POLITEHNICA UNIVERSITY
OF BUCHAREST

Doctoral School of Electronics, Telecommunications
and Information Technology

Ph.D. THESIS
SUMMARY

Cezar Gabriel DUMITRACHE

CONTRIBUTIONS IN SECURING IoT USING SMART SECURITY SYSTEMS

Conf. Dr. Ing. Constantin Viorel Marian
Politehnica Univ. of Bucharest — PhD Supervisor

BUCHAREST 2025

Chapter 1

Introduction

Communications and information technology have expanded to an unimaginable extent compared to just a few years ago, becoming the driving force of global development. They have experienced this rapid growth because new technologies have emerged, bringing added value through lower production costs and greater energy efficiency. However, even though they were designed by leveraging extensive experience accumulated over time, they also came with a series of vulnerabilities that were only discovered once they were implemented in production. One such example, still widely used today despite its vulnerabilities, is the WPA2 protocol. It is heavily used by all of us, being even "imposed" by internet providers through the devices they give to customers for the duration of their internet services. This has led to an increased demand for specialists in this field, since no type of organization can function with security breaches—ensuring security is essential to prevent unwanted incidents that could result in significant losses.

1.1 Presentation of the field of the doctoral thesis

I chose to approach the concept of cybersecurity by studying the WPA2-Enterprise data encryption protocol in a wireless network, as it brings significant security benefits and can also serve as a foundation for studying the security of the Internet of Things (IoT). These objects have grown exponentially, being introduced into our daily lives in increasing numbers. The vast majority of them have been connected via Wi-Fi in a completely unsecured manner, thus becoming very easy targets for anyone with minimal technical knowledge. Although ENISA reports a decrease in the number of security incidents, it is imperative for organizations to establish specialized security teams, especially since their complexity has increased according to the same study.

1.2 Scope of the doctoral thesis

The purpose of this thesis is to contribute to securing IoT by implementing security systems and policies applicable to most operating systems, as well as by developing a

device designed to secure these objects at the wireless network level, thereby reducing the likelihood of breaches in such networks. At the same time, it aims to provide a practical, scalable, and adaptable solution for various usage scenarios, capable of efficiently preventing and responding to cyberattacks specific to IoT environments, which are found in public, domestic, and industrial settings alike. In this context, I set out to conduct an in-depth analysis of the particularities of cyber threats associated with IoT environments, focusing on frequently encountered attack vectors, their impact on data security, and the functioning of connected devices. The study includes a comparative evaluation of the existing security mechanisms in the main operating systems used in IoT, highlighting the advantages and limitations of each. Furthermore, the thesis proposes an original architecture for securing wireless networks, integrating hardware and software components designed for the protection of the targeted network.

1.3 Content of the doctoral thesis

The doctoral thesis is structured into seven chapters, along with the list of tables, figures, and abbreviations used. In Chapter 1, I presented the introductory part of the thesis domain, where I described various types of attacks, including emerging ones, targeting all existing devices. I studied these types of attacks and defense techniques against them in order to develop different methods and techniques of protection.

In Chapter 2, I described the techniques that can be implemented to protect a network against such attacks at different levels.

Chapter 3 begins with different types of implementations for intrusion detection and continues with methods and actions for protecting various devices running different operating systems.

In Chapter 4, I described encryption and authentication standards and protocols used in wireless environments, as well as protection methods applied in this medium. I also detailed the 802.1X protocol, which I used to secure older IoT devices that are not capable of interacting via this protocol.

Chapter 5 starts with the description of the topology used for device testing, continues with an explanation of device operation, and also presents the conclusions drawn from the implementation of security devices.

Chapter 6 is the most interesting, as it addresses device authentication in a Wi-Fi network through digital certificates and by using blockchain as an additional security mechanism. To my knowledge, this has not yet been implemented elsewhere. Device authentication in a Wi-Fi network using blockchain represents the result of my research, introducing a new concept for securing device authentication with minimal resources.

The final conclusions, results, and my original contributions are presented in Chapter 7.

Chapter 2

Techniques for Mitigating Emerging Types of Attacks

In order to protect the organization's network and, implicitly, its users, measures must be taken in accordance with the current threats. The first actions that need to be implemented are those applied both on network devices and on the devices used by end-users. Among these are the security policies that must be applied both at the software level and at the hardware level.

2.1 At the physical layer

Protecting servers and communication devices by placing them in rooms equipped with locks, motion sensors, thermal sensors, and surveillance cameras. Physical security can be defined as having two main components: the architecture of the building and the types of equipment and devices used.

Protection against the disclosure of essential data can be achieved by positioning monitors in such a way that only the intended user can view the information displayed.

Protecting physical input/output ports so that they can only be accessed by authorized personnel designated to use them.

2.2 At the software level

Various security policies will be implemented both at the network device level and at the end-user device level, with centralized management. Among these, the following can be highlighted:

Creation of VLANs, enabling network micro-segmentation, which reduces the likelihood of compromising the entire network. If a compromise occurs, it will be limited to a single network segment.

Disabling unused protocols on devices, such as DTP.

Creation of administrators with medium-level access for troubleshooting and resolving issues.

Creation of users who will only have access to the resources necessary to perform their tasks and responsibilities.

Policy to lock the device after a period of inactivity, requiring re-entry of the username and password to unlock it.

Temporary lockout after multiple failed login attempts, preventing the user from authenticating for a set period of time.

Centralized time-based access policies, ensuring a user can only use the device during designated working hours. To access the device outside these hours, the user must request permission from the network administrator.

Implementation of daily or weekly backup processes.

Disaster recovery policies, designed to quickly reinitialize the infrastructure in the event of a disaster.

2.3 ARP Attack Mitigation Techniques

For ARP protection, DAI (Dynamic ARP Inspection) can be configured on Layer 2 devices, as well as on Layer 3 devices through enabling DHCP snooping or defining trusted interfaces. However, great care must be taken during implementation, because improper configuration can result in the blocking of legitimate ARP packets. Additionally, the computational load must be considered, as it can affect network performance, even though it provides a very high level of security.

2.4 IP Attack Mitigation Techniques

Considering that communication protocols were created only for remote communication and were not designed to be secure, over time malicious actors have developed various types of attacks to access the information transmitted over these networks. Here are some ways to protect against them:

- IP filtering using firewalls to block unauthorized access;
- Using public names instead of IP addresses;
- Employees should not initiate connections from the host network;
- Securing subdomains;
- IP validation to prevent IP spoofing attacks;
- Use of encryption protocols.

2.5 ICMP Attack Mitigation Techniques

We can protect against this type of attack by enforcing access control lists, thereby blocking traffic for this protocol and allowing its use only where necessary.

2.6 DoS and DDoS Attack Mitigation Techniques

It is recommended that networks, applications, and IT infrastructure be protected through a multi-layered security approach. This methodology can integrate solutions such as firewalls, virtual private networks (VPNs), anti-spam systems, content filters, and other complementary mechanisms. The goal is to ensure continuous monitoring of network activities and the identification of user identity anomalies, which may indicate attempts at DDoS attacks. It is necessary to implement a response plan in the event that such an attack occurs. Another highly effective tool in combating DDoS attacks is the creation of access control lists.

2.7 Spoofing Attack Mitigation Techniques

Such an attack can be countered by implementing preventive measures, such as:

- DHCP Snooping on high-trust ports. DHCP Snooping relies on source MAC addresses and can determine whether they originate from a trusted or less-trusted source, thereby filtering DHCP messages from untrusted sources.
- uRPF (Unicast Reverse Path Forwarding). When a router receives a unicast packet, it normally looks only at the destination to forward it, without verifying the source IP. With this technology, the router also checks the source, and if it is not among the IPs designated as legitimate, the packet is dropped, effectively stopping spoofing attacks.

This approach significantly contributes to preventing spoofing attacks, enhancing the resilience of network infrastructure against compromise or unauthorized access attempts. Additionally, it aligns with modern network security principles, promoting bidirectional filtering and strict control of data flows based on verified identities of communication participants.

Chapter 3

Measures for Preventing Intrusions

3.1 Security Policies for End-User Devices

Recommendations for Workstation Security and Prevention. Workstation security can be achieved by implementing methods that allow control on multiple levels, since the concept of confidentiality is not naturally associated with security for an ordinary user who lacks the necessary skills to take online security measures.

Therefore, it is necessary to implement security policies on workstations, such as:

- Identification – A measure required to identify the workstation user in order to determine the group they belong to, so that the necessary security policies can be applied.
- Authentication – This can be performed through various types of verification: tokens, SIDs, passwords, SIM cards, biometric data verification, and others. For enhanced security, it is recommended to use at least two methods simultaneously.

3.2 Information System Intrusion Prevention Measures

Updating hardware devices and the software in use. By keeping software up to date, especially the operating system, the risk of it being compromised is significantly reduced.

- Each user must ensure that they have installed, configured, and activated an antivirus product on their system. It is recommended that, at regular intervals, a full scan is performed to check for malicious software.
- Disable Telnet and SSH services if they are not used.
- Develop policies that enforce the use of complex passwords by users. A password must:
- Be at least 12 characters long (14+ is even better).
- Include uppercase and lowercase letters, numbers, and special symbols.
- Not be stored in insecure places, but only memorized.

- Use two-factor authentication wherever possible.
- To make complex passwords easier to remember, a mnemonic expression or passphrase can be used.

3.2.1 With WINDOWS operating system

Disable Remote Desktop:
Control Panel \ System and Security \ Allow remote access \ Properties \ Remote Desktop \ Don't allow remote connections to this computer
Password policy:
Policies applied to domain or local accounts that determine their settings, such as enforcement and password lifetime.
Settings for password age.
Settings for the duration of account lockout.
Security policies can be configured using:
GROUP POLICY OBJECT EDITOR:
Computer Configuration \ Windows Settings \ Security Settings \ Account Policies
Create a backup of the operating system.

3.2.2 With a LINUX operating system

Creating users with different privileges depending on each person's responsibilities within the organization.
Changing the default password.
Implementing system auditing and a disaster recovery system.
For remote connections, it is recommended to implement SSH or HTTPS.
Configuring IP Tables (closing unused ports).
Implementing a policy that requires users to change their authentication passwords periodically, with passwords meeting complexity requirements.
Implementing a policy that automatically locks the system after a period of inactivity, requiring the user to enter their username and password to unlock it.
Restricting access rights to the command terminal.

3.2.3 Methods and actions for protecting Layer 2 devices

- The first measure to be taken is physical protection. Devices should be placed so that access is granted only to authorized personnel for maintenance and configuration changes, using locked rooms, motion sensors, surveillance cameras, or none of these measures depending on the importance of the information transmitted through them.
- The second measure is to disable unused ports and assign them to an unused VLAN.
- The next measure involves micro-segmenting the network by implementing VLANs.
- Always use the same VLAN ID dedicated to all ports in a trunk.
- Implement SNMP redundancy for each VLAN individually.

- Implement port security to prevent MAC flooding attacks and DHCP starvation attacks.
- Disable trunking on all unused ports and set them to access mode, assigning them to an unused VLAN in the network.
- Disable automatic trunk negotiation on all ports. Negotiation should be performed manually via configuration.
- Create accounts with limited access for maintenance purposes.
- Limit remote access as much as possible; if remote access is necessary, use secure communication protocols such as SFTP or SSL/TLS and a dedicated management VLAN.
- Micro-segment the network by implementing VLANs.
- Implement DHCP Snooping on VLANs considered vulnerable to DHCP starvation and DHCP spoofing attacks.
- Implement Dynamic ARP Inspection (DAI) to prevent ARP spoofing and ARP poisoning attacks.

3.2.4 Methods and actions for protecting Layer 3 devices

- Disable the HTTP protocol and enable the HTTPS protocol.
- Create users with different privileges based on each person's responsibilities within the organization.
- Depending on the routing protocol used, protect adjacency links between routing devices with passwords. By encrypting EIGRP or OSPF, routers will authenticate each other, preventing false routing updates from unknown sources.
- Implement Access Control Lists (ACLs).
- Disable all unused services.
- Encrypt all passwords.
- Implement IDS (Intrusion Detection System) and IPS (Intrusion Prevention System).
- Implement uPRF (User-Defined Packet Rate Filtering).

3.2.5 Implementation of IDS

At the level of a large organization, it is necessary to create a department dedicated to protection against cybercrime, which must include security analysts. Analysts are often asked to investigate incidents and provide an action plan in case of an attack. To protect against such attacks, a strong and properly configured antivirus must be installed throughout the network, including on end-user devices. Additionally, the network should be protected through next-generation firewalls, intrusion prevention systems, and the inclusion of analysts to monitor and analyze traffic.

This department must be capable of deploying and managing traffic analyzers, firewalls, implementing SIEM, and configuring SNMP.

To detect abnormal behavior of devices within the network, SNMP allows security analysts to review logs generated by devices, providing valuable information for identifying potential threats. Using this data, an analyst can consult various threat intelligence sources to determine the specific family of a potential threat. A platform

that assists in identifying a hash or file by providing relevant information is VirusTotal.com.

3.3 User training

Another very important component for protecting an organization is user training.

To enhance organizational security in the virtual environment, it is necessary to train employees on the proper use of interconnected devices and to avoid attacks such as phishing, vishing, smishing, and typosquatting. This activity is essential to familiarize users with the methods and techniques attackers use to obtain sensitive information. Such information can later be used to compromise system security, either through data theft or by introducing malicious software.

Types of activities that should be carried out include:

Informing and training users to prevent data leaks. Users must be aware of the risks of unintentionally disclosing important organizational information. For example, sending a laptop for repair without removing storage devices.

Encouraging users to treat any email requesting personal data with caution.

Avoiding access to web addresses included in unsolicited emails.

Verifying the legitimacy of emails requesting personal information by contacting the company directly via official phone numbers and checking the sender's domain.

Establishing rules and procedures so that in the event of a disaster, each user follows predetermined steps to minimize damage to the organization.

In addition to technical measures, implementing a clear internal policy regarding the use of personal devices and access to the organization's network is imperative. Applying these policies can prevent the spread of cyber threats via unsecured or compromised devices.

Chapter 4

Protection measures in a Wi-Fi environment

4.1 Standards and protocols in the Wi-Fi environment

To better understand the environment in which I will work, a brief history of the different Wi-Fi standards and their evolution over time is necessary. In Table 4.1, the evolution of Wi-Fi standards from the IEEE 802.11 family is highlighted, with each standard introducing significant improvements in terms of transfer speed, frequency bands, and network efficiency.

The 802.11a standard, developed in 1999, was an early solution for high-speed networks in the 5 GHz band, although its coverage was limited due to the lower signal penetration. In parallel, 802.11b, although slower (11 Mbps), became widely used because of its stability in the 2.4 GHz band. Later, 802.11g combined the advantages of both, offering speeds up to 54 Mbps in the 2.4 GHz band, with good coverage and acceptable performance.

A major leap occurred with 802.11n in 2009, which introduced MIMO technology, increasing theoretical speeds up to 300 Mbps. Development continued with 802.11ac, finalized in 2013, which introduced dual-band operation and the capacity to serve multiple clients simultaneously, achieving gigabit speeds (up to 2.3 Gbps). Currently, 802.11ax (Wi-Fi 6) optimizes performance in crowded networks and reduces power consumption through the Target Wake Time (TWT) feature, while being compatible with all three bands: 2.4 GHz, 5 GHz, and 6 GHz.

Most recently, 802.11be (Wi-Fi 7) builds on Wi-Fi 6, focusing on higher speeds and lower latency for advanced applications such as virtual reality and 8K video streaming. The introduction of Multi-Link Operation allows simultaneous connections across multiple bands, significantly improving network stability and overall performance.

Overall, each generation has brought important innovations, reflecting the growing demands of users for fast, stable, and efficient connectivity.

4.2 Wi-Fi encryption protocols

WPA – WPA is a protocol used to encrypt data in a wireless network. Although it was introduced to address the weaknesses of its predecessor, WEP, over time it has become vulnerable to attacks despite initially improving security.
WPA2 – Similarly, WPA2 is a data encryption protocol for wireless networks, which, like WPA, has become susceptible to attacks over time.
WPA2 Enterprise – This protocol currently provides strong protection, as it authenticates users in a wireless network via a RADIUS or Kerberos server. User identification can also be based on digital certificates, offering higher security.
RADIUS Server – RADIUS (Remote Authentication Dial-In User Service) is a client-server protocol and software that allows a device (controller or access point) to communicate with a central server to authenticate users and authorize access to resources based on requested privileges.
IEEE 802.1X and RADIUS – This is a data encryption protocol that works with a RADIUS server. Its ability to integrate with LDAP is an advantage for organizations, enabling optimized authentication and access control management across the IT infrastructure. It is commonly referred to in the literature as WPA2 Enterprise. Combined with EAP-TLS, it forms the foundation of a highly secure wireless network.

The major advantage of this system is that authentication does not rely on simple passwords but on individually signed certificates issued by a certification authority, significantly reducing the risk of account compromise via brute-force or phishing attacks. In enterprise environments, this architecture is often complemented by role-based access policies, restricting available resources according to user identity and privilege level. Furthermore, integration with existing infrastructures, such as Active Directory, allows for easier management and rapid response in case of incidents.

4.3 Authentication protocols

LEAP – Developed by Cisco, LEAP uses an adapted version of the MS-CHAP protocol. Although it provides authentication, it has vulnerabilities in credential encryption. Therefore, it is recommended to implement strong passwords to compensate for inherent security weaknesses.

EAP-FAST – This protocol uses two phases for authentication. In the first phase, a TLS handshake establishes a protected tunnel for secure key exchange between the client and the authentication server. Once the tunnel is successfully established, the second phase verifies the credentials of the user or device. Client and server also enforce authorization and authentication policies. EAP-FAST supports a wide range of configurations, including whether certificates are used.

PEAP – PEAP encapsulates EAP within a TLS-encrypted and authenticated tunnel. The process involves two steps:

- Phase One: Establish a secure encrypted channel.

- Phase Two: Perform the actual authentication.

EAP-TLS – Internationally recognized as a very effective data security technique, EAP-TLS provides mutual authentication. Both the client and server must have certificates for authentication. Once verified, session keys are created and used for the session. This protocol is particularly effective against Man-in-the-Middle attacks and provides clear visibility of who accesses the network at any given time, since authentication relies on embedded digital certificates.

EAP-TTLS – Unlike EAP-TLS, EAP-TTLS uses two phases of authentication. The first phase uses standard TLS procedures, followed by a client-authentication phase over a tunnel using EAP, PAP, or MS-CHAP.

EAP-TTLS/PAP – This hybrid authentication method combines transport security with a simple authentication mechanism. However, passwords are transmitted in clear text, making them vulnerable to interception. Configuration is complex, requiring advanced knowledge, and relying on end-user devices for setup can lead to delays and other issues.

EAP-TTLS/MSCHAP – This PPP-based protocol uses CHAP negotiation. The access server challenges the client, which responds with authentication packets ("Response" and "Success") to enable mutual authentication. The client also requests verification from the server to ensure it knows the user's password. If the server fails to respond, the client automatically terminates the connection.

4.4 Methods for Wi-Fi protection

- Change the access password at regular, short intervals.
- Disable public broadcasting of the network name (SSID).
- Implement MAC address filtering.
- Implement the 802.1X authentication protocol.
- Use digital certificates.
- Use TLS (Transport Layer Security).
- Limit signal strength to reduce network visibility in unauthorized areas, minimizing the risk of interception.
- Segment the network using VLANs to isolate sensitive devices and prevent lateral access in case of a breach.
- Monitor traffic in real time to detect suspicious activity or intrusions quickly.
- Update router and access point firmware to close known vulnerabilities and increase resistance to exploits.
- Disable unnecessary services (e.g., UPnP, WPS) to eliminate potential weak points in device configuration.
- Enforce strict password policies to make brute-force attacks more difficult.
- Implement two-factor authentication (2FA) for an additional layer of security even if passwords are compromised.
- Isolate clients in a wireless network to prevent direct communication between devices.
- Log authentication and access events.

- Conduct periodic security testing through vulnerability scans.
- Block suspicious IP addresses.
- Backup device configurations to ensure quick recovery in case of failure or attack, which is a very good disaster recovery practice.
- To ensure Wi-Fi network security, it is essential to adopt multiple complementary measures. First and foremost, regularly changing the access password prevents unauthorized access if the password is compromised, thereby limiting the time an attacker can exploit the network.

Chapter 5

Wireless Architecture for IoT Security

5.1 Proposed Architecture

Within this project, a security system for IoT devices was implemented using two Raspberry Pi boards. The first board hosts a FreeRADIUS server, responsible for authenticating users connected to the Wi-Fi network. The second board is dedicated to connecting wireless IoT devices that do not support the 802.1X protocol.To analyze the performance of the established connections, the Iperf3 application was used. An important aspect of device-level security is the use of a bridge between the wireless and Ethernet interfaces, allowing Layer 2 packet forwarding. In this way, communication with the secured wireless network for any IoT device can be carried out through the Ethernet interface.

5.2 Experimental Setup

To establish secure communication, the following experimental setup was implemented: two Raspberry Pi development boards, an Asus router, and an IoT device. In this setup, a laptop was used to simulate an IoT device.

This configuration allows testing a real-world scenario commonly encountered in industrial or smart home environments, where certain devices cannot be modified to support advanced authentication protocols but still need to be integrated into a secure network.

The first step involved installing an operating system on the Raspberry Pi board used to run the FreeRADIUS server, specifically a Linux distribution for Raspberry Pi to enable the WPA2 Enterprise protocol.

In the proposed architecture, the two Raspberry Pi boards play a central role. Essential components include a RADIUS server, an authenticator (wireless controller or access point), a packet-bridging device, and clients (IoT devices).

For a client to connect to a wireless network and initiate a connection, it must be capable of using the 802.1X protocol. The first step was installing FreeRADIUS. The second step involved entering the IP address of the authenticator (Access Point or

Wi-Fi controller) and the password used for its authentication. At this stage, a Wi-Fi network secured via WPA2-Enterprise or the 802.1X protocol was created.

For simplicity during testing, a secure connection based only on username and password was used initially, without digital certificates. Once this connection is functional, digital certificates can be created and implemented to further enhance connection security.

To secure connections for older devices that normally cannot use the 802.1X protocol, the following solution was implemented:

Another Raspberry Pi 4 board was used to connect an IoT device that has an Ethernet interface. Any device added to the topology is connected via a patch cord from the Raspberry Pi's Ethernet port to the network port of the respective device.

5.3 Conclusions

Through this type of architecture, a secure connection was established between the IoT device (laptop) and the secured wireless network. I consider this implementation to be very useful for interconnecting different devices securely using the 802.1X protocol. It is also worth noting that this architecture achieves very low energy consumption.

To further enhance security, the SSID will be disabled in the Access Point configuration panel, and digital certificates will be installed on both the server and the clients connecting to the Wi-Fi network. This makes the network nearly impossible to penetrate.

As a final conclusion, although several studies discuss the concept of a bridge between Wi-Fi and Ethernet, none of them implement it with a secured wireless network using 802.1X, as proposed in this project.

Chapter 6

Wi-Fi Authentication System Using Blockchain

6.1 Wi-Fi Authentication

From the vast field of cybersecurity, I chose to focus on wireless network security. To enhance security, a wide range of tools is available and accessible, considering the complexity of this domain.

When discussing network security, the technical aspect often receives the most attention, but physical security should not be overlooked, as it frequently plays a crucial role. Additionally, administrative measures are important for assigning roles and defining access rights, ensuring each user can only access the resources needed to perform their tasks.

This was achieved using blockchain technology, which represents an innovation in the field. Currently, this type of authentication in Wi-Fi networks is not yet widely used. By combining blockchain-based authentication with digital certificates, nearly impenetrable security can be achieved. The implementation of smart contracts automates authentication and authorization, reducing the risk of unauthorized access.

For the experiment, a topology was created consisting of: a Raspberry Pi board running FreeRADIUS and the Ganache blockchain, connected via Ethernet to an Asus router serving as the authenticator. The hashes generated from digital certificates were stored on the blockchain, allowing identity verification without relying on a centralized database. This eliminates a critical vulnerability point and increases the network's resilience against spoofing or identity forgery attacks.

Moreover, using FreeRADIUS ensures compatibility with standard authentication protocols, facilitating integration with existing infrastructure. The use of smart contracts allows automatic enforcement of access policies without manual intervention, significantly reducing human errors and increasing the efficiency of access control.

6.2 Creation of Digital Certificates

As IoT networks, critical infrastructures, and industrial systems become increasingly interconnected, it is essential that each device is securely identified and authenticated. To prevent unauthorized access, current solutions implement the TLS protocol along with digital certificates issued by a trusted entity, known as a Certification Authority (CA).

A Certification Authority is a trusted institution that verifies the identity of various entities and issues them digital certificates. By signing these certificates with its private key, the CA ensures the authenticity of the entity's identity. This method provides a high level of trust in device-to-device communication, guaranteeing that transmitted data comes from verified sources.

The main functions of a CA with respect to connected devices are:

- Verify the identity of the manufacturer or device.
- Issue a customized digital certificate.
- Enable validation of the device's identity within a system.
- A digital certificate is an electronic document that links a public key to the identity of a device. It is signed by a CA and contains:

- Information about the device.
- The device's public key.
- Information about the issuing CA.
- Validity period.
- The CA's digital signature.
- Authentication is performed through a process called mutual TLS authentication, in which both the server and the device prove their identities using certificates signed by a trusted CA.

Preparation process:

The device generates a key pair: one private and one public key.

For the experiment, FreeRADIUS and the Ganache blockchain were installed on a Raspberry Pi board to store the hashes generated from the created digital certificates.

A Certification Authority was created to sign the digital certificates. The first step was setting up the CA using OpenSSL.

6.3 Determination of EAP Usage

To use authentication via the TLS protocol, I modified the "authorize" section within the sites-enabled file used by FreeRADIUS. I added a check_blockchain section, which initializes the verification of hashes stored in the local Ganache blockchain.

After verification via a Python script, if the response returns any value other than 0, the authenticator receives a "reject" message. If the response is 0, the authenticator receives an "access accepted" message.

6.4 Python Script Implementation

In the first phase, the script reads the certificate, then calculates the hash of the certificate. It then adds the prefix 0x in front of the calculated hash to match the format already stored. Next, the script connects to the local blockchain and compares the calculated hash with the one stored in the blockchain.

If the comparison returns 0, the certificate matches one of the previously stored certificates. If not, it returns 1, indicating that the certificate does not match.

6.5 Generating Hashes from Digital Certificates

To generate hashes from the digital certificates created for clients, the following command was used:

- openssl x509 -in cez.crt -outform DER | sha256sum | awk '{print $1}'

Thus, these hashes were created manually by entering the command in the console. The command generates a hash using the OpenSSL utility for the existing certificates.

The first part of the command, -in cez.crt, specifies the input certificate file in PEM format (most commonly Base64 encoded with -----BEGIN CERTIFICATE-----).

The parameter -outform DER converts the certificate from PEM format to binary DER format, which is widely used for generating hashes.

The command sha256sum calculates the digital fingerprint of the certificate in DER format.

Finally, awk '{print $1}' extracts only the hash value, ignoring the filename.

The resulting hash is then stored on the blockchain for later verification. This process ensures a precise and consistent hash of the certificate content, necessary for reliable authentication and validation.

6.5 Storing Hashes in Ganache

Using Remix Ethereum, the previously generated hashes were stored in Ganache. Remix is a useful tool for Ethereum application development, as it automatically generates multiple blockchain accounts with preset test Ether balances. This Ether has no real value, allowing tests to be conducted without cost.

Interaction with this local simulated network is extremely fast, and every transaction and event can be monitored in real time. This enables developers to test the functionality of smart contracts in a controlled and predictable environment.

In my project, this local network was used to simulate the storage and verification of hashes generated from digital certificates. This process ensures that the identity of each client can be securely validated without exposing sensitive data.

With the help of Remix Ethereum, I wrote a smart contract in which the hashes generated from digital certificates were stored. During authentication, a Python script verifies whether the hash generated from the digital certificate presented by the device matches the hash stored on the blockchain. This adds an extra layer of security to FreeRADIUS authentication.

Chapter 7

Conclusions

7.1 Obtained Results

The results obtained from integrating these technologies to secure IoT devices in a Wi-Fi network are:

High security through the use of digital certificates and blockchain technology, which significantly increases resistance to attacks.

Very low energy consumption due to the use of Raspberry Pi boards.

Cost reduction by implementing smart contracts, which automate the authentication process, reduce human intervention, and consequently lower operational costs..

7.2 Original Contributions

In this work, I designed and implemented a secure network topology characterized by minimal energy consumption, low implementation costs, and modest hardware requirements. This was achieved through the use of Raspberry Pi boards in combination with open-source software applications, which do not require purchasing commercial licenses.

Additionally, the authentication process was extended and customized using Python and Solidity. These languages were used to develop functional modules that enable both logical control of authentication and integration with a smart contract deployed on the blockchain.

The innovative element of the proposed architecture lies in reinforcing the security mechanism by adding an additional layer of protection during device authentication in the network. Specifically, fingerprints (hashes) of user identities were generated and stored in a smart contract on the blockchain. Verification of these fingerprints is performed automatically via a custom Python script integrated into the FreeRADIUS authorization flow.

This mechanism ensures decentralized, verifiable, and tamper-resistant authentication, significantly enhancing the trust and security level of the network.

7.3 List of Original Works

During the entire doctoral period, ten articles were published, six of which as first author..

.

[1] C.-G. Dumitrache, C. V. Marian, G. Predusca, F.M. Barbu, M. Neferu, "Wireless Authentication System for Internet of Things using FreeRADIUS and Blockchain ," in 2025.

[2] C.-G. Dumitrache and P. Anghelescu, "Design and Implementation of a Secure Communication Architecture for IoT Devices," JSAN, vol. 14, no. 4, p. 64, Jun. 2025, doi: 10.3390/jsan14040064.

[3] C.-G. Dumitrache, G. Gheorghe, P. Gabriel, and G. Mariuca-Roxana, "Device For Securing IoT In The Wireless Environment," in 2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Iasi, Romania: IEEE, Jun. 2024, pp. 1–6. doi: 10.1109/ecai61503.2024.10607458.

[4] C.-G. Dumitrache, C. A. Ghita, G. Predusca, L. D. Circiumarescu, N. Angelescu, and D. C. Puchianu, "The Efficiency of Routing Protocols, OSPF and EIGRP, Using Graphical Network Simulator-3 and Python," in 2024 28th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania: IEEE, Oct. 2024, pp. 194–199. doi: 10.1109/icstcc62912.2024.10744665.

[5] C.-G. Dumitrache, G. Predusca, G. Gavriloaia, N. Angelescu, D. Circiumarescu, and D. C. Puchianu, "Comparative analysis of routing protocols using GNS3, Wireshark and IPerf3," in 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, Romania: IEEE, Jun. 2022, pp. 1–6. doi: 10.1109/ecai54874.2022.9847499.

[6] C. G. Dumitrache, G. Predusca, L. D. Circiumarescu, N. Angelescu, and D. C. Puchianu, "Comparative study of RIP, OSPF and EIGRP protocols using Cisco Packet Tracer," in 2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE), Galati: IEEE, Oct. 2017. doi: 10.1109/iseee.2017.8170694.

[7] A. Frâncu, G. Preduşcă, L. D. Cîrciumărescu, N. Angelescu, D. C. Puchianu, and C.G Dumitrache, "Examining IP Network Routing Protocols with Packet Tracer and Cisco Equipment," The Scientific Bulletin of Electrical Engineering Faculty, vol. 25, no. 1, pp. 68–73, Jun. 2025, doi: 10.2478/sbeef-2025-0011.

[8] S. Cristian, P. Gabriel, C. Denisa, A. Nicoleta, P. Dan, and C.-G. Dumitrache, "Analysis of web traffic using MPLS technology," in 2022 International Symposium

on Electronics and Telecommunications (ISETC), Timisoara, Romania: IEEE, Nov. 2022, pp. 1–4. doi: 10.1109/isetc56213.2022.10009987.

[9] A. Frâncu, G. Predușcă, L. D. Cîrciumărescu, N. Angelescu, D. C. Puchianu, and C.-G. Dumitrache, "Examining IP Network Routing Protocols with Packet Tracer and Cisco Equipment," The Scientific Bulletin of Electrical Engineering Faculty, vol. 25, no. 1, pp. 68–73, Jun. 2025, doi: 10.2478/sbeef-2025-0011.

[10] G. Predușcă, D. Circiumarescu, C.-G. Dumitrache, and C. Stănescu, "Analysis of a Fiber Optical Link Using Octave," The Scientific Bulletin of Electrical Engineering Faculty, vol. 22, no. 1, pp. 1–6, Jul. 2022, doi: 10.2478/sbeef-2022-0007.

[11] C. Stănescu, G. Predușcă, D. Circiumarescu, and C.-G. Dumitrache, "Determination of Distance in Fiber Optical Communications Networks," The Scientific Bulletin of Electrical Engineering Faculty, vol. 22, no. 1, pp. 1–6, Jul. 2022, doi: 10.2478/sbeef-2022-0012.

7.4 Perspectives for further developments

A possible extension of the functionalities proposed in this work involves the implementation of an additional security mechanism based on unique hardware identifiers. Specifically, a supplementary hash could be generated from the MAC address of the device requesting network access. This hash could be stored on the blockchain, alongside other authentication information, thereby further strengthening the access control architecture.

By including this physical identification element, an extra verification layer is added, making it significantly more difficult, if not impossible, for an unauthorized device to gain access to the network, even in scenarios where authentication credentials are compromised. The proposed mechanism would enhance not only the overall security of the system but also its resilience against spoofing or man-in-the-middle attacks.