I.J. Wireless and Microwave Technologies, **2026, 1, 37-49**
Published Online on February 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.01.03

Data Protection through the Integration of TPM and Cryptography

Rafael A. Menezes
State University of Ceará (UECE), Fortaleza, Brazil
E-mail: menezes.almeida@aluno.uece.br
ORCID iD: https://orcid.org/0009-0008-1208-3110

Ramon S. Araujo
State University of Ceará (UECE), Fortaleza, Brazil
E-mail: ramon.araujo@aluno.uece.br
ORCID iD: https://orcid.org/0009-0000-7063-7689

Lyedson S. Rodrigues
State University of Ceará (UECE), Fortaleza, Brazil
E-mail: lyedson.silva@aluno.uece.br
ORCID iD: https://orcid.org/0009-0009-0179-4707

Erick S. Nascimento
State University of Ceará (UECE), Fortaleza, Brazil
E-mail: erick.nascimento@aluno.uece.br
ORCID iD: https://orcid.org/0009-0006-8861-6935

Rafael L. Gomes*
State University of Ceará (UECE), Fortaleza, Brazil
E-mail: rafa.lopes@uece.br
ORCID iD: https://orcid.org/0000-0001-7922-0695
*Corresponding Author

Received: 14 May, 2025; Revised: 25 August, 2025; Accepted: 27 September, 2025; Published: 08 February, 2026

Abstract: The growing number of cyber threats has made the protection of sensitive data critical. This work presents a solution integrating the Trusted Platform Module (TPM) with AES-CBC and RSA cryptography to mitigate threats like unauthorized key access and data tampering. The architecture uses the TPM as a hardware root of trust and implements a secure device authentication process using the TPM's Endorsement Key (EK). To evaluate its practical viability, we conducted comparative experiments on multiple hardware configurations, measuring the performance impact of the TPM on encryption and decryption tasks for files up to 1GB. Our findings show a clear performance trade-off: TPM integration introduces a measurable overhead that is most significant on lower-end hardware and for smaller files. As file size increases, the relative performance penalty diminishes, though the absolute overhead grows. For instance, decryption operations consistently showed less performance variability than encryption. The results demonstrate that the solution effectively enhances security through hardware-based key isolation, and we conclude that the observed performance cost is a predictable and justifiable price for the robust protection offered against modern cyber threats.

Index Terms: Trusted Platform Module (TPM), Cryptography, Data Security, AES-CBC, RSA, Key Management, Hardware- based Security, Challenge-Response Authentication

1. Introduction

The alarming rise in cyberattack incidents targeting data theft has raised global concern [1]. The increasing sophistication of hacker techniques, combined with society's growing dependence on digital technologies, has rendered both personal and corporate information vulnerable to virtual threats [2]. Companies, government institutions, and even individuals have frequently become targets of cyberattacks, leading to the exposure and theft of sensitive data [3].

Mal- ware plays a significant role, exploiting system vulnerabilities to gain unauthorized access and manipulate confidential data. Prominent examples include spyware, Trojan horses, and ransom ware [4]. To mitigate these threats, regulations such as the European Union's GDPR and Brazil's LGPD establish stringent guidelines for the protection of sensitive information [5, 6]. The GDPR mandates organizations to implement appropriate technical and organizational measures to ensure data security, including encryption and mechanisms to prevent unauthorized access [7].

In light of this scenario, it becomes essential to adopt advanced data protection strategies. In this context, the Trusted Platform Module (TPM) emerges as a robust solution to ensure the integrity and confidentiality of information [8]. TPM is a hardware component that provides a secure environment for the storage and management of cryptographic keys, playing a central role in the security architecture of modern systems. It functions as a root of trust, ensuring that only legitimate software and configurations are allowed to execute [3].

Moreover, TPM enables functionalities such as remote attestation (allowing third parties to verify system integrity) and secure cryptographic key storage, which is critical for preventing attacks that compromise reliability [3]. Its architecture includes Platform Configuration Registers (PCRs), which store integrity measurements and detect unauthorized hardware or software modifications [9]. The application of TPM in file encryption has proven to be an effective approach to protect data against unauthorized access [10]. Integrating this technology into critical systems reduces the risk of credential compromise, as the keys are physically isolated within the hardware, mitigating common threats associated with software- only solutions.

Although various malware detection solutions exist, TPM-based protection stands out by integrating hardware and software securely and efficiently [4]. However, widespread adoption of TPM faces challenges, such as compatibility with legacy systems, implementation complexity in large-scale infrastructures, and the need for greater awareness regarding its capabilities. Nevertheless, its ability to enhance cyber security, especially in environments handling critical data, makes it an indispensable solution.

The primary motivation of this work is to develop a mechanism that combines sensitive file encryption with TPM, ensuring that cryptographic keys are stored and used within a trusted environment isolated from external threats. TPM offers a hardware-based root of trust, safeguarding cryptographic operations from tampering and unauthorized access, which is vital for strengthening the protection of stored data. By integrating file encryption with TPM, this work aims to reduce the risk of key compromise and improve the integrity of protected information.

Furthermore, there is a growing need for efficient solutions to ensure the confidentiality, authenticity, and integrity of critical information, particularly in sectors dealing with sensitive data such as finance, government, and corporate environments. With the increasing adoption of TPM in modern devices, leveraging it as a central mechanism for cryptographic protection presents a promising pathway for mitigating risks and establishing a more robust digital security standard.

Within this context, this work aims to bridge the existing gap by demonstrating the feasibility of integrating TPM with file encryption and contributing to the construction of more secure and resilient systems against cyber threats. Therefore, a TPM-based file protection mechanism is proposed, employing encryption to guarantee the confidentiality, integrity, and authenticity of stored data. The system is capable of securely encrypting files while protecting cryptographic keys against unauthorized access, ensuring that only authenticated devices can decrypt them.

Although existing literature explores hybrid cryptography for data transmission [5] and the integration of post-quantum algorithms with TPM [7], a notable gap remains in the design and empirical evaluation of a complete, hardware-anchored security architecture for data-at-rest using current cryptographic standards. Previous software-only solutions [5] fail to protect cryptographic keys from endpoint compromises, while forward-looking studies on quantum resistance [7] do not provide a performance baseline for readily deployable systems today. Our work addresses this gap by proposing and validating an end-to-end system that not only combines TPM with AES-CBC and RSA but also introduces a novel device authentication mechanism.

This work presents the following novel contributions: (I) **A complete system architecture** that integrates TPM, AES- CBC, and RSA for securing data-at-rest, moving beyond software-only approaches [5] by establishing a hardware root of trust that protects keys from memory-based threats. (II) **A specific device identity and authentication mechanism** that leverages the TPM's Endorsement Key (EK) to generate a unique identifier and implements a Nonce-based Challenge- Response protocol, ensuring that data is only accessible by verified hardware. (III) **A comprehensive performance analysis** on diverse hardware, quantifying the security-performance trade-off of using a TPM for cryptographic operations, which serves as a practical benchmark for developers, a contribution not found in theoretical or future-focused studies [7]. The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 details the proposed solution; Section 4 presents the experiments and results; and Section 6 conclude the paper and outlines directions for future research.

2. Related Work

This section presents studies related to data security using TPM and cryptographic techniques, examining their approaches and contributions. Table 1 summarizes the addressed problems and the approaches adopted in the reviewed references.

In the study conducted by Chaloop and Abdullah [11], a security approach based on encryption algorithms, including AES and RSA, was presented, along with a hybrid solution to protect sensitive data during network transmission. The experiments demonstrated that the hybrid approach improves both security and efficiency in information exchange, providing greater resilience against external attacks. The methodology focused on the comparative analysis of algorithm execution time, showing that the hybrid solution offers a balance between performance and security, making it a viable alternative for applications requiring robust data protection. Furthermore, the study explores the application of these algorithms in different scenarios, assessing the impact of key size and computational complexity on system performance. Despite its significant contribution, the study does not address integration with TPM modules. Consequently, the crypto- graphic keys remain vulnerable to software-level attacks on the host machine. Our work directly mitigates this threat by anchoring the entire key management process within the TPM's hardware-isolated environment.

Another relevant work is by Kil et al. [12], which provides a comprehensive survey on the use of TPM in defense systems. The study highlights the importance of hardware-based security modules in protecting against physical and logical attacks, emphasizing their capability to offer a trusted environment for storing cryptographic keys and performing other critical security operations. In addition, the article analyzes the evolution of TPM specifications and their adoption across different industry sectors. One of the key points addressed is the need for standardization and enhancement of TPM technology to ensure greater compatibility and efficiency in modern systems. However, despite discussing the advantages of TPM, the work does not propose a concrete solution to improve its integration with distributed architectures or to optimize its performance in real-time applications.

In the study by Fiolhais and Sousa [13], the authors explore the incorporation of quantum-resistant cryptography into TPM. The work investigates the feasibility of integrating algorithms such as Kyber and Dilithium to ensure future-proof security against quantum computing threats. The results indicate that these algorithms can replace traditional methods based on RSA and ECC, ensuring the robustness of TPM in emerging scenarios. The research highlights the necessity to adapt current security architectures to address the challenges posed by quantum advancements. In addition to implementing post-quantum algorithms, the study conducts a comparative performance analysis between these new methods and conventional approaches. The tests reveal that, while security is enhanced, implementing the new algorithms may introduce significant computational overhead, requiring optimizations to enable their adoption in embedded systems and low-power devices. While this research is crucial for future-proofing against quantum threats, our work focuses on providing a practical and thoroughly evaluated solution for the current threat landscape. We offer a performance baseline for widely adopted algorithms like RSA and AES, addressing the immediate needs of developers and system architects today.

Zeitouni et al. [14] propose enhancements to data integrity management using TPM in virtualized environments. The research introduces a request scheduling mechanism to avoid overloading the physical TPM when multiple virtual ma- chines utilize an anchored vTPM. The proposed approach proved effective in reducing latency and increasing processing efficiency. The study details the implementation of the request manager and presents metrics demonstrating system performance improvements. Another relevant aspect discussed is the analysis of challenges faced in integrating vTPM into large-scale virtualized systems. The authors point out that, although the proposed solution effectively mitigates TPM over- load, its implementation may require specific adjustments for each computing environment, which could hinder adoption in heterogeneous systems.

Table 1. Summary of Related Work

Reference	Problem Addressed	Approach
[11]	Secure data transmission	AES, RSA encryption and hybrid scheme to protect sensitive data
[12]	Security in defense systems	Survey on TPM usage and protection against physical and logical attacks
[13]	Post-quantum security	Implementation of Kyber and Dilithium in TPM to with-stand quantum threats
[14]	TPM in virtualized environments	Request scheduling to reduce TPM overload in multi-VMsystems

3. Proposed Architecture

3.1 Threat Model

The threat model for this work is centered on a common and highly relevant real-world scenario: an attacker who has achieved software-level access to the system but lacks direct physical access to the hardware. This boundary is crucial and is defined as follows:

- **In-Scope Threats (Software-Level Access):** We assume the adversary can achieve full control over the operating system. This includes executing malicious code (malware, ransomware, spyware), exploiting remote code execution vulnerabilities, escalating privileges to administrator/root, and reading the system's main memory. The primary goal of our solution is to protect the cryptographic keys even when the OS is fully compromised. The TPM enforces this boundary by design; as a separate hardware chip, it performs cryptographic operations internally. Sensitive private keys are created and used within the chip and are never

exposed to the OS or the main system memory, thus protecting them from software-based extraction. This principle of leveraging a hardware root of trust to mitigate software-based threats, including malware that operates with high privileges, is a foundational concept in trusted computing [15].

- **Out-of-Scope Threats (Physical Access):** We consider attacks requiring physical possession and manipulation of the device to be out of scope. This includes, but is not limited to, device theft, opening the chassis to attach hardware probes (e.g., to sniff the LPC/SPI bus), advanced side-channel attacks (such as power or timing analysis) against the TPM chip itself, and cold boot attacks to recover data from RAM. Our system architecture relies on the security guarantees of the certified TPM hardware, whose design and implementation are intended by its manufacturer to be resilient against such low-level physical attacks.

This distinction is justified because software-based attacks are significantly more scalable, common, and can be per- formed remotely, representing the vast majority of threats faced by modern systems. Physical attacks, in contrast, are targeted, costly, and complex. Therefore, securing data against a compromised OS represents a substantial and critical improvement in security. While physical security (e.g., locked server rooms, tamper-evident seals) is an essential and complementary layer of defense, our work focuses on solving the pressing challenge of data protection against remote, software-based adversaries.

3.2 Architecture Overview

The architecture of the proposed solution, illustrated in Fig. 1, comprises several modules and services, each responsible for a specific part of the process. The main components are as follows:

- **M-API**: Handles message exchange between the device and the API.
- **M-ENCRYPT**: Responsible for encrypting sensitive data.
- **M-DECRYPT**: Responsible for decrypting previously encrypted data.
- **M-TPM**: Manages communication with the TPM and performs security operations such as secure key storage.
- **AGTO**: Agent Orchestrator acts as an intermediary, coordinating communication among modules.
- **API**: A service that exposes endpoints for registration, authentication, data transmission, and retrieval.
- **Database**: A NoSQL database used to store encrypted data, registered device records, and operation logs.

The architecture begins with an initialization phase in which the M-TPM module checks for the presence of the TPM and generates an RSA key pair, stored in the TPM's non-volatile memory. The TPM's Endorsement Key (EK) public key is also retrieved, used to generate a unique device identifier (UUID), along with the EK certificate, which attests to the authenticity of the hardware.

Following initialization, the device registration process takes place. In this stage, the UUID, EK certificate, and public signing key are sent to the API, which validates the data using trusted certificates, ensuring that only legitimate devices can access the resources. Upon successful validation, a random number (Nonce) is returned to the device, confirming successful registration and allowing the process to proceed to secure authentication.

Authentication is performed using a Nonce-based Challenge-Response (CR) mechanism, a standard protocol for preventing replay attacks [16]. By requiring the device to sign a fresh, single-use random number (Nonce) for each authentication attempt, the system ensures that an attacker cannot reuse a previously intercepted signature to gain unauthorized access. The process begins when the M-API module sends the UUID to the API, which responds with a specific Nonce. The TPM signs this Nonce using its private key, and the resulting signature is returned to the API for verification. Once authenticated, the device receives a JWT token, enabling secure and authenticated communication for accessing encrypted data. To enhance security, JWT tokens are configured with a short expiration time (e.g., one hour), and a refresh token is issued to allow session continuation without re-authentication. In case of device compromise or revocation needs, the system supports token invalidation through a server-side blacklist mechanism.

During the encryption phase, the M-ENCRYPT module generates a 256-bit AES symmetric key, used to encrypt data in AES-CBC mode. This AES key is then encrypted by the TPM using the previously stored RSA key, ensuring the confidentiality of the information. The encrypted file is also digitally signed by the TPM to guarantee its integrity during storage in the NoSQL database, managed by the AGTO module.

Finally, when data needs to be retrieved, the M-DECRYPT module first verifies the integrity of the encrypted content using the digital signature. After successful validation, the TPM is used to decrypt the AES key using the securely stored RSA private key, allowing the encrypted data to be decrypted. This combination of stages and components, each performing a crucial role, ensures that the data remains confidential, tamper-proof, and accessible only to properly authenticated and authorized devices.

The next section will detail each step of the solution, as well as describe the specific functionalities of the components designed to enable data protection using TPM and cryptographic techniques.

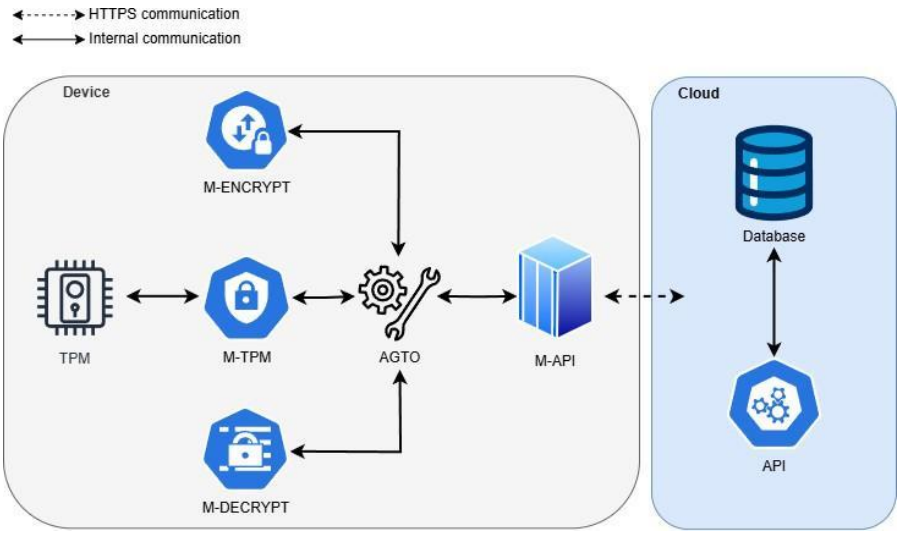


Fig. 1. Solution architecture.

3.3 System Initialization

The initial startup of the solution involves a series of operations where the AGTO triggers the M-TPM modules to establish the foundation for system security and integrity. The initial steps are as follows:

1. Verification of the existence and access to the TPM. If the M-TPM successfully completes this step, it proceeds with the initialization process.
2. RSA Key Generation:

- One key for data encryption and decryption.
- One key for the digital signing of encrypted data, also used for signature verification before decryption.

The generated keys are saved in the TPM's non-volatile memory, allowing their recovery in subsequent initializations.

3. Recovery of the EK public key: The EK public key is retrieved.
4. Dynamic generation of the UUID: Based on the EK, a unique identifier (UUID) for the device is deterministically generated.
5. Generation of the EK certificate: A certificate for the EK is created, ensuring the authenticity of the device.

In subsequent startups, the AGTO retrieves the previously saved keys through the M-TPM, eliminating the need for re-generation.

3.4 Device Registration

After initialization, the device must be registered in the system to enable secure information exchange. The registration process consists of the following steps:

1. The device sends the following data to a specific endpoint:

- UUID generated during initialization;
- EK certificate;
- Public key for signing encrypted data and for use in the CR mechanism.

2. The API validates the EK certificate, comparing it with known manufacturer certificates. If validation is successful, the device is registered.
3. The server returns a Nonce (random number) as a response, serving as confirmation of the registration.

If the device is already registered, the AGTO proceeds to the authentication step.

3.5 Authentication

Once the device is registered, the next step is authentication to ensure that only authorized devices can interact with the API. Fig. 2 illustrates the challenge-response process using the Nonce method, commonly employed in secure authentication protocols

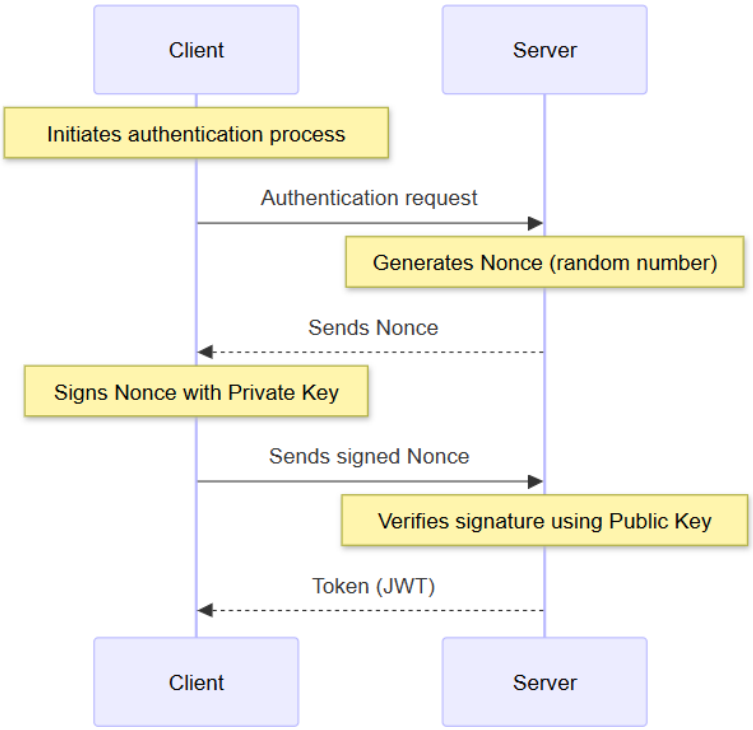


Fig. 2. Challenge-Response Sequence Diagram.

The authentication process is performed using the CR method, as shown in Fig. 2, as follows:

1. The M-API sends the device's UUID to a login endpoint.
2. If the device is pre-registered, the API identifies it and returns a Nonce (random number).
3. The AGTO triggers the M-TPM, which signs the Nonce with its private signing key and returns it to the AGTO.
4. The AGTO triggers the M-API, which requests login by sending the UUID and the Nonce again.
5. The API identifies the device through the UUID and verifies the Nonce signed with the device's public signing key.
6. If everything is correct, the API returns a JWT with a validity of 1 hour, along with a refresh token, which allows extending the authentication time without requiring a new login.
7. The M-API saves the JWT and uses it for future interactions with the API.

3.6 Encryption and Storage

To ensure the confidentiality and integrity of sensitive data, the encryption and storage process is performed in multiple steps:

1. The M-ENCRYPT generates a random AES-256 symmetric key.
2. The sensitive data is encrypted using the AES algorithm in CBC mode.
3. The AES key is encrypted using the RSA-OAEP key, via the M-TPM, which returns the key stored in the TPM.
4. A SHA-256 hash of the encrypted file is calculated, and this hash is then signed with the RSA signing key, also retrieved via the M-TPM.

With the resulting data – encrypted data, encrypted AES key, digital signature, and file hash – the AGTO sends a request to the API via the M-API. Upon receiving the request, the API performs the following actions:

- Verifies the integrity of the data using the hash and digital signature.
- Stores the data in the NoSQL database.
- Returns a unique operation identifier (ID) representing the stored data.

3.7 Decryption and Retrieval

When the user wishes to retrieve stored data, the system initiates the decryption process:

1. The AGTO requests the file associated with the operation ID from the API through the M-API.
2. The AGTO receives the encrypted data, digital signature, and encrypted AES key. It then calls the M-DECRYPT.
3. The M-DECRYPT performs the following steps:

(a) Verifies the digital signature using the RSA private signing key of the TPM, ensuring the data has not been altered.
(b) Decrypts the AES key using the RSA private encryption and decryption key from the TPM.
(c) Uses the AES key to decrypt the data with the AES-CBC algorithm.

4. After decryption, the sensitive data is saved in the user's Downloads folder, allowing access and use.

4. Evaluation and Results

This section presents and analyzes the results obtained from evaluating the impact of using TPM in the encryption (M-ENCRYPT) and decryption (M-DECRYPT) modules. The experiments compare the performance of these modules in two configurations: one using TPM for secure cryptographic key management and another storing key only in memory during execution. Additionally, the developed code and experimental data are available in the project repository[1], along with detailed instructions to ensure reproducibility.

4.1 Experimental Setup

To ensure comprehensive evaluation, the experiments were conducted on three machines with distinct configurations, as shown in Table 2. The hardware diversity allows for assessing the system's behavior under different usage scenarios.

Table 2. Machine Configuration

Component	M1	M2	M3
Memory	16.0 GB	16.0 GB	8.0 GB
Processor	Ryzen 7 5700G	i7-12700	i5-12400
OS	Ubuntu 22.04	PopOS 22.04	PopOS 22.04

The experiments followed a rigorous methodology to ensure the reliability of the results. For each machine, encryption and decryption tests were performed with and without TPM usage. The test dataset included files of various sizes: 1MB, 10MB, 50MB, 100MB, 250MB, 500MB, and 1024MB. Each operation was repeated 10 times to allow for statistically meaningful analysis.

4.2 Results

The analysis was conducted using 95% confidence intervals, which estimate the range within which the true average execution time lies, with 95% certainty. Table 3 presents the confidence intervals for each tested configuration.
To formally assess the significance of the differences suggested by the confidence intervals in Table 3, we conducted a series of independent two-sample t-tests. For each machine and file size combination, we compared the execution times of TPM-enabled versus non-TPM operations. The results consistently showed statistically significant differences ($p < 0.001$) across all tested scenarios, confirming that the performance impact of TPM usage is not due to random variation. The statistical significance was most pronounced on Machine 3 ($p < 0.0001$ for all comparisons), which also exhibited the largest performance gaps in absolute terms. This reinforces the finding that hardware capabilities moderate the TPM's performance impact. The combination of non-overlapping confidence intervals and extremely low p-values provides strong statistical evidence that the observed performance differences are both real and significant.

The results revealed several important aspects regarding the impact of TPM on the performance of encryption and decryption operations. The confidence intervals show that operations involving TPM present greater variability in execution time compared to those without TPM. This variability is especially noticeable for larger files and on Machine 3, which has the most modest hardware configuration among the tested systems.

Machine 2, equipped with an Intel® Core™ i7-12700 processor, demonstrated the best overall performance and the lowest variability in execution time. This suggests that hardware configuration significantly influences the efficiency of TPM-based operations. Machine 3, with lower specifications, exhibited higher instability and longer execution times.

The performance impact of TPM is not linear with respect to file size. For small files (1MB), the relative difference between operations with and without TPM is more pronounced. As file size increases, although the absolute execution time difference also increases, the percentage difference tends to decrease.

The results indicate that decryption operations tend to exhibit lower variability than encryption operations, especially when executed without TPM. This suggests that decryption is more stable and predictable, regardless of the underlying configuration.

Table 3. Confidence Intervals for Execution Time (ms)

Encryption						
Size	**Machine 1**		**Machine 2**		**Machine 3**	
	TPM	**No TPM**	**TPM**	**No TPM**	**TPM**	**No TPM**
1MB	21–26	3–4	33–46	2–3	92–131	2–3
10MB	49–61	17–31	47–66	14–21	109–184	16–24
50MB	195–225	85–139	188–197	67–96	228–364	76–108
100MB	347–408	196–245	307–336	134–184	406–457	155–204
500MB	1724–1931	930–1221	966–1384	660–883	1753–3259	766–975
1024MB	3390–3922	1721–3007	1716–2173	1322–1831	3780–6379	1494–2138
Decryption						
1MB	15–50	2.3–2.9	46–88	1.7–1.9	104–106	1.7–1.8
10MB	47–71	16–21	73–96	11–12	142–155	13–16
50MB	171–211	79–100	150–164	55–57	209–241	61–63
100MB	330–352	159–198	243–252	107–111	311–337	120–124
500MB	1499–1746	768–957	961–1091	528–540	1174–1335	580–615
1024MB	2899–3234	1571–1962	1859–2084	1085–1092	2270–2498	1195–1220

To complement the analysis of confidence intervals and provide better insights into system behavior, a series of graphs was generated showing the evolution of execution time as a function of file size for each machine. Figures 3, 4, 5, 6, 7, and 8 present the results for encryption and decryption across the three machines.

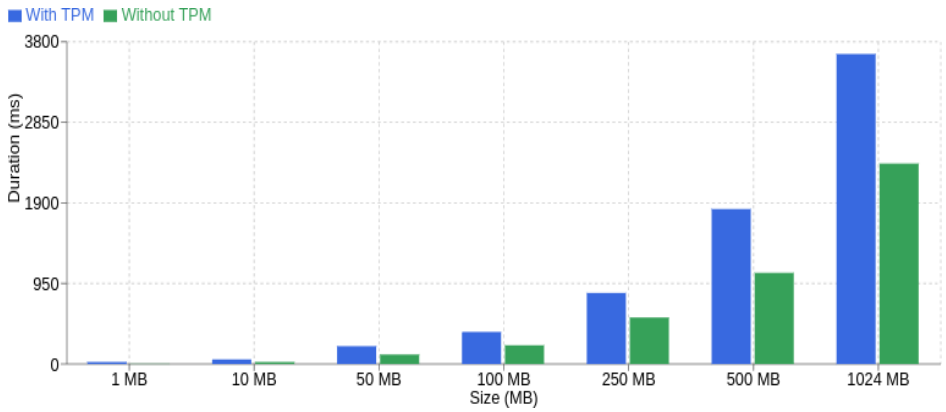


Fig. 3. Encryption time on Machine 1

The analysis of the graphs reveals important patterns in system behavior:

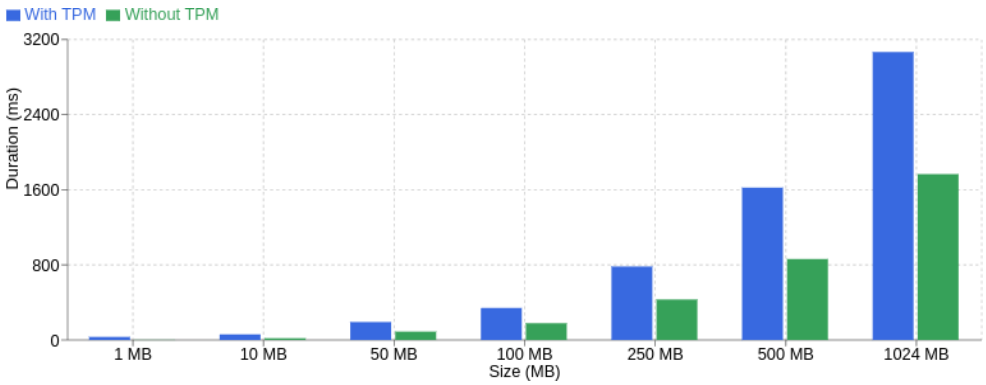


Fig. 4. Decryption time on Machine 1

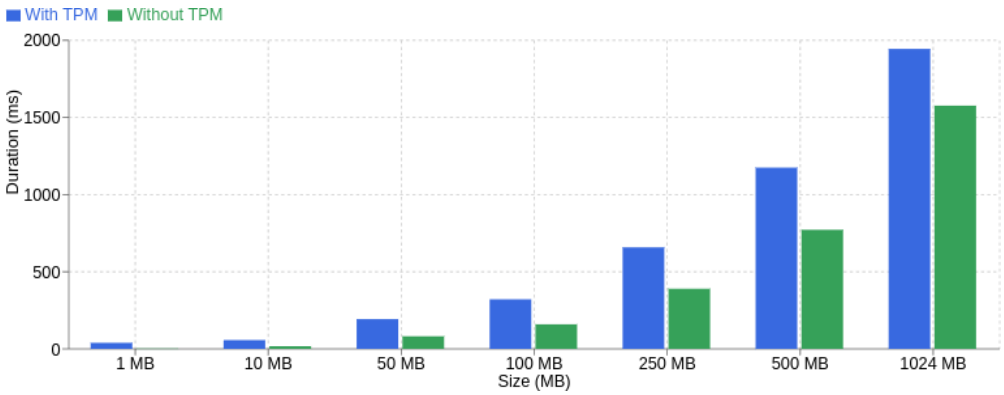


Fig. 5. Encryption time on Machine 2

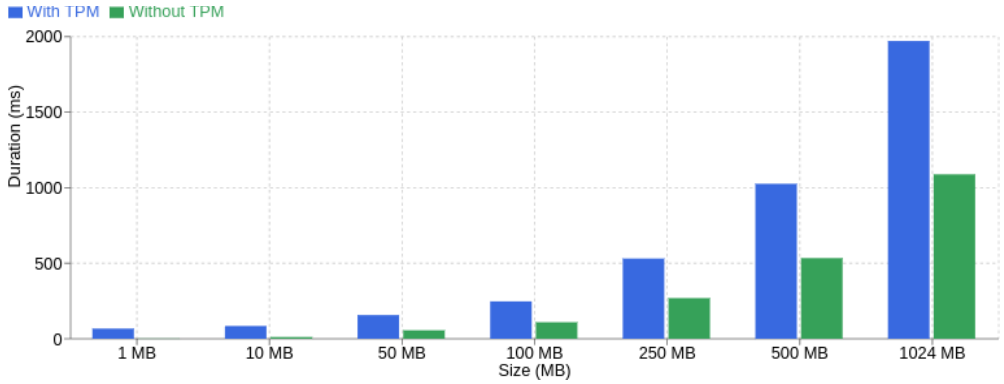


Fig. 6. Decryption time on Machine 2

- **Non-Linear Growth:** Execution time exhibits non-linear growth with respect to file size, with a more significant increase observed from 250MB onward. This pattern is present in both TPM and non-TPM operations, though more pronounced with TPM.
- **Encryption vs. Decryption:** Decryption operations are more efficient, with smoother curves and smaller differences between TPM and non-TPM versions. This suggests that the TPM has less impact during decryption.
- **Scalability Pattern:** For small files (1–50MB), there is a relatively minor difference between TPM and non-TPM operations. However, this gap increases significantly for files larger than 100MB, indicating that the TPM overhead becomes more significant as file size grows.
- **Machine Consistency:** All machines exhibit similar trends, suggesting that the TPM's impact follows a predictable pattern, regardless of hardware. However, the magnitude of this impact varies with the processing capabilities of each system.
- **Hardware Influence:** Machine 2, powered by an Intel® Core™ i7-12700 processor, shows the smoothest curves and smallest differences between configurations. In contrast, Machine 3 (Intel® Core™ i5-12400) demonstrates the highest performance gap. This suggests that more powerful hardware can help mitigate TPM-induced overhead.

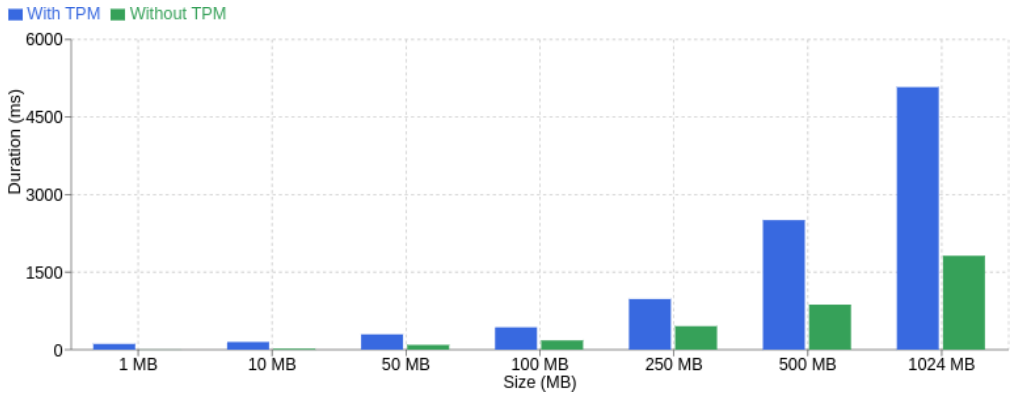


Fig. 7. Encryption time on Machine 3

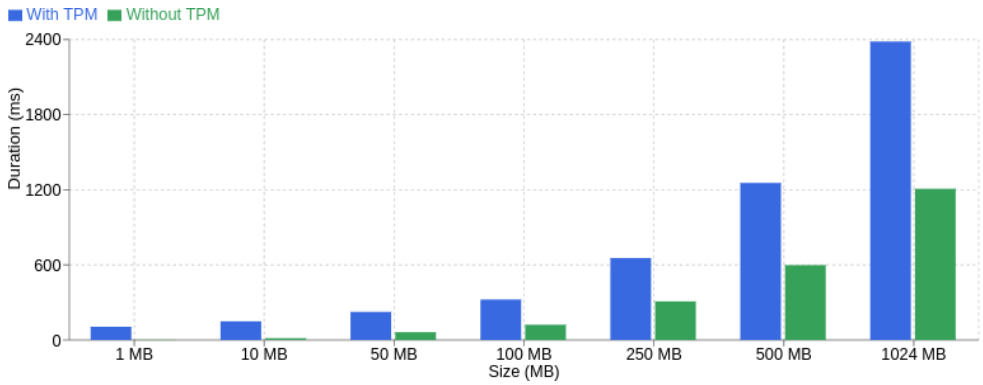


Fig. 8. Decryption time on Machine 3

This visual analysis complements the confidence interval, providing a clearer view of how file size and hardware configuration affect system performance. The graphs also highlight that although TPM introduces significant overhead, this impact is predictable and can be minimized with suitable hardware.

The findings demonstrate that using TPM introduces additional execution time during encryption and decryption processes. This overhead is more pronounced in systems with limited computational resources and in operations involving smaller files. However, it is essential to note that this performance cost must be weighed against the security benefits provided by TPM, which offers a more secure environment for cryptographic key management.

5. Final Discussion

The results obtained in this study reinforce the relevance of our proposed architecture, which advances beyond existing literature by providing a practical and empirically validated integration of TPM with classical cryptography. Unlike software-only models such as [5], our approach demonstrates the tangible security benefits of a hardware-based root of trust, effectively protecting cryptographic keys from endpoint compromise. Furthermore, while works like [7] focus on future threats, our comprehensive performance analysis provides a crucial, real-world benchmark for the overhead associated with TPM-based security today. The experiments confirmed that this overhead is predictable and, we argue, a justifiable trade-off for the robust integrity and confidentiality guarantees our system provides, establishing a foundation for scalable, hardware-anchored security.

From a contribution standpoint, this work advances the state of the art by offering a modular and practical architecture that unifies symmetric (AES-CBC) and asymmetric (RSA) cryptography under a trusted execution environment provided by TPM. The implementation of secure device registration and authentication based on the TPM's Endorsement Key (EK) not only strengthens trust in device identity but also mitigates risks of impersonation and replay attacks. The availability of an open-source implementation further amplifies the contribution, enabling reproducibility and adoption by the wider community.

The benefits of the proposal extend beyond its immediate application in data confidentiality. By leveraging TPM as a root of trust, the system establishes a foundation for scalable, hardware-anchored security that can be adapted to diverse domains, including cloud infrastructures, corporate environments, and critical sectors such as finance and government.

This integration ensures that sensitive data remains not only encrypted but also bound to authenticated devices, thereby elevating both the confidentiality and authenticity guarantees.

Furthermore, the evaluation highlighted that the system performs consistently across different hardware setups, demonstrating its robustness in real-world scenarios. While scalability challenges remain for environments with constrained resources, the architecture's modularity provides a pathway for optimizations and future enhancements, such as the incorporation of post-quantum algorithms and support for virtualized TPM (vTPM) in cloud deployments. These directions will further extend the applicability of the solution to emerging contexts.

In summary, the proposed solution contributes a balanced and practical approach to data security by combining the reliability of cryptographic algorithms with the trustworthiness of TPM hardware. The findings confirm that, despite performance trade-offs, the architecture effectively addresses modern cyber threats, enhances trust in digital systems, and lays the groundwork for future advancements in hardware-based security.

6. Conclusion

This study demonstrated that the integration of a Trusted Platform Module (TPM) with robust cryptographic mechanisms provides effective protection against a critical class of modern cyber threats: those originating from a complete software-level system compromise. Our conclusion is that, despite a measurable performance overhead, the proposed architecture is a highly effective solution for ensuring data confidentiality, integrity, and authenticity under the conditions defined in our threat model.

Specifically, our findings confirm that the solution successfully mitigates the following threats:

- **Malware-based Key Compromise:** By performing all private key operations within the TPM's hardware-isolated environment, the system prevents malware, even with root privileges, from extracting the cryptographic keys necessary to decrypt data.
- **Unauthorized Data Access on a Compromised OS:** Even on a fully compromised operating system, an attacker cannot decrypt sensitive files without successfully authenticating the hardware-bound identity of the device, thus protecting data-at-rest from unauthorized access and tampering.
- **Replay Attacks:** The implemented Nonce-based Challenge-Response authentication mechanism ensures that each session is unique, effectively thwarting attempts by an adversary to reuse previously captured authentication credentials.

The experimental results quantified the performance cost of these security guarantees, showing a predictable overhead that varies with hardware and file size. We conclude that this cost is a justifiable trade-off for the robust, hardware- anchored security provided. The modular architecture not only proved effective but also established a solid foundation for future extensions.

While the proposed solution demonstrates practical viability and security robustness, several limitations should be acknowledged. First, the system's performance overhead, particularly for encryption tasks using TPM, is noticeable on lower-end hardware. This may impact scalability in environments where computational resources are constrained. Additionally, as a deliberate choice to establish a performance baseline using current standards, the implementation does not yet support post-quantum cryptographic algorithms, leaving it vulnerable to future quantum-based threats despite its strong present-day protection.

Another limitation is the reliance on physical TPM availability, which may hinder adoption in legacy or virtualized infrastructures where TPM or vTPM support is lacking or inconsistent. Furthermore, while JWT-based authentication is implemented with expiration and revocation strategies, real-time detection and response to token compromise are not yet integrated.

For future work, our research agenda will expand in several key directions. First, we will conduct a deeper investigation into the cryptographic performance of the TPM itself. This includes ensuring long-term security by integrating post-quantum cryptography (PQC) standards, and also comparing the performance of the currently used RSA-2048 keys against Elliptic Curve Cryptography (ECC) alternatives to identify more efficient security-performance trade-offs. Second, we will address scalability in cloud environments by evaluating our architecture with virtual TPMs (vTPMs). Third, we will conduct a comprehensive comparative analysis against other solutions, such as Hardware Security Modules (HSMs). Finally, to enhance practical deployability, we will design and evaluate mechanisms for robustness and error handling, including secure key backup and recovery protocols. These steps are essential for creating a secure, forward-looking, robust, and thoroughly benchmarked hardware-rooted security solution.

Acknowledgment

The authors would like to thank the CNPq (N^o 303877/2021-9 and N^o 405940/2022-0) and CAPES (N^o 88887.954253/2024- 00 and N^o 88887.972043/2024-00) of Brazil for the financial support.

References

[1] Marcus A. Costa, Yago M. Costa, Yanne O. Almeida, Francisco J. Cardoso, and Rafael L. Gomes. Connection management using automated firewall based on threat intelligence. In *Proceedings of the 2024 Latin America Networking Conference*, LANC '24, page 32–37, New York, NY, USA, 2024. Association for Computing Machinery.

[2] Marcus de V. D. da Silva, Alexandre Rocha, Rafael L. Gomes, and Michele Nogueira. Lightweight data compression for low energy consumption in industrial internet of things. In *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2, 2021.

[3] Shohreh Hosseinzadeh, Bernardo Sequeiros, Pedro R. M. Ina ́cio, and Ville Leppa ̈nen. Recent trends in applying tpm to cloud computing. *SECURITY AND PRIVACY*, 3(1):e93, 2020.

[4] Omar Jarkas, Ryan Ko, Naipeng Dong, and Redowan Mahmud. A container security survey: Exploits, attacks, and defenses. *ACM Comput. Surv.*, January 2025. Just Accepted.

[5] Matheus Silveira, Danielle Santos, Michael Souza, Douglas Silva, Maria Mesquita, Jonas Neto, and Rafael Lopes Gome. An anonymization service for privacy in data mining. In *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, LADC '23, page 214–219, New York, NY, USA, 2023. Association for Computing Machinery.

[6] Ivo Pimenta, Douglas Silva, Evellin Moura, Matheus Silveira, and Rafael Lopes Gomes. Impact of data anonymiza- tion in machine learning models. In *Proceedings of the 13th Latin-American Symposium on Dependable and Secure Computing*, LADC '24, page 188–191, New York, NY, USA, 2024. Association for Computing Machinery.

[7] European Parliament and Council of the European Union. General Data Protection Regulation (GDPR) – Regulation (EU) 2016/679, 2016. Article 32 – Security of processing.

[8] Mohammed Ali Shaik. Protecting agents from malicious hosts using trusted platform modules (tpm). In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pages 559–564, 2018.

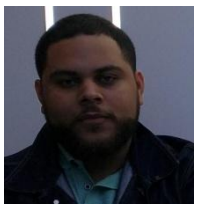
[9] Damiano Turriziani. Protection of private keys with tpm 2.0. Master's thesis, Politecnico di Torino, 2023.
[10] Daniel Moghimi, Berk Sunar, Thomas Eisenbarth, and Nadia Heninger. TPM-FAIL: TPM meets timing and lattice attacks. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2057–2073. USENIX Association, August 2020.
[11] Samir G. Chaloop and Mahmood Z. Abdullah. Enhancing hybrid security approach using aes and rsa algorithms. *Journal of Engineering and Sustainable Development*, 25(4):58–66, Jul. 2021.
[12] DoHyung Kim, Young-Sae Kim, Jin-Hee Han, Jeong-Nyeo Kimand , , and . A comprehensive survey of tpm for defense systems. *KSII Transactions on Internet and Information Systems*, 18(7):1953–1967, 2024.
[13] Lu´ıs Fiolhais and Leonel Sousa. Qr tpm in programmable low-power devices, 2023.
[14] Mariane Zeitouni, Marcela Santos, and Reinaldo Gomes. Melhorias no processo de armazenamento de dados em tpm para gerenciamento de integridade. In *Anais Estendidos do XXIV Simpo´sio Brasileiro de Seguranc¸a da Informac¸a˜o e de Sistemas Computacionais*, pages 270–278, Porto Alegre, RS, Brasil, 2024. SBC.
[15] Seunghun Han, Wook Shin, Jun-Hyeok Park, and HyoungChun Kim. A bad dream: Subverting trusted platform module while you are sleeping. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1229–1246, Baltimore, MD, August 2018. USENIX Association.
[16] Muhammad Nadeem, Ali Arshad, Saman Riaz, Syeda Wajiha Zahra, Ashit Kumar Dutta, and Sultan Almotairi. A secure architecture to protect the network from replay attacks during client-to-client data transmission. *Applied Sciences*, 12(16), 2022.

Authors' Profiles

Rafael A. Menezes is a Master degree student in Computer Science at the State University of Cear á(UECE), being a member of the Laboratory of Computer Networks and Security (LARCES). He has the following research interests: Data Protection, Statistical Analysis, and Security Modeling.

Ramon S. Araujo is an undergraduate student in Computer Science at the State University of Cear á(UECE), being a member of the Laboratory of Computer Networks and Security (LARCES). He has the following research interests: Network Management, Information Modeling, and Statistical Analysis.

Lyedson S. Rodrigues is an undergraduate student in Computer Science at the State University of Cear á (UECE), being a member of the Laboratory of Computer Networks and Security (LARCES). He has the following research interests: Data Communication, Privacy Modeling, and Cybersecurity.



Erick S. Nascimento is an undergraduate student in Computer Science at the State University of Cear á (UECE), being a member of the Laboratory of Computer Networks and Security (LARCES). He has the following research interests: Cloud Computing, Data Science, and Serverless Computing.

Rafael L. Gomes is an Associate Professor at the State University of Cear á(UECE) and the coordinator of the Laboratory of Computer Networks and Security (LARCES). In addition, Rafael is a Productivity Scholarship (DT Level 2) of the National Council for Scientific and Technological Development (CNPq). He received a Ph.D. degree in Computer Science from the University of Campinas (UNICAMP) in Brazil. He was a research visitor at Network Research Lab at the University of California Los Angeles (UCLA) in 2014. He is part of technical program committees of several international conferences, such as the International Symposium on Integrated Network Management (IM), IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE Latin-American Conference on Communications (LATINCOM), and others. He has experience and research on the following

topics: Network Management, Cybersecurity, Software Defined Networks, Resilience Planning, Wireless Networks, and Internet of Things.

How to cite this paper: Rafael A. Menezes, Ramon S. Araujo, Lyedson S. Rodrigues, Erick S. Nascimento, Rafael L. Gomes, "Data Protection through the Integration of TPM and Cryptography", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.1, pp. 37-49, 2026. DOI:10.5815/ijwmt.2026.01.03