



Article

Toward Smart Home Authentication Using PUF and Edge-Computing Paradigm

Tsu-Yang Wu ¹, Fangfang Kong ¹, Liyang Wang ¹, Yeh-Cheng Chen ², Saru Kumari ³ and Jeng-Shyang Pan ^{1,4,*}

¹ College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China

² Department of Computer Science, University of California, Davis, CA 001313, USA

³ Department of Mathematics, Chaudhary Charan Singh University, Meerut 250004, India

⁴ Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan

* Correspondence: jengshyangpan@gmail.com

Abstract: The smart home is a crucial embodiment of the internet of things (IoT), which can facilitate users to access smart home services anytime and anywhere. Due to the limited resources of cloud computing, it cannot meet users' real-time needs. Therefore, edge computing emerges as the times require, providing users with better real-time access and storage. The application of edge computing in the smart home environment can enable users to enjoy smart home services. However, users and smart devices communicate through public channels, and malicious attackers may intercept information transmitted through public channels, resulting in user privacy disclosure. Therefore, it is a critical issue to protect the secure communication between users and smart devices in the smart home environment. Furthermore, authentication protocols in smart home environments also have some security challenges. In this paper, we propose an anonymous authentication protocol that applies edge computing to the smart home environment to protect communication security between entities. To protect the security of smart devices, we embed physical unclonable functions (PUF) into each smart device. Real-or-random model, informal security analysis, and ProVerif are adopted to verify the security of our protocol. Finally, we compare our protocol with existing protocols regarding security and performance. The comparison results demonstrate that our protocol has higher security and slightly better performance.

Keywords: IoT; edge computing; smart home; PUF



Citation: Wu, T.-Y.; Kong, F.; Wang, L.; Chen, Y.-C.; Kumari, S.; Pan, J.-S. Toward Smart Home Authentication Using PUF and Edge-Computing Paradigm. *Sensors* **2022**, *22*, 9174. <https://doi.org/10.3390/s22239174>

Academic Editors: Muhammad Naveed Aman, Chien-Ming Chen and Shehzad Ashraf Chaudhry

Received: 14 October 2022

Accepted: 22 November 2022

Published: 25 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The internet of things (IoT) [1–4] is a network connected with everything, which can collect various types of information in real time and communicate with other devices. The development of the IoT has brought significant achievements in different fields, such as smart city [5–8], healthcare [9–11], vehicular ad hoc network (VANET) [12–16], smart home [17–19], and artificial intelligence [20,21]. The smart home is the embodiment of IoT. It is an environment in which smart devices are deployed in the house, and various devices provide services to users through connecting to the internet. People can access smart home services anytime and anywhere through voice assistants or applications and easily control smart devices. In the smart home environment, people's lives have become more comfortable, their lifestyle has become more intelligent, and people's quality of life is also constantly improving.

Many smart devices are deployed in the smart home environment, such as smart air conditioners, smart desk lamps, and smart curtains. These smart devices can provide users with various services. The traditional framework of the smart home is shown in Figure 1. The framework consists of four entities: registration authority (RA), users, gateway, and smart devices. The primary responsibilities of RA include the registration of users and smart devices as well as the distribution of system parameters. Gateway is a bridge

between smart devices and users. Only smart devices registered in RA can provide services for users. Users use mobile devices (such as smartphones, tablets, and smartwatches) to control smart devices in their homes at any time. For example, users can turn on the air conditioner and close the curtains outdoors; users can master the family situation by viewing the smart camera.

The traditional smart home architecture relies on centralized cloud computing, which is used for data collection and processing. There are some problems in the traditional architecture; for example, in monitoring this application scenario that requires real-time feedback, cloud computing [22–24] will process a great deal of data, which may not meet users' real-time needs [25,26]. Edge computing [27–29] is closer to the data source than cloud computing. It can better process data and provide real-time access, solving the above problems. An edge gateway is the node of edge computing, which can give real-time computing and storage in the smart home environment instead of going to the remote cloud center. The edge gateway can locally process the data collected between the user's mobile device and the smart device. First, the user and the smart device are registered in the registration center, and the registered legal user negotiates the session key with the smart device with the help of the edge gateway. Only legal users can enjoy smart home services. Although smart homes bring convenience to people's lives, users and smart devices communicate through public channels. Due to the openness of the public channel, the information transmitted in the public channel may be intercepted by malicious attackers, which will lead to user privacy disclosure. Therefore, protecting users and smart devices for secure communication is very important.

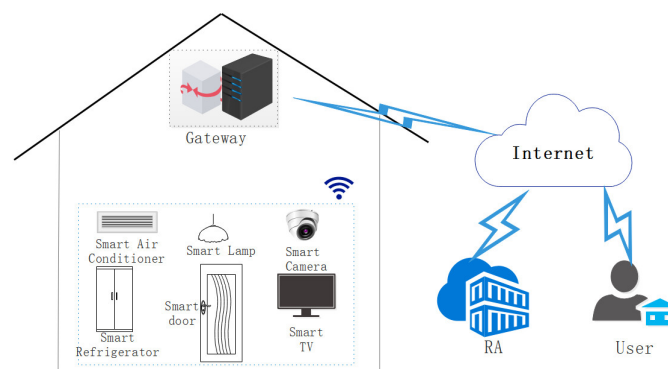


Figure 1. Traditional architecture of smart home.

The physical unclonable function (PUF) [30,31] is a function that can be embedded in an integrated circuit. The integrated circuit takes a bit string as input (or called challenge) and generates a random response string as the output. For various PUF modules manufactured on the same integrated circuit, no two PUF modules will produce the same response if faced with the same challenges. If a malicious attacker wants to change or destroy the PUF, it will change the corresponding internal circuit and logic gate delay. At this time, even if the same challenge is entered, the malicious attacker cannot obtain the same response. According to the microstructure and response of a given PUF, it is difficult for a malicious attacker to guess or infer the correct challenge. Moreover, the PUF is available on demand and does not require secure storage.

In this paper, a smart home authentication protocol using PUF and edge computing paradigm is proposed. The following are the novelty and contributions of this paper:

- (1) To the best of our knowledge, we are the first to introduce an edge-computing-based smart home architecture and propose an authentication protocol based on this architecture. In our protocol, the user and the smart device realize mutual authentication with the help of the edge gateway and successfully establish a session key for secure communication.

- (2) We apply PUF to smart devices to prevent data-leakage attacks launched by attackers, thus ensuring data security. According to the security properties of PUF, even if an attacker gets the same challenge, they cannot get the same response. Therefore, using PUF in our protocol can resist tampering and biological cloning attacks.
- (3) We verify the security of our protocol by using the real or random (ROR) model, informal security analysis and simulation software (ProVerif). The results are shown that the proposed protocol can resist several well-known attacks.
- (4) Finally, we compare our protocol with existing protocols regarding security and performance. The comparisons demonstrate that our protocol guarantees better security and slightly lower communication cost.

The remainder of this paper is structured as follows. The relevant research on smart homes, edge computing, and PUF is briefly reviewed in Section 2. In Section 3, we describe the system model and detailed protocol. We prove the security of our proposed protocol in Section 4. In Section 5, we compare our protocol with existing protocols in terms of security and performance. In Section 6, we set forth our conclusions.

2. Related Work

Many researchers proposed several authentication and key agreement (AKA) protocols in different environments. In 2008, Jeong et al. [32] proposed a lightweight user authentication protocol in the home network environment. This protocol could not guarantee the anonymity of users, and users were easily tracked. In addition, the protocol could not resist attacks by privileged insiders. Vaidya et al. [33] proposed a strong cryptographic-based AKA protocol in the home network environment. The author showed that this protocol has strong security. However, Kim et al. [34] performed cryptanalysis on the protocol of Vaidya et al. [33] and found that their protocol could not provide forward security and suffered from stolen smart card attacks. Kim et al. [34] indicated the security vulnerabilities of Vaidya et al.'s protocol [33] and proposed an enhanced AKA. Unfortunately, the protocol of Kim et al. [34] could not resist privileged insider attacks and was unable to guarantee users' anonymity and untraceability. In 2017, Wazid et al. [35] proposed a lightweight AKA for remote users. They proved that their protocol was secure and had good performance. However, Lyu et al. [36] discovered that the protocol of Wazid et al. [35] was unable to withstand stolen verifier attacks and synchronization attacks. Lyu et al. [36] introduced IFTTT as a home gateway and used it as the executor and supervisor of commands. In 2019, Shuai et al. [37] proposed an efficient AKA protocol using elliptic curve cryptography (ECC) and showed that the protocol could resist existing attacks. However, Kaur and Kumar [38] found that Shuai et al.'s protocol [37] was vulnerable to insider attacks, replay attacks, and offline password guessing attacks. Subsequently, Kaur and Kumar [38] proposed a two-factor AKA protocol to enhance security. Unfortunately, Yu et al. [18] found that the protocol of Kaur and Kumar [38] could not resist impersonation attacks and session key disclosure attacks and proposed a secure AKA protocol based on three factors. However, Alzahrani et al. [39] discovered that Yu et al.'s protocol [18] was unable to achieve mutual authentication. Banerjee et al. [40] found that Shuai et al.'s protocol [37] could not resist stolen smart card attacks and user impersonation attacks and then proposed an efficient anonymous authentication protocol. Unfortunately, this protocol cannot guarantee the anonymity and untraceability of users [41]. Oh et al. [42] proposed an efficient authentication protocol using the hush function for IoT-based smart home environments. They proved that the protocol can resist known attacks.

In edge-computing environments, Tsai and Lo [43] proposed an authentication protocol using identity-based encryption technology. This protocol is based on bilinear pairing and the identity-based cryptosystem, which reduced the computation of users and servers. However, Jiang et al. [44] proved that the protocol of Tsai and Lo [43] was vulnerable to server impersonation attacks. Irshad et al. [45] also found that Tsai and Lo's protocol [43] could not resist the de-synchronization attacks. They designed an improved multi-server authentication protocol and proved that the designed protocol could resist known attacks.

However, Xiong et al. [46] pointed out that the protocol of Irshad et al. [45] lacked the registration and revocation of users and designed a new protocol. Later, Jia et al. [47] designed an identity-based authentication protocol. However, Li et al. [26] found that Jia et al.'s protocol [47] could not resist man-in-the-middle (MITM) attacks and then proposed a novel mobile edge computing environment architecture and designed a lightweight AKA protocol on this architecture. Unluckily, Li et al.'s protocol [26] cannot resist replay attacks and denial of service attacks. Kaur et al. [48] proposed a lightweight privacy-preserving AKA protocol, which adopts elliptic curve cryptography to resist various attacks, thus ensuring secure communication between entities.

Numerous PUF-based AKA protocols were recently proposed to address the aforementioned well-known security issues. Aysu et al. [49] proposed a secure and efficient end-to-end AKA protocol based on PUF between servers and resource-limited devices. Chatterjee et al. [50] designed a PUF-based AKA protocol for the IoT to realize authentication and secure information transfer between devices. Braeken [51] analyzed Chatterjee's protocol [50] and found that it could not resist MITM attacks and replay attacks and proposed an efficient AKA protocol. Gope et al. [52] proposed a lightweight AKA protocol for user privacy protection in industrial wireless sensor networks. In this protocol, user and sensor nodes can authenticate and negotiate the session key with the aid of the gateway. Chen et al. [53] found that PUF-authentication protocols are vulnerable to machine learning attacks. Therefore, they adopted the concept of Shamir's secret sharing to design an AKA protocol to resist the attacks. Ebrahimabadi et al. [54] designed a novel authentication protocol based on PUF and showed that the protocol has better security and efficiency. In order to ensure that users can obtain secure and timely services in a smart city environment, Yu et al. [55] proposed a lightweight authentication protocol based on PUF in an internet of drones environment. Shao et al. [56] proposed an AKA protocol using PUF in a wireless medical sensor environment with limited resources to ensure data security and patient privacy. Some significant relevant works are listed in Table 1.

Table 1. The summary of authentication protocols.

Protocols	Cryptographic Techniques and Properties	Limitations
Yu et al. [18]	(1) Utilized one-way hash function (2) Utilized symmetric encryption	(1) Cannot provide mutual authentication
Jeong et al. [32]	(1) Utilized one-way hash function (2) Based on one-time password	(1) Cannot resist insider attacks (2) Cannot guarantee user anonymity
Vaidya et al. [33]	(1) Utilized one-way hash function (2) Utilized symmetric encryption (3) Utilized HMAC-based one-time password algorithm (4) Based on smart card	(1) Cannot resist provide perfect forward security (2) Cannot resist stolen smart card attacks
Wazid et al. [35]	(1) Utilized one-way hash function (2) Utilized symmetric encryption	(1) Cannot resist stolen verifier attacks (2) Cannot resist synchronization attacks
Shuai et al. [37]	(1) Utilized ECC (2) Utilized one-way hash function (3) Anonymity	(1) Cannot resist insider attacks (2) Cannot resist replay attacks (3) Cannot resist offline password guessing attacks
Kaur and Kumar [38]	(1) Utilized one-way hash function (2) Based on smart card (3) Utilized ECC (4) Two-factor	(1) Cannot resist impersonation attacks (2) Cannot resist ssession key disclosure attacks
Jia et al. [47]	(1) Utilized one-way hash function (2) Utilized ECC (3) Utilize bilinear pairing	(1) Cannot resist MITM attacks
Chen et al. [53]	(1) Utilized one-way hash function (2) Based on PUF (3) Utilized Shamir's secret sharing	–
Banerjee et al. [40]	(1) Utilized one-way hash function	(1) Cannot guarantee user anonymity and untraceability
Oh et al. [42]	(1) Utilized one-way hash function (2) Based on smart card	–

3. Proposed Protocol

In this section, an authentication protocol using PUF and the edge-computing paradigm for the smart home environment is proposed. Four entities, trusted third party TTP , edge gateway EGW , user U_i , and smart device SD_j , are involved in our protocol. The system model is shown in Figure 2. Details on each entity are described below:

- (1) Trusted third party *TTP*: *TTP* is a trusted entity, mainly responsible for the registration of home users and smart devices. Additionally, it stores a few users and smart device registration parameters in the edge gateway’s secure database.
- (2) Edge gateway *EGW*: *EGW* is a trusted entity and is deployed in the home. *EGW* can collect data from various smart devices, process the data, and send the processed data to users who need data. It also serves as a bridge between smart devices and users.
- (3) Home user U_i : U_i refers to the legal users who have successfully registered through *TTP*. With the help of the *EGW*, legal home users can enjoy the services provided by smart devices and remotely control them through mobile devices (such as smartphones, tablets, and smartwatches) anytime and anywhere.
- (4) Smart device SD_j : SD_j deployed in the smart home environment (such as cameras, smart refrigerators, smart desk lamps, and smart locks) must be registered with *TTP*. Each smart device is embedded with a PUF module. In the smart home, it can execute the instructions transmitted by the user through the edge gateway and collect the data.

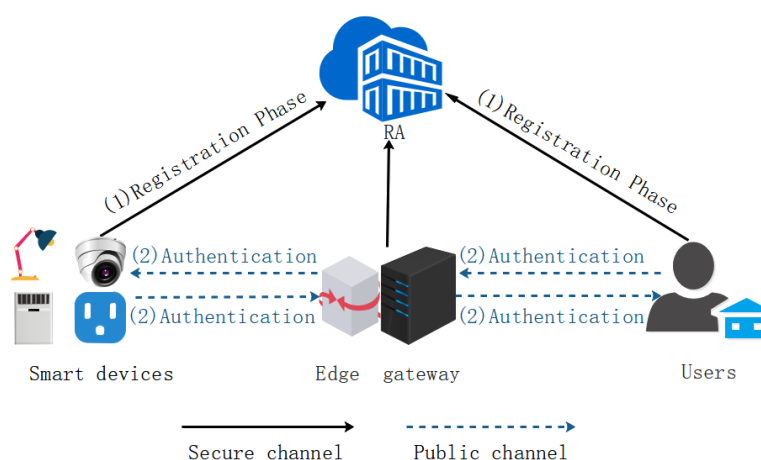


Figure 2. System model.

Our protocol is divided into the registration, login and authentication phases. Before U_i and SD_j are deployed in the smart home environment, *TTP* generates a master key x . Each SD_j has a unique identity $SMID_j$ and PUF module. The symbols used in the protocol are shown in Table 2, and the following thoroughly explains each phase.

Table 2. Notations.

Notations	Description
U_i	i th user
SD_j	j th smart device
UID_i	Identity of U_i
PID_i	Pseudo-identities of U_i
UPW_i	Password of U_i
<i>TTP</i>	Trusted third party
<i>EGW</i>	Edge gateway
$SMID_j$	Identity of SD_j
x	Private key of <i>TTP</i>
$Gen(\cdot)$	Fuzzy extractor probabilistic generation
$Rep(\cdot)$	Reproduction function
$PUF(\cdot)$	PUF function
SK	Session key
$h(\cdot)$	Secure-hash function
\parallel	Concatenation operation
\oplus	XOR operation

3.1. Registration Phase

In the smart home environment, U_i and SD_j must register with TTP via a secure channel. There are two phases of registration: U_i registration and SD_j registration.

User Registration Phase. If U_i wants to enjoy smart home services, he or she must first register as a legal user in TTP . The process of U_i registration is shown in Figure 3. The steps of U_i registration are described in detail below.

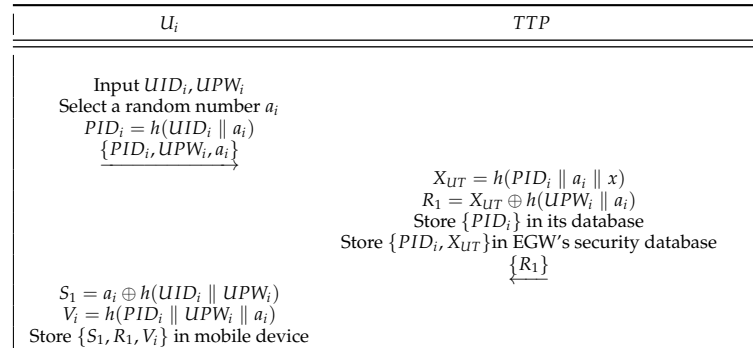


Figure 3. U_i registration phase.

- (1) To begin with, U_i uses the mobile device to enter the identity UID_i , password UPW_i , and selects a random number a_i . Then, the mobile device calculates

$$PID_i = h(UID_i || a_i). \quad (1)$$

Finally, U_i passes $\{PID_i, UPW_i, a_i\}$ to TTP through the secure channel.

- (2) When TTP receives $\{PID_i, UPW_i, a_i\}$, first retrieves PID_i from the database of TTP . If retrieved in the database, TTP rejects the U_i 's registration. Or else, TTP calculates

$$\begin{aligned} X_{UT} &= h(PID_i || a_i || x), \\ R_1 &= X_{UT} \oplus h(UPW_i || a_i). \end{aligned} \quad (2)$$

Thereafter, TTP calculation is completed, stores $\{PID_i\}$ in its database, and stores $\{PID_i, X_{UT}\}$ in the secure database of EGW . Finally, TTP sends $\{R_1\}$ to U_i .

- (3) When U_i receives $\{R_1\}$, it calculates

$$\begin{aligned} S_1 &= a_i \oplus h(UID_i || UPW_i), \\ V_i &= h(PID_i || RPW_i || a_i). \end{aligned} \quad (3)$$

Finally, U_i stores $\{S_1, R_1, V_i\}$ in the mobile device.

Smart Device Registration Phase. SD_j must be registered at TTP before it can provide smart home services to U_i . The SD_j registration process is shown in Figure 4. The following are the specific SD_j registration process.

- (1) Initially, SD_j selects an identity $SMID_j$, generates a challenge C_j . Then, SD_j calculates

$$\begin{aligned} R_j &= PUF(C_j), \\ Gen(R_j) &= (\sigma_j, \delta_j). \end{aligned} \quad (4)$$

Finally, SD_j sends $\{SMID_j, C_j, \delta_j\}$ to TTP .

- (2) After receiving $\{SMID_j, C_j, \delta_j\}$, TTP retrieves $SMID_j$ from the database. If not retrieved in the database, TTP calculates

$$X_{ST} = h(SMID_j || x || \delta_j). \quad (5)$$

Then, TTP stores $\{SMID_j\}$ in its database and stores $\{SMID_j, C_j, \delta_j, X_{ST}\}$ in EGW 's security database. Finally, TTP sends $\{X_{ST}\}$ to SD_j .

- (3) After receiving $\{X_{ST}\}$, SD_j generates a random number b_j , and then calculates

$$\begin{aligned} S_2 &= X_{ST} \oplus h(SMID_j \parallel b_j), \\ S_3 &= \delta_j \oplus h(SMID_j \parallel b_j \parallel X_{ST}). \end{aligned} \quad (6)$$

Finally, SD_j stores $\{S_2, S_3, b_j\}$ in memory.

SD_j	TTP
Choose an identity $SMID_j$ Generate a challenge set C_j $R_j = PUF(C_j)$ $Gen(R_j) = (\sigma_j, \delta_j)$ $\{SMID_j, C_j, \delta_j\}$	$X_{ST} = h(SMID_j \parallel x \parallel \delta_j)$ Store $\{SMID_j\}$ in its database Store $\{SMID_j, C_j, \delta_j, X_{ST}\}$ in EGW 's database $\{X_{ST}\}$
Generate a random number b_j $S_2 = X_{ST} \oplus h(SMID_j \parallel b_j)$ $S_3 = \delta_j \oplus h(SMID_j \parallel b_j \parallel X_{ST})$ Store $\{S_2, S_3, b_j\}$ in memory	

Figure 4. SD_j registration phase.

3.2. Login and Authentication Phase

In this phase, all entities communicate via a public channel. With the help of the EGW , the legal U_i establishes a session key SK with the SD_j . The established SK facilitates the U_i to safely obtain the service of the SD_j and future communication. The detailed login and authentication process is shown in Figure 5. The steps of this process are described in detail below.

- (1) First, U_i uses the mobile device to input his own identity UID_i , password UPW_i , and then calculates

$$\begin{aligned} a_i &= h(UID_i \parallel UPW_i) \oplus S_1, \\ PID_i &= h(UID_i \parallel a_i), \\ V_i^* &= h(PID_i \parallel RPW_i \parallel a_i). \end{aligned} \quad (7)$$

Next, U_i check $V_i^* \stackrel{?}{=} V_i$. If it holds, U_i successfully logs in. Otherwise, U_i will be denied login. Then, U_i calculates

$$\begin{aligned} X_{UT} &= R_1 \oplus h(a_i \parallel UPW_i), \\ RID_i &= h(PID_i \parallel a_i \parallel X_{UT}). \end{aligned} \quad (8)$$

Additionally, U_i selects unique identity $SMID_j$ of the SD_j , random number r_i , and T_1 . Then, U_i calculates

$$\begin{aligned} W_1 &= (SMID_j \parallel r_i) \oplus h(X_{UT} \parallel T_1), \\ W_2 &= RID_i \oplus X_{UT} \oplus T_1, \\ V_{UE} &= h(RID_i \parallel X_{UT} \parallel r_i \parallel T_1). \end{aligned} \quad (9)$$

At last, U_i sends the message $M_1 = \{W_1, W_2, PID_i, V_{UE}, T_1\}$ through the public channel to EGW .

- (2) When receiving M_1 sent by U_i , EGW first checks $|T_1 - T_s| \leq \Delta T$. If T_1 is valid, EGW uses PID_i retrieve X_{UT} from the secure database, and calculates

$$\begin{aligned}(SMID_j \parallel r_i) &= W_1 \oplus h(X_{UT} \parallel T_1), \\ RID_i &= W_2 \oplus X_{UT} \oplus T_1, \\ V_{UE}^* &= h(RID_i \parallel X_{UT} \parallel r_i \parallel T_1).\end{aligned}\quad (10)$$

Next, EGW checks $V_{UE}^* \stackrel{?}{=} V_{UE}$. If it is correct, EGW authenticates SD_j , and uses $SMID_j$ retrieves $\{C_j, \delta_j, X_{ST}\}$ from the secure database. Then, EGW generates a timestamp T_2 , and calculates

$$\begin{aligned}W_3 &= (C_j \parallel RID_i \parallel r_i) \oplus SMID_j \oplus \delta_j, \\ V_{ED} &= h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2).\end{aligned}\quad (11)$$

Eventually, EGW sends the message $M_2 = \{W_3, V_{ED}, T_2\}$ to SD_j .

- (3) When SD_j receives M_2 sent from EGW , first checks $|T_2 - T_s| \leq \Delta T$. Then, SD_j calculates

$$\begin{aligned}X_{ST} &= S_2 \oplus h(SMID_j \parallel b_j), \\ \delta_j &= S_3 \oplus h(SMID_j \parallel b_j \parallel X_{ST}), \\ (C_j \parallel RID_i \parallel r_i) &= W_3 \oplus SMID_j \oplus \delta_j, \\ V_{ED}^* &= h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2).\end{aligned}\quad (12)$$

Next, SD_j checks $V_{ED}^* \stackrel{?}{=} V_{ED}$. If it is correct, the identity of the EGW is authenticated, then SD_j calculates

$$\begin{aligned}\sigma_j &= Rep(PUF(C_j), \delta_j), \\ PSMID_j &= h(SMID_j \parallel \sigma_j \parallel X_{ST}).\end{aligned}\quad (13)$$

Additionally, SD_j generates r_j and T_3 , then calculates

$$\begin{aligned}SK &= h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j)), \\ W_4 &= (PSMID_j \parallel r_j) \oplus X_{ST} \oplus \delta_j, \\ V_{DE} &= h(PSMID_j \parallel r_j \parallel \delta_j \parallel T_3).\end{aligned}\quad (14)$$

Finally, SD_j sends the message $M_3 = \{W_4, V_{DE}, T_3\}$ to EGW .

- (4) When EGW receives M_3 sent by SD_j , first checks $|T_3 - T_s| \leq \Delta T$. If T_3 is valid, EGW calculates

$$\begin{aligned}(PSMID_j \parallel r_j) &= W_4 \oplus X_{ST} \oplus \delta_j, \\ V_{DE} &= h(PSMID_j \parallel r_j \parallel \delta_j \parallel T_3).\end{aligned}\quad (15)$$

Next, EGW checks $V_{DE}^* \stackrel{?}{=} V_{DE}$. If it is correct, EGW authenticates the SD_j . Next, EGW generates a timestamp T_4 , and calculates

$$\begin{aligned}W_5 &= (PSMID_j \parallel r_j) \oplus RID_i \oplus X_{UT}, \\ V_{EU} &= h(PSMID_j \parallel r_j \parallel RID_i \parallel T_4).\end{aligned}\quad (16)$$

At last, EGW sends the message $M_4 = \{W_5, V_{EU}, T_4\}$ to U_i .

- (5) When receiving M_4 sent by EGW , U_i first checks $|T_4 - T_s| \leq \Delta T$. Then, U_i calculates

$$\begin{aligned}(PSMID_j \parallel r_j) &= W_5 \oplus RID_i \oplus X_{UT}, \\ V_{EU}^* &= h(PSMID_j \parallel r_j \parallel RID_i \parallel T_4).\end{aligned}\quad (17)$$

Finally, U_i checks $V_{EU}^* \stackrel{?}{=} V_{EU}$. If the verification is successful, U_i calculates

$$SK = h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j)). \quad (18)$$

The SK of the U_i and SD_j is successfully established, indicating the complete login and authentication process.

U_i	EGW	SD_j
Input UID_i and UPW_i $a_i = h(UID_i \parallel UPW_i) \oplus S_1$ $PID_i = h(UID_i \parallel a_i)$ $V_i^* = h(PID_i \parallel UPW_i \parallel a_i)$ Check $V_i^* \stackrel{?}{=} V_i$ $X_{UT} = R_1 \oplus h(a_i \parallel UPW_i)$ $RID_i = h(PID_i \parallel a_i \parallel X_{UT})$ Generate r_i and timestamp T_1 Choose $SMID_j$ $W_1 = (SMID_j \parallel r_i) \oplus h(X_{UT} \parallel T_1)$ $W_2 = RID_i \oplus X_{UT} \oplus T_1$ $V_{UE} = h(RID_i \parallel X_{UT} \parallel r_i \parallel T_1)$ $M_1 = \{W_1, W_2, PID_i, V_{UE}, T_1\}$	Check $ T_1 - T_s \leq \Delta T$ Retrieve $\{X_{UT}\}$ in database using PID_i $(SMID_j \parallel r_i) = W_1 \oplus h(X_{UT} \parallel T_1)$ $RID_i = W_2 \oplus X_{UT} \oplus T_1$ $V_{UE}^* = h(RID_i \parallel X_{UT} \parallel r_i \parallel T_1)$ Check $V_{UE}^* \stackrel{?}{=} V_{UE}$ Generate a timestamp T_2 Retrieve $\{C_j, \delta_j, X_{ST}\}$ in database using $SMID_j$ $W_3 = (C_j \parallel RID_i \parallel r_i) \oplus SMID_j \oplus \delta_j$ $V_{ED} = h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2)$ $M_2 = \{W_3, V_{ED}, T_2\}$	Check $ T_2 - T_s \leq \Delta T$ $X_{ST} = S_2 \oplus h(SMID_j \parallel b_j)$ $\delta_j = S_3 \oplus h(SMID_j \parallel b_j \parallel X_{ST})$ $(C_j \parallel RID_i \parallel r_i) = W_3 \oplus SMID_j \oplus \delta_j$ $V_{ED}^* = h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2)$ Check $V_{ED}^* \stackrel{?}{=} V_{ED}$ $\sigma_j = Rep(PUF(C_j), \delta_j)$ $PSMID_j = h(SMID_j \parallel \sigma_j \parallel X_{ST})$ Generate r_j and timestamp T_3 $SK = h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j))$ $W_4 = (PSMID_j \parallel r_j) \oplus X_{ST} \oplus \delta_j$ $V_{DE} = h(PSMID_j \parallel r_j \parallel \delta_j \parallel T_3)$ $M_3 = \{W_4, V_{DE}, T_3\}$
Check $ T_4 - T_s \leq \Delta T$ $(PSMID_j \parallel r_j) = W_5 \oplus RID_i \oplus X_{UT}$ $V_{EU}^* = h(PSMID_j \parallel r_j \parallel RID_i \parallel T_4)$ Check $V_{EU}^* \stackrel{?}{=} V_{EU}$ $SK = h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j))$	Check $ T_3 - T_s \leq \Delta T$ $(PSMID_j \parallel r_j) = W_4 \oplus X_{ST} \oplus \delta_j$ $V_{DE}^* = h(PSMID_j \parallel r_j \parallel \delta_j \parallel T_3)$ Check $V_{DE}^* \stackrel{?}{=} V_{DE}$ Generate a timestamp T_4 $W_5 = (PSMID_j \parallel r_j) \oplus RID_i \oplus X_{UT}$ $V_{EU} = h(PSMID_j \parallel r_j \parallel RID_i \parallel T_4)$ $M_4 = \{W_5, V_{EU}, T_4\}$	

Figure 5. Login and authentication phase.

4. Security Analysis

4.1. Formal Security Analysis

In this section, we verify the security of the proposed protocol by using the ROR [57–59] model. Under the ROR model, different rounds of games are set up to simulate whether an attacker (A) can crack the protocol in polynomial time and calculate the SK so as to verify the security of the proposed protocol.

Adversarial Model. In this paper, we use commonly used Dolev–Yao [60] and Canetti–Krawczyk [61] models. The following describes the capabilities of A in the above model.

- (1) A can eavesdrop, update, delete, intercept and modify information in the public channel.
- (2) A can steal the U_i 's mobile device and then through physical analysis to obtain U_i 's private information stored in the mobile device [62].

- (3) Through a dictionary attack, A can guess the U_i 's identity or password, but A cannot simultaneously speculate U_i 's identity and password.
- (4) A can obtain the temporary value of any entity.
- (5) A cannot access information stored in the EGW security database.

Security Model. The proposed protocol involves U_i , EGW , and SD_j . We define $\Pi_{U_i}^x$, Π_{EGW}^y , and $\Pi_{SD_j}^z$ represents the x -th U_i instance, the y -th EGW instance, and the z -th SD_j instance respectively. Here, assume that the A can implement the following operations under the ROR model.

- (1) *Execute*(E): A can eavesdrop on messages transmitted between entities, where $E = \{\Pi_{U_i}^x, \Pi_{EGW}^y, \Pi_{SD_j}^z\}$.
- (2) *Send*(E, M_i): A sends the message M_i to E and get E 's response.
- (3) *Hash*(*string*): A enters a string and obtain the string's hash value.
- (4) *Corrupt*($\Pi_{U_i}^x$): A can get U_i information stored in the mobile device.
- (5) *Test*(E): A guesses the correct SK by flipping the coin C . If $C = 1$, A can obtain the correct SK . If $C = 0$, A can obtain a random string with the same length as the SK .

According to both models, we adopt Theorem 1 to show the security of our proposed protocol.

Theorem 1. Under the ROR model, the advantages of the A 's ability to break the proposed protocol in polynomial time ξ are: $Adv_A^P(\xi) \leq \frac{q_h^2}{|Hash|} + \frac{q_s^2}{|PUF|} + 2C' \cdot q_{send}'$. Here, q_h refers to the number of hash operations performed, $|Hash|$ refers to the space of the hash function, $|PUF|$ refers to the PUF function, and C' and s' refer to two constants.

Proof. We defined five games: GM_0 - GM_4 to simulate the process of A attacking our proposed protocol. In the process of proof, $Succ_A^{GM_i}(\xi)$ is defined as the probability of A winning in GM_i , Adv_A^P is defined as the advantage of A to crack the protocol. The specific proof steps are as follows:

GM_0 : In GM_0 , A starts the game by tossing a coin C and does not perform any operation in the game. Therefore, we can obtain

$$Adv_A^P(\xi) = |2Pr[Succ_A^{GM_0}(\xi)] - 1|. \quad (19)$$

GM_1 : By executing *Execute*(E), A can eavesdrop $M_1 = \{W_1, W_2, PID_i, V_{UE}, T_1\}$, $M_2 = \{W_3, V_{ED}, T_2\}$, $M_3 = \{W_4, V_{DE}, T_3\}$, and $M_4 = \{W_5, V_{EU}, T_4\}$. When GM_1 at the end of the session, calculate the SK by executing *Test*(\cdot) query, where $SK = h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j))$. However, A cannot obtain values $\{RID_i, PSMID_j, r_i, r_j\}$, so A cannot calculate SK . Therefore, there is no difference between the probabilities of GM_1 and GM_0 :

$$Pr[Succ_A^{GM_1}(\xi)] = Pr[Succ_A^{GM_0}(\xi)]. \quad (20)$$

GM_2 : Add *Send*(\cdot) operation and *Hash*(\cdot) operation in GM_2 . Because the authentication values $\{V_{UE}, V_{ED}, V_{DE}, V_{EU}\}$ are composed of the private value generated by each entity and is secured by the hash function, so A cannot tamper with the message. In addition, the random number in the authentication value is different in each session, so a hash collision does not occur. Therefore, according to the birthday paradox, we can obtain

$$|Pr[Succ_A^{GM_2}(\xi)] - Pr[Succ_A^{GM_1}(\xi)]| \leq \frac{q_h^2}{2|Hash|}. \quad (21)$$

GM_3 : In GM_3 , the difference from GM_2 is to delete the *Hash*(\cdot) operation and add *PUF* query. As described in Section 1, according to the security attributes of the *PUF*(\cdot), we can obtain the probability of GM_3 as

$$|Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)]| \leq \frac{q_s}{2|PUF|}. \quad (22)$$

GM_4 : In GM_4 , A obtains the information $\{A_1, R_1, V_i\}$ in the mobile device by executing the *Corrupt*() query, and attempts to exploit the offline password guessing attacks to obtain the user's correct password UPW_i . Since A cannot obtain the U_i 's PID_i and random number a_i . The U_i 's password cannot be guessed. Therefore, according to Zipf's law [63], we can conclude that

$$|Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)]| \leq C' \cdot q_{send}^{s'} \quad (23)$$

Finally, A can only guess bit C to obtain the correct SK so as to win the game. Therefore, we can obtain

$$\begin{aligned} \frac{Adv_{\mathcal{A}}^{\mathcal{P}}(\xi)}{2} &= |Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - \frac{1}{2}| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)]| \\ &= |Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)]| \\ &\leq \sum_{i=0}^3 |Pr[Succ_{\mathcal{A}}^{GM_{i+1}}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_i}(\xi)]| \\ &= \frac{q_h^2}{2|Hash|} + \frac{q_s^2}{2|PUF|} + C' \cdot q_{send}^{s'} \end{aligned} \quad (24)$$

Finally, we can conclude that

$$Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq \frac{q_h^2}{|Hash|} + \frac{q_s^2}{|PUF|} + 2C' \cdot q_{send}^{s'} \quad (25)$$

□

4.2. Informal Security Analysis

MITM Attacks. It is assumed that A can intercept all information transmitted in the public channel. Let us take message M_2 as an example, message M_2 contains the authentication value $V_{ED} = h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2)$, A tried to tamper with the value of V_{ED} , but A does not know RID_i , δ_j , and X_{ST} , so A cannot tamper with the authentication value V_{ED} . Similarly, A cannot tamper with the message M_1 , M_3 , and M_4 . Therefore, evil intermediaries cannot break our protocol.

Smart Device Stolen Attacks. Suppose A obtains the information $\{S_2, S_3, b_j\}$, which is stored in the memory of S_j . Since each S_j is embedded with a PUF module, A is unable obtain the value of δ_j and A cannot calculate $PSMID_j$. Similarly, A cannot calculate RID_i and r_i , so A is incapable of successfully calculating SK. Thus, our protocol can resist smart device stolen attacks.

Temporary Value Disclosure Attacks. Suppose A can obtain the random number generated in any entity. Let us take A can obtain r_i generated by U_i as an example, where $SK = h((RID_i \oplus r_i) \parallel (PSMID_j \oplus r_j))$. Although A can intercept messages in the public channel, A cannot know RID_i , $PSMID_j$ and r_j , so A cannot figure out the correct SK. Similarly, even if A obtains r_j generated by S_j , it cannot figure out the correct SK. Therefore, even if A obtains the random number of any entity, it cannot break our protocol.

Replay Attacks. In our proposed protocol, each message delivered in the public channel contains a timestamp. When each entity receives a message, it first checks whether the timestamp is valid. The entity will perform subsequent calculations if the timestamp is within the valid range. Here, take message $M_2 = \{W_3, V_{ED}, T_2\}$ as an example. Suppose A intercepts the message M_2 and sends M_2 to S_j repeatedly. When S_j receiving M_2 sent by A , S_j first checks $|T_2 - T_s| \leq \Delta T$. S_j will terminate the session because the timestamp in message M_2 is not within the valid time range. Consequently, our proposed protocol can withstand replay attacks.

Mutual Authentication. In our proposed protocol, the validity of the entity is verified by the authentication value. The message passed in the public channel contains the authentication value, wherein

$$\begin{aligned} V_{UE} &= h(RID_i \parallel X_{UT} \parallel r_i \parallel T_1), \\ V_{ED} &= h(RID_i \parallel \delta_j \parallel X_{ST} \parallel T_2), \\ V_{DE} &= h(PSMID_j \parallel r_j \parallel RID_i \parallel T_3), \\ V_{EU} &= h(PSMID_j \parallel r_j \parallel RID_i \parallel T_4). \end{aligned} \quad (26)$$

EGW through calculation of V_{UE} verify the validity of U_i , S_j through calculation of V_{ED} verify the validity of EGW , EGW through calculation of V_{DE} verify the validity of S_j , U_i through calculation of V_{EU} verify the validity of EGW . Therefore, our protocol can ensure that each entity realizes mutual authentication.

Anonymity and Untraceability. In our proposed protocol, random numbers and hash functions are used to hide the real identities of U_i and S_j . The pseudonym of U_i and S_j are used in the authentication process. Even if the attacker intercepts the messages M_1 , M_2 , M_3 and M_4 transmitted in the public channel, it cannot track the U_i and S_j . In addition, random numbers are different during each session, ensuring that U_i and S_j are not traceable. As a result, the proposed protocol can guarantee the anonymity and untraceability of entities.

4.3. ProVerif

ProVerif [64,65] is a formal simulation tool developed by Bruno Blanchett for automatically verifying cryptographic protocols. It describes cryptographic primitives, such as hash functions, fuzzy extraction, etc. In this paper, we use ProVerif software to simulate the smart home environment, mainly by executing code to simulate the registration and authentication process of U_i , EGW , TTP , and SD_j to verify the security of our protocol.

The symbols and operations used in ProVerif are defined in Figure 6a. We use ProVerif to query whether A can calculate SK through the information transmitted on the public channel. Our proposed protocol proof includes six events: event UserStarted(), event UserAuthed(), event EGWAcUser(), event SmartdeviceAcEGW(), event EGWAcSmartdevice(), and event UserAcEGW(), which indicate that U_i starts authentication, U_i completes authentication, EGW completes the authentication of the U_i , SD_j completes the authentication of the EGW , EGW completes the authentication of SD_j , and U_i completes the authentication of the EGW . The specific query and event definitions are shown in Figure 6b.

The process of ProVerif simulating U_i , SD_j , TTP , and EGW in Figure 6c–e. TTP includes two sub-processes: U_i registration and SD_j registration. “UiReg” represents the user registration phase, and “SDjReg” represents the smart device registration phase. ProVerif describes the detailed steps of each entity, such as the definition of new parameters and sending and receiving messages. Take the U_i process as an example, where “new UIDi: bitstring” represents the definition of the U_i identity, “out (sch, (PIDi, UPWi, ai))” represents that the U_i sends messages to EGW , and “in (sch, (xR1: bitstring))” means that the U_i receives messages sent from EGW . Finally, we use ProVerif to verify the proposed protocol, as shown in Figure 6f. We can conclude from the results that A cannot calculate SK , which proves that we propose a secure protocol.

According to the presentations in Sections 4.1–4.3, we demonstrated the security of our protocol in terms of formal proof (using RoR model), informal proof, and simulation software (ProVerif). The results show that the proposed authentication protocol can resist several well-known attacks, such as insider, gateway impersonation, session key disclosure, offline password guessing, and replay, and provides mutual authentication, anonymity, and untraceability.

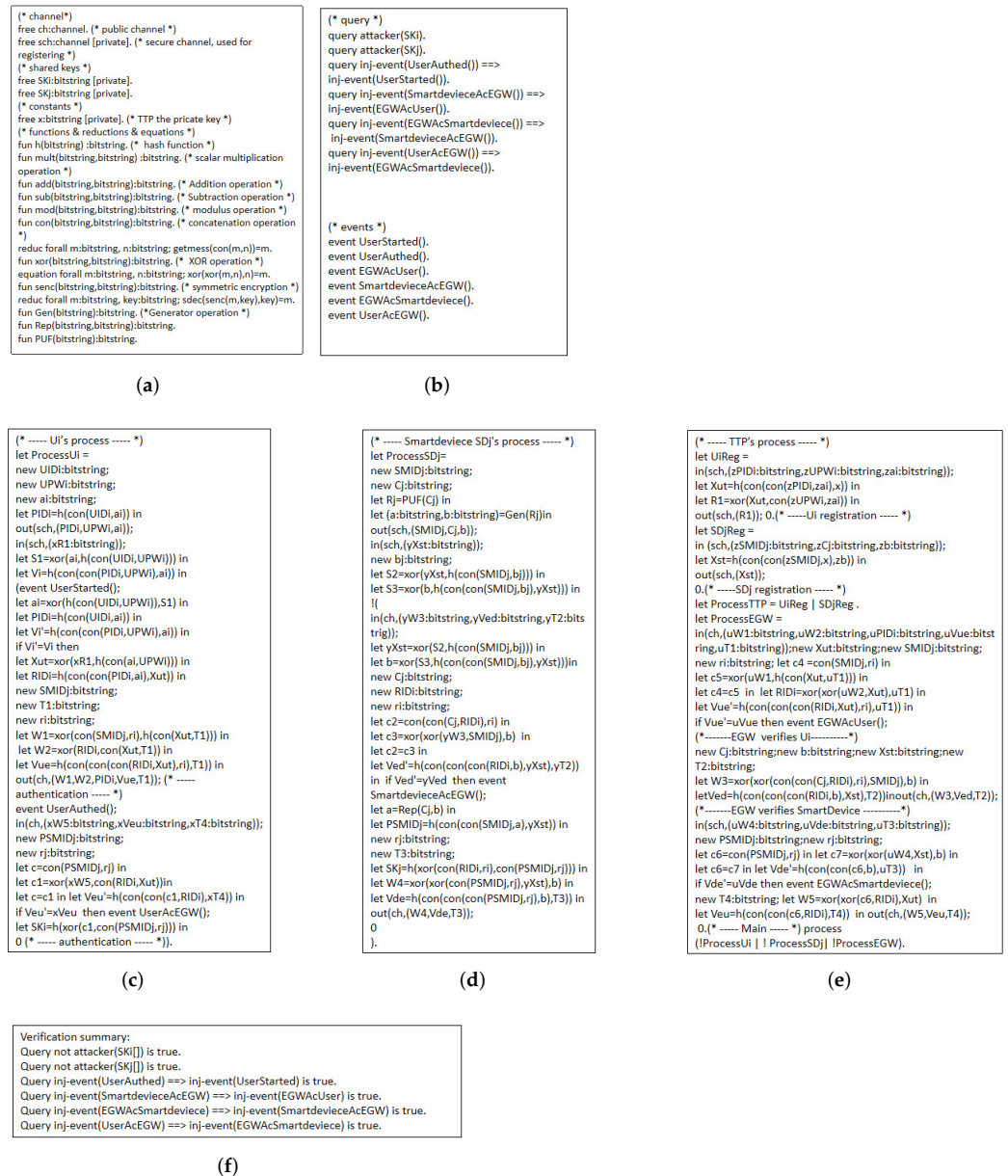


Figure 6. Simulation process in ProVerif. (a) Definitions; (b) the queries and events; (c) execution process of U_i ; (d) execution process of SD_j ; (e) execution process of TTP and EGW ; (f) verification results.

5. Security and Performance Comparisons

In this section, we compare the proposed protocol with four existing related protocols [18,37,40,42] in terms of security and performance.

5.1. Security Comparisons

We compare the security of our proposed protocol with that of Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42]. Table 3 shows the security comparison results. \checkmark demonstrates that the protocol can resist this attack, and \times demonstrates that the protocol suffers from this attack. Shuai et al.'s protocol [37] suffers from insider attacks, gateway impersonation attacks, session key disclosure attacks, offline password guessing attacks, and replay attacks. Banerjee et al.'s protocol [40] cannot provide anonymity and untraceability. Yu et al.'s protocol [18] is unable to provide mutual authentication. Oh et al. [42] and our protocol can resist these attacks.

Table 3. Comparison of security.

Security Properties	[37]	[40]	[18]	[42]	Ours
Insider Attacks	×	✓	✓	✓	✓
Gateway Impersonation Attacks	×	✓	✓	✓	✓
Session Key Disclosure Attacks	×	✓	✓	✓	✓
Offline Password Guessing Attacks	×	✓	✓	✓	✓
Replay Attacks	×	✓	✓	✓	✓
Mutual Authentication	✓	✓	×	✓	✓
Anonymity and untraceability	✓	×	✓	✓	✓

5.2. Performance Comparisons

We compare the performance from two aspects: computational cost and communication cost.

5.2.1. Computational Cost Comparisons

We compare and analyze the computational costs of each protocol in the login and authentication phase. Additionally, we perform simulation experiments to evaluate the computational cost of the protocol. We use HONOR Play3 to simulate users, Lenovo desktop to simulate edge gateway, and Lenovo laptop to simulate smart devices. The specific configuration of these three devices is shown in Table 4, where the operation time is obtained by averaging 20 times of operation. Here we will ignore hash and join operations. We can see the comparison results of the computational cost from Table 5. Because the running time of the fuzzy extractor is almost the same as that of the hash function, we use the hash function's running time to represent the fuzzy extractor's running time in the calculation cost comparison.

Table 4. Configuration parameters and running time of equipment.

	HONOR Play3	Lenovo Desktop	Lenovo Laptop
Operating System	Android System	Windows 10	Windows 10
Running Memory	4G	16G	8G
CPU	HUAWEI Kirin 710F	Intel(R) Core(TM) i5-9500 CPU @ 3.00 GHz	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz
Hash Function	0.0041 ms	0.0024 ms	0.0035 ms
Point Multiplication	0.5354 ms	0.3354 ms	0.4129 ms
Point Addition	0.1604 ms	0.0633 ms	0.0977 ms

Table 5. Computational cost comparison.

Protocols	U_i (ms)	EGW (ms)	SD_j (ms)
Shuai et al. [37]	$2T_C + 6T_H \approx 1.095$	$T_C + 7T_H \approx 0.556$	$3T_H \approx 0.012$
Banerjee et al. [40]	$10T_H + T_P \approx 0.045$	$9T_H \approx 0.22$	$4T_H \approx 0.014$
Yu et al. [18]	$T_D + 12T_H + T_P \approx 0.309$	$11T_H \approx 0.045$	$7T_H \approx 0.029$
Oh et al. [42]	$16T_H \approx 0.066$	$15T_H \approx 0.036$	$8T_H \approx 0.028$
Our	$9T_H \approx 0.037$	$5T_H \approx 0.020$	$6T_H + T_P \approx 0.029$

Here, T_C represents the execution time of ECC point multiplication, T_D represents the execution time of symmetric encryption/decryption operation, T_H represents the running time of hash function, and T_P represents the execution time of the fuzzy extraction function.

In the framework of the smart home environment, there can be multiple U_i and SD_j and only one edge gateway. We describe the relationship between the change in the number of entities and the calculated cost as follows. The relationship between the number of U_i and the computational cost is shown in Figure 7. Shuai et al. [37] used point multiplication in the protocol, so the computational cost of this protocol is higher than that of other protocols. Yu et al. [18] used symmetric key encryption/decryption and fuzzy extractor in the protocol, and its computational cost is lower than that of Shuai et al. [37]. Moreover,

the computational cost of other protocols is not different. The computational cost of *EGW* is shown in Figure 8. We can conclude from Figure 8 that the *EGW* computational cost of the proposed protocol is lower than that of other protocols. The relationship between the number of SD_j and the computational cost is shown in Figure 9. We can conclude from Figure 9 that the SD_j computational cost of the proposed protocol is lower than that of Oh et al.’s protocol [42], the same as that of Yu et al.’s protocol [18], but slightly higher than that of other protocols.

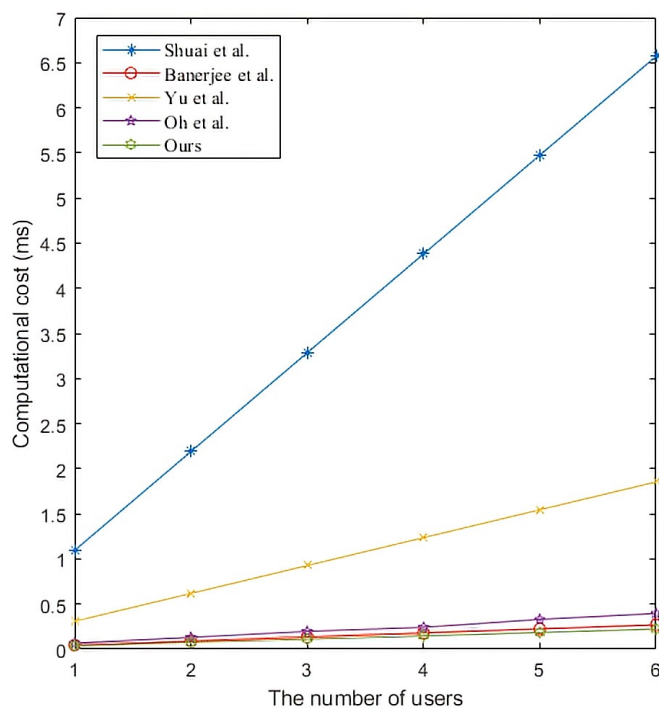


Figure 7. The computational cost of users. Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42].

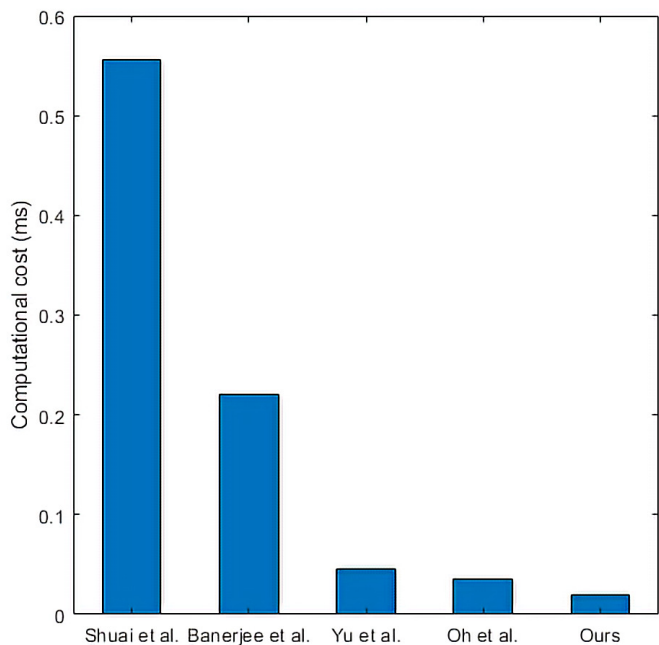


Figure 8. The computational cost of EGW. Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42].

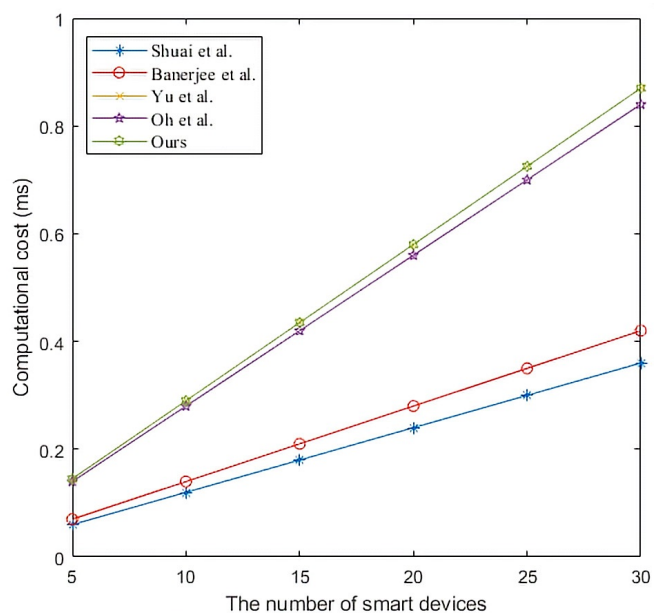


Figure 9. The computational cost of smart devices. Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42].

5.2.2. Communication Cost Comparisons

This part assumes that the length of timestamp, random number, identity, hash function, point multiplication, and symmetric encryption/decryption are 32, 128, 160, 256, 320, and 256 bits. Take our protocol as an example to explain the calculation process of communication cost. In our protocol, the messages transmitted in the public channel are $M_1 = \{W_1, W_2, PID_i, V_{UE}, T_1\}$, $M_2 = \{W_3, V_{ED}, T_2\}$, $M_3 = \{W_4, V_{DE}, T_3\}$, $M_4 = \{W_5, V_{EU}, T_4\}$. Where, PID_i is the identity, $\{W_1, W_2, W_3, W_4, W_5\}$ are random numbers, $\{V_{UE}, V_{ED}, V_{DE}, V_{EU}\}$ are hash functions, $\{T_1, T_2, T_3, T_4\}$ are time stamps. It is calculated that the communication cost of our protocol is 1952 bits. The communication costs of Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42] are 2016, 1696, 1792, and 2368 bits, respectively. We can draw a conclusion from Table 6 and Figure 10 that the communication cost of the proposed protocol is lower than that of Shuai et al. [37] and Oh et al. [42], and slightly higher than that of Banerjee et al. [40] and Yu et al. [18].

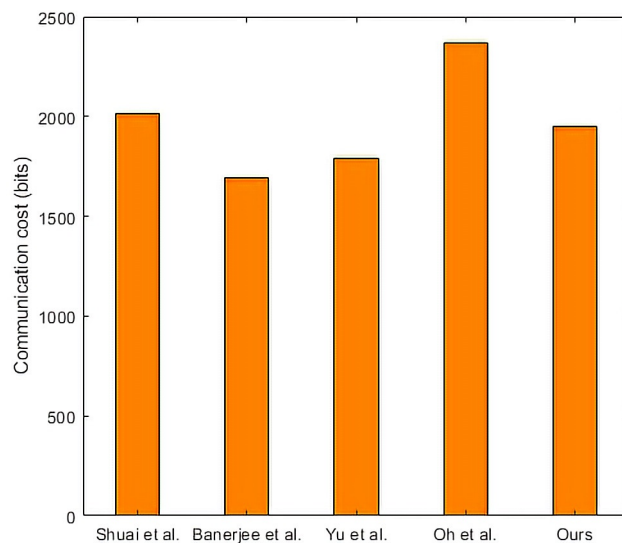


Figure 10. Comparisons of communication cost. Shuai et al. [37], Banerjee et al. [40], Yu et al. [18], and Oh et al. [42].

Table 6. Communication cost comparison.

Protocols	Rounds	Communication Cost
Shuai et al. [37]	4	2016 bits
Banerjee et al. [40]	4	1696 bits
Yu et al. [18]	4	1792 bits
Oh et al. [42]	5	2368 bits
Our	4	1952 bits

6. Conclusions

Communication security is an essential factor for the sustainable development of smart homes. It ensures that users can obtain secure smart home services and protects users' privacy. Due to the openness of wireless channels prone to data leakage, using cryptographic methods to ensure communication security has attracted many researchers' attention. To the best of our knowledge, we introduce the first edge-computing-based smart home architecture. Meanwhile, based on this architecture, a PUF-based authentication protocol is proposed. Precisely, the properties of PUF are provided to resist physical tampering and biological cloning attacks. The standard security verification approaches which are formal security analysis using RoR model, informal security analysis, and ProVerif simulation software are made to demonstrate the security of our protocol. The security and performance comparisons are indicated that our protocol has higher security and slightly better performance. In the future, we will adopt several lightweight cryptographic operations to design the new authentication protocol in smart home environments. Without loss of security, the new protocol is more suitable for users' IoT devices.

Author Contributions: Conceptualization, T.-Y.W.; methodology, T.-Y.W. and F.K.; software, L.W.; formal analysis, Y.-C.C.; investigation, S.K. and J.-S.P.; writing—original draft preparation, T.-Y.W., F.K., L.W., Y.-C.C., S.K. and J.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is included in the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
ROR	Real or random
RA	Registration authority
PUF	Physical unclonable functions
AKA	Authentication and key agreement
ECC	Elliptic curve cryptography

References

- Chen, X.; Zhang, J.; Lin, B.; Chen, Z.; Wolter, K.; Min, G. Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 683–697. [[CrossRef](#)]
- Shen, S.; Yang, Y.; Liu, X. Toward data privacy preservation with ciphertext update and key rotation for IoT. *Concurr. Comput. Pract. Exp.* **2021**, e6729. [[CrossRef](#)]
- Namasudra, S. A secure cryptosystem using DNA cryptography and DNA steganography for the cloud-based IoT infrastructure. *Comput. Electr. Eng.* **2022**, *104*, 108426. [[CrossRef](#)]
- Wu, T.Y.; Meng, Q.; Kumari, S.; Zhang, P. Rotating behind Security: A Lightweight Authentication Protocol Based on IoT-Enabled Cloud Computing Environments. *Sensors* **2022**, *22*, 3858. [[CrossRef](#)] [[PubMed](#)]

5. Yu, Z.; Zheng, X.; Huang, F.; Guo, W.; Sun, L.; Yu, Z. A framework based on sparse representation model for time series prediction in smart city. *Front. Comput. Sci.* **2021**, *15*, 1–13. [[CrossRef](#)]
6. Huang, H.; Lu, S.; Wu, Z.; Wei, Q. An efficient authentication and key agreement protocol for IoT-enabled devices in distributed cloud computing architecture. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 1–21. [[CrossRef](#)]
7. Luo, Y.; Zheng, W.M.; Chen, Y.C. An anonymous authentication and key exchange protocol in smart grid. *J. Netw. Intell.* **2021**, *6*, 206–215.
8. Liu, S.; Chen, C.M. Comments on “A Secure and Lightweight Drones-Access Protocol for Smart City Surveillance”. *IEEE Trans. Intell. Transp. Syst.* **2022**. [[CrossRef](#)]
9. Yang, Y.; Zheng, X.; Guo, W.; Liu, X.; Chang, V. Privacy-preserving fusion of IoT and big data for e-health. *Future Gener. Comput. Syst.* **2018**, *86*, 1437–1455. [[CrossRef](#)]
10. Wu, T.Y.; Yang, L.; Luo, J.N.; Ming-Tai Wu, J. A Provably Secure Authentication and Key Agreement Protocol in Cloud-Based Smart Healthcare Environments. *Secur. Commun. Netw.* **2021**, 2299632. [[CrossRef](#)]
11. Das, S.; Namasudra, S. MACPABE: Multi-Authority-based CP-ABE with efficient attribute revocation for IoT-enabled healthcare infrastructure. *Int. J. Netw. Manag.* **2022**. [[CrossRef](#)]
12. Jiang, Q.; Zhang, X.; Zhang, N.; Tian, Y.; Ma, X.; Ma, J. Two-factor authentication protocol using physical unclonable function for IoV. In Proceedings of the 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 11–13 August 2019; pp. 195–200. [[CrossRef](#)]
13. Yu, S.; Lee, J.; Park, K.; Das, A.K.; Park, Y. IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. *IEEE Access* **2020**, *8*, 167875–167886. [[CrossRef](#)]
14. Chaudhry, S.A. Combating identity de-synchronization: An improved lightweight symmetric key based authentication scheme for IoV. *J. Netw. Intell.* **2021**, *6*, 656–667.
15. Kumar, V.; Kumar, R.; Kumar, V.; Kumari, A.; Kumari, S. RAVCC: Robust Authentication Protocol for RFID based Vehicular Cloud Computing. *J. Netw. Intell.* **2022**, *7*, 526–543.
16. Li, Z.; Miao, Q.; Chaudhry, S.A.; Chen, C.M. A provably secure and lightweight mutual authentication protocol in fog-enabled social Internet of vehicles. *Int. J. Distrib. Sens. Netw.* **2022**, *18*, 15501329221104332. [[CrossRef](#)]
17. Naoui, S.; Elhdhili, M.E.; Saidane, L.A. Lightweight and secure password based smart home authentication protocol: LSP-SHAP. *J. Netw. Syst. Manag.* **2019**, *27*, 1020–1042. [[CrossRef](#)]
18. Yu, S.; Jho, N.; Park, Y. Lightweight three-factor-based privacy-preserving authentication scheme for iot-enabled smart homes. *IEEE Access* **2021**, *9*, 126186–126197. [[CrossRef](#)]
19. Tanveer, M.; Abbas, G.; Abbas, Z.H.; Bilal, M.; Mukherjee, A.; Kwak, K.S. LAKE-6SH: Lightweight user authenticated key exchange for 6LoWPAN-based smart homes. *IEEE Internet Things J.* **2021**, *9*, 2578–2591. [[CrossRef](#)]
20. Xue, X.; Jiang, C. Matching sensor ontologies with multi-context similarity measure and parallel compact differential evolution algorithm. *IEEE Sens. J.* **2021**, *21*, 24570–24578. [[CrossRef](#)]
21. Xue, X.; Huang, Q. Generative adversarial learning for optimizing ontology alignment. *Expert Syst.* **2022**. [[CrossRef](#)]
22. Yang, Y.; Zheng, X.; Chang, V.; Ye, S.; Tang, C. Lattice assumption based fuzzy information retrieval scheme support multi-user for secure multimedia cloud. *Multimed. Tools Appl.* **2018**, *77*, 9927–9941. [[CrossRef](#)]
23. Zhang, J.; Li, M.; Chen, Z.; Lin, B. Computation offloading for object-oriented applications in a UAV-based edge-cloud environment. *J. Supercomput.* **2022**, *78*, 10829–10853. [[CrossRef](#)]
24. Wu, T.Y.; Wang, L.; Guo, X.; Chen, Y.C.; Chu, S.C. SAKAP: SGX-Based Authentication Key Agreement Protocol in IoT-Enabled Cloud Computing. *Sustainability* **2022**, *14*, 11054. [[CrossRef](#)]
25. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [[CrossRef](#)]
26. Li, Y.; Cheng, Q.; Liu, X.; Li, X. A secure anonymous identity-based scheme in new authentication architecture for mobile edge computing. *IEEE Syst. J.* **2020**, *15*, 935–946. [[CrossRef](#)]
27. Chen, X.; Chen, S.; Ma, Y.; Liu, B.; Zhang, Y.; Huang, G. An adaptive offloading framework for android applications in mobile edge computing. *Sci. China Inf. Sci.* **2019**, *62*, 1–17. [[CrossRef](#)]
28. Liu, G.; Chen, X.; Zhou, R.; Xu, S.; Chen, Y.C.; Chen, G. Social learning discrete Particle Swarm Optimization based two-stage X-routing for IC design under Intelligent Edge Computing architecture. *Appl. Soft Comput.* **2021**, *104*, 107215. [[CrossRef](#)]
29. Wu, T.Y.; Meng, Q.; Yang, L.; Guo, X.; Kumari, S. A provably secure lightweight authentication protocol in mobile edge computing environments. *J. Supercomput.* **2022**, *78*, 13893–13914. [[CrossRef](#)]
30. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical one-way functions. *Science* **2002**, *297*, 2026–2030. [[CrossRef](#)]
31. Tahavori, M.; Moazami, F. Lightweight and secure PUF-based authenticated key agreement scheme for smart grid. *Peer-to-Peer Netw. Appl.* **2020**, *13*, 1616–1628. [[CrossRef](#)]
32. 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), Walkoloa, HI, USA, 7–10 January 2008; p. 294. [[CrossRef](#)]
33. Vaidya, B.; Park, J.H.; Yeo, S.S.; Rodrigues, J.J. Robust one-time password authentication scheme using smart card for home network environment. *Comput. Commun.* **2011**, *34*, 326–336. [[CrossRef](#)]

34. Kim, H.J.; Kim, H.S. AUTH HOTP-HOTP based authentication scheme over home network environment. In Proceedings of the International Conference on Computational Science and Its Applications, Santander, Spain, 20–23 June 2011; pp. 622–637. [[CrossRef](#)]
35. 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](#)]
36. Lyu, Q.; Zheng, N.; Liu, H.; Gao, C.; Chen, S.; Liu, J. Remotely access “my” smart home in private: An anti-tracking authentication and key agreement scheme. *IEEE Access* **2019**, *7*, 41835–41851. [[CrossRef](#)]
37. 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](#)]
38. 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](#)]
39. Alzahrani, B.A.; Barnawi, A.; Albarakati, A.; Irshad, A.; Khan, M.A.; Chaudhry, S.A. SKIA-SH: A Symmetric Key-Based Improved Lightweight Authentication Scheme for Smart Homes. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8669941. [[CrossRef](#)]
40. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Park, Y. An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors* **2020**, *20*, 1215. [[CrossRef](#)]
41. Fadi, A.T.; Deebak, B.D. Seamless authentication: For IoT-big data technologies in smart industrial application systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 2919–2927.
42. Oh, J.; Yu, S.; Lee, J.; Son, S.; Kim, M.; Park, Y. A secure and lightweight authentication protocol for IoT-based smart homes. *Sensors* **2021**, *21*, 1488. [[CrossRef](#)]
43. Tsai, J.L.; Lo, N.W. A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2015**, *9*, 805–815. [[CrossRef](#)]
44. Jiang, Q.; Ma, J.; Wei, F. On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Syst. J.* **2016**, *12*, 2039–2042. [[CrossRef](#)]
45. Irshad, A.; Sher, M.; Ahmad, H.F.; Alzahrani, B.A.; Chaudhry, S.A.; Kumar, R. An improved multi-server authentication scheme for distributed mobile cloud computing services. *KSII Trans. Internet Inf. Syst. (TIIS)* **2016**, *10*, 5529–5552.
46. Xiong, L.; Peng, D.; Peng, T.; Liang, H. An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services. *KSII Trans. Internet Inf. Syst. (TIIS)* **2017**, *11*, 6169–6187.
47. Jia, X.; He, D.; Kumar, N.; Choo, K.K.R. A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing. *IEEE Syst. J.* **2019**, *14*, 560–571. [[CrossRef](#)]
48. Kaur, K.; Garg, S.; Kaddoum, G.; Guizani, M.; Jayakody, D.N.K. A lightweight and privacy-preserving authentication protocol for mobile edge computing. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [[CrossRef](#)]
49. Aysu, A.; Gulcan, E.; Moriyama, D.; Schaumont, P.; Yung, M. End-to-end design of a PUF-based privacy preserving authentication protocol. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Saint-Malo, France, 13–16 September 2015; pp. 556–576. [[CrossRef](#)]
50. Chatterjee, U.; Chakraborty, R.S.; Mukhopadhyay, D. A PUF-based secure communication protocol for IoT. *ACM Trans. Embed. Comput. Syst. (TECS)* **2017**, *16*, 1–25. [[CrossRef](#)]
51. Braeken, A. PUF based authentication protocol for IoT. *Symmetry* **2018**, *10*, 352. [[CrossRef](#)]
52. Gope, P.; Das, A.K.; Kumar, N.; Cheng, Y. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4957–4968. [[CrossRef](#)]
53. Chen, S.; Li, B.; Chen, Z.; Zhang, Y.; Wang, C.; Tao, C. Novel Strong-PUF-based Authentication Protocols Leveraging Shamir’s Secret Sharing. *IEEE Internet Things J.* **2021**, *9*, 14408–14425. [[CrossRef](#)]
54. Ebrahimabadi, M.; Younis, M.; Karimi, N. A PUF-based modeling-attack resilient authentication protocol for IoT devices. *IEEE Internet Things J.* **2021**, *9*, 3684–3703. [[CrossRef](#)]
55. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and Lightweight Authentication Protocol Using Physical Unclonable Functions for Internet of Drones in Smart City Environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388. [[CrossRef](#)]
56. Shao, X.; Guo, Y.; Guo, Y. A PUF-based anonymous authentication protocol for wireless medical sensor networks. *Wirel. Netw.* **2022**, *28*, 3753–3770. [[CrossRef](#)]
57. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Proceedings of the International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84. [[CrossRef](#)]
58. Wu, T.Y.; Lee, Z.; Yang, L.; Luo, J.N.; Tso, R. Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks. *J. Supercomput.* **2021**, *77*, 6992–7020. [[CrossRef](#)]
59. Wu, T.Y.; Meng, Q.; Yang, L.; Kumari, S.; Nia, M.P. Amassing the Security: An Enhanced Authentication and Key Agreement Protocol for Remote Surgery in Healthcare Environment. *Comput. Model. Eng. Sci.* **2023**, *134*, 317–341. [[CrossRef](#)]
60. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [[CrossRef](#)]
61. 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. [[CrossRef](#)]

62. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [[CrossRef](#)]
63. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf’s law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [[CrossRef](#)]
64. Blanchet, B. An efficient cryptographic protocol verifier based on prolog rules. In Proceedings of the 14th IEEE Computer Security Foundations Workshop, Cape Breton, NS, Canada, 11–13 June 2001; Volume 1, pp. 82–96.
65. Yang, L.; Chen, Y.C.; Wu, T.Y. Provably Secure Client-Server Key Management Scheme in 5G Networks. *Wirel. Commun. Mob. Comput.* **2021**, 4083199. [[CrossRef](#)]