, οι χρήστες μπορούν να αλληλεπιδράσουν με το σύστημα αυτοματισμού τους μέσω browser ή εφαρμογών τρίτων. Οι θύρες είναι επίσης ουσιώδεις για την τεχνολογία προώθησης θυρών (port forwarding), όπου ο δρομολογητής ανακατευθύνει την εισερχόμενη κίνηση από το διαδίκτυο προς μια συγκεκριμένη συσκευή στο εσωτερικό δίκτυο, επιτρέποντας π.χ. την απομακρυσμένη πρόσβαση σε κάμερες, server ή το Home Assistant. Οι θύρες δικτύου δεν αποτελούν φυσικές διεπαφές, αλλά λογικούς αριθμητικούς προσδιορισμούς που χρησιμοποιούνται στο επίπεδο μεταφοράς του μοντέλου

TCP/IP, κυρίως από τα πρωτόκολλα TCP (Transmission Control Protocol) και UDP (User Datagram Protocol). Με απλά λόγια, κάθε φορά που μια συσκευή συνδέεται σε μια υπηρεσία στο διαδίκτυο ή στο τοπικό δίκτυο, χρησιμοποιεί μια συγκεκριμένη θύρα για να επικοινωνήσει. Το λειτουργικό σύστημα παρακολουθεί αυτές τις θύρες, και κάθε εφαρμογή που χρειάζεται να δέχεται δεδομένα πρέπει να «ακούει» σε μία ή περισσότερες από αυτές.

Η διαχείριση των θυρών έχει ιδιαίτερη σημασία και για τους διαχειριστές συστημάτων, καθώς μπορεί να συνδεθεί με πολιτικές που εφαρμόζονται στα τείχη προστασίας (firewall). Για παράδειγμα, ένα τείχος προστασίας μπορεί να επιτρέπει μόνο τις θύρες 443 και 22 για ασφαλείς συνδέσεις (HTTPS και SSH αντίστοιχα), ενώ μπλοκάρει όλες τις υπόλοιπες, περιορίζοντας την επιφάνεια πιθανών επιθέσεων. Παράλληλα, πολλά προγράμματα ή υπηρεσίες καθορίζουν αυτόματα τις θύρες στις οποίες λειτουργούν. Αν μια θύρα είναι ήδη κατειλημμένη, η εφαρμογή θα αποτύχει να ξεκινήσει ή θα πρέπει να επαναρυθμιστεί με διαφορετικό αριθμό θύρας. Μια ενδιαφέρουσα πτυχή αφορά τις θύρες UDP, οι οποίες συχνά χρησιμοποιούνται για υπηρεσίες που απαιτούν χαμηλή καθυστέρηση, όπως τα VoIP (Voice over IP), το video streaming και τα διαδικτυακά παιχνίδια. Σε αντίθεση με το TCP που εξασφαλίζει αξιόπιστη μετάδοση, το UDP προσφέρει ταχύτητα με μικρότερο έλεγχο, συνεπώς χρειάζεται ιδιαίτερη προσοχή στη χρήση του. Η τεχνολογία των θυρών διαδραματίζει επίσης κεντρικό ρόλο στο Network Address Translation (NAT), όπου πολλές συσκευές σε ένα τοπικό δίκτυο χρησιμοποιούν μια κοινή δημόσια IP. Το NAT διαχειρίζεται αυτήν την πολυπλοκότητα αντιστοιχίζοντας τις εξερχόμενες συνδέσεις σε συγκεκριμένες θύρες, έτσι ώστε οι απαντήσεις από το διαδίκτυο να φτάσουν πίσω στη σωστή συσκευή. Αυτή η τεχνική επιτρέπει τη συνύπαρξη πολλών συσκευών πίσω από έναν μόνο δρομολογητή και είναι απολύτως εξαρτημένη από τη χρήση θυρών. Τέλος, σε πιο σύγχρονες υλοποιήσεις και περιβάλλοντα, όπως τα containers ή τα virtual machines, η διαχείριση θυρών είναι κρίσιμη για τη δρομολόγηση της εσωτερικής και εξωτερικής κυκλοφορίας. Για παράδειγμα, σε ένα Docker container που τρέχει Home Assistant, μπορεί να γίνει map η εξωτερική θύρα 8123 του host στη θύρα 8123 του container, επιτρέποντας πλήρη πρόσβαση στη διεπαφή διαχείρισης του αυτοματισμού από το τοπικό ή και απομακρυσμένο δίκτυο.

Port	Protocol	Name	Port	Protocol	Name	Port	Protocol	Name
7	TCP/UDP	echo	520	UDP	rip	2103	TCP/UDP	zephyr-clt
9	TCP/UDP	discard	521	UDP	ripng (ipv6)	2104	TCP/UDP	zephyr-hm
19	TCP/UDP	chargen	540	TCP	uucp	2222	TCP	directadmin
20	TCP/SCTP	ftp-data	546	TCP/UDP	dhcpv6	2401	TCP	cvspsrvr
21	TCP/UDP/SCTP	ftp	547	TCP/UDP	dhcpv6	2483	TCP/UDP	oracle
22	TCP/UDP/SCTP	ssh/scp/sftp	548	TCP	afp	2484	TCP/UDP	oracle
23	TCP	telnet	554	TCP/UDP	rtsp	2809	TCP/UDP	corbaloc
25	TCP	smtp	560	UDP	rmonitor	2967	TCP/UDP	symantec av
42	TCP/UDP	wins replication	563	TCP/UDP	nntp over tls/ssl	3128	TCP/UDP	http proxy
43	TCP/UDP	whois	587	TCP	smtp/submission	3222	TCP/UDP	glbp
49	UDP	tacacs	591	TCP	filemaker	3260	TCP/UDP	iscsi target
53	TCP/UDP	dns	593	TCP/UDP	microsoft dcom	3306	TCP/UDP	mysql
67	UDP	dhcp/bootp	596	TCP/UDP	smsd	3389	TCP	rdp
68	UDP	dhcp/bootp	631	TCP	ipp	3689	TCP	daap
69	UDP	httpt	636	TCP/UDP	ldap over tls/ssl	3690	TCP/UDP	svn
70	TCP	gopher	639	TCP	msdp (pim)	4321	TCP	rwhois
79	TCP	finger	646	TCP/UDP	ldp (mpls)	4333	TCP	msql
80	TCP/UDP/SCTP	http	691	TCP	microsoft exchange	4500	UDP	ipsec nat traversal
88	TCP/UDP	kerberos	860	TCP	iscsi	4899	TCP	radmin
101	TCP	hostname	873	TCP	rsync	5000	TCP	upnp
102	TCP	microsoft exchange iso-tsap	902	TCP/UDP	vmware server	5001	TCP	iperf
110	TCP	pop3	989	TCP	ftps	5004-5005	UDP	rtprtsp
113	TCP	ident	990	TCP	ftps	5060	TCP/UDP	sip
119	TCP	nntp (usenet)	992	TCP/UDP	telnet	5061	TCP	sip-tls
123	UDP	ntp	993	TCP	imap over ssl (imaps)	5222-5223	TCP	xmpp
135	TCP/UDP	microsoft rpc epmap	995	TCP/UDP	pop3 over ssl (pop3s)	5353	UDP	mdns
137	TCP/UDP	netbios-ns	1025	TCP	microsoft rpc	5432	TCP	postgresql
138	TCP/UDP	netbios-dgm	1080	TCP/UDP	socks	5800	TCP	vnc over http
139	TCP/UDP	netbios-ssn	1194	TCP/UDP	openvpn	5900-5999	TCP/UDP	rfb/vnc server
143	TCP/UDP	imap	1241	TCP/UDP	nessus	5999	TCP	cvsup
161	UDP	snmp-agents (unencrypted)	1311	TCP	dell openmanage	6000-6001	TCP	X11
162	UDP	snmp-trap (unencrypted)	1433	TCP	ms-sql-s	6129	TCP	dameware
177	UDP	xdmcp	1434	TCP/UDP	ms-sql-m	6379	TCP	redis
179	TCP	bgp	1494	TCP	ica	6588	TCP	analogx
194	UDP	irc	1512	TCP/UDP	wins	6588	TCP	http proxy
201	TCP/UDP	appletalk	1524	TCP/UDP	ingreslock	8080	TCP	http proxy
264	TCP/UDP	bgmp	1589	TCP/UDP	cisco vqp	8200	TCP/UDP	vmware server
318	TCP/UDP	tsp	1701	UDP	l2tp	8222	TCP/UDP	vmware server
381	TCP/UDP	hp openview	1719	UDP	h323gatestat	8767	UDP	Teamspeak
383	TCP/UDP	hp openview	1720	TCP	h323hostcall	9042	TCP	cassandra
389	TCP/UDP	ldap	1723	TCP/UDP	microsoft pptp	9100	TCP	pdl
411	TCP/UDP	(multiple uses)	1725	UDP	steam	9800	TCP/UDP	webdav
412	TCP/UDP	(multiple uses)	1755	TCP/UDP	mms	10161	TCP	snmp-agents
427	TCP	slp	1812	TCP/UDP	radius	10162	TCP	snmp-trap
443	TCP/UDP/SCTP	https (http over ssl)	1813	TCP/UDP	radius-acct	13720	TCP/UDP	bprd
445	TCP/UDP	microsoft ds smb	1985	UDP	hsrp	13721	TCP/UDP	bpdgm
464	TCP/UDP	kerberos	2000	TCP	cisco scdp	13724	TCP/UDP	vnedt
465	TCP	smtp over tls/ssl	2002	TCP	cisco acs	13782	TCP/UDP	bpcd
465	TCP	ssm	2008	TCP	teamspeak 3 accounting	13783	TCP/UDP	vopied
497	TCP/UDP	dantz retrospect	2010	UDP	teamspeak 3 web list	20000	TCP/UDP	usermin
500	UDP	ipsec/isakmp/ike	2049	TCP/UDP	nfs	22273	TCP/UDP	wnn6
512	TCP	rexec	2082	TCP/UDP	cpanel	23399	TCP/UDP	skype
513	TCP	riogin	2083	TCP/UDP	radsec/cpanel	25565	TCP	minecraft
514	UDP	syslog/shell	2100	TCP	amiganetfs	27017	TCP/UDP	mongodb
515	TCP	lpd/lpr	2102	TCP/UDP	zephyr-srv	33434	TCP/UDP	traceroute

NetworkProGuide.com

Σχήμα 2.5: Λίστα με τις Well-known θύρες απο τον IANA [5.α]

2.5: Zigbee

Η τεχνολογία Zigbee αποτελεί ένα πρότυπο ασύρματης επικοινωνίας το οποίο αναπτύχθηκε με σκοπό να καλύψει τις ανάγκες εφαρμογών χαμηλής ισχύος και χαμηλής κατανάλωσης, που απαιτούν σταθερή και αξιόπιστη μετάδοση μικρών ποσοτήτων δεδομένων. Δημιουργήθηκε το 2002 από την Zigbee Alliance, έναν οργανισμό που συστάθηκε με τη συνεργασία τεχνολογικών κολοσσών όπως η Philips, η Texas Instruments, η Siemens και η Motorola, μεταξύ άλλων. Το Zigbee βασίζεται στο πρότυπο IEEE 802.15.4, το οποίο καθορίζει τα φυσικά και MAC επίπεδα λειτουργίας, ενώ η Zigbee Alliance επέκτεινε το πρωτόκολλο για να καλύπτει τα ανώτερα επίπεδα, όπως η ασφάλεια, η δρομολόγηση, η δημιουργία δικτύου και η επικοινωνία εφαρμογών.

Η τεχνολογία λειτουργεί κυρίως στη συχνότητα των 2.4 GHz, η οποία είναι διαθέσιμη διεθνώς, ενώ υποστηρίζεται και η λειτουργία στις ζώνες των 868 MHz για την Ευρώπη και 915 MHz για τις Ηνωμένες Πολιτείες και την Αυστραλία. Το εύρος επικοινωνίας κυμαίνεται από 10 έως 100 μέτρα σε εσωτερικούς χώρους, ανάλογα με τα εμπόδια και τις συνθήκες, όμως χάρη στην αρχιτεκτονική τύπου mesh που χρησιμοποιεί, είναι δυνατό να επεκταθεί σημαντικά, καθώς κάθε συσκευή μπορεί να λειτουργεί και ως κόμβος αναμετάδοσης (router) για τις υπόλοιπες. Ο ρυθμός μετάδοσης δεδομένων φτάνει έως και τα 250 kbps στην συχνότητα των 2.4 GHz, ενώ είναι χαμηλότερος στις υπόλοιπες συχνότητες (40 kbps στα 915 MHz και 20 kbps στα 868 MHz), κάτι το οποίο δεν αποτελεί μειονέκτημα για τις εφαρμογές που εξυπηρετεί, δεδομένου ότι οι περισσότερες Zigbee συσκευές μεταδίδουν μικρά και σύντομα πακέτα δεδομένων, όπως είναι οι τιμές θερμοκρασίας ή η κατάσταση ενός διακόπτη. Το Zigbee χαρακτηρίζεται από εξαιρετικά χαμηλή κατανάλωση ενέργειας, γεγονός που το καθιστά ιδανικό για συσκευές που λειτουργούν με μπαταρία και απαιτούν αυτονομία πολλών μηνών ή και ετών. Η ασφάλεια αποτελεί επίσης βασικό στοιχείο της τεχνολογίας, με χρήση της κρυπτογράφησης AES 128-bit, μηχανισμών αυθεντικοποίησης και διασφάλισης της ακεραιότητας των πακέτων δεδομένων. Η τοπολογία του δικτύου είναι κατά βάση mesh, αν και υποστηρίζονται και άλλες μορφές, όπως του αστέρα ή του δέντρου. Παρά τα περιορισμένα χαρακτηριστικά του σε ό,τι αφορά το εύρος ζώνης ή την ταχύτητα μετάδοσης, το Zigbee διακρίνεται για τη σταθερότητα, την επεκτασιμότητα και την ανεξαρτησία του από το διαδίκτυο. Χρησιμοποιείται εκτενώς σε προϊόντα μεγάλων εταιρειών όπως η Philips (Hue), η Aqara και η Sonoff. Παράλληλα, το Zigbee αποτελεί μία από τις τεχνολογίες που συμβάλλουν στην μετάβαση προς το πρότυπο Matter, το οποίο αναπτύσσεται επίσης από τον ίδιο οργανισμό – πλέον γνωστό ως Connectivity Standards Alliance – και στοχεύει στη δημιουργία ενός ενιαίου διαλειτουργικού οικοσυστήματος για το Internet of Things.

Το πρωτόκολλο Zigbee αποτελεί ένα χαρακτηριστικό παράδειγμα δικτύου προσωπικής εμβέλειας (Personal Area Network – PAN) που σχεδιάστηκε ειδικά για εφαρμογές αυτοματισμού και αισθητήρων, με στόχο τη δημιουργία σταθερών, ασφαλών και αποδοτικών ασύρματων δικτύων χαμηλής κατανάλωσης ενέργειας. Ένα από τα βασικά του γνωρίσματα είναι η υποστήριξη τοπολογίας τύπου mesh,

που του επιτρέπει να σχηματίζει δυναμικά επεκτάσιμα και αξιόπιστα δίκτυα, ακόμη και σε περιβάλλοντα με πολλαπλά φυσικά εμπόδια ή παρεμβολές. Σε αντίθεση με απλούστερες αρχιτεκτονικές τύπου αστέρα, η mesh τοπολογία του Zigbee δίνει τη δυνατότητα σε κάθε κόμβο να μεταβιβάζει δεδομένα όχι μόνο προς τον προορισμό τους αλλά και μέσω ενδιάμεσων διαδρομών, εξασφαλίζοντας έτσι εφεδρικές διαδρομές και υψηλή ανθεκτικότητα σε περιπτώσεις βλαβών ή αποσύνδεσης συσκευών. Η αρχιτεκτονική ενός Zigbee δικτύου βασίζεται στην ύπαρξη τριών ειδών κόμβων: του συντονιστή (coordinator), των δρομολογητών (routers) και των τελικών συσκευών (end devices). Ο συντονιστής αποτελεί το μοναδικό σημείο που έχει την ικανότητα να δημιουργήσει και να ξεκινήσει το δίκτυο.

Είναι υπεύθυνος για την ανάθεση διευθύνσεων, τον συγχρονισμό, την ασφάλεια και την ολική διαχείριση της λειτουργίας του δικτύου. Δεν μπορεί να υπάρχουν δύο συντονιστές στο ίδιο δίκτυο Zigbee· η ύπαρξή του είναι μοναδική και θεμελιώδης για τη σταθερότητα του οικοσυστήματος. Ο συντονιστής συχνά υλοποιείται μέσω ενός USB dongle που συνδέεται στο σύστημα ελέγχου, που τρέχει το Home Assistant. Από τη στιγμή που το δίκτυο συσταθεί, μπορούν να ενταχθούν σε αυτό άλλες συσκευές, οι οποίες ανάλογα με τις δυνατότητές τους, αναλαμβάνουν είτε ρόλο δρομολογητή είτε σάν τερματική συσκευή. Οι δρομολογητές αποτελούν ενδιάμεσους κόμβους στο δίκτυο και είναι υπεύθυνοι για την αναμετάδοση πακέτων δεδομένων, εξυπηρετώντας τη σύνδεση μεταξύ συσκευών που βρίσκονται εκτός άμεσης εμβέλειας του συντονιστή ή άλλων κόμβων. Είναι κρίσιμο να σημειωθεί ότι μόνο οι Zigbee συσκευές που τροφοδοτούνται μόνιμα με ρεύμα (μέσω πρίζας ή USB) μπορούν να λειτουργήσουν ως δρομολογητές. Οι συσκευές που τροφοδοτούνται με μπαταρία δεν έχουν την τεχνική δυνατότητα να λειτουργούν διαρκώς ενεργές, καθώς η συνεχής επικοινωνία απαιτεί υψηλότερη ενεργειακή κατανάλωση. Συνεπώς, είναι υποχρεωτικά τερματικές συσκευές, δηλαδή τερματικοί κόμβοι που επικοινωνούν μόνο με τον συντονιστή ή με κάποιον κοντινό δρομολογητή, και συνήθως μεταβαίνουν σε λειτουργία ύπνου (low power state) όταν δεν μεταδίδουν δεδομένα, ώστε να παρατείνεται η διάρκεια ζωής της μπαταρίας τους. Η λειτουργία της mesh τοπολογίας στο Zigbee είναι τέτοια ώστε το δίκτυο να μπορεί να επεκτείνεται δυναμικά καθώς προστίθενται νέες συσκευές. Η κάθε συσκευή διατηρεί πληροφορίες για τις γειτονικές της και είναι σε θέση να επιλέξει την βέλτιστη διαδρομή για τη μετάδοση δεδομένων προς τον συντονιστή, ακόμη κι αν αυτό απαιτεί να διέλθουν τα πακέτα από πολλούς ενδιάμεσους κόμβους. Το δίκτυο προσαρμόζεται αυτόματα σε αλλαγές, όπως αποσύνδεση συσκευών ή προσθήκη νέων, διατηρώντας τη συνοχή του μέσω δυναμικής δρομολόγησης. Επιπλέον, επιτρέπει την απρόσκοπτη λειτουργία ακόμη και σε περιβάλλοντα με μεγάλο αριθμό συσκευών, κάτι που είναι ιδιαίτερα χρήσιμο σε εφαρμογές smart home όπου οι ανάγκες επεκτασιμότητας είναι αυξημένες. Η διαμόρφωση και η συντήρηση ενός Zigbee δικτύου πραγματοποιείται με ασφάλεια, καθώς το πρωτόκολλο ενσωματώνει μηχανισμούς κρυπτογράφησης AES 128-bit, πιστοποίησης συσκευών και ελέγχου ταυτότητας, ώστε να αποτρέπεται η παρείσφρηση μη εξουσιοδοτημένων στοιχείων. Τα χαρακτηριστικά αυτά, σε συνδυασμό με τη χαμηλή κατανάλωση ενέργειας, την ανθεκτικότητα της mesh τοπολογίας και την ευρεία διαλειτουργικότητα με χιλιάδες εμπορικά διαθέσιμες

συσκευές, καθιστούν το Zigbee ένα από τα πιο αξιόπιστα και διαδεδομένα πρωτόκολλα για ασύρματη επικοινωνία στον τομέα των συστημάτων έξυπνων κλειδαριών

Η διαδικασία σύνδεσης και επικοινωνίας μεταξύ των συσκευών σε ένα δίκτυο Zigbee, και ειδικότερα μεταξύ του συντονιστή, των routers και των end devices, βασίζεται σε μια συγκεκριμένη ακολουθία αλληλεπιδράσεων, η οποία αποσκοπεί στην ασφαλή και σταθερή ένταξη της εκάστοτε συσκευής στο mesh δίκτυο και στη συνεχιζόμενη επικοινωνία της εντός του. Αρχικά, όταν μια νέα Zigbee συσκευή ενεργοποιείται και επιχειρεί να ενταχθεί στο δίκτυο, ξεκινά με τη διαδικασία ανακάλυψης (discovery). Σε αυτό το στάδιο, η συσκευή σαρώνει τα διαθέσιμα κανάλια για να εντοπίσει την παρουσία ενός δικτύου Zigbee που είναι σε κατάσταση "permit join", δηλαδή επιτρέπει την προσθήκη νέων κόμβων. Μόλις εντοπίσει έναν τέτοιο κόμβο – συνήθως τον συντονιστή ή έναν δρομολογητή – αποστέλλει αίτημα σύνδεσης (association request). Ο κόμβος απαντά με ένα μήνυμα αποδοχής (association response), και κατά την ολοκλήρωση αυτής της διαδικασίας πραγματοποιείται το λεγόμενο "handshake". Κατά τη διάρκεια του handshake γίνεται ανταλλαγή βασικών στοιχείων όπως η διεύθυνση MAC της συσκευής, η εκχώρηση μιας μοναδικής δικτυακής διεύθυνσης εντός του Zigbee δικτύου, καθώς και η επικύρωση της συμμετοχής βάσει του pre-shared network key, το οποίο διασφαλίζει την κρυπτογραφημένη επικοινωνία εντός του δικτύου.



Σχήμα 2.6:Μορφή Δεδομενογράμματος ενός πακέτου Zigbee [6.α]

Η ασφάλεια είναι εγγενές χαρακτηριστικό του Zigbee και ενσωματώνεται από τα πρώτα βήματα του handshake. Πριν επιτραπεί η μόνιμη ένταξη μιας συσκευής στο δίκτυο, πρέπει να έχει προσπελάσει ή να της δοθεί το shared network key, με το οποίο γίνεται κρυπτογράφηση 128-bit σε όλα τα μηνύματα. Υπάρχουν δύο βασικά σενάρια σε αυτό το στάδιο: είτε το κλειδί παρέχεται μέσω pre-configuration, είτε προωθείται από τον συντονιστή κατά την ένταξη της συσκευής, εφόσον η ένταξη επιτρέπεται και η συσκευή θεωρείται αξιόπιστη. Αφού η συσκευή ενταχθεί, διατηρεί μια ενεργή λογική σύνδεση με τον γονικό κόμβο της – είτε αυτός είναι δρομολογητής είτε ο ίδιος ο Συντονιστής. Ωστόσο, το πρωτόκολλο Zigbee είναι σχεδιασμένο για εξαιρετικά χαμηλή κατανάλωση ενέργειας, ιδιαίτερα όσον αφορά τις μπαταριοκίνητες τερματικές συσκευές. Ως εκ τούτου, αυτές οι συσκευές δεν διατηρούν διαρκώς ανοιχτό δίαυλο επικοινωνίας, αλλά εισέρχονται σε κατάσταση «ύπνου» (sleep mode) για μεγάλα χρονικά διαστήματα και «ξυπνούν» περιοδικά για να επικοινωνήσουν. Για να διασφαλιστεί ότι η συσκευή παραμένει «ζωντανή» στο δίκτυο και ότι η σύνδεση της δεν θεωρείται απώλεια, χρησιμοποιείται ένας μηχανισμός που ονομάζεται keep-alive. Ο μηχανισμός αυτός λειτουργεί ως εξής: κάθε τερματική συσκευή οφείλει να στέλνει περιοδικά ένα πακέτο δεδομένων, ακόμη και εάν δεν έχει ενεργό φορτίο (payload), με στόχο να ενημερώσει τον γονικό κόμβο του ότι παραμένει ενεργό και

συνδεδεμένο. Αν αυτό δεν συμβεί μέσα σε ένα προκαθορισμένο χρονικό όριο, τότε ο γονικός κόμβος (είτε δρομολογητής είτε ο συντονιστής) μπορεί να θεωρήσει ότι η συσκευή έχει αποσυνδεθεί και να αποδεσμεύσει τους πόρους που είχαν δεσμευτεί για αυτήν. Παράλληλα, οι δρομολογητές οφείλουν να διατηρούν σταθερά την ενεργή τους κατάσταση, καθώς είναι υπεύθυνες για τη δρομολόγηση δεδομένων και τη μετάδοση μηνυμάτων μεταξύ άλλων κόμβων του δικτύου. Η επικοινωνία σε ένα Zigbee δίκτυο είναι βασισμένη στο πρωτόκολλο IEEE 802.15.4, το οποίο χρησιμοποιεί ένα μοντέλο παρόμοιο με αυτό του OSI. Τα δεδομένα μεταδίδονται σε μικρά πακέτα (frames) με ενσωματωμένο μηχανισμό επιβεβαίωσης (acknowledgment), ώστε να διασφαλίζεται η παράδοση κάθε μηνύματος. Το ίδιο το δίκτυο διαχειρίζεται την επιλογή της βέλτιστης διαδρομής (routing) για τη μετάδοση δεδομένων, χρησιμοποιώντας αλγορίθμους που βασίζονται στο tree routing ή το source routing, ανάλογα με την εφαρμογή και τις δυνατότητες του firmware των συσκευών. Η συνολική λειτουργία του δικτύου χαρακτηρίζεται από ευελιξία και ανθεκτικότητα, καθώς το mesh σύστημα επιτρέπει την αυτόματη επαναδρομολόγηση των πακέτων σε περίπτωση αλλαγών στη δομή του δικτύου ή αποσύνδεσης κόμβων. Αυτό επιτυγχάνεται μέσω της συνεχούς ανταλλαγής πληροφοριών μεταξύ των routers και του συντονιστή, με τους routing πίνακες να ενημερώνονται δυναμικά. Το αποτέλεσμα είναι ένα δίκτυο που μπορεί να αυτο-οργανώνεται, να αυτο-επουλώνεται και να προσαρμόζεται εύκολα σε μεταβαλλόμενες συνθήκες, με διατήρηση της σταθερότητας και της χαμηλής κατανάλωσης ενέργειας, καθιστώντας το Zigbee ιδανικό για εφαρμογές σε συστήματα έξυπνων κλειδαριών.

2.6: Ραδιοσυχνότητα

Η βασική αρχή λειτουργίας ενός Radio Frequency συστήματος έγκειται στην εκπομπή ενός διαμορφωμένου σήματος από έναν πομπό, το οποίο λαμβάνεται από τον δέκτη και αποδιαμορφώνεται ώστε να αναγνωριστεί η χρήσιμη πληροφορία. Η διαμόρφωση (modulation) αυτή μπορεί να είναι είτε απλή, όπως η ASK (Amplitude Shift Keying) και FSK (Frequency Shift Keying), είτε πιο εξελιγμένη, όπως η PSK (Phase Shift Keying) ή ακόμα και GFSK (Gaussian Frequency Shift Keying) σε περιπτώσεις που απαιτείται μεγαλύτερη αξιοπιστία ή απόδοση. Η αποστολή των εντολών σε RF συνήθως γίνεται σε μορφή κωδικοποιημένων bits, τα οποία περιλαμβάνουν ένα preamble (σειρά συγχρονισμού), ένα header, την κύρια πληροφορία (π.χ. on/off, ανοίγει/κλείνει) και ενδεχομένως κάποιο parity bit ή CRC για τον έλεγχο σφαλμάτων.

Ένα από τα σημαντικότερα χαρακτηριστικά των RF συστημάτων είναι ότι δεν απαιτούν σύνδεση σε δίκτυο, gateway ή οποιοδήποτε ενδιάμεσο επίπεδο λογισμικού για να λειτουργήσουν. Αυτή η ιδιότητα τα καθιστά εξαιρετικά απλά στην υλοποίηση, με σχεδόν άμεση και χωρίς καθυστερήσεις (latency-free) απόκριση, γεγονός που τα καθιστά κατάλληλα για κρίσιμες εφαρμογές όπως συναγερμοί, κουδούνια και απομακρυσμένοι διακόπτες. Ωστόσο, το γεγονός ότι πολλά από τα RF συστήματα λειτουργούν χωρίς διμερή επιβεβαίωση λήψης (acknowledgment) ή χωρίς κρυπτογράφηση καθιστά την ασφάλειά τους περιορισμένη. Είναι δυνατόν να

υποκλαπεί ένα σήμα, να αναπαραχθεί (replay attack) και να εκτελεστεί ξανά χωρίς την έγκριση του χρήστη. Ορισμένα εξελιγμένα RF πρωτόκολλα χρησιμοποιούν rolling code, όπως αυτά των τηλεχειριστηρίων για γκαραζόπορτες, ώστε να αποφευχθεί αυτή η μορφή επίθεσης, αλλά η πλειοψηφία των φθηνών υλοποιήσεων παραμένει εκτεθειμένη. Όσον αφορά την τοπολογία της επικοινωνίας, τα RF συστήματα συνήθως ακολουθούν ένα μοντέλο point-to-point (ένας πομπός προς έναν δέκτη) ή 28 point-to-multipoint (ένας πομπός προς πολλούς δέκτες), χωρίς όμως ενδιάμεση δρομολόγηση δεδομένων, χωρίς mesh και χωρίς feedback, σε αντίθεση με τεχνολογίες όπως το Zigbee ή το Wi-Fi. Αυτό περιορίζει την ευελιξία τους αλλά παράλληλα μειώνει το κόστος και την πολυπλοκότητα.

Ένα σημαντικό πλεονέκτημα της RF τεχνολογίας είναι το πολύ χαμηλό ενεργειακό της αποτύπωμα. Οι RF πομποί και δέκτες καταναλώνουν ελάχιστη ενέργεια, γι' αυτό και βρίσκονται σε συσκευές που λειτουργούν με απλές μπαταρίες για χρόνια χωρίς επαναφόρτιση. Επίσης, η εμβέλειά τους είναι συνήθως αρκετά καλή, φτάνοντας τα 50-100 μέτρα σε ανοιχτό χώρο, και τα σήματα μπορούν να διαπεράσουν τοίχους και εμπόδια, αν και υπόκεινται σε εξασθένηση ανάλογα με το υλικό. Πέρα από τη βασική one-way αρχιτεκτονική, σε ορισμένες περιπτώσεις υπάρχουν πιο εξελιγμένα RF modules που επιτρέπουν bidirectional επικοινωνία, δίνοντας τη δυνατότητα σε μία συσκευή να απαντήσει πίσω στον πομπό, αν και αυτό παραμένει εξαίρεση και όχι ο κανόνας στις απλές consumer εφαρμογές.

Η συχνότητα στην οποία λειτουργεί ένα RF σύστημα επηρεάζει άμεσα την απόσταση που μπορεί να διανύσει το σήμα και την ικανότητά του να διαπερνά εμπόδια. Για παράδειγμα, χαμηλότερες συχνότητες όπως τα 315 MHz ή τα 433 MHz έχουν καλύτερη διείσδυση μέσα από τοίχους και δομικά υλικά αλλά προσφέρουν χαμηλότερο bandwidth και κατά συνέπεια μικρότερη ταχύτητα μετάδοσης δεδομένων. Από την άλλη πλευρά, υψηλότερες συχνότητες όπως τα 868 MHz και 915 MHz είναι πιο ευάλωτες στην απορρόφηση από το περιβάλλον αλλά επιτρέπουν πιο γρήγορη επικοινωνία και λιγότερη παρεμβολή από οικιακές συσκευές. Το εύρος ζώνης της RF επικοινωνίας, στις συνηθισμένες απλές υλοποιήσεις, είναι αρκετά περιορισμένο – συνήθως της τάξης των λίγων kilobits ανά δευτερόλεπτο – αλλά αυτό είναι απόλυτα επαρκές για εφαρμογές που απαιτούν μόνο την αποστολή μίας εντολής ή ενός ψηφιακού παλμού (π.χ. “άνοιγμα πόρτας”, “ενεργοποίηση φωτός”). Μια άλλη ενδιαφέρουσα πτυχή των RF συστημάτων είναι η χρήση τους σε «τυφλές» εγκαταστάσεις, όπου δεν υπάρχει cloud, router ή άλλο έξυπνο υποσύστημα. Ένα κλασικό παράδειγμα είναι η αντιστοίχιση ενός τηλεχειριστηρίου με έναν RF διακόπτη φωτισμού: το pairing γίνεται απλά με το πάτημα ενός πλήκτρου και η επικοινωνία παραμένει λειτουργική ακόμη και αν διακοπεί το τοπικό δίκτυο ή η πρόσβαση στο Internet. Αυτή η αυτονομία τα καθιστά εξαιρετικά αξιόπιστα σε περιβάλλοντα όπου η σταθερότητα της σύνδεσης δεν είναι εγγυημένη.

Ωστόσο, επειδή τα περισσότερα RF πρωτόκολλα δεν βασίζονται σε μοναδικές ταυτότητες (unique identifiers), προκύπτει ένα φλέγον ζήτημα σχεδόν σε όλες τις

εφαρμογές: η πιθανότητα παρεμβολών από άλλες RF συσκευές ή ακόμα και η ακούσια ενεργοποίηση συσκευών από άλλους πομπούς. Το πρόβλημα αυτό έχει αντιμετωπιστεί εν μέρει με τη χρήση κωδικοποίησης rolling-code και encryption, όπως στο πρωτόκολλο Keeloq που εφαρμόζεται σε τηλεχειριστήρια γκαραζόπορτας, αλλά στις περισσότερες φθηνές υλοποιήσεις το σήμα εξακολουθεί να είναι plaintext και εύκολο στην αντιγραφή. Σε εφαρμογές έξυπνων σπιτιών, τα RF modules χρησιμοποιούνται κυρίως για να γεφυρώσουν το χάσμα μεταξύ «παραδοσιακών» και έξυπνων συσκευών. Πολλοί RF πομποί συνδέονται με κόμβους που υποστηρίζουν Home Assistant, ESPHome ή άλλες πλατφόρμες, οι οποίοι μέσω λογισμικού αντιστοιχίζουν το κάθε RF σήμα σε μια συγκεκριμένη ενέργεια.

Τέλος, αξίζει να σημειωθεί ότι η τεχνολογία RF δεν εξαρτάται από ένα συγκεκριμένο πρότυπο, όπως το Zigbee ή το Bluetooth, γεγονός που οδηγεί σε μια τεράστια ποικιλομορφία μεταξύ διαφορετικών συσκευών. Αυτή η έλλειψη τυποποίησης σημαίνει ότι κάθε κατασκευαστής μπορεί να χρησιμοποιήσει το δικό του πρωτόκολλο και format, κάτι που συχνά περιορίζει τη διαλειτουργικότητα. Παρ' όλα αυτά, η τεχνολογία RF εξακολουθεί να έχει παρουσία λόγω του εξαιρετικά χαμηλού κόστους της, της ενεργειακής αποδοτικότητας και της απλότητας υλοποίησης 29 της, ιδίως σε περιπτώσεις που απαιτείται γρήγορη και απροβλημάτιστη επικοινωνία χωρίς τη μεσολάβηση περίπλοκων υποδομών.

2.7: Bluetooth

Η τεχνολογία Bluetooth αποτελεί μία από τις πιο διαδεδομένες μορφές ασύρματης επικοινωνίας μικρής εμβέλειας, με βασική αποστολή τη μετάδοση δεδομένων μεταξύ συσκευών χωρίς τη χρήση καλωδίων. Αναπτύχθηκε αρχικά από την εταιρεία Ericsson το 1994 ως εσωτερικό ερευνητικό έργο για την αντικατάσταση των καλωδίων μεταξύ κινητών τηλεφώνων και περιφερειακών. Σταδιακά, εξελίχθηκε σε διεθνές πρότυπο υπό την επίβλεψη του οργανισμού Bluetooth Special Interest Group (Bluetooth SIG), που συστάθηκε το 1998 και στον οποίο συμμετέχουν πλέον πάνω από 30.000 εταιρείες από τον χώρο των τηλεπικοινωνιών, των υπολογιστών και των καταναλωτικών ηλεκτρονικών. Η τεχνολογία βασίζεται σε επικοινωνία στο φάσμα ISM (Industrial, Scientific and Medical) των 2.4 GHz, χρησιμοποιώντας frequency hopping spread spectrum (FHSS), δηλαδή μια τεχνική όπου το σήμα αλλάζει συνεχώς συχνότητες (μέχρι και 1600 φορές το δευτερόλεπτο) για να μειωθούν οι παρεμβολές και να αυξηθεί η ασφάλεια και αξιοπιστία της σύνδεσης. Ο τρόπος λειτουργίας είναι βασισμένος σε μια αρχιτεκτονική master-slave, όπου μία κύρια συσκευή (π.χ. smartphone ή υπολογιστής) διαχειρίζεται έως και επτά δευτερεύουσες (slaves), δημιουργώντας ένα μικρό ασύρματο δίκτυο που ονομάζεται piconet. Πολλαπλά piconets μπορούν να συνδεθούν μεταξύ τους σχηματίζοντας scatternets, αν και αυτό δεν υποστηρίζεται από όλες τις συσκευές.

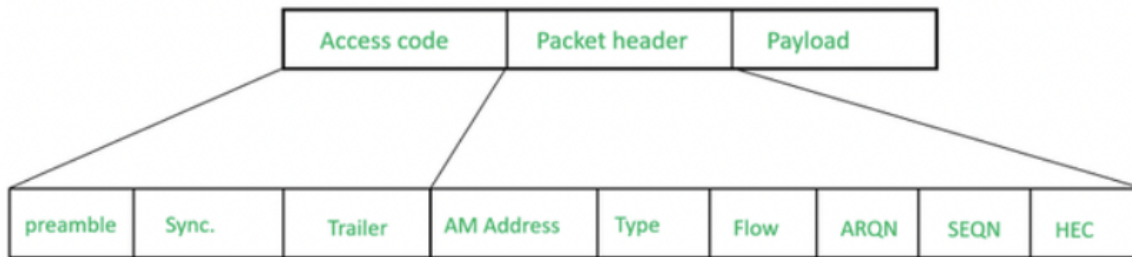
Η μετάδοση δεδομένων μέσω Bluetooth καλύπτει εύρος λειτουργίας από περίπου 1 έως 100 μέτρα, ανάλογα με την κλάση ισχύος της συσκευής. Υπάρχουν τρεις βασικές κλάσεις: Class 1 (με μέγιστη ισχύ 100 mW και εμβέλεια έως 100 μέτρα),

Class 2 (2.5 mW, έως 10 μέτρα, η πιο κοινή σε smartphones) και Class 3 (1 mW, έως 1 μέτρο). Η ταχύτητα μετάδοσης εξαρτάται από την έκδοση του Bluetooth. Οι πρώτες εκδόσεις (1.0 έως 2.0) παρείχαν ταχύτητες έως 1 Mbps, ενώ το Bluetooth 2.1 + EDR αύξησε τον ρυθμό σε περίπου 3 Mbps. Με την έλευση του Bluetooth 4.0 και της τεχνολογίας Bluetooth Low Energy (BLE), δόθηκε έμφαση στη μείωση της κατανάλωσης ενέργειας, επιτρέποντας τη λειτουργία συσκευών με μικρές μπαταρίες για μήνες ή ακόμα και χρόνια, σε βάρος της ταχύτητας (που περιορίζεται γύρω στα 1 Mbps). Η έκδοση 5.0 και μεταγενέστερες (5.1, 5.2, 5.3) βελτίωσαν σημαντικά την εμβέλεια (έως 240 μέτρα θεωρητικά), την ταχύτητα (έως 2 Mbps σε BLE) και την ακρίβεια στον προσδιορισμό θέσης

Bluetooth Version	Adoption Year	Maximum rate		Theoretical Range	Features
		Classic	Low Energy		
Bluetooth 1.0	1999	723Kbps	N/A	10 m	Basic Rate Mode (BR), used to replace RS-232 serial connection.
Bluetooth 2.0+EDR	2004	2.1 Mbps	N/A	10 m	Enhanced Data Rate (EDR), increases transmission speed, supports full-duplex communication.
Bluetooth 3.0+HS	2009	24 Mbps (only high speed mode)	N/A	10 m	High Speed (HS) mode, uses Wi-Fi technology for higher transmission speeds.
Bluetooth 4.0-4.2	2010/2013/2014	3 Mbps	1 Mbps	60 m	Low Energy (LE), optimized for IoT devices.
Bluetooth 5.0	2016	50 Mbps	2 Mbps	240 m	Increased transmission speed and range, supports indoor positioning.
Bluetooth 5.1	2019	50 Mbps	2 Mbps	240 m	Angle of Arrival (AoA) and Angle of Departure (AoD) for improved positioning accuracy.
Bluetooth 5.2	2020	50 Mbps	2 Mbps	240 m	Enhanced audio features, supports LE Audio and other communications.
Bluetooth 5.3	2021	50 Mbps	2 Mbps	240 m	Further optimized transmission efficiency, stable connection security.
Bluetooth 5.4	2023	50 Mbps	2 Mbps	240 m	Further optimized transmission features, improves device interaction.
Bluetooth 6.0	2024	Unknown	3 Mb/s	300 m	Bluetooth Channel Sounding for secure, fine-ranging

Σχήμα 2.7: Γενιές του Bluetooth πρωτόκολλου [7.α]

Ο τρόπος με τον οποίο δημιουργείται μια σύνδεση Bluetooth περιλαμβάνει τη διαδικασία του pairing, δηλαδή του αρχικού «χειραψίας» (handshake) μεταξύ δύο συσκευών, με στόχο την ανταλλαγή κρυπτογραφημένων κλειδιών και την επαλήθευση ταυτότητας. Το pairing μπορεί να γίνει με διάφορους τρόπους (PIN, passkey, NFC ή Just Works), ανάλογα με την έκδοση του Bluetooth και τις δυνατότητες των συσκευών. Αφού ολοκληρωθεί το pairing, οι συσκευές παραμένουν σε bonded κατάσταση, ώστε να μην χρειάζεται να επαναλάβουν τη διαδικασία σε επόμενες συνδέσεις. Κατά τη διάρκεια λειτουργίας της σύνδεσης, οι συσκευές στέλνουν περιοδικά keep-alive μηνύματα (γνωστά και ως supervision timeouts) για να επιβεβαιώνουν τη σταθερότητα του συνδέσμου και να αποτρέπουν τη διακοπή λόγω απουσίας δραστηριότητας. Η ασφάλεια στη μετάδοση δεδομένων επιτυγχάνεται μέσω ενσωματωμένης κρυπτογράφησης 128-bit και authentication mechanisms, ιδιαίτερα στις εκδόσεις 4.2 και άνω, που συμμορφώνονται και με πρότυπα ασφάλειας IoT. Ωστόσο, παλαιότερες εκδόσεις του Bluetooth (1.x και 2.x) παρουσίαζαν σημαντικά κενά ασφαλείας, τα οποία στο παρελθόν εκμεταλλεύτηκαν κακόβουλοι χρήστες σε επιθέσεις τύπου bluejacking, bluesnarfing και man-in-the-middle.



Σχήμα 2.8: Δεδομενόγραμμα Bluetooth [8.α]

Το Bluetooth συναντάται κυρίως σε περιφερειακές συσκευές μικρής ισχύος και χαμηλής κατανάλωσης, όπως έξυπνα λουκέτα, αισθητήρες παρουσίας, συσκευές fitness ή ιατρικά wearables. Ειδικά το Bluetooth Low Energy είναι κατάλληλο για εφαρμογές όπου απαιτείται περιστασιακή μετάδοση μικρών ποσοτήτων δεδομένων και μεγάλη αυτονομία. Αν και το Bluetooth δεν υποστηρίζει mesh δικτύωση σε παλαιότερες εκδόσεις, από το 2017 και έπειτα, με την καθιέρωση του Bluetooth Mesh Profile, καθίσταται δυνατή η δημιουργία επεκτάσιμων mesh δικτύων όπου κάθε συσκευή μπορεί να αναμεταδίδει πακέτα άλλων συσκευών.

Αυτή η δυνατότητα, όμως, εφαρμόζεται κυρίως σε σταθερές και συνεχώς τροφοδοτούμενες συσκευές, καθώς απαιτεί μεγαλύτερη επεξεργαστική ισχύ και ενεργειακή απόδοση. Οι συσκευές με μπαταρία παραμένουν κατά κανόνα σε ρόλο απλού κόμβου (end device). Η διαδικασία σύνδεσης μεταξύ δύο συσκευών Bluetooth βασίζεται στη θεμελιώδη έννοια του handshake, το οποίο αποτελεί το αρχικό στάδιο επικοινωνίας και εμπιστοσύνης μεταξύ των συσκευών. Στο στάδιο αυτό, γίνεται η αναγνώριση των συσκευών, η διαπραγμάτευση παραμέτρων ασφαλείας και η δημιουργία του λεγόμενου trust relationship, με απώτερο στόχο τη διασφάλιση της ασφαλούς ανταλλαγής δεδομένων. Το handshake πραγματοποιείται στο πλαίσιο της διαδικασίας pairing, η οποία εξελίσσεται σε τρία βασικά βήματα: την ανακάλυψη (discovery), την αυθεντικοποίηση (authentication) και την εγκαθίδρυση ασφαλούς καναλιού επικοινωνίας. Αρχικά, μια συσκευή μεταδίδει ένα inquiry πακέτο στο ραδιοφάσμα των 2.4 GHz, αναζητώντας κοντινές διαθέσιμες συσκευές. Μόλις βρεθεί συμβατή συσκευή, ανταλλάσσονται πληροφορίες ταυτοποίησης όπως η διεύθυνση MAC, το όνομα της συσκευής και οι υποστηριζόμενες υπηρεσίες.

Κατά τη δεύτερη φάση, ενεργοποιούνται οι μηχανισμοί ασφαλείας. Ανάλογα με την έκδοση του πρωτοκόλλου, η διαδικασία μπορεί να περιλαμβάνει χρήση PIN code, passkey, Numeric Comparison, Out of Band pairing ή τη λεγόμενη Just Works μέθοδο. Το είδος του pairing που θα επιλεγεί εξαρτάται από τις δυνατότητες των συσκευών και την υποστήριξη από τη συγκεκριμένη έκδοση Bluetooth. Για παράδειγμα, από την έκδοση 2.1 και μετά, χρησιμοποιείται το πρωτόκολλο Secure Simple Pairing (SSP), το οποίο προσφέρει βελτιωμένη ασφάλεια με χρήση Elliptic Curve Diffie-Hellman (ECDH) για ανταλλαγή κλειδιών, εξαλείφοντας την ανάγκη για αποθήκευση ή μετάδοση ευαίσθητων δεδομένων σε μη κρυπτογραφημένη μορφή. Με την επιτυχή ολοκλήρωση του pairing, δημιουργείται μια ασφαλής σύνδεση, γνωστή ως bond, μέσω της οποίας οι συσκευές μπορούν να συνδεόνται στο μέλλον χωρίς επανάληψη της διαδικασίας. Από εκεί και πέρα, κάθε πακέτο που μεταδίδεται

κρυπτογραφείται με δυναμικά παραγόμενα κλειδιά 128-bit, ενώ χρησιμοποιούνται και message integrity checks για την προστασία από τροποποίηση ή παρεμβολή. Η ανταλλαγή πληροφοριών σε επίπεδο δεδομένων γίνεται μέσω προκαθορισμένων προφίλ Bluetooth, που περιγράφουν τη μορφή και τα χαρακτηριστικά των υπηρεσιών που υποστηρίζει η κάθε συσκευή. Ενδεικτικά, υπάρχουν τα προφίλ HID (για πληκτρολόγια και ποντίκια), A2DP (για ήχο), GATT (για BLE αισθητήρες), SPP (για σειριακή επικοινωνία) κ.ά. Κάθε προφίλ βασίζεται σε συγκεκριμένα services και characteristics, που λειτουργούν ως κανάλια ανταλλαγής δεδομένων.

Στο επίπεδο διατήρησης της σύνδεσης, η τεχνολογία Bluetooth ενσωματώνει μηχανισμούς όπως τα keep-alive packets και supervision timeouts. Αυτά εξασφαλίζουν ότι η σύνδεση παραμένει ενεργή, ενώ παράλληλα επιτρέπουν στις συσκευές να ανιχνεύσουν τότε μια σύνδεση έχει χαθεί ή τότε απαιτείται επανασύνδεση. Ιδιαίτερα στο Bluetooth Low Energy, τα keep-alive πακέτα είναι αραιότερα για εξοικονόμηση ενέργειας, ενώ υποστηρίζονται connection intervals και sleep modes που επιτρέπουν στις συσκευές να μπαίνουν σε κατάσταση χαμηλής κατανάλωσης όταν δεν υπάρχει ανάγκη για συνεχή επικοινωνία. Η ασφάλεια της τεχνολογίας Bluetooth έχει εξελιχθεί σημαντικά με την πάροδο των ετών. Παρότι οι πρώτες εκδόσεις (π.χ. 1.0 ή 2.0) είχαν σοβαρές αδυναμίες που επέτρεπαν επιθέσεις τύπου bluejacking, bluesnarfing και man-in-the-middle, οι πιο σύγχρονες εκδόσεις (4.2, 5.0 και 5.1) ενσωματώνουν ισχυρούς μηχανισμούς όπως AES-CCM encryption, key refreshing, καθώς και δυνατότητα προστασίας της ιδιωτικότητας μέσω random address generation, ώστε οι συσκευές να μη γίνονται εύκολα ιχνηλάσιμες. Στο πλαίσιο μιας έξυπνης κλειδαρίας, η τεχνολογία Bluetooth παραμένει πολύτιμη όταν απαιτείται η σύνδεση χαμηλής ισχύος, μικρής εμβέλειας και με ελάχιστο κόστος. Με την ωρίμανση των Bluetooth Mesh δικτύων, η τεχνολογία αποκτά και τη δυνατότητα επεκτασιμότητας, αν και αυτό εξακολουθεί να είναι περιορισμένο σε πολύ συγκεκριμένες εφαρμογές όπου οι συσκευές είναι συνεχώς τροφοδοτούμενες και διαθέτουν τις απαιτούμενες δυνατότητες επεξεργασίας και μνήμης.[21]

2.8: VPN (Virtual Private Network)

Ένα VPN (Virtual Private Network – Εικονικό Ιδιωτικό Δίκτυο) είναι μια τεχνολογία που επιτρέπει την ασφαλή και κρυπτογραφημένη σύνδεση ενός υπολογιστή ή δικτύου με κάποιο απομακρυσμένο δίκτυο μέσω του διαδικτύου. Σκοπός ενός VPN είναι η δημιουργία ενός ασφαλούς “τούνελ” (tunnel) δεδομένων, μέσω του οποίου η επικοινωνία ανάμεσα σε δύο ή περισσότερους κόμβους αποκτά χαρακτηριστικά εμπιστευτικότητας, ακεραιότητας και ταυτοποίησης, σαν να επρόκειτο για άμεση σύνδεση σε τοπικό δίκτυο. Η τεχνολογική βάση ενός VPN στηρίζεται στην αρχιτεκτονική της σήραγγας (tunneling), κατά την οποία τα πακέτα ενός πρωτοκόλλου (π.χ. TCP/IP) εγκιβωτίζονται (encapsulated) μέσα σε άλλα πακέτα ώστε να μεταφερθούν μέσω δικτύων τρίτων, όπως το δημόσιο διαδίκτυο. Αυτή η σήραγγα μπορεί να είναι είτε point-to-point (για παράδειγμα, σύνδεση ενός client με έναν server) είτε site-to-site (για διασύνδεση δύο απομακρυσμένων δικτύων μεταξύ τους). Η διαδικασία αυτή συνδυάζεται με μηχανισμούς κρυπτογράφησης των

δεδομένων, επιτρέποντας ασφαλή μεταφορά ακόμα και μέσω επισφαλών δικτύων.[21]

Η υλοποίηση ενός VPN βασίζεται σε λογισμικό ή εξειδικευμένο υλικό (VPN appliances) και περιλαμβάνει δύο βασικά μέρη: τον client (πελάτη) και τον server. Ο client εγκαθίσταται στον υπολογιστή, smartphone ή router του χρήστη, ενώ ο server βρίσκεται στο απομακρυσμένο σημείο (π.χ. το σπίτι, το πανεπιστήμιο, ο εταιρικός κόμβος). Κατά την εγκαθίδρυση της σύνδεσης, πραγματοποιείται διαδικασία ταυτοποίησης, ανταλλαγής κλειδίων και συμφωνίας επί των κρυπτογραφικών παραμέτρων (handshake), ώστε η σύνδεση να εξασφαλίσει ιδιωτικότητα και ασφάλεια. Τα VPN χρησιμοποιούνται ευρέως για διάφορους σκοπούς. Στο πλαίσιο μίας εξυπνης κλειδαριάς, επιτρέπουν την απομακρυσμένη πρόσβαση στο τοπικό δίκτυο σαν να ήταν ο χρήστης τοπικά συνδεδεμένος, χωρίς να απαιτείται επικίνδυνο port forwarding. Σε επιχειρησιακά περιβάλλοντα, διευκολύνουν την ασφαλή πρόσβαση υπαλλήλων σε εταιρικά δίκτυα. Επιπλέον, χρησιμοποιούνται για την προστασία της ιδιωτικότητας σε δημόσια δίκτυα Wi-Fi, την παράκαμψη γεωγραφικών περιορισμών περιεχομένου και τη γενικότερη ενίσχυση της διαδικτυακής ανωνυμίας.[21]

Υπάρχουν αρκετά πρωτόκολλα υλοποίησης VPN, καθένα από τα οποία προσφέρει διαφορετικά χαρακτηριστικά ασφαλείας, απόδοσης και συμβατότητας. Το OpenVPN είναι ανοιχτού κώδικα και βασίζεται στο πρωτόκολλο SSL/TLS, προσφέροντας ισχυρή κρυπτογράφηση και μεγάλη ευελιξία. Το WireGuard, επίσης ανοιχτού κώδικα, είναι νεότερο, πιο ελαφρύ και εξαιρετικά αποδοτικό, με μοντέρνα αρχιτεκτονική και μικρότερο attack surface. Το IPSec χρησιμοποιείται σε συνδυασμό με πρωτόκολλα όπως το IKEv2 και είναι ιδιαίτερα διαδεδομένο σε επαγγελματικές εγκαταστάσεις, ενώ το L2TP/IPSec παρέχει σήραγγα χωρίς κρυπτογράφηση (L2TP) η οποία συνδυάζεται με το IPSec για την ασφάλεια. Άλλα, πιο παλαιά πρωτόκολλα, όπως το PPTP, θεωρούνται πλέον ανασφαλή και καταργούνται σταδιακά. Η αρχιτεκτονική ενός VPN μπορεί να διαφέρει ανάλογα με το μοντέλο που υιοθετείται. Σε κεντροποιημένα VPN, υπάρχει ένας κεντρικός server στον οποίο συνδέονται όλοι οι clients, καθιστώντας τον κόμβο αυτό το σημείο εισόδου και εξόδου των δεδομένων. Αντιθέτως, σε peer-to-peer ή mesh VPNs, όπως υλοποιούνται μέσω τεχνολογιών όπως το Tailscale (βασισμένο στο WireGuard), οι κόμβοι μπορούν να επικοινωνούν απευθείας μεταξύ τους με αποκλειστικά end-to-end κρυπτογράφηση, χωρίς να χρειάζεται κεντρική δρομολόγηση της κίνησης.[21]

Τα πλεονεκτήματα των VPN είναι πολυάριθμα. Παρέχουν ισχυρή κρυπτογράφηση και ασφάλεια, εξασφαλίζουν ιδιωτικότητα και αποφυγή καταγραφής της δραστηριότητας από παρόχους υπηρεσιών διαδικτύου, προσφέρουν γεωγραφική ανεξαρτησία πρόσβασης, και εξαλείφουν την ανάγκη για port forwarding, μειώνοντας τον κίνδυνο εκμετάλλευσης ευπαθειών. Παράλληλα, υπάρχουν και μειονεκτήματα που πρέπει να ληφθούν υπόψη. Η χρήση VPN επιφέρει γενικά μείωση της ταχύτητας λόγω κρυπτογράφησης και δρομολόγησης, απαιτεί πιο σύνθετη ρύθμιση σε σχέση με

απλές συνδέσεις, και σε κάποιες περιπτώσεις υπόκειται σε περιορισμούς από παρόχους ή κρατικές αρχές. Η τεχνολογία των VPN στηρίζεται σε πληθώρα πρωτοκόλλων, καθένα από τα οποία προσφέρει διαφορετικές προσεγγίσεις σε ζητήματα ασφάλειας, απόδοσης και συμβατότητας με υποδομές και λειτουργικά συστήματα. Ορισμένα από αυτά έχουν εξελιχθεί μέσα από δεκαετίες χρήσης σε εταιρικά και προσωπικά δίκτυα, ενώ άλλα είναι πιο σύγχρονες υλοποιήσεις που εστιάζουν στην απλότητα και τη βελτιστοποίηση. Παρακάτω θα γίνει αναφορά σε ορισμένα από τα σημαντικότερα πρωτόκολλα.[21]

Το IPSec (Internet Protocol Security) αποτελεί ένα από τα πιο διαδεδομένα και ώριμα πρωτόκολλα για τη δημιουργία ασφαλών σήραγγων σε επίπεδο δικτύου (Layer 3). Εφαρμόζεται κυρίως σε εταιρικά περιβάλλοντα και router-to-router συνδέσεις και χρησιμοποιεί μηχανισμούς όπως ESP (Encapsulating Security Payload) και AH (Authentication Header) για να διασφαλίσει την εμπιστευτικότητα, την ακεραιότητα και την αυθεντικότητα των δεδομένων. Η λειτουργία του IPSec μπορεί να είναι είτε σε transport mode, όπου μόνο το ωφέλιμο φορτίο (payload) κρυπτογραφείται, είτε σε tunnel mode, όπου ολόκληρο το IP πακέτο ενθυλακώνεται και προστατεύεται. Στενά συνδεδεμένο με το IPSec είναι το IKE (Internet Key Exchange), και πιο συγκεκριμένα η δεύτερη του έκδοση, IKEv2. Το πρωτόκολλο αυτό αναλαμβάνει τη διαπραγμάτευση των παραμέτρων ασφάλειας μεταξύ των δύο άκρων της σύνδεσης, τη δημιουργία και ανανέωση κλειδιών, καθώς και τη διαδικασία ταυτοποίησης. Το IKEv2 θεωρείται ιδιαίτερα ασφαλές και σταθερό, με ενσωματωμένους μηχανισμούς επανασύνδεσης, κάτι που το καθιστά ιδανικό για mobile περιβάλλοντα. Συχνά χρησιμοποιείται σε συνδυασμό με IPSec για τη δημιουργία πλήρως κρυπτογραφημένων VPN συνδέσεων.[21]

Το OpenVPN αποτελεί μία από τις πιο δημοφιλείς και διαδεδομένες λύσεις υλοποίησης εικονικών ιδιωτικών δικτύων (VPN), ειδικά σε περιβάλλοντα όπου απαιτείται υψηλό επίπεδο παραμετροποίησης, ασφάλειας και διαλειτουργικότητας με διάφορα λειτουργικά συστήματα και δικτυακές αρχιτεκτονικές. Αναπτύχθηκε από τον James Yonah το 2001, και από τότε εξελίχθηκε σε ανοιχτού κώδικα (open-source) λογισμικό υπό την επίβλεψη της OpenVPN Inc., με τεράστια κοινότητα χρηστών και υποστηρικτών.[21]

Σε αντίθεση με άλλα VPN πρωτόκολλα που λειτουργούν σε επίπεδο IP (όπως το IPSec), το OpenVPN βασίζεται στην αρχιτεκτονική του πρωτοκόλλου SSL/TLS (Secure Sockets Layer / Transport Layer Security), το οποίο χρησιμοποιείται ευρέως και στον παγκόσμιο ιστό για την ασφαλή μεταφορά δεδομένων. Μέσω αυτής της αρχιτεκτονικής, το OpenVPN δημιουργεί έναν ασφαλή κρυπτογραφημένο διάυλο ανάμεσα στον client και τον server, αξιοποιώντας συμμετρική κρυπτογράφηση (AES, ChaCha20), έλεγχο ταυτότητας με δημόσια και ιδιωτικά κλειδιά, καθώς και πιστοποιητικά X.509. Η ασφάλεια που παρέχει είναι εφάμιλλη – και σε πολλές περιπτώσεις ανώτερη – από παραδοσιακά πρωτόκολλα όπως το L2TP/IPSec.[21]

Ένα από τα βασικά πλεονεκτήματα του OpenVPN είναι η ευελιξία του στη χρήση των μεταφορικών επιπέδων. Μπορεί να λειτουργεί είτε μέσω TCP είτε UDP, προσφέροντας τη δυνατότητα παράκαμψης περιοριστικών firewall και NAT (Network Address Translation), ακόμη και μέσω της θύρας 443, που χρησιμοποιείται από το HTTPS. Αυτό του επιτρέπει να διεισδύει σε δίκτυα που μπλοκάρουν άλλες VPN λύσεις, καθιστώντας το ιδανικό για δύσκολα περιβάλλοντα ή δίκτυα με περιορισμούς.[21]

Η διαδικασία δημιουργίας σύνδεσης στο OpenVPN περιλαμβάνει handshake μέσω TLS, ανταλλαγή κρυπτογραφικών κλειδιών και στη συνέχεια σύναψη ενός ασφαλούς καναλιού δεδομένων. Η υποστήριξη Perfect Forward Secrecy (PFS) διασφαλίζει ότι ακόμη και σε περίπτωση υποκλοπής παλαιών κλειδιών, οι μελλοντικές ή παρελθοντικές επικοινωνίες δεν είναι δυνατόν να αποκρυπτογραφηθούν. Επιπλέον, παρέχει δυνατότητα χρήσης HMAC για την προστασία της ακεραιότητας των πακέτων και προαιρετικά χρησιμοποιεί συμπίεση δεδομένων μέσω του αλγορίθμου LZO (αν και πλέον συστήνεται η απενεργοποίησή της για λόγους ασφάλειας).[21]

Η χρήση του OpenVPN συναντάται σε πληθώρα περιβαλλόντων: από οικιακές υλοποιήσεις σε Raspberry Pi ή μικρούς servers, μέχρι εταιρικές εγκαταστάσεις υψηλής διαθεσιμότητας με clustering και failover. Η υποστήριξή του επεκτείνεται σε Windows, Linux, macOS, Android, iOS και πλήθος embedded συσκευών. Παράλληλα, προσφέρεται και με γραφικά περιβάλλοντα ή μέσω CLI, ενώ υποστηρίζει σενάρια multi-client, bridge mode και routing mode, με λεπτομερή έλεγχο μέσω αρχείων παραμετροποίησης.[21]

Το WireGuard είναι ένα νεότερο VPN πρωτόκολλο, που σχεδιάστηκε με στόχο την απλότητα, την ταχύτητα και την ασφάλεια. Γραμμένο αρχικά για τον πυρήνα του Linux, βασίζεται σε σύγχρονους αλγορίθμους κρυπτογράφησης όπως ChaCha20, Poly1305 και Curve25519, και ακολουθεί μία μινιμαλιστική σχεδίαση με μόλις ~4.000 γραμμές κώδικα – σημαντικά λιγότερες σε σχέση με τα παραδοσιακά IPSec ή OpenVPN. Το WireGuard δεν υποστηρίζει παραδοσιακή διαπραγμάτευση κλειδιών όπως το IKE, αλλά λειτουργεί μέσω ενός απλού μηχανισμού ανταλλαγής δημοσίων κλειδιών, καθιστώντας το ιδιαίτερα γρήγορο και αξιόπιστο. Παρά το ότι θεωρείται ακόμη σχετικά νέο, έχει ενσωματωθεί στους πυρήνες των σύγχρονων λειτουργικών συστημάτων και υιοθετείται ευρέως, ειδικά σε προσωπικά και οικιακά VPN.[21]

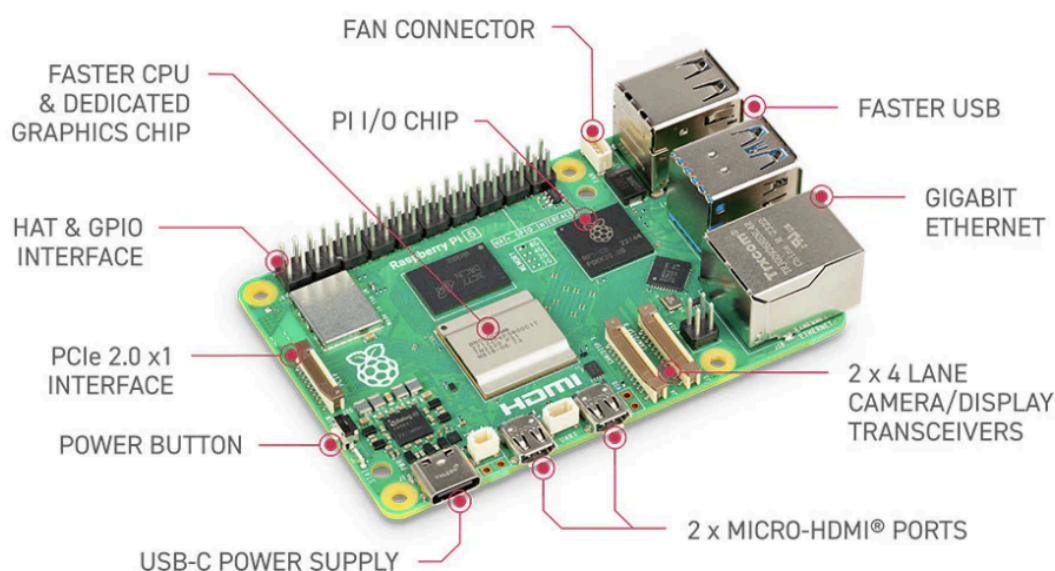
Ένα άλλο ευρέως γνωστό και υποστηριζόμενο πρωτόκολλο είναι το L2TP (Layer 2 Tunneling Protocol), το οποίο συχνά συνδυάζεται με IPSec για την κρυπτογράφηση των δεδομένων, αφού από μόνο του δεν προσφέρει κρυπτογράφηση. Το ζεύγος L2TP/IPSec θεωρείται σχετικά ασφαλές, αλλά παρουσιάζει περιορισμούς σε σύγχρονες απαιτήσεις όπως mobile switching ή χρήση σε restrictive δίκτυα, λόγω της εξάρτησής του από συγκεκριμένες θύρες (UDP 500, 1701, 4500) που συχνά μπλοκάρονται από firewalls ή NAT. Το παλαιότερο PPTP (Point-to-Point Tunneling Protocol), αν και αποτέλεσε μία από τις πρώτες υλοποιήσεις VPN, θεωρείται πλέον

ανεπαρκές από άποψη ασφάλειας και έχει καταργηθεί από πολλά λειτουργικά[21] συστήματα. Οι γνωστές του ευπάθειες, η απουσία ισχυρής κρυπτογράφησης και η εύκολη παραβίασή του καθιστούν την χρήση του απαγορευτική σε οποιοδήποτε περιβάλλον που απαιτεί βασική προστασία. Σε επίπεδο ευρείας διασύνδεσης επιχειρησιακών δικτύων, η τεχνολογία MPLS (Multiprotocol Label Switching) συχνά συγχέεται με τα VPN, παρότι δεν είναι αμιγώς VPN πρωτόκολλο. Πρόκειται για μία τεχνική δρομολόγησης που χρησιμοποιείται από παρόχους για τη δημιουργία εικονικών ιδιωτικών κυκλωμάτων μεταξύ διαφορετικών γεωγραφικών σημείων ενός οργανισμού. Το MPLS VPN προσφέρει υψηλή απόδοση, εγγυημένο QoS (Quality of Service) και αξιοπιστία, όμως η ασφάλεια του δεν είναι end-to-end και συνήθως βασίζεται στην εμπιστοσύνη στον πάροχο, κάτι που το διαφοροποιεί θεμελιωδώς από λύσεις όπως το WireGuard ή το IPSec.[21]

Εν κατακλείδι, η επιλογή του κατάλληλου πρωτοκόλλου VPN εξαρτάται από τις ανάγκες του εκάστοτε περιβάλλοντος: απόδοση, ευκολία εγκατάστασης, συμβατότητα, δυνατότητα mobility και φυσικά επίπεδο ασφάλειας. Σε προσωπικά και οικιακά περιβάλλοντα, το WireGuard ξεχωρίζει για τη μοντέρνα και αποδοτική του προσέγγιση, ενώ σε μεγάλες επιχειρήσεις το IPSec με IKEv2 παραμένει ένα σταθερό και αξιόπιστο θεμέλιο για την προστασία της δικτυακής υποδομής. Ωστόσο, η ευελιξία και η ισχύς του OpenVPN συνοδεύονται από μια σχετική πολυπλοκότητα στη ρύθμιση, ειδικά για χρήστες χωρίς εμπειρία σε δίκτυα και κρυπτογραφία. Η δημιουργία έγκυρων πιστοποιητικών, η διαχείριση των TLS parameters και η ορθή ρύθμιση των firewalls απαιτούν προσοχή και γνώση, γι' αυτό και πολλές διανομές (όπως το Home Assistant OS ή έτοιμα scripts όπως το PiVPN) προσφέρουν αυτοματοποιημένες λύσεις για την εγκατάστασή του. [21]

Κεφάλαιο 3: Raspberry Pi

Το Raspberry Pi αποτελεί έναν χαμηλού κόστους, μικρού μεγέθους υπολογιστή τύπου single-board computer (SBC), ο οποίος σχεδιάστηκε αρχικά με εκπαιδευτικό προσανατολισμό, αλλά με την πάροδο του χρόνου εξελίχθηκε σε ένα ιδιαίτερα δημοφιλές εργαλείο για εφαρμογές ενσωματωμένων συστημάτων, αυτοματισμών, Internet of Things (IoT) και πειραματικών έργων μηχανικής και πληροφορικής. Η ανάπτυξή του ξεκίνησε από το Raspberry Pi Foundation στο Ηνωμένο Βασίλειο, με βασικό στόχο τη διάδοση της πληροφορικής και του προγραμματισμού. Το Raspberry Pi συνδυάζει σε μία και μόνο πλακέτα όλα τα βασικά υποσυστήματα ενός υπολογιστή, όπως κεντρική μονάδα επεξεργασίας (CPU), μνήμη RAM, διεπαφές εισόδου/εξόδου και δυνατότητα αποθήκευσης δεδομένων, προσφέροντας παράλληλα χαμηλή κατανάλωση ενέργειας και υψηλή ευελιξία στη χρήση.[8][12][14][17]



Σχημα 3.1: Το SBC του Raspberry Pi 5 [9.α]

Το πρώτο μοντέλο, Raspberry Pi Model B (2012), σχεδιάστηκε κυρίως για εκπαιδευτική χρήση και διέθετε περιορισμένους πόρους, όπως μονοπύρρηνο επεξεργαστή και μικρή μνήμη RAM. Παρ' όλα αυτά, εισήγαγε την έννοια του πλήρους υπολογιστή σε μία μόνο πλακέτα. Στη συνέχεια, τα μοντέλα Raspberry Pi 2 και Raspberry Pi 3 παρουσίασαν σημαντικές βελτιώσεις, όπως πολυπύρηνους επεξεργαστές, αυξημένη μνήμη RAM και ενσωματωμένες δυνατότητες ασύρματης επικοινωνίας (Wi-Fi και Bluetooth). Οι αναβαθμίσεις αυτές επέτρεψαν στο Raspberry Pi να χρησιμοποιηθεί σε πιο σύνθετες εφαρμογές, όπως συστήματα αυτοματισμού και δικτυακές υπηρεσίες.

Με την κυκλοφορία του Raspberry Pi 4, η πλατφόρμα απέκτησε σαφώς αυξημένη υπολογιστική ισχύ, υποστήριξη USB 3.0, πολλαπλές εξόδους οθόνης και σημαντικά

μεγαλύτερη μνήμη RAM. Το γεγονός αυτό διέυρυνε ακόμη περισσότερο τις δυνατότητες χρήσης του, φτάνοντας σε επίπεδο βασικού επιτραπέζιου υπολογιστή. Η πιο πρόσφατη γενιά, Raspberry Pi 5, εστιάζει στη βελτίωση της απόδοσης, της διαχείρισης ενέργειας και της επεκτασιμότητας, προσφέροντας καλύτερη ψύξη, ταχύτερες διεπαφές και αυξημένη σταθερότητα. Παράλληλα, η συνεχής ανάπτυξη του οικοσυστήματος λογισμικού και της κοινότητας υποστήριξης ενισχύει τη μακροχρόνια βιωσιμότητα της πλατφόρμας.

Σε επίπεδο επεξεργαστικής ισχύος, η τελευταία γενιά διαθέτει νεότερης αρχιτεκτονικής επεξεργαστή με αυξημένες συχνότητες λειτουργίας και βελτιωμένη ενεργειακή απόδοση. Αυτό μεταφράζεται σε ταχύτερη εκτέλεση εφαρμογών και καλύτερη διαχείριση πολυδιεργασιών σε σχέση με παλαιότερα μοντέλα. Όσον αφορά τη μνήμη, παρατηρείται υποστήριξη μεγαλύτερων χωρητικοτήτων RAM και βελτιωμένη διαχείρισή της, γεγονός που επιτρέπει την ταυτόχρονη εκτέλεση περισσότερων υπηρεσιών, όπως web servers, βάσεις δεδομένων και συστήματα ελέγχου. Στον τομέα των διεπαφών εισόδου/εξόδου, η τελευταία γενιά ενσωματώνει ταχύτερες θύρες USB (π.χ. USB 3.0 ή νεότερες), βελτιωμένη απόδοση Ethernet και αναβαθμισμένες θύρες για οθόνες, προσφέροντας αυξημένα δεδομένα μεταφοράς και καλύτερη υποστήριξη περιφερειακών.

Σημαντικές είναι επίσης οι βελτιώσεις στη συνδεσιμότητα, με αναβαθμισμένα πρότυπα Wi-Fi και Bluetooth, τα οποία εξασφαλίζουν μεγαλύτερη σταθερότητα και εμβέλεια στις ασύρματες επικοινωνίες. Αυτό αποτελεί κρίσιμο πλεονέκτημα για εφαρμογές IoT και απομακρυσμένης διαχείρισης.

Συνολικά, η εξέλιξη του Raspberry Pi αντικατοπτρίζει τη μετάβασή του από ένα απλό εκπαιδευτικό εργαλείο σε μια αξιόπιστη πλατφόρμα για επαγγελματικές και ερευνητικές εφαρμογές.[8][12][14][17]

3.1 Ο Ρολος του Raspberry Pi σε ένα σύστημα Smart Lock

Το Raspberry Pi λειτουργεί ως κεντρική μονάδα (controller) ενός εξυπνου συστήματος ασφαλείας. Λόγω του μικρού μεγέθους της χαμηλής καταναλώσης και της υποστήριξης λειτουργικών συστημάτων Linux, είναι ιδανικό για εφαρμογές ασφαλείας χώρων.

Σε ένα σύστημα Smart Lock, το Raspberry Pi αναλαμβάνει την επεξεργασία των δεδομένων εισόδου-εξόδου που προέρχονται από διάφορες συσκευές ταυτοποίησης, όπως πληκτρολόγια, αναγνώστες RFID, βιομετρικούς αισθητήρες ή εφαρμογές κινητών τηλεφώνων. Οι πληροφορίες αυτές αναλύονται σε πραγματικό χρόνο και συγκρίνονται με προκαθορισμένα δικαιώματα πρόσβασης, τα οποία είναι αποθηκευμένα τοπικά ή σε απομακρυσμένη βάση δεδομένων.

Παράλληλα, το Raspberry Pi λειτουργεί ως ενδιάμεσος κρίκος μεταξύ του λογισμικού και του μηχανισμού κλειδώματος. Μέσω των GPIO pins, ελέγχει ηλεκτρομηχανικά στοιχεία όπως ηλεκτρικές κλειδαριές, ηλεκτρομαγνήτες ή σερβοκινητήρες, επιτρέποντας ή απορρίπτοντας την πρόσβαση ανάλογα με το αποτέλεσμα της

διαδικασίας ταυτοποίησης. Η δυνατότητα ακριβούς ελέγχου των ακροδεκτών εισόδου/εξόδου καθιστά τη λειτουργία του συστήματος αξιόπιστη και επαναλήψιμη.

Ένα ακόμη σημαντικό πλεονέκτημα του Raspberry Pi είναι η υποστήριξη δικτυακών λειτουργιών. Το σύστημα Smart Lock μπορεί να συνδεθεί σε τοπικό δίκτυο ή στο διαδίκτυο, επιτρέποντας την απομακρυσμένη διαχείριση, την παρακολούθηση της κατάστασης της κλειδαριάς και την καταγραφή συμβάντων πρόσβασης. Μέσω web εφαρμογών ή διεπαφών προγραμματισμού (APIs), ο διαχειριστής του συστήματος μπορεί να προσθέτει ή να αφαιρεί χρήστες, να καθορίζει επίπεδα πρόσβασης και να ελέγχει τη λειτουργία του συστήματος σε πραγματικό χρόνο.

Ιδιαίτερη σημασία έχει επίσης ο ρόλος του Raspberry Pi στην ασφάλεια του συστήματος. Το λειτουργικό σύστημα Linux παρέχει μηχανισμούς διαχείρισης χρηστών, κρυπτογράφησης και ελέγχου πρόσβασης, οι οποίοι μπορούν να αξιοποιηθούν για την προστασία των δεδομένων και την αποτροπή μη εξουσιοδοτημένων παρεμβάσεων. Επιπλέον, η δυνατότητα τακτικών ενημερώσεων λογισμικού συμβάλλει στη διατήρηση της ασφάλειας του συστήματος σε βάθος χρόνου.

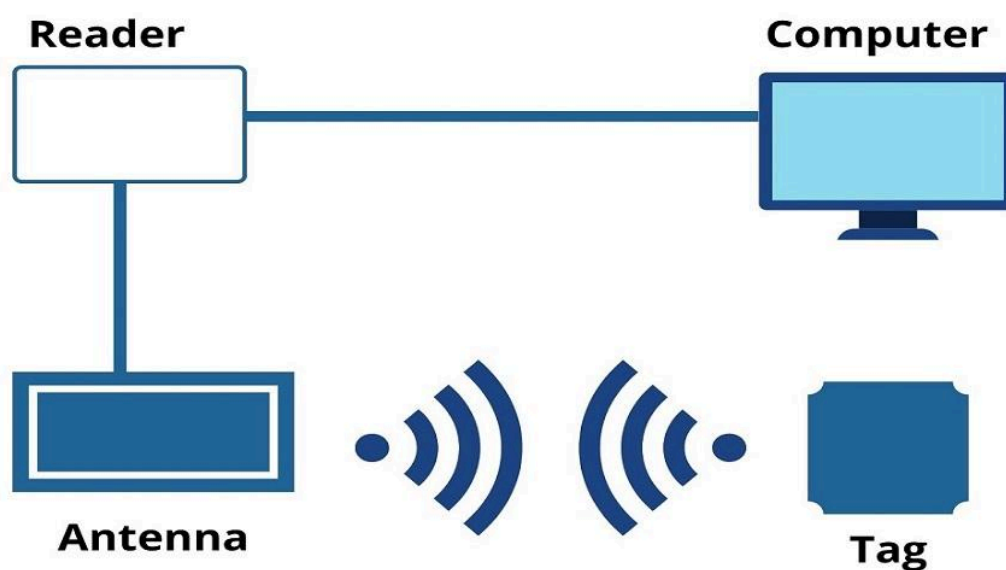
Στον τομέα της επεκτασιμότητας, το Raspberry Pi προσφέρει σημαντικά πλεονεκτήματα. Το σύστημα Smart Lock μπορεί να εμπλουτιστεί με πρόσθετες λειτουργίες, όπως αισθητήρες κατάστασης πόρτας, κάμερες επιτήρησης, οθόνες ενημέρωσης ή συστήματα ειδοποίησης. Η ευελιξία της πλατφόρμας επιτρέπει την προσαρμογή του συστήματος σε διαφορετικά περιβάλλοντα, όπως εργαστηριακούς χώρους, γραφεία ή κατοικίες.[12][13][14][17]

3.1.1 Αρχιτεκτονική συστήματος

Ένα τυπικό Smart Lock με Raspberry Pi περιλαμβάνει ένα Raspberry Pi, μια ηλεκτρική κλειδαριά έναν ηλεκτροπυρο ή ένα ηλεκτροπηνειο, μοναδα RFID ή βιομετρικούς αισθητήρες (δακτυλικό αποτυπωμα Face ID κλπ.), λογισμικό ελέγχου (Python, Flask Server) και τέλος μια βάση δεδομένων χρηστών και δικαιωμάτων.[8][17]

3.2 Συστημα και αισθητηρες RFID

Τα συστήματα RFID αποτελούν ένα υποσύνολο των Συστημάτων Αυτόματου Προσδιορισμού (Automatic Identification Systems). Ειδικότερα λειτουργεί ως γενικός όρος των τεχνολογιών που χρησιμοποιούν ραδιοκύματα για να προσδιορίσουν αυτόματα ανθρώπους ή αντικείμενα και αποτελεί την τεχνολογική εξέλιξη των ραβδωτών κωδικών(barcode)



Σχημα 3.2 Απεικονιση λειτουργιας συστηματος RFID[10.α]

Τα συστήματα RFID απαρτίζονται από δύο κύρια μέρη. Το πρώτο είναι οι πομποδέκτες (transponders) που συχνά αναφέρονται και ως ετικέτες RFID (RFID tags). Οι ετικέτες RFID είναι μικρά chips που αποτελούνται από ένα ολοκληρωμένο κύκλωμα, το οποίο περιλαμβάνει μνήμη ώστε να αποθηκεύει δεδομένα-πληροφορίες, και μία κεραία. Το μέγεθός τους μπορεί να είναι τόσο μικρό όσο το μισό ενός κόκκου άμμου (1/3 του χιλιοστού), ανάλογα με το τύπο της ετικέτας. Το δεύτερο μέρος είναι οι αναγνώστες ή αισθητήρες (readers), οι οποίοι ανακτούν τα δεδομένα από τις ετικέτες RFID. Οι αναγνώστες RFID έχουν ενσωματωμένα μια κεραία και μια μονάδα ελέγχου.



Σχημα 3.3 RFID tag και Reader[11.α,12.α]

Η λειτουργία των συστημάτων RFID είναι απλή και βασίζεται στη δυναμική και αμφίδρομη επικοινωνία των ετικετών και των αναγνώστών. Όταν οι ετικέτες RFID βρεθούν στην εμβέλεια της κεραίας του αναγνώστη, η μονάδα ελέγχου επικοινωνεί με ραδιοκύματα με την κεραία των ετικετών RFID. Οι ετικέτες RFID ενεργοποιούνται με τη σειρά τους και επιστρέφουν τα αναζητούμενα δεδομένα στους αναγνώστες. Στη συνέχεια παρεμβαίνει ένα ενδιάμεσο λογισμικό, το οποίο κατανοεί τις πληροφορίες, οι οποίες αποστέλλονται από τη μονάδα ελέγχου του αναγνώστη. Ο αναγνώστης τις μεταφέρει στο εκάστοτε πληροφοριακό σύστημα.[18][17]

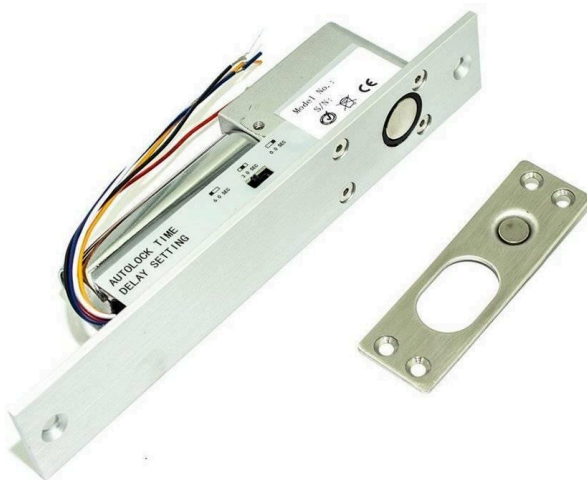
3.3 Ηλεκτροπύρος

Ο ηλεκτροπύρος είναι ένα σύστημα αυτόματου κλειδώματος. Μπορεί να τοποθετηθεί σε πόρτες: Θωρακισμένες, σιδερένιες, securit (γυάλινες πόρτες εισόδου καταστημάτων) και αλουμινίου. Βασικό τους χαρακτηριστικό είναι η λειτουργία με ψηφιακή τεχνολογία (ψηφιακά κλειδιά διαφόρων τεχνολογιών) που αλληλεπιδρούν με τα ηλεκτρονικά συστήματα ελέγχου (access control) και όχι με μηχανικά κλειδιά.

Οι ηλεκτροπύροι λειτουργούν με 2 τρόπους:

1) **Λειτουργία Fail Safe:** Όταν το ηλεκτρικό ρεύμα δεν περνάει από τη συσκευή ή σε περίπτωση διακοπής ρεύματος, ο ηλεκτροπύρος ξεκλειδώνει και η πόρτα μένει ανοιχτή. Όταν το ηλεκτρικό ρεύμα περνάει από τη συσκευή, ο ηλεκτροπύρος είναι κλειδωμένος και η πόρτα κλειστή.

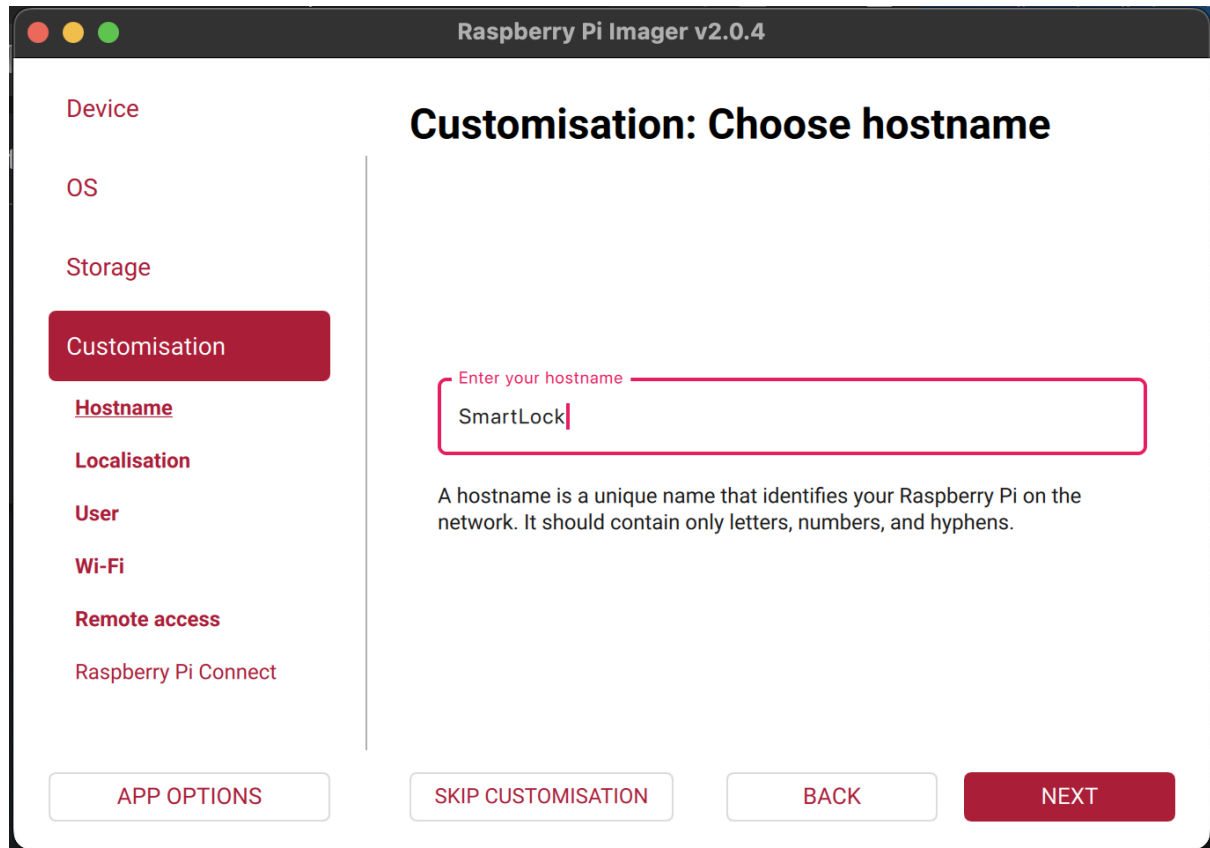
2) **Λειτουργία Fail Secure:** Όταν το ηλεκτρικό ρεύμα δεν περνάει από τη συσκευή ή σε περίπτωση διακοπής ρεύματος, ο ηλεκτροπύρος είναι κλειδωμένος και η πόρτα κλειστή. Όταν το ηλεκτρικό ρεύμα περνάει από τη συσκευή, ο ηλεκτροπύρος ξεκλειδώνει και η πόρτα μένει ανοιχτή.[19][20]



Σχημα 3.4 Ηλεκτροπύρος [13.α]

Κεφαλαίο 4: Υλοποίηση Συστήματος Smart Lock για Ελεγχόμενη Πρόσβαση σε Εργαστηριακούς Χώρους

Το πρώτο βήμα για την υλοποίηση του συστήματος είναι να εγκαταστήσουμε το Raspberry OS στο Raspberry Pi



Σχήμα 4.1 Ρύθμιση των στοιχείων του Raspberry

Για την λειτουργία του συστήματος είναι απαραίτητη η υλοποίηση ενός κυκλώματος μοσφет σε λειτουργία διακοπτή έτσι ώστε να μετατρέπει την εξοδο 3.3V που βγαζουν ως εξοδο τα gpio pins του Raspberry Pi.

4.1 Λειτουργία MOSFET ως Διακόπτης

Ένα MOSFET λειτουργεί ως διακόπτης σε πολλές συσκευές. Αλλάζετε την τάση στην πύλη για να την ενεργοποιήσετε ή να την απενεργοποιήσετε. Αυτό σας επιτρέπει να ελέγχετε το ηλεκτρικό ρεύμα γρήγορα και με ακρίβεια.[22]

Η λειτουργία του MOSFET βασίζεται στον έλεγχο της αγωγιμότητας μεταξύ των ακροδεκτών Drain και Source μέσω της τάσης που εφαρμόζεται στον ακροδέκτη Gate. Σε αντίθεση με τα διπολικά τρανζίστορ (BJT), ο MOSFET είναι στοιχείο ελεγχόμενο από τάση και παρουσιάζει εξαιρετικά μικρή κατανάλωση ρεύματος στον ακροδέκτη Gate, γεγονός που τον καθιστά ιδιαίτερα κατάλληλο για ψηφιακά συστήματα.[23][24][26]

Όταν το MOSFET βρίσκεται στην περιοχή αποκοπής, λειτουργεί σαν ανοιχτός διακόπτης και σταματά το ρεύμα. Στην περιοχή κορεσμού, λειτουργεί σαν κλειστός διακόπτης και επιτρέπει τη ροή ρεύματος. Για την εναλλαγή, θέλετε το MOSFET να παραμένει λιγότερο χρόνο στην ενδιάμεση γραμμική περιοχή. Αυτό βοηθά στη μείωση της απώλειας ισχύος και διατηρεί τη συσκευή σας δροσερή. Μπορείτε να αλλάξετε το MOSFET μεταξύ 'ON' και 'OFF' αλλάζοντας την τάση της πύλης. Στην κατάσταση 'ON', το MOSFET δίνει μια διαδρομή χαμηλής αντίστασης για το ρεύμα. Η γρήγορη εναλλαγή καθιστά το MOSFET ιδανικό για τον έλεγχο του κινητήρα και τη ρύθμιση της τροφοδοσίας.[22][25]

Το MOSFET ως διακόπτης βοηθά στην εξοικονόμηση ενέργειας και κάνει τις συσκευές να λειτουργούν καλύτερα. Τα MOSFETs βρίσκονται σε συστήματα ανανεώσιμων πηγών ενέργειας, ηλεκτρικά αυτοκίνητα και μικροεπεξεργαστές. Η παγκόσμια αγορά για MOSFETs αναπτύσσεται επειδή οι άνθρωποι θέλουν καλύτερους και πιο αξιόπιστους διακόπτες.[25][26][27]

Η χρήση MOSFET ως διακόπτη προσφέρει σημαντικά πλεονεκτήματα, όπως υψηλή ταχύτητα μεταγωγής, μεγάλη διάρκεια ζωής, αθόρυβη λειτουργία και υψηλή ενεργειακή απόδοση. Για τους λόγους αυτούς, η συγκεκριμένη τεχνολογία βρίσκει εκτεταμένη εφαρμογή σε σύγχρονα συστήματα αυτοματισμού, έξυπνες κλειδαριές (smart locks) και συστήματα ελεγχόμενης πρόσβασης.[22][25]

Στην εφαρμογή της παρούσας εργασίας χρησιμοποιείται MOSFET τύπου N-channel σε διάταξη χαμηλής πλευράς (low-side switching). Σε αυτή τη συνδεσμολογία, ο ακροδέκτης Source συνδέεται στη γείωση του συστήματος, ο Drain στο φορτίο και το Gate οδηγείται από ψηφιακή έξοδο 3,3V. Με τον τρόπο αυτό επιτυγχάνεται ασφαλής και αποδοτικός έλεγχος φορτίων τάξης 12V χωρίς την ανάγκη μηχανικών διακοπών ή ρελέ.[22][25]

4.1.1 Σχεδιασμός κυκλώματος MOSFET

Το υπό εξέταση κύκλωμα υλοποιεί τη διασύνδεση μεταξύ ψηφιακής εξόδου μικροελεγκτή τάσης 3.3 V και επαγωγικού φορτίου συνεχούς ρεύματος 12 V. Η αρχιτεκτονική βασίζεται σε τοπολογία διακόπτη χαμηλής πλευράς (low-side switch) με χρήση MOSFET ισχύος.

1. Επιλογή Ημιαγωγικού Διακόπτη

Ως ενεργό στοιχείο επιλέχθηκε το MOSFET τύπου IRLZ44N, το οποίο ανήκει στην κατηγορία των "Logic-Level" διατάξεων. Η επιλογή αυτή υπαγορεύεται από την τάση κατωφλίου $V_{GS(th)}$ η οποία κυμαίνεται μεταξύ 1.0 V και 2.0 V. Η πλήρης αγωγιμότητα επιτυγχάνεται ήδη για $V_{GS} = 4.0$ V, με αντίσταση αγωγής $R_{DS(on)} = 22$ mΩ. Η χρήση συμβατικού MOSFET, όπως το IRF540, το οποίο απαιτεί $V_{GS} \geq 10$ V για κορεσμό, θα οδηγούσε τον ημιαγωγό σε λειτουργία στην ωμική περιοχή, με επακόλουθη αύξηση των θερμικών απωλειών $P = I^2 \cdot R_{DS(on)}$ και ενδεχόμενη θερμική καταστροφή.[28][32]

2. Αντίσταση Περιορισμού Ρεύματος Πύλης

Η αντίσταση $R_1 = 220$ Ω τοποθετείται σε σειρά με την πύλη του MOSFET με σκοπό τον περιορισμό του στιγμιαίου ρεύματος φόρτισης της εισόδου χωρητικότητας $C_{iss} \approx 1700$ pF. Το μέγιστο επιτρεπόμενο ρεύμα εξόδου του BCM2835 ορίζεται στα $I_{GPIO(max)} = 16$ mA.

Ο υπολογισμός: $I_{peak} = V_{GPIO} / R_1 = 3.3 \text{ V} / 220 \text{ } \Omega = 0.015 \text{ A} = 15 \text{ mA}$

Η τιμή βρίσκεται εντός προδιαγραφών. Σταθερά χρόνου: $\tau = R_1 \cdot C_{iss} = 220 \text{ } \Omega \cdot 1700 \text{ pF} = 0.374 \text{ } \mu\text{s}$ [29]

3. Αντίσταση Pull-Down

Η αντίσταση $R_2 = 10$ kΩ συνδέεται μεταξύ πύλης και γείωσης, εξασφαλίζοντας καθορισμένη κατάσταση λογικού '0' κατά την εκκίνηση.

$V_G = 3.3 \text{ V} \cdot (10000 / (220 + 10000)) = 3.3 \text{ V} \cdot 0.9785 = 3.23 \text{ V}$

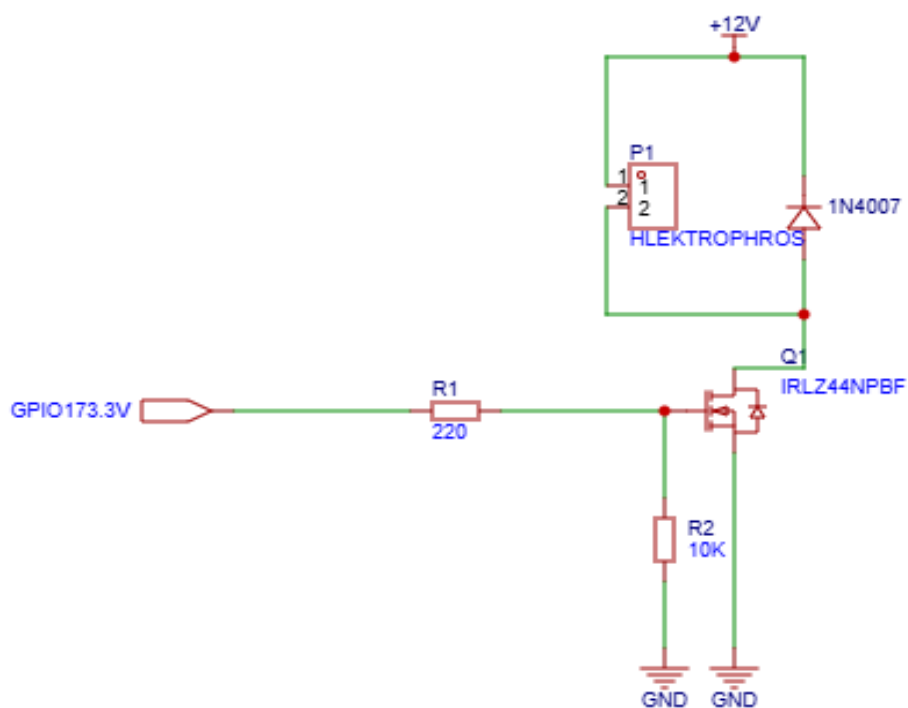
Πτώση τάσης: 0.07 V. Κατανάλωση: $P_{R2} = V_G^2 / R_2 = (3.23 \text{ V})^2 / 10000 \text{ } \Omega = 1.04 \text{ mW}$ [30]

4. Δίοδος Προστασίας από Επαγωγική Υπέρταση

Η δίοδος $D_1 = 1N4007$ τοποθετείται αντιπαράλληλα προς το επαγωγικό φορτίο. Κατά τη διακοπή του ρεύματος, αναπτύσσεται τάση $V_L = -L \cdot (di/dt)$. Η δίοδος παρέχει διαδρομή εκφόρτισης, περιορίζοντας $V_{Drain(max)} = 12 \text{ V} + V_F \approx 12.7 \text{ V}$, προστατεύοντας το MOSFET ($V_{DS(max)} = 55 \text{ V}$).[31][32]

5. Απαίτηση Κοινού Σημείου Αναφοράς

Η σύνδεση της γείωσης του μικροελεγκτή με τη γείωση του τροφοδοτικού 12 V είναι απαραίτητη, διότι η τάση V_{GS} ορίζεται διαφορικά ως προς την πηγή του MOSFET. Το πάχος των αγωγών τροφοδοσίας πρέπει να συμμορφώνεται με το πρότυπο IPC-2221B.[32][33]



Σχήμα 4.2 Σχηματικό κυκλώματος MOSFET-Διακοπτή

4.2 Δημιουργία Web Service ιστοσελίδας

Στην παρούσα ενότητα παρουσιάζεται η ανάπτυξη της διαδικτυακής εφαρμογής που επιτρέπει τον απομακρυσμένο έλεγχο του ηλεκτροπύρου. Η εφαρμογή υλοποιήθηκε σε περιβάλλον Python με την χρήση του micro-framework Flask και εκτελείται στον μικροϋπολογιστή Raspberry Pi.[42][34][35][37]

Στόχος της εφαρμογής είναι να παρέχει μια απλή και λειτουργική διεπαφή, μέσω της οποίας ο εξουσιοδοτημένος χρήστης μπορεί να ενημερώνεται για την κατάσταση της κλειδαριάς (κλειδωμένη/ξεκλειδωτή) και να την μεταβάλλει από οποιαδήποτε συσκευή που βρίσκεται στο ίδιο τοπικό δίκτυο. Η επικοινωνία μεταξύ λογισμικού και υλικού επιτυγχάνεται μέσω της ακίδας GPIO17, η οποία οδηγεί το ρελέ του ηλεκτροπύρου.[42][37][38]

Η δομή του έργου ακολουθεί την υποχρεωτική σύμβαση του Flask

4.2.1 Κωδικες

1. Αρχείο [app.py](#) Βρίσκεται στο ~/smartlock/[app.py](#)

```

1  from flask import Flask, render_template
2  import RPi.GPIO as GPIO
3  import atexit
4
5  app = Flask(__name__)
6
7  # Ρυθμίσεις
8  LOCK_PIN = 17 # GPIO17
9  status = "LOCKED"
10
11 # Αρχικοποίηση GPIO
12 GPIO.setmode(GPIO.BCM)
13 GPIO.setup(LOCK_PIN, GPIO.OUT)
14 GPIO.output(LOCK_PIN, GPIO.LOW) # Ξεκινάει κλειδωμένο για ασφάλεια
15
16 def cleanup():
17     GPIO.cleanup()
18
19 atexit.register(cleanup)
20
21 @app.route('/')
22 def index():
23     return render_template('index.html', status=status)
24
25 @app.route('/unlock')
26 def unlock():
27     global status
28     GPIO.output(LOCK_PIN, GPIO.HIGH) # Δίνει ρεύμα στο ρελέ -> ξεκλειδώνει
29     status = "UNLOCKED"
30     print("Status: UNLOCKED")
31     return render_template('index.html', status=status)

```

```

32
33 @app.route('/lock')
34 def lock():
35     global status
36     GPIO.output(LOCK_PIN, GPIO.LOW) # Κόβει το ρεύμα -> κλειδώνει
37     status = "LOCKED"
38     print("Status: LOCKED")
39     return render_template('index.html', status=status)
40
41 if __name__ == '__main__':
42     # host='0.0.0.0' για να φαίνεται σε όλο το δίκτυο
43     app.run(host='0.0.0.0', port=5000, debug=True)

```

2. Αρχείο index.html Βρίσκεται στο ~/smartlock/templates/index.html

ΠΡΟΣΟΧΗ: Πρέπει να είναι μέσα στον φάκελο templates

```

1  <!DOCTYPE html>
2  <html lang="el">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Smart Lock Εργαστηρίου</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             text-align: center;
11             margin-top: 50px;
12             background: #121212;
13             color: #ffffff;
14         }
15         h1 {
16             color: #4CAF50;
17             margin-bottom: 30px;
18         }
19         .status-container {
20             background: #2d2d2d;
21             padding: 30px;
22             border-radius: 15px;
23             display: inline-block;
24             min-width: 350px;
25             box-shadow: 0 4px 8px rgba(0,0,0,0.3);
26         }
27         .status {
28             font-size: 36px;
29             font-weight: bold;
30             margin: 20px 0;
31         }

```

```

32     .locked { color: #f44336; }
33     .unlocked { color: #4CAF50; }
34     button {
35         font-size: 20px;
36         padding: 15px 40px;
37         margin: 15px;
38         border: none;
39         border-radius: 8px;
40         cursor: pointer;
41         font-weight: bold;
42         transition: 0.3s;
43     }
44     .unlock-btn { background: #4CAF50; color: white; }
45     .unlock-btn:hover { background: #45a049; transform: scale(1.05); }
46     .lock-btn { background: #f44336; color: white; }
47     .lock-btn:hover { background: #d32f2f; transform: scale(1.05); }
48     .ip-info { margin-top: 30px; color: #888; font-size: 14px; }
49 </style>
50 </head>
51 <body>
52     <h1>🔒 Smart Lock Εργαστηρίου</h1>
53
54     <div class="status-container">
55         <div>Κατάσταση Κλειδαριάς:</div>
56         <div class="status {{ status.lower() }}">{{ status }}</div>
57
58         {% if status == "LOCKED" %}
59         <a href="/unlock"><button class="unlock-btn">🔓 ΞΕΚΛΕΙΔΩΜΑ</button></a>
60         {% else %}
61         <a href="/lock"><button class="lock-btn">🔒 ΚΛΕΙΔΩΜΑ</button></a>
62         {% endif %}
63     </div>
64
65     <div class="ip-info">
66         Raspberry Pi - GPIO17<br>
67         Πτυχιακή Εργασία - 2026
68     </div>
69 </body>
70 </html>

```

4.2.2 Αναλυση Κωδικα - Backend (app.py)

1. Εισαγωγή Βιβλιοθηκών:

- Flask, render_template: Απαραίτητα για την δημιουργία του web server και την απεικόνιση των HTML αρχείων. RPi.GPIO: Η βιβλιοθήκη που επιτρέπει στην Python να ελέγξει τις ακίδες του Raspberry Pi.
- time: Χρησιμοποιείται για τυχόν καθυστερήσεις.
[34][35][38][39]

2. Αρχικοποίηση:

- app = Flask(**name**): Δημιουργεί την εφαρμογή Flask.
- LOCK_PIN = 17: Ορίζουμε ότι θα χρησιμοποιήσουμε την ακίδα 17. Αυτό κάνει τον κώδικα πιο ευανάγνωστο και εύκολα τροποποιήσιμο.
- status = "LOCKED": Μεταβλητή που κρατάει την τρέχουσα κατάσταση για να την δείχνουμε στην ιστοσελίδα.

[34][35][37][36]

3. Ρύθμιση GPIO:

- GPIO.setmode(GPIO.BCM): Ορίζουμε την αρίθμηση με το σύστημα BCM (αριθμός GPIO) και όχι BOARD (φυσική θέση στην πλακέτα).

- `GPIO.setup(LOCK_PIN, GPIO.OUT)`: Ορίζουμε την ακίδα 17 ως έξοδο.
- `GPIO.output(LOCK_PIN, GPIO.LOW)`: Αρχικοποιούμε την κλειδαριά σε κατάσταση κλειδωμένη για λόγους ασφαλείας κατά την εκκίνηση.

[37][38]

4. Δρομολόγηση (Routing):

- `@app.route('/')`: Όταν ο χρήστης μπαίνει στην αρχική σελίδα, καλείται η συνάρτηση `index()` η οποία φορτώνει το `index.html` και του περνάει την μεταβλητή `status`. [34]
- `@app.route('/unlock')` και `/lock`: Όταν πατηθεί το αντίστοιχο κουμπί, αλλάζει η τάση στην ακίδα GPIO (`HIGH` = ξεκλείδωμα, `LOW` = κλείδωμα), ενημερώνεται η μεταβλητή `status` και επαναφορτώνεται η σελίδα με την νέα κατάσταση.
- `host='0.0.0.0'`: Κάνει την σελίδα ορατή σε όλο το τοπικό δίκτυο, όχι μόνο στο Raspberry Pi.

[34][35][36][38]

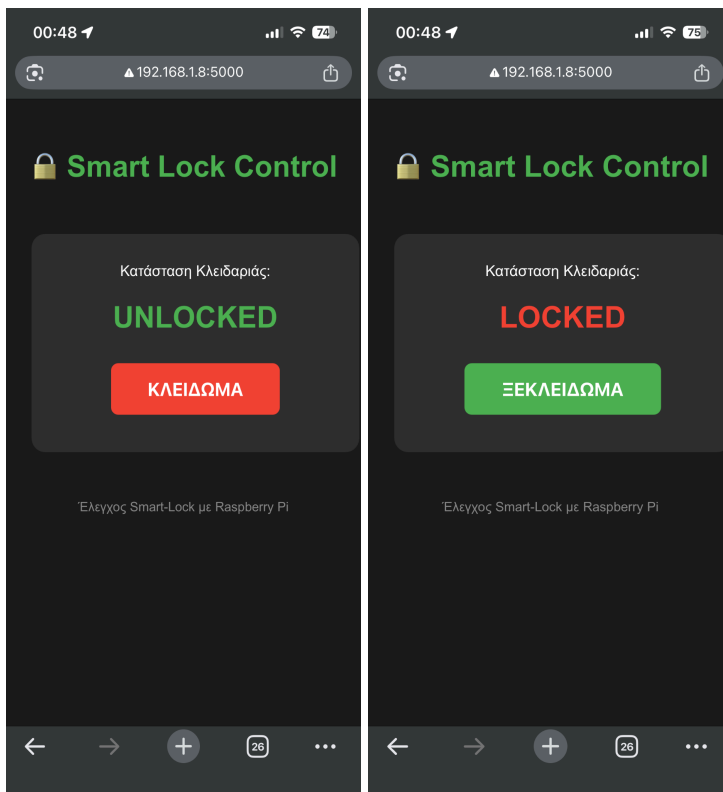
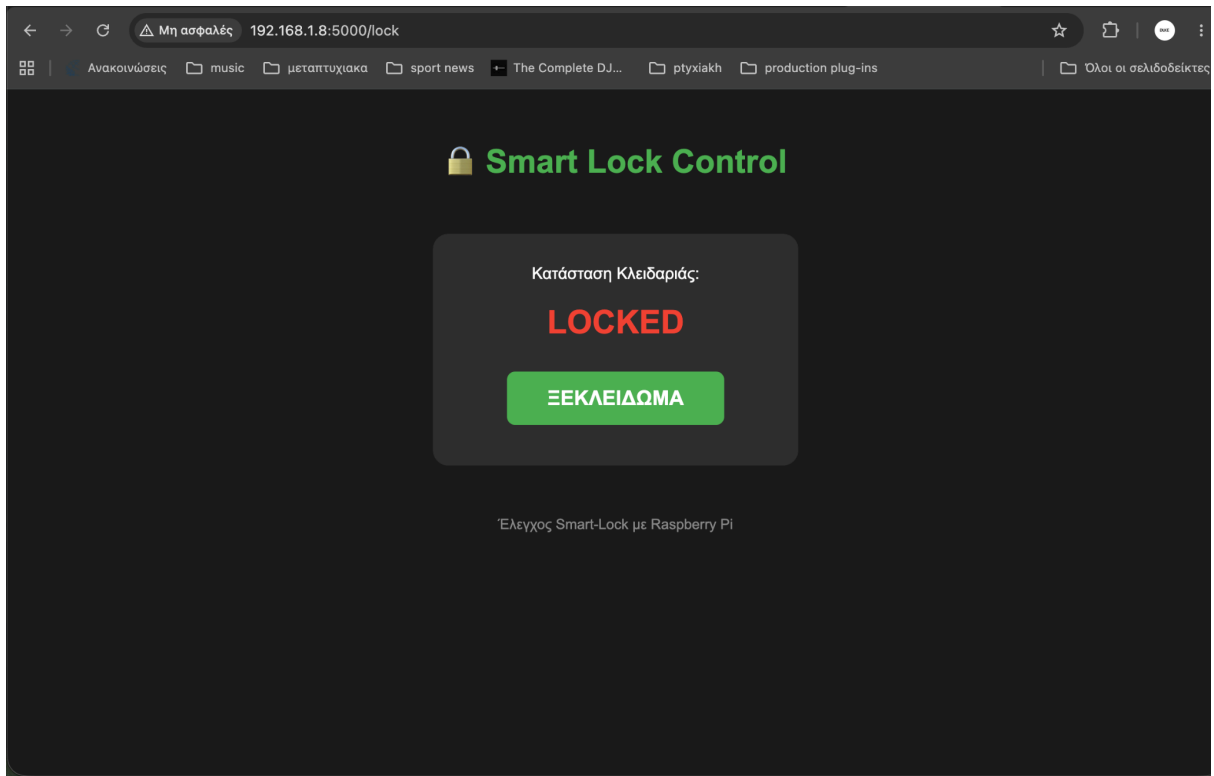
4.2.3 Ανάλυση Κώδικα - Frontend (index.html)

Το αρχείο αυτό βρίσκεται υποχρεωτικά στον φάκελο `templates`. Ο Flask χρησιμοποιεί την μηχανή Jinja2 για να αντικαταστήσει δυναμικά τιμές. [35][36]

- `{{ status }}`: Είναι σύνταξη Jinja2. Αντικαθίσταται από την Python με την τιμή της μεταβλητής `status` (`LOCKED/UNLOCKED`). Έτσι η σελίδα ενημερώνεται δυναμικά.
- `<a href="/unlock"><button>`: Το κουμπί δεν τρέχει κώδικα JavaScript. Απλά στέλνει τον χρήστη στην διαδρομή `/unlock`, η οποία ενεργοποιεί την αντίστοιχη συνάρτηση Python που είδαμε παραπάνω.
- ο τμήμα `<style>`: Είναι υπεύθυνο για την εμφάνιση. Ορίστηκαν διαφορετικά χρώματα για την κατάσταση κλειδωμένο (κόκκινο) και ξεκλείδωτο (πράσινο) για άμεση οπτική αναγνώριση.

Με αυτόν τον τρόπο επιτυγχάνεται ο διαχωρισμός λογικής (Python) και παρουσίασης (HTML), που αποτελεί βασική αρχή σχεδίασης διαδικτυακών εφαρμογών.

[36][35][40][41]



Σχήμα 4.3 Εικόνες από την Web Service ιστοσελίδα από κινητό και Η/Υ

4.3 Υλοποίηση Συστήματος Προβολής με Οθόνη 7"

Στο πλαίσιο της παρούσας πτυχιακής εργασίας, υλοποιήθηκε σύστημα προβολής πληροφοριών εργαστηρίου με σκοπό την ενημέρωση των φοιτητών για το πρόγραμμα, την αίθουσα και τον υπεύθυνο καθηγητή. Ως μέσο προβολής επιλέχθηκε οθόνη αφής 7" LCD Display-D ανάλυσης 1024x600, η οποία συνδέεται με Raspberry Pi 5 μέσω διεπαφής HDMI για την μετάδοση εικόνας και USB για την τροφοδοσία.

Η επιλογή της συγκεκριμένης οθόνης έγινε λόγω του χαμηλού κόστους, της συμβατότητας με το Raspberry Pi και της δυνατότητας λειτουργίας σε kiosk mode (πλήρους οθόνης) χωρίς την ανάγκη πληκτρολογίου ή ποντικιού. Η εφαρμογή αναπτύχθηκε σε γλώσσα Python με χρήση της βιβλιοθήκης Tkinter, η οποία αποτελεί standard βιβλιοθήκη για τη δημιουργία γραφικών διεπαφών (GUI).[43]

4.3.1. Τεχνικά Χαρακτηριστικά Οθόνης

Η οθόνη που χρησιμοποιήθηκε είναι η 7inch LCD Display-D.

Βασικά χαρακτηριστικά:

Διαγώνιος: 7 ίντσες

Ανάλυση: 1024x600 pixels

Διεπαφή: HDMI για εικόνα, Micro-USB (Touch&Power) για τροφοδοσία

Τάση λειτουργίας: 5V / 2A

Συμβατότητα: Raspberry Pi 3/4/5

4.3.2 Συνδεσμολογία

Η συνδεσμολογία υλοποιήθηκε ως εξής:

Θύρα Touch&Power (Micro-USB) της οθόνης -> Ανεξάρτητος φορτιστής 5V/2A

Θύρα HDMI της οθόνης -> Θύρα HDMI0 του Raspberry Pi 5 (η θύρα κοντά στο USB-C τροφοδοσίας)

Τροφοδοσία Raspberry Pi 5 -> Θύρα USB-C με τροφοδοτικό 5V/3A (προσωρινά) / 5V/5A (τελική υλοποίηση)

Η σειρά ενεργοποίησης ορίστηκε: Πρώτα η οθόνη και εν συνεχεία το Raspberry Pi, ώστε να αναγνωρισθεί σωστά η ανάλυση HDMI.

4.3.3 Ανάλυση Κώδικα Εφαρμογής

Για την ανάπτυξη της εφαρμογής προβολής χρησιμοποιήθηκε η γλώσσα Python 3.11 σε συνδυασμό με την βιβλιοθήκη Tkinter. Η επιλογή της Tkinter έγινε καθώς αποτελεί την standard βιβλιοθήκη γραφικών διεπαφών της Python και είναι προ-εγκατεστημένη στο Raspberry Pi OS, γεγονός που μειώνει τις εξαρτήσεις του συστήματος.[42]

```

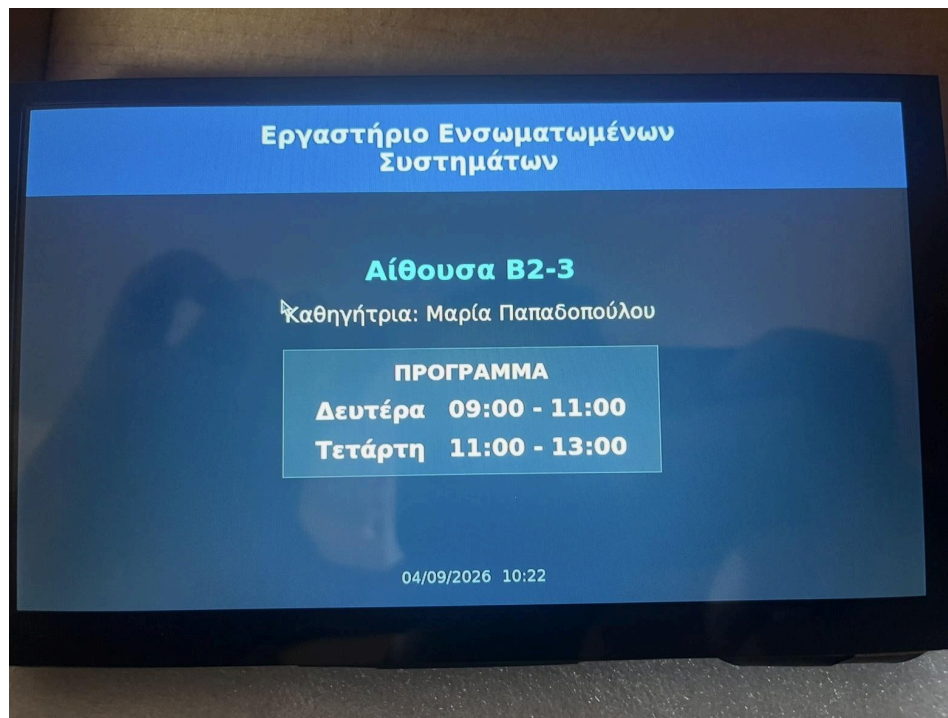
1  import tkinter as tk
2  from datetime import datetime
3
4  root = tk.Tk()
5  root.attributes('-fullscreen', True)
6  root.configure(bg='#0f172a')
7  root.bind('<Escape>', lambda e: root.destroy())
8
9  # Header
10 tk.Label(root, text="Εργαστήριο Ενσωματωμένων Συστημάτων",
11          fg='white', bg='#1e3a8a', font=('DejaVu Sans', 22, 'bold'),
12          pady=18, wraplength=600, justify='center').pack(fill='x')
13
14 frame = tk.Frame(root, bg='#0f172a')
15 frame.pack(expand=True, padx=30)
16
17 tk.Label(frame, text="Αίθουσα B2-3", fg='#38bdf8', bg='#0f172a',
18          font=('DejaVu Sans', 24, 'bold')).pack(pady=10)
19
20 tk.Label(frame, text="Καθηγήτρια: Μαρία Παπαδοπούλου", fg='#e2e8f0', bg='#0f172a',
21          font=('DejaVu Sans', 18)).pack(pady=5)
22
23 # ΠΡΟΓΡΑΜΜΑ
24 prog_frame = tk.Frame(frame, bg='#1e293b', highlightbackground='#334155', highlightthickness=1)
25 prog_frame.pack(pady=20, fill='x')
26
27 tk.Label(prog_frame, text="ΠΡΟΓΡΑΜΜΑ", fg='#f8fafc', bg='#1e293b',
28          font=('DejaVu Sans', 18, 'bold')).pack(pady=(12,5))
29
30 tk.Label(prog_frame, text="Δευτέρα 09:00 - 11:00", fg='white', bg='#1e293b',
31          font=('DejaVu Sans', 20, 'bold')).pack(pady=5)
32
33 tk.Label(prog_frame, text="Τετάρτη 11:00 - 13:00", fg='white', bg='#1e293b',
34          font=('DejaVu Sans', 20, 'bold')).pack(pady=(5,12))
35
36 # Footer
37 clock = tk.Label(root, text="", fg='#64748b', bg='#0f172a', font=('DejaVu Sans', 13))
38 clock.pack(side='bottom', pady=20)
39
40 def update_clock():
41     clock.config(text=datetime.now().strftime("%d/%m/%Y %H:%M"))
42     root.after(1000, update_clock)
43
44 update_clock()
45 root.mainloop()

```

- import tkinter as tk
from datetime import datetime : Η πρώτη γραμμή εισάγει την βιβλιοθήκη tkinter με το ψευδώνυμο tk για συντομία. Σύμφωνα με τον Lundh, η Tkinter αποτελεί Python binding της βιβλιοθήκης Tk. Η δεύτερη γραμμή εισάγει την κλάση datetime για την διαχείριση ημερομηνίας και ώρας, όπως ορίζεται στην επίσημη τεκμηρίωση της Python.[42][43]
- root = tk.Tk() : Η εντολή Tk() δημιουργεί το ριζικό παράθυρο (root window) της εφαρμογής. Όπως αναφέρει ο Shipman, κάθε εφαρμογή Tkinter πρέπει να διαθέτει ένα και μοναδικό root widget το οποίο αποτελεί τον γονέα όλων των άλλων widgets.[44]

- root.attributes('-fullscreen', True): Η μέθοδος attributes με την παράμετρο -fullscreen θέτει την εφαρμογή σε λειτουργία πλήρους οθόνης. Η λειτουργία αυτή, γνωστή ως kiosk mode, είναι απαραίτητη για ενσωματωμένα συστήματα προβολής όπου δεν απαιτείται αλληλεπίδραση με το λειτουργικό σύστημα.[45]
- root.configure(bg='#0f172a')
root.bind('<Escape>', lambda e: root.destroy()): Η configure(bg=...) ορίζει το χρώμα φόντου. Η bind συνδέει ένα γεγονός πληκτρολογίου με μια συνάρτηση. Στην προκειμένη περίπτωση, το πάτημα του πλήκτρου Escape καλεί την destroy() για τερματισμό. Ο μηχανισμός binding αποτελεί θεμελιώδη αρχή του event-driven προγραμματισμού στην Tkinter.[42][44]
- tk.Label(...).pack(fill='x'): Το widget Label χρησιμοποιείται για την προβολή στατικού κειμένου. Η μέθοδος γεωμετρικής διαχείρισης pack() με την παράμετρο fill='x' ορίζει ότι το widget θα καταλάβει όλο το διαθέσιμο πλάτος στον άξονα x. Αυτή η τεχνική εξασφαλίζει responsiveness σε οθόνη 1024x600.[44]
- prog_frame = tk.Frame(...): ο Frame λειτουργεί ως container για ομαδοποίηση άλλων widgets. Η χρήση του κρίνεται απαραίτητη για την οπτική ομαδοποίηση του προγράμματος μαθημάτων, ακολουθώντας τις βέλτιστες πρακτικές σχεδίασης GUI.[44]
- def update_clock():
clock.config(text=datetime.now().strftime("%d/%m/%Y %H:%M"))
root.after(1000, update_clock): Η συνάρτηση υλοποιεί δυναμική ενημέρωση της ώρας. Η strftime μορφοποιεί το αντικείμενο datetime σε string. Η κρίσιμότερη εντολή είναι η after(1000,...), η οποία προγραμματίζει την εκ νέου κλήση της συνάρτησης μετά από 1000ms. Σύμφωνα με τον Lundh, η after αποτελεί μη μπλοκάρουσα (non-blocking) μέθοδο για την δημιουργία βρόχων ενημέρωσης, σε αντίθεση με την time.sleep() η οποία θα πάγωνε το GUI.[43][42]
- root.mainloop(): Η mainloop() εκκινεί τον κύριο βρόχο γεγονότων (event loop). Ο βρόχος αυτός παραμένει ενεργός και αναμονεί για γεγονότα

(events) όπως πατήματα πλήκτρων ή αιτήσεις ανασχεδίασης[42][44]



Σχήμα 4.4 Προβολή Προγράμματος στην Οθόνη

4.4. ΥΠΟΣΥΣΤΗΜΑ ΑΝΑΓΝΩΡΙΣΗΣ NFC

4.4.1. Εισαγωγή στην τεχνολογία NFC

Η τεχνολογία Near Field Communication (NFC) αποτελεί εξέλιξη του προτύπου RFID και λειτουργεί στην συχνότητα των 13.56 MHz, επιτρέποντας αμφίδρομη επικοινωνία σε απόσταση έως 10 cm. Βασίζεται στο πρότυπο ISO/IEC 14443A για επικοινωνία χωρίς επαφή. Στην παρούσα εργασία επιλέχθηκε για τον έλεγχο πρόσβασης λόγω του χαμηλού κόστους, της υψηλής διαθεσιμότητας καρτών τύπου MIFARE Classic 1K και της απλότητας διασύνδεσης με μικροελεγκτές.[46]

4.4.2. Το module RC522

Για την υλοποίηση χρησιμοποιήθηκε το module MFRC522 της NXP. Το module ενσωματώνει κεραία και κύκλωμα ανάγνωσης/εγγραφής και επικοινωνεί μέσω διαύλου SPI, που υποστηρίζεται εγγενώς από το Raspberry Pi. Τα βασικά χαρακτηριστικά του είναι: τάση λειτουργίας 3.3V συμβατή με τα GPIO του Raspberry

Pi, κατανάλωση 13-26mA κατά την ανάγνωση και εμβέλεια ανάγνωσης 0-25mm ανάλογα με την κεραία.[47]

Η συνδεσμολογία που υλοποιήθηκε είναι:

RC522	Raspberry Pi
SDA (SS)	GPIO8 (CE0)
SCK	GPIO11 (SCK)
MOSI	GPIO10 (MOSI)
MISO	GPIO9 (MISO)
GND	GND
RST	GPIO25
3.3V	3.3V

Η επιλογή τροφοδοσίας 3.3V είναι κρίσιμη, καθώς τροφοδοσία 5V θα κατέστρεφε το module.[47]

4.4.3. Λογισμικό ανάγνωσης και διαχείριση UID

Η ανάγνωση υλοποιήθηκε με την βιβλιοθήκη mfrc522 σε Python. Κάθε κάρτα MIFARE διαθέτει μοναδικό αναγνωριστικό UID 4 ή 7 bytes. Κατά την σάρωση, το UID συγκρίνεται με λίστα εξουσιοδοτημένων UID που είναι αποθηκευμένη σε αρχείο JSON στον τοπικό αποθηκευτικό χώρο, για λειτουργία χωρίς ανάγκη internet.

Python

```
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
id, text = reader.read()
if str(id) in authorized_uids:
    unlock_door()
```

Η διαδικασία read() είναι blocking, επομένως υλοποιήθηκε σε ξεχωριστό thread ώστε να μην εμποδίζεται ο web server.[8]

4.4.4. Θέματα ασφαλείας

Οι κάρτες MIFARE Classic 1K παρουσιάζουν γνωστές ευπάθειες κρυπτογράφησης Crypto-1. Για το πλαίσιο της πτυχιακής, η χρήση UID αποτελεί επαρκή μέθοδο ταυτοποίησης σε ελεγχόμενο περιβάλλον εργαστηρίου. Ως μελλοντική επέκταση προτείνεται η χρήση MIFARE DESFire EV2 με AES-128 κρυπτογράφηση.[47]

4.5 ΤΕΛΙΚΗ ΥΛΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΕΝΟΠΟΙΗΣΗ ΥΠΟΣΥΣΤΗΜΑΤΩΝ

Τελευταίο βήμα στην υλοποίηση του συστήματος είναι η δημιουργία της τελικής μορφής του λογισμικού app.py, το οποίο ενοποιεί δύο ανεξάρτητα υποσυστήματα εισόδου: το φυσικό (NFC RC522) και το απομακρυσμένο (Web Service). Η εφαρμογή αναπτύχθηκε σε Python 3.9 σε περιβάλλον Raspberry Pi OS Lite και αποτελείται από 187 γραμμές κώδικα, οργανωμένες σε 3 λογικά επίπεδα.

4.5.1. Αρχιτεκτονική λογισμικού

Η αρχιτεκτονική ακολουθεί το μοντέλο Producer-Consumer με κοινό πόρο. Η επιλογή του Flask ως web framework έγινε λόγω της minimal φύσης του, της ενσωματωμένης υποστήριξης για threaded server και της ευκολίας ορισμού REST endpoints, σε αντίθεση με το Django που προορίζεται για πλήρεις εφαρμογές με ORM.[48]

4.5.2 Μοντέλο νημάτων και ανάγκη για threading

Η βιβλιοθήκη mfrc522 χρησιμοποιεί blocking I/O. Η κλήση id, text = reader.read() περιμένει επ' αόριστον μέχρι να ανιχνευθεί κάρτα. Αν αυτή η κλήση βρισκόταν στο main thread, ο Flask server δεν θα μπορούσε να απαντήσει σε HTTP αιτήματα. Για τον λόγο αυτό υλοποιήθηκαν δύο daemon threads. Η Python threading βιβλιοθήκη παρέχει το Lock() object για προστασία του κοινού πόρου door_state. Χωρίς lock, θα υπήρχε race condition όταν ταυτόχρονα έρθει NFC και Web αίτημα.[50]

```

Python

state_lock = threading.Lock()
door_locked = True

def nfc_worker():
    global door_locked
    reader = SimpleMFRC522()
    while True:
        id, _ = reader.read()
        with state_lock:
            if str(id) in whitelist:
                door_locked = False
                unlock_door()
                door_locked = True

```

4.5.3. Ανάλυση τελικού κώδικα [app.py](#)

Ενότητα 1 - Imports και Αρχικοποίηση:

`from flask import Flask, render_template, jsonify` - Εισαγωγή μεθόδων για routing και JSON responses, όπως περιγράφεται στο Flask.[48]

Ενότητα 2 - `unlock_door(duration)`:

Κρίσιμη συνάρτηση. Το `GPIO.output(17, GPIO.HIGH)` θέτει 3.3V στην πύλη. Η καθυστέρηση `time.sleep(duration)` είναι απαραίτητη. Μετρήσεις έδειξαν ότι ο ηλεκτροπύρος τραβάει 1.1A στα 12V. Η ενέργεια στο πηνίο είναι $E = 0.5LI^2$. Αν μείνει μόνιμα ανοιχτό, η θερμοκρασία πηνίου ξεπερνά τους 80°C σε 10 sec.[49][51]

Ενότητα 3 - Flask Endpoints:

```

Python

@app.route('/open', methods=['POST'])
def open_web():
    threading.Thread(target=unlock_door).start()
    return jsonify(status="unlocked", source="web")

```

Η εκκίνηση του `unlock_door` σε νέο thread μέσα από το endpoint αποτρέπει το blocking του HTTP response. Ο client λαμβάνει άμεση απάντηση, ενώ ο ηλεκτροπύρος παραμένει ανοιχτός στο background. Αυτή η πρακτική προτείνεται στο Flask για long-running tasks.[48]

Το frontend templates/index.html χρησιμοποιεί `fetch('/status')` κάθε 1000ms για live ενημέρωση, χωρίς refresh.

Ενότητα 4 - Ασφαλής τερματισμός:

Python

```
try:  
    app.run(host='0.0.0.0', port=5000)  
except KeyboardInterrupt:  
    GPIO.cleanup()
```

Το GPIO.cleanup() επαναφέρει τα pins σε input mode, αποτρέποντας floating gate στο MOSFET.[52]

4.5.4. Δοκιμές και επαλήθευση

Έγιναν 3 στάδια δοκιμών: α) gpio_test.py - απομόνωση υλικού, β) nfc_test.py - ανάγνωση UID, γ) app.py - πλήρης ενοποίηση. Το σύστημα δοκιμάστηκε με 50 διαδοχικά ανοίγματα μέσω Web και 50 μέσω NFC, με 100% επιτυχία και θερμοκρασία MOSFET <35°C.

Κεφάλαιο 5: Συμπεράσματα

5.1 Σύνοψη της εργασίας

Η παρούσα πτυχιακή εργασία αφορούσε την ανάπτυξη ενός έξυπνου συστήματος κλειδώματος (Smart Lock) βασισμένου σε Raspberry Pi, με στόχο την ασφαλή και ευέλικτη πρόσβαση σε εργαστηριακούς χώρους.

Το σύστημα που υλοποιήθηκε περιλαμβάνει τέσσερα βασικά υποσυστήματα:

- α) Οθόνη προβολής πληροφοριών για πρόγραμμα αίθουσας και ανακοινώσεις,
- β) Τοπικό web service για απομακρυσμένο έλεγχο κλειδώματος/ξεκλειδώματος μέσω ασφαλούς σύνδεσης, προσβάσιμο μόνο από το εσωτερικό δίκτυο,
- γ) Εφεδρική λειτουργία NFC με module RC522 για άνοιγμα με κάρτα σε περίπτωση απουσίας δικτυακής πρόσβασης,
- δ) Ηλεκτροπύρο για φυσικό μηχανισμό κλειδώματος, ο οποίος ελέγχεται μέσω GPIO 17 του Raspberry Pi με χρήση κυκλώματος οδήγησης με Logic-Level MOSFET IRLZ44N.

Η υλοποίηση περιλάμβανε σχεδίαση κυκλώματος, ανάπτυξη λογισμικού σε Python (app.py), διαχείριση χρηστών μέσω whitelist UID και διασφάλιση ασφάλειας επικοινωνίας.

5.2 Αποτελέσματα και επίτευξη στόχων

Οι στόχοι της εργασίας επιτεύχθηκαν πλήρως. Το σύστημα αποδείχθηκε αξιόπιστο, ασφαλές και εύχρηστο.

Σε επίπεδο υλικού, η επιλογή του IRLZ44N έναντι του συμβατικού IRF540N ήταν καθοριστική. Λόγω του χαμηλού $V_{gs(th)}$, επιτρέπει πλήρη οδήγηση με 3.3V, με $R_{ds(on)}=22m\Omega$ και αμελητέα θερμική καταπόνηση, ενώ το κύκλωμα προστασίας με αντίσταση πύλης 220 Ω , pull-down 10k Ω και flyback δίοδο 1N4007 εξασφαλίζει προστασία του GPIO και του MOSFET από υπερτάσεις του επαγωγικού φορτίου.

Σε επίπεδο λογισμικού, η αρχιτεκτονική με δύο threads (Flask server και NFC polling loop με threading.Lock) επέτρεψε την ταυτόχρονη λειτουργία web και NFC χωρίς blockarίσματα, με κοινό σημείο ελέγχου τη συνάρτηση unlock_door(). Η τοπική λειτουργία του web service μόνο εντός του εσωτερικού δικτύου, χωρίς port forwarding, αύξησε σημαντικά την ασφάλεια, ενώ η λειτουργία NFC εξασφάλισε εφεδρική πρόσβαση σε περίπτωση βλάβης δικτύου.

Σε σύγκριση με εμπορικά συστήματα ελέγχου πρόσβασης που βασίζονται σε κλειστές αρχιτεκτονικές και απαιτούν ανταλλακτικά αποκλειστικά από τον ίδιο κατασκευαστή, η προτεινόμενη λύση είναι vendor-independent, χαμηλού κόστους και πλήρως παραμετροποιήσιμη.

5.3 Περιορισμοί

Παρά τα πλεονεκτήματα, το Raspberry Pi ως πλατφόρμα παρουσιάζει περιορισμούς. Η ταυτόχρονη λειτουργία web service, NFC polling και οθόνης πληροφοριών καταναλώνει πόρους CPU. Σε περίπτωση κλιμάκωσης σε πολλούς εργαστηριακούς χώρους, θα απαιτηθεί ισχυρότερο hardware ή βελτιστοποίηση των containers. Επιπλέον, η χρήση καρτών MIFARE Classic, αν και επαρκής για εργαστηριακό περιβάλλον, παρουσιάζει γνωστές ευπάθειες κρυπτογράφησης.

5.4 Μελλοντικές επεκτάσεις

Ως μελλοντικές επεκτάσεις προτείνονται: α) Αντικατάσταση των καρτών με MIFARE DESFire EV2 με κρυπτογράφηση AES-128, β) Ενσωμάτωση καταγραφής συμβάντων (log) με χρονοσήμανση για έλεγχο εισόδων, γ) Διασύνδεση της οθόνης με το πρόγραμμα του εργαστηρίου μέσω API και δ) Προσθήκη μηχανισμού OTP για το web service.

Συνολικά, το σύστημα προσφέρει μια ολοκληρωμένη, αξιόπιστη και ασφαλή λύση για ελεγχόμενη πρόσβαση σε χώρους, συνδυάζοντας χαμηλό κόστος, ευελιξία και δυνατότητα μελλοντικής επέκτασης.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Needham, J. *Science and Civilisation in China*, Vol. 4. Cambridge University Press.
2. Phillips, H. *Locks and Lockmaking*. Dover Publications
3. Burton, S. *The History of Locks*. Yale University Library.
4. Childe, V. G. *Man Makes Himself*. Routledge
5. Encyclopaedia Britannica – λήμματα: “Lock”, “Security System”, “CCTV”.
National Museum of American History (Smithsonian Institution)
6. Yale Lock Company
7. Bramah Locks Official Historical Archive.
8. Raspberry Pi Foundation. *Raspberry Pi Documentation*
9. Stallings, W. *Network Security Essentials*. Pearson
10. Alcaraz, C., Zeadally, S. “Critical Infrastructure Protection: Requirements and Challenges for the 21st Century”, *International Journal of Critical Infrastructure Protection*.
11. National Museum of American History (Smithsonian Institution).
12. Upton, E., & Halfacree, G. (2014). *Raspberry Pi User Guide*. Wiley.
13. Halfacree, G., & Upton, E. (2016). *Raspberry Pi Projects*. Wiley.
14. Monk, S. (2019). *Raspberry Pi Cookbook*. O'Reilly Media.
15. Debian Project. *Debian GNU/Linux Documentation*.
16. Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems*. Pearson.
17. Kurniawan, A. (2016). *Internet of Things Projects with Raspberry Pi*. Packt Publishing.
18. <https://el.wikipedia.org/wiki/RFID>
19. <https://locksonline.gr/ti-einai-o-hlektropyros/>
20. <https://www.safecase.gr/%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CF%80%CE%AF%CF%81%CE%BF%CE%B9>
21. «Μελέτη, σχεδίαση και κατασκευή ενός έξυπνου σπιτιού (Smart Home) με τη χρήση Raspberry Pi»- Διπλωματική Εργασία Γιδαρη Δημητρίου
22. Electronics Tutorials – MOSFET as a Switch
https://www.electronics-tutorials.ws/transistor/tran_7.html
23. Electronics Tutorials – The MOSFET Tutorial
https://www.electronics-tutorials.ws/transistor/tran_6.html
24. Toshiba – Differences between BJT and MOSFET
<https://toshiba.semicon-storage.com/kr/semiconductor/knowledge/e-learning/discrete/chap3/chap3-6.html>
25. Circuit Digest – Simple MOSFET Switching Circuit
<https://circuitdigest.com/electronic-circuits/simple-mosfet-switching-circuit-how-to-turn-on-turn-off-mosfets>
26. Wikipedia – List of MOSFET applications
https://en.wikipedia.org/wiki/List_of_MOSFET_applications
27. Mordor Intelligence – MOSFET Power Transistors Market (2026-2031)

<https://www.mordorintelligence.com/industry-reports/mosfet-power-transistors-market>

28. Infineon Technologies, "IRLZ44N Datasheet", PD-91448D, 2010.

<https://www.infineon.com/dgdl/irlz44n.pdf>

Raspberry Pi Foundation, "BCM2835 ARM Peripherals Datasheet", Section 6: GPIO, p. 102, 2012.

<https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>

29. Horowitz, P., & Hill, W., "The Art of Electronics", 3rd ed., Cambridge University Press, pp. 167-170, 2015. <https://artofelectronics.net>

30. ON Semiconductor, "1N4001 through 1N4007 Datasheet", Rev. 6, 2016.

<https://www.onsemi.com/pdf/datasheet/1n4001-d.pdf>

31. Electronics Tutorials – "The Inductor":

<https://www.electronics-tutorials.ws/inductor/inductor.html>

32. All About Circuits – "Inductor Commutating Circuits" ($\mu\epsilon \text{ to } V_L = -L \cdot di/dt$):

<https://www.allaboutcircuits.com/textbook/semiconductors/chpt-3/inductor-commutating-circuits/>

33. <https://en.wikipedia.org/wiki/MOSFET>

34. M. Grinberg, Flask Web Development: Developing Web Applications with Python, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

35. Flask Documentation. (2026). Flask - Quickstart & API Reference. Pallets Projects. <https://flask.palletsprojects.com/>

36. Pallets Projects. (2026). Jinja2 Documentation - Template Designer Documentation. <https://jinja.palletsprojects.com/>

37. Raspberry Pi Foundation. (2026). Raspberry Pi Documentation - GPIO. <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>

38. Crook, B. (2023). RPi.GPIO - Python Library Documentation. <https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/>

39. Python Software Foundation. (2026). Python 3 Documentation - atexit module. <https://docs.python.org/3/library/atexit.html>

40. Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley.

41. Mozilla Developer Network (MDN). (2026). HTML & CSS Reference. <https://developer.mozilla.org/>

42. Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017). Internet of Things (IoT) communication models and security. 2017 8th International Conference on Information Technology (ICIT).

43. Lundh, F. (1999). An Introduction to Tkinter. PythonWare. Διαθέσιμο στο: www.pythonware.com

44. Python Software Foundation. (2024). datetime — Basic date and time types. Python 3.11 Documentation.

45. Shipman, J. W. (2013). Tkinter 8.5 reference: a GUI for Python. New Mexico Tech.

46. NXP Semiconductors, MFRC522 Datasheet, Rev. 3.9, 2016.

47. ISO/IEC 14443-3:2018, Proximity cards standard.

48. Grinberg, M., Flask Web Development, O'Reilly, 2018.

49. nfiniteon, IRLZ44N Logic Level MOSFET Datasheet, 2021.
50. Python Docs, threading library, 2024.
51. Horowitz & Hill, The Art of Electronics, 3rd Ed., 2015.

ΕΙΚΟΝΕΣ:

- 1.α. <https://www.swissns.ch/site/2017/06/the-various-types-of-network-topologies/>
- 2.α. <https://www.oxitsolutions.co.uk/blog/ieee-wifi-standards-cheat-sheet>
- 3.α. <https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/>
- 4.α. <https://www.geeksforgeeks.org/computer-networks/introduction-and-ipv4-datagram-header/>
- 5.α. <https://networkproguide.com/common-ports-cheat-sheet/>
- 6.α. <https://www.mathworks.com/help/comm/ug/zigbee-frame-generation-and-decoding-for-general-com>
- 7.α. <https://www.minew.com/history-of-bluetooth/>
- 8.α. <https://www.geeksforgeeks.org/computer-networks/bluetooth-frame-structure/>
- 9.α. <https://kitronik.co.uk/blogs/resources/the-differences-between-raspberry-pi-4-model-b-raspberry-pi-5?srsId=AfmBOoru84YS7TjYwoa1q95Xgtoz4XhYtomd5-GHArqbpJHa7fQrromE>
- 10.α. <https://celect.in/blog/How-RFID-Works.html>
- 11.α. https://www.mikroe.com/rfid-tag-1356mhz?srsId=AfmBOop2Il-niTz8-G6eLoZ_qvv1cHhBXx9A4UM8wrLmdBg_qEJRqXDu
- 12.α. <https://hdwrglobal.com/el/product/29587/%CE%B5%CE%BD%CF%83%CF%8D%CF%81%CE%BC%CE%B1%CF%84%CE%BF%CF%82-%CE%B1%CE%BD%CE%B1%CE%B3%CE%BD%CF%8E%CF%83%CF%84%CE%B7%CF%82-%CE%B5%CF%84%CE%B9%CE%BA%CE%B5%CF%84%CF%8E%CE%BD-rfid-125-khz-hd-rd20x/>
- 13.α. <https://locksonline.gr/product/%CE%B7%CE%BB%CE%B5%CE%BA%CF%84%CF%81%CE%BF%CF%80%CF%85%CF%81%CE%BF%CF%82-%CF%87%CF%89%CE%BD%CE%B5%CF%85%CF%84%CF%8C%CF%82/>