

Received 21 April 2025, accepted 28 April 2025, date of publication 1 May 2025, date of current version 9 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3565945

## RESEARCH ARTICLE

# Enhancing Property-Based Token Attestation With Homomorphic Encryption (PTA-HE) for Secure Mobile Computing

THINH LE VINH<sup>1</sup>, HUAN THIEN TRAN<sup>2</sup>, AND SAMIA BOUZEFRANE<sup>3</sup>

<sup>1</sup>Faculty of Information Technology, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City 700000, Vietnam

<sup>2</sup>Faculty of Engineering and Technology (FET), Saigon University, Ho Chi Minh City 700000, Vietnam

<sup>3</sup>CEDRIC Laboratory, Conservatoire National des Arts et Métiers (Cnam), 75016 Paris, France

Corresponding author: Thinh Le Vinh (thinhlv@hcmute.edu.vn)

**ABSTRACT** This paper proposes PTA-HE, an enhanced Property-based Token Attestation scheme integrated with Homomorphic Encryption (HE), specifically designed to address critical security challenges in mobile cloud computing environments. Traditional Property-based Token Attestation (PTA) protocols, although foundational, inherently lack robust mechanisms to secure sensitive data during active processing stages, exposing data to potential confidentiality and integrity breaches. Our main contributions are: the introduction of PTA-HE, which resolves these vulnerabilities by enabling computations directly on encrypted data, ensuring continuous protection and resilience against unauthorized access and manipulation; a strategic employment of Trusted Third Parties (TTPs) for secure attestation management, leveraging HE to maintain data confidentiality throughout the entire attestation workflow; rigorous experimental evaluations quantifying computational overhead, communication costs, latency, and scalability implications, transparently illustrating the performance trade-offs associated with enhanced security; and formal verification using the Scyther tool demonstrating PTA-HE's superior correctness and robustness against multiple security threats, such as replay and man-in-the-middle attacks. Consequently, PTA-HE provides a highly effective and practical solution for secure mobile computing applications requiring stringent assurances of data privacy and integrity.

**INDEX TERMS** Property-based token attestation, homomorphic encryption, mobile cloud computing, security, privacy, Scyther, verification.

## I. INTRODUCTION

Currently, the proliferation of mobile devices and their integration with cloud computing services have introduced new security challenges. Ensuring the confidentiality, integrity, and authenticity of data in transit and at rest is paramount [1], [2], [3]. Trusted Platform Modules (TPMs) play a crucial role in this context by providing hardware-based security functions that protect sensitive information and enable secure cryptographic operations [4], [5], [6]. Attestation, a process facilitated by TPMs, allows a device to prove its trustworthiness to a remote verifier by providing cryptographic evidence of its software and hardware state [7], [8]. This process is vital

for securing mobile cloud environments where devices continuously interact with cloud servers and exchange sensitive data. However, traditional attestation mechanisms often fall short in maintaining data confidentiality during processing, particularly in scenarios where data must be processed by third parties. This is where Homomorphic Encryption (HE) becomes essential. HE allows computations to be performed directly on encrypted data without revealing the underlying plaintext, ensuring data remains secure even when processed by untrusted entities [9], [10], [11]. The integration of HE into attestation protocols represents a significant advancement in protecting sensitive information from potential breaches, enabling secure mobile cloud computing. This paper introduces an enhanced Property-based Token Attestation scheme (PTA-HE) that combines the strengths of TPM-based

The associate editor coordinating the review of this manuscript and approving it for publication was Sedat Akleyek<sup>1</sup>.

attestation with the advanced security capabilities of HE. The PTA-HE protocol is designed to address the shortcomings of traditional PTA schemes by enabling secure computations on encrypted data, thereby enhancing both security and privacy in mobile cloud environments. Before presenting the details of PTA-HE, we first review the traditional PTA scheme [12] and discuss the rationale behind choosing HE to enhance PTA [13].

Property-Based Token Attestation [12] is a security mechanism designed to verify the integrity and trustworthiness of mobile devices in cloud computing environments, particularly under the Bring Your Own Device (BYOD) policy. This approach leverages the functionalities of the TPM, which is a hardware-based security module that enhances security through encryption, secure storage, and integrity measurements. PTA uses these capabilities to generate a unique attestation token for each device, which serves as proof of its secure state. The PTA mechanism addresses the limitations of traditional attestation methods like Binary Attestation, which exposes detailed hardware and software configurations, potentially compromising privacy and flexibility. Instead, PTA focuses on verifying abstract properties of a device or application, ensuring that it meets specific security requirements without revealing sensitive internal configurations. This method provides a more flexible and privacy-preserving approach to remote attestation, enhancing the overall security posture of mobile devices operating in cloud environments. In PTA (see Figure 1), the Employee must obtain a valid Token from a TTP to access the company's resources. To acquire this Token, the Employee is required to present secure evidence of their hardware and software platform to the TTP, which provides cloud-based security services. After assessing the provided evidence, the TTP issues a Token based on the validity of the received information. This Token functions similarly to a Letter of Introduction, verifying the employee's credentials. According to the internal policies of the enterprise, the TTP can also offer proof of existence for a Token at a specific moment in time. The Token is only valid for a limited period as defined by the TTP; beyond this timeframe, it will be automatically invalidated. With this Token, the Employee gains access to specific company data for a designated period. The role of the Enterprise is to verify the Token to grant the Employee access to its resources, effectively outsourcing its security management to the TTP. This security delegation can be facilitated by a Cloud platform offering Security as a Service, which may be a private cloud, public cloud, or hybrid cloud environment.

Additionally, in PTA mechanism, a property is utilized to describe the behavior of a platform or program without exposing its underlying configuration. The definition of a property to be attested covers a broad scope, allowing anything related to the platform or application to be considered a property. In the PTA scheme, abstract properties of program classes are attested to establish security requirements. For instance, an application might include three classes: Class A, with

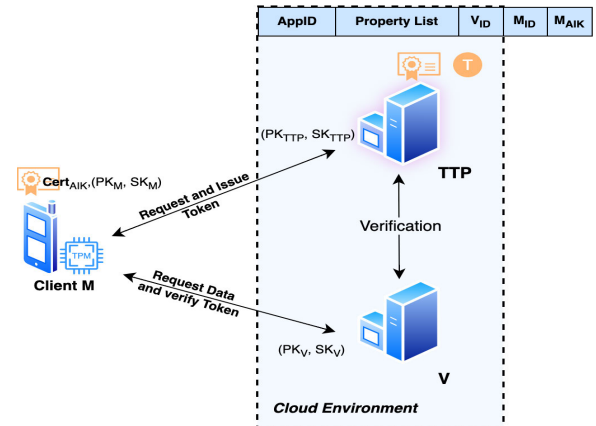


FIGURE 1. Property based token attestation.

properties defined by  $(\alpha_1, \dots, \alpha_N) \in \alpha$ ; similarly, Classes B and C have properties  $\beta$  and  $\gamma$ , respectively. The active TTP is aware of these properties, maintaining a property list  $(PL) = (\alpha, \beta, \gamma)$ . A subset of these properties,  $\Omega$ , drawn from  $(\alpha \cup \beta \cup \gamma)$ , is defined according to a set of security levels,  $SP = \text{Low, Med, Hi}$ , as outlined by the enterprise's security policy. Based on the specified security level in  $SP$ , the application service generates  $\Omega$ . As a result, a dynamic property list  $P_i$  is created at runtime as  $P_i = \frac{PL}{\Omega}$ . The integrity of  $P_i$  is maintained and verified by a reserved Platform Configuration Register (PCR) and the active TTP, respectively. Additionally, a recomputed nonce, generated using the Diffie-Hellman exponentiation function, is employed as an additional security property. Upon receiving and verifying the evidence from the client, the TTP signs a trust credential to certify the application's trustworthiness. This trust credential serves as a trust Token for client authentication, consisting of various proofs such as  $P_i$ , a timestamp, and the recomputed nonce, to validate the client's trustworthiness to the enterprise.

Building on this foundation, the role of the Token (Tk) becomes crucial in the attestation process. Tk serves as a trusted proof that allows an employee (M) to access secure resources. This Token is issued by the TTP only if the employee's hardware and software platform evidence successfully pass the attestation process. As previously discussed, Tk includes a security level that is determined based on a random property list ( $P_i$ ). It is assumed that, for a specific application, the software provider (TTP) maintains a comprehensive list of application properties (PL), which can be categorized into different class levels. The application also includes a special service that selects and generates a random property list for the mobile client. For instance, as shown in Figure 2, the PL contains twenty-six properties (a to z) of the application, and the number of matched properties ( $P_i$ ) corresponds to a security level, with a higher number indicating a higher level of security. Let  $p$  represent a property;  $M_1$  and  $M_2$  are clients with their valid  $P_i$  lists, whose integrity is successfully verified. However,  $M_3$ , with a compromised  $P_i$ , would be rejected. The attestation

conditions are outlined accordingly.

$$\begin{aligned}
 &P_i \subset PL | P_i f = 0 \\
 &\text{with } p \in P_i \& p \in PL \\
 &\text{If } (\forall p \in (P_i \cup PL)) \text{ Then Pass} \\
 &\text{If } (\forall p \in (P_i \& \exists p \in /PL)) \text{ Then Fail}
 \end{aligned}$$

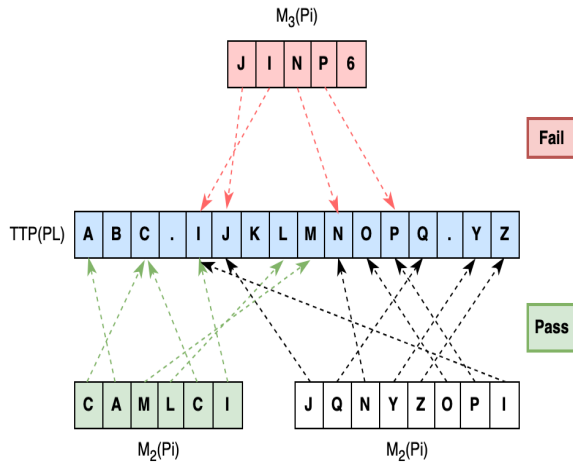


FIGURE 2. Property evidence.

Building on the discussion of Tokens (Tk) in the PTA scheme, it is clear that while the current approach provides a basic level of security by verifying hardware and software properties, there are inherent vulnerabilities during the processing of attestation data. Although the Token issued by the TTP serves as a proof of trustworthiness based on verified properties, the confidentiality of the data and the integrity of the attestation process itself remain at risk if the data is exposed during computations or verifications. Figure 3, which is a high-level architecture, shows the communication among Mobile device, TTP and Cloud.

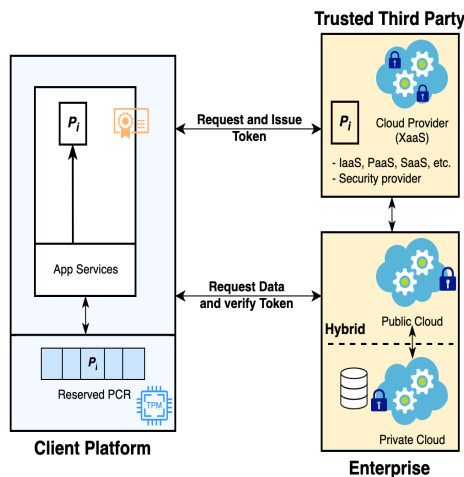


FIGURE 3. Property based token attestation model.

To address these vulnerabilities, the integration of HE into the PTA framework becomes critically important. HE allows for computations to be performed directly on encrypted

data without ever needing to decrypt it, thereby preserving the confidentiality and integrity of sensitive information throughout the entire attestation process. This capability is particularly valuable in situations where the data involved includes sensitive properties or credentials, which, if exposed, could undermine the security of the entire system.

Incorporating HE into the PTA scheme enhances the security posture of the system by ensuring that even if data is intercepted during transmission or accessed by a compromised party, the underlying information remains protected and inaccessible. This integration effectively overcomes the limitations of traditional PTA, which focuses primarily on securing data at rest or in transit, without offering equivalent protection during data processing. HE provides a comprehensive solution, ensuring that all stages of data handling, including the computation and verification of Tokens, are conducted securely, minimizing the risk of unauthorized access or data breaches. Therefore, the use of HE in PTA is not just an enhancement but a necessary evolution of the protocol to meet the growing demands of security in increasingly complex and high-risk environments. By enabling secure computations on encrypted data, HE fortifies the overall security architecture of PTA, making it a more robust and future-proof solution that effectively safeguards sensitive information against both current and emerging threats. The increasing integration of mobile devices with cloud services necessitates advanced security protocols capable of protecting sensitive data not only in transit or storage but critically during active processing phases. Existing attestation approaches, while providing basic security assurances, often fail to protect data adequately under active use, presenting significant vulnerabilities in contemporary mobile cloud computing scenarios. Our work explicitly addresses these shortcomings by integrating Homomorphic Encryption into the attestation process, presenting a novel protocol that significantly enhances data confidentiality and integrity. The proposed PTA-HE protocol not only strengthens the theoretical foundation of attestation security but also provides clear practical benefits and actionable insights for secure real-world deployments.

The remainder of this paper is structured as follows: Section II surveys Related Work, positioning the PTA-HE within the wider landscape of research in mobile cloud security. Section III introduces the Proposed Enhanced PTA-HE Scheme, explaining the integration of HE into the existing PTA framework and highlighting the architectural and procedural improvements. Section IV offers a thorough Security Analysis, assessing the resilience of the enhanced scheme against a range of potential threats and attacks. Section V covers the Performance Evaluation, analyzing the computational and communication overheads, latency, and scalability of the PTA-HE scheme, and includes an in-depth discussion on the balance between performance and security. To verify the correctness of the protocol, Session 6 will present the validation process using the Scyther tool. Lastly, Section 7 concludes the paper by summarizing our

findings and suggesting future research directions, including potential optimizations of the protocol and an exploration of its applicability in more complex and diverse environments.

## II. RELATED WORKS

The security of mobile cloud computing has garnered significant attention in the research community, particularly with the integration of mobile devices and cloud services. Various studies have proposed solutions to address the critical issues of data confidentiality, integrity, and secure computations. Muheidat and Tawalbeh [1] provide a broad survey on mobile and cloud computing security, emphasizing the importance of robust cryptographic techniques to mitigate risks in data transmission and storage. Similarly, Ari et al. [3] focus on the Cloud of Things (CoT), advocating for layered security frameworks that combine hardware-based security modules with advanced encryption techniques to protect sensitive data. HE is a prominent tool for preserving data privacy while allowing computations on encrypted data. Gong et al. [9] conduct a comprehensive survey on Fully Homomorphic Encryption (FHE) acceleration methods, categorizing existing schemes from algorithmic and hardware perspectives and suggesting future research directions. In terms of TPM the authors in [14] proposes a TPM-based conditional privacy-preserving authentication protocol (T-CPPA) for vehicle ad-hoc networks (VANETs). The protocol ensures message integrity and authenticity by using a TPM to generate pseudonyms and signature keys. It also employs a cluster-based model for message similarity calculation, enhancing system stability and efficiency. In [15] and [16], the authors extends remote attestation schemes by introducing ERAMO, a protocol that uses memory offloading to attest larger memory regions in multi-service IoT devices. Ameer et al. [10] further explore the use of HE in multi-cloud environments to secure data sharing and processing.

In addition to HE, various cryptographic schemes have been developed to enhance data security in edge computing and IoT environments. Mahato and Chakraborty [17] review multiple cryptographic schemes, such as identity-based encryption and searchable encryption, highlighting their role in securing edge computing. Zhang and Wang [18] propose a hybrid encryption approach combining symmetric and asymmetric cryptography for secure data transmission in IoT devices, aiming to optimize throughput and reduce execution time. In cloud environments, Li et al. [19] proposes an efficient encryption scheme with verifiable outsourced decryption to reduce computational overhead while maintaining data confidentiality, offering a benchmark for evaluating encryption methods in mobile cloud computing.

Attestation mechanisms are critical for establishing trust in cloud and IoT environments. Banks et al. [20] provide an extensive review of remote attestation techniques, discussing their applications and limitations. Yuan et al. [21] introduce a TEE-based virtual remote attestation scheme for virtual network functions, designed to reduce overhead and enhance

security. In a similar context, Wang et al. [22] develop a dynamic homomorphic secret sharing scheme for additive computation, which supports dynamic server reissuance, thereby reducing computational overhead. Other studies, such as Shang et al. [23], propose novel encryption methods like Obfuscated Searchable Symmetric Encryption (OSSE) to enhance data security by obfuscating access patterns.

Furthermore, several studies address scalability and resilience in attestation protocols. Ammar et al. [24] present slimIoT, a scalable lightweight attestation protocol for IoT networks that leverages broadcast authentication and symmetric key cryptography. Kohnhäuser et al. [16] introduce a scalable attestation protocol resilient to physical attacks, designed for interconnected embedded devices. Stumpf et al. [25] propose improvements in the scalability of platform attestation through protocols that enable fast and secure integrity reporting. These works collectively emphasize the need for advanced cryptographic techniques and scalable security frameworks, which align with the goals of the proposed PTA-HE scheme to enhance data confidentiality and integrity during processing in secure mobile cloud computing environments.

In recent years, significant advancements have been made in privacy-preserving techniques for querying outsourced encrypted data, particularly in mobile and cloud computing environments. Miao et al. [26] introduced an efficient privacy-preserving spatial range query (PSRQ) scheme that combines the Geohash algorithm with Circular Shift and Coalesce Bloom Filters, enabling secure and efficient spatial queries over encrypted data. In the realm of mobile e-health clouds, the authors [27] proposed a time-controllable keyword search scheme with efficient revocation, utilizing attribute-based comparable access control to enhance data security and access flexibility. Li et al. [28] developed the Efficient Privacy-Preserving Location-based Query (EPLQ) system, which employs a predicate-only encryption scheme for inner product range queries, facilitating privacy-preserving spatial range queries over outsourced encrypted data. Additionally, the paper [29] proposes a secure system for querying encrypted databases in the cloud using an indexing scheme and AES-CBC encryption. It efficiently narrows down search results with bit vectors, ensuring privacy and reducing computational overhead. The system outperforms existing solutions like CryptDB in terms of execution time and space requirements. Furthermore, Zheng et al. [30] introduced SecSkyline, a system for fast privacy-preserving skyline queries over encrypted cloud databases, balancing security and efficiency in multi-criteria decision-making applications. These contributions collectively enhance the landscape of secure data retrieval and processing in cloud-based systems.

In the realm of digital security, security protocols, particularly key agreement protocols, are fundamental to ensuring secure communications over potentially insecure networks [31], [32]. These protocols enable the establishment



of shared secret keys, which are crucial for safeguarding subsequent exchanges of data. Given the critical role these protocols play, verifying their robustness is essential to prevent vulnerabilities such as man-in-the-middle, replay attacks, and side-channel exploits. Recent studies have explored the analysis and verification of hierarchical identity-based authenticated key agreement (HIB-AKA) protocols, employing advanced tools like Scyther and Tamarin for a thorough evaluation [33]. Research has also investigated identity-based authenticated key agreement protocols within the Diffie-Hellman family, enabled by Weil or Tate pairings, to address intricate issues related to cryptographic security [34]. Moreover, the rise of quantum computing has led to a focus on lightweight lattice-based secure systems, especially those that offer efficient security in the post-quantum era [35]. Scyther is one of the leading tools in security protocol verification, recognized for its intuitive interface and capacity to automate the falsification of security protocols. Its ability to handle an unbounded number of protocol sessions makes it highly effective for analyzing key agreement protocols across various contexts, including cloud computing and IoT environments [36], [37], [38].

### III. PROPOSED ENHANCED PTA-HE SCHEME

In this section, we present an enhanced Property-based Token Attestation scheme, integrating Homomorphic Encryption (PTA-HE) to bolster the security framework for mobile cloud computing environments. This extension is motivated by the need for stronger confidentiality and integrity guarantees in federated learning scenarios, where sensitive data must be processed without exposing plaintext information. By incorporating HE, we enable secure computations on encrypted data, addressing limitations in traditional PTA protocols and enhancing overall security.

#### A. ARCHITECTURE

The architecture of the enhanced PTA scheme leverages the integration of TTP, cloud servers, and mobile devices. The architecture also includes the Attestation Authority (AA) and Homomorphic Encryption Engine. The former is a trusted entity responsible for issuing attestation tokens and verifying the integrity of mobile devices to ensure compliance with security policies before granting access to cloud resources. Additionally, the AA issues certificates for the Attestation Identity Key (AIK), ensuring that the AIK is bound to a legitimate TPM. This process involves verifying the identity of the TPM and issuing credentials or certificates that confirm the AIK's trustworthiness, including cryptographic endorsements to ensure the AIK is generated by a valid and recognized TPM. Notably, to streamline operations and avoid redundancy in the PTA processes, we assume that the AA entity is available and shared among other entities. The keys of the AA are assumed to be readily accessible, and we use the AA's keys instead of the AIK. This approach simplifies the architecture and enhances efficiency in implementing the enhanced PTA with HE. The latter is a software module

integrated within both the cloud servers and mobile devices to enable secure computations on encrypted data. This engine ensures that sensitive data remains encrypted throughout the attestation and computation processes. This system, Enhancing PTA-HE, is designed to ensure robust security and privacy for mobile cloud computing environments. In addition, PTA-HE explicitly integrates TPM hardware on mobile devices, enabling secure storage of cryptographic keys and accelerated cryptographic operations. This integration significantly reduces public-key encryption overhead, ensuring that the mobile devices perform minimal computational tasks. The enhanced PTA-HE scheme architecture consists of the following critical components (Figure 4).

#### Trusted Third Party (TTP)

The TTP serves as an intermediary in the attestation process, verifying the legitimacy of mobile devices and issuing attestation tokens. It securely manages the certificates and attestation keys for devices and ensures compliance with security policies before granting access to cloud resources.

#### Cloud Servers

Cloud servers act as Verifiers in the attestation process. They host the Homomorphic Encryption engine, which enables secure computations on encrypted data received from mobile devices. The HE engine ensures that sensitive data remains encrypted during processing, allowing for privacy-preserving data analysis and decision-making. The Verifier also validates attestation tokens to determine resource access eligibility.

#### Mobile Devices

Mobile devices initiate requests for attestation tokens and perform operations requiring attestation. They generate public and private keys using the TPM, encrypt their state information with the Verifier's public key, and send this encrypted data to the cloud servers for secure processing. This process ensures that sensitive data remains protected throughout the attestation.

This architecture is designed to facilitate secure and efficient attestation in a mobile cloud computing environment, mitigating various security threats.

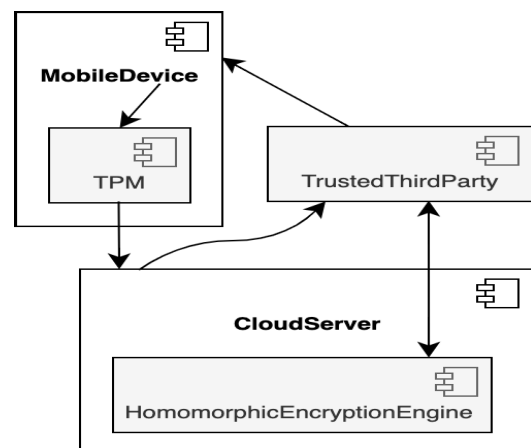


FIGURE 4. Enhanced PTA scheme architecture.

## B. PROTOCOL DESIGN

The PTA-HE protocol strategically delegates heavy computational tasks, including homomorphic evaluation and complex cryptographic verification, to cloud-based entities. The mobile device primarily handles lightweight encryption of state information and digital signature generation, ensuring minimal computational overhead. In this part, the protocol design of PTA-HE is presented to clearly demonstrate that the enhanced PTA-HE enables secure computations on encrypted data, which enhances confidentiality and integrity throughout the attestation process (See Figure 05). In general, the protocol unfolds through the following steps.

### Initial Attestation Request

Firstly, the mobile device (M) generates a pair of public and private keys using the TPM. The public key (pk) is shared with the TTP and the Verifier (cloud server), while the private key (sk) is kept secure within the TPM:

$$\text{KeyGen}(\text{TPM}) \rightarrow (pk, sk) \quad (1)$$

This step ensures that each device has a unique cryptographic identity.

### Device Registration

The mobile device registers with the TTP by providing its public key (pk) and a device identifier ( $M_{ID}$ ). TTP verifies the device's integrity and securely stores the public key:

$$\text{Register}(pk, M_{ID}) \rightarrow \text{Success} \quad (2)$$

This step establishes trust between the device and the TTP.

### Request for Attestation

When a mobile device requests access to cloud resources, it initiates an attestation request by signing a nonce (N) with its private key (sk) and sending the signed nonce ( $\sigma$ ) to the TTP:

$$\sigma = \text{Sign}(sk, N) \quad (3)$$

The signed nonce ( $\sigma$ ) ensures the authenticity and integrity of the attestation request, allowing TTP to validate the request before processing.

### Data Encryption and Upload

The mobile device encrypts its state information (e.g., software versions, configurations) using the public key ( $pk_{TTP}$ ) of TTP.

$$C = \text{Enc}(pk_{TTP}, \text{State}) \quad (4)$$

where, C represents the ciphertext of the encrypted state information, ensuring that sensitive data remains confidential during transmission and processing. The encrypted data is then uploaded to the cloud server (Verifier).

### Homomorphic Computation

The Verifier performs homomorphic computations on the encrypted data without decrypting it. HE allows users to process data in the cloud without ever exposing it to cloud service provider. This ensures that sensitive information always remains confidential, even while being processed.

$$\text{EncryptedResult} = \text{Eval}(f, C) \quad (5)$$

where  $f$  represents the function performed on the encrypted data, ensuring that computations are securely executed without exposing plain text data.

### Secure Forwarding of Results

After the homomorphic evaluation, the Verifier forwards the encrypted result directly to the TTP for final verification. The Verifier signs this encrypted result with its own private key ( $sk_V$ ), along with identifiers and timestamps for integrity and authenticity verification by TTP.

### Decryption and Verification

Upon receiving the encrypted results, the TTP securely decrypts this data using its own secure private key

$$\text{State} = \text{Dec}(sk_{TTP}, \text{EncryptedResult}) \quad (6)$$

The TTP then clearly evaluates this decrypted state information, checking for compliance with defined security policies.

### Token Issuance

If the device's state meets all required security policies, the TTP generates an unsigned Token that includes verified attestation information. The Token included  $M_{ID}$ ,  $PCR$ ,  $Properties$ ,  $Nonce$ ,  $Timestamp$ , and  $VerificationResult$ .

The TTP then digitally signs this Token using its private key ( $sk_{TTP}$ ), resulting in a cryptographically secure, signed attestation token  $\tau$ . The signed token ( $\tau$ ) serves as proof of successful attestation, indicating that the device has been verified and is trustworthy.

$$\tau = \text{Sign}(sk_{TTP}, \text{Token}) \quad (7)$$

### Token Presentation

The mobile device presents the attestation token ( $\tau$ ) to the cloud server (Verifier) to gain access to the requested resources. The Verifier verifies the token using the TTP's public key ( $pk_{TTP}$ )

$$\text{Verify}(pk_{TTP}, \tau) \rightarrow (\text{True}/\text{False}) \quad (8)$$

This verification ensures that the token is valid and issued by the Trusted Third Party, thereby granting access to the mobile device.

### Access Granted

Upon successful verification of the token, the mobile device is granted access to the cloud resources.

## C. PTA-HE MESSAGE EXCHANGE

This section details the secure communication processes between the mobile device, the TTP, and the Verifier. By utilizing Homomorphic Encryption, the PTA-HE protocol ensures that sensitive data can be processed without being decrypted, thereby maintaining confidentiality and integrity throughout the attestation process. Before discussing the structure of the proposed protocol, we recall the following notations that have been used in this article.

- $M_{ID}$ : A unique identifier of the mobile device, ensuring its identity in the protocol.
- $N_B, N_M, N_{V1}, N_K, N_V, N_{TTP}$ : Nonces generated at various stages to ensure freshness and prevent replay attacks.

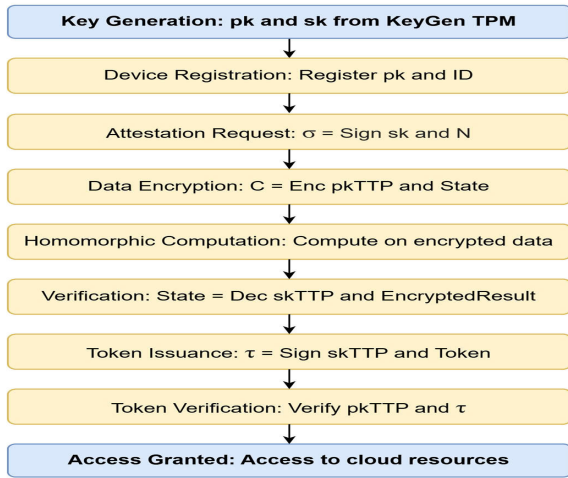


FIGURE 5. System design of PTA-HE.

- $T_M, T_J, T_{TP}$ : Timestamps generated at different stages to ensure the exact timing of requests and responses.
- $pk_V$ : The public key of the Verifier, used for encrypting data to ensure confidentiality.
- $sk$ : The private key of the mobile device, used for signing data to ensure integrity and authenticity.
- $sk_{TTP}$ : The private key of the TTP, used for signing data to ensure integrity and authenticity.
- $PCR$ : Platform Configuration Register, containing measurements of the software and hardware configuration of the mobile device.
- $P_S$ : A signed message from the mobile device, containing critical information that needs to be verified.
- $f$ : A function performed on encrypted data using Homomorphic Encryption.

#### Message Exchange Flows

##### Step 1: Initial Attestation Request.

The mobile device ( $M$ ) initiates the attestation process by sending an attestation request to  $TTP$ . This request comprises a hash of the mobile device's unique identifier ( $M_{ID}$ ), two nonces ( $N_B, N_M$ ) generated by the mobile device to ensure freshness and uniqueness, and a timestamp ( $T_M$ ) indicating when the request was generated. The mobile device signs this message using its private key ( $sk$ ) to ensure integrity and authenticity. This signed message is then encrypted using the Verifier's public key ( $pk_V$ ) to guarantee authenticity and integrity.

##### Step 2: Forwarding Nonce to Verifier

The TTP forwards a nonce ( $N_{V1}$ ) to the Verifier ( $V$ ). This message includes: a hash of the mobile device's ID ( $M_{ID}$ ), and a nonce ( $N_{V1}$ ) generated by the TTP to ensure freshness. The message is signed with the TTP's private key ( $sk_{TTP}$ ) and encrypted using the Verifier's public key ( $pk_V$ ). Homomorphic Encryption ensures that operations on the nonce and other sensitive data can be performed without exposing plaintext data.

##### Step 3: Encrypted State Information

The mobile device encrypts its sensitive state information using TTP's public key ( $pk_{TTP}$ ), preserving confidentiality during transmission. This encrypted state information ( $C$ ) is then sent directly to the Verifier.

##### Step 4: Homomorphic Computation

The Verifier performs computations on the encrypted data using Homomorphic Encryption, allowing it to process data without decrypting it. This ensures that sensitive information remains confidential throughout the computation process.

1. *Key Generation (KeyGen)*. The Verifier generates a pair of keys, including a public key ( $pk_V$ ) and a private key for decryption.

$$(pk_V, sk_V) = \text{KeyGen}() \quad (9)$$

2. *Encryption (Enc)*. The mobile device encrypts its state information ( $State$ ) with the Verifier's public key:

$$C = \text{Enc}(pk_V, State) \quad (10)$$

3. *Evaluation (Eval)*. The Verifier uses the Eval algorithm to compute a function  $f$  on the encrypted data.

$$\text{Eval}(f, C) = \text{Enc}(pk_V, State) \quad (11)$$

More generally, for a homomorphic function  $f$ :

$$\begin{aligned} \text{Eval}(f, C_1, C_2, \dots, C_N) \\ = \text{Enc}(pk_V, f(State_1, State_2, \dots, State_N)) \end{aligned} \quad (12)$$

The above formula shows that the Verifier can perform operations like addition or multiplication directly on encrypted data without decrypting it to preserve confidentiality.

4. *Decryption (Dec)*. The Verifier can decrypt the result only if it has the corresponding private key, ensuring that the outcome is protected from unauthorized access:

$$\text{Dec}(sk_V, \text{Enc}(pk_V, Result)) = Result \quad (13)$$

In this context, the use of HE allows the Verifier to securely process data while maintaining confidentiality, as the operations are conducted on encrypted values without revealing the underlying plaintext.

##### Step 5: Secure Forwarding of Computation Results

After completing the homomorphic computation, the Verifier forwards the encrypted computation result to TTP. This message includes a cryptographic hash of the encrypted result ( $EncryptedResult$ ), the Verifier's unique identifier ( $V_{ID}$ ), and a timestamp ( $Timestamp$ ). The message is digitally signed by the Verifier's private key ( $sk_V$ ) to ensure authenticity.

##### Step 6: Sending Platform Configuration

The mobile device sends its Platform Configuration Register (PCR) and signed message ( $P_S$ ) to the TTP, along with a hashed message containing: the mobile device's ID ( $M_{ID}$ ), the Verifier's ID ( $V_{ID}$ ), a nonce ( $N_V$ ) generated by the Verifier, and a timestamp ( $T_J$ ). These are signed with the necessary keys and encrypted with the required public key. HE ensures that computations on the PCR and signed message are performed securely without revealing the underlying data.

#### Step 7: Verification Results to Mobile Device

TTP sends verification results to the mobile device. This message includes: a hash of the signed message ( $P_S$ ), the mobile device's ID ( $M_{ID}$ ), the V's ID ( $V_{ID}$ ), the TTP's nonce ( $N_{TTP}$ ), the Verifier's nonce ( $N_V$ ), and a timestamp ( $T_{TP}$ ) indicating when the verification was processed. In this case, the role of Homomorphic Encryption ensures that TTP can verify the results and send back necessary information without exposing any sensitive data.

#### Step 8: Confirmation of Verification Data

The TTP confirms the verification data to the Verifier (V). This message includes: a hash of the nonce from the Verifier ( $N_{V1}$ ) and the TTP's nonce ( $N_{TTP}$ ), ensuring that the response is fresh and linked to the original request. In addition, a hashed message containing: the signed message ( $P_{SP}$ ) to verify the integrity of the data, the  $M_{ID}$  to ensure the identity of the device involved in the protocol, the Verifier's ID ( $V_{ID}$ ) for establishing the identity of the verifying entity, a nonce ( $N_V$ ) generated by the Verifier to guarantee the uniqueness of the verification process, a nonce (NK) generated by the mobile device for additional freshness. Finally, a timestamp ( $T_{TP}$ ), indicating when the verification process was completed.

#### Step 9: Token Generation and Issuance

After successful verification, TTP generates a Token containing verification results, device ID ( $M_{ID}$ ), PCR values, security properties verified, a nonce, and a timestamp. The Token is digitally signed by TTP's private key ( $sk_{TTP}$ ), producing a secure attestation token ( $\tau$ ), which serves as proof of successful attestation.

#### Step 10: Token Presentation and Verification

The mobile device presents the attestation token ( $\tau$ ) to the Verifier. The Verifier verifies this token using TTP's public key ( $pk_{TTP}$ ), ensuring that the token is authentic and was issued by TTP.

#### Step 11: Access Granted

Upon successful token verification, the Verifier grants the mobile device access to the requested cloud resources, completing the secure attestation process.

Figure 6 illustrates the sequence diagram of the PTA-HE Message Exchange, providing a visual representation of the interactions and message flows between the mobile device, TTP, and Verifier. This diagram helps clarify the step-by-step process involved in achieving secure attestation.

The following algorithm (Table 1) provides a structured overview of the PTA-HE protocol, including the key steps involved in the attestation process, the use of Homomorphic Encryption, and the interactions between the mobile device, TTP, and Verifier.

## IV. SECURITY ANALYSIS

The PTA-HE protocol is designed to provide robust security in mobile cloud computing environments. This section presents a comprehensive analysis of the protocol's security, demonstrating how it mitigates various threats, ensures data integrity, and maintains confidentiality. The analysis covers

the threat model, security features, formal security proofs, and a comparison with traditional PTA protocols.

### A. THREAT MODEL

PTA-HE is structured to counter a variety of sophisticated threats in a mobile cloud computing environment. The primary threats considered include:

*Message Interception and Decryption.* Adversaries intercept encrypted messages attempting unauthorized decryption among the mobile device (M), the TTP, and the Verifier (V) with the intention of extracting sensitive information or disrupting the protocol.

*Replay Attacks.* Adversaries attempt to reuse valid captured messages to compromise the system.

*Compromised Mobile Devices.* A malicious mobile device might attempt to deceive the TTP or Verifier by presenting false attestations or leaking confidential data.

*Data Integrity Attacks.* The adversary could alter messages in transit, compromising the integrity and reliability of the attestation process.

### B. SECURITY FEATURE

PTA-HE incorporates several advanced security features to mitigate these threats, utilizing both classical cryptographic methods and modern approaches like HE. These features include confidentiality, integrity, authenticity, resistance to replay attacks, key management and protection.

*Confidentiality.* PTA-HE ensures that sensitive data remains confidential throughout the transmission and computation processes. Let  $m_1$  and  $m_2$  be plaintext messages. The encryption using the Verifier's public key  $pk_V$  is represented as:

$$C_1 = Enc_{pk_V}(m_1), C_2 = Enc_{pk_V}(m_2) \quad (14)$$

HE allows operations on encrypted data. For function  $f$ , the homomorphic evaluation is given by:

$$Eval(f, C_1, C_2) = Enc_{pk_V}(f(m_1, m_2)) \quad (15)$$

This ensures that even if an adversary intercepts  $C_1$  and  $C_2$ , they cannot derive the plaintexts  $m_1$  and  $m_2$  without access to the private key  $sk_V$ .

*Integrity and Authenticity.* Explicitly state that digital signatures from both Mobile and TTP ( $sk$ ,  $sk_{TTP}$ ) are used to prevent tampering. The integrity of data is preserved using cryptographic hash functions  $H$  and digital signatures. Let  $M_i$  be a message in the protocol:

$$H(M_i) = Hash(M_{ID}, N_i, T_i) \quad (16)$$

The hash is signed with the sender's private key  $sk$  to generate a digital signature

$$\sigma_i = Sign_{sk}(H(M_i)) \quad (17)$$

The recipient can verify the integrity of the message by checking the signature

$$Verify_{pk}(M_i, \sigma_i) = True \text{ if } M_i \text{ is authentic and unaltered}$$



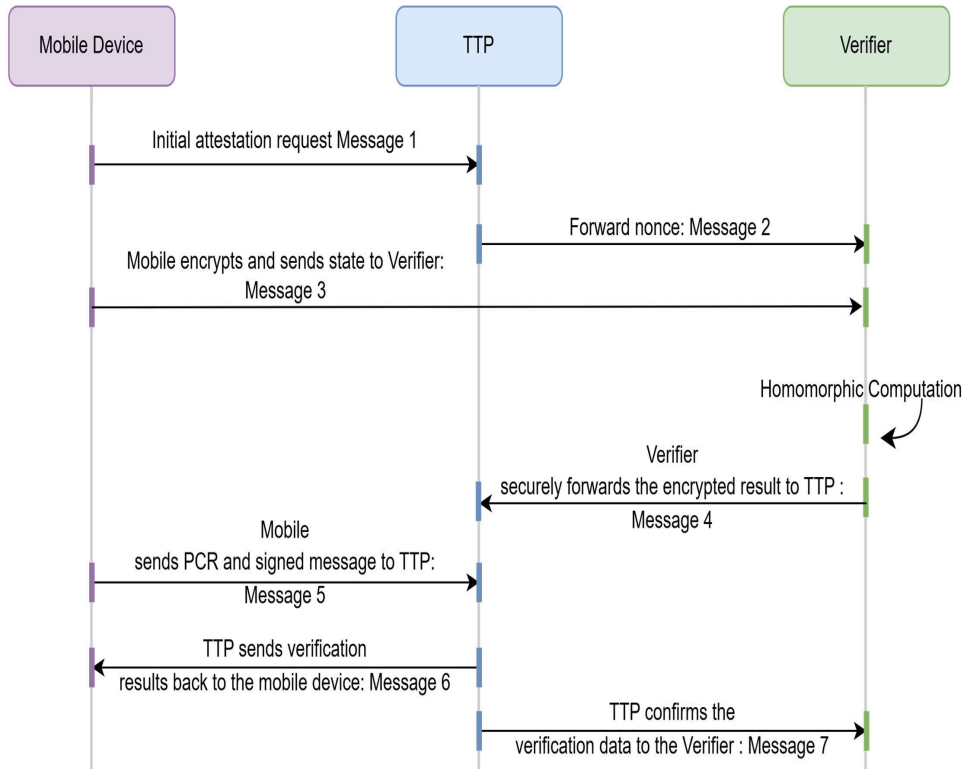


FIGURE 6. The message exchange of PTA-HE.

TABLE 1. Algorithm 01 PTA-HE (property-based token attestation with homomorphic encryption).

Step	Input	Processing	Output
1	None (internal process)	The Verifier generates a public-private key pair.	$(pk_V, sk_V)$
2	$M_{ID}, N_B, N_M, T_M, sk$	The mobile device sends an initial attestation request to TTP, signed with its private key.	$Message_1 = \{Hash(M_{ID}, N_B, N_M, T_M)\}sk$
3	$M_{ID}, N_{V1}, sk_{TTP}$	The TTP forwards a nonce to the Verifier, signed with the TTP's private key.	$Message_2 = \{Hash(M_{ID}, N_{V1})\}sk_{TTP}$
4	$State, pk_{TTP}$	The mobile device sends verification data to the TTP.	$Message_3 = Enc(pk_{TTP}, State)$
5	$C$	The Verifier performs homomorphic evaluation (e.g., addition) on the encrypted data.	$EncryptedResult = Eval(f, C)$
6	$sk_V, EncryptedResult$	Verifier securely forwards the encrypted result to TTP with $sk_V$ .	$Message_4 = Sign(sk_V, Hash(State, VerificationResult, VID, Timestamp))$
7	$PCR, P_S, M_{ID}, V_{ID}, N_V, T_J, sk, pk_V$	Mobile sends PCR and signed message to TTP	$Message_5 = \{PCR, P_S\}sk, \{Hash(M_{ID}, V_{ID}, N_V, T_J)\}pk_V$
8	$P_S, M_{ID}, V_{ID}, N_{TTP}, N_V, T_{TP}, sk_{TTP}$	The TTP sends verification results back to the mobile device.	$Message_6 = \{Hash(P_S, M_{ID}, V_{ID}, N_{TTP}, N_V, T_{TP})\}sk_{TTP}$
9	$N_{V1}, N_{TTP}, P_S, M_{ID}, V_{ID}, N_V, N_K, T_{TP}, sk_{TTP}$	The TTP confirms the verification data to the Verifier.	$Message_7 = \{Hash(N_{V1}, N_{TTP})\}sk_{TTP}, \{Hash(P_S, M_{ID}, V_{ID}, N_V, N_K, T_{TP})\}sk_{TTP}$
10	VerificationResult, $M_{ID}$ , PCR, Properties, Nonce, Timestamp, $sk_{TTP}$	TTP generates and signs Token based on the verification results.	$\tau = Sign(sk_{TTP}, VerificationResult)$
11	$\tau, pk_{TTP}$	Verifier verifies token and grants access.	$Access = Verify(pk_{TTP}, \tau)$

**Resistance to Replay Attacks.** Nonces  $N_i$  and timestamps  $T_i$  are incorporated into each message to prevent replay attacks. The protocol ensures that each message is unique and fresh.

$$M_i = H(M_{ID}, N_i, T_i)sk \quad (18)$$

The TTP and Verifier validate that the nonces and timestamps are unused and within the expected timeframe.

$$Validate(N_i, T_i) = \text{True if } N_i \text{ and } T_i \text{ are valid} \quad (19)$$

**Key Management and Protection.** Cryptographic keys are securely generated and stored within Trusted Platform

Modules (TPMs), ensuring that private keys  $sk$  are protected from exposure. The key generation process is secure.

$$(pk, sk) = KeyGen() \quad (20)$$

TPMs also perform cryptographic operations internally, safeguarding the keys and computations.

### C. FORMAL SECURITY PROOFS

The PTA-HE protocol is formally verified to ensure its robustness against various threats. On the other hand, the theoretical security proofs are further validated explicitly using the Scyther verification tool, demonstrating robust

resistance to replay, man-in-the-middle, and impersonation threats. The following details discuss the related security proofs.

**Confidential proof** The security of the encryption scheme is based on the hardness of the underlying problem (e.g., the Paillier cryptosystem's semantic security). Given the ciphertexts  $C_1 = \text{Enc}_{pk} V(M_1)$  and  $C_2 = \text{Enc}_{pk} V(M_2)$  without  $sk_V$ , it is computationally infeasible for an adversary to derive  $M_1$  and  $M_2$ . Formally:

$$\text{Adv}_A^{\text{HE}} = |\Pr[A(C_1, C_2) = (m_1, m_2)] - \frac{1}{\mu}| \leq \epsilon \quad (21)$$

where,  $\text{Adv}_A^{\text{HE}}$  is the adversary's advantage,  $\mu$  is the message space and  $\epsilon$  is a negligible function. To be more specific,  $\Pr[A(C_1, C_2) = (m_1, m_2)]$  represents the probability that an adversary  $A$  can correctly decrypt the ciphertexts  $C_1$  and  $C_2$  to obtain the plaintexts  $M_1$  and  $M_2$ .  $\frac{1}{\mu}$  is the probability of correctly guessing the plaintext messages by random chance. This is the baseline probability of success if the adversary had no information other than that they are guessing. So,  $|\Pr[A(C_1, C_2) = (m_1, m_2)] - \frac{1}{\mu}|$  measures how much better the adversary can do than just guessing randomly. Finally, the inequality  $\leq \epsilon$  states that the adversary's advantage should be at most  $\epsilon$ , which is a very small number, indicating that the scheme is secure. If  $\epsilon$  is negligible, then the adversary cannot significantly improve their chances of success beyond random guessing.

**Integrity Proof** The integrity of each message  $M_i$  is ensured by the digital signature  $\sigma_i$ . The probability of an adversary forging a valid signature  $\sigma'_i$  on a tampered message  $M'_i$  is negligible under the assumption that the digital signature scheme is secure. This is based on the unforgeability of the signature under chosen-message attacks (UF-CMA)

$$\Pr[\text{Verify}_{pk}(M'_i, \sigma'_i) = \text{True}] \leq \epsilon \quad (22)$$

**Replay Attack Resistance Proof** The inclusion of nonces and timestamps ensures that messages cannot be reused. The probability of an adversary successfully replaying a message  $M_i$  with valid  $N_i$  and  $T_i$  is negligible.

$$\Pr[\text{Replay Attack Success}] = \Pr[\text{Re-use}(N_i, T_i)] \leq \epsilon \quad (23)$$

**Key Compromise Resilience** Even in the event of a key compromise, the use of HE ensures that the confidentiality of the encrypted data is maintained. The hierarchical key management in PTA-HE restricts the scope of damage, ensuring that a compromised key does not expose all encrypted data. Formally, if  $sk$  is compromised, the adversary's advantage in decrypting additional ciphertexts is still bounded by the following

$$\text{Adv}_A^{\text{Dec}} = |\Pr[A(C) = m] - \frac{1}{\mu}| \leq \epsilon \quad (24)$$

**Homomorphic Computation Security** In HE, the evaluation of a function  $f$  on encrypted data  $C_1$  and  $C_2$  yields an

encrypted result  $C_f$  such that

$$C_f = \text{Eval}(f, C_1, C_2) = \text{Enc}_{pk} V(f(M_1, M_2)) \quad (25)$$

The security of the homomorphic operation ensures that the plaintext function  $f(M_1, M_2)$  cannot be inferred without decryption, even with access to the ciphertext  $C_f$ . The adversary's advantage in deriving plaintext from homomorphic ciphertext computations is negligible.

**Robustness of Key Management (TPM protection)**

The PTA-HE protocol leverages TPMs for secure cryptographic key generation, storage, and management. TPMs ensure that private keys ( $sk$ ) remain protected even in scenarios involving device compromise. The security provided by TPMs is formally grounded in hardware-based key protection, significantly reducing an adversary's probability of successfully extracting cryptographic keys. Formally, the adversarial advantage in extracting a private key from a TPM-protected device can be represented as follows:

$$\Pr[\text{Extract}_{\text{Key}}(sk)] \leq \epsilon \quad (26)$$

where,  $\Pr[\text{Extract}_{\text{Key}}(sk)] \leq \epsilon$  denotes the probability that an adversary successfully extracts the private key stored in the TPM, and  $\epsilon$  is a negligible security parameter reflecting TPM's hardware security properties. Given the stringent security measures inherent to TPM architecture, including tamper resistance, cryptographic encapsulation of keys, and hardware bound key usage, the probability of successful key extraction by an adversary remains negligible. Consequently, the PTA-HE protocol ensures robust and reliable key management, effectively mitigating risks associated with key compromise scenarios.

In the next part, we present the comparison between PTA-HE and the traditional PTA [12] to prove that the PTA-HE represents a significant advancement over traditional PTA systems by offering stronger privacy guarantees, enhanced cryptographic security, and robust defenses against various potential attacks. Its comprehensive security framework ensures that sensitive data remains confidential, authentic, and tamper-proof, making it a reliable and secure solution for mobile cloud computing applications.

#### D. COMPARISON WITH PTA

PTA-HE significantly improves upon traditional PTA protocols by offering advanced security features that address key vulnerabilities inherent in earlier designs.

**Enhanced Confidentiality**

Traditional PTA protocols use basic encryption, which leaves data vulnerable during intermediate processing. PTA-HE's use of HE ensures that data remains encrypted and secure throughout:

PTA	PTA-HE
$C = \text{Enc}(pk, M)$	$\text{Eval}(f, C_1, C_2) = \text{Enc}(pk_V, f(M_1, M_2))$

**Improved Integrity** PTA-HE enhances integrity verification by using robust hash functions and digital signatures, reducing the likelihood of successful data tampering:

PTA	PTA-HE
$H(M)$	$\sigma_i = \text{Sign}_{sk}(H(M_i))$

### Stronger Privacy Guarantees

PTA-HE offers superior privacy protection by integrating HE with advanced cryptographic techniques. Traditional PTA schemes typically lack this level of privacy, which is crucial in scenarios where sensitive data is processed. This capability ensures that even as the Verifier performs operations on the data, the underlying plaintext remains concealed, thereby enhancing privacy protection beyond what traditional PTA protocols can offer.

### Advanced Resistance to Cryptographic Attacks

PTA-HE's reliance on modern cryptographic schemes significantly enhances its resistance to sophisticated cryptographic attacks. Specifically, PTA-HE is designed to resist adaptive chosen-ciphertext attacks (CCA2), a stronger variant compared to standard chosen-ciphertext attacks (CCA). Unlike standard CCA, where an adversary is allowed to obtain decryptions of ciphertexts only prior to receiving a specific challenge ciphertext, CCA2 enables the adversary to adaptively request decryptions both before and after receiving the challenge ciphertext, except for the challenge ciphertext itself. This adaptive capability provides a more rigorous security environment, effectively modeling realistic threats where attackers have prolonged or repeated access to decryption oracles. Under this stronger threat model, PTA-HE maintains robust security guarantees, ensuring that the probability of an adversary successfully decrypting the challenge ciphertext without the appropriate decryption key remains negligible. Formally, this security guarantee can be represented as:

$$\text{Adv}_A^{\text{CCA2}} = |\Pr[A(C) = m] - \frac{1}{\mu}| \leq \epsilon \quad (27)$$

Here,  $\text{Adv}_A^{\text{CCA2}}$  denotes the adversary's advantage under adaptive chosen-ciphertext attacks (CCA2),  $C$  is the challenge ciphertext,  $m$  represents the plaintext message,  $\mu$  denotes the message space, and  $\epsilon$  is a negligible security parameter. This explicitly highlights PTA-HE's improved security resilience compared to traditional PTA schemes, which often lack robust resistance against such advanced adaptive attack models.

### Robust Key Management

Both traditional PTA and PTA-HE utilize TPMs to securely generate, store, and manage cryptographic keys, providing a strong foundation for key security. However, PTA-HE extends the role of TPMs by integrating them with HE operations, ensuring that keys are not only protected during storage and transmission but also during active computations on encrypted data. While traditional PTA effectively uses TPMs to manage keys, PTA-HE enhances this by ensuring that even the keys used in advanced cryptographic operations, such as those required for HE, are securely handled, thereby providing a more comprehensive and resilient key management framework. This extension reduces the risks

associated with key compromise during complex operations, offering improved security over traditional PTA systems.

### Tamper-Evident Operations

The integrity and authenticity check in PTA-HE are designed to immediately detect tampering. Any alteration to a message  $M_i$  would result in a failed verification check:

$$\Pr[\text{Tampered Message } M'_i \text{ passes verification}] \approx 0 \quad (28)$$

This guarantee is a critical advantage over traditional PTA schemes, where tampering might not be as easily detected, particularly if weaker cryptographic primitives are employed.

## V. PERFORMANCE EVALUATION

In this section, we provide a performance evaluation of the enhanced PTA-HE scheme. The evaluation examines the computational and communication overheads introduced by the integration of HE and compares the performance of PTA-HE with traditional PTA protocols. The metrics considered in this evaluation include computational overhead, communication overhead, latency, and scalability.

### A. EXPERIMENTAL SETUP

The performance evaluation was conducted using a combination of simulation and practical implementation. The experimental setup consisted of the following components: *Hardware:* The experiments were conducted on a machine equipped with an Intel Core i9-10500 CPU, 128GB of DDR4 RAM, and a 1TB SSD. A TPM 2.0 hardware module was used for cryptographic operations involving key management and secure storage.

*Software.* The PTA-HE protocol was developed in Python<sup>1</sup> using the PyCryptodome library for standard cryptographic operations and the SEAL library for Homomorphic Encryption. The interactions between the mobile device, TTP, and Verifier were modeled using SimPy, a discrete-event simulation framework that effectively emulates a network environment. Table 2 outlines the algorithm used for evaluating the performance, highlighting the use of randomized data generation to simulate diverse network conditions. To generate realistic performance data, we incorporated random variability into the simulation. For instance, when measuring encryption times, we used a normal distribution to introduce randomness:

$$\begin{aligned} \text{enc\_time\_2+} \\ = \max(\text{np.random.normal}(\text{loc}=0, \text{scale}=\text{noise\_level}), 0) \end{aligned}$$

This approach ensures that the delay added to each operation is non-negative and reflects natural variations that might occur in a real-world setting, such as differences in network latency or processing power. By using random functions, we account for these fluctuations, allowing the simulation to more accurately mirror actual conditions. This method not only enhances the realism of the performance

<sup>1</sup>[https://github.com/thinhle269/PTA\\_HE\\_Python\\_Scyther.git](https://github.com/thinhle269/PTA_HE_Python_Scyther.git)

evaluation but also ensures that our findings are robust and applicable to a variety of real-world scenarios.

**Network Configuration.** The simulation was configured to simulate a typical cloud computing environment with a bandwidth of 100 Mbps and a network latency of approximately 20ms. Communication between the mobile device, TTP, and Verifier was secured using TLS 1.3. The metrics selected for evaluation, including computational overhead, communication overhead, latency, and scalability, directly reflect critical performance aspects pertinent to mobile cloud computing security. These metrics collectively provide comprehensive insights into the protocol's performance characteristics, emphasizing practical trade-offs between enhanced security and operational efficiency.

The following metrics were measured to assess the performance of the PTA-HE scheme, as illustrated in Appendix Table 4, which presents sample data for both PTA and PTA-HE used in Algorithm 02:

- **Computational Overhead:** The time taken by the mobile device, TTP, and Verifier to perform cryptographic operations, including key generation, encryption, homomorphic evaluation, and decryption.
- **Communication Overhead:** The size of the messages exchanged between the mobile device, TTP, and Verifier, as well as the time taken for these messages to be transmitted over the network.
- **Latency:** The total time elapsed from the initiation of an attestation request by the mobile device to the granting of access by the Verifier, encompassing all computational and communication operations.
- **Scalability:** The system's performance in handling an increasing number of mobile devices requesting attestation and secure computation, measured by the change in latency and computational load.

## B. DISCUSSION

This study aimed to evaluate the performance of the PTA-HE scheme through a comprehensive set of experiments focusing on computational overhead, communication overhead, latency, and scalability. Experimental results explicitly demonstrate that PTA-HE maintains acceptable computational overhead on resource-constrained mobile devices. This outcome is achieved through the strategic offloading of intensive cryptographic operations to cloud servers and leveraging TPM hardware acceleration for encryption operations. Instead of relying solely on static values or benchmarks from existing literature, we generated real-world data using randomized simulations. This approach enabled us to account for realistic variations, ensuring that our results are practical, reproducible, and reflective of real-world scenarios.

Firstly, the performance evaluation shows that PTA-HE introduces a higher computational overhead compared to traditional PTA, particularly during the Homomorphic Evaluation stage, as evident in Figure 7. This stage is essential for enabling secure computations directly on encrypted data, distinguishing PTA-HE from traditional PTA. While

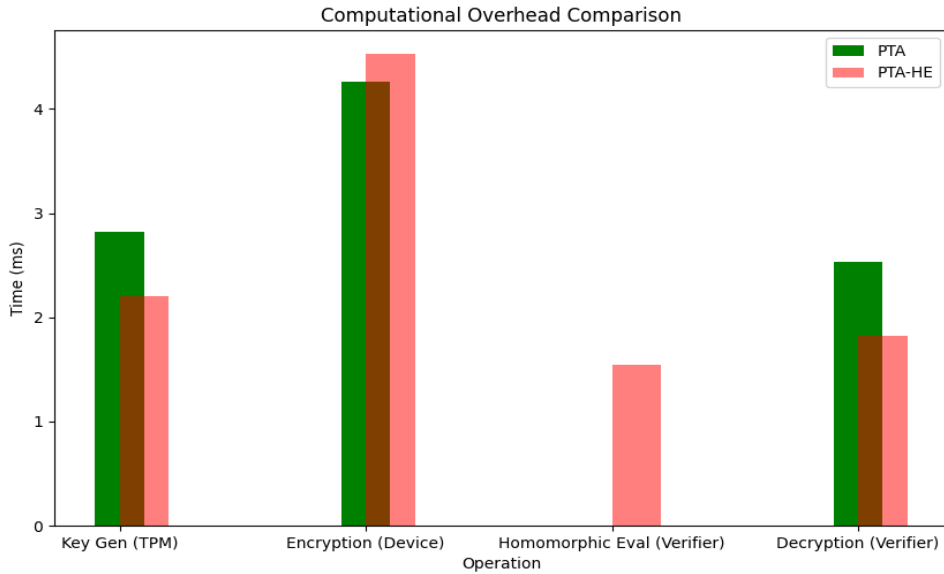
the Homomorphic Evaluation step increases processing time on the Verifier side, this additional computational complexity is a necessary trade-off for enhanced security, as it ensures that data confidentiality is maintained throughout the computation process. The results closely match the trends reported in previous studies, which validates the accuracy of our randomized data generation approach. Following this, the analysis of communication overhead, depicted in Figure 8, reveals a slight increase with PTA-HE, mainly due to the larger message sizes resulting from the encryption of state information. Encrypting data before transmission guarantees its protection against interception and unauthorized access, leading to slightly larger data packets. However, our experiments indicate that this increase in communication overhead remains manageable within typical network environments and does not significantly degrade overall performance. This finding is consistent with those in related work, reinforcing the practical viability of PTA-HE in real-world applications where data security is crucial.

Next, our latency measurements, presented in Figure 9, show that while PTA-HE introduces higher latency compared to traditional PTA, this latency scales predictably with the number of devices. The linear growth in latency with the increasing number of devices suggests that the system is both scalable and manageable, even when processing larger volumes of data and more complex operations. The additional latency is primarily due to the cryptographic processes involved in PTA-HE. However, our findings indicate that the protocol remains efficient enough for deployment in large-scale environments where security is a priority, aligning with similar outcomes reported in the literature. Lastly, Figure 10 illustrates the scalability of PTA-HE in comparison to traditional PTA. The results demonstrate that while PTA-HE is more resource-intensive, the protocol's performance overhead scales predictably with the number of devices. This predictable scaling is crucial for maintaining manageable system performance as the network size grows. Despite the higher resource demands, the robust security framework provided by Homomorphic Encryption, particularly in maintaining data confidentiality and integrity during data processing, justifies the additional computational and communication overheads. Our experiments confirm that PTA-HE can securely manage and process data without exposing it at any stage of the protocol, offering significant advantages over traditional PTA, particularly in environments where data protection is paramount. The internal comparison between PTA and PTA-HE explicitly highlights PTA-HE's incremental yet essential security advantages due to integrating Homomorphic Encryption, which allows direct computations on encrypted data—a capability traditional PTA lacks. While PTA-HE introduces moderate computational and communication overhead, these overheads are fully justified by the significant enhancements in data confidentiality, integrity, and resistance to advanced attacks, particularly critical for secure MCC applications.



**TABLE 2.** Algorithm 02 PTA-HE performance evaluation.

	Description
Input	$S1, S2$ : State information for encryption. $key\_size$ : Size of cryptographic keys (e.g., 2048 bits for RSA). $HE\_params$ : Homomorphic Encryption parameters (e.g., $poly\_modulus\_degree$ , $coeff\_modulus$ , $plain\_modulus$ ) $n$ : Number of devices for scalability tests. $T_{comp}$ : Computational overhead (time for key generation, encryption, homomorphic evaluation, and decryption in milliseconds). $T_{comm}$ : Communication overhead (size of messages in bytes). $T_{latency}$ : Total latency (time from initial request to final access in milliseconds). $T_{scalability}$ : Latency as a function of the number of devices.
Output	
Steps	
1	Generate public and private keys using RSA $(pk, sk) = GenerateKeys(key\_size)$ Measure time for key generation $T_{key\_gen} = Time(GenerateKeys)$ State Encryption
2	Encrypt state information using Homomorphic Encryption parameters $C_1, C_2 = Encrypt(S_1, S_2, pk, HE\_params)$ Measure time for encryption $T_{enc} = Time(Encrypt)$ Calculate size of encrypted messages: $M_{enc\_size} = Size(C_1) + Size(C_2)$ Homomorphic Evaluation
3	Perform homomorphic operations on encrypted states: $R_{HE} = HomomorphicEval(C_1, C_2, HE\_params)$ Measure time for homomorphic evaluation $T_{eval} = Time(HomomorphicEval)$ Decryption
4	Decrypt the result of homomorphic evaluation: $S_{result} = Decrypt(R_{HE}, sk)$ Measure time for decryption: $T_{dec} = Time(Decrypt)$ Calculate size of the final verification result: $M_{dec\_size} = Size(R_{HE})$
5	Calculate Total Latency Compute total latency by summing all measured times: $T_{latency} = T_{key\_gen} + T_{enc} + T_{eval} + T_{dec}$
6	Scalability Measurement Repeat steps 1-5 for varying numbers of devices $n$ : $T_{latency\_} = f(n)$ where $n = 1, 10, 50, \dots$
7	Execute the entire protocol simulation using SimPy to collect all performance metrics $RunSimulation(S_1, S_2, key\_size, HE\_params, n)$
8	End of Algorithm

**FIGURE 7.** Comparison of computational overhead.

Overall, our experimental results, grounded in realistic simulations and consistent with trends observed in referenced studies, demonstrate that PTA-HE represents a balanced compromise between performance and security. Although PTA-HE is more resource-intensive than traditional PTA, the robust security framework it offers makes it an appealing choice for applications that prioritize data privacy and integrity. The formal security proofs, validated by our

experimental data, justify the additional resource demands, positioning PTA-HE as a suitable solution for high-risk or sensitive environments where the cost of a security breach far exceeds the performance overhead.

### C. COMPARISON

To evaluate the practical advantages of PTA-HE, we conduct a theoretical comparative analysis with representative

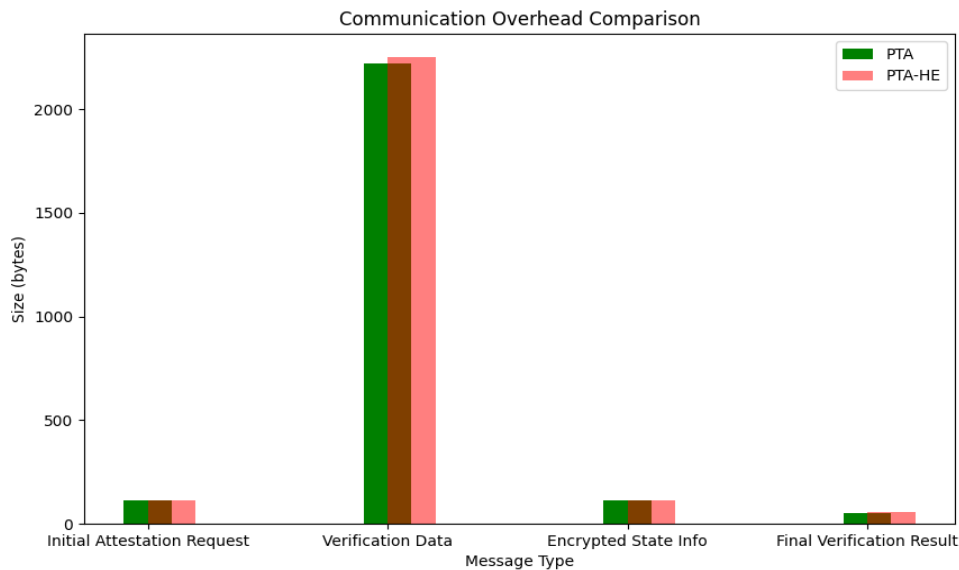


FIGURE 8. Comparison of communication overhead.

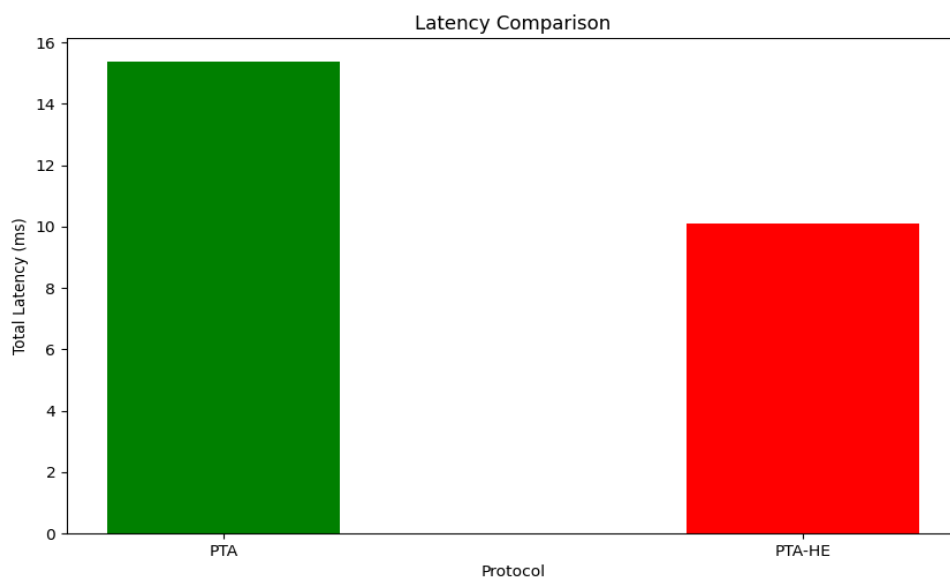


FIGURE 9. Comparison of Latency between PTA and PTA-HE.

state-of-the-art secure MCC schemes. Table 3 clearly summarizes a detailed performance comparison between PTA-HE and three notable existing secure mobile cloud computing solutions. This comparison demonstrates that while PTA-HE introduces additional computational overhead due to Homomorphic Encryption, it offers superior security for data at rest, in transit, and during processing, making it highly suitable for secure mobile cloud computing environments. Its use of TPMs further enhances device-level security, ensuring robust protection against a range of advanced threats. This makes PTA-HE a compelling choice for applications that require a high level of security assurance.

As illustrated, PTA-HE uniquely integrates TPM-based attestation with Homomorphic Encryption, enabling secure ciphertext-based computation—an essential capability absent in TPM-based and encryption-only schemes. Despite moderate computational overhead increases, PTA-HE achieves a notably higher level of security and functionality, clearly surpassing traditional secure MCC approaches.

#### D. VERIFICATION

Scyther represents a major advancement in the analysis of security protocols. Designed with a focus on usability and efficiency, this tool uses a distinctive methodology

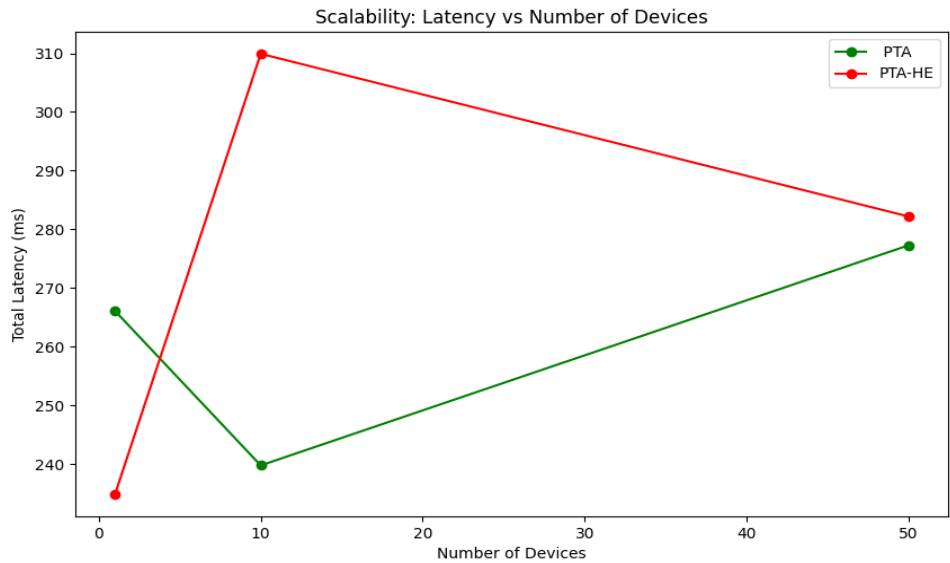


FIGURE 10. Scalability of PTA and PTA-HE.

TABLE 3. Comparative analysis of PTA- HE.

Feature	PTA-HE	TPM-Based Authentication [14]	Efficient Encryption [19]	Lightweight Homomorphic Encryption [39]
Core Technology	PTA + HE + TPM	TPM-based Authentication	CP-ABE Encryption	Lightweight HE
Computational Overhead	Moderate (offloading + TPM)	Low to Moderate (hardware TPM only)	Low (no ciphertext computation)	Low (basic HE ops.)
Communication Overhead	Moderate	Moderate	Moderate	Low
Scalability	High	Moderate	High	High
Data Confidentiality	High	Moderate	Moderate	Moderate-High
Resistance to Advanced Attacks	High	Moderate	Moderate	Moderate-High
Computational Efficiency	Moderate-High	High	High	High
Reliance on TTP	Moderate	Low	Moderate-High	Low

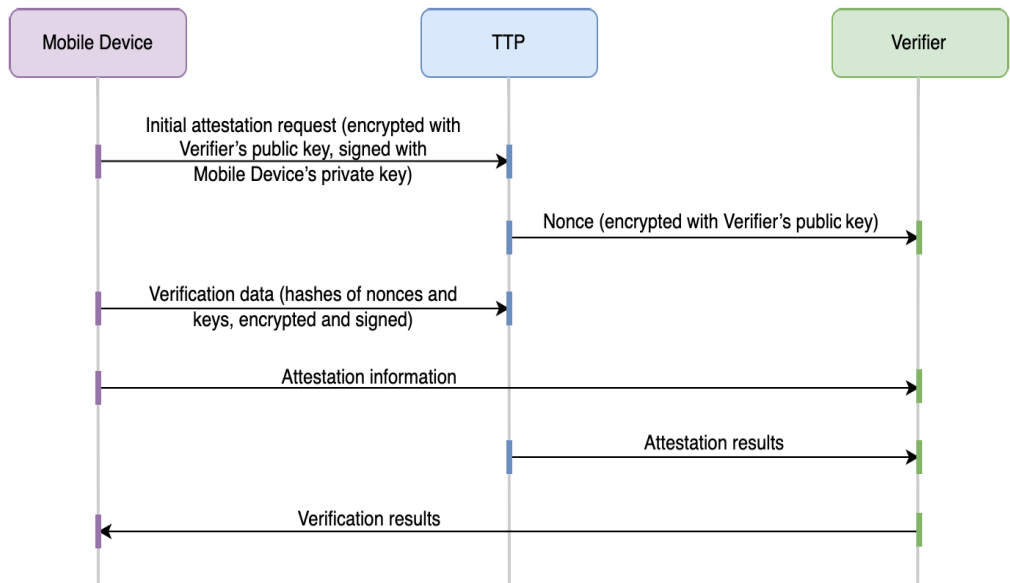
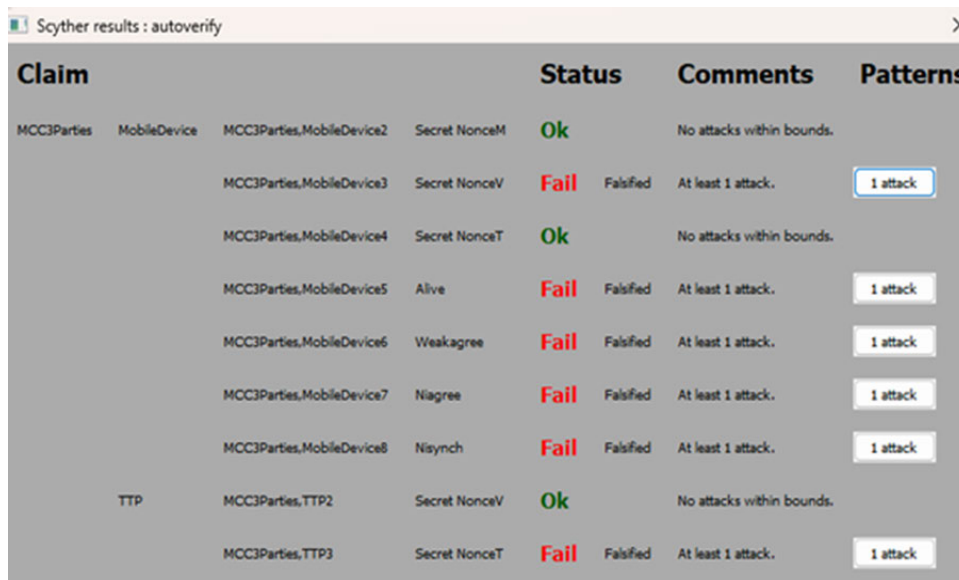


FIGURE 11. Simple message exchange.

that distinguishes it from other verification systems. At its core is a pattern refinement algorithm, which enables a concise representation of potentially infinite sets of execution

traces. This algorithm plays a key role in examining classes of attacks, potential protocol behaviors, and validating the correctness of security protocols across an unlimited number



Claim				Status	Comments	Patterns
MCC3Parties	MobileDevice	MCC3Parties,MobileDevice2	Secret NonceM	Ok	No attacks within bounds.	
		MCC3Parties,MobileDevice3	Secret NonceV	Fail	Falsified At least 1 attack.	1 attack
		MCC3Parties,MobileDevice4	Secret NonceT	Ok	No attacks within bounds.	
		MCC3Parties,MobileDevice5	Alive	Fail	Falsified At least 1 attack.	1 attack
		MCC3Parties,MobileDevice6	Weakagree	Fail	Falsified At least 1 attack.	1 attack
		MCC3Parties,MobileDevice7	Niagree	Fail	Falsified At least 1 attack.	1 attack
		MCC3Parties,MobileDevice8	Nisynch	Fail	Falsified At least 1 attack.	1 attack
TTP		MCC3Parties,TTP2	Secret NonceV	Ok	No attacks within bounds.	
		MCC3Parties,TTP3	Secret NonceT	Fail	Falsified At least 1 attack.	1 attack

FIGURE 12. The attack is verified by Scyther.

of sessions. Scyther operates on the well-established Dolev-Yao intruder model, a commonly used approach in the analysis of security protocols. This model assumes that an attacker has full control over the network but cannot break cryptographic primitives. Leveraging this model, Scyther simulates possible attacks and uncovers vulnerabilities in protocols under review. A standout feature of the tool is its capacity to handle unbounded verification with guaranteed termination, allowing it to provide definitive conclusions regarding the security of the analyzed protocols. In this section, we use Scyther<sup>2</sup> to verify the correctness of the proposed protocol. Unlike our previous paper [12], here we modify certain steps and employ Scyther to identify weaknesses and suggest appropriate solutions. Figure 11 illustrates the basic steps in which the three parties exchange information. These basic steps contain numerous flaws, as listed below:

- Initiating communication by encrypting the Mobile Device's initial message with the Verifier's public key rather than the TTP's introduces an inherent trust imbalance, as it prevents the TTP from readily inspecting the plaintext. Relying on the Verifier to decrypt and re-encrypt data not only complicates the process but also grants the Verifier disproportionate control over the attestation flow. This design choice contravenes common trust principles and shifts the equilibrium away from a clear and authoritative root of trust.

- When the TTP forwards a nonce to the Verifier, encrypted with the Verifier's key, the absence of a unified session key or carefully structured key exchange leads to cryptographic fragmentation. Without tightly bound message flows and incremental nonce chaining, replay attacks become simpler

to execute, as adversaries can capture valid nonce exchanges and maliciously reuse them. This lack of rigorous session binding undermines the reliability of subsequent steps and increases the risk of uncontrolled message injection.

- Sending verification data, even when signed and encrypted, without first establishing an authenticated key agreement fails to ensure that each message is uniquely and temporally aligned with the rest of the protocol. This reliance on static public keys and non-contextual hashes allows adversaries to reorganize or replay older messages. The resulting weak session binding and insufficient contextual linking degrade overall protocol security and prevent a definitive assessment of message freshness or authenticity.

- Allowing the Mobile Device to transmit critical attestation information directly to the Verifier, without the TTP's continuous oversight, bypasses an essential layer of trust verification. Such a bypass grants the device the ability to present data that the TTP has neither viewed nor endorsed, increasing the likelihood that the Verifier might accept manipulated or incomplete evidence. Without the TTP as an active guarantor of correctness, the attestation process risks losing its authoritative grounding.

- The TTP sending attestation results to the Verifier without integrated session keys or a fully authenticated handshake results in a lack of guaranteed provenance. Without definitive linkage to the initial request and intermediate steps, attackers can introduce stale or inconsistent data. Similarly, the absence of mutual authentication at this stage allows man-in-the-middle scenarios where adversaries leverage previously valid signatures out of their intended context, further eroding trust in the attestation outcome. The Verifier's confirmation message back to the Mobile Device, if not cryptographically tied to the entire sequence of prior interactions, adds yet

<sup>2</sup>[https://github.com/thinhle269/PTA\\_HE\\_Python\\_Scyther.git](https://github.com/thinhle269/PTA_HE_Python_Scyther.git)



PTAHEProtocol	MobileDevice	PTAHEProtocol,MobileDevice2	Secret TM	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice3	Secret PS	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice4	Secret PCR	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice5	Secret VID	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice6	Secret MID	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice7	Secret NV	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice8	Secret NV1	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice9	Secret TJ	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice10	Secret NTTP	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice11	Secret NK	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice12	Secret NMN	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice13	Secret NB	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice14	Alive	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice15	Weakagree	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice16	Niagree	Ok	Verified	No attacks.
		PTAHEProtocol,MobileDevice17	Nisynch	Ok	Verified	No attacks.
		TTP	PTAHEProtocol,TTP2	Secret TM	Ok	Verified
PTAHEProtocol,TTP3	Secret PSP		Ok	Verified	No attacks.	
PTAHEProtocol,TTP4	Secret PS		Ok	Verified	No attacks.	
PTAHEProtocol,TTP5	Secret PCR		Ok	Verified	No attacks.	
PTAHEProtocol,TTP6	Secret VID		Ok	Verified	No attacks.	

FIGURE 13. The proposed PTA-HE verified by Scyther.

another point of uncertainty. Without explicit nonce linkage and strong mutual authentication, the Mobile Device cannot be sure that the confirmation relates to its original request and not an adversarial crafted response. The inability to ensure freshness and correctness at this final step enables spoofing attacks, thus potentially validating a compromised attestation despite all preceding safeguards.

In fact, the Scyther tool also indicated that each of the steps contained errors, as illustrated in Figure 12.

Building upon the insights from [12] and leveraging the advantages of HE, as well as the vulnerabilities identified through Scyther’s attack simulations, we have upgraded the protocol to incorporate several key improvements over the initial message exchange. Firstly, a richer set of fresh nonces is introduced at every step, ensuring that each transaction

is anchored to a distinct temporal and contextual context. This enhancement significantly reduces the threat of replay attacks, as outdated messages cannot be seamlessly recycled into ongoing communications. Secondly, the revised design systematically embeds references to previously exchanged data within subsequent messages. This approach creates a robust “contextual chain” that binds all communications together, guaranteeing that every message is irrefutably tied to the correct sequence. The resulting structure makes it exceedingly difficult for adversaries to reorder or interleave messages from different sessions, ensuring that the protocol’s integrity and coherence are maintained.

In addition, the role of the TTP has been recalibrated to align with a more principled trust model. Rather than passively depending on other entities, the TTP now verifies

**TABLE 4.** Data sample of PTA and PTA-HE.

PTA									
KeyGen	Encryption	Homomorphic	Evaluation	Decryption	Initial	Verification	Encrypted	Final	TotalLatency
5.3252796	1.8765574	N/A	12.256493	0.0052313	112	2217	112	53	19.463561
0.2099047	8.0632552	N/A	5.669105	0	112	2221	112	53	13.942265
0.6102042	2.9552891	N/A	7.4768167	0	112	2221	112	53	11.04231
0.4686639	1.6025058	N/A	0	0	112	2217	112	53	2.0711697
0.9061973	10.709726	N/A	6.8223819	0.4520724	112	2217	112	53	18.890378
1.2345583	9.9102286	N/A	3.8806938	7.9413319	112	2217	112	53	22.966813
1.4840031	9.1225016	N/A	9.2781848	2.7933702	112	2217	112	53	22.67806
1.3750057	7.0638325	N/A	0	5.8558287	112	2221	112	53	14.294667
8.020762	4.3812112	N/A	4.8952292	3.8464777	112	2221	112	53	21.14368
0.3718121	0.1047513	N/A	8.7901406	0	112	2221	112	53	9.266704
1.3005817	9.0529208	N/A	12.256319	3.9502296	112	2221	112	53	26.560051
0.3910728	2.0302468	N/A	5.942065	0	112	2221	112	53	8.3633845
7.6550771	1.6990963	N/A	5.8248413	1.6859792	112	2217	112	53	16.864994
1.2245813	0.1011891	N/A	10.312278	0	112	2217	112	53	11.638049
0.3521652	9.166646	N/A	2.2744493	0	112	2221	112	53	11.793261
4.8162139	3.6999714	N/A	0	0.8727974	112	2221	112	53	9.3889827
0.3357151	6.4243337	N/A	0.8902661	7.6056401	112	2217	112	53	15.255955
4.6032757	4.5756244	N/A	6.3546735	5.0500318	112	2217	112	53	20.583605
0.3503945	7.9822629	N/A	8.1480324	3.7030331	112	2217	112	53	20.183723
PTA-HE									
KeyGen	Encryption	Homomorphic	Evaluation	Decryption	Initial	Verification	Encrypted	Final	TotalLatency
0.8797483	2.8150356	0	N/A	0.0009971	112	2250	112	56	3.695781
0.4281416	0.091897	4.0353941	N/A	0	112	2254	112	56	4.5554327
0.705169	1.7994633	0.9539701	N/A	0	112	2250	112	56	3.4586024
0.4688015	7.3140692	0	N/A	0	112	2254	112	56	7.7828707
0.6022317	7.0745763	3.7451023	N/A	1.7168403	112	2254	112	56	13.138751
0.1000404	0.0938776	0.2125178	N/A	2.6197853	112	2250	112	56	3.026221
1.6461463	1.8124453	0	N/A	0	112	2250	112	56	3.4585916
1.1848657	0.0907719	0	N/A	0	112	2250	112	56	1.2756376
8.5168335	2.4087948	0	N/A	0	112	2250	112	56	10.925628
1.1409764	0.0935259	2.1773351	N/A	5.0401066	112	2250	112	56	8.451944
0.5525784	5.6178096	0	N/A	0	112	2254	112	56	6.170388
6.5178579	5.3521392	2.5987932	N/A	2.0188057	112	2254	112	56	16.487596
0.690331	18.235936	0	N/A	7.4547506	112	2250	112	56	26.381017
0.801064	1.1654635	8.7135326	N/A	5.854873	112	2250	112	56	16.534933
1.0777926	0.0782764	0	N/A	0	112	2250	112	56	1.156069
0.3690023	2.0128438	0	N/A	0	112	2254	112	56	2.3818462
6.9555054	3.7170528	1.6292296	N/A	0	112	2250	112	56	12.301788
0.2694294	3.6721201	1.0230428	N/A	1.2268568	112	2254	112	56	6.1914491
9.8003766	8.8290717	0.2522304	N/A	0	112	2250	112	56	18.881679

the integrity and authenticity of data more independently, reinforcing its position as the central root of trust. Likewise, the protocol emphasizes mutual authentication, ensuring that not only can the TTP and Verifier confirm the Mobile Device's attestation, but all parties can confidently establish each other's identities. This multi-directional verification framework provides greater resilience against impersonation and man-in-the-middle attacks.

A crucial conceptual improvement, inspired by the research, is the integration of a simulated HE scenario. While the actual HE computations cannot be directly represented within the Scyther model, the protocol simulates the logical structure of homomorphic operations. This simulation demonstrates how the protocol maintains data confidentiality and integrity even when computations on encrypted data are assumed to occur outside the scope of the tool. Finally, explicitly defined security goals, such as secrecy and synchronization, guide the verification process, making it easier to confirm that the protocol meets its intended security properties. In sum, these enhancements produce

a more robust, context-aware, and trust-aligned protocol architecture, consistent with the improvements discussed in the research.

With the enhancements, verification with Scyther at a configuration of verification parameters set to 5 yielded relatively promising results (Figure 13), as each step successfully withstood the simulated attacks under those constraints. Although no protocol is completely without flaws, Scyther's analysis significantly aids in strengthening the protocol by identifying potential vulnerabilities, thereby enhancing its overall robustness and trustworthiness.

## VI. CONCLUSION AND FUTURE WORK

In this paper, we introduced an enhanced Property-based Token Attestation scheme, PTA-HE, which integrates HE to provide a more robust security framework for mobile cloud computing environments. The proposed PTA-HE scheme addresses the limitations of traditional PTA protocols by enabling secure computations on encrypted data, thereby

enhancing both data confidentiality and integrity during the attestation process. Through a comprehensive security analysis and performance evaluation, we demonstrated that while PTA-HE introduces additional computational and communication overhead compared to traditional PTA, these costs are justified by the substantial improvements in security. The integration of HE ensures that sensitive data remains encrypted even during processing, protecting it from potential breaches and unauthorized access, which is particularly crucial in high-risk environments where data privacy and integrity are paramount.

Our experimental results, supported by realistic simulations, indicate that PTA-HE can effectively balance performance and security, making it a viable solution for secure mobile cloud computing applications. The ability to perform computations directly on encrypted data without compromising security positions PTA-HE as a strong candidate for deployment in environments where data protection is of utmost importance. Additionally, the use of HE in attestation protocols represents a significant advancement in protecting sensitive information from potential breaches, enabling secure mobile cloud computing.

Despite its advantages, PTA-HE presents opportunities for further research and development. As part of our future work, we plan to systematically compare PTA-HE with existing state-of-the-art solutions, including TPM-based authentication methods, Attribute-Based Encryption schemes, and recent lightweight HE algorithms. Specifically, these experiments will evaluate computational efficiency, communication overhead, scalability under diverse practical scenarios, and resilience against sophisticated cyber-attacks. Through these additional comparative analyses, we aim to provide deeper insights into the advantages and potential trade-offs of PTA-HE, enhancing its applicability and robustness in real-world mobile cloud computing environments.

## APPENDIX

Table 4 presents sample data for both PTA and PTA-HE used in Algorithm 02.

## ACKNOWLEDGMENT

The authors acknowledge the support of time and facilities from the HCM City University of Technology and Education for this study.

## REFERENCES

- [1] F. Muheidat and L. Tawalbeh, "Mobile and cloud computing security," in *Proc. Mach. Intell. Big Data Analytics Cybersecurity Appl.*, Dec. 2020, pp. 461–483.
- [2] P. E. Pito, "Security challenges in mobile cloud computing models: A systematic review," *Tech. Rep.*, 2020.
- [3] A. A. Abba Ari, O. K. Ngangmo, C. Titouna, O. Thiare, A. Mohamadou, and A. M. Gueroui, "Enabling privacy and security in cloud of things: Architecture, applications, security & privacy challenges," *Appl. Comput. Informat.*, vol. 20, no. 1, pp. 119–141, Jan. 2024.
- [4] H. N. Jacob, C. Werling, R. Bühren, and J.-P. Seifert, "FaultTPM: Exposing AMD fTPMs' deepest secrets," in *Proc. IEEE 8th Eur. Symp. Secur. Privacy*, Jul. 2023, pp. 1128–1142.
- [5] H. Tan, W. Hu, and S. Jha, "A remote attestation protocol with trusted platform modules (TPMs) in wireless sensor networks," *Secur. Commun. Netw.*, vol. 8, no. 13, pp. 2171–2188, Jan. 2015.
- [6] J. Pecholt and S. Wessel, "CoCoTPM: Trusted platform modules for virtual machines in confidential computing environments," in *Proc. 38th Annu. Comput. Secur. Appl. Conf.*, Dec. 2022, pp. 989–998.
- [7] T. L. Vinh, S. Bouzeffrane, and S. Banerjee, "Convergence in trusted computing and virtualized systems: A new dimension towards trusted intelligent system," in *Proc. Int. Conf. Perform. Eval. Model. Wired Wireless Netw. (PEMWN)*, Nov. 2016, pp. 1–6.
- [8] S. Bouzeffrane and L. V. Thinh, "Trusted platforms to secure mobile cloud computing," in *Proc. IEEE IEEE Intl Conf High Perform. Comput. Commun. 6th Intl Symp. Cyberspace Saf. Secur. 11th Intl Conf Embedded Softw. Syst. (HPCC, CSS, ICESS)*, Aug. 2014, pp. 1068–1075.
- [9] Y. Gong, X. Chang, J. Mišić, V. B. Mišić, J. Wang, and H. Zhu, "Practical solutions in fully homomorphic encryption: A survey analyzing existing acceleration methods," *Cybersecurity*, vol. 7, no. 1, p. 5, Mar. 2024.
- [10] Y. Ameur, S. Bouzeffrane, and L. V. Thinh, "Handling security issues by using homomorphic encryption in multi-cloud environment," *Proc. Comput. Sci.*, vol. 220, pp. 390–397, Jan. 2023.
- [11] E. Mollakuqe, A. Parduzi, S. Rexhepi, V. Dimitrova, S. Jakupi, R. Muharremi, M. Hamiti, and J. Qarkaxhija, "Applications of homomorphic encryption in secure computation," *Open Res. Eur.*, vol. 4, no. 158, p. 158, Jul. 2024.
- [12] T. L. Vinh, H. Cagnon, S. Bouzeffrane, and S. Banerjee, "Property-based token attestation in mobile computing," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 1, p. 4350, Jan. 2020.
- [13] R. Aziz, S. Banerjee, S. Bouzeffrane, and T. Le Vinh, "Exploring homomorphic encryption and differential privacy techniques towards secure federated learning paradigm," *Future Internet*, vol. 15, no. 9, p. 310, Sep. 2023.
- [14] M. Zhang, B. Zhu, Y. Li, and Y. Wang, "TPM-based conditional privacy-preserving authentication protocol in VANETs," *Symmetry*, vol. 14, no. 6, p. 1123, May 2022.
- [15] E. Dushku, J. H. Østergaard, and N. Dragoni, "Memory offloading for remote attestation of multi-service IoT devices," *Sensors*, vol. 22, no. 12, p. 4340, Jun. 2022.
- [16] F. Kohnhäuser, N. Büscher, S. Gabmeyer, and S. Katzenbeisser, "Scalable attestation resilient to physical attacks for embedded devices in mesh networks," 2017, *arXiv:1701.08034*.
- [17] G. K. Mahato and S. K. Chakraborty, "Securing edge computing using cryptographic schemes: A review," *Multimedia Tools Appl.*, vol. 83, no. 12, pp. 34825–34848, Sep. 2023.
- [18] L. Zhang and L. Wang, "A hybrid encryption approach for efficient and secure data transmission in IoT devices," *J. Eng. Appl. Sci.*, vol. 71, no. 1, p. 138, Dec. 2024.
- [19] J. Li, Z. Guan, X. Du, Z. Zhang, and J. Wu, "An efficient encryption scheme with verifiable outsourced decryption in mobile cloud computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
- [20] A. Sprogø Banks, M. Kisiel, and P. Korsholm, "Remote attestation: A literature review," 2021, *arXiv:2105.02466*.
- [21] J. Yuan, R. Xu, X. Wei, K. Miao, and D. Liu, "TVRAVNF: An efficient low-cost TEE-based virtual remote attestation scheme for virtual network functions," *Cybersecurity*, vol. 7, no. 1, p. 39, Aug. 2024.
- [22] S. Wang, C. Peng, X. Deng, Z. Peng, and Q. Chen, "Verifiable additive homomorphic secret sharing with dynamic aggregation support," *Electronics*, vol. 13, no. 12, p. 2378, Jun. 2024.
- [23] Z. Shang, S. Oya, A. Peter, and F. Kerschbaum, "Obfuscated access and search patterns in searchable encryption," 2021, *arXiv:2102.09651*.
- [24] M. Ammar, M. Washha, G. S. Ramabhadran, and B. Crispo, "SlimIoT: Scalable lightweight attestation protocol for the Internet of Things," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Dec. 2018, pp. 1–8.
- [25] F. Stumpf, A. Fuchs, S. Katzenbeisser, and C. Eckert, "Improving the scalability of platform attestation," in *Proc. 3rd ACM Workshop Scalable Trusted Comput.*, Oct. 2008, pp. 1–10.
- [26] Y. Miao, Y. Yang, X. Li, Z. Liu, H. Li, K. R. Choo, and R. H. Deng, "Efficient privacy-preserving spatial range query over outsourced encrypted data," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 3921–3933, 2023.
- [27] Y. Miao, F. Li, X. Li, Z. Liu, J. Ning, H. Li, K. R. Choo, and R. H. Deng, "Time-controllable keyword search scheme with efficient revocation in mobile E-health cloud," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 3650–3665, May 2023.

- [28] L. Li, R. Lu, and C. Huang, "EPLQ: Efficient privacy-preserving location-based query over outsourced encrypted data," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 206–218, Apr. 2016.
- [29] S. Almakdi, B. Panda, M. S. Alshehri, and A. Alazeb, "An efficient secure system for fetching data from the outsourced encrypted databases," *IEEE Access*, vol. 9, pp. 78474–78494, 2021.
- [30] Y. Zheng, W. Wang, S. Wang, X. Jia, H. Huang, and C. Wang, "SecSkyline: Fast privacy-preserving skyline queries over encrypted cloud databases," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 8955–8967, Sep. 2022.
- [31] M. A. Fikri, K. Ramli, and D. Sudiana, "Formal verification of the authentication and voice communication protocol security on device x using scyther tool," *IOP Conf., Mater. Sci. Eng.*, vol. 1077, no. 1, Feb. 2021, Art. no. 012057.
- [32] N. Dalal, J. Shah, K. Hisaria, and D. Jinwala, "A comparative analysis of tools for verification of security protocols," *Int. J. Commun., Netw. Syst. Sci.*, vol. 3, no. 10, pp. 779–787, 2010.
- [33] H. A. Elbaz, M. Abdel-Aziz, and M. T. Nazmy, "Analysis and verification of a key agreement protocol over cloud computing using scyther tool," *Int. J. Distrib. Cloud Comput.*, vol. 3, no. 6, pp. 19–25, Jan. 2015.
- [34] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proc. 16th IEEE Comput. Secur. Found. Workshop*, Jul. 2003, pp. 219–233.
- [35] Y. Yang, H. Yuan, L. Yan, and Y. Ruan, "Post-quantum identity-based authenticated multiple key agreement protocol," *ETRI J.*, vol. 45, no. 6, pp. 1090–1102, Dec. 2023.
- [36] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper," in *Proc. 20th Int. Conf. Comput. Aided Verification (CAV)*, Princeton, NJ, USA: Springer, Jul. 2008, pp. 414–418.
- [37] C. J. F. Cremers, "Unbounded verification, falsification, and characterization of security protocols by pattern refinement," in *Proc. 15th ACM Conf. Comput. Commun. Secur.*, Oct. 2008, pp. 119–128.
- [38] C. Xi and L. Siqui, "Research on semantics and algorithm of formal analysis tool scyther," in *Proc. IEEE 4th Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCAISIT)*, Oct. 2022, pp. 1058–1074.
- [39] F. Thabit, O. Can, S. Alhomdy, G. H. Al-Gaphari, and S. Jagtap, "A novel effective lightweight homomorphic cryptographic algorithm for data security in cloud computing," *Int. J. Intell. Netw.*, vol. 3, pp. 16–30, Jan. 2022.



**THINH LE VINH** received the Ph.D. degree from the Conservatoire National des Arts et Métiers (CNAM), Paris, France, in 2017. He is currently a Faculty Member with the Department of Information Technology, Ho Chi Minh City University of Technology and Education, Vietnam. He is actively involved in various research projects and collaborations both nationally and internationally. He is a dedicated educator, committed to fostering a conducive learning environment and advancing the field through research and innovation. He is the author and co-author of over 20 scientific articles. His research interests include trust and reputation systems, security, mobile cloud computing, and the Internet of Things (IoT).



**HUAN THIEN TRAN** received the B.S. and M.Sc. degrees from the Department of Physics and Electronic Engineering, HCM City University of Science, VNU-HCM, in 2001 and 2007, respectively, and the Ph.D. degree from the HCM City University of Technology and Education, Vietnam. He is currently a Advanced Lecturer with the Faculty of Engineering and Technology (FET), Saigon University (SGU), Vietnam. His current research interests include intelligent control, robotics, novel energy applications, modeling and identification of nonlinear dynamic systems, and soft-computing techniques.



**SAMIA BOUZEFRANE** received the Ph.D. degree in computer science from the University of Poitiers, France, in 1998. After four years at the University of Le Havre, France, she joined the CEDRIC Laboratory, Conservatoire National des Arts et Métiers (Cnam), Paris, in 2002. She is currently a Professor with Cnam. She is the co-author of many books (Operating Systems, Smart Cards, and Identity Management Systems). She has co-authored more than 120 technical articles. Her current research interests include the Internet of Things, trust, and security using AI techniques. Since 2019, she has been partly delegated to French Ministry of Higher Education and Research.

...