*sustainability*

MDPI

*Article*

# Provable Secure Authentication Protocol in Fog-Enabled Smart Home Environment

**Qi Xie *, Jinming Han and Zixuan Ding**

Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China
* Correspondence: qixie68@126.com

**Abstract:** People can access and obtain services from smart home devices conveniently through fog-enabled smart home environments. The security and privacy-preserving authentication protocol play an important role. However, many proposed protocols have one or more security flaws. In particular, almost all the existing protocols for the smart home cannot resist gateway compromised attacks. The adversary can not only know the user's identity but also launch impersonation attacks. Designing a provable secure authentication protocol that avoids all known attacks on smart homes is challenging. Recently Guo et al. proposed an authentication scheme based on symmetric polynomials in the fog-enabled smart home environment. However, we found that their scheme suffers from gateway compromised attack, desynchronization attack, mobile device loss/stolen and attack, and has no untraceability and perfect forward secrecy. Therefore, we adopt a Physical Unclonable Function (PUF) to resist gateway compromised attack, adopt Elliptic Curve Diffie–Hellman (ECDH) key exchange protocol to achieve perfect forward secrecy, and propose a secure and privacy-preserving authentication protocol, which is provably secure under the random oracle model. According to the comparisons with some related protocols, the proposed protocol has better security and transmission efficiency with the same computation cost level.

**Keywords:** authentication protocol; smart home; fog-enabled; privacy-preserving; PUF



**Citation:** Xie, Q.; Han, J.; Ding, Z. Provable Secure Authentication Protocol in Fog-Enabled Smart Home Environment. *Sustainability* **2022**, *14*, 14367. https://doi.org/10.3390/su142114367

Academic Editor: Mouloud Denai

Received: 30 September 2022
Accepted: 29 October 2022
Published: 2 November 2022



## 1. Introduction

In recent years, with the iteration of communication technology and smart devices, the Internet of Things (IoT) has been gradually applied in many aspects, such as logistics, transportation, security, pollutants monitoring, smart home, etc. The smart home is one of the applications of IoT that connect the user and the devices in residence by using a common communication system and control technology [1,2], such as air conditioners, televisions, monitors, water heaters, etc. The new way of controlling devices provided by the smart home brings people safety, energy conservation, comfort, convenience, and healthcare [3].

Due to the smart devices having limited computation and storage, the smart home must have nodes that provide reliable computing services, storage services, and network services to build a communication system [4]. In general, the cloud is more suitable for resource nodes. However, real-time response is a requirement for some emergency applications in the smart home, so nodes also have to meet the requirements of high bandwidth and low latency. Cloud latency is often determined by physical distance, so real-time requirements cannot be met. To meet real-time requirements, distributing computation and storage to edge devices is an idea called fog computing [5].

Fog computing has the characteristics of low latency and high response, and it has been applied in healthcare and smart home [6–9]. Because the computing and storage resources of smart devices in the smart home are limited, they cannot afford much computation. A scheme is proposed in [9] which connects the sensor devices of the terminal based on the IoT controller as a gateway in the smart home. In the fog computing network, the fog layer

composed of the smart gateways undertakes the message forwarding and the distributed computation and storage [10,11].

The fog-computing smart home enhances the user's control over computation and storage nodes and provides more privacy. However, it is still vulnerable to malicious attacks because messages are transmitted on open channels. The data, after being maliciously attacked, will transmit false information, induce users to make wrong decisions, and directly affect residential security and privacy. There are great concerns about the security and privacy of remote access in emergencies and dangers. The security and privacy-preserving authentication protocol play an important role in the smart home.

Until now, many authentication protocols have been proposed. These authentication protocols have some shortcomings in terms of security, anonymity, and perfect forward secrecy [12]. Jeong et al. [13] proposed a user authentication (UA) protocol based on the one-time password (OTP) protocol. The scheme provides authentication of users and gateway, mainly used in remote access to the home network. In the smart home, secure communication between devices and gateways is essential. Xue et al. [14] proposed a temporal-credential-based scheme for Wireless Sensor Networks (WSN) using hash and XOR. Saqib et al. [15] indicated that Xue et al.'s scheme is not immune to smart card theft and server fraud attack. Shuai et al. [16] designed an anonymous authentication scheme based on Elliptic Curve Cryptography (ECC) for the smart home environment. The protocol avoided storing the validation table to reduce the harm caused by theft and resisted replay attacks and clock synchronization attacks. Unfortunately, Kaur et al. [17] pointed out that Shuai et al.'s scheme is vulnerable to offline password guessing attacks, insider attacks, replay attacks, gateway bypass attacks, and insecure session key agreement problems and proposed an improvement scheme for the smart home. However, the scheme is vulnerable to gateway compromised and replay attacks. Santoso et al. [18] proposed a scheme based on ECC for the smart home system. The scheme cannot provide anonymity and untraceability and cannot resist privileged-insider and smart card stolen attacks. Guo et al. [19] presented a new authentication mode based on a symmetric bivariate polynomial [20], which includes the edge negotiation phase and the authentication phase, and reduces communication consumption. The scheme has extremely low computational consumption, but we show it is vulnerable to gateway compromised attacks, desynchronization attacks, mobile device loss/stolen attacks, etc.

Some authentication schemes consider the gateway is trusted and store sensitive information. Wazid et al. [21] proposed a lightweight authentication protocol for the smart home environment based on XOR, symmetric cipher, and hash functions. The authentication table is stored in the gateway. Haseeb-ur-Rehman et al. [22] proposed a lightweight protocol for the smart home and declared that the gateway is trusted. Lee et al. [23] proposed a three-factor authentication protocol in an IoT environment; the gateway is also fully trusted and stores the long-term key. Gateway compromised attacks may lead to the disclosure of user identity, long-term secret values, and other information and lead to suffering impersonation attacks, privilege attacks, etc.

On the other hand, privacy-preserving is a necessary security requirement in the smart home. Yeh et al. [24] proposed an authentication scheme established on Elliptic Curve Cryptography (ECC) for WSN. The message in their scheme contains the real identity of the user, so it does not provide anonymity. In addition, the ECC multiplication is used many times, which makes the protocol have high computational complexity [15]. Yang et al. [25] proposed an ID-based authentication protocol for mobile devices, and the scheme has less computation time because it does not require users' public keys. However, Islam et al. [26] stated that Yang et al.'s scheme has no anonymity.

Perfect forward secrecy (PFS) is an extremely harsh security condition; it is a security feature that can still maintain the confidentiality of previously transmitted messages even if long-term keys are leaked [12]. Although it will increase the computation costs, a better method is to use the Diffie–Hellman key exchange algorithm to design the protocol to guarantee PFS. However, many proposed protocols can not achieve PFS [16,17,19,21,25].

Physical capture attacks often have security implications, and PUF [27] is a way to provide physical device security. A PUF is a one-way function derived from the randomness of physical features caused by the manufacturing variation process. PUFs are unreproducible, unpredictable, and tamper-resistant and are widely used in security. Yi et al. [28] proposed an authentication protocol in WSN. The protocol introduced a PUF chip to provide the physical integrity of sensors. Yu et al. [29] proposed a physically secure privacy-preserving scheme in telecare medical information systems, and PUF is used to store long-term keys. The introduction of PUF to protect devices against physical capture attacks is effective.

*Motivation and Contributions*

According to the analysis of the existing protocols for smart homes, we found that most of them have one or more security flaws, which cannot achieve perfect forward secrecy, privacy protection, etc. In particular, almost all the existing protocols for a smart home cannot resist gateway compromised attacks. The adversary can not only know the user's identity but also launch impersonation attacks. Designing a provable secure authentication protocol that avoids gateway compromised attacks for smart homes is challenging. The contributions of this paper are as follows:

- We pointed out that Guo et al.'s protocol in a fog-enabled smart home is vulnerable to smart gateway compromised attacks, desynchronization attacks, and mobile device lost/stolen attacks, and traceability has no perfect forward secrecy.
- We propose the first secure and privacy-preserving authentication protocol in fog-enabled smart homes to avoid a gateway compromised attack. We adopt PUF to resist gateway compromised attack, adopt ECDH key exchange protocol to achieve perfect forward secrecy, and redesign the process to provide privacy-preserving and makes it resistant to desynchronization attack and mobile device lost/stolen attack.
- We prove the security of the proposed protocol formally under the random oracle model. According to the comparisons with some related protocols, the proposed protocol has better security and transmission efficiency with the same computation cost level.

## 2. System and Attack Models

In this section, we introduce the system and attack models in a fog-enabled smart home environment.

### *2.1. System Model*

Figure 1 shows the system model of Guo et al.'s scheme. The communication system consists of smart devices, users & mobile devices, a smart gateway, and the cloud. The smart device is connected to the smart gateway via the home network, such as Wi-Fi and wired network. The smart gateway is connected to the cloud via the internet, and users can access the smart gateway remotely. The smart gateway provides computing and storage resources in the communication system to ensure real-time data transmission. The smart gateway in this architecture is responsible for collecting from smart devices and processing user requests. The entity negotiation model is shown in Figure 2.

In the edge negotiation stage, the smart gateway and the smart device establish a persistent connection and re-establish the session after the session expires. After the session is established, the gateway can collect real-time data securely from the smart device and fast-forward commands to the smart device from the user.

In addition, the smart gateway will also establish a temporary session with the user. The user negotiates a one-time session key with the gateway after identity verification, accepts the smart device data sent by the gateway, and quickly sends instructions to the smart device via the user's portable mobile device. The above two stages achieve the user's security management of smart devices.

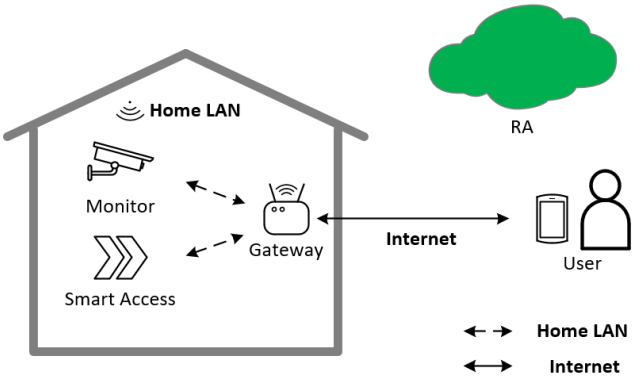


**Figure 1.** The system model of a smart home.

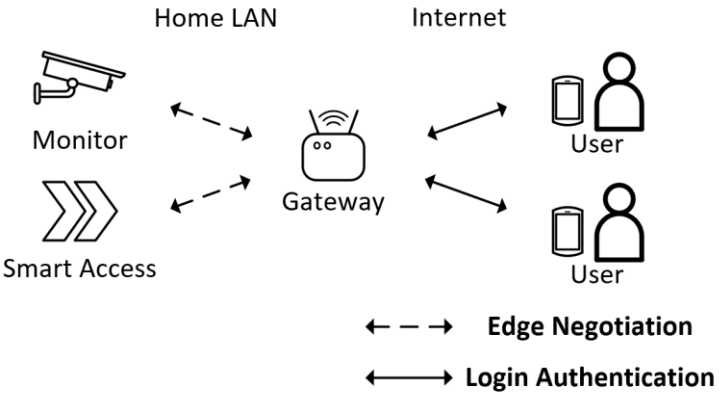


**Figure 2.** The entity negotiation mode.

### *2.2. Attack Model*

We provide an attack model consistent with the original protocol; according to the Dolev-Yao [30] threat model and CK-adversary [31] model in wireless networks, the attacker can compromise information such as session key and session state.

- The attacker can eavesdrop, delay, modify, and delete the messages transmitted on public communication channels;
- The attacker can compromise temporary information such as session key and session state;
- The smart gateway is not considered fully trusted. And it can be compromised by the attacker;
- The smart devices are considered untrusted because the device can be physically lost or stolen, and all data can be extracted;
- The user's mobile device considers an untrusted entity; the mobile device can be captured or compromised by the attacker; the user's mobile devices can be obtained by the attacker;
- Registration authority (RA) is completely credible and cannot be compromised;
- The private channels are secure and cannot be controlled or eavesdropped on by attackers.

## 3. Review of the Guo et al.'s Scheme

In this section, we describe Guo et al.'s scheme. The scheme includes a registration phase, edge negotiation phase, and login authentication phase. The secret sharing of Guo et al.'s scheme is based on symmetric bivariate polynomials. For symmetric bivariate polynomial $func(x, y)$, there is $func(x, y) = func(y, x)$. Table 1 shows the notations used in this paper.

**Table 1.** Notations.

| Notations | Descriptions |
|---|---|
| $U_i$, $GW$, $D_j$, $MD_{U_i}$ | User, gateway, smart home device, and user's mobile device |
| $ID_{U_i}$, $ID_{GW}$, $ID_{D_j}$ | Identities of $U_i$, $GW$, $D_j$ |
| $PID_{U_i}$, $PID_{GW}$, $PID_{D_j}$ | Pseudo identities of $U_i$, $GW$, $D_j$ |
| $TID_{U_i}$, $TID_{D_j}$ | Temporary identities of $U_i$, $D_j$ |
| $RT_{U_i}$, $RT_{GW}$, $RT_{D_j}$ | Registration timestamps of $U_i$, $GW$, $D_j$ |
| $TC_{U_i}$, $TC_{GW}$, $TC_{D_j}$. | Token of $U_i$, $GW$, $D_j$ |
| $PW_{U_i}$, $BIO_{U_i}$ | $U_i$'s password and biometric information |
| $f(\cdot), g(\cdot)$ | Symmetric bivariate polynomial |
| $Rep(\cdot), Gen(\cdot)$ | Reproduction and generation of fuzzy extractor |
| $\sigma_{U_i}$, $\tau_{U_i}$. | $U_i$'s biometric private key and public key of fuzzy extractor |
| $SK$ | Session key |
| $r_{GW}, r_{U_i}, r_{D_j}$ | Random nonce |
| $RA$ | Registration authority |
| $K$ | Private key of $RA$ |
| $T$, $\Delta T$ | Timestamp and maximum transmission delay time |
| $h(\cdot)$ | Hash function |
| $\oplus$,$\|$ | XOR operation and concatenation |

### *3.1. Registration Phase*

$RA$ generates a long-term private key $K$ and chooses two symmetric bivariate polynomials $f(x,y)$ and $g(x,y)$ over the finite field $GF(p)$.

The registration phase includes gateway registration, smart home device registration, and user registration.

(1) Gateway registration

$RA$ chooses a unique identity $ID_{GW}$ for the gateway and computes a pseudo-identity $PID_{GW} = h(ID_{GW} \| K)$, a token $TC_{GW} = h(ID_{GW} \| RT_{GW} \| K)$ and two polynomial functions $f(PID_{GW}, y), g(PID_{GW}, y)$. Finally, $GW$ stores $\{PID_{GW}, (TID_{D_j}^{old} = null, TID_{D_j}^{new} = TID_{D_j}), PID_{D_j}, TID_{U_i}, TC_{GW}, f(PID_{GW}, y), g(PID_{GW}, y), h(\cdot)\}$ in its memory, where $TID_{D_j}^{new}$ is the latest temporary identity of $D_j$, and $TID_{D_j}^{old}$ is the last temporary identity of $D_j$.

(2) Smart home device registration

$RA$ chooses a unique identity $ID_{D_j}$, a temporary identity $TID_{D_j}$ for a smart home device, and computes a pseudo-identity $PID_{D_j} = h(ID_{D_j} \| K)$, the function $g(PID_{D_j}, y)$. Finally, $RA$ stores $\{TID_{D_j}^{old} = null, TID_{D_j}^{new} = TID_{D_j}, PID_{D_j}, g(PID_{D_j}, y), h(\cdot)\}$ in the memory of $D_j$.

(3) User registration

Step UR1: $U_i$ inputs a unique identity $ID_{U_i}$ and biological information $BIO_{U_i}$ into mobile device $MD_{U_i}$. $MD_{U_i}$ generates a nonce $r_{U_i}$, and computes a pseudo-identity $PID_{U_i} = h(ID_{U_i} \| r_{U_i})$, $Gen(BIO_{U_i}) = (\sigma_{U_i}, \tau_{U_i})$ [32], $HPW_{U_i} = h(PW_{U_i} \| \sigma_{U_i} \| r_{U_i})$, $S = HPW_{U_i} \oplus \sigma_{U_i} \oplus r_{U_i}$. The message $\{PID_{U_i}, S\}$ is sent to $RA$ via a private channel.

Step UR2: On receiving the message from $U_i$, $RA$ generates a nonce $R_{U_i}$ and a timestamp $RT_{U_i}$, then calculates a token $TC_{U_i} = h(PID_{U_i} \| RT_{U_i} \| K)$, $A_{U_i} = TC_{U_i} \oplus S$, $B_{U_i} = R_{U_i} \oplus TC_{U_i}$, $C_{U_i} = PID_{GW} \oplus S$. Then $RA$ picks a temporary identity $TID_{U_i}$, calcu-

lates a function $f(PID_{U_i}, y)$, and sends a message $\{A_{U_i}, B_{U_i}, C_{U_i}, TID_{U_i}, f(PID_{U_i}, y), h(\cdot)\}$ to $U_i$ via a secure channel.

Step UR3: Once the message $\{A_{U_i}, B_{U_i}, C_{U_i}, TID_{U_i}, f(PID_{U_i}, y), h(\cdot)\}$ is received, $U_i$ computes $TC_{U_i} = A_{U_i} \oplus S$, $R_{U_i} = B_{U_i} \oplus TC_{U_i}$, $Auth_{U_i} = h(TC_{U_i} \oplus R_{U_i} \oplus HPW_{U_i})$, $D_{U_i} = r_{U_i} \oplus h(ID_{U_i} \oplus PW_{U_i} \oplus \sigma_{U_i})$, $B^*_{U_i} = B_{U_i} \oplus HPW_{U_i}$, $PID^*_{U_i} = PID_{U_i} \oplus HPW_{U_i}$, $TC^*_{U_i} = TC_{U_i} \oplus HPW_{U_i}$, $C^*_{U_i} = C_{U_i} \oplus TC_{U_i}$. Finally, $U_i$ stores $\{TID_{U_i}, Auth_{U_i}, D_{U_i}, B^*_{U_i}, PID^*_{U_i}, TC^*_{U_i}, C^*_{U_i}, f(PID_{U_i}, y), \tau_{U_i}, Rep(\cdot), Gen(\cdot), h(\cdot)\}$ in the memory of $MD_{U_i}$.

In this phase, the user can update password and biometrics information.

### *3.2. Edge Negotiation Phase*

This phase establishes a session key between the gateway and the smart home device. The steps are as follows.

Step EN1: The smart device $D_j$ sends message $\{TID_{D_j}, r_{D_j}, T_1\}$ to the smart gateway $GW$ via the public channel, where $T_1$ is current timestamp and $r_{D_j}$ is a random number.

Step EN2: On receiving the message, $GW$ generates the current timestamp $T_2$, checks the freshness of the timestamp $T_1$, chooses a random number $R_{D_j}$, and finds $PID_{D_j}$ by $TID_{D_j}$. Then $GW$ computes $g(PID_{GW}, PID_{D_j})$, $M_1 = R_{D_j} \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| T_2)$, $M_2 = h(TC_{GW} \| T_2 \| R_{D_j}) \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{GW-D_j} = h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$, $M_3 = h(SK_{GW-D_j} \| r_{D_j} \| R_{D_j} \| T_2)$, $TID^*_{D_j} = TID_{D_j} \oplus SK_{GW-D_j}$. Finally, $GW$ updates $TID^{old}_{D_j} = TID_{D_j}$, $TID^{new}_{D_j} = TID^*_{D_j}$ and sends $\{PID_{GW}, M_1, M_2, M_3, T_2\}$ to $D_j$.

Step EN3: On receiving the message from $GW$, $D_j$ checks the freshness of the timestamp and calculates $g(PID_{D_j}, PID_{GW})$, $R_{D_j} = M_1 \oplus h(g(PID_{D_j}, PID_{GW}) \| r_{D_j} \| T_2)$, $h(TC_{GW} \| T_2 \| R_{D_j}) = M_2 \oplus h(g(PID_{D_j}, PID_{GW}) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{D_j-GW} = h(g(PID_{D_j}, PID_{GW}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$, $M^*_3 = h(SK_{D_j-GW} \| r_{D_j} \| R_{D_j} \| T_2)$. Then $D_j$ checks $M_3 = M^*_3$. If yes, $D_j$ updates $TID^*_{D_j} = TID_{D_j} \oplus SK_{D_j-GW}$ in its memory.

### *3.3. Login Authentication Phase*

When $U_i$ wants to access a smart home device, the user does the following steps.

Step LA1: $U_i$ inputs $ID_{U_i}$, $PW^{in}_{U_i}$ and $BIO^{in}_{U_i}$ into the mobile device $MD_{U_i}$.

Step LA2: $MD_{U_i}$ computes $\sigma'_{U_i} = Rep(BIO^{in}_{U_i}, \tau_{U_i})$, $r'_{U_i} = D_i \oplus h(ID_{U_i} \| PW^{in}_{U_i} \| \sigma'_{U_i})$, $HPW'_{U_i} = h(PW^{in}_{U_i} \| \sigma'_{U_i} \| r'_{U_i})$, $TC'_{U_i} = TC^*_{U_i} \oplus HPW'_{U_i}$, $B'_{U_i} = B^*_{U_i} \oplus HPW'_{U_i}$, $R'_{U_i} = B'_{U_i} \oplus TC'_{U_i}$, $Auth'_{U_i} = h(TC'_{U_i} \oplus R'_{U_i} \oplus HPW'_{U_i})$. If $Auth'_{U_i} = Auth_{U_i}$, do the next step; otherwise, re-do the step LA1.

Step LA3: $MD_{U_i}$ generates a random number $n_{U_i}$ and current timestamp $T_3$ and calculates $PID_{GW} = C^*_{U_i} \oplus TC^*_{U_i} \oplus HPW_{U_i}$, $f(PID_{U_i}, PID_{GW})$, $M_1 = PID_{U_i} \oplus h(PID_{GW})$, $M_2 = n_{U_i} \oplus h(f(PID_{U_i}, PID_{SG}))$, $M_3 = h(M_2 \| TID_{U_i} \| n_{U_i} \| T_3)$, $M_4 = h(TC_{U_i} \| T_3 \| n_{U_i})$. Finally, $MD_{U_i}$ sends message $\{M_1, M_2, M_3, M_4, TID_{U_i}, T_3\}$ to $GW$ publicly.

Step LA4: When $GW$ receives the message from $MD_{U_i}$, $GW$. checks the freshness of the timestamp $T_3$. Then $GW$ computes $PID'_{U_i} = M_1 \oplus h(PID_{GW})$, $n'_{U_i} = M_2 \oplus h(f(PID_{GW}, PID_{U_i}))$, $M'_3 = h(M_2 \| TID_{U_i} \| n'_{U_i} \| T_3)$. If $M'_3 \neq M_3$, terminate. Otherwise, $GW$ generates a random number $N_{U_i}$ and current timestamp $T_4$, and computes $M_5 = N_{U_i} \oplus h(f(PID_{GW}, PID_{U_i}) \| TID_{U_i} \| T_3)$, $M_6 = h(TC_{GW} \| T_4 \| N_{U_i}) \oplus h(f(PID_{GW}, PID_{U_i}) \| M_4 \| T_4)$, $SK_{GW-U_i} = h(TID_{U_i} \| f(PID_{GW}, PID_{U_i}) \| h(TC_{GW} \| T_4 \| N_{U_i}) \| M_4 \| T_4)$, $M_7 = h(SK_{GW-U_i} \| n_{U_i} \| N_{U_i} \| T_3 \| T_4)$. Subsequently, $GW$ chooses a new random number $TID^{new}_{U_i}$ and computes $M_8 = TID^{new}_{U_i} \oplus h(TID_{U_i} \| SK_{GW-U_i} \| n_{U_i} \| N_{U_i} \| T_3 \| T_4)$. Finally, $GW$ sends $\{M_5, M_6, M_7, M_8, T_4\}$ to $MD_{U_i}$.

Step LA5: $MD_{U_i}$ checks the freshness of the timestamp $T_4$ after receiving the message, and calculates $N'_{U_i} = M_5 \oplus h(f(PID_{U_i}, PID_{GW}) \| TID_{U_i} \| T_3)$, $h(TC_{GW} \| T_4 \| N_{U_i}) = M_6 \oplus h(f(PID_{U_i}, PID_{GW}) \| M_4 \| T_4)$, $SK_{U_i-GW} = h(TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| h(TC_{GW} \| T_4 \| N_{U_i}) \| M_4 \| T_4)$, $M^*_7 = h(SK_{U_i-GW} \| n_{U_i} \| N'_{U_i} \| T_3 \| T_4)$. $MD_{U_i}$ checks

$M_7^* = M_7$. If yes, the authentication is successful, $MD_{U_i}$ updates $TID_{U_i}^{new} = M_8 \oplus h(TID_{U_i} \| SK_{U_i-GW} \| n_{U_i} \| N_{U_i} \| T_3 \| T_4)$.

## 4. Cryptanalysis of the Guo et al.'s Scheme

In this section, we show the weaknesses of Guo et al.'s scheme.

### *4.1. Smart Gateway Compromised Attack*

In Guo et al.'s scheme, the smart gateway stores sensitive information and is not fully trusted. Suppose that an attacker compromises the smart gateway and steals information $\{PID_{GW}, (TID_{D_j}^{old} = null, TID_{D_j}^{new} = TID_{D_j}), PID_{D_j}, TID_{U_i}, TC_{GW}, f(PID_{GW}, y), g(PID_{GW}, y), h(\cdot)\}$, each shared secret $g(PID_{GW}, PID_{D_j})$ between $D_j$ and $GW$ can be calculated from $g(PID_{GW}, y)$ and $PID_{D_j}$.

(1) The attacker impersonates the smart device.

The attacker extracts $TID_{D_j}$ from the stolen information and sends $TID_{D_j}$, $r_{D_j}$, $T_1$ to $GW$ in Step EN1, where $r_{D_j}$ is a random number and $T_1$ is a current timestamp. In Step EN3, the attacker gets $PID_{GW}, M_1, M_2, M_3, T_2$ from $GW$, picks $PID_{D_j}$, $g(PID_{GW}, y)$ from the stolen information and can calculate $g(PID_{GW}, PID_{D_j})$, $R_{D_j} = M_1 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| T_2)$, $h(TC_{GW} \| T_2 \| R_{D_j}) = M_2 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{GW-D_j} = h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$.

(2) The attacker eavesdrops on the smart device.

The attacker eavesdrops $TID_{D_j}$, $r_{D_j}$, $T_1$ in Step EN1 and $PID_{GW}$, $M_1$, $M_2$, $M_3$, $T_2$ in Step EN2, and picks $TID_{D_j}, PID_{D_j}$, $g(PID_{GW}, y)$ from the stolen information, then calculates $g(PID_{GW}, PID_{D_j})$, $R_{D_j} = M_1 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| T_2)$, $h(TC_{GW} \| T_2 \| R_{D_j}) = M_2 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| R_{D_j} \| T_2)$ and $SK_{GW-D_j} = h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$ in Step EN3. The attacker gets $SK_{GW-D_j}$ and $TID_{D_j}^* = TID_{D_j} \oplus SK_{GW-D_j}$ successfully. The attacker can also eavesdrop on the next session by $TID_{D_j}^*$.

### *4.2. Desynchronization Attack*

The attacker intercepts and modifies $r_{D_j}$ in Step EN1. In Step EN2, the smart gateway will receive the tampered $r_{D_j}$ and calculate $M_1 = R_{D_j} \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| T_2)$, $M_2 = h(TC_{GW} \| T_2 \| R_{D_j}) \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{GW-D_j} = h(g\left(PID_{GW}, PID_{D_j}\right) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$, $M_3 = h(SK_{GW-D_j} \| r_{D_j} \| R_{D_j} \| T_2)$, $TID_{D_j}^* = TID_{D_j} \oplus SK_{GW-D_j}$ by the tampered $r_{D_j}$. In Step EN3, the smart device calculates $R_{D_j} = M_1 \oplus h(g(PID_{D_j}, PID_{GW}) \| r_{D_j} \| T_2)$, $h(TC_{GW} \| T_2 \| R_{D_j}) = M_2 \oplus h(g\left(PID_{D_j}, PID_{GW}\right) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{D_j-GW} = h(g(PID_{D_j}, PID_{GW}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$, $M_3^* = h(SK_{D_j-GW} \| r_{D_j} \| R_{D_j} \| T_2)$ by the real $r_{D_j}$, so $M_3^* \neq M_3$ and $D_j$ won't update $TID_{D_j}$ in Step EN3.

### *4.3. Traceability*

In the login authentication phase, $M_1 = PID_{U_i} \oplus h(PID_{GW})$, where $PID_{U_i}$ and $PID_{GW}$ are constant, and $M_1$ doesn't change. Therefore, the attacker can trace $U_i$ by eavesdropping on the constant value $M_1$.

### *4.4. Mobile Device Lost/Stolen Attack*

Assuming that $U_i$'s mobile device is lost or stolen, all the information $\{TID_{U_i}, Auth_{U_i}, D_{U_i}, B_{U_i}^*, PID_{U_i}^*, TC_{U_i}^*, C_{U_i}^*, f(PID_{U_i}, y), \tau_{U_i}, Rep(\cdot), Gen(\cdot), h(\cdot)\}$ stored in the mobile device will be obtained by the attacker. The attacker eavesdrops $PID_{GW}$ in Step EN2 and

$M_1$ in Step LA3, and computes $PID_{U_i} = M_1 \oplus h(PID_{GW})$. The attacker can perform the following attacks.

(1) The attacker impersonates the user.

The attacker picks $f(PID_{U_i}, y)$, $TID_{U_i}$. and $PID_{GW}$, generates a random number $n_{U_i}$, a fake token $TC_{U_i}$, a current timestamp $T_3$, and calculates $f(PID_{U_i}, PID_{GW})$, $M_1 = PID_{U_i} \oplus h(PID_{GW})$, $M_2 = n_{U_i} \oplus h(f(PID_{U_i}, PID_{GW}))$, $M_3 = h(M_2 \| TID_{U_i} \| n_{U_i} \| T_3)$ and $M_4 = h(TC_{U_i} \| T_3 \| n_{U_i})$. Finally, the attacker sends $M_1$, $M_2$, $M_3$, $M_4$, $TID_{U_i}$, $T_3$ in Step LA3. In Step LA5, the attacker calculates $N'_{U_i} = M_5 \oplus h(f(PID_{U_i}, PID_{GW}) \| TID_{U_i} \| T_3)$, $h(TC_{GW} \| T_4 \| N_{U_i}) = M_6 \oplus h(f(PID_{U_i}, PID_{GW}) \| M_4 \| T_4)$, $SK_{U_i-GW} = h(TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| h(TC_{GW} \| T_4 \| N_{U_i}) \| M_4 \| T_4)$.

(2) The attacker eavesdrops on the user.

The attacker picks $f(PID_{U_i}, y)$, $TID_{U_i}$ and $PID_{GW}$ and intercepts $M_2$, $M_4$, $T_3$ in Step LA3, calculates $f(PID_{U_i}, PID_{GW})$, $n_{U_i} = M_2 \oplus h(f(PID_{U_i}, PID_{GW}))$. Then the attacker eavesdrops $M_5$, $M_6$, $M_8$, $T_4$ in Step LA4 and calculates $N_{U_i} = M_5 \oplus h(f(PID_{U_i}, PID_{GW}) \| TID_{U_i} \| T_3)$, $h(TC_{GW} \| T_4 \| N_{U_i}) = M_6 \oplus h(f(PID_{U_i}, PID_{GW}) \| M_4 \| T_4)$, $SK_{U_i-GW} = h(TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| h(TC_{GW} \| T_4 \| N_{U_i}) \| M_4 \| T_4)$, $TID^{new}_{U_i} = M_8 \oplus h(TID_{U_i} \| SK_{U_i-GW} \| n_{U_i} \| N_{U_i} \| T_3 \| T_4)$ in Step LA5. The next session key can be calculated in the same way.

### *4.5. No Perfect Forward Secrecy*

In Guo et al.'s scheme, $f(PID_{U_i}, PID_{GW})$ and $g(PID_{GW}, PID_{D_j})$ are long-term for the session key agreement.

In the login and authentication phase, if $f(PID_{U_i}, PID_{GW})$ leaks, the attacker can eavesdrop $M_2$, $M_4$, $T_3$, $TID_{U_i}$ in Step LA3 and $M_5$, $M_6$, $M_8$, $T_4$ in Step LA4, then calculate $n_{U_i} = M_2 \oplus h(f(PID_{U_i}, PID_{GW}))$, $N_{U_i} = M_5 \oplus h(f(PID_{U_i}, PID_{GW}) \| TID_{U_i} \| T_3)$, $h(TC_{GW} \| T_4 \| N_{U_i}) = M_6 \oplus h(f(PID_{GW}, PID_{U_i}) \| M_4 \| T_4)$, $SK_{GW-U_i} = h(TID_{U_i} \| f(PID_{GW}, PID_{U_i}) \| h(TC_{GW} \| T_4 \| N_{U_i}) \| M_4 \| T_4)$. In the same way, the attacker can calculate the previous session key.

In the edge negotiation phase, If $g(PID_{GW}, PID_{D_j})$ leaks, the attacker can compute $R_{D_j} = M_1 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| T_2)$, $h(TC_{GW} \| T_2 \| R_{D_j}) = M_2 \oplus h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| R_{D_j} \| T_2)$, $SK_{GW-D_j} = h(g(PID_{GW}, PID_{D_j}) \| r_{D_j} \| h(TC_{GW} \| T_2 \| R_{D_j}) \| T_2)$, where $r_{D_j}$ can be obtained in Step EN1 and $M_1$, $M_2$, $T_2$ can be obtained in Step EN2. The previous session key can also be calculated in the same way.

## 5. The Proposed Scheme

In this section, we propose a security-enhanced scheme. The scheme consists of a system initialization phase, entity registration phase, edge negotiation phase, and login authentication phase.

In the proposed scheme, the PUF can improve security. PUF is a one-way function derived from complex physical and environmental characteristics. When a challenge stimulates the device, the device calculates a response from the complex physical functions, and the response is unpredictable and repeatable. The attackers cannot predict the response to the challenge and build the same PUF based on the same design and blueprint [27].

### *5.1. System Initialization Phase*

$RA$ generates private key $K$ and two symmetric bivariate polynomials $f(x, y)$ and $g(x, y)$ with degree $\tau$ over the field $GF_p$, and selects the elliptic curve $E_p(a, b)$ over finite field $F_p$, $G$ is the base point.

### *5.2. Entity Registration Phase*

In this phase, we use the PUF to protect the private data of the smart gateway and the smart devices. Figures 3–5 describe the registration processes.

| ***GW*** | ***RA*** |
|---|---|
| ***Step GR 1***<br>*Generates a challenge* $Ch_{GW}$*, computes*<br>$ID_{GW} = PUF(Ch_{GW})$.<br>*Sends* $ID_{GW}$ *to* $RA$ via *private channel.* → | |
| | ***Step GR 2***<br>*Generates current timestamp* $RT_{GW}$*, computes*<br>$PID_{GW} = h(ID_{GW} \parallel RT_{GW} \parallel K)$.<br>*Sends* $PID_{GW}, RT_{GW}$ *to* $GW$ via *private channel.* ← |
| ***Step GR 3***<br>*Computes*<br>$SID_{GW} = PID_{GW} \oplus h(Ch_{GW} \parallel ID_{GW})$.<br>*Stores* $SID_{GW}, Ch_{GW}, RT_{GW}, PUF, h, G$. | |

**Figure 3.** Summary of smart gateway registration.

| ***SD$_j$*** | ***RA*** | ***GW*** |
|---|---|---|
| ***Step DR 1***<br>*Sends a unique* $ID_{D_j}$ *to* $RA$ via *private channel.* → | | |
| | ***Step DR 2***<br>*Generates current timestamp* $RT_{D_j}$ *and* $TID_{D_j}$*, computes*<br>$PID_{D_j} = h(ID_{D_j} \parallel RT_{D_j} \parallel K)$,<br>$g(PID_{GW}, y)$,<br>$g(PID_{D_j}, y)$.<br>*Sends* $<TID_{D_j}, g(PID_{D_j}, y)>$ *to* $GW$ via *private channel.* → | |
| | *Sends* $g(PID_{GW}, y)$, $PID_{D_j}$, $RT_{D_j}$, $TID_{D_j}$ *to* $SD_j$ via *private channel.* ← | |
| ***Step DR 4***<br>*Generates a challenge* $Ch_{D_j}$*,computes*<br>$DID_{D_j} = PID_{D_j} \oplus h(PUF(Ch_{D_j}))$<br>*Stores* $TID_{D_j}, DID_{D_j}, Ch_{D_j}, g(PID_{GW}, y), PUF, h, G$. | | ***Step DR 3***<br>*Stores* $<TID_{D_j}, g(PID_{D_j}, y)>$. |

**Figure 4.** Summary of smart device registration.

| ***$U_i$*** | ***RA*** | ***GW*** |
|---|---|---|
| ***Step UR 1***<br>*Picks a unique* $ID_{U_i}$*, a nonce* $r_{U_i}$*, computes*<br>$PID_{U_i} = h(ID_{U_i} \parallel r_{U_i})$.<br>*Sends* $PID_{U_i}$ *to* $RA$ via *private channel.* → | | |
| | ***Step UR 2***<br>*Generates current timestamp* $RT_{U_i}$ *and* $TID_{U_i}$*, computes*<br>$TC_{U_i} = h(PID_{U_i} \parallel RT_{U_i} \parallel K)$,<br>$f(PID_{U_i}, y)$,<br>$f(PID_{GW}, y)$.<br>*Sends* $<TID_{U_i}, f(PID_{U_i}, y)>$ *to* $GW$ via *private channel.* → | |
| | *Sends* $TID_{U_i}, TC_{U_i}, f(PID_{GW}, y)$ *to* $U_i$ via *private channel.* ← | |
| ***Step UR 4***<br>*Inputs* $BIO_{U_i}$ *and* $PW_{U_i}$*, computes*<br>$(\sigma_{U_i}, \tau_{U_i}) = Gen(BIO_{U_i})$,<br>$HPW_{U_i} = h(ID_{U_i} \parallel PW_{U_i} \parallel \sigma_{U_i} \parallel r_{U_i})$,<br>$B_{U_i} = r_{U_i} \oplus h(PW_{U_i} \parallel \sigma_{U_i})$.<br>*Stores* $B_{U_i}, TID_{U_i}, TC_{U_i}, \tau_{U_i}, HPW_{U_i}, f(PID_{GW}, y), Rep, Gen, h, G$. | | ***Step UR 3***<br>*Stores* $<TID_{U_i}, f(PID_{U_i}, y)>$. |

**Figure 5.** Summary of user registration.

5.2.1. Smart Gateway Registration

Step GR1: The smart gateway $GW$ picks a $PUF$ and generates a challenge $Ch_{GW}$, then computes $ID_{GW} = PUF(Ch_{GW})$. The smart gateway sends the message $ID_{GW}$ to the $RA$ via the private channel.

Step GR2: After receiving the message $ID_{GW}$, $RA$ generates the current timestamp $RT_{GW}$, and computes $PID_{GW} = h(ID_{GW} \parallel RT_{GW} \parallel K)$. $RA$ returns a response $PID_{GW}, RT_{GW}$ via a private channel.

Step GR3: After receiving the response, the smart gateway calculates $SID_{GW} = PID_{GW} \oplus h(Ch_{GW} \parallel ID_{GW})$ and stores $SID_{GW}$, $Ch_{GW}$, $RT_{GW}$, $PUF$, $h$, $G$ in the memory.

5.2.2. Smart Device Registration

Step DR1: The smart device $SD_j$ sends a unique identity $ID_{D_j}$ to $RA$ via the private channel.

Step DR2: After receiving the message, $RA$ generates a random number $TID_{D_j}$, current registration timestamp $RT_{D_j}$, then calculates $PID_{D_j} = h(ID_{D_j} \parallel RT_{D_j} \parallel K)$ and two functions $g(PID_{GW}, y)$, $g(PID_{D_j}, y)$ from the symmetric bivariate polynomial $g(x, y)$, $PID_{GW}$ and $PID_{D_j}$. Then $RA$ sends the message $<TID_{D_j}, g(PID_{D_j}, y)>$ to $GW$ and the message $\left\langle g(PID_{GW}, y), PID_{D_j}, RT_{D_j}, TID_{D_j} \right\rangle$ to $SD_j$ via the private channel.

Step DR3: $GW$ stores the message $<TID_{D_j}, g(PID_{D_j}, y)>$ in its memory.

Step DR4: The smart device $SD_j$ chooses a $PUF$ and generates a challenge $Ch_{D_j}$, and computes $DID_{D_j} = PID_{D_j} \oplus h(PUF(Ch_{D_j}))$. Finally, $SD_j$ stores $\{TID_{D_j}, DID_{D_j}, Ch_{D_j}, g(PID_{GW}, y), PUF, h, G\}$ in its memory.

5.2.3. User Registration

Step UR1: $U_i$ selects a unique identity $ID_{U_i}$, a random number $r_{U_i}$, computes $PID_{U_i} = h(ID_{U_i} \parallel r_{U_i})$, and sends $PID_{U_i}$ to $RA$ via the private channel.

Step UR2: $RA$ generates a random $TID_{U_i}$, a current registration timestamp $RT_{U_i}$, two functions $f(PID_{U_i}, y)$ and $f(PID_{GW}, y)$ by the symmetric bivariate polynomial $f(x, y)$, and computes $TC_{U_i} = h(PID_{U_i} \parallel RT_{U_i} \parallel K)$. Then $RA$ sends $<TID_{U_i}, f(PID_{U_i}, y)>$ to $GW$ and $TID_{U_i}, TC_{U_i}, f(PID_{GW}, y)$ to $U_i$ via the private channel.

Step UR3: $GW$ stores $<TID_{U_i}, f(PID_{U_i}, y)>$ in its memory.

Step UR4: $U_i$ inputs $BIO_{U_i}$, $PW_{U_i}$, and computes $(\sigma_{U_i}, \tau_{U_i}) = Gen(BIO_{U_i})$, where $BIO_{U_i}$ is $U_i$'s biological information. Later $U_i$ computes $HPW_{U_i} = h(ID_{U_i} \parallel PW_{U_i} \parallel \sigma_{U_i} \parallel r_{U_i})$, $B_{U_i} = r_{U_i} \oplus h(PW_{U_i} \parallel \sigma_{U_i})$, where $PW_{U_i}$ is the password. Finally, $U_i$ stores $\{B_{U_i}, TID_{U_i}, TC_{U_i}, \tau_{U_i}, HPW_{U_i}, f(PID_{GW}, y), Rep, Gen, h, G\}$ in the memory of a mobile device.

### *5.3. Edge Negotiation Phase*

In the edge negotiation phase, the session key is negotiated between the smart gateway and the smart devices. Figure 6 describes the executive process.

Step EN1: The smart device $SD_j$ generates a random number $r_{D_j}$, current timestamp $T_1$, and computes $PID_{D_j} = DID_{D_j} \oplus h(PUF(Ch_{D_j}))$, the secret value $g(PID_{GW}, PID_{D_j})$, $M_1 = r_{D_j} \cdot G, V_1 = h(M_1 \parallel TID_{D_j} \parallel g(PID_{GW}, PID_{D_j}) \parallel T_1)$, and sends $\{TID_{D_j}, M_1, V_1, T_1\}$ to $GW$ via the public channel, where $G$ is the base point.

| $SD_j$ | $GW$ |
|---|---|
| $TID_{D_j}, DID_{D_j}, Ch_{D_j}, g(PID_{GW}, y), PUF, h, G$ | $SID_{GW}, Ch_{GW}, RT_{GW}, PUF, h, G$<br>$<TID_{D_j}, g(PID_{D_j}, y)>$<br>$<TID_{U_i}, f(PID_{U_i}, y)>$ |
| ***Step EN1***<br>*Generates random number* $r_{D_j}$*, current timestamp* $T_1$*, and computes*<br>$PID_{D_j} = DID_{D_j} \oplus h(PUF(Ch_{D_j}))$,<br>$M_1 = r_{D_j}.G$,<br>$V_1 = h(M_1 \| TID_{D_j} \| g(PID_{GW}, PID_{D_j}) \| T_1)$.<br>*Sends* $TID_{D_j}, M_1, V_1, T_1$ via *public channel.* → | |
| | ***Step EN2***<br>*Checks if* $\|T_1^{'} - T_1\| \leq \Delta TS$*. If so generates random number* $R_{D_j}$*, current timestamp* $T_2$*, computes*<br>$ID_{GW} = PUF(Ch_{GW})$,<br>$PID_{GW} = SID_{GW} \oplus h(Ch_{GW} \| ID_{GW})$.<br>*Finds* $g(PID_{D_j}, y)$ *by* $TID_{D_j}$*, and computes*<br>$V_1^{'} = h(M_1 \| TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| T_1)$.<br>*If* $V_1 \neq V_1^{'}$ *exit, else computes*<br>$M_2 = R_{D_j}.G$,<br>$SK_{GW-D_j} = h(TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| R_{D_j}.M_1 \| T_1 \| T_2)$,<br>$TID_{D_j}^{new} = h(TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| R_{D_j}.M_1 \| T_2)$,<br>$V_2 = h(TID_{D_j}^{new} \| T_1)$.<br>*When session is established, updates* $TID_{D_j} = TID_{D_j}^{new}$.<br>*Sends* $M_2, V_2, T_2$ via *public channel.* ← |
| ***Step EN3***<br>*Checks if* $\|T_2^{'} - T_2\| \leq \Delta TS$*, if so, computes*<br>$SK_{D_j-GW} = h(TID_{D_j} \| g(PID_{GW}, PID_{D_j}) \| r_{D_j}.M_2 \| T_1 \| T_2)$,<br>$TID_{D_j}^{new} = h(TID_{D_j} \| g(PID_{GW}, PID_{D_j}) \| r_{D_j}.M_2 \| T_2)$,<br>$V_2^{'} = h(TID_{D_j}^{new} \| T_1)$.<br>*If* $V_2 = V_2^{'}$ *negotiates success, updates* $TID_{D_j} = TID_{D_j}^{new}$. | |

**Figure 6.** Summary of edge negotiation phase.

Step EN2: On receiving the request, $GW$ checks the freshness of the timestamp $T_1$. After that, $GW$ finds the function $g(PID_{D_j}, y)$ by $TID_{D_j}$, and calculates $ID_{GW} = PUF(Ch_{GW})$, $PID_{GW} = SID_{GW} \oplus h(Ch_{GW} \| ID_{GW})$, the secret value $g(PID_{D_j}, PID_{GW})$, $V_1' = h(M_1 \| TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| T_1)$. If $V_1 = V_1'$, the request is integrity. Then $GW$ generates a random number $R_{D_j}$ and timestamp $T_2$, and calculates $M_2 = R_{D_j} \cdot G$, $SK_{GW-D_j} = h(TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| R_{D_j} \cdot M_1 \| T_1 \| T_2)$, $TID_{D_j}^{new} = h(TID_{D_j} \| g(PID_{D_j}, PID_{GW}) \| R_{D_j} \cdot M_1 \| T_2)$, $V_2 = h(TID_{D_j}^{new} \| T_1)$. Then $GW$ returns $M_2$, $V_2$, $T_2$ via the public channel. It is worth noting that the $GW$ does not immediately update $TID_{D_j}$ at the time, and updates $TID_{D_j} = TID_{D_j}^{new}$ after a secure session is established. Even if the session establishment fails, it will not cause the smart device $SD_j$ to get out of sync.

Step EN3: On receiving the response, the smart device $SD_j$ checks the freshness of timestamp $T_2$. Then $SD_j$ calculates $SK_{D_j-GW} = h(TID_{D_j} \| g(PID_{GW}, PID_{D_j}) \| r_{D_j} \cdot M_2 \| T_1 \| T_2)$, $V_2' = h(TID_{D_j}^{new} \| T_1)$. If $V_2 = V_2'$, the negotiation is successful, and $SD_j$ calculates $TID_{D_j}^{new} = h(TID_{D_j} \| g(PID_{GW}, PID_{D_j}) \| r_{D_j} \cdot M_2 \| T_2)$, updates $TID_{D_j} = TID_{D_j}^{new}$ and sends a secure message to notify the $GW$ to update $TID_{D_j}$.

### *5.4. Login and Authentication Phase*

When the user needs to access the smart home, Figure 7 describes the executive process.

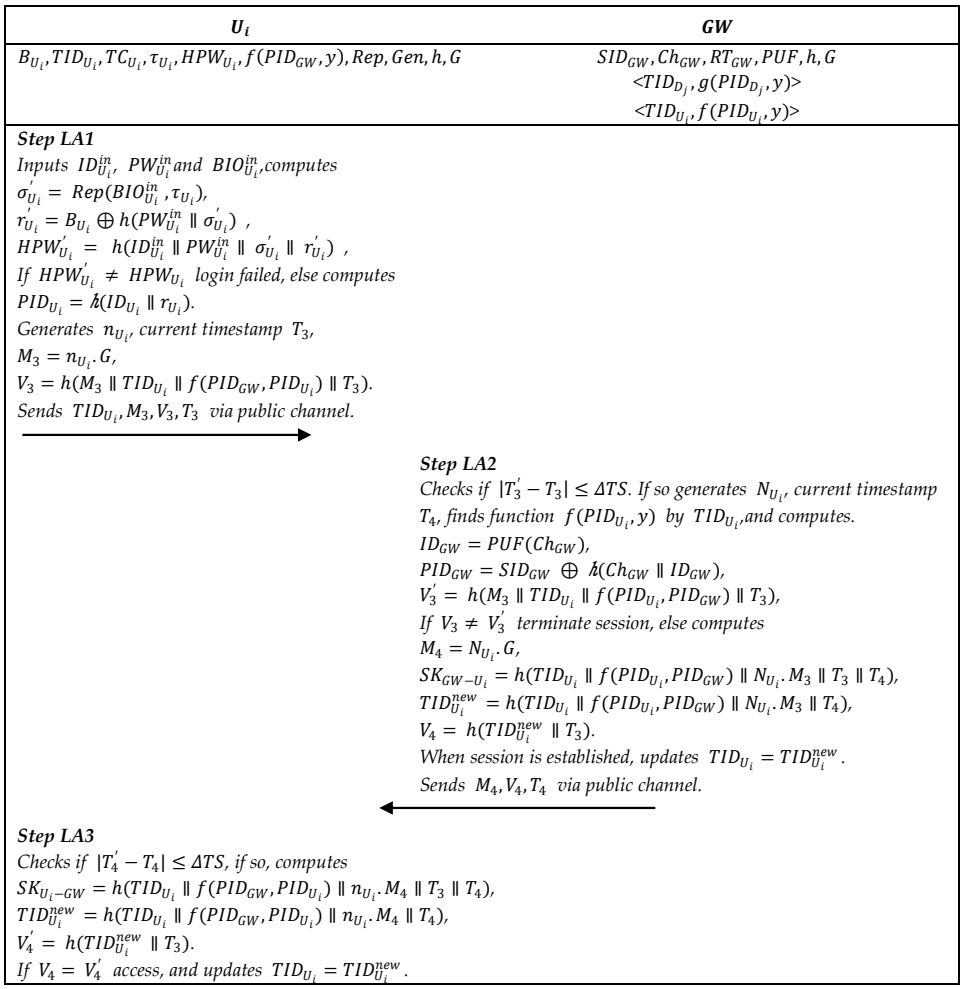


**Figure 7.** Summary of login and authentication phase.

Step LA1: The user $U_i$ inputs identity information $ID_{U_i}^{in}$, $PW_{U_i}^{in}$ and biometric information $BIO_{U_i}^{in}$ into the mobile device, the mobile device computes $\sigma'_{U_i} = Rep(BIO_{U_i}^{in}, \tau_{U_i})$, $r'_{U_i} = B_{U_i} \oplus h(PW_{U_i}^{in} \| \sigma'_{U_i})$, $HPW'_{U_i} = h(ID_{U_i}^{in} \| PW_{U_i}^{in} \| \sigma'_{U_i} \| r'_{U_i})$. If $HPW'_{U_i} \neq HPW_{U_i}$, login failed. Otherwise, the mobile device generates a random number $n_{U_i}$ and current timestamp $T_3$, computes $PID_{U_i} = h(ID_{U_i} \| r_{U_i})$, $M_3 = n_{U_i} \cdot G$, $V_3 = h(M_3 \| TID_{U_i} \| f(PID_{GW}, PID_{U_i}) \| T_3)$. Then the mobile device sends $TID_{U_i}, M_3, V_3, T_3$ to the $GW$ via the public channel.

Step LA2: After receiving the message, $GW$ checks the freshness of the timestamp $T_3$. Then $GW$ extracts the information $ID_{GW} = PUF(Ch_{GW})$, $PID_{GW} = SID_{GW} \oplus h(Ch_{GW} \| ID_{GW})$, gets $f(PID_{U_i}, y)$ by $TID_{U_i}$ and calculates the secret value $f(PID_{U_i}, PID_{GW})$. Similarly, the $GW$ calculates $V'_3 = h(M_3 \| TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| T_3)$. If $V_3 \neq V'_3$, terminate. Otherwise, $GW$ generates random number $N_{U_i}$ and current timestamp $T_4$, calculates $M_4 = N_{U_i} \cdot G$, $SK_{GW-U_i} = h(TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| N_{U_i} \cdot M_3 \| T_3 \| T_4)$, $TID_{U_i}^{new} = h(TID_{U_i} \| f(PID_{U_i}, PID_{GW}) \| N_{U_i} \cdot M_3 \| T_4)$, $V_4 = h(TID_{U_i}^{new} \| T_3)$, and updates the $TID_{U_i}$ with $TID_{U_i}^{new}$ after the session is established. Finally, $GW$ sends the message $M_4$, $V_4$, $T_4$ to $U_i$ via the public channel.

Step LA3: On receiving the response, the user $U_i$'s mobile device checks the freshness of the timestamp $T_4$. And the mobile device computes $SK_{U_i-GW} = h(TID_{U_i} \| f(PID_{GW}, PID_{U_i}) \| n_{U_i} \cdot M_4 \| T_3 \| T_4)$, $V'_4 = h(TID_{U_i}^{new} \| T_3)$, $TID_{U_i}^{new} = h(TID_{U_i} \| f(PID_{GW}, PID_{U_i}) \| n_{U_i} \cdot M_4 \| T_4)$. If $V_4 = V'_4$, the authentication succeeded, and the mobile device updates $TID_{U_i}$ with $TID_{U_i}^{new}$ and securely informs the smart gateway $GW$ to update $TID_{U_i}$.

## 6. Formal Security Proof

### *6.1. Random Oracle Model*

**Definition 1** (Participants & partnering)**.** *The participants are composed of User (U), Smart Device (SD), and Gateway (GW). In the i-th instance, the participants are denoted as* $In^i_{Ui}$, $In^i_{SDj}$, *and* $In^i_{GW}$, *respectively. The state Accept represents that an oracle receives a correct message.*

If two oracles are in *Accept* and the session keys have been agreed upon, the oracles get their session identities and participant identities. The oracles can be considered partners if the following conditions are satisfied:

Their session keys are the same;

Their session identities are the same;

The participant's identity is equal to each other's identity.

**Definition 2** (Queries)**.** *The queries simulate the capabilities of attackers*.

$Execute(In^i_{Ui}, In^i_{GW}, In^i_{SDj})$: All the messages transmitted openly can be intercepted by the adversary $A$.

$Send(In^i_{Ui}, In^i_{GW}, In^i_{SDj}, m)$: $A$ forges and sends the message $m$ to $In^i_{Ui}$, $In^i_{GW}$, or $In^i_{SDj}$, if $m$ is correct, $In^i_{Ui}$, $In^i_{GW}$, or $In^i_{SDj}$ responses $A$.

$Reveal(In^i_{Ui}, In^i_{GW}, In^i_{SDj})$: $A$ can get the current session key between $In^i_{Ui}$, $In^i_{GW}$, and $In^i_{SDj}$.

$Test(In^i_{Ui}, In^i_{GW}, In^i_{SDj}, r)$: This query is allowed to be executed at most once, which generates a random bit $r$, if $r = 1$, the real session key is returned.

$CorruptUser(In^i_{Ui})$: Which simulates the side-channel attack on the user's device and returns the stored information $\{B_{U_i}, TID_{U_i}, TC_{U_i}, \tau_{U_i}, HPW_{U_i}, f(PID_{GW}, y), Rep, Gen, h, G\}$.

$CorruptDevice(In^i_{SDj})$: Which simulates the attack of capturing a smart device and returns the stored information $\{TID_{D_j}, DID_{D_j}, Ch_{D_j}, g(PID_{GW}, y), PUF, h, G\}$.

$CorruptGateway(In^i_{GW})$: Which simulates the attack of capturing the smart gateway and returns the stored information $\{SID_{GW}, TC_{GW}, Ch_{GW}, RT_{GW}, PUF, h, G\}$, $<TID_{D_j}, g(PID_{D_j}, y)>$, and $<TID_{U_i}, f(PID_{U_i}, y)>$.

**Definition 3** (Freshness)**.** *An instance can be regarded as fresh if it satisfies:*

$In^i_{Ui}$, $In^i_{GW}$, *and* $In^i_{SDj}$ *are in* $Accept$.

*The query* $Reveal(In^i_{Ui}, In^i_{GW}, In^i_{SDj})$ *has not been executed.*

*The queries* $Corrupt$ *have been executed at most once.*

**Definition 4** (Semantic security)**.** *A is allowed to execute at most once* $Test(In^i_{Ui}, In^i_{GW}, In^i_{SDj}, r)$ *and multiple other queries to determine the correctness of the return value of* $Test(In^i_{Ui}, In^i_{GW}, In^i_{SDj}, r)$. *That is A guesses the random bit r generated by Test. The possibility is* $Adv^A_P = |2Pr[suc(A)] - 1|$, $Adv^A_P < \eta$ *represents the protocol is secure, where* $\eta$ *is sufficiently small.*

### *6.2. Formal Security Proof under the Random Oracle Model*

**Theorem 1.** *The advantage of obtaining the session key in polynomial time by A is* $Adv^A_P \leq \frac{q^2_{HA}}{2^{l_{HA}}} + \frac{(q_{SE}+q_{EX})^2}{n} + \frac{q_{SE}}{2^{l_{bio}-1}} + 2q_{SE}Adv^A_{PUF} + 2Adv^A_{ECDLP} \cdot Adv^A_{SBP}$.

*Where* $q_{HA}$, $q_{SE}$, *and* $q_{EX}$ *represents the times of executing Hash, Send, and Execute, respectively.* $l_{HA}$, $n$, *and* $l_{bio}$ *are the length of hash, transcripts, and biological key, respectively. The advantage of breaking PUF, ECDLP, and the symmetric bivariate polynomial by A are* $Adv^A_{PUF}$, $Adv^A_{ECDLP}$, *and* $Adv^A_{SBP}$, *respectively.*

**Proof.** The games $Game_i(0 \leq i \leq 4)$ are defined to simulate the attacks launched by $A$. $Win_i(0 \leq i \leq 4)$ means $A$ guesses the random bit $r$ in the $Game_i$. The games are defined as:

$Game_0$: This game simulates the real attack first launched by $A$. According to the definition, we get:

$$Adv_P^A = |2Pr[Win_0] - 1| \tag{1}$$

$Game_1$: This game simulates the eavesdropping attack. $A$ gets all the messages transmitted publicly. Then, $A$ guesses the random bit $r$. However, because of the ECDLP, the attacker cannot judge the association between the captured messages and the session keys. Therefore, we get:

$$Pr[Win_0] = Pr[Win_1] \tag{2}$$

$Game_2$: This game simulates the collision attack on the transcripts and hash results according to the definition of the birthday paradox, the probability of hash collision is less than $\frac{q_{HA}^2}{2^{l_{HA}+1}}$, and the collision probability of other transcripts is less than $\frac{(q_{SE}+q_{EX})^2}{2n}$. Therefore, we have:

$$Pr[Win_2] - Pr[Win_1] \leq \frac{q_{HA}^2}{2^{l_{HA}+1}} + \frac{(q_{SE}+q_{EX})^2}{2n} \tag{3}$$

$Game_3$: This game simulates $A$ executes $CorruptUser(In_{Ui}^i)$, $CorruptDevice(In_{SDj}^i)$, and $CorruptGateway(In_{GW}^i)$ to obtain the stored information $\{B_{U_i}, TID_{U_i}, TC_{U_i}, \tau_{U_i}, HPW_{U_i}, f(PID_{GW}, y), Rep, Gen, h, G\}$ in the user's device, $\{TID_{D_j}, DID_{D_j}, Ch_{D_j}, g(PID_{GW}, y), PUF, h, G\}$ in the smart device, and $\{SID_{GW}, TC_{GW}, Ch_{GW}, RT_{GW}, PUF, h, G\}$, $<TID_{D_j}, g(PID_{D_j}, y)>$, and $<TID_{U_i}, f(PID_{U_i}, y)>$ in the smart gateway. Where $r'_{U_i} = B_{U_i} \oplus h(PW_{U_i}^{in} \parallel \sigma'_{U_i})$, $PID_{U_i} = h(ID_{U_i} \parallel r_{U_i})$, $\sigma'_{U_i}$ is the biometric key, $PID_{D_j} = DID_{D_j} \oplus h(PUF(Ch_{D_j}))$, $PID_{GW} = SID_{GW} \oplus h(Ch_{GW} \parallel PUF(Ch_{GW}))$, $PID_{U_i}$, $PID_{D_j}$, and $PID_{GW}$ are used for verification and session key agreement. If $A$ wants to obtain the valuable parameters, $A$ must guess $\sigma'_{U_i}$ or break PUF. Suppose the probability of breaking PUF by $A$ is $Adv_{PUF}^A$. Therefore, we have:

$$Pr[Win_3] - Pr[Win_2] \leq q_{SE}(\frac{1}{2^{l_{bio}}} + Adv_{PUF}^A) \tag{4}$$

$Game_4$: $A$ can obtain $M_1 = r_{D_j} \cdot G$, $M_2 = R_{D_j} \cdot G$, $M_3 = n_{U_i} \cdot G$, and $M_4 = N_{U_i} \cdot G$ publicly, the session key agreements are based on ECDLP and the symmetric bivariate polynomial. This game simulates that $A$ calculates the session keys according to the transcripts. We have:

$$Pr[Win_4] - Pr[Win_3] \leq Adv_{ECDLP}^A \cdot Adv_{SBP}^A \tag{5}$$

The session keys are generated independently and randomly. Hence, the advantage of guessing $r$ is equal to guessing the session key. We have:

$$Pr[Win_4] = \frac{1}{2} \tag{6}$$

Combining the above formulas, we have:

$$\frac{1}{2} Adv_P^A = \left| Pr[Win_0] - \frac{1}{2} \right|$$

$$\leq \frac{q_{HA}^2}{2^{l_{HA}+1}} + \frac{(q_{SE}+q_{EX})^2}{2n} + \frac{q_{SE}}{2^{l_{bio}}} + q_{SE} Adv_{PUF}^A + Adv_{ECDLP}^A \cdot Adv_{SBP}^A$$

That is:

$$Adv_P^A \leq \frac{q_{HA}^2}{2^{l_{HA}}} + \frac{(q_{SE}+q_{EX})^2}{n} + \frac{q_{SE}}{2^{l_{bio}-1}} + 2q_{SE}Adv_{PUF}^A + 2Adv_{ECDLP}^A \cdot Adv_{SBP}^A$$

□

## 7. Informal Security Analysis

### *7.1. Anonymity and Untraceability*

In our proposed scheme, all messages are calculated from random numbers, and the temporary identities are changed in each session. Therefore, the proposed scheme is anonymous and untraceable.

### *7.2. Perfect Forward Secrecy*

In our proposed scheme, the session key negotiation is based on long-term shared secrets and ECDLP. Even if the long-term key $g(PID_{D_j}, PID_{GW})$, $f(PID_{U_i}, PID_{GW})$ and the current session key is leaked, the attacker cannot get the random number and cannot calculate the previous or later session key. Therefore, the protocol has perfect forward secrecy.

### *7.3. Impersonation Attack*

If an attacker wants to impersonate $SD_j$, stealing the secret $g(PID_{GW}, PID_{D_j})$ is a precondition. The attacker can't get the $PID_{D_j}$ because of the security features of PUF, and the secret will not be revealed even if the gateway is compromised, so the attacker can't impersonate $SD_j$.

If an attacker wants to impersonate $U_i$. Calculating $PID_{U_i} = h(ID_{U_i} \parallel r_{U_i})$, where $ID_{U_i}$ is public. But calculating $r_{U_i} = B_{U_i} \oplus h(PW_{U_i} \parallel \sigma_{U_i})$ is difficult because of the biological key $\sigma_{U_i}$, so it is impossible to impersonate $U_i$.

If an attacker wants to impersonate $GW$ negotiate with $SD_j$ or communicate with $U_i$, the attacker needs to know the $PID_{GW}$ to calculate the secret $g(PID_{D_j}, PID_{GW})$ or $f(PID_{U_i}, PID_{GW})$. However, the $PID_{GW}$ of the smart gateway also needs to be calculated by the PUF, so the attacker can't impersonate $GW$.

### *7.4. Replay Attack*

If an attacker captures previous data transmitted on the public channel and resends it, the data recipient will verify the freshness of timestamp. The integrity of the message is combined with the timestamps. The modified timestamp cannot pass the verification.

### *7.5. Mobile Device Loss/Stolen Attack*

If an attacker gets the user's mobile device, all the information is extracted in the $U_i$'s mobile device. Because the secret value $r_{U_i}$ is calculated from the $U_i$'s bioinformatic features $\sigma_{U_i}$, the attacker cannot calculate $PID_{U_i}$ and secret value $f(PID_{U_i}, PID_{GW})$. Therefore, the scheme can resist the mobile device loss/stolen attack.

### *7.6. Smart Device Captured Attack*

Suppose an attacker captures the smart device $SD_j$ and extracts the information stored in it. Due to the security features of PUF, the attacker cannot calculate $PUF(Ch_{D_j})$, $PID_{D_j}$ and $g(PID_{GW}, PID_{D_j})$. So, the attacker cannot impersonate $SD_j$, and it is impossible to affect other smart devices.

### *7.7. Smart Gateway Compromised Attack*

Suppose an attacker compromises the smart gateway; he can get the information $\{SID_{GW}, TC_{GW}, Ch_{GW}, RT_{GW}, PUF, h, G, \langle TID_{D_j}, g(PID_{D_j}, y)\rangle, \langle TID_{U_i}, f(PID_{U_i}, y)\rangle\}$ stored in the memory. Because of the features of PUF, the attacker cannot calculate $PID_{GW}$, $g(PID_{D_j}$,

$PID_{GW}$), $g(PID_{D_j}, PID_{GW})$. Therefore, the scheme can resist the smart gateway compromised attack.

### *7.8. Man-in-the-Middle Attack*

Since an attacker can neither impersonate $GW$, nor $U_i$ and $SD_j$, which is described in the Impersonation Attack. Therefore, the scheme can resist the man-in-the-middle attack.

### *7.9. Desynchronization Attack*

The reason our protocol can resist the desynchronization attack is that $TID_{D_j}$ is updated after passing the authentication and establishing the session key.

### *7.10. The Ephemeral Secret Leakage (ESL) Attack*

Suppose an attacker gets an ephemeral secret during a negotiation, such as $r_{D_j}$, $R_{D_j}$, $n_{U_i}$, $N_{U_i}$. However, the session key calculation requires the long-term key $g(PID_{D_j}, PID_{GW})$ or $f(PID_{U_i}, PID_{GW})$. Therefore, the scheme is resistant to the ephemeral secret leakage (ESL) attack.

## 8. Comparisons

In this section, we compared the security features and performance between our protocol and some related schemes [16,17,19,21]. The performance evaluation consists of communication and computation costs.

In terms of computational consumption, TI MSP430 microcontrollers are widely used in measurement and control equipment, so we take them to carry the TMP36 sensor to simulate the calculation consumption of the smart device. The mobile device chip uses ARM Cortex-A9 MPCore@890 Mhz CPU, and the smart gateway uses Intel Core i5-2500@3.3 GHz. To ensure the consistency of statistical methods, we adopt the conversion method of [19], $T_h \approx T_{mac} \approx T_{hmac}, T_{epm} \approx T_{me}, T_p \approx 16T_{mac}$, where $T_h$ is the time of the hash function, $T_e$ is the time of symmetric key encryption or decryption, $T_p$ is the time of symmetric polynomial, $T_{epm}$ is the time of ECC point multiplication, $T_{mac}$ is the time of message authentication code (MAC), $T_{hmac}$ is the time of hashed MAC and $T_{me}$ the time of is modular exponentiation; the notations are shown in Table 2. According to the smart device simulation result, $T_h \approx 1.42$ ms, $T_e \approx 2.18$ ms, $T_{epm} \approx 21.82$ ms. According to the mobile device simulation, $T_h \approx 0.067$ ms, $T_e \approx 0.085$ ms, $T_{epm} \approx 13.56$ ms. According to the smart gateway simulation, $T_h \approx 0.037$ ms, $T_e \approx 0.055$ ms, $T_{epm} \approx 8.77$ ms. The result is shown in Table 3.

**Table 2.** Notations.

| Symbol | Algorithm |
|---|---|
| $T_h$ | Hash function |
| $T_e$ | Symmetric key encryption or decryption |
| $T_p$ | Symmetric polynomial |
| $T_{epm}$ | ECC point multiplication |
| $T_{mac}$ | Message authentication code (MAC) |
| $T_{hmac}$ | Hashed MAC |
| $T_{me}$ | Modular exponentiation |

**Table 3.** Algorithm execution time (ms).

| Device | $T_h$ | $T_e$ | $T_{epm}$ |
|---|---|---|---|
| Mobile device | 0.067 | 0.085 | 13.56 |
| Smart device | 1.42 | 2.18 | 21.82 |
| Gateway | 0.037 | 0.055 | 8.77 |

In terms of communication consumption, the length of the output, such as random number, identity, timestamp, hash, MAC, symmetric encryption block, and ECC point are 128 bits, 128 bits, 32 bits, 256 bits, 256 bits, 128 bits, and 1024 bits, respectively.

Table 4 is the comparison of the security features, and the comparison of the computation and communication consumptions are shown in Tables 5 and 6.

**Table 4.** Security features.

| Features | [16] | [17] | [21] | [19] | Ours |
|---|---|---|---|---|---|
| Offline password-guessing attack | ✗ | ✗ | ✓ | ✓ | ✓ |
| Mobile device stolen attack | ⊘ | ⊘ | ✓ | ✗ | ✓ |
| Smart device captured attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gateway compromised attack | ✗ | ✗ | ⊘ | ✗ | ✓ |
| Replay attack | ✗ | ✗ | ✓ | ✓ | ✓ |
| User impersonation attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Smart device impersonation attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Gateway impersonation attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| User anonymity | ✓ | ✓ | ✓ | ✓ | ✓ |
| Un-traceability | ✓ | ✓ | ✓ | ✗ | ✓ |
| Man-in-the-middle attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mutual authentication | ✓ | ✓ | ✓ | ✓ | ✓ |
| Desynchronization attack | ⊘ | ⊘ | ✗ | ✗ | ✓ |
| Key agreement | ✓ | ✓ | ✓ | ✓ | ✓ |
| Perfect forward secrecy | ✗ | ✗ | ✗ | ✗ | ✓ |

✓: secure; ✗: insecure; ⊘: not applicable.

**Table 5.** Computation costs (ms).

| Scheme | Mobile Device | Smart Device | Gateway | Total |
|---|---|---|---|---|
| [16] | $6T_h + 2T_{epm} \approx 27.52$ | $3T_h \approx 4.26$ | $7T_h + T_{epm} \approx 9.03$ | 40.81 |
| [17] | $6T_h + 2T_{epm} \approx 27.52$ | $3T_h \approx 4.26$ | $7T_h + T_{epm} \approx 9.03$ | 40.81 |
| [19] | $27T_h \approx 1.81$ | - | $24T_h \approx 0.89$ | 2.7 |
| [21] | $8T_h + T_e \approx 0.62$ | $6T_h + T_e \approx 10.7$ | $10T_h + 2T_e \approx 0.48$ | 11.8 |
| Ours | $21T_h + 2T_{epm} \approx 28.53$ | - | $19T_h + 2T_{epm} \approx 18.24$ | 46.77 |

**Table 6.** Communication costs (bits).

| Scheme | Total Messages | Communication Cost |
|---|---|---|
| [16] | 4 | 4480 |
| [17] | 4 | 4608 |
| [21] | 4 | 4000 |
| [19] | 2 | 2112 |
| Ours | 2 | 2752 |

In the scheme [16], the user inputs the identity and the password to achieve verification. When facing the offline password guessing attack, the password will no longer be secure. In addition, the gateway stores the long-term keys, the keys of users and devices will be leaked when the gateway is compromised, which threatens the security of communication data and session keys. The protocol cannot resist the gateway compromised attack and has no perfect forward secrecy. In addition, there is no timestamp verification in the authentication process; messages can be replayed.

In terms of communication consumption, the scheme [16] performs a total of 4 transmissions, the message sent by the user is 2560 bits (1024 + 1024 + 256 + 256), the gateway sent to the device is 768 bits (512 + 256), the returned message from the device is 512 bits (256 + 256), and the gateway sends to the user is 640 bits (384 + 256), so a total consumption is 4480 bits (2560 + 768 + 512 + 640).

In scheme [17], protocol is improved based on [16]. However, it can not resist the offline password-guessing attack and the replay attack. Similarly, because the gateway stores keys and sensitive data, the protocol does not provide perfect forward secrecy and cannot resist the gateway compromised attack. On the communication consumption, this protocol makes 4 transmissions. The user sends to the gateway 2592 bits (1024 + 1024 + 256 + 256 + 32), the gateway sends to the device 800 bits (512 + 256 + 32), the device returns 544 bits (256 + 256 + 32) message, then the gateway returns 672 bits (384 + 256 + 32) message to the user, the total consumption is 4608 bits (2592 + 800 + 544 + 672).

The scheme [21] stores the keys in the gateway, and the authors claimed that the gateway is fully trusted and cannot be compromised. So, the gateway compromised attack does not apply to this scheme. The biometric information and password are used for user verification, which can resist offline password guessing attacks and mobile device stolen attacks. A temporary identifier is used in the scheme to provide anonymity, which brings the issue of updating the temporary identifier. If the attacker intercepts the last message, the temporary identifier will be out of sync. In addition, the session key is built based on the shared secret key, so the scheme has no perfect forward secrecy. The data transmission communication consumptions in the protocol are 672 bits (128 + 256 + 256 + 32), 1056 bits (768 + 256 + 32), 800 bits (256 + 256 + 256 + 32), 1472 bits (896 + 256 + 256 + 32 + 32) respectively, so the total communication consumption is 4000 bits (672 + 1056 + 800 + 1472). The scheme [21] is based on symmetric encryption, and the shared key is preset in the gateway, so the calculation is fast.

We have pointed out that Guo et al.'s scheme [19] is not immune to the mobile device stolen attack, the gateway compromised attack, and the desynchronization attack. There is traceability and no perfect forward secrecy. In terms of communication consumption, access to the smart home requires 2 message transfers, consuming 1056 bits (256 + 128 + 256 + 256 + 128 + 32), 1056 bits (256 + 256 + 256 + 256 + 32), respectively, so total consumption is 2112 bits (1056 + 1056).

Our proposed scheme can support all security features. In terms of communication, our authentication requires 2 message transfers, sending 1440 bits (128 + 1024 + 256 + 32) and 1312 bits (1024 + 256 + 32), respectively, so total consumption is 2752 bits (1440 + 1312).

Tables 2 and 5 show that our protocol has better security and transmission efficiency. To achieve perfect forward secrecy (PFS), it needs at least four times elliptic curve point multiplication. Since only our scheme achieves PFS, the computation cost of our scheme is a little more than others but at the same computation cost level.

## 9. Conclusions

In this paper, we first pointed out that many existing authentication protocols for a smart home have one or more security flaws. It further showed that almost all these protocols may suffer from gateway compromised attacks. Then we described that Guo et al.'s protocol in a fog-enabled smart home is vulnerable to smart gateway compromised attacks, desynchronization attacks, and mobile device lost/stolen attacks and has no perfect forward secrecy and untraceability. To overcome the shortcomings of Guo et al.'s protocol, we adopt PUF to resist gateway compromised attacks, adopt ECDH to achieve perfect forward secrecy, and propose a secure and privacy-preserving authentication protocol that avoids gateway compromised attacks in fog-enabled smart homes, and formally prove the security of the proposed protocol under random oracle model. Finally, we compare our protocol with some related protocols. The proposed protocol has better security and transmission efficiency with the same computation cost level.

**Author Contributions:** Conceptualization, Q.X. and J.H.; methodology, Q.X.; validation, Q.X., J.H., and Z.D.; formal analysis, Z.D.; investigation, J.H.; writing—original draft preparation, J.H.; writing—review and editing, Q.X.; supervision, Q.X.; project administration, Q.X.; funding acquisition, Q.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Natural Science Foundation of China (Grant No. U21A20466).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lutolf, R. Smart home concept and the integration of energy meters into a home based system. In Proceedings of the Seventh International Conference on Metering Apparatus and Tariffs for Electricity Supply, Glasgow, UK, 17–19 November 1992; pp. 277–278.
2. Berlo, A.V.; Allen, B. *Design Guidelines on Smart Homes: A COST 219bis Guidebook*; COST, European Co-Operation in the Field of Scientific and Technical Research: Cham, Switzerland, 1999.
3. Zemrane, H.; Baddi, Y.; Hasbi, A. Internet of things smart home ecosystem. In *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*; Springer: Cham, Switzerland, 2020; pp. 101–125.
4. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of things for smart cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. [CrossRef]
5. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog computing and its role in the internet of things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–16.
6. Rahimi, M.; Songhorabadi, M.; Kashani, M.H. Fog-based smart homes: A systematic review. *J. Netw. Comput. Appl.* **2020**, *153*, 102531. [CrossRef]
7. Alatoun, K.; Matrouk, K.; Mohammed, M.A.; Nedoma, J.; Martinek, R.; Zmij, P. A Novel Low-Latency and Energy-Efficient Task Scheduling Framework for Internet of Medical Things in an Edge Fog Cloud System. *Sensors* **2022**, *22*, 5327. [CrossRef] [PubMed]
8. Rocha Filho, G.P.; Brandão, A.H.; Nobre, R.A.; Meneguette, R.I.; Freitas, H.; Gonçalves, V.P. HOsT: Towards a Low-Cost Fog Solution via Smart Objects to Deal with the Heterogeneity of Data in a Residential Environment. *Sensors* **2022**, *22*, 6257. [CrossRef]
9. Chen, Y.-Y.; Chen, M.-H.; Chang, C.-M.; Chang, F.-S.; Lin, Y.-H. A Smart Home Energy Management System Using Two-Stage Non-Intrusive Appliance Load Monitoring over Fog-Cloud Analytics Based on Tridium's Niagara Framework for Residential Demand-Side Management. *Sensors* **2021**, *21*, 2883. [CrossRef]
10. Debauche, O.; Nkamla Penka, J.B.; Mahmoudi, S.; Lessage, X.; Hani, M.; Manneback, P.; Lufuluabu, U.K.; Bert, N.; Messaoudi, D.; Guttadauria, A. RAMi: A New Real-Time Internet of Medical Things Architecture for Elderly Patient Monitoring. *Information* **2022**, *13*, 423. [CrossRef]
11. Verma, P.; Sood, S.K. Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet Things J.* **2018**, *5*, 1789–1796. [CrossRef]
12. Hu, B.; Tang, W.; Xie, Q. A Two-factor Security Authentication Scheme for Wireless Sensor Networks in IoT Environments. *Neurocomputing* **2022**, *500*, 741–749. [CrossRef]
13. Jeong, J.; Chung, M.Y.; Choo, H. Integrated OTP-based user authentication scheme using smart cards in home networks. In Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), Waikoloa, HI, USA, 7–10 January 2008; p. 294.
14. Xue, K.; Ma, C.; Hong, P.; Ding, R. A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 316–323. [CrossRef]
15. Saqib, M.; Jasra, B.; Moon, A.H. A lightweight three factor authentication framework for IoT based critical applications. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6925–6937. [CrossRef]
16. Shuai, M.; Yu, N.; Wang, H.; Xiong, L. Anonymous authentication scheme for smart home environment with provable security. *Comput. Secur.* **2019**, *86*, 132–146. [CrossRef]
17. Kaur, D.; Kumar, D. Cryptanalysis and improvement of a two-factor user authentication scheme for smart home. *J. Inf. Secur. Appl.* **2021**, *58*, 102787. [CrossRef]
18. Santoso, F.K.; Vun, N.C. Securing IoT for smart home system. In Proceedings of the 2015 International Symposium on Consumer Electronics (ISCE), Madrid, Spain, 24–26 June 2015; pp. 1–2.
19. Guo, Y.; Zhang, Z.; Guo, Y. SecFHome: Secure remote authentication in fog-enabled smart home environment. *Comput. Netw.* **2022**, *207*, 108818. [CrossRef]
20. Blundo, C.; Santis, A.D.; Herzberg, A.; Kutten, S.; Vaccaro, U.; Yung, M. *Perfectly-Secure Key Distribution for Dynamic Conferences. Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 1992; pp. 471–486.
21. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans. Dependable Secur. Comput.* **2017**, *17*, 391–406. [CrossRef]
22. Haseeb-ur-rehman, R.M.A.; Liaqat, M.; Aman, A.H.M.; Almazroi, A.A.; Hasan, M.K.; Ali, Z.; Ali, R.L. LR-AKAP: A Lightweight and Robust Security Protocol for Smart Home Environments. *Sensors* **2022**, *22*, 6902. [CrossRef]

23. Lee, J.; Oh, J.; Kwon, D.; Kim, M.; Yu, S.; Jho, N.-S.; Park, Y. PUFTAP-IoT: PUF-Based Three-Factor Authentication Protocol in IoT Environment Focused on Sensing Devices. *Sensors* **2022**, *22*, 7075. [CrossRef]
24. Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779. [CrossRef]
25. Yang, J.H.; Chang, C.C. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Comput. Secur.* **2009**, *28*, 138–143. [CrossRef]
26. Islam, S.H.; Biswas, G.P. A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *J. Syst. Softw.* **2011**, *84*, 1892–1898. [CrossRef]
27. Zerrouki, F.; Ouchani, S.; Bouarfa, H. A survey on silicon PUFs. *J. Syst. Archit.* **2022**, *127*, 102514. [CrossRef]
28. Yi, F.; Zhang, L.; Xu, L.; Yang, S.; Lu, Y.; Zhao, D. WSNEAP: An Efficient Authentication Protocol for IIoT-Oriented Wireless Sensor Networks. *Sensors* **2022**, *22*, 7413. [CrossRef] [PubMed]
29. Yu, S.; Park, K. PUF-PSS: A Physically Secure Privacy-Preserving Scheme Using PUF for IoMT-Enabled TMIS. *Electronics* **2022**, *11*, 3081. [CrossRef]
30. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
31. Canetti, R.; Krawczyk, H. Analysis of key-exchange protocols and their use for building secure channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques; Springer: Berlin/Heidelberg, Germany, 2001; pp. 453–474.
32. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.