

TPM Attestation in Home Environments based on Online and Offline Technologies

Messias D. P. de Mesquita Filho   [State University of Ceará | messias.filho@aluno.uece.br]

Gustavo C. V. Monteiro  [State University of Ceará | gustavo.cesar@aluno.uece.br]

Luiz G. G. Monteiro  [State University of Ceará | luiz.monteiro@aluno.uece.br]

Eliaquim B. Pimentel  [State University of Ceará | eliaquim.pimentel@aluno.uece.br]

Rafael A. Menezes  [State University of Ceará | menezes.almeida@aluno.uece.br]

Rafael L. Gomes  [State University of Ceará | rafa.lopes@uece.br]

Paulo H. M. Maia  [State University of Ceará | pauloh.maia@uece.br]

 Center of Science and Technology, State University of Ceará, Fortaleza, CE, 60714-903, Brazil.

Received: 23 February 2026 • Accepted: 29 July 2026 • Published: 22 September 2026

Abstract Nowadays, society is applying a home office, since the isolation demanded by the pandemic. However, this remote access may lead to boot attack scenarios that are difficult or impossible to detect remotely. This reality led to the necessity of attestation, i.e., a way to prove the state of computers to a corporate server. One promising approach to address this situation is the usage of TPM (Trusted Platform Module) attestation, which is typically used to quarantine tampered computers in the corporate network. However, the usage of TPM attestation at home is still an open challenge, due to the lack of corporate control over network access outside the corporate firewall. Within this context, this paper presents a mechanism to support the TPM Attestation in a home environment, applying online and offline technologies, such as QR-Code and Bluetooth. In this way, the users at home may attest to the security of their workstations before using their login credentials. Results from real experiments using a real TPM chip suggest that the proposed solution is suitable to attest device security in both online and offline scenarios.

Keywords: TPM, Attestation at Home, QR-Code, Bluetooth

1 Introduction

In the last few years, the number of cyber-attacks has been increasing Li and Liu [2021]; Pimenta *et al.* [2026], raising the importance of information security. According to Forbes Brooks [2022], the worldwide number of data breaches reached the mark of 17.2 billion from 2004 to 2022. Additionally, the home office working model was applied recently, due to the isolation demanded by the pandemic Portela *et al.* [2024]; Brito *et al.* [2026]. This remote working model can create another concern about vulnerabilities and threats, since the user is outside the controlled computational environment of the company Costa *et al.* [2024]; Nobre *et al.* [2026]. For example, cyber-attacks on users' workstations, such as boot attacks, became frequent and are difficult or impossible to detect remotely Won and Bhasin [2021].

This reality presents challenges and makes it necessary to create mechanisms that can mitigate the impact of malicious attacks, whether in a corporate or personal environment. The main goal of Trusted Platform Modules is to provide reliable evidence about the state of hardware or software executing on a system Menezes *et al.* [2026]. This evidence is intended to ensure that targets do not engage in certain types of inappropriate behavior. To this end, the Trusted Computing Group (TCG) introduced the Trusted Platform Module (TPM) and the associated concept of Remote Attestation Coker *et al.* [2011]. The attestation process aims to guarantee data integrity, one of the three foundational pillars of information security Death [2017]. Thus, attestation is a method for a

device to prove its state to a corporate server, typically used to quarantine tampered or misconfigured computers from the corporate network.

Existing solutions, such as references Ryu *et al.* [2024]; Zeitouni *et al.* [2024]; Oliver [2021]; Cheng *et al.* [2021], explore traditional attestation using TPM, assuming the existence of a trusted corporate network, continuous connectivity, and centralized enforcement mechanisms. These assumptions are not present in home environments, where devices can operate on untrusted networks or may have intermittent connectivity Nobre *et al.* [2025]. This difference creates a gap between existing attestation solutions and the requirements of home office scenarios.

To address this limitation, this paper presents a hybrid TPM-based attestation approach designed specifically for home environments. The proposed solution combines both online and offline attestation mechanisms. In the online mode, a cloud environment is used to verify the integrity of the home device, using the same concepts as traditional remote attestation. On the other hand, in the offline mode, the attestation is performed locally using a smartphone acting as a verifier, through QR-Code or Bluetooth technologies, making the solution more resilient and portable Gomes *et al.* [2016].

Results from real experiments, as a case study, using a real TPM chipset and mobile phones, suggest that the proposed solution is suitable for attesting the security of a workstation in both online and offline scenarios. The experiments evaluated the support of a mobile phone to provide a mechanism of offline attestation, which makes it much more viable from a

portability point of view.

This work presents the following contributions: (i) A novel attestation model, introducing a method to apply TPM attestation outside corporate-controlled networks, specifically tailored for home office environments; (ii) Hybrid Attestation Architecture, supporting Online attestation (cloud-based attestation server) and Offline attestation (using smartphones as verifiers, enabling device integrity checks without internet access); and (III) Security Enforcement, performing integrity checks before allowing credential input, mitigating risks of credential theft and unauthorized access in remote environments.

The remainder of this paper is organized as follows: Section 2 presents the background concepts related to attestation and TPM. Section 3 discusses the related works. Section 4 introduces the proposed mechanism, while Section 5 discusses the case study performed, and Section 6 presents results. Section 7 details the proposal, its performance, and its applicability in industry. Finally, Section 8 concludes the paper and presents future work.

2 Background

2.1 Trusted Platform Module (TPM)

A TPM is a low-cost, hardware-based standardized security solution that delivers the basis of a safe computing environment TCG [2020]; Wang *et al.* [2016]. The TPM was designed to bridge the gap between a purely software-based realization and the added protection that hardware adds to the security of any system. The TPM protects against snooping for keying material contained within a hard drive. Embedded systems can benefit from the deployment of the TPM device within their platforms and can use this device for secure bootstrap routines and to protect secure application threads running within a system. In conjunction with a secure software stack, the TPM makes a formidable security platform that is much more resistant to software- and hardware-based attacks Kinney [2006].

Currently TPM is in its second version, namely the TPM 2.0 TCG [2019] and provides a fundamental set of security features which have been defined by the Trusted Computing Group (TCG) Wang *et al.* [2016], including (i) health attestation and identification of devices; (ii) secure generation of keys; (iii) enhanced authorization; (iv) quick key loading; and, (v) flexible authorization management.

Devices that have a TPM are capable to generate symmetrical or asymmetrical encryption keys. An Asymmetrical key, mainly through the RSA (Rivest-Shamir-Adleman) algorithm, is actually a key pair: a private key known only to one party and a public key known to everyone. These keys can be used for a variety of purposes, the main ones being signing and data encryption Challener *et al.* [2008]. TPM also uses an asymmetric key for secrecy in some protocols. Along with public keys, a secure hash algorithm is used to provide integrity Challener *et al.* [2008], which is directly related to the attestation.

2.2 TPM-based Attestation

Attestation is a trusted platform's ability to provide evidence about the software or hardware state of a platform. It requires a process in the platform to provide a log of measurements of hardware or software, and the TPM to provide signed summaries of those measurements.

Formally, an attestation process can be defined as a protocol involving the following entities:

$$Attestation = (A, V, E, P)$$

where :

- $A(Attester)$ is the device being verified;
- $V(Verifier)$ is the entity responsible for validating the device;
- $E(Evidence)$ represents the attestation data (e.g., PCR values and signed Quote);
- $P(Policy)$ defines the criteria used to determine whether the system is in a trusted state.

In general, an attestation mechanism basically needs to guarantee a few simple things: that the measurements truly represent the current state of the system (integrity), that they came from a legitimate TPM (authenticity), that they are not old or reused (freshness), and that the device cannot deny that it generated this information (non-repudiation) Coker *et al.* [2011].

TCG utilizes this functionality to demonstrate to a remote entity (e.g., service provider) that a platform seeking to access the service meets specific integrity requirements Achemlal *et al.* [2011].

In summary, to perform attestation properly, every TPM has a key called Endorsement Key (EK), which allows verifying the authenticity of that specific TPM for other entities that want to recognize it; and an Attestation Key (AK), which could be used as a TPM identity and can be shared with other parts for attestation purposes. Finally, there are components called Platform Configuration Register (PCR), which are a set of registers that store integrity metrics containing a hash of instructions executed in a platform Proudler *et al.* [2014].

In TPM 2.0, there are typically 24 PCRs available, though this number can vary depending on the implementation. These registers are divided into banks based on different hash algorithms. For example, a SHA-1 PCR bank may have 24 PCRs, and a SHA-256 PCR bank may have another 24 PCRs. Each PCR is identified by an index number (e.g., PCR[0], PCR[1], ..., PCR[23]). The specific purpose and usage of each PCR index are defined by platform specifications and OEM implementations.

The metrics measure the integrity of any code before launching, ranging from BIOS to application codes Wang *et al.* [2016]. The values stored in PCRs are updated by using the operation of an extended hash. Instead of replacing the current value of the PCR, the TPM performs a cryptographic hash operation to "extend" the PCRs value. This ensures that historical measurements cannot be overwritten or tampered with. The formula for the Extend operation is:

$$PCR[n] = Hash(PCR[n] || NewMeasurement)$$

Basically, these measurements represent the current state of a certain machine. PCRs are used to generate a cryptographic proof of the system state. This proof, called a Quote, is signed by the TPM's AK and can be sent to a remote Verifier. The Verifier checks the PCR values to ensure the system is in a trusted state.

A third party who wishes to know the state of a trusted platform, (i) verifies the TPM's signature and establishes that the PCR values were provided by a genuine TPM (Quote process) installed in a genuine trusted platform and (ii) verifies that the PCR values signed by the TPM are digests of the hardware or software measurements in the unsigned log Proudler *et al.* [2014].

3 Related Works

Regarding the existing works for attestation and TPM, several solutions in the literature were proposed. About this, the Davi *et al.* [2009] explains that there are several ways to verify data integrity using TPM. It can be either dynamically or statically. The way that seems most relevant will be presented in our proposal, but we can say that it is a dynamic verification based on the PCR values. Du *et al.* [2018] brings a proposal to perform data verification dynamically and statically ensuring that the VMs used in that model could have their data intact.

In reference Hosseinzadeh *et al.* [2020], it is discussed how TPM helps to mitigate possible attacks that happen in the cloud. This shows us that using the cloud, we can create an environment to guarantee trust in a computer that can be certified and avoid various types of threats. These attacks are categorized and described as network attack, application attack, malicious/untrusted cloud service providers, attacks on cloud back-end, data tampering, and data leakage. One of the differentials of this work in relation to the others is the way in which we can carry out the communication for the attestation. Our architecture allows us to perform online attestation, a very common type that many works mentioned above can do, and in addition, we can also perform offline attestation, through the use of Bluetooth and QR-Code technologies, with the use of a smartphone.

A relevant work is by Kil *et al.* Ryu *et al.* [2024], 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 cybersecurity in home environments.

Zeitouni *et al.* Zeitouni *et al.* [2024] 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 machines utilize an anchored vTPM. The

proposed approach proved to be 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 overload, its implementation may require specific adjustments for each computing environment, which could hinder adoption in heterogeneous systems.

Oliver *et al.* Oliver [2021] analyse the role of TPM to support security solutions in cloud computing and networking technologies such as 5G and forthcoming 6G, establishing identity, integrity, and provenance of devices to address zero-trust security requirements. In the same way, Cheng *et al.* Cheng *et al.* [2021] propose CloudTA, an extensible remote attestation architecture, which groups all virtual machines on each cloud server and generates trusted evidence by a group. Both proposals use TPM as the root of the security approach, but none of them considers the issues related to remote work in home environments.

Despite existing work on TPM-based attestation, most solutions deal with a traditional remote attestation model, in which devices are verified in a controlled network environment and maintain continuous connectivity with centralized verification servers. In contrast, home environments introduce a fundamentally different context, characterized by:

- (i) operation on unreliable or heterogeneous networks;
- (ii) intermittent or unavailable connectivity.

As can be observed, previous work has not explored a hybrid attestation architecture that combines online and fully offline verification while preserving the security guarantees provided by TPM. This is one of the main aspects that differentiates our proposal. In particular, the approach extends attestation to more decentralized environments, introducing local verifiers and alternative communication methods such as QR codes and Bluetooth.

4 Proposal

4.1 Mitigation of Attacks and Threat Model

The security of the proposed attestation mechanism is analyzed under a defined threat model. We assume that adversaries may attempt to compromise endpoint integrity, intercept communications, replay attestation data, or manipulate firmware and boot components. Attackers may operate over untrusted home networks, attempt man-in-the-middle interception, or exploit insecure connectivity environments.

The proposed solution assumes that the TPM provides a hardware root of trust and that its cryptographic primitives remain intact. Physical invasive attacks against the TPM hardware and side-channel extraction are considered out of scope due to their high cost and specialized requirements. We assume the operating system and firmware may be compromised before attestation, which motivates measurement verification.

One of the primary threats in home office environments is boot and firmware tampering, where attackers compromise the system at a pre-OS level Kirmani and Banday [2024]. The proposed attestation mechanism collects and verifies PCR values generated by the TPM, which contain cryptographic measurements of the system's boot chain. By comparing these values to predefined Golden Measurements, the solution ensures that the system has not been tampered with at the firmware or pre-boot level.

Another major concern is unauthorized access and credential theft, which can occur when users enter login data into compromised systems Arfaoui *et al.* [2023]. The proposed model enforces attestation before login, meaning access credentials are only submitted after confirming the device is in a trusted state. This measure substantially reduces the risk of exposing sensitive user credentials to attackers in untrusted environments.

The solution also acts to mitigate replay and spoofing attacks by using cryptographic nonces and timestamps to validate the freshness and authenticity of each attestation response Vardakis *et al.* [2024]. This prevents attackers from reusing previously captured QR-codes or signed TPM outputs. Moreover, the use of TPM-bound keys (EK and AK) ensures that only legitimate, uniquely identifiable devices can complete the attestation process, mitigating identity forgery or device cloning.

Additionally, the system offers protection against man-in-the-middle and local network attacks, which are more prevalent in home networks Mohamed and Ahmed [2024]. By supporting offline modes through Bluetooth and QR-code scanning (secured with asymmetric encryption and Android Keystore integration), the solution avoids relying on potentially compromised local networks. These mechanisms guarantee that sensitive data is exchanged only between trusted components, maintaining the integrity of the attestation process even in fully disconnected or insecure network scenarios.

In offline attestation scenarios, the smartphone acts as a local verifier and, in practice, ends up becoming part of the trusted computing base. If this device is compromised, it may end up validating incorrect results or even exposing sensitive process information.

To reduce this risk, the solution uses several mechanisms, such as secure storage with hardware support (for example, the Android Keystore), protection of the integrity of locally saved measurements, and secure pairing procedures between devices. Also, the accuracy of offline attestation depends heavily on trust in the smartphone, which is assumed as a system premise. This limitation is inherent in decentralized models and needs to be considered, especially in environments that require higher levels of security.

4.2 Extending Remote Attestation to Home Environments

The proposed mechanism provides a security solution based on TPM to workstations in home environments (or any other environment outside the company), working in two modes: Online, executing over a cloud environment, bringing scalability, reliability, and elasticity to the solution; and Offline, applying QR-Code and Bluetooth technologies. Fig-

ure 1 illustrates the architecture designed for the solution, covering the entities of both online and offline attestation modes.

Before describing the solution, including how it works, it is important to have an understanding of the entities that make up the proposal for this work, as well as to demonstrate how these entities interact with each other, in order to facilitate understanding of the solution. There are a series of entities and communication points that are part of the solution design and will be described next.

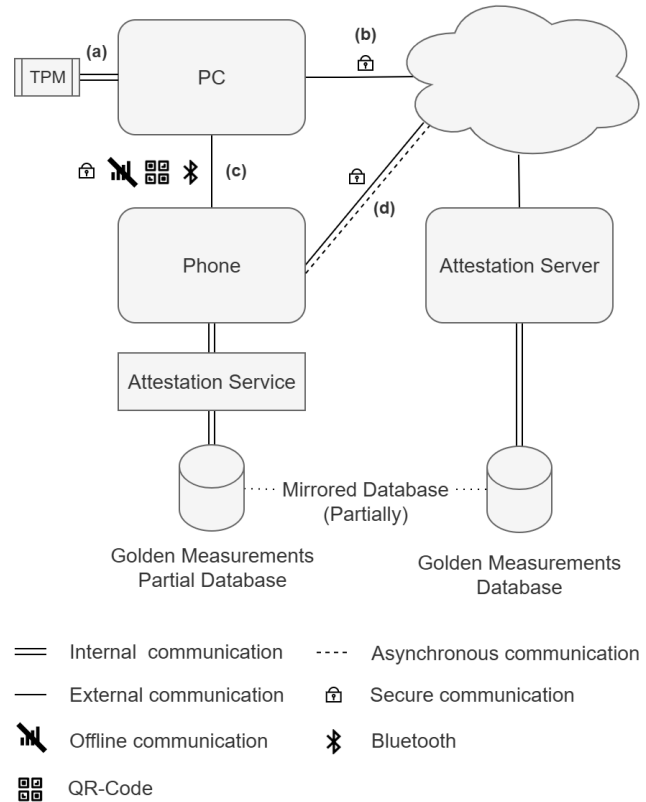


Figure 1. Online and Offline Attestation Architecture

The proposed architecture relies on multiple trust anchors. The TPM serves as the primary hardware root of trust, ensuring the integrity of platform measurements and cryptographic key protection. The attestation server functions as a trust authority responsible for validating device state and maintaining reference measurements. In offline scenarios, the smartphone acts as a local verifier and relies on hardware-backed key storage and secure execution mechanisms to preserve attestation integrity.

To prevent rollback or tampering of locally stored reference measurements, the mobile application stores Golden Measurements using secure storage APIs with integrity protection. Additionally, device pairing procedures ensure that only previously provisioned machines can be verified offline.

4.3 Entities and Components

4.3.1 HTTPS connection

A variation of the HTTP connection, the secure HTTP connection via the Security Sockets Layer (SSL). All communication

between any entity and the Attestation Server must be done securely using an encrypted connection.

4.3.2 Bluetooth/QR-Code connections

These are forms of direct communication between the machine being attested and the Phone when using Offline Attestation.

4.3.3 Attestation Server

The Attestation Server is primarily responsible for authenticating and attesting machines via TPM. It has the logic required to securely validate the health of devices using data and structures provided by the TPM. The attestation allows changes to be detected by authorized parties. This process allows administrators to validate the health of the computer being attested and to identify unauthorized changes. In addition, it has an infrastructure capable of storing information about the devices to be attested, their security certificates, TPM public keys, and any relevant information related to the devices.

4.3.4 Golden Measurements

As explained earlier, PCRs are special-purpose registers inside the TPM that store cryptographic “measurements” of the system’s state. These measurements are essentially hash values that represent the integrity of specific components (e.g., firmware, bootloader, operating system). In our solution, the Golden Measurements are formed by the values of all PCRs in the TPM and represent the trusted state of a machine at a given point in time. Therefore, these values are considered a reference point for future comparisons. If something changes, it means that there has been some change in the machine, and therefore it is untrusted, until the changes are validated.

4.3.5 Partial Golden Measurements

Partial Golden Measurements are basically a cut of the Golden Measurements present in the Attestation Server. However, they are specifically aimed at just one machine. These measurements are used to save the trusted state of the machine in offline attestation mode and are stored on the mobile device, which acts as the Verifier.

4.3.6 Client TPM

Performs the process of interacting with the machine’s TPM to obtain the measurements that will be used during the machine’s attestation process, e.g., to obtain PCR digests.

4.3.7 Phone Application

The smartphone application acts as a verifier in offline scenarios. It validates attestation evidence received from the client device using locally stored reference measurements and secure cryptographic operations.

4.4 Key Provisioning

Initially, there is a step called Key Provisioning, in which the machine makes its information available that identifies it as a trusted platform. The provisioning and device identification process is based on and adapted from [TCG] and takes into account the provisioning of the AK, used for device attestation.

This process provides the following assurances:

1. A TPM is authentic, by validating the EK certificate;
2. This specific TPM is authentic and is represented by the EK certificate (i.e., the EK private key resident in this TPM corresponds to the public key in this EK certificate);
3. This AK is authentic (the AK is resident in the same TPM as the EK private key corresponding to the public key in this EK certificate).

The detailed procedure is shown in Figure 2 and is described below:

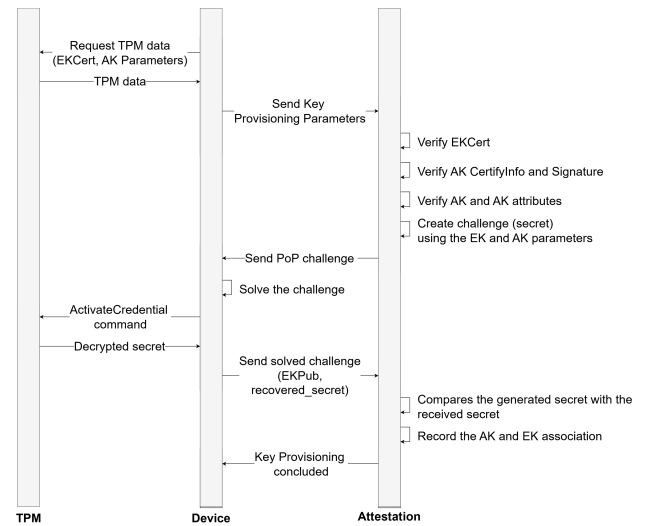


Figure 2. Key Provisioning for Attestation

1. Client software uses the TPM to create a Restricted, fixedParent, fixedTPM Ordinary signing key (the AK) using TPM2_Create.
2. Client software uses TPM_Certify to certify the AK key (the key created in step 1) with the OEM EK, producing a signed TPMB_Attest structure. This will show that the AK key is in the same TPM as the EK and, by including the Name field, serves as a signature of the AK key public data, which specifies the required key attributes verified in step 5.c.
3. Client software builds the Key Provisioning Parameters (KPP), including the:
 - (a) TPM EK Certificate.
 - (b) AK Public Area (TPMT_PUBLIC structure of the AK).
 - (c) AK key TPM2B_ATTEST structure returned by TPM2_Certify.
 - (d) AK key TPMB_Attest structure with signature.
4. The device software sends the KPP to the Attestation Server.

5. The Server verifies the received data:
 - (a) Compare the EK Pub present in the EK Certificate with the data present in the Server Database.
 - (b) Verify the EK Certificate using the indicated TPM manufacturer's public key.
 - (c) Verify TPM residency of AK key using the EK public key to validate the signature of the TPMB_Attest structure.
 - (d) Verify the attributes (TPMA_OBJECT bits) of the AK public area to ensure that the key is a Restricted, fixedTPM, fixedParent signing key.
6. The Verifier issues a challenge to the device to validate that the TPM has the EK matching the certificate in step 3a and that the AK whose public area is included in 3b is loaded in the device's TPM. The challenge data blob is created using the following procedure:
 - (a) Calculate the cryptographic Name of the AK, by hashing its public area with its associated hash algorithm, previously defined with the Algorithm ID of the hashing algorithm;
 - (b) Using the sequence described in the TPM 2.0 Library Specification [TCG] Part 3, section 12.6.3 (TPM2_MakeCredential Detailed Actions), create the encrypted "credential" structure to be sent to the device. When building this encrypted structure, objectName is the Name of the AK calculated in step 'a';
 - (c) The Verifier sends the encrypted "credential" blob to the enrolling device.
7. The Client software uses the TPM2_ActivateCredential command. During this operation, the TPM verifies the AK's Name (and therefore the entire AK public area) using the EK. Because the EK is used, this operation requires Endorsement Hierarchy Admin Authorization. The returned "secret" is the Proof of Possession (POP).
8. The device returns the recovered secret to the Server Verifier. The Server Verifier matches the received value with the secret retained earlier. The Server Verifier's receipt and validation of the "secret" is a Proof of Possession that demonstrates that the AK is resident in the TPM described by the EK Certificate, is non-duplicable, and fixedTPM.
9. The Server records the association between the public part of the EK (EKPub) and the public part of the AK (AKPub) in its local database for future identification of the machine.

4.5 Online Attestation Solution

In an online environment, the Device is attested through online communication. Before users type their login credentials, a mechanism of attestation is started to verify the Device's integrity.

Key Provisioning is the first step necessary before performing attestation, which was already discussed in the Section 4.4. At the end of this process, the Attestation Server stores the EKPUB and AKPub in a local database, usually integrated with the corporate Public Key Infrastructure (PKI). Another object typically stored in the Attestation Server's local database

is a policy that identifies an expected initial state (measurements) of the Device. The measurements are stored in a local database and are used later, during attestation, to determine if a system is in a good state. In our solution, initial base values, or "Golden Measurements", can be obtained after the first attestation. From that point on, PCR values are considered reliable and serve as a basis for future measurement comparisons. If the values eventually change on the machine, for example, due to a hardware change, the Attestation Server must be able to identify the changes and analyze them. Verification can be based on other resources, such as the known TCG Event Log TCG [2025]. However, this type of treatment is not the focus of this work.

The Device starts a communication with the Attestation Server to request an attestation of the machine, sending its EKPUB to serve as identity. The Server then verifies whether the Device in question has already gone through the provisioning process and has an associated AK. Once this verification is successful, the Verifier generates a "challenge", which may contain information such as¹:

1. Selection of PCRs, which should be used in the Quote;
2. Nonce, which should be used to prevent replay attacks and inserted into the Quote.

The Device receives the data from the Verifier and begins loading the necessary TPM keys, which include the EK and AK. Once the keys are loaded, the TPM2_Quote() command is executed, passing as main parameters the list of selected PCRs, the received Nonce, and the AK. This command generates an instance of the TPMS_ATTEST attestation structure defined by the TCG. It consists of two other structures, which form the Quote and its respective Signature. In addition to this data, the Device reads the TPM PCRs by executing the TPM2_PCR_Read() command several times to capture the hash values of the desired PCRs, as requested by the Attestation Server. These Attestation Parameters (AP) are then aggregated and sent to the Verifier for Appraisal.

The Attestation Server analyses the received information by verifying:

1. Device identity: done through EKPUB check on the database;
2. The Quote signature: using the AKPub associated with the Device;
3. The Quote attributes, in addition to the Nonce stored in the "extraData" attribute;
4. The integrity of the received PCR values: validated by hashing the result of concatenating all PCR values. The resulting hash is then compared with the PCR Hash received from the Quote. If the values are the same, the PCR values are intact and trustworthy;
5. The change in PCR values: Once the received PCR values are validated and trustworthy, the Verifier compares each received PCR value with the PCR values already known in the database ("Golden Measurements". If any value differs, it indicates that the device is not in a trusted state. Otherwise, the attestation is considered successful.

¹The data sent may vary, depending on the needs and policies of the Attestation Server

At the end of this process, the Attestation Server informs the Device of the attestation result. Figure 3 depicts the attestation and communication flow between the entities involved.

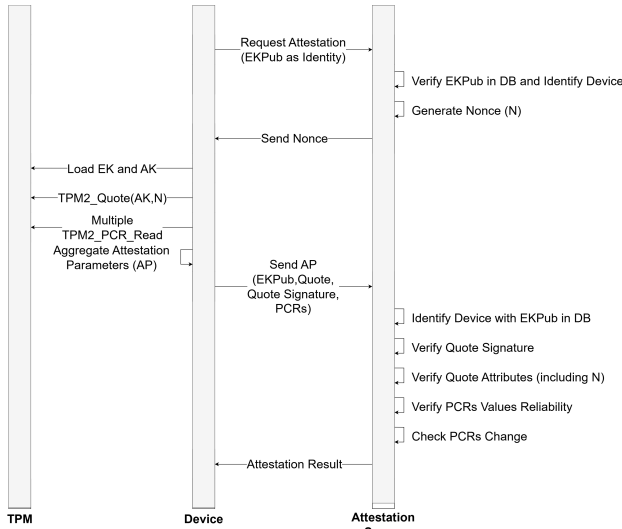


Figure 3. Cloud-based Attestation flow diagram

4.6 Offline Attestation Solution

The Offline Attestation Solution is designed to address scenarios where there is no availability of connection to the corporate or cloud infrastructure, offering a secure and reliable means of verifying the integrity of home or remote workstations, even in disconnected environments.

This solution extends the architecture already designed and presented, providing attestation mechanisms for devices that operate independently of the Internet, allowing the attestation process to be performed locally using a smartphone, which acts as the verifier, based on the online solution. The phone receives or exchanges information directly with the user's computer via QR code or Bluetooth, ensuring data confidentiality and integrity through asymmetric cryptography and secure key storage mechanisms (such as the Android Keystore).

Based on the online approach, where a centralized server performs verification, the offline solution is supported by a smartphone, acting as a centralized local server, and relies on previously obtained Golden Measurements (reliable reference values from the system's TPM registers) securely stored on the smartphone. By comparing these values with the current values collected from the machine's TPM, the phone is able to determine if the device remains trusted.

For the offline solution to work, at least once, the machine must be attested online, through the process of provisioning keys and challenge. Offline attestation introduces unique security challenges due to the absence of continuous connectivity with centralized infrastructure. The proposed design mitigates these risks through cryptographic protections and attestation freshness guarantees.

The following subsections detail how the offline attestation process works, describing the two approaches (via QR code and via Bluetooth) and how each enables secure, portable, and resilient validation of device integrity.

4.7 QR-Code Offline Attestation

This approach consists of using QR-Codes as a means of data transfer between the user's computer and a verifying smartphone, having two main roots of trust: (i) the TPM itself, which is on the client side, and (ii) the Android Keystore and security mechanisms, on the Phone side. This is a simple, accessible, and secure method for validating the integrity of a device without the need for a network connection.

During the process, the computer generates different types of QR-Codes containing essential information for pairing and verification between devices, such as TPM measurements and the signed quote, also encapsulating an Access Code. This Access Code, generated and protected by the TPM for each new attestation request before the user's login, is only revealed on the smartphone if the attestation is completed, ensuring that access to the system only occurs in an intact and reliable state.

To guarantee the confidentiality of both the data and the Access Code contained within the Quote transmitted through the QR-Code, the Quote is encrypted using a 2048-bit RSA asymmetric key generated by the mobile application via the Android Keystore during the Pairing Phase. The public portion of this key is securely shared with the computer in advance, enabling the device to encrypt the Quote before embedding it into the QR-Code.

The smartphone acts as a local verifier. Through it, the QR-Code is read by a dedicated application, which decodes the displayed data, compares the received information with the previously registered Golden Measurements, and performs all the necessary attestation steps to identify the actual state of the machine. Since the mobile application is used to scan the QR-Code and has the private part of the RSA asymmetric key, it is possible to decrypt the Quote, perform the machine attestation, and, if the machine's state is as expected, the Access Code is extracted from the Quote and displayed to the user, who can enter the Access Code on the machine, allowing for secure authentication.

The use of QR codes makes the process portable, low-cost, and compatible with different platforms, significantly expanding the applicability of TPM-based attestation in decentralized home and corporate contexts.

The following sections detail the technical operation of this approach and present its two main phases, device pairing and the attestation process itself.

4.7.1 Phase 1 — Device Pairing with QR-Code

The pairing process establishes a secure trust relationship between the client machine and the smartphone, enabling future offline attestations to be performed without requiring a direct network connection.

The procedure begins with the generation of a QR code on the client machine, containing its Endorsement Key (EK). Once the smartphone scans this QR code using the attestation application, it proceeds to generate a 2048-bit RSA asymmetric key pair within the Android Keystore, ensuring that the private key remains securely confined to the hardware-backed storage of the mobile device.

Next, the smartphone exports the public portion of this key,

referred to as the Android Encryption Key (AEK). This key is securely transmitted to the client machine and will later be used to encrypt the TPM Quote that the PC generates during the attestation phase. This encrypted Quote, embedded in a QR code, forms the secure channel through which attestation data is exchanged between both devices.

After generating the AEK, the smartphone contacts the Attestation Server, sending both the EK (to identify the device) and the AEK (to establish encryption parameters). If the EK is recognized by the server, it indicates that the device has already undergone at least one successful online attestation. In this case, the server retrieves the corresponding Golden Measurements and transmits them to the smartphone. These measurements are then securely stored locally using the Secure Device Storage library, ensuring their integrity and availability for future offline attestations.

Additionally, the public portion of the Attestation Key (AK), which is cryptographically associated with the machine's EK, is also transmitted to the smartphone during this process and is stored securely along with the Golden Measurements. This allows the verifier to later authenticate the TPM signatures and validate the provenance of the attestation data received from the client.

The entire process can be seen in the flow diagram depicted by Figure 4.

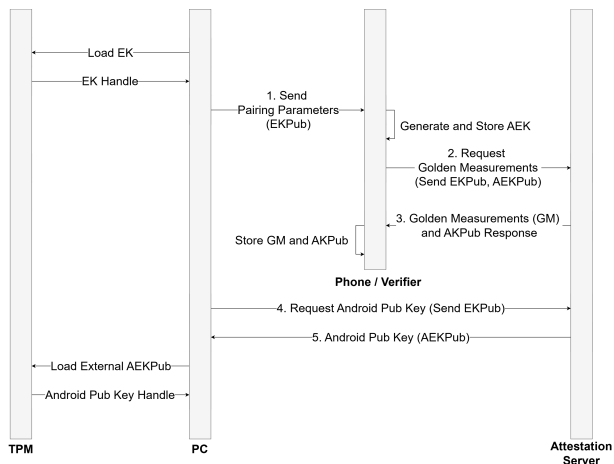


Figure 4. Flow diagram detailing Phase 1, device pairing, for offline attestation with QR-Code Technology

4.7.2 Phase 2 — Device Attestation with QR-Code

After the successful completion of the pairing phase, the offline attestation process can be executed whenever the user needs to validate the integrity of their workstation before logging in. In this step, attestation occurs entirely locally, through the secure exchange of information between the client computer and the verifying smartphone, using a QR code as a means of transferring encrypted data.

Upon initiating attestation, the computer generates a new QR-Code, called the Attestation QR-Code, which encapsulates what is called Attestation Parameters (AP), which include: the public portion of the Endorsement Key (EKPUB), the values of the selected PCRs, and the Quote signed by the AK, accompanied by its respective digital signature.

Before generating the QR code, the machine creates an Access Code, a short-lived credential used to grant the user access after successful attestation. This code is generated and securely stored within the TPM using the TPM2_Seal() operation, ensuring that it can only be released if the system state remains unchanged. The Access Code is then inserted into the extraData field of the Quote structure, linking the credential to the verified system state.

To preserve the confidentiality and integrity of the attestation data, all Quote content (including the Access Code) is encrypted using the public portion of the Android Encryption Key (AEK), previously established during the Pairing Phase. Therefore, only the paired smartphone, which keeps the corresponding private key securely stored in the Android Keystore, can decrypt and interpret the information contained in the QR code.

After reading the Attestation QR code, the smartphone decodes the information and start checking the PubEK at its local database to identify the device; after, it decrypts the Quote using the AEK's private key and verifies the received data against the Golden Measurements and the AEK's public key stored locally. The verification process includes checking the integrity of the PCR values, the authenticity of the TPM signatures, and validating the current attestation using the Clock (timestamp) contained in the Quote.

If all verification steps are completed and the device is considered to be in a trusted state, the smartphone extracts the Access Code from the Quote and displays it to the user. The user then enters this code into the computer, which validates it internally via the TPM and grants access to the system. If the attestation fails, indicating system tampering or discrepancies in the PCR values, the Access Code is not displayed, and access to the system remains blocked.

This two-step attestation process, based on QR codes, ensures secure and network-independent validation of device integrity, leveraging the hardware-based security of both the TPM and the smartphone. By combining asymmetric cryptography, reliable measurements, and hardware-sealed credentials, the approach provides a robust mechanism for authenticating devices even in completely offline scenarios. Figure 5 depicts the process in a more detailed way.

4.8 Bluetooth Offline Attestation

The offline attestation approach via Bluetooth was designed to allow for direct and bidirectional verification of a device's integrity, without the need for an internet connection. Unlike the QR-Code solution, which uses unidirectional exchange of visual information, the Bluetooth approach enables interactive communication between the computer and the smartphone, replicating the behavior of a traditional online attestation, but in a completely local environment. However, similar to the QR-Code offline solution, at least one fully online attestation must have been performed. The Bluetooth Offline Attestation Solution is described in the next sections.

4.8.1 Pairing and Attestation Process with Bluetooth

The operation of the offline Bluetooth solution is based on the identification of events that happen during the exchange of

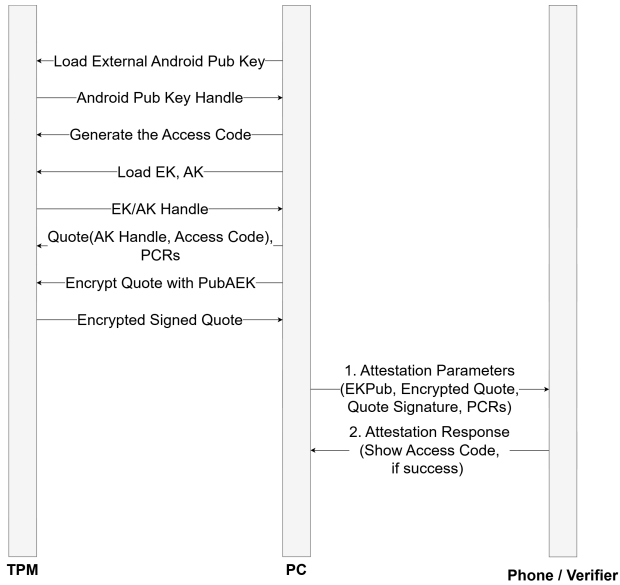


Figure 5. Flow diagram detailing Phase 2 of the offline attestation via QR-Code Technology

data between the two main entities involved: PC (acting as the Attester) and Phone (acting as the Verifier). The approach is structured in a sequence of events identified by TAGs, which control the ordered exchange of messages between devices. The entire Pairing and Attestation Process can be visualized in Figure 6.

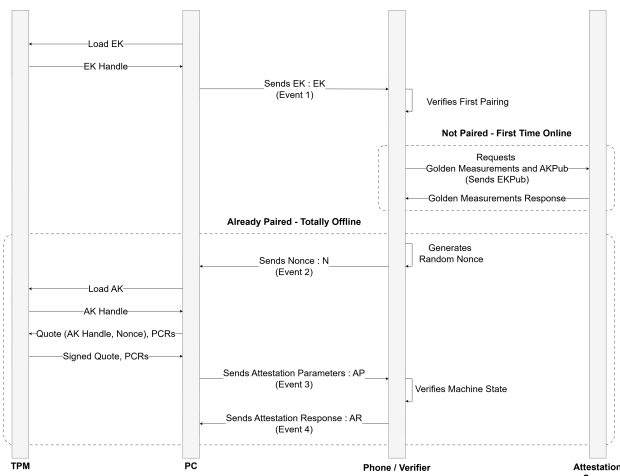


Figure 6. Offline Pairing and Attestation Process via Bluetooth

The events and their respective TAGs are:

1. *PC* → *Phone* : EK Shipment — The PC initiates the process by sending its Endorsement Key (EK) to the Phone, requesting the start of the attestation.
 - (a) TAG: EK
2. *Phone* → *PC* : Nonce Shipment — The Phone generates a random 20-byte Nonce, used to guarantee originality and prevent replay attacks. This Nonce is sent to the PC as a cryptographic challenge.
 - (a) TAG: N
3. *PC* → *Phone* : Sending the Attestation Parameters (AP) — The PC loads the AK and EK keys into the TPM and executes the TPM2_Quote() command, passing the Nonce and the selection of PCRs to be verified. The

result, consisting of the Signed Quote, Quote Signature, PCR values, and associated keys, is then sent to the Phone.

(a) TAG: AP

4. *Phone* → *PC* : Attestation Response — The Phone validates the received content by verifying the Quote signature with AK's public key and comparing the PCR values with the Golden Measurements stored locally. The verification result is then returned to the PC.

(a) TAG: AR

The process is started by loading the EK from the PC. Then, the PC requests attestation by sending the EK to the Phone (**Event 1**). The Phone recognizes the event through the EK tag. At this point, a check is performed. During this process, if the Phone does not have previous device records (that is, there is no association between the received EK and the stored reliable measurements), it automatically initiates a pairing process with the Attestation Server to retrieve and locally store the corresponding Golden Measurements and associated keys. This mechanism ensures that only previously recognized and provisioned machines can be attested. The main purpose of this pairing is to obtain the PC-related Golden Measurements, using the EK as the identifier, and to store them securely locally. If the EK is on the Attestation Server, then that PC has already performed at least one online attestation. On the server side are stored the Golden Measurements from that machine that will be stored securely on the Phone with the Secure Device Storage library support, in order to be used in the offline solution. If the pairing has already been performed, the Attestation Process is started.

The Attestation Process is started by the Phone through the generation of a random 20-byte Nonce. Then, the Phone requests the Attestation Parameters, sending the Nonce to the PC (**Event 2**). The PC receives the Nonce and starts the Quote generation and PCR selection. For this, the PC executes the TPM2_Load() command to load its Attestation (AK) and Endorsement (EK) keys. It then executes the TPM2_Quote() command, providing the Nonce and the selection of PCRs and signing with the Attestation Key (AK) loaded. The PC also runs the TPM2_PCR_Read() command multiple times to read the selected PCRs. By default, all PCRs, are selected (PCR-0 to PCR-23), and the Verifier (Phone) is responsible for the verification policy of the PCRs of interest. Finally, the PC sends the Attestation Parameters to the Phone, consisting of the EKPub, Quote, Quote Signature, and PCRs (**Event 3**).

The Phone will check the EK in its local database to identify the device, and then will verify the other information by performing the appraisal policy, comparing the software measurement logs against the good-known-state of the system software configuration stored in the local database. Finally, the Phone sends the Attestation Response to the PC (**Event 4**), concluding the attestation process.

The main advantage of the Bluetooth approach is that it allows for the continuous and cryptographically protected exchange of information between devices, eliminating the need for network access and maintaining the same level of security as online attestation. Furthermore, the use of the Bluetooth channel (combined with mutual authentication measures and

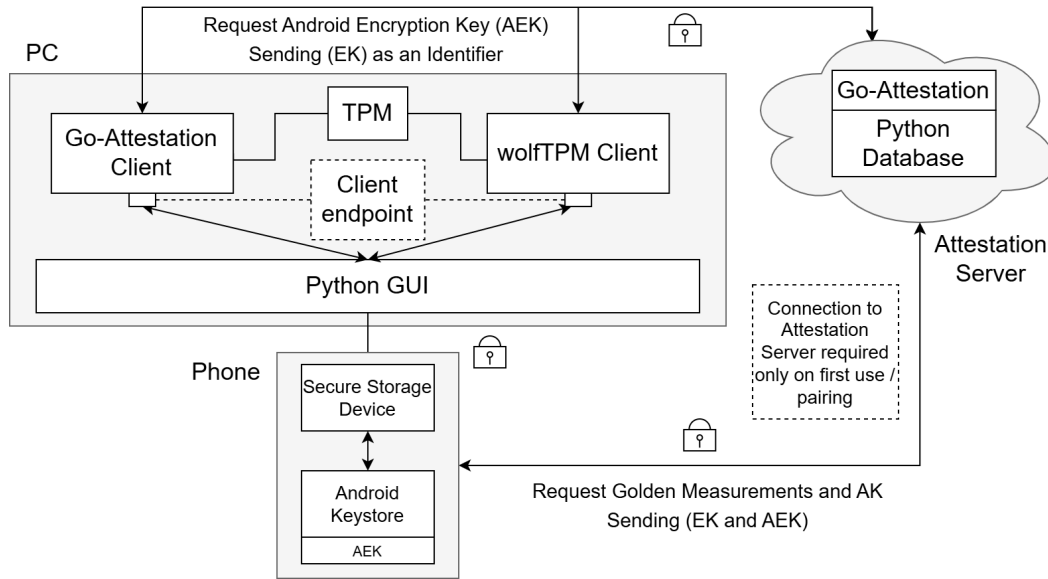


Figure 7. Online and Offline Attestation Architecture

cryptographic protection) reduces exposure to man-in-the-middle attacks and local interceptions, common in unprotected home networks.

Thus, offline attestation via Bluetooth offers an efficient, secure, and automated solution for validating the integrity status of workstations in environments without connectivity, reinforcing the reliability of the proposed hybrid attestation ecosystem.

5 Case Study

Two Proof-of-Concept (PoC) were developed to validate, in real-world scenarios, the applicability and robustness of the proposed architecture, covering both online attestation and offline approaches (QR-Code and Bluetooth). The experimental environment included three main entities: the client machine (Attester), the Attestation Server, and the Smartphone acting as a Local Verifier. Figure 7 illustrates the topology and components used.

5.1 Online Attestation

The online attestation solution is intended to allow users working on a non-corporate network to attest to their computers before using their login credentials. The solution has two main entities: PC (Attester) and Attestation Server.

On the Attester side, two main programs were developed: the first one, using the Go-Attestation library², through the Golang language; and the second, using the wolfTPM library³, through the C programming language. Both clients were responsible for making calls to the TPM chip, collecting the measurements (PCRs), and transmitting the information to the Attestation Server.

The user interacted with the system through a graphical interface in Python, which allowed them to select the type of verification (online or offline) and view the verification result,

with success or failure messages depending on the device's status.

On the server, a verification service based on the Go-Attestation library was developed, capable of receiving, validating, and processing data sent by the client. This service executed the main steps of the attestation process, challenge generation, cryptographic key verification, quote authentication, and PCR value validation, maintaining a local database.

A second service, developed in Python, was responsible for managing and verifying the Golden Measurements, ensuring that the received measurements corresponded to the reliable state stored in the database. Communication between client and server was entirely protected via the HTTPS protocol, ensuring confidentiality and integrity in the exchange of information.

5.2 Offline Attestation

The offline solution was designed and tested with an Android application developed in Java that acted as a local verifier. The app supported two forms of communication with the machine: QR code and Bluetooth.

The user could interact with the attestation program through a user-friendly graphical interface on the machine. Initially, two main options were available: run "Offline Attestation" and "Online Attestation". Selecting the Offline Option, two ways of attestation were presented, via QR-Code or Bluetooth.

Selecting the "Attestation via QR-Code" option opened a new screen. On this screen, a Pairing QR Code was displayed, which the user should scan using the mobile application. Then, after pairing, a new screen containing a new Attestation QR-Code was displayed, which should be scanned by the user in the same way as the previous QR-Code. Once the last QR-Code was scanned, the attestation process was started in the mobile application, and at the end of the process, in case of success, a 6 (six) digit Access Code was displayed to the user in the application itself. The user entered this code into the machine's attestation program in order to gain access to it. In the event of an error, for example, the machine was not

²<https://github.com/google/go-attestation>

³<https://www.wolfssl.com/products/wolftpm/>

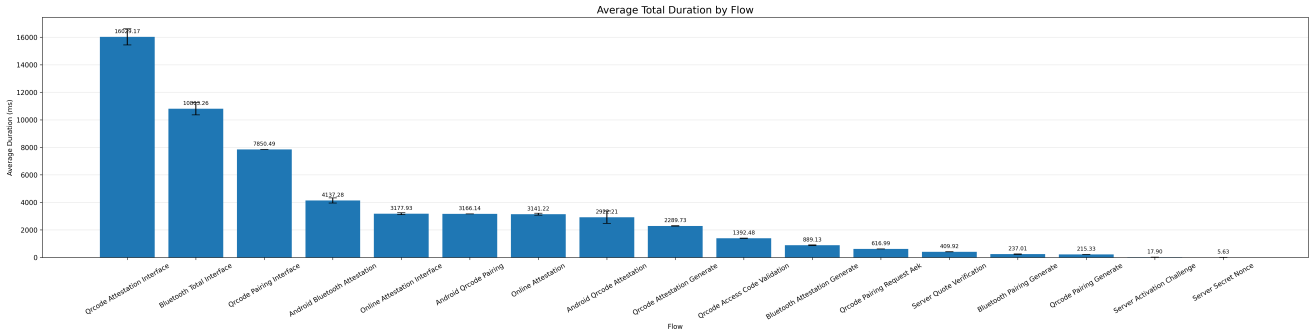


Figure 8. Total duration

in a trusted state, the Access Code was not displayed, and access to the machine was not granted.

When selecting “Attestation Via Bluetooth”, a new screen was launched. On this screen, an alert is displayed indicating that the attestation process would start after pairing the machine with the mobile device. Thus, using the application on the mobile device, the user could select the Attestation via Bluetooth option and then choose the device they would like to attest, displayed in a list of nearby Bluetooth devices. When selecting the device (machine to be attested), a Bluetooth connection was established between the machine and the mobile device, and then the pairing and attestation process was started. In the Bluetooth flow, the communication is bidirectional: the smartphone sends a nonce, the client generates the Attestation Parameters, and the verifier performs the local validation. This model reproduces the logic of online attestation, but without dependence on external connectivity. At the end, after completion of the process, a warning message was displayed to the user, both on the machine and in the mobile application, indicating the success or failure of the attestation.

6 Performance Experiments

Despite the functional evaluation, we performed performance experiments to analyze the execution time and memory consumption of the main components involved in both the online and offline attestation workflows. The evaluation considered all major execution paths, including online attestation, QR Code-based offline attestation, Bluetooth-based offline attestation, server-side verification services, and the Android application acting as the local verifier.

The experiments were conducted using a workstation with Windows 10 Pro, an Intel® Core™ i7-9750H processor running at 2.60 GHz, 32 GB of RAM, and a Trusted Platform Module (TPM) 2.0. The TPM was configured with 2048-bit RSA Endorsement and Attestation Keys, following the recommendations of the TPM 2.0 specification. The online attestation server, TPM client applications, and QR-Code/Bluetooth attestation components were executed on this platform. For the offline attestation experiments, a Samsung Galaxy S22 Ultra smartphone running Android 16 with 12 GB of RAM was used as the local verifier. This hardware configuration represents a typical enterprise workstation and a modern high-end mobile device, providing a realistic environment for evaluating the performance and practicality of the proposed hybrid attestation architecture. Additionally, each experiment

was executed 25 times under the same hardware and software configuration, and the reported values correspond to the average execution time and memory allocation observed during the experiments.

Figure 8 presents the average execution time of the evaluated workflows. The results indicate that the overall delay varies according to the complexity of each attestation mode and the amount of user interaction required. As expected, the complete user interfaces responsible for orchestrating the attestation process exhibit the highest execution times due to TPM operations, communication establishment, cryptographic processing, user interaction, and graphical interface management.

It is worth observing that the main execution costs come from communication establishment and user interaction rather than TPM processing itself. In particular, QR-Code workflows are affected by image acquisition and decoding, while Bluetooth workflows spend most of their execution time establishing the wireless connection. Once the communication channel is established, TPM quote generation and verification execute within a relatively short period, demonstrating that the proposed security mechanisms impose limited computational overhead.

Figure 9 shows the total memory allocated during the execution of each workflow. The results show that the memory footprint remains suitable across all evaluated components. The Bluetooth Attestation Generation and Bluetooth Pairing Generation exhibited the highest average memory allocation (approximately 36.6 MiB), while the QR-Code pairing procedures presented significantly lower memory requirements, requiring approximately 16 MiB throughout execution, with almost no variation between average and maximum allocation. On the other hand, the online attestation flow allocated approximately 6.9 MiB on average, remaining well below the offline workflows. Finally, the server-side cryptographic operations required less than 4 MiB of average allocated memory.

Therefore, the memory allocation experiments demonstrate that the proposal introduces a small computational overhead while keeping low memory requirements. Even the most demanding workflows need less than 42 MiB of allocated memory, which is suitable for modern desktop and mobile devices. Furthermore, the server-side verification services exhibit very low latency and memory consumption.

From a real-world deployment perspective, these performance results indicate that the proposed hybrid attestation architecture is practical for real-world deployment, since the

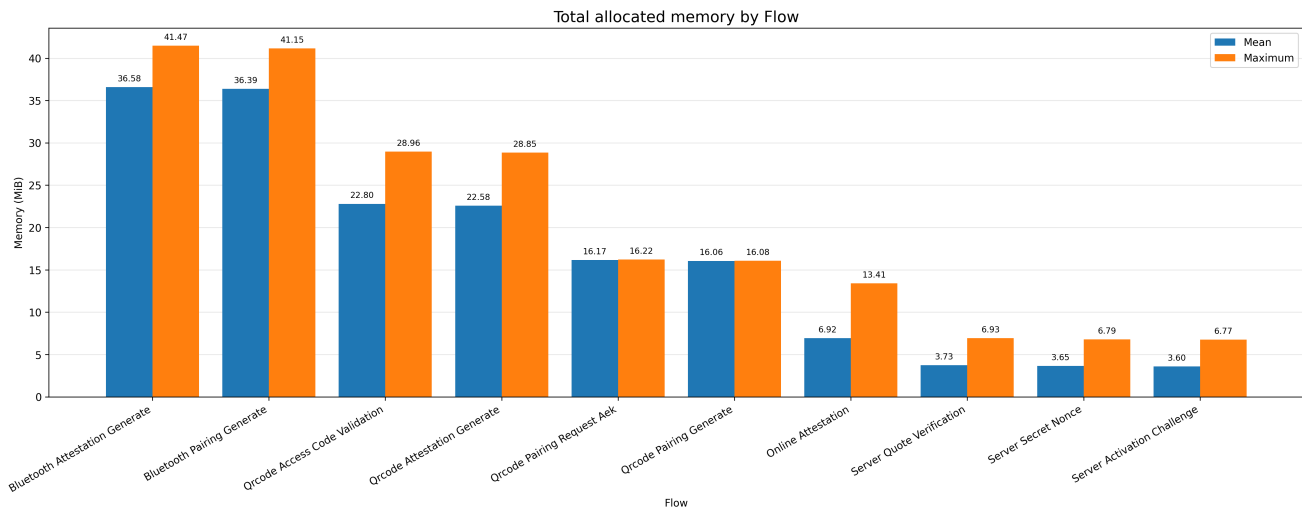


Figure 9. Total Memory Allocation

TPM cryptographic operations and server-side verification introduce only a small fraction of the total execution time, with the dominant overhead originating from communication setup and human interaction. This indicates that the proposed architecture is computationally efficient and that future performance improvements should primarily target communication protocols and user experience rather than the attestation mechanism itself.

7 Final Discussion

The experiments confirmed that both methods (online and offline) are technically feasible and guarantee equivalent levels of security in the verification process. The results demonstrated that:

- Online attestation demonstrated stable performance compatible with cloud-based corporate environments, with low response times and consistent validation of keys and measurements.
- Offline attestation via QR-Code proved simple and practical for end users, suitable for environments without a network;
- Offline attestation via Bluetooth provided an interactive and secure channel, appropriate for scenarios where bidirectional information exchange is desirable.

These proof-of-concept studies demonstrated that the proposed model is functional in real-world environments and offers a reliable and scalable means of hybrid attestation, integrating trusted hardware, mobility, and robust cryptographic mechanisms.

In addition to functional validation, performance observations indicated that online attestation responses were typically completed within a few seconds under standard broadband conditions. Offline QR-code attestation required additional user interaction but remained practical for real-world usage. Bluetooth attestation provided faster interaction due to automated message exchange.

Security robustness was evaluated by altering PCR values to simulate unauthorized modifications. In such cases, the

attestation process correctly denied access, demonstrating the effectiveness of integrity verification.

8 Conclusion

This paper presented a TPM-based attestation mechanism designed to support secure verification of workstations in home environments. By combining online and offline approaches (such as QR-Code and Bluetooth), the proposed solution enables users to validate device integrity before entering their credentials, even in scenarios without reliable network connectivity.

The results obtained from experiments using real TPM hardware and mobile devices show that the approach works in practice and can be used in both connected and offline scenarios. In the case of offline attestation, the use of smartphones as local verifiers proved to be a viable alternative, allowing this type of validation even without a network, while still maintaining the main security guarantees that already exist in TPM-based mechanisms.

In addition to ensuring device integrity, the proposed solution processes platform measurements that may indirectly act as device identifiers. For this reason, implementations should consider data protection aspects and ensure compliance with applicable regulations, such as GDPR and LGPD. In this context, only strictly necessary data should be collected and stored, following principles of minimization and controlled retention.

As future work, we intend to evaluate other approaches for TPM, expanding the pool of security solutions that can be deployed by the tool and, consequently, improving the security level of the companies.

Acknowledgements

The authors would like to thank the CNPq (N° 305946/2025-0) for the financial support.

Funding

The authors would like to thank the CNPq (N° 305946/2025-0) for

the financial support.

Authors' Contributions

All authors contributed to the development of the proposal and the writing of this manuscript.

Competing interests

The authors declare no conflicts of interest.

Availability of data and materials

The code developed, as well as the data used in the experiments, will be available upon request with the necessary instructions for reproducibility.

References

- Achemlal, M., Gharout, S., and Gaber, C. (2011). Trusted platform module as an enabler for security in cloud computing. In *2011 Conference on Network and Information Systems Security*, pages 1–6. DOI: 10.1109/SAR-SSI.2011.5931361.
- Arfaoui, G., Jacques, T., Lacoste, M., Onete, C., and Robert, L. (2023). Towards a privacy-preserving attestation for virtualized networks. *IACR Cryptol. ePrint Arch.*, 2023:735. DOI: 10.1007/978-3-031-51482-1_18.
- Brito, M. L. L., Ferreira, M. C. M., Portela, A. L. C., and Gomes, R. L. (2026). Ai-based estimation of bandwidth availability for data offloading in edge-cloud computing. *IEEE Networking Letters*, 8:69–73. DOI: 10.1109/LNET.2025.3614770.
- Brooks, C. (2022). *Global Thought Leader in Cybersecurity and Emerging Tech*. Web site Forbes. Book.
- Challener, D., Yoder, K., Catherman, R., Safford, D., and Doorn, L. V. (2008). *A Practical Guide to Trusted Computing*. Book.
- Cheng, J., Zhang, K., and Tu, B. (2021). Remote attestation of large-scale virtual machines in the cloud data center. In *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 180–187. DOI: 10.1109/TrustCom53373.2021.00041.
- Coker, G., Guttman, J., Loscocco, P., Herzog, A., Millen, J., O'Hanlon, B., Ramsdell, J., Segall, A., Sheehy, J., and Sniffen, B. (2011). Principles of remote attestation. *Int. J. Inf. Secur.*, 10(2):63–81. DOI: 10.1007/s10207-011-0124-7.
- Costa, M. A., Costa, Y. M., Almeida, Y. O., Cardoso, F. J., and Gomes, R. L. (2024). 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. Association for Computing Machinery. DOI: 10.1145/3685323.3685331.
- Davi, L., Sadeghi, A.-R., and Winandy, M. (2009). Dynamic integrity measurement and attestation: towards defense against return-oriented programming attacks. page 49–54. DOI: 10.1145/1655108.1655117.
- Death, D. (2017). *Information Security Handbook*. Packt. Book.
- Du, R., Pan, W., and Tian, J. (2018). Dynamic integrity measurement model based on vtpm. *China Communications*, 15(2):88–99. DOI: 10.1109/CC.2018.8300275.
- Gomes, R. L., Bittencourt, L. F., Madeira, E. R. M., Cerqueira, E. C., and Gerla, M. (2016). Software-defined management of edge as a service networks. *IEEE Transactions on Network and Service Management*, 13(2):226–239. DOI: 10.1109/TNSM.2016.2538821.
- Hosseinazadeh, S., Sequeiros, B., Inácio, P. R., and Leppänen, V. (2020). Recent trends in applying tpm to cloud computing. *Security and privacy*, 3(1):e93. DOI: 10.1002/spy2.93.
- Kinney, S. L. (2006). *Trusted Platform Module Basics: Using TPM in Embedded Systems*. Newnes, USA. Book.
- Kirmani, M. S. and Banday, M. T. (2024). Exploring firmware-based anti-forensics in iot devices: Techniques and implications. *SN Computer Science*, 5(8):1–23. DOI: 10.1007/s42979-024-03476-y.
- Li, Y. and Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. *Energy Reports*, 7:8176–8186. DOI: 10.1016/j.egyr.2021.08.126.
- Menezes, R. A., Araujo, R. S., Rodrigues, L. S., Nascimento, E. S., and Gomes, R. L. (2026). Data protection through the integration of tpm and cryptography. *International Journal of Wireless and Microwave Technologies(IJWMT)*, 16(1):37–49. DOI: 10.5815/ijwmt.2026.01.03.
- Mohamed, N. and Ahmed, A. A. (2024). Ai in combatting man-in-the-middle attacks: A comprehensive review. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE. DOI: 10.1109/iccant61001.2024.10725789.
- Nobre, F. V. J., Alves, D. O., Araujo, R. S., Campos, G. A., and Gomes, R. L. (2026). Risk classification of ip addresses using machine learning with weighted voting approach. In Rodrigues, L. A. and Oliveira, R., editors, *Dependable and Secure Computing*, pages 320–328, Cham. Springer Nature Switzerland. DOI: 10.1007/978-3-032-11539-3_19.
- Nobre, F. V. J., Silva, D. d. S., Ferreira, M. C. M. M., Brito, M. L. M. L., de Araújo, T. P., and Gomes, R. L. (2025). Time-weighted correlation approach to identify high delay links in internet service providers. *Journal of Internet Services and Applications*, 16(1):419–430. DOI: 10.5753/jisa.2025.5218.
- Oliver, I. (2021). Trust, security and privacy through remote attestation in 5g and 6g systems. In *2021 IEEE 4th 5G World Forum (5GWF)*, pages 368–373. DOI: 10.1109/5GWF52925.2021.00071.
- Pimenta, I. A., Lee, M. H., Bittencourt, L. F., and Gomes, R. L. (2026). Adaptive privacy based on mutual information for machine learning in edge-cloud environments. *IEEE Networking Letters*, 8:49–53. DOI: 10.1109/LNET.2025.3607555.
- Portela, A., Linhares, M. M., Nobre, F. V. J., Menezes, R., Mesquita, M., and Gomes, R. L. (2024). The role of tcp congestion control in the throughput forecasting. In *Proceed-*

- ings of the 13th Latin-American Symposium on Dependable and Secure Computing, LADC '24, page 196–199, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3697090.3699869.
- Proudler, G., Chen, L., and Dalton, C. (2014). *Trusted Computing Platforms - TPM2.0 in Context*. DOI: 10.1007/978-3-319-08744-3.
- Ryu, C., Lee, J.-H., Kim, D.-H., Lee, H.-S., Kim, Y.-S., Han, J.-H., and nyeo Kim and, J. (2024). A comprehensive survey of tpm for defense systems. *KSII Transactions on Internet and Information Systems*, 18(7):1953–1967. DOI: 10.3837/tiis.2024.07.012.
- TCG (2019). Trusted computing group: Trusted platform module library specification, family 2.0. Available at:<https://trustedcomputinggroup.org/resource/tpm-library-specification/>.
- TCG (2020). Trusted platform module library - part 1: Architecture. Available at:https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM2_r1p59_Part1_Architecture_pub.pdf.
- TCG (2025). TCG Guidance on Integrity Measurements and Event Log Processing. Available at:https://trustedcomputinggroup.org/wp-content/uploads/TCG-Guidance-on-Integrity-Measurements-and-Event-Log-Processing-Version-1.0-RC-1_24February25.pdf.
- (TCG), T. C. G. (2025a). TPM 2.0 Keys for Device Identity and Attestation. Available at:<https://trustedcomputinggroup.org/resource/tpm-2-0-keys-for-device-identity-and-attestation>.
- (TCG), T. C. G. (2025b). Trusted Platform Module Library Specification. Available at:<https://trustedcomputinggroup.org/tpm-library-specification>.
- Vardakis, G., Hatzivasilis, G., Koutsaki, E., and Papadakis, N. (2024). Review of smart-home security using the internet of things. *Electronics*, 13(16):3343. DOI: 10.3390/electronics13163343.
- Wang, J., Shi, Y., Peng, G., Zhang, H., Zhao, B., Yan, F., Yu, F., and Zhang, L. (2016). Survey on key technology development and application in trusted computing. *China Communications*, 13(11):70–90. DOI: 10.1109/CC.2016.7781720.
- Won, Y.-S. and Bhasin, S. (2021). Are cold boot attacks still feasible: A case study on raspberry pi with stacked memory. In *2021 Workshop on Fault Detection and Tolerance in Cryptography (FDTC)*, pages 56–60. IEEE. DOI: 10.1109/fdte53659.2021.00017.
- Zeitouni, M., Santos, M., and Gomes, R. (2024). Melhorias no processo de armazenamento de dados em tpm para gerenciamento de integridade. In *Anais Estendidos do XXIV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais*, pages 270–278. SBC. DOI: 10.5753/sbseg_estendido.2024.243344.