OPTIMIZING IOT SECURITY VIA TPM INTEGRATION: AN ENERGY EFFICIENCY CASE STUDY FOR NODE AUTHENTICATION

Anestis Papakotoulas[1], Theodoros Mylonas[2], Kakia Panagidi[1], Stathes Hadjiefthymiades[1]
[1]Department of Informatics & Telecommunications, National & Kapodistrian University of Athens, Greece, [2]Department of Science & Technology, Hellenic Open University, Greece

NOTE: Corresponding author: Anestis Papakotoulas, apapakot@di.uoa.gr

Abstract – *The widespread adoption of Internet of Things (IoT) applications in different technical fields has resulted in a significant increase in connected devices while amplifying concerns regarding security and privacy. The presence of security vulnerabilities in various layers of IoT design has emerged as an important issue. Trusted computing, particularly leveraging the Trusted Platform Module (TPM), is seen as a promising approach to counter these vulnerabilities. This paper investigates thoroughly the utilization of TPM technology to enhance node authentication with a focus on energy efficiency. Researchers closely examine each layer to carefully outline an adversary model that is tailored to the IoT ecosystem. The node authentication scheme that is proposed leverages TPM, which has advantages both in terms of processing time and energy. The outcome of this study can be applied to Flying AdHoc Network (FANET) nodes that operate in areas with high levels of traffic, where there are strict safety and reliability standards. Experiments conducted present the essential significance of TPM in ensuring secure node authentication across various application environments. The adoption of TPM technology is validated through rigorous performance assessments, revealing significant improvements in both energy efficiency and security.*

Keywords – Cryptography, energy efficiency, Internet of Things, node authentication, trusted platform module 2.0, wireless sensor networks

1. INTRODUCTION

The IoT technology is becoming more and more important and vastly adopted nowadays. The associated environment features great heterogeneity in aspects like hardware, networking and sensing. The majority of devices are lightweight nodes that operate in low power and have limited computing and memory resources. Their main function is efficient data collection, initial processing and transmission further up in the IoT architectural hierarchy. Due to the diversity of IoT devices and their multitude, security is of paramount importance. Relevant solutions that are robust and resource-optimized need to be developed and applied to sustain performance and operational efficiency.

Although incorporating embedded devices into a network provides more effective control and maintenance, it also increases the possibility of security issues. Hence, the establishment of a reliable connection, supported by robust security services, becomes essential in order to ensure the provision of reliable services across the aforementioned technological domains and to support the development of a trustworthy IoT ecosystem accessible to all stakeholders. Trust is an essential element of security that can be acquired or bestowed, but should never be presumed. Numerous applications are constructed with server-side and client-side components that are beyond the user's control. The Trusted Computing Group (TCG) is developing and promoting the emerging technology of trusted computing [1]. The TCG is an industry consortium that develops standards based on trusted computing approaches [2]. One of TCG's standards is the implementation of TPM.

According to the ISO/IEC 11889 standard, TPM is a security module that is based on hardware. The device functions as a specialized microcontroller that is specifically designed to enhance the security of a device's hardware and software by incorporating cryptographic keys and algorithms. This facilitates the secure storage of passwords, certificates, encryption keys, and authentication algorithms. In addition, TPMs provide a variety of security functionalities, including secure boot, storage capabilities, and key management, thereby establishing a resilient framework for the protection of confidential information. Due to their ability to function across various operating systems and their extensive adoption, TPMs have become an essential solution for enhancing the dependability of IoT applications. TPMs have a crucial role in guaranteeing the integrity of computing devices in various environments by enabling the storage of essential metrics. TPM facilitates the implementation of secure bootstrapping mechanisms, enabling remote entities to verify the integrity of their systems by ensuring that only authorized code is executed [3].

FANETs have significantly transformed road traffic management in a vibrant urban environment. In this dynamic network, autonomous drones serve as nodes, equipped with a variety of sensors such as cameras, Light Detection And Ranging (LiDAR), and GPS. These Unmanned Aerial

Vehicles (UAVs) engage in real-time collaboration, fulfilling various functions such as monitoring traffic, responding to emergencies, planning dynamic routes, monitoring air quality, and serving as security surveillance systems. Nevertheless, the primary focus is on node authentication. Every individual drone is subjected to rigorous authentication procedures prior to integration into the network, thereby guaranteeing that only devices with established trustworthiness are able to engage in communication and make contributions to the traffic management system.

Examining security concepts pertaining to a typical FANET, wherein a Roadside Unit (RSU) acts as the resource controller, could be valuable. The primary duties associated with the role of a resource controller encompass the supervision of nodes and the management of sensors. This architectural framework contains UAVs, referred to as "nodes," which establish a connection with the RSU responsible for each node's authentication. Fig. 1 depicts a node in its comprehensive representation, comprising multiple sensors responsible for capturing environmental data. Within this particular node, a multitude of applications are currently operational, engaging in interactions with various other applications within the cloud computing environment. The server and other nodes establish communication with each other through a wireless interface. In order to enhance the level of security at the node, a TPM could be incorporated. The TPM serves as a root of trust for each node, enhancing its security as a whole.

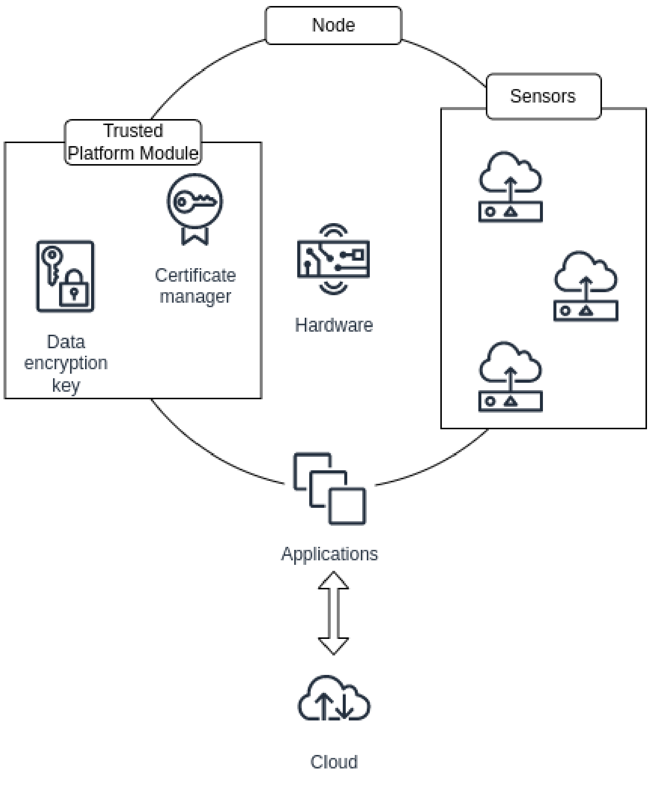

Fig. 1 – IoT architecture in trusted computing

Each application and sensor may utilize the TPM, which operates cryptographic algorithms to encrypt its data. TPM could be either an embedded device or software. A trusted component could also be installed on the RSU. A node authentication process could be used in this FANET scenario to serve as a fundamental element of security, effectively protecting against possible cyber threats and ensuring the overall integrity of the system.

1.1 Contribution

In this paper, we present a comprehensive analysis of the use of TPM technology to improve node authentication and provide an energy-efficient solution. Focusing on each layer, we present a detailed adversary model for the IoT ecosystem. Next, a node authentication scheme employing TPM and CPU and performing time comparisons is described. Our application consists of FANET nodes that function in high-traffic areas. Enhanced safety and dependability requirements are characteristic of road traffic areas. We demonstrate how TPM can be employed wherever secure node authentication is required. Additionally, we evaluate the effectiveness of our measurements. Our evaluation results indicate that using TPM technology significantly improves energy efficiency and security. Finally, we discuss the limitations of our methodology and directions for future research.

This study's contribution is considered significant because it aims to achieve the following goals:

- Development of a library that ensures operational node authentication.
- Demonstration of the use of TPM and CPU to secure an IoT device.
- Quantification of the time required to complete the node authentication process when using TPM or CPU.
- Quantification of the energy efficiency of the node authentication process in comparison between TPM and CPU.
- Contribution to the field of IoT security and the application of trusted computing to the security of FANETs.

1.2 Structure of the paper

The paper is organized as follows: In Section 2, we present related work. A description of the TPM adversary model is presented in Section 3, which defines the main concepts and components studied in this work. Section 4 explains the proposed implementation's system architecture. Using TPM, Section 5 describes a hardware-based approach for node authentication on an IoT device. The sixth section provides experimental results and evaluates the performance of the experiments. Section 7 concludes the paper and discusses directions for future research.

2. RELATED WORK

Security is regarded as one of the primary concerns that impede the rapid and widespread adoption and implementation of Internet of Things (IoT) and Wireless Sensor Networks (WSNs). In fact, security in IoT and WSN is a popular topic of discussion [4], [5]. The paper [6] gives an overview of the most important security issues in WSN, with a focus on the security vulnerabilities in the different protocol layers.

Several authentication systems have been developed to protect the security of the IoT, as authentication is the cornerstone of delivering effective security [7], [8], [9].

To accommodate weak connections and changeable network topologies, [10] suggested a Task-oriented Authentication Model (ToAM) for UAV-based networks that utilized blockchain technology and Public Key Infrastructure (PKI). In ToAM, UAVs were implanted with TPM chips for secure data storage, and support the Transport Layer Security (TLS) handshake to prevent common attacks against software-only implementations. In addition, the authentication mechanism invoked two distinct phases for group building authentication and intra-group authentication, with hash values saved in the blockchain representing authentication data. Authors in [10] present a secure and globally operable UAV authentication system based on dependable security mechanisms and established protocols; nevertheless, this method does not guarantee that UAVs will connect to a trustworthy environment.

Researchers in [11] present the cybersecurity challenges faced in IoT environments. A literature comparative study based on trusted computing schemes is presented, along with different implementations of critical analysis. The implementations are based on an algorithmic approach with no experimental results (like in the present paper). The TPM used is based on TPM 1.2 and not on TPM 2.0.

In [12], the One Drone One Block (ODOB) distributed network architecture was introduced, which depends on a blockchain-based framework with its nodes as independent as possible from each other. Compared to standard blockchain implementations, this framework aims to protect the network, minimize the compute and communication overhead of maintaining the blockchain, and reduce delay and storage needs. In addition, the drones were paired with individual blocks and fitted with TPM chips for secure identification, although these were not utilized when the algorithmic technique was implemented in real-world situations.

According to the authors of [13], there is no adequate solution in terms of security services for the various security mechanisms based on trusted IoT computing (confidentiality, integrity, authentication, and availability). The authors suggest the incorporation of TPM into IoT computing in order to perform cryptographic operations and provide hardware-based security; however, there is no evidence of experimentation or implementation.

In [14], the authors describe how to conduct a self-identification procedure to achieve secure auto-configuration of IoT devices entering the cloud. In [15], the authors analyze the security concerns associated with the collection of IoT data and highlight three threats: i) the process involves IoT devices; ii) data exchange between devices and other parties, such as users and clouds; iii) security concerns associated with the configuration chosen by users on their devices.

In [16], the authors propose and evaluate a remote attestation protocol for IoT networks with different schemes for both TPM-enabled IoT devices and IoT devices that could not run TPMs. The authors apply their model MTRA to two single-board computers: the Odroid-XU4 for TPM-enabled devices and the Raspberry Pi for non-TPM devices. The authors present their results applied to a different area of node authenticity. The experiments used TPM 1.2 emulator (contrary to the present paper which is based on TPM 2.0). The attacks that were studied are a subset of the attacks covered in Table 6. They are the man-in-the-middle attack, the wormhole attack, the Time-Of-Check-To-Time-Of-Use (TOCTTOU) Attack, and the verifier-based DoS attack.

TPM functionality applied on a TPM 1.2 emulator is presented in [17] applied on energy-constrained resources. The base TPM functionality is sliced and distributed across multiple IoT devices within a cluster. Authors implemented a distributed TPM with 4 and 10 shares and illustrated that for each IoT device there is a 90%+ energy advantage to this approach measured in energy units (Joule). However, the authors do not indicate the type of the resources and the sliced approach can be implemented with at least four nodes in a cluster raising restrictions for the possible underlying infrastructure.

In [18], the authors propose a TPM extension scheme (xTSeH) to ensure system integrity and authentication for SEDs lacking TPM chips, demonstrating its feasibility through an RPi. However, the measurements are performed with a hardware TPM 1.2 (Infineon SLB9645) and not on TPM 2.0, where no arguments have been made for energy efficiency or key generation.

Despite the extensive research on TPM benefits and its adaptation, an evaluation of the security, performance, and usability of both types of TPM in an IoT environment with the same resources is lacking.

3. ADVERSARY MODEL - TPM AS COUNTERMEASURE

IoT is used in a wide variety of services, applications, and products, necessitating the need for a dependable, secure, and interoperable ecosystem. Due to rapid development, some pathogens have been created, which, combined with the security issues faced by distributed systems, do not leave the IoT unscathed and free from security and privacy challenges. Such security challenges include privacy, authentication, management and information storage issues. The subsequent paragraphs will clarify the different techniques through which the layers of the IoT architecture, as shown in Fig. 2, can be subjected to attacks.

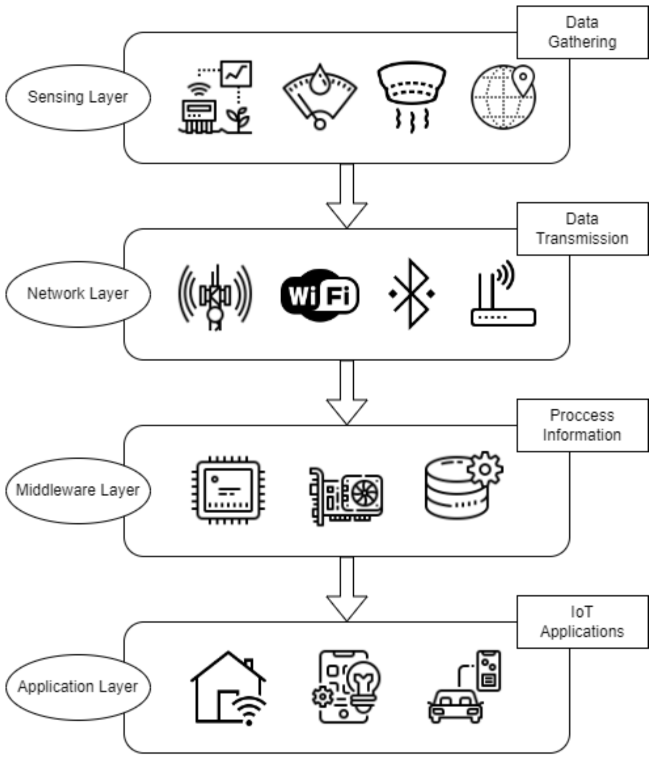

Fig. 2 – IoT layers

3.1 Sensor-level security issues

At this layer, various IoT elements (sensors) are deployed throughout the environment and collect data. At this level, the following attacks are possible [19], [20], [21]:

Node capturing: Many sensors in the IoT are connected to low-power nodes. These nodes can be attacked in a number of ways by malicious users who want to intercept data being sent from the node or even pretending to be the node themselves.

Malicious code injection attack: In this kind of attack, the attacker injects malicious code into the memory of the node. Taking advantage of the nodes being remotely updated, the malicious user may compel the nodes to execute undesirable actions or even obtain complete access to the IoT system.

False data injection attack: Once an attacker has gained control of the node, it may be exploited to inject bogus data into the IoT system. The attacker may continue with a Distributed Denial-of-Service (DDoS) assault using this strategy.

Side-Channel Attack (SCA): In addition to direct assaults on the nodes, channel attacks may also result in the disclosure of sensitive information. Processor design, electromagnetic radiation, and power usage may divulge confidential communication data.

Eavesdropping and interference: IoT applications sometimes include a large number of nodes spread over a vast region, making them vulnerable and susceptible to eavesdropping.

Booting attacks: In order to maintain their batteries for as long as possible, the nodes function in sleep/wake-up cycles, where they are restarted during the transition from the sleep phase to the wake-up phase. During the boot phase, remote endpoints are susceptible to a variety of assaults as security functions are disabled.

3.2 Network-level security issues

The basic function of the network layer is to transmit the information received from the sensing layer to the computing unit for processing. Below are the main security issues encountered at the network level [19], [20], [21], [22], [23]:

Phishing site attack: Phishing attacks are often reported when multiple IoT devices can be targeted with minimal effort by the attacker, leading users to deceptive websites and potentially gaining access to user passwords. In this case, the entire IoT environment becomes vulnerable to cyberattacks.

Access attack: An IoT network may become vulnerable to access attacks, also known as Advanced Persistent Threats (APTs), which occur when unauthorised individuals gain access. It is essential to note, however, that although some access attacks may meet the criteria for APTs, not all access attacks fall into this classification. The attacker can remain undetected on the network for an extended period of time. Typically, these types of assaults are more concerned with getting data and information than with harming IoT devices.

Data transit attacks: IoT applications necessitate the storage and interchange of data. The value of data makes it a constant target for hackers. The security of data kept on local servers or in the cloud is at danger, but data in transit, or traveling between locations, is much more susceptible to intrusions. There is a great deal of data traffic in IoT applications between sensors, actuators, the cloud, etc. Due to the diversity of connectivity methods used to transmit data, IoT applications are susceptible to data breaches.

Routing attacks: In this form of attack, hostile IoT nodes may attempt to divert data routing. Sink attacks are a form of routing attack in which a malicious node provides the shortest routing path and induces other nodes to direct traffic to the sink through it. A worm-hole assault is another technique that, when combined with other attacks such as a sinkhole attack, can pose a significant security risk. A worm hole is an out-of-band link between two nodes that facilitates quick packet transport. An attacker can establish a worm hole between a network node and a device to attempt to circumvent the fundamental security procedures of an IoT application.

Spoofing and Sybil attacks: Spoofing and Sybil attacks primarily seek to detect RFID and MAC addresses in order to get unauthorized access to the IoT system. As the TCP/IP suite lacks a robust security protocol, IoT devices are rendered more susceptible to spoofing attacks. These two types of attacks, along with man-in-the-middle and denial-of-service attacks, are considered among the most severe.

3.3 Middleware security issues

Although the middleware layer is useful for providing a reliable and robust IoT application, it is also susceptible to various attacks. These attacks can take control of the entire IoT application by infecting the middleware. Database security and cloud security are also security challenges at the middleware level. Various possible attacks in the level are shown below [19], [20], [21], [22]:

Man-in-the-middle attack: The Message Queuing Telemetry Transport (MQTT) protocol employs a publish-subscribe form of communication between clients and subscribers through the MQTT mediator, which operates as a proxy. This helps to decouple publishers from client subscribers, allowing messages to be transmitted without knowing the destination. If the attacker is able to gain control of the intermediary and become the man-in-the-middle, he can assume complete control of all communications without the customers' awareness.

SQL injection attack: Middleware may be the target of SQL Injection (SQLi) attacks. In these assaults, the attacker can implant malicious SQL queries into an application to obtain the private information of any user and modify database files.

Signature wrapping attack: In a signature refactoring attack, an attacker breaks the signature algorithm and uses vulnerabilities in the simple object access protocol to perform operations on or modify the intercepted message.

Cloud malware injection: An attacker can take control, inject malicious code, or inject a virtual machine into the cloud via cloud malware injection. Using a malicious virtual machine or malicious module, the attacker impersonates a legitimate service. The attacker can then get access to device services and obtain or alter sensitive data.

3.4 Application-level security issues

The security issues between the application and task layers overlap, as specific security issues exist at these layers that are not presented at other layers, such as privacy issues. Security issues at these levels are specialized for different applications. Many IoT applications also consist of a secondary layer between the network layer and the application layer, commonly referred to as the application support layer or middleware layer, as described above. Attacks that can be performed at these levels are given below [20], [21], [22], [23]:

Access control attacks: Access control is the authorization method that grants authorized individuals or processes access to the data or account. In IoT applications, an access control attack is crucial because when secure access is breached, the entire IoT program becomes susceptible to attack.

Malicious code injection attacks: Typically, attackers employ the simplest means available to break into a system or network. If the system is susceptible to malicious scripts and misconfigurations due to insufficient code controls, then this would be the initial entry point used by an attacker. Cross-site scripting (XSS) is typically used by attackers to inject malicious code into a trusted website. If an XSS attack is successful, an IoT account can be compromised, and the IoT system cannot function.

Sniffing attacks: Attackers may use sniffer applications to monitor network traffic in IoT applications. This can make it possible for an attacker to get to private and confidential user data if there aren't enough security measures in place to stop it.

Reprogram attacks: If the programming process is not secured, unauthorized people will be able to remotely reprogram IoT objects. This can result in IoT network hacking.

3.5 TPM as countermeasure

TPM is a cryptocontroller that ensures the authenticity and integrity of embedded systems and devices, performs remote attestation, and performs cryptographic tasks. It provides a secure communication channel between smart factories to prevent theft of data, processes, and intellectual property. Key characteristics include encrypted key, certificate, and password storage, dedicated key management, and support for a variety of encryption methods, such as the Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithms.

The AES algorithm, as a replacement for the Data Encryption Standard (DES) encryption algorithm, can support multiple block sizes, but AES chooses to have a fixed block size of 128 bits with three key variations of 128 bits, 192 bits, and 256 bits. The following are the main characteristics of AES [24], [25]:

- Symmetric encryption key;
- 128-bit information, which uses a 128, 192, or 256-bit key;
- It has stronger encryption than its predecessor;
- It is faster than its predecessor.

The RSA encryption algorithm is an asymmetric encryption algorithm as it uses different keys for encryption and decryption without the requirement of sending secret keys while also having the ability to digitally sign [26], [27].

3.5.1 Sensor-level security

The TPM is essential for enhancing the security of IoT ecosystems, particularly at the sensor level. Its robust cryptographic capabilities and secure storage offer a formidable defence against a wide variety of security threats. TPM can counter node capturing by preventing unauthorised entities from gaining control over IoT nodes. Validating the integrity of software running on devices mitigates malicious code injection attacks. In addition, TPM protects against false data injection attacks by allowing the validation of data authenticity via cryptographic signatures. In addition, it offers formidable defence against side-channel attacks, preventing adversaries from exploiting information leakage. With its encryption capabilities, TPM protects against eavesdropping and interference, ensuring that all communications remain secure and private. Lastly, TPM's secure boot process thwarts booting attacks, ensuring that only authentic, unmodified firmware is executed. In essence, the TPM is a formidable defence against a variety of sensor-level security threats, playing a crucial role in bolstering the integrity and reliability of IoT deployments.

3.5.2 Network-level security

While the TPM operates primarily at the device level, it has a significant impact on network-level security. TPM plays a crucial role in preventing phishing site attacks by establishing only authenticated and trusted connections. It protects against access attacks by providing secure authentication mechanisms, thereby preventing unauthorised parties from gaining network access. The cryptographic functions of the TPM are essential for protecting against data transit attacks, encrypting data in transit to prevent interception or modification. In addition, it protects against routing attacks by ensuring that network traffic follows authorised paths, thwarting any attempts to manipulate or redirect data flows. TPM plays a crucial role in preventing spoofing and Sybil attacks; it ensures the authenticity of network devices and prevents the creation of fraudulent identities. While the TPM is primarily a device-level security measure, it significantly contributes to enhancing network-level security, providing a robust defence against a variety of sophisticated threats.

3.5.3 Middleware security

The TPM is crucial to strengthening the security of middleware. It closes key security holes in this portion of IoT systems. It proves invaluable in thwarting man-in-the-middle attacks by facilitating secure communication channels and ensuring that data exchanges remain private and impervious to interception or manipulation. In addition, TPM's cryptographic capabilities offer a robust defence against SQL injection attacks, preventing malicious attempts to manipulate or compromise databases via injected commands. TPM ensures the integrity of digital signatures and prevents adversaries from tampering with authentication credentials, playing a crucial role in preventing signature wrapping attacks. In addition, TPM's secure boot process and secure storage capabilities play an essential role in preventing cloud malware injection by ensuring that only trusted and unmodified software components are executed. The TPM is, in essence, a stalwart defender of middleware security, providing essential protections against a spectrum of sophisticated attacks in this fundamental layer of IoT ecosystems.

3.5.4 Application-level security

In IoT ecosystems, the TPM is essential for bolstering application-level security. It plays a crucial role in preventing access control attacks by enforcing strong authentication and authorization mechanisms, ensuring that only authorised users and processes have access to vital resources. The secure storage capabilities of the TPM offer a formidable defence against malicious code injection attacks, preventing unauthorised code injections or modifications to applications. In addition, the TPM's encryption and secure communication features effectively defend against sniffing attacks, ensuring that sensitive data remains private during transit. In addition, the TPM's integrity measurement capabilities are essential for preventing reprogram attacks, ensuring that only authorised, unmodified applications are executed. In essence, the TPM serves as a stalwart defender of application-level security, providing essential protections against an array of sophisticated attacks, thereby enhancing the overall integrity and dependability of IoT applications.

4. SYSTEM ARCHITECTURE

For the purpose of this research, the Raspberry Pi (RPi) 3B, a computing device commonly used in IoT applications, was selected to mimic a node in a FANET[28]. FANET nodes need to be lightweight devices like RPi in order to be easily integrated into a unmanned vehicle managing system's primary tasks (autopilot, sensing) along with other, lateral operations. RPis are commonly used, as shown in [28] and [29], as they offer a powerful operating system with several functionalities in a lightweight package.

The utilization of the RPi 3 Model B board in IoT environments encompasses a wide array of applications. In the context of smart home applications, the RPi 3 Model B functions as a central hub, facilitating the interconnection of diverse devices such as thermostats, lighting systems, and security cameras. The high computational capacity of this device facilitates the smooth integration and automation of various functions, empowering users to remotely monitor and regulate their domestic surroundings. The RPi 3 Model B is commonly used in industrial environments for the purpose of process automation. It serves as a gateway device, facilitating the collection of data from various sensors and effectively coordinating the operation of machinery. The efficiency of data exchange across different devices and systems is facilitated by the compatibility of the system with various communication protocols. Furthermore, within educational environments, the RPi 3 Model B is a highly valuable resource for instructing programming and electronics, enabling students to develop IoT solutions and gain comprehension of the complexities associated with interconnected systems. The RPi 3 Model B has demonstrated its versatility and capability as a platform for constructing various IoT environments, including smart homes, industrial automation, and educational settings.

The device in question can be described as a 64-bit miniature personal computer, functioning with the computational capabilities provided by a quad-core 1.2 GHz Broadcom BCM2837 processor. The 40-pin port serves as a means of communication with various environmental sensors, actuators, and interface devices, employing protocols such as SPI and I2C. This port facilitates the TPM's connectivity. The RPi commonly employs different iterations of the Linux operating system, although it is also capable of running Windows 10 IoT Core.

For the purpose of the node authentication scheme, the Infineon IRIDIUM SLB 9670 TPM2.0 board will be used, on which the Infineon OPTIGA™ SLB 9670 TPM 2.0 circuit is installed. The OPTIGA™ TPM SLB 9670 microcontroller follows the specifications of the TCG 2.0 family (Fig. 3), is compatible with the RPi Model 3 B, and offers a set of security functions such as key management, authentication, signature functions,encryption/decryption, and secure recording [30] [31].

5. IMPLEMENTATION

In FANETs, mobility, lack of central control, and the extremely volatile topology of the nodes are the main features. Assuming they are acting to perform road traffic checks using sensing systems. All this transmitted data is very sensitive; therefore, it must be fully protected.

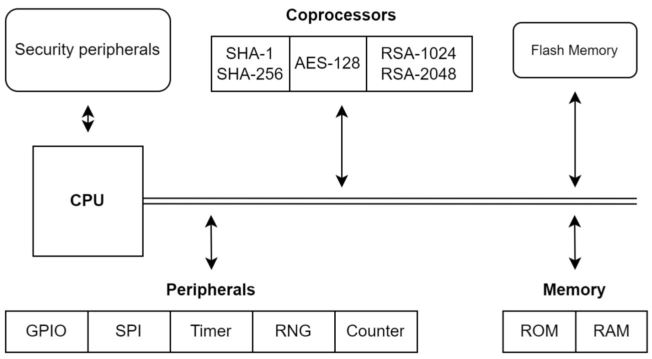

Fig. 3 – Architecture diagram of OPTIGA™ TPM SLI 9670 [30]

In order to achieve integrity, authenticity, and confidentiality, the Secure Hash Algorithm (SHA), the AES symmetric key algorithm, and the RSA asymmetric key algorithm, combined with hardware TPM, will be used. The TPM can perform encryption and decryption operations in addition to digital signature verification.

The utilization of the previously mentioned encryption algorithms in combination with a TPM guarantees the verification of the operating node's authentication. In this manner, the duration required to accomplish the task can be assessed, along with the amount of energy consumed. In addition to the utilization of the CPU for the purpose of authentication, it is possible to assess and compare their respective performance levels.

The novelty of this research is that it proposes an energy-efficient method for authenticating a node in a distributed network, such as FANET, by utilizing the capabilities of TPM 2.0.

5.1 Node authentication

In order to determine the most efficient encryption and decryption method in terms of time and the consumption of IoT resources (memory, CPU usage, etc.) for the generated information, both the performance of delegating the encryption and decryption processes exclusively to the processor (CPU) and the system performance of delegating the encryption and decryption processes to TPM 2.0 were evaluated.

Fig. 4 depicts the client's data exchange with the server, as well as encryption and decryption using TPM 2.0.

5.1.1 Client-side procedures

Create public and private key: Create the first RSA key for the initial exchange of encrypted authentication data.

Encrypt client's public key with server's key: Create the client's public RSA key with the server's public key using the RSA algorithm, and store the key in the TPM, if TPM is used.

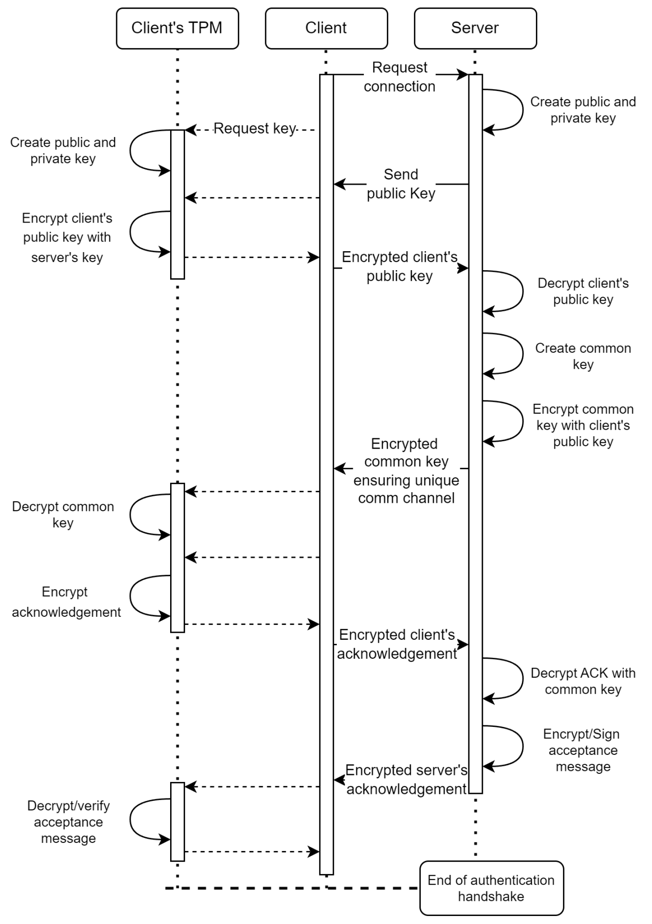
Client's TPM
Client
Server
Request connection
Create public and private key
Request key
Create public and private key
Send public Key
Encrypt client's public key with server's key
Encrypted client's public key
Decrypt client's public key
Create common key
Encrypt common key with client's public key
Encrypted common key ensuring unique comm channel
Decrypt common key
Encrypt acknowledgement
Encrypted client's acknowledgement
Decrypt ACK with common key
Encrypt/Sign acceptance message
Encrypted server's acknowledgement
Decrypt/verify acceptance message
End of authentication handshake

Fig. 4 – Data flow of node authentication mechanism.

Decrypt common key with client's private key: Decrypt common communication key with the client's private key using the RSA algorithm and store the key in the TPM.

Encrypt acknowledgement: Encrypt the acknowledgement of receiving successfully the common communication key using the selected encryption algorithm.

Decrypt/verify acceptance message: Decrypt the message received from the server in the case of the AES algorithm with the shared key, or to confirm the digital signature of the server's acceptance message with the shared public in the case of the RSA algorithm.

5.1.2 Server-side procedures

Create public and private key: Generate the first RSA key for the authentication's initial encrypted data exchange and store the key in the TPM, if one is used.

Decrypt client's public key: Decrypt the public key sent from the client to the server using the RSA algorithm and store the key in the TPM.

Create common communication key: Generate the common communication key depending on the selected encryption algorithm for the initial exchange of encrypted authentication data and store the key in the TPM.

Encrypt common key with client's public key: Encrypt the common key with the key received from the client by the server using the RSA algorithm.

Decrypt acknowledgement with common key: Decrypt the message of successful reception of the common key received on the server from the client, according to the selected encryption algorithm.

Encrypt/Sign acceptance message: Encrypt the client's acceptance message to the server when using the AES algorithm, or sign it when using the RSA algorithm.

5.2 Assumptions

According to implementation assumptions, IPv4 is employed. Prior to the authentication process, the time offset between the server and client clocks was determined. The client sends an unencrypted packet of data to the server along with its settings (TPM support, selected encryption algorithm, key size, and maximum encryption packet size) at the beginning of the communication, and the server sends an identical packet to the client. The application uses the method (CPU or TPM) for which it is configured for encryption or decryption, regardless of the settings of the other, i.e. if the client is configured to encrypt with the CPU, it will not change the method if the server uses TPM, while the server will decrypt with TPM. Multiple clients can connect to the server simultaneously. In order to obtain a representative sample of encryption and decryption measurements, each cryptographic method was executed for thirty repetitions for a minimum of eight minutes.

The client and server applications were written in the Java programming language, with the Java libraries as well as the Microsoft TPM Software Stack (TSS) libraries [32] for TPM 2.0 control. The Operating System and Hardware Information (OSHI) library version 4.4.2 [33] was used to record system characteristics and resources (CPU used, memory used, network traffic etc.). Furthermore, to support the above libraries, there is the Simple Logging Facade for Java (SLF4J) version 1.7 [34], the Java Native Access (JNA and JNA-platform) version 5.5 [35], the bcprov-jdk15on version 1.64, and Apache Commons Lang common-lang3-3.9 [36].

6. EXPERIMENTAL ANALYSIS

6.1 Experimental results

The experimental part was conducted with two approaches: one with the RPi as the client and a PC as the server, and one with the RPi as the server and a PC as the client. The PC used to send data to the RPi, either as a client or a server, has an i7 processor of the seventh generation, eight cores, and a frequency of 2.8 GHz, which is significantly higher than the RPi's 1.2 GHz. As a result, it completes the tasks assigned to it more quickly.

Tables 1, 2, and 3 provide the average, max and min times recorded for client and server applications during the authentication process regarding the selected encryption algorithm.

All presented measurements are recorded for the RPi. It is mentioned that the server generates keys twice, which is the most time-intensive operation. As the PC generates the two required keys in a shorter amount of time, the time required to complete the authentication process when the RPi is acting as a client should be less than when the RPi is acting as a server.

Regarding the data packet size, the maximum amount of RSA-1024 information that can be encrypted is 117 bytes, using the remaining 11 bytes used for the header. The RSA-2048 algorithm can encrypt 245 bytes of data, plus 11 bytes for the header. Larger amounts of data are divided into packets of 117 bytes or 245 bytes or less, respectively.

For each cryptographic algorithm, the average, maximum, and minimum CPU and memory consumption values were also recorded. These metrics are presented in tables 4 and 5.

6.2 Node authentication outcomes

According to the node authentication process, it is observed that there is no overhead on the response times of the encryption algorithms (key generation, encryption, and decryption) between the two applications (client and server). At the end of the authentication process using the RSA algorithm, the process of digital signature and signature confirmation is performed. Digital signatures take longer to process than for encryption. On the contrary, signature confirmation times show a different pattern with decryption between CPU usage and TPM, where CPU usage shows confirmation times much shorter than decryption (the percentage varies from 10% in RSA-2048 to 50% in RSA-1024), while using TPM, signature confirmation times are much longer, almost double, than for decryption. In any case, both digitally signing a message and checking the signature using the TPM takes longer than using the CPU. From tables 1 and 2, where the total times to achieve authentication are presented, it becomes clear that the authentication takes place in a shorter time using the CPU as opposed to the TPM, where there are observed long delays in all encryption algorithms, both on the client side and on the server side. Comparing the total execution time of the encryption algorithms, AES-128 is the fastest, followed by RSA-1024 and RSA-2048 that use the CPU, and then RSA-1024 and RSA-2048 that use the TPM. It becomes clear that in each case (CPU and TPM), RSA-1024 authenticates faster than RSA-2048.

Notably, by performing the key generation using the TPM at a trusted geolocation before deploying the node into FANET, the overall authentication time is comparable to or even better than for CPU. Figures 5 and 6 present the aforementioned assumption by comparing the total authentication time with (blue line) or without (red line) the pre-establishment of the keys on TPM for each cryptographic algorithm.

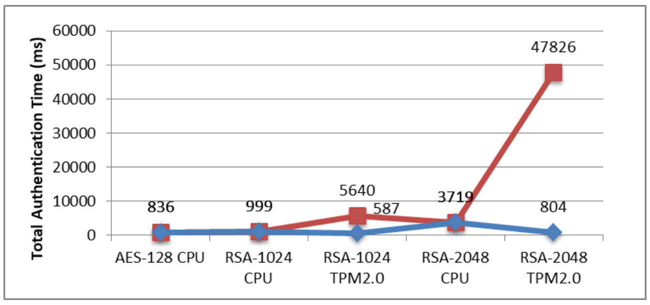

Fig. 5 – Total client authentication time

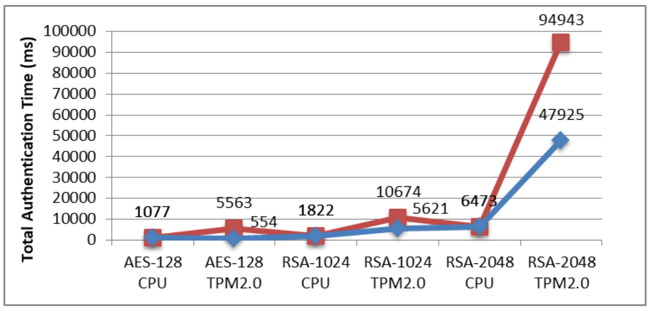

Fig. 6 – Total server authentication time

Overall, the authentication process occurs more quickly using the CPU compared to the TPM. However, by generating keys using the TPM before deploying the node into FANET, the authentication time can be made comparable to, or even better than, using CPU-based node authentication.

6.3 Resources' consumption outcomes

In relation to the utilization of resources during the authentication process, it is evident that the execution of encryption algorithms by the CPU in the client-side application necessitates approximately twice and nearly five times the processing power of the RPi when compared to the execution of these algorithms by the TPM.

Upon analysis of the encryption algorithms implemented through the CPU or TPM, it is clear that RSA-2048 demands a lower computational capacity compared to RSA-1024. Specifically, when considering TPM, RSA-2048 demands nearly half the processing power of RSA-1024.

The AES-128 algorithm necessitates a greater amount of processing power in comparison to the alternative algorithms employed within the client application. Moreover, there are no discernible disparities in memory usage between encryption algorithms that employ the CPU or TPM.

Table 1 – Client node authentication measurements

Procedure [Algorithm] (ms)	AES-128 CPU			RSA-1024 CPU			RSA-1024 TPM2.0*	RSA-2048 CPU			RSA-2048 TPM2.0*
	Min	*Max*	*Avg.*	*Min*	*Max*	*Avg.*	*Avg.*	*Min*	*Max*	*Avg.*	*Avg.*
Create Public & Private key [RSA]	507	1207	815	599	1382	918	5047	1710	5386	3516	47022
Encrypt Client's public key with Server's key [RSA]	4	14	7	1	4	6	136	4	9	7	148
Decrypt common key with Client's private key [RSA]	1	5	3	43	74	62	308	164	201	181	504
Encrypt acknowledgement [AES/RSA]	6	14	10	2	5	4	70	4	12	7	72
Decrypt/Verify acceptance message [AES/RSA]	1	2	1	8	9	9	79	8	8	8	80

*Measurements using TPM2.0 are quite similar, therefore no max/min values are provided.

Table 2 – Server node authentication measurements using RSA for communication

Procedure (ms)	RSA-1024 CPU			RSA-1024 TPM2.0*	RSA-2048 CPU			RSA-2048 TPM2.0*
	Min	*Max*	*Avg.*	*Avg.*	*Min*	*Max*	*Avg.*	*Avg.*
Create Public & Private key	345	1316	887	5053	1574	4934	3303	47018
Decrypt Client's public key	84	106	96	311	261	314	282	503
Create common communication key	480	1114	817	4920	1843	3771	2802	46894
Encrypt communication key with Client's public key	2	3	2	182	6	7	7	186
Decrypt acknowledgement with common key	1	2	2	152	2	3	3	250
Sign acceptance message	1	30	18	29	74	81	76	92

*Measurements using TPM2.0 are quite similar, therefore no max/min values are provided.

Table 3 – Server node authentication measurements using AES for communication

Procedure [Algorithm] (ms)	AES-128 CPU			AES-128 TPM2.0*
	Min	*Max*	*Avg.*	*Avg.*
Create Public & Private key [RSA-1024]	333	1500	968	5009
Decrypt Client's public key [RSA-1024]	83	106	92	299
Create common communication key [AES]	1	2	1	176
Encrypt communication key with Client's public key [RSA-1024]	3	6	4	67
Decrypt acknowledgement with common key [AES]	9	13	11	-
Encrypt acceptance message [AES]	1	2	1	-

*Measurements using TPM2.0 are quite similar, therefore no max/min values are provided. The device cannot support encryption/decryption processes in AES mode (manufacture's note).

Table 4 – CPU utilization during node authentication

Algorithm	Avg.	Min.	Max.
AES-128 CPU	44.72%	4.13%	71.57%
RSA-1024 CPU	43.72%	3.17%	73.52%
RSA-1024 TPM	21.67%	3.18%	63.29%
RSA-2048 CPU	37.43%	5.26%	73.07%
RSA-2048 TPM	8.44%	3.60%	53.23%

Table 5 – Memory utilization during node authentication

Algorithm	Avg.	Min.	Max.
AES-128 CPU	28.19%	27.13%	29.19%
RSA-1024 CPU	27.92%	26.84%	28.63%
RSA-1024 TPM	28.71%	27.86%	30.15%
RSA-2048 CPU	28.48%	27.29%	29.16%
RSA-2048 TPM	28.29%	26.42%	30.47%

In the context of memory utilization, there is a lack of variability in the observed data, with measurements exhibiting fluctuations that consistently hover around 30 percent.

To summarize, the authentication process shows different resource usage patterns. CPU-based execution requires much more processing power than TPM for encryption algorithms. Particularly, RSA-2048 has lower computational requirements compared to RSA-1024.

6.4 Addressed security challenges

Section 3 of the research delved into an examination of the security issues and various forms of attacks that may be encountered within an IoT ecosystem. Additionally, the study highlighted the role of the TPM as a countermeasure against these threats.

The TPM chip has a dedicated tamper-resistant storage area, which protects data even if the computer is physically compromised or attacked by malware. Any attempt to modify or tamper with the register values will result in the TPM detecting the change and triggering a system response. This capability consistently maximizes the benefits of TPM over software-based TPM.

Table 6 presents a comparison between TPM and software-based approaches in terms of the types of attacks that can be mitigated within an IoT ecosystem.

Table 6 – CPU and TPM security comparison in IoT attacks

Layers	***Attack Types***	***CPU usage***	***TPM usage***
Sensor	Node Capturing	✓	✓
	False Data Injection Attack	✗	✓
	Side-Channel Attacks	✓	✓
	Eavesdropping and Interference	✓	✓
Network	Access Attack	✗	✓
	Data Transit Attack	✓	✓
	Spoofing and Sybil attack	✓	✓
Middleware	Man-in-the-Middle Attack	✓	✓
	Signature Wrapping Attack	✓	✓
Application	Data Thefts	✓	✓
	Access Control Attack	✗	✓
	Sniffing Attack	✓	✓

The TPM offers a hardware-based and secure framework for IoT devices, providing strong defense mechanisms against various forms of attacks. Even though software-based methods are very important, they depend on the security of the software infrastructure they are built on and may be more vulnerable to certain types of attacks, especially those that involve physical tampering.

In conclusion, the importance of the TPM in enhancing the security of IoT ecosystems is evident. It is achieved through its specialized tamper-resistant storage and hardware-based defence mechanisms, which are more effective than software-based methods. Software-based approaches rely on the security of the underlying software infrastructure and are more vulnerable to physical tampering attacks.

6.5 Energy efficiency analysis

The power consumption of an RPi 3B during idle periods, with CPU usage at approximately 10%, is estimated to be approximately 2.5 watts. Under conditions of moderate usage, specifically when the CPU is operating at a level of 40-60% utilization, it is expected that power consumption will exhibit a discernible increase, potentially falling within the range of 3-3.5 watts.

Considering our measurements, CPU utilization during the node authentication process is using TPM ranges from 8% to 22%, whereas during the node authentication process using CPU, it ranges from 35% to 45%. Based on the aforementioned data, it is clear that there exists a minimum difference of 0.5 watts in energy consumption.

According to [37], the TPM functions at a voltage of 3.3 volts, exhibiting a deviation of 0.3 volts. The TPM demonstrates an impressive energy-efficient design in its inactive state, exhibiting a low standby power consumption of approximately 110 µA. This feature guarantees minimal power depletion during periods of inactivity. Hence, the power consumption of TPM is extremely low, measuring at approximately 0.363 mW. However, during the transitioning into active mode, the TPM consumes a maximum of 25 mA. Based on the aforementioned data, it is determined that the energy consumption amounts to 0.0825 watts.

Based on the aforementioned computations, we argue that the utilization of TPM for node authentication results in a decrease in power consumption of approximately 25% compared to CPU-based node authentication. Consequently, the extended battery life will facilitate the execution of longer-duration missions for each FANET node.

7. CONCLUSIONS

By implementing the node authentication procedure, we are able to evaluate the operational trustworthiness of an IoT ecosystem that has constraints in terms of available resources. The node authentication scheme employed in our system is both simple and efficient. It leverages the AES and RSA cryptographic algorithms to ensure robust security measures. Additionally, it successfully delivers the desired operational status information.

It is possible that the implementation of TPM-based security measures may introduce system performance degradation due to increased communication requirements. This is primarily attributed to the relatively slower clock speed of the TPM compared to the CPU, resulting in reduced efficiency and slower execution of functions. Therefore, a crucial challenge that needs to be addressed is the enhancement of the clock-speed capacity of a TPM in order to execute the associated cryptographic operations at a faster rate. Nevertheless, it is imperative to carefully consider these concerns in light of the benefits provided by TPM, such as enhanced system security, integrity, and confidentiality. One of the primary benefits of utilizing a hardware-based TPM is its inherent ability to provide a tamper-resistant storage capability.

Based on the aforementioned capability of being tamper-resistant, it can be assumed that prior to deploying nodes in the FANET, key creation procedures have been performed to mitigate the time delay. Consequently, it is expected that the total duration of execution will be faster in comparison to that of the CPU.

Aggregating the results, a preliminary assessment suggests that the most optimal algorithm, considering the mean values of our measurements, is AES-128 implemented on the CPU. However, what is remarkable is that on CPU usage, the upper and lower bounds of measurements demonstrate huge variance. In contrast, cryptographic processes with TPM show a deterministic approach. The main reason is that the CPU must handle the entire system load, which causes processing delays. Concerning the variance of the measurements, the most efficient algorithm is RSA-1024 using TPM. Regarding CPU consumption, the TPM has a significantly lower demand for system resources during authentication. Therefore, as mentioned in Section 6.5, the power consumption during node authentication using TPM is optimized by reducing it by 25%.

Essentially, our study of node authentication in FANET using TPM in comparison to software-based approaches indicates that:

- The time required to complete the node authentication procedure is considered acceptable based on the enhanced capabilities of the TPM.
- No modifications have been made to the system during its operational duration.
- The encryption keys are securely protected and stored within the TPM, ensuring that they cannot be intercepted or compromised.
- The secure environment of the TPM allows for the encryption of data.
- Execution variances are not observed during the processes of encryption and decryption.
- The information that is being transmitted undergoes encryption.
- The authentication process utilizes less computational resources compared to other software-based methodologies, thereby enhancing its lifecycle.

Going a step further, this work can be extended to more complex scenarios, in which the resources involved form a dynamic topology. Let's assume an emergency scenario, like a fire in a warehouse. Unmanned ground vehicles are responsible for tracking humans in buildings and guiding them outside of them. The unmanned vehicles are equipped with TPM devices that will be dynamically registered in the warehouse security ecosystem and start to exchange encoded messages. The challenges in these dynamic environments are considered to be: i) quickly initiate (cold start) an encrypted communication as an "external" device in a secure ecosystem; ii) to secure the integrity of the rest of the infrastructure; and iii) to be energy efficient, especially when using battery-based devices like unmanned vehicles.

Undoubtedly, the implementation of a FANET in a road-traffic environment necessitates significant effort and the resolution of various challenges. In this research, we present a comprehensive analysis of security aspects and experimental performance metrics using TPM and CPU, which could be valuable for individuals seeking to deploy a secure FANET. The precision of time in the road-traffic environment is a requirement that is fulfilled by TPM, surpassing the CPU. The measurements indicate that node authentication using TPM varies slightly, with the upper and lower bounds being pretty close. Another crucial factor to consider is the management of resource consumption, wherein it is evident that TPM shows a significantly less energy consumption compared to CPU utilization.

Given that hardware-based TPM is widely regarded as the most secure type of TPM, it is imperative to enhance its performance in order to meet the growing need for enhanced security. Hence, our forthcoming research agenda encompasses the examination of energy efficiency for other encryption algorithms, such as Elliptic Curve Cryptography (ECC), as well as the utilization of TPM in additional aspects of IoT security including, but not limited to, securing communication and localization.

ACKNOWLEDGEMENT

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] Will Arthur, David Challener, and Kenneth Goldman. "Using the New Trusted Platform Module in the New Age of Security". In: *A Practical Guide to TPM 2.0*. Springer Nature, 2015, pp. 285–288. DOI: 10.1007/978-1-4302-6584-9.

[2] *Trusted Computing*. https://trustedcomputinggroup.org/trusted-computing/. [Online; accessed 23 4 2022]. 2020.

[3] Ronald Toegl, Thomas Winkler, Mohammad Nauman, Theodore W. Hong, Johannes Winter, and Michael Gissing. "Programming Interfaces for the TPM". In: *Trusted Computing for Embedded Systems*. Ed. by Bernard Candaele, Dimitrios Soudris, and Iraklis Anagnostopoulos. Cham: Springer International Publishing, 2015, pp. 3–32. ISBN: 978-3-319-09420-5. DOI: 10.1007/978-3-319-09420-5_1. URL: https://doi.org/10.1007/978-3-319-09420-5%5C_1.

[4] Pradeep Singh, Bharat Bhargava, Marcin Paprzycki, Narottam Kaushal, and Wei-Chiang Hong. *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*. Jan. 2020. ISBN: 978-3-030-40304-1. DOI: 10.1007/978-3-030-40305-8.

[5] Palvi Aggarwal, Cleotilde Gonzalez, and Varun Dutt. "HackIt: A Real-Time Simulation Tool for Studying Real-World Cyberattacks in the Laboratory". In: *Handbook of Computer Networks and Cyber Security, Principles and Paradigms*. Ed. by Brij B. Gupta, Gregorio Martínez Pérez, Dharma P. Agrawal, and Deepak Gupta. Springer, 2020, pp. 949–959. DOI: 10.1007/978-3-030-22277-2_39. URL: https://doi.org/10.1007/978-3-030-22277-2%5C_39.

[6] Bharat Bhushan and Gadadhar Sahoo. "Recent Advances in Attacks, Technical Challenges, Vulnerabilities and Their Countermeasures in Wireless Sensor Networks". In: *Wirel. Pers. Commun.* 98.2 (Jan. 2018), pp. 2037–2077. ISSN: 0929-6212. DOI: 10.1007/s11277-017-4962-0. URL: https://doi.org/10.1007/s11277-017-4962-0.

[7] Luciano Barreto, Antonio Celesti, Massimo Villari, Maria Fazio, and Antonio Puliafito. "An authentication model for IoT clouds". In: *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 2015, pp. 1032–1035. DOI: 10.1145/2808797.2809361.

[8] Alireza Esfahani, Georgios Mantas, Rainer Matischek, Firooz B. Saghezchi, Jonathan Rodriguez, Ani Bicaku, Silia Maksuti, Markus G. Tauber, Christoph Schmittner, and Joaquim Bastos. "A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment". In: *IEEE Internet of Things Journal* 6.1 (2019), pp. 288–296. DOI: 10.1109/JIOT.2017.2737630.

[9] S. Rajalakshmi E. Padma. "Trusted Attestation System for Cloud Computing Environment Using Trusted Platform Module". In: *Internet of Things and Cloud Computing Journal* 5.1 (2017), pp. 38–43. DOI: 10.11648/j.iotcc.20170503.11.

[10] Dominic Pirker, Thomas Fischer, Christian Lesjak, and Christian Steger. "Global and Secured UAV Authentication System based on Hardware-Security". In: *2020 8th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. 2020, pp. 84–89. DOI: 10.1109/MobileCloud48802.2020.00020.

[11] Mohammad Faisal, Ikram Ali, Muhammad Khan, Su Kim, and Hong Kim. "Establishment of Trust in Internet of Things by Integrating Trusted Platform Module: To Counter Cybersecurity Challenges". In: *Complexity* 2020 (Dec. 2020), p. 9. DOI: 10.1155/2020/6612919.

[12] Maninderpal Singh, Gagangeet Singh Aujla, and Rasmeet Singh Bali. "ODOB: One Drone One Block-based Lightweight Blockchain Architecture for Internet of Drones". In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2020, pp. 249–254. DOI: 10.1109/INFOCOMWKSHPS50562.2020.9162950.

[13] Mohammad Faisal, Ikram Ali, Muhammad Khan, Su Kim, and Hong Kim. "Establishment of Trust in Internet of Things by Integrating Trusted Platform Module: To Counter Cybersecurity Challenges". In: *Complexity* 2020 (Dec. 2020), p. 9. DOI: 10.1155/2020/6612919.

[14] Antonio Puliafito, Antonio Celesti, Massimo Villari, and Maria Fazio. "Towards the Integration between IoT and Cloud Computing: An Approach for the Secure Self-Configuration of Embedded Devices". In: 2015 (Jan. 2015). ISSN: 1550-1329. DOI: 10.1155/2015/286860. URL: https://doi.org/10.1155/2015/286860.

[15] Vineeta Soni, Samriddhi Koolwal, Sahiti Balantrapu, Devershi Bhatt, and Narendra Yadav. "Security Requirements in Internet of Things: Challenges and Methods". In: Oct. 2021, pp. 600–604. DOI: 10.1109/ISPCC53510.2021.9609355.

[16] Hailun Tan, Gene Tsudik, and Sanjay Jha. "MTRA: Multi-Tier randomized remote attestation in IoT networks". In: *Computers Security* 81 (2019), pp. 78–93. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2018.10.008. URL: https://www.sciencedirect.com/science/article/pii/S0167404818307697.

[17] Hala Hamadeh, Soma Chaudhuri, and Akhilesh Tyagi. "Area, Energy, and Time Assessment for a Distributed TPM for Distributed Trust in IoT Clusters". In: *2015 IEEE International Symposium on Nanoelectronic and Information Systems*. 2015, pp. 225–230. DOI: 10.1109/iNIS.2015.17.

[18] Di Lu, Ruidong Han, Yulong Shen, Xuewen Dong, Jianfeng Ma, Xiaojiang Du, and Mohsen Guizani. "xTSeH: A Trusted Platform Module Sharing Scheme Towards Smart IoT-eHealth Devices". In: *IEEE Journal on Selected Areas in Communications* 39.2 (2021), pp. 370–383. DOI: 10.1109/JSAC.2020.3020658.

[19] Syeda Tahsien, Hadis Karimipour, and P. Spachos. "Machine learning based solutions for security of Internet of Things (IoT): A survey". In: *Journal of Network and Computer Applications* 161 (Apr. 2020), p. 102630. DOI: 10.1016/j.jnca.2020.102630.

[20] Rashid Hussain and Irfan Abdullah. "Review of Different Encryption and Decryption Techniques Used for Security and Privacy of IoT in Different Applications". In: *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. 2018, pp. 293–297. DOI: 10.1109/SEGE.2018.8499430.

[21] Nataliia Neshenko, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". In: *IEEE Communications Surveys Tutorials* 21.3 (2019), pp. 2702–2733. DOI: 10.1109/COMST.2019.2910750.

[22] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures". In: *IEEE Access* 7 (2019), pp. 82721–82743. DOI: 10.1109/ACCESS.2019.2924045.

[23] Asmaa Munshi, Nouf Ayadh Alqarni, and Nadia Abdullah Almalki. "DDOS Attack on IOT Devices". In: *2020 3rd International Conference on Computer Applications Information Security (ICCAIS)*. 2020, pp. 1–5. DOI: 10.1109/ICCAIS48893.2020.9096818.

[24] Sana Fatima, Tanazzah Rehman, Muskan Fatima, Shahmeer Khan, and Mir Arshan Ali. "Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing". In: *Engineering Proceedings* 20.1 (2022). ISSN: 2673-4591. DOI: 10.3390/engproc2022020014. URL: https://www.mdpi.com/2673-4591/20/1/14.

[25] Zhenghong Jiang, Hanchen Jin, G. Edward Suh, and Zhiru Zhang. "Designing Secure Cryptographic Accelerators with Information Flow Enforcement: A Case Study on AES". In: *Proceedings of the 56th Annual Design Automation Conference 2019*. DAC '19. Las Vegas, NV, USA: Association for Computing Machinery, 2019. ISBN: 9781450367257. DOI: 10.1145/3316781.3317798. URL: https://doi.org/10.1145/3316781.3317798.

[26] Muhammad Hafiz Mazlisham, Syed Farid Syed Adnan, Mohd Anuar Mat Isa, Zahari Mahad, and Muhammad Asyraf Asbullah. "Analysis of Rabin-P and RSA-OAEP Encryption Scheme on Microprocessor Platform". In: *2020 IEEE 10th Symposium on Computer Applications Industrial Electronics (ISCAIE)*. 2020, pp. 292–296. DOI: 10.1109/ISCAIE47305.2020.9108811.

[27] Purnima Gupta, Deepak Kumar Verma, and Aswani Kumar Singh. "Improving RSA Algorithm Using Multi-Threading Model for Outsourced Data Security in Cloud Storage". In: *2018 8th International Conference on Cloud Computing, Data Science Engineering (Confluence)*. 2018, pp. 14–15. DOI: 10.1109/CONFLUENCE.2018.8442788.

[28] Inam Ullah Khan, Ijaz Mansoor Qureshi, Muhammad Adnan Aziz, Tanweer Ahmad Cheema, and Syed Bilal Hussain Shah. "Smart IoT Control-Based Nature Inspired Energy Efficient Routing Protocol for Flying Ad Hoc Network (FANET)". In: *IEEE Access* 8 (2020), pp. 56371–56378. DOI: 10.1109/ACCESS.2020.2981531.

[29] Inam Khan, Asrin Abdollahi, Abdul Jamil, Bisma Baig, Muhammad Aziz, and Fazal Subhan. "A Novel Design of FANET Routing Protocol Aided 5G Communication Using IoT". In: *Journal of Mobile Multimedia* (Apr. 2022). DOI: 10.13052/jmm1550-4646.1851.

[30] *OPTIGATM TPM SLI9670*. https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-tpm/sli-9670/. [Online; accessed 23 4 2022]. 2020.

[31] *IRIDIUM SLI 9670 TPM2.0*. Available : https://www.infineon.com/cms/en/product/evaluation-boards/iridium-sli-9670-pm2.0/. [Online; accessed 23 4 2022]. 2020.

[32] Microsoft. *The TPM Software Stack from Microsoft Research*. https://github.com/microsoft/TSS.MSR/. [Online; accessed 23 4 2022]. 2020.

[33] *OSHI*. https://github.com/oshi/oshi/. [Online; accessed 23 4 2022]. 2020.

[34] *Simple Logging Facade for Java*. http://www.slf4j.org//. [Online; accessed 23 4 2022]. 2020.

[35] *Java Native Access (JNA)*. https://github.com/java-native-access/jna/. [Online; accessed 23 4 2022]. 2020.

[36] *The Legion of the Bouncy Castle*. http://www.bouncycastle.org/java.html/. [Online; accessed 23 4 2022]. 2020.

[37] *IRIDIUM SLB 9670 TPM2.0 - Data Sheet*. Available : https://www.infineon.com/dgdl/Infineon-SLB%209670VQ2.0-DataSheet-v01_04-EN.pdffileId=5546d4626fc1ce0b016fc78270350cd6. [Online; accessed 20 10 2023]. 2018.

AUTHORS

Anestis Papakotoulas is currently a Ph.D student in the Dept. of Informatics and Telecommunications of the National & Kapodistrian University of Athens (NKUA), Greece. His Ph.D. research focuses on trusted computing in distributed systems. In 2015, he received an M.Sc. (hons) in computer technology and computer systems technology from the Hellenic Open University, Patras, Greece, and, in 2016, he received an M.Sc. (hons) in systems engineering from the Technical University of Crete, Chania, Crete. He also holds a B.Sc. (hons) in military and defence from the Hellenic Military Academy and graduated in 2011 as a signals officer. Since 2018, Anestis serves as a software engineer with the Hellenic Army (Captain) and is a research associate of the Network Technologies Services and Applications (NETSA) Research Lab at NKUA. Anestis has contributed to several R&D projects funded by the EU Horizon Europe, Horizon 2020 and FP7 Research Funding Frameworks. His research interests are on the application of trusted computing in wireless sensor networks and the Internet of Things.

Theodoros Mylonas received in 2003 his B.Sc. on "Geotechnology and Environmental Engineering" from the Technological Educational Institute of Kozani, Greece. In 2010, he received his B.Sc. in geology from the National and Kapodistrian University of Athens, Greece. In 2016 received a B.Sc. in informatics from the Hellenic Open University, Greece. In 2020, he received an M.Sc. in "Mobile and Distributed Computing Systems" from the Hellenic Open University, Greece. Since 2007, he has been working with the company ENVISTA, as an environmental consultant for the construction and operation of large infrastructure projects. Since 2016, he has assumed the duties of the company's IT systems manager. Moreover, from 2020, he has been teaching the courses "Computer Architecture", "Computer Systems Security", "Computer Programming" and "Operating Systems" at vocational training institutes.

Kakia Panagidi received her Ph.D in "Informatics and Telecommunications" from the Department of "Informatics and Telecommunications" of the NKUA in 2019. She holds also an M.Sc in "Administration and Economics of Telecommunications Networks" receiving honors scholarship and a B.Sc. in "Informatics and Telecommunications" from the same department. Furthermore, since January 2012, she has been a member of Communication Networks Laboratory (CNL) of the NKUA as a senior research engineer in the Network Technologies, Services and Applications Lab [http://netsa.di.uoa.gr]. She teaches as an adjunct professor the course of "Project Management", "Design of Databases" and "Algorithms and Complexity" at the Department of Digital Industry Technologies in NKUA. Her research interests focus on key problems on machine learning in pervasive computing and Internet of Things.

Stathes Hadjiefthymiades received his B.Sc. (hons) and M.Sc. (hons) in computer science from the Department of Informatics at the University of Athens, Athens, Greece, in 1993 and 1996 respectively. In 1999 he received his Ph.D. from the University of Athens (Department of Informatics and Telecommunications). In 2002 he received a joint engineering-economics M.Sc. degree from the National Technical University of Athens. From 1992, he was with the Greek consulting firm Advanced Services Group (ASG), working on the design and implementation of telematic applications and services. Since 1995, he has been a member of the Communication Networks Laboratory of the University of Athens. He has participated in numerous projects realised in the context of EU Programs (e.g., ACTS, ORA, TAP, IST, H2020, HE) as well as national initiatives (e.g., Telematique, RETEX). During the period Sept.2001-Jul.2002 he served as an adjunct assistant professor at the University of Aegean, Dept. of Information and Communication Systems Engineering. He joined the faculty of the Hellenic Open University (Patras, Greece) on 2002 as an assistant professor. Since the beginning of 2004 he has been working for to the Faculty of the Department of Informatics and Telecommunications, University of Athens, where he is presently a professor (on large-scale sofware systems). His research interests are in the area of distributed/mobile/pervasive computing and networked multimedia. He is the author of over 250 publications in the above areas. He led the H2020 project RAWFIE (Road-, Air- and Water-based Future Internet Experimentation, www.rawfie.eu) and currently leads the Horizon Europe projects EO4EU (Earth Observation big data processing and application support) and TRACE (intelligent logistics). He also heads the University of Athens M.Sc. programme "Management and Economics of Telecommunication Networks and Information Systems".