IoT Security: A Critical Evaluation and Improvement of ChatGPT-Generated Authentication Protocols Using Bilinear Pairing

WenBin Hsieh

d9802106@gmail.com

National Taitung University

Research Article

Keywords: ChatGPT, Pairing, authentication, IoT, security

Posted Date: December 4th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5487468/v1

Additional Declarations: No competing interests reported.

IoT Security: A Critical Evaluation and Improvement of ChatGPT-Generated Authentication Protocols Using Bilinear Pairing

IoT Security: A improvement ChatGPT-Generated Authentication Protocols Using Bilinear Pairing

WENBIN HSIEH*

Department of Electronic and Computer Engineering, National Taipei University of Technology, Taipei, 106344, Taiwan, wbhsieh@ntut.edu.tw

Authentication in Internet of Things (IoT) is critical to protect devices, ensure data integrity, prevent unauthorized access, and protect against cyber threats in interconnected systems. It maintains trust and privacy in the IoT ecosystem. ChatGPT, which is a Generative Pre-trained Transformer, is a powerful tool that can provide a variety of knowledge, including the principles and common practices involved in designing network protocols. In this paper, we use interactive interrogation to let ChatGPT design the authentication protocols required for IoT and explore the security of the resulting protocol. A fatal flaw was discovered in the protocol created by ChatGPT during the security analysis process. In addition, we examine the pros and cons of the generated protocol, provide considerations necessary to protect IoT communications, and further propose an identity-based, enhanced and scalable authentication protocol for IoT devices using bilinear pairing. We also compared the performance of the improved protocol with the original. Research results can be fed back to ChatGPT to improve its knowledge of designing protocols and the improved protocol is more secure, efficient and practical. Moreover, this paper proposes a future research motif aimed at stimulating researchers to enhance artificial intelligence by proposing counterarguments to suggested perspectives on artificial intelligence.

CCS CONCEPTS • Security and Privacy • Networks • Information System

Additional Keywords and Phrases: ChatGPT, Pairing, authentication, IoT, security

1 INTRODUCTION

Over the years, a variety of compelling and significant research topics have emerged, including artificial intelligence, information security, and so on. Soon many people figure out to combine artificial intelligence and information security to study cyber-attacks and defenses. Shortly thereafter, researchers figured out how to combine artificial intelligence with information security to study network attacks and defenses. For example, Shachar et al. [20] introduced an innovative IoT security testbed framework in 2019 that uses machine learning algorithms for comprehensive standard and advanced security testing across diverse IoT devices. In 2020, Bagaa et al. [16] proposed a novel machine learning-based security framework for IoT, integrating Software Defined Networking and Network Function Virtualization to enhance threat mitigation and anomaly detection. Following this, in 2021, Keserwani et al. [17] combined Grey Wolf Optimization [18] and Particle Swarm Optimization [19] to extract relevant features and used a Random Forest classifier to develop an intrusion

* Corresponding author: WenBin Hsieh, e-mail: wbhsieh@ntut.edu.tw

detection system for IoT networks. However, in 2022, a significant advancement in artificial intelligence emerged, fundamentally transforming research in machine learning for cybersecurity. Chat Generative Pre-trained Transformer (ChatGPT) [21, 22] based large language models (LLMs) [23] was launched on November 30, 2022. ChatGPT can be used for a variety of tasks, such as developing and debugging computer programs, producing books and papers, answering exam questions, and coming up with research proposals. It can also provide solutions and explanations for problems and topics, offering significant convenience to users in acquiring new knowledge.

Cryptographic protocols are essential for many of our digital activities, such as connecting to a Wi-Fi hotspot, accessing HTTPS websites, managing IoT networks, and making electronic payments. Designing and deploying security protocols requires advanced expertise in cryptographic primitives, making it a complex and challenging task. Even established, peer-reviewed, and standardized protocols are frequently found to contain flaws or have a room for potential improvement. For instance, in 2017, Aman et al. [24] use a physical unclonable function (PUF) to propose a lightweight mutual authentication protocol that is asserted to be well-suited for real-time IoT applications. After analysing the weaknesses and vulnerabilities of Aman's protocols, Amir et al. [25] proposed an improved lightweight two-factor authentication protocol for IoT applications in 2023. Soon after, in 2024, Ghazaleh et al. [26] examined the proposed protocol and presented a lightweight multi-factor authentication protocol based on PUFs that addresses the shortcomings of existing solutions.

After the above discussion, we can understand that general users have become popular in using ChatGPT to query cryptography-related issues. However, when asked to provide a secure IoT authentication protocol, we found that the advice given by ChatGPT was incorrect on some basic concepts of cryptography.

The novel contributions and distinctive advancements presented in this research are delineated as follows:

(1) First, we highlight the issue of ChatGPT's unreliability in specific advanced mathematical fields, such as cryptography, by systematically exposing and scrutinizing the inaccuracies in its generated responses.
(2) Next, in addressing the aforementioned issue, we have meticulously integrated the concepts of identity-based cryptography with bilinear pairing-based public key cryptography to propose a method that leverages the identities of IoT nodes for secure identity authentication and key agreement.
(3) Finally, tthrough comprehensive security and performance analyses, we demonstrate the security and scalability of the proposed protocol, proving its practicality in real-world scenarios.

The structure of this paper is as follows: Section 2 introduces the key concepts necessary for understanding the paper. Section 3 discusses the protocol provided by ChatGPT. In Section 4, we propose an innovative authentication protocol. Section 5 presents the security and performance analyses, and finally, Section 6 concludes our research.

2 PRELIMINARY

In this section, we embark on the foundational concepts that are germane to the subsequent discussion. We will delve into the essential preliminaries that serve as the foundation for subsequent topics, providing the necessary background and context. By understanding these preliminary notions, the depth and nuances of the content that will be introduced in the forthcoming sections can be more easily understood.

2.1 Elliptic curve cryptography [12, 13]

Definition 2.1. An elliptic curve E over a field K of characteristic $\neq 2$ or 3 is defined as a cubic curve represented by the equation $y^2 = x^3+ax+b$, comprising points (x, y) along with a designated element O, known as the "point at infinity." Figure 1 illustrates elliptic curves following four different formulas.

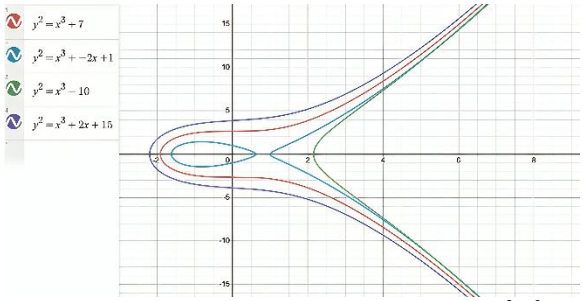

Figure 1 A point on an elliptic curve is a coordinate pair (x, y) that satisfies the equation $y^2=x^3+ax+b$, where a and b are constants defining the curve's structure. [11]

Definition 2.2. Let P, $Q \in E$ and given a line L passing through P and Q (if $P = Q$, then the line L is the tangent to the elliptic curve E at the point P). The line L intersects the curve E at the third point $-R$. Let L' be the line through O and $-R$ intersects the curve E at the third point denoted as $R = P \oplus Q$. Therefore, $P \oplus Q$ represents the point obtained by adding the points P and Q on the curve. The law is illustrated in Figure 2 and 3.

Proposition 2.3. Given three points P, Q, R (not necessarily distinct from each other) lying on the intersection of the line L and the curve E. Thus, the composition law demonstrates the following properties:

- $O = (P \oplus Q) \oplus R$
- For all $P \in E$, $O = P \oplus O$
- If $P \in E$, $\exists -P \in E$ s.t. $O = P \oplus (-P)$
- $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$

The aforementioned properties, in conjunction with the closure property, make E into abelian group.

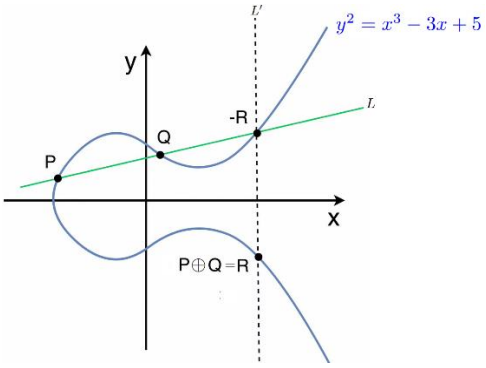

Figure 2 Addition of a point P to itself

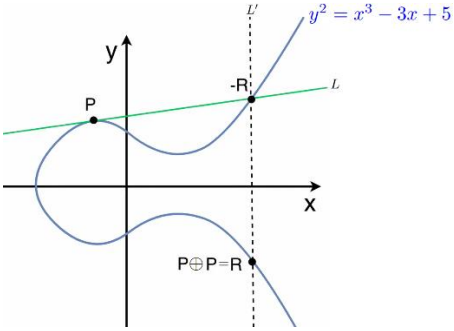

Figure 3 Addition of two distinct points *P* and *Q*

2.2 Bilinear Pairings [9, 10]

Given that n is a prime number, consider $G_1 = \langle P \rangle$, an additive group of order n with the identity element denoted by ∞. Likewise, let G_T be a multiplicative group of order n, where 1 represents the identity member.

Definition 2.4. A bilinear pairing on the groups $\langle G_1, G_T \rangle$ is defined as a mapping $\hat{e}$ such that:

$$\hat{e} = \ G_1 \times G_1 \to G_T$$

The following properties are satisfied.

(1) ***Bilinearity.*** For all $R, S, T \in G_1$, the pairing meets the equalities below:

$$\hat{e}(R + S, T) = \hat{e}(R, T)\hat{e}(S, T)$$

and

$$\hat{e}(R, S + \ T) = \hat{e}(R, S)\hat{e}(S, T)$$

(2) ***Non-degeneracy.*** The pairing is considered non-degenerate, meaning there exists $P \in G_1$ such that

$$\hat{e}(P, P) \neq 1$$

(3) ***Computability.*** The map $\hat{e}$ can be computed efficiently.

The following properties of bilinear pairings can be formally established. Property (5) provides an alternative characterization of non-degeneracy. For all $S, T \in G_1$.

(1) $\hat{e}(S, \infty) = 1$ and $\hat{e}(\infty, S) = 1$
(2) $\hat{e}(S, -T) = \hat{e}(-S, T) = (\hat{e}(S, T))^{-1}$
(3) $\hat{e}(aS, bT) = \hat{e}(S, T)^{ab}$ for all $a, b \in Z$
(4) $\hat{e}(S, T) = \hat{e}(T, S)$
(5) If $\hat{e}(S, R) = 1$ for all $R \in G_1$, then $S = \infty$

The Discrete Logarithm Problem (DLP) in G_1 can be efficiently reduced to the DLP in G_T as a direct result of the bilinear property. Specifically, if (P, Q) is an instance of the DLP in G_1 where $Q = xP$, then $\hat{e}(P, Q) = \hat{e}(P, xP) = \hat{e}(P, P)^x$. Consequently,$\log_g h$, where $g = \hat{e}(P, P)$ and $h = \hat{e}(P, Q)$ are elements of G_T.

Additionally, the security of numerous pairing-based protocols is contingent upon the computational intractability of the following problems.

Definition 2.5. Consider $\hat{e}$ as a bilinear pairing on the groups $\langle G_1, G_T \rangle$. The Bilinear Diffie-Hellman Problem (BDHP) is defined as follows: Given P, aP, bP, cP, the goal is to compute $\hat{e}(P, P)^{abc}$.

The hardness of the BDHP indicates a corres ponding difficulty in solving the Diffie-Hellman problem (DHP) in the groups G_1 and G_T. If the DHP in G_1 can be efficiently resolved, one can feasibly address an instance of the DBHP by computing abP and subsequently calculating $\hat{e}(abP, cP) = \hat{e}(P, P)^{abc}$. Similarly, if the DHP in G_T is solvable, then the BDHP can be tackled by determining $g = \hat{e}(P, P)$, $g^{ab} = \hat{e}(aP, bP)$, $g^c = \hat{e}(P, cP)$ and finally g^{abc}. Although the precise hardness of the BDHP remains unknown, it is widely presumed to be as computationally infeasibility as the DHP in G_1 and G_T.

Moreover, the decisional Diffie-Hellman problem (DDHP) in G_1 is efficiently solvable. The DDHP involves determining whether a given quadruple (P, aP, bP, cP) within G_1 forms a valid Diffie-Hellman tuple, i.e., whether $cP = abP$. This can be verified by computing $\gamma_1 = \hat{e}(P, cP) = \hat{e}(P, P)^c$ and $\gamma_2 = \hat{e}(aP, bP) = \hat{e}(P, P)^{ab}$, where $cP = abP$ holds if and only if $\gamma_1 = \gamma_2$.

2.3 Identity-based cryptography (IBC) [14, 15]

IBC depends on a trusted third party known as the Private Key Generator (PKG). At the beginning, the PKG is responsible for generating a master public/private key pair, denoted as mpk_{PKG} and msk_{PKG}. The fundamental procedure for performing encryption and decryption is as follows:

- **Setup**(1^k)**:** The algorithm is executed by the PKG and the input is the security parameter 1^k. Upon execution, the system parameters, along with the PKG's master public key (*mpk*) and master secret key (*msk*), are generated and exported.
- **Extract**($\boldsymbol{msk}$, $\boldsymbol{ID} \in \{\mathbf{0}, \mathbf{1}\}^*$)**:** The PKG executes the algorithm which is employed to determine an individual's private key. The input to the **Extract** algorithm is the identity of the individual. The output is the corresponding private key.
- **Encrypt**($\boldsymbol{m}, \boldsymbol{ID}$)**:** The sender of plaintext *m* is required to execute this algorithm. The encryption algorithm takes the plaintext *m* and the receiver's identity as inputs, and it outputs the encrypted message *c*.

3 CHATGPT'S PROTOCOL FOR INTERNET OF THINGS

5.1. The Fundamentals of ChatGPT Protocol

In this section begins with an overview of ChatGPT's IoT-specific protocol. After receiving a query for providing a secure protocol for IoT device, ChatGPT provides the five key principles without detailed explanation. Therefore, we further introduce these principles as follows:

(1) Ensuring device identity. In order to guarantee only genuine and authorized devices can access the network, each device must be verified. The goal is also to prevent unauthorized malicious devices from launching attacks to the networks, maintain data integrity and secure the system. This can be achieved through unique device identifiers, cryptographic certificates and so on.

(2) Maintaining data integrity. IoT involves transmission, collection and storage of large amounts of data by resource-constraint devices. This data needs to be accuracy, consistent, and immutable. For this purpose, encryption or error checking mechanisms can be used to protect data from corruption and unauthorized modification. Thus, the reliability and credibility can be ensured.

(3) Protecting confidentiality. Data collected and transmitted through IoT may contain sensitive data, including personal information, health indicators, and financial details. Without strong confidentiality measures, this information can easily be intercepted by malicious users, resulting in privacy violations, financial losses, or personal harm. To ensure the confidentiality of data, encryption mechanisms are often used.

(4) Ensuring scalability. [1] As IoT networks grow, managing thousands of devices and large amounts of data requires efficient data processing and minimizing human intervention to prevent failures. The scalable architecture features efficient addressing and authentication protocols such as MQTT and CoAP with SSL/TLS to ensure secure, automated and reliable communications to support the expanding IoT ecosystem.

(5) Efficiency. IoT devices have to operate independently in remote or hostile situations for extended periods of time without maintenance since they have limited energy, storage, and computing capability.

Efficiency is essential for preventing malfunctions that compromise agreements or interrupt services, and it emphasizes the best use of resources for long-term dependability and performance. The list of efficiencies that should be considered include energy [2, 3], network [2, 4] and time [2, 4].

With these principles in mind, ChatGPT designed an IoT authentication protocol, as shown in Figure 1. There are four phases in ChatGPT's protocol which are Device Registration, Mutual Authentication, Secure Communication Establishment, and Data Integrity and Confidentiality. Each phase is summarized below.

· Device Registration

(1) When the product exits the manufacturing facility, each device is assigned a unique identifier, its Device ID (ID_i).
(2) A public and private key pair is generated for the device.
(3) The manufacturer's certificate authority (CA) issues a certificate for each device that contains the device ID, the device's public key, and the manufacturer's signature.
(4) Each device stored private keys and the certificates securely in its hardware security module (HSM) or trusted platform module (TPM).

· Mutual Authentication

After powering on or connecting to the Internet, the device will initiate a secure connection to the IoT gateway or server.

(1) The server sends an authentication request to the device containing the server's certificate and a nonce signed by the server's private key.
(2) After receiving the authentication request, the device executes the following steps:
   i. Use the CA public key provided by the manufacturer to validate the server's certificate.
   ii. Use the private key on the device to encrypt the nonce that was received from the server.
   iii. Return the server with a response containing the encrypted nonce and the certificate signed by the device's private key.
(3) The following actions are then taken by the server:
   i. Use the CA public key provided by the manufacturer to verify the device's certificate.
   ii. Use the device's public key to decrypt the nonce.
   iii. Compare the sent nonce with the decrypted version. The device is authenticated if they match. Otherwise, terminate the connection.

· Secure Communication Establishment

Based on Diffie-Hellman key exchange algorithm, the server and device use each other's public keys obtained from the certificate together with their private keys to calculate the session key.

· Data Integrity and Confidentiality

(1) Transmitted data is encrypted by a session key using a symmetric algorithm such as AES-GCM which provides message authentication.
(2) Use Diffie-Hellman key exchange to update session keys periodically.

5.2. Disadvantages of ChatGPT protocol

In this section, we point out the shortcomings of the protocol proposed by ChatGPT. Details are as follows:

(1) The fatal mistake of misusing public key cryptography：During the mutual authentication phase, the step where the device encrypts the received nonce using its private key is incorrect.

Fig. 1. The lightweight cryptographic protocol proposed by ChatGPT to ensure efficiency on resource-constrained IoT devices (The original, unaltered flowchart provided by ChatGPT)

Since the corresponding public key is known to other devices or servers, the purpose of encryption is not achieved. In fact, this is the same behavior as the last step of signing the message using the device's private key. Additionally, if one of the certificate authorities of the many manufacturers is compromised, it could lead to various attacks.

(2) Certificate Authority risks：In ChatGPT's protocol, CA plays an important role in the entire system. This may suffer from the following risks.

i. Management Overhead： The issue, renewal, and revocation of certificates by CAs necessitate a large amount of management overhead. It may not be feasible to do this in a large-scale IoT deployment where there may be millions of devices. Moreover, all manufacturers' CAs must be linked together, which is a challenging task.

ii. A single point of failure：The CA serves as a focal point for trust. The security of the entire system is jeopardized if one of the several manufacturers' CAs is breached.

(3) Man-in-the-middle attack：ChatGPT recommends using the Diffie-Hellman key exchange algorithm to calculate session key. As mentioned in CA risks, if one of the many manufacturers' CAs is compromised, a malicious or unauthorized device, server, or gateway could issue tampered credentials to launch a man-in-the-middle attack, as shown in Figure 1.

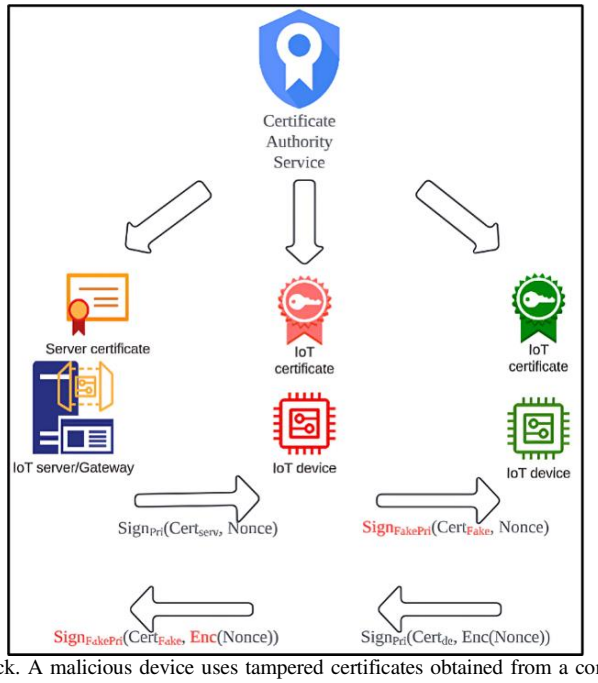

Figure 2. Man-in-the-middle attack. A malicious device uses tampered certificates obtained from a compromised CA to sign messages

Lack of scalability：As mentioned earlier, , the adoption of CAs limits the scalability of IoTs. Additionally, the Diffie-Hellman key exchange algorithm lacks scalable flexibility. If a group of devices wants to communicate simultaneously, the algorithm must be implemented between every pair of devices, which is very time-consuming and inefficient.

4 THE INNOVATIVE BILINEAR PAIRING AUTHENTICATED KEY AGREEMENT FOR IOT NETWORKS

The proposed scalable identity-based authentication key agreement protocol is described in this section. First, we go over the protocol's definitions and security features.

4.1 Definition and Properties of the proposed protocol

Definition 4.1. An identity-based authenticated key agreement scheme combines digital signature and key agreement algorithms, consisting of the following five processes:

- **Setup**: Upon inputting the security parameter 1^k, the probabilistic algorithm generates the initial key pair for the trusted third party, also known as the private key generator (PKG), which contains the public key Y and the secret key S. At the same time, common parameters are produced, which are used to generate the public and private key pairs for IoT devices.
- **Join**: The process between the IoT group proxy and the new IoT device results in the device becoming a new group member. The output is the key pair for the new IoT device.
- **Sign:** An algorithm that takes in a group public key, a new IoT device's secret, and an ephemeral key pair. The output is the signature $\widetilde{Sig}$ and the ephemeral public key R.
- **Verify**: An algorithm takes as input partial verification values from group members, the public key of the device intending to join the group, and the signature Sig. The output is either a new group public key that includes the new IoT device or an error message.

- **Key agreement**: The algorithm takes as input the public keys of group members and new participants. The output is the new group session key.
- **Refresh**: The process between the IoT group proxy, group members, and the removal of IoT devices results in the output of a new group public key.

The design of the proposed identity-based authenticated key agreement protocols is based on the following properties.

- **Correctness**: The signature implicitly generated by the new IoT device using **Sign** must be accepted by **Verify**.
- **Unforgeability**: Only devices with a true identity and genuine possession of the corresponding secret can generate a signature.
- **Key Confirmation**: Ensure that both parties possess the same key after the key agreement process.
- **Coalition-resistance**: A new malicious IoT device and a colluding subset of group members cannot generate a signature that can pass verification.
- **Efficiency:** Efficiency depends on the following parameters: the size of the group public key, the length of the signature, and the efficiency of the whole protocol.

4.2 A novel identity-based bilinear pairing authenticated key agreement protocol

This section describes the detailed process of the identity-based authenticated key agreement protocol using bilinear pairing. The conceptual diagram is illustrated in Figure 1. As preliminarily introduced in the previous section, the proposed protocol consists of five processes, described as follows:

[Setup]

When the security parameter 1^k is entered, the trusted third-party acts as a private key generator (PKG), selecting a random number $t \in Z_q^*$ and setting a public key $T = tP_2$, where $P, P_1, P_2 \in G_1$ and G_1 is a cyclic additive group generated by P. Let G_T be be a cyclic multiplicative group of the same order q. The public parameters of the systems are $params = \{G_1, G_T, e, q, P, P_1, P_2, T, H_1, H_2, H_3\}$, where $H_1: \{0,1\}^* \to Z_q^*$, $H_2: \{0,1\}^* \to \{0,1\}^k$ and $H_3: G_T \to \{0,1\}^l$. PKG keeps t secret as the master key. The IoT group proxy is assigned a key pair (d_{GP}, D_{GP}) where d_{GP} is the private key defined a $d_{GP} = tx_{GP}H_1(ID_{GP})$, and D_{GP} is the public key defined as $D_{GP} = tx_{GP}H_1(ID_{GP})P_1$.

[Join]

The manufacturer produces a new IoT device and assigns it to PKG. The PKG uses the device's identity to generate a key pair (d_i, D_i) where d_i is the private key defined a $d_i = tx_iH_1(ID_i)$, and D_i is the public key defined as $D_i = tx_iH_1(ID_i)P_1$. The private key is embedded in the device and is safeguarded in secure storage, such as a Trusted Execution Environment (TEE). The public key is distributed to the IoT group proxy.

[Sign]

The new IoT device first generates a temporary key pair (r_A, R_A), where $r_A \in Z_q^*$ is a random number and $R_A = r_AP_1$.

Now suppose this new device wants to join a group of *n* IoT devices, it must calculate the following process.

(1) $\partial = H_3\big(\hat{e}(d_AP_1, P_2)\big)r_A$
, where $d_A = tx_AH_1(ID_A)$

(2) $\widetilde{Sig} = H_2(\hat{e}(\sum_{i=1}^{n} R_i, \sum_{i=1}^{n} D_i\, \partial))$
, where $R_i = r_iP_1$, $D_i = tx_iH_1(ID_i)P_1$ and $i \in \{1,2,\dots,n\}$

The output is a signature $\widetilde{Sig}$. The new device then sent $\{R_A, \widetilde{Sig}\}$ to the IoT group proxy.

[Verify]

After receiving the signature from the new device, the IoT group proxy will distribute the participant's public key to the group members and request the group members to generate their own partial authentication tokens for verification. Each group member computes as follows.

(1) $(r_i P_1, d_i R_A H_3(\hat{e}(D_A, P_2)))$
$= (r_i P_1, d_i r_A P_1 H_3(\hat{e}(tx_A H_1(ID_i) P_1, P_2)))$
$= (r_i P_1, d_i P_1 H_3(\hat{e}(tx_A H_1(ID_i) P_1, P_2)) r_A)$
$= (r_i P_1, D_i \partial)$

Each member sends the above result to the group proxy, which then calculates the product of all results.

$\rho = \prod_{j=1}^{n} \prod_{i=1}^{n} \hat{\boldsymbol{e}}(r_i P_1, D_j \partial)$

(2) Finally, the IoT group proxy computes the hash value of ρ, denoted as $\widetilde{Sig}' = H_2(\rho)$ and then compares $\widetilde{Sig}'$ with $\widetilde{Sig}$.

(3) If $\widetilde{Sig}'$ matches $\widetilde{Sig}$, the new device is authenticated and proceeds to the next key agreement phase. Otherwise, the proxy will reject the new device and terminate this session.

[Key agreement]

After verifying the legitimacy of the new device, each group member will calculate a new group session key. The process is divided into two rounds.

Round 1.

The Internet of Things Group Proxy (IGP) exchanges the public key with each device. Note that this step needs to be performed only once. Subsequently, only new devices will require this step.

Communication.

$IGP \xrightarrow{D_{GP}} M_i$

$M_i \xrightarrow{D_i} IGP$

, where M_i represents IoT devices and $1 \le i \le n$.

Computation.

$IGP: K_{Pi} = d_{GP} D_i = (x_{pi}, y_{pi}), 1 \le i \le n.$

$M_i: K_{ip} = d_i D_{GP} = (x_{pi}, x_{pi}), 1 \le i \le n.$

Every device and the group proxy will obtain the same point, $K_{Pi} = K_{ip}$ and $K_{Pi} \neq K_{Pj}$ for $i \neq j$ and $1 \le i, j \le n$. The hash value $H_1(x_{pi})$ is then used as *n* shared keys between the IGP and each member of the group.

Round 2.

Computation.

$IGP: O_i = \left[\prod_{j=1, j\neq i}^{n} H_1(x_{pj})\right] P, 1 \le i \le n$

Communication.

$IGP \xrightarrow{O_i} M_i, 1 \le i \le n$

Group Key Agreement.

$M_i: P_K = H_1(x_{pi}) O_i = H_1(x_{pi}) \left[\prod_{j=1, j\neq i}^{n} H_1(x_{pj})\right] P = \left[\prod_{i=1}^{n} H_1(x_{pi})\right] P = (x_K, y_K)$

$K = H_2(x_K)$, which now serves as a group key among the members.

[Refresh]

Suppose one of the devices in the group is damaged or malfunctioning and needs to be removed from the group. The most important security constraint for a leaving member is the concealment of future information

(keys) with respect to both the group members and external devices. After not receiving a heartbeat signal from a specific device for a certain period of time, the IoT group proxy will perform the following steps.

(1) Assume M_j is deemed non-operational.
IGP generates a random number $V_j \in Z_q^*$ and
computes an inverse $\varepsilon = H_1(x_{pj})^{-1} \in Z_q^*$.

(2) IGP encrypts $[\varepsilon V_j]$ with $H_1(x_{pi})$ and transmits the ciphertext to other group members M_i, $i \neq j$.

$$IGP \xrightarrow{Enc_{H_1(x_{pi})}([\varepsilon V_j])} M_i, 1 \leq i \leq n, i \neq j$$

(3) After receiving the ciphertext, the corresponding device use the shared key $H_1(x_{ip})$ to decrypt the message and calculates the new group session key, as shown below. M_i: $P_{K_{new}} = H_1(x_{pi})[\varepsilon V_j]O_i$

$$\begin{aligned} &= H_1(x_{pi})[\varepsilon V_j]\left[\prod_{j=1,j\neq i}^{n} H_1(x_{pj})\right]P \\ &= V_j[H_1(x_{pj})^{-1}]\left[\prod_{i=1}^{n} H_1(x_{pi})\right]P \\ &= \left[\prod_{i=1,i\neq j}^{n} H_1(x_{pi})V_j\right]P \\ &= (x_{K_{new}}, y_{K_{new}}) \end{aligned}$$

$$K_{new} = H_2(x_{K_{new}})$$

5 SECURITY AND PERFORMANCE ANALYSIS

In this section, we will start by performing a thorough and comprehensive security analysis of the suggested protocol in this section. The computational complexity associated with each phase of the protocol will next be evaluated.

5.1. Security Analysis

The proposed protocol guarantees the following critical security properties.

(1) Certificate authority-free:
As previously mentioned in this paper, many protocols, including the one proposed by GPT, rely on Certificate Authorities (CAs) to issue certificates that contain identity information and public keys. However, this approach is not without its flaws; for instance, if a CA is compromised, the trustworthiness of the entire ecosystem is jeopardized. The proposed protocol, however, uses an identity-based mechanism that provides greater flexibility and scalability. This proposed protocol generates cryptographic key pairs based on the identities embedded within Internet of Things (IoT) devices. The authenticity of these devices is verified through the validation of signatures generated by private keys, thus ensuring the legitimacy of the devices. This methodology eliminates the need for the widespread deployment of hierarchical CAs, thereby significantly mitigating potential security threats within the system. By obviating the necessity for a full Public Key Infrastructure (PKI) and certificate management, the process is significantly simplified.

(2) Resist to man-in-the-middle attacks:
To authenticate its identity, the IoT device must generate a signature by computing $H_3(\hat{e}(d_A P_1, P_2))r_A$. This computation utilizes an identity-based private key derived from the manufacturer's secret. The possibility of a man-in-the-middle attack is rendered infeasible due to the secure nature of the identity-based cryptographic mechanism and the trustworthiness of the Private Key Generator (PKG), making the forgery of any identity impracticable.

(3) Forward secrecy:

As defined in [5], forward secrecy ensures that the compromise of long-term keying material, which is utilized for the authentication and negotiation of session keys, does not compromise the confidentiality of session keys established prior to the breach. Canetti et al. [6] further pointed out that achieving perfect forward secrecy is not feasible within the context of a two-round authenticated key-exchange protocol. Therefore, in light of this, the proposed protocol employs ephemeral key pairs (r_i, R_i) , which are used only once to establish a session key. Furthermore, when a group member departs, a random number is introduced to ensure the unpredictability and confidentiality of the session key agreement.

(4) No key control (KC): In the proposed group key agreement protocol, a group session key is established using ephemeral public keys, D_i , contributed by each participant in the group. The design of the protocol ensures that no single participant can unilaterally determine the final value of the shared secret key or exert undue influence over the generation process. Each participant contributes equally to the formation of the group session key, such that even if one participant attempts to introduce bias by selecting a particular value, the collective contributions of the other honest participants preserve the randomness and security properties of the group session key. This ensures a fair and secure key agreement process where the final key is not predictable or controllable by any single party.

(5) Replay attack: The signature generation process involves a pair of ephemeral keys, which are different whenever a member joins or leaves the group. This protocol design enhances security by ensuring that each signature is unique and non-repetitive, thereby effectively mitigating the risk of replay attacks. By preventing the reuse of identical signatures, the proposed protocol maintains the integrity and authenticity of group communications

(6) Revocation handling: Former members should not retain the privilege or ability to access group messages. To ensure this, the IoT group proxy monitors the heartbeat signals of all members. If a device ceases to transmit a heartbeat signal, indicating it is no longer present, the agent generates a random number V_j unknown to the removed device, along with an inverse of the data $H_1(x_{pj})$ to offset the contribution of the removed device. The group session keys are then recalculated using this new information, thus preserving the confidentiality of group communications from devices that are no longer part of the group.

5.2. Performance evaluation

In our proposed protocol, there are four primary phases for which the computational costs must be considered. To ensure ease of understanding for readers, we analyse the protocol step by step from the perspective of fundamental operations. The summary of the computational complexity for the signing and verification processes is discussed below.

· **Signing Complexity**.

(1) Ephemeral key pair generation:
A random number r_i is selected and the corresponding public key $R_i = r_i\ P_1$ is calculated. The operation involves scalar multiplication on an elliptic curve, resulting in a computational complexity of $O(k)$ where k denotes the size of the scalar, which representing the random number r_i.

(2) Signature calculation:
The value ∂ is derived by performing a hash operation on $\left(\hat{e}(d_A P_1, P_2)\right)$ and then multiplying by a scalar r_A. Subsequently, the computation of the signature $\widetilde{Sig}$ primarily dominated by the pairing operation, which typically has a computational complexity of $O(klog(k)^2)$.

· **Verification Complexity**.

(1) Token generation by each group member:

Each group member is required to compute $r_i P_1$ and $d_i R_A H_3(\hat{e}(D_A, P_2))$, which are essentially scalar multiplications with a computational complexity of $O(k)$.

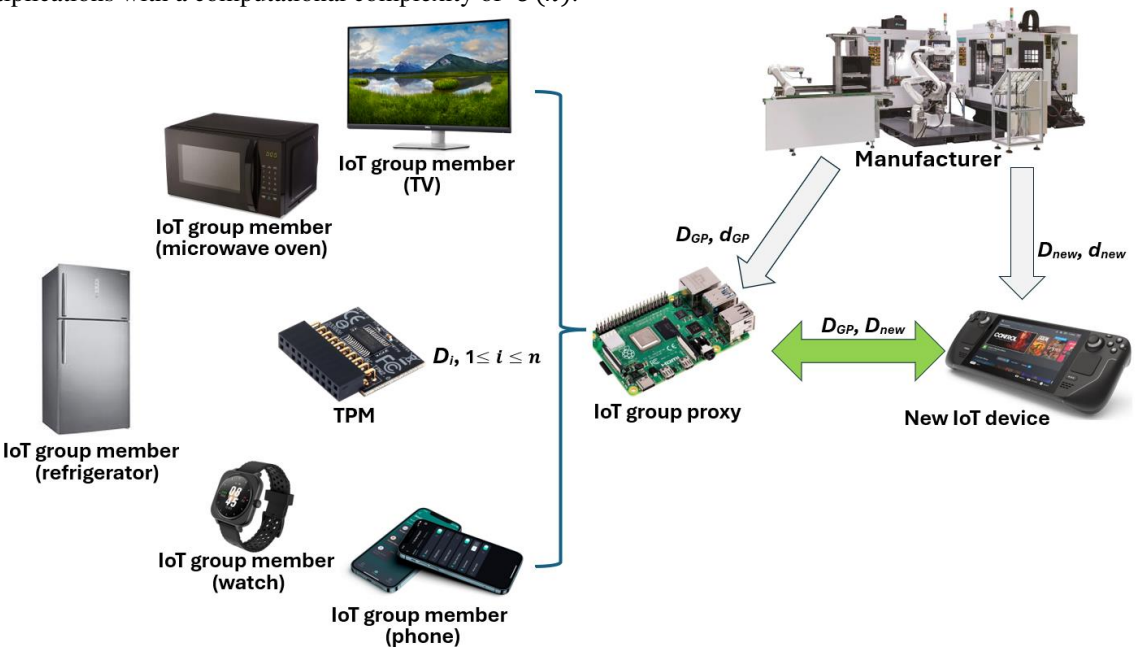

Figure 3 Cryptographic key pairs, generated by the manufacturer and securely embedded in a Trusted Execution Environment (TEE), such as a Trusted Platform Module (TPM), are installed in IoT devices and proxies prior to their shipment from the factory

(2) Aggregation computation

The IoT group proxy aggregates the partial results and calculates the final signature $\widetilde{Sig}'$. The computational complexity of this step involves the product of individual results. Compared to the elliptic curve operations, the hash operation is negligible. Therefore, the total complexity is $O(n^2 klog(k)^2)$.

- **Key agreement Complexity**.

(1) Roun 1 (Public key exchange):

For the first time, each device and IGP calculate the shared secret point $K_{Pi} = d_{GP} D_i = d_i D_{GP} = K_{ip} = (x_{pi}, y_{pi})$. This process involves scalar multiplication, which necessitates a computational complexity of $O(k)$, where k represents the bit length of the scalar.

(2) Roun 2 (Group key computation):

i. Computation by IGP — The IGP computes the combined value O_i , which represents a scalar multiplication operation with a computational complexity of $O(k)$.

ii. Computation by IoT devices — Each device computes the group key P_K by performing scalar multiplication and a hashing operation. The hashing operation is negligible compared to the elliptic curve operation. Consequently, the computational complexity for each device is $O(k)$.

- **Experimental results**

The experiment referenced [7] to utilize the following wearable device for result analysis. As illustrated in Figure 2, these experimental devices encompass smartphones, smartwatches, and IoT development boards. The specifications for each of these devices are detailed in Table 1.

Figure 4 Wearable devices employed in the referenced performance evaluation study

As demonstrated in [7], the results of bilinear pairing operations are evaluated on the jPBC benchmark framework [8]. The selected curves include Type A, defined by the equation $y^2 = x^3 + ax$ over the finite field F_q, and a Type D curve, characterized by an embedding degree of 6 and a prime order. The experimental results for the Type A curve are presented in Figure 3. The Intel Edison, running JDK 1.8.0, emerges as the most efficient device in this study, executing a single pairing operation in approximately 580 milliseconds. However, it is important to note that the initialization phase for pairing on the Intel Edison necessitates more than 8 seconds. Furthermore, the results of pairing-based cryptography (PBC) operations utilizing the Type D curve are illustrated in Figure 4.

Table 1 SELECTED DEVICES WITH THEIR CORRESPONDING SPECIFICATIONS [7]

Type	Device	RAM	SoC	Processor
IoT Dev. Board	Raspberry Pi 1 model B	512MB	BCM2835	700MHz Single-Core ARM Cortex-A6
IoT Dev. Board	Raspberry Pi 2 model B	1GB	BCM2836	900MHz Quad-Core ARM Cortex-A7
IoT Dev. Board	Intel® Edison	1GB	Atom+ Quark	500MHz Dual-Core Intel AtomTM CPU, 100Mhz MCU
Smartphone	Samsung I9500 Galaxy S4	2GB	APQ8064T	1.6GHz Dual-Core Cortex-A15
Smartphone	Jiayu S3 Advanced	3GB	MT6752	1.7GHzOcta-Core 64bit Cortex A53
Smart Watch	Sony SmartWatch 3 SWR50	512MB	BCM47531	1.2GHz Quad-Core ARM A7

The findings suggest that optimized PBC schemes with a minimal number of pairing operations (i.e., fewer than two) can be effectively deployed within the security layers of non-real-time IoT applications running on contemporary smartphones and wearable devices.

Based on the performance depicted in Figures 3 and 4, it is evident that bilinear pairings can be efficiently implemented on devices with limited computational resources or older hardware. This is demonstrated by the relatively low initialization and pairing times observed on smartphones such as the Huawei S3 Advanced and

Samsung Galaxy S4. In IoT networks, where the deployment of cost-effective, low-power devices is common, this efficiency is particularly advantageous. It is demonstrated that the bilinear pairing operation may be used to a variety of Internet of Things devices, ranging from wearables like the Sony Smart Watch 3 to more computationally demanding systems like the Intel Edison and Raspberry Pi models. This adaptability suggests that bilinear pairings can be used in a variety of IoT settings, allowing secure cryptographic operations to take place without being restricted to particular hardware.

Although the proposed identity-based protocol uses bilinear pairing operations, its computational complexity exhibits quadratic growth of $O(n^2)$. However, the rewards are high security and scalability. In IoT systems, security vulnerabilities can lead to significant risks, so the enhanced security provided by such operations is worth the trade-off. Furthermore, as IoT devices advance, the actual performance of the proposed protocol is expected to become increasingly feasible and practical with the advent of more powerful CPUs and larger memories.

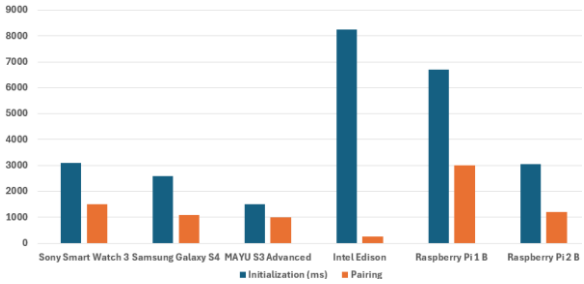

Figure 5 Initialization and Pairing operation time on curve $y^2 = x^3 + x$ over F_q [7]

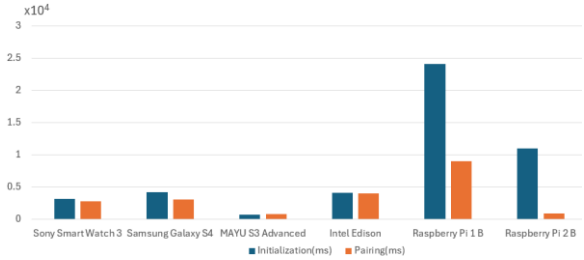

Figure 6 Initialization and Pairing operation time on curve with embedding degree 6 and prime order [7]

6 CONCLUSION AND FUTURE RESEARCH

In this paper, we uncover a critical error in ChatGPT that highlights a significant concern when seeking advanced information from artificial intelligence without a basic understanding of the subject. If this guidance is misused, it can have disastrous consequences. Subsequently, we propose an identity-based, enhanced, and scalable authentication protocol for IoT devices utilizing bilinear pairing. This protocol obviates the need for traditional Public Key Infrastructure (PKI), thereby reducing computational overhead and simplifying key management. Security and performance analyses confirm the protocol's practicality. Future research will be investigating the protocol's performance under different network conditions and adversarial models will provide deeper insights into its practical applicability. The integration of post-quantum cryptographic techniques may also constitute a pivotal area of focus.

ACKNOWLEDGMENTS

This research has received no external funding.

7 HISTORY DATES

In case of submissions being prepared for Journals or PACMs, please add history dates after References as (*please note revised date is optional*):

Received October 2024

REFERENCES

[1] T. S. Nikoui and A. M. Rahmani, "Internet of Things architecture challenges: A systematic review," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 7575-7590, Dec. 2022, doi: 10.1016/j.jksuci.2022.04.001.

[2] Catarinucci L, De Donno D, Mainetti L, et al. An IoT-aware architecture for smart healthcare systems. IEEE Internet Things J. 2015; 2(6):515-526.

[3] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon and R. Guizzetti, "OSCAR: Object security architecture for the Internet of Things," Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, Sydney, NSW, Australia, 2014, pp. 1-10, doi: 10.1109/WoWMoM.2014.6918975.

[4] Tracey D, Sreenan C. A holistic architecture for the Internet of Things, sensing services and big data. In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pages 546-553. IEEE, 2013.

[5] Krawczyk, H., Perfect Forward Secrecy. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA, 2011. doi:10.1007/978-1-4419-5906-5_90.

[6] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. pages 453–474, 2001.

[7] A. Ometov *et al*., "Feasibility characterization of cryptographic primitives for constrained (wearable) IoT devices," *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Sydney, NSW, Australia, 2016, pp. 1-6, doi: 10.1109/PERCOMW.2016.7457161.

[8] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in Proc. of IEEE Symposium on Computers and Communications (ISCC), pp. 850–855, IEEE, 2011.

[9] A. Menezes, "An Introduction to Pairing-Based Cryptography," University of Waterloo. [Online]. Available: https://www.math.uwaterloo.ca/pairings. [Accessed: Aug. 18, 2024].

[10] N. El Mrabet and M. Joye, *Guide to Pairing-Based Cryptography*. Boca Raton, FL, USA: CRC Press, 2016.

[11] R. Bowen, "Elliptic Curve Point Addition," *RareSkills*, Aug. 2023. [Online]. Available: https://www.rareskills.io/post/_elliptic-curve-addition. [Accessed: Aug. 20, 2024].

[12] O. Shevchuk, "Introduction to Elliptic Curve Cryptography," The University of Chicago, 2020. [Online]. Available: https://math.uchicago.edu/REUPapers/Shevchuk. [Accessed: 20-Aug-2024].

[13] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987. doi: 10.2307/2007884.

[14] A. Shamir, "Identity-based cryptosystems and signature schemes," in Advances in Cryptology - CRYPTO '84, Lecture Notes in Computer Science, vol. 196, G. R. Blakley and D. Chaum, Eds. Berlin, Heidelberg: Springer, 1985, pp. 47-53.

[15] D. S. Gupta, S. Ray, T. Singh, and M. Kumari, "Post-quantum lightweight identity-based two-party authenticated key exchange protocol for Internet of Vehicles with probable security," *Computer Communications*, vol. 180, pp. 69-79, 2022.

[16] M. Bagaa, T. Taleb, J. B. Bernabe and A. Skarmeta, "A Machine Learning Security Framework for Iot Systems," in IEEE Access, vol. 8, pp. 114066-114077, 2020, doi: 10.1109/ACCESS.2020.2996214.

[17] P. K. Keserwani, M. C. Govil, E. S. Pilli, et al., "A smart anomaly-based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model," J. Reliable Intell. Environ., vol. 7, no. 1, pp. 3–21, 2021.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[20] S. Siboni et al., "Security Testbed for Internet-of-Things Devices," in IEEE Transactions on Reliability, vol. 68, no. 1, pp. 23-44, March 2019, doi: 10.1109/TR.2018.2864536.

[21] OpenAI, "ChatGPT (GPT-4)," 2023. [Online]. Available: https://www.openai.com/chatgpt. [Accessed: Jun. 31, 2024].

[22] "ChatGPT," *Wikipedia, The Free Encyclopedia*, accessed August 31, 2024. [Online]. Available: https://en.wikipedia.org/wiki/ChatGPT.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. K. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS 2017)*, vol. 30, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[24] M. N. Aman, K. C. Chua and B. Sikdar, "Mutual authentication in IoT systems using physical unclonable functions", IEEE Internet Things J., vol. 4, no. 5, pp. 1327-1340, Oct. 2017.

[25] A. M. A. Modarres and G. Sarbishaei, "An Improved Lightweight Two-Factor Authentication Protocol for IoT Applications," in IEEE Transactions on Industrial Informatics, vol. 19, no. 5, pp. 6588-6598, May 2023, doi: 10.1109/TII.2022.3201971.

[26] G. Sarbishaei, A. Masoud Aminian Modarres, F. Jowshan, F. Zahra Khakzad and H. Mokhtari, "Smart Home Security: An Efficient Multi-Factor Authentication Protocol," in IEEE Access, vol. 12, pp. 106253-106272, 2024, doi: 10.1109/ACCESS.2024.3437294.