LR-AKAP: A Lightweight and Robust Security Protocol for Smart Home Environments

Rana Muhammad Abdul Haseeb-ur-rehman 1,*, Misbah Liaqat 2, Azana Hafizah Mohamed Aman 1, Abdulwahab Ali Almazroi 2, Mohammad Kamrul Hasan 1, Zeeshan Ali 3 and Rana Liaqat Ali 4, *

1 Faculty of Information Science and Technology, University Kebangsaan Malaysia, Bangi 43600, Malaysia
2 Department of Information Technology, College of Computing and Information Technology, University of Jeddah, Khulais 21959, Saudi Arabia
3 Department of Computer Science and Software Engineering, International Islamic University Islamabad, Islamabad 44000, Pakistan
4 Department of Physics, COMSATS University Islamabad, Islamabad 45550, Pakistan
* Correspondence: p108391@siswa.ukm.edu.my (R.M.A.H.-u.-r.); liaqat.ali@comsats.edu.pk (R.L.A.)

Abstract: For the betterment of human life, smart Internet-of-Things (IoT)-based systems are needed for the new era. IoT is evolving swiftly for its applications in the smart environment, including smart airports, smart buildings, smart manufacturing, smart homes, etc. A smart home environment includes resource-constrained devices that are interlinked, monitored, controlled, and analyzed with the help of the Internet. In a distributed smart environment, devices with low and high computational power work together and require authenticity. Therefore, a computationally efficient and secure protocol is needed. The authentication protocol is employed to ensure that authorized smart devices communicate with the smart environment and are accessible by authorized personnel only. We have designed a novel, lightweight secure protocol for a smart home environment. The introduced novel protocol can withstand well-known attacks and is effective with respect to computation and communication complexities. Comparative, formal, and informal analyses were conducted to draw the comparison between the introduced protocol and previous state-of-the-art protocols.

Keywords: Internet-of-Things (IoT); automation; sensors; smart home; authentication; protocols; session key; stolen verifier; impersonation; security; AVISPA

1. Introduction

Human life is enviable. Science and technology play a vital role in providing safety to human life by developing safer devices using smarter technology. [1,2]. The advances in wireless and sensor technology have played an indispensable role in the rapid adaption of the smart home as an interesting new paradigm of the Internet of Things (IoT) [3–8]. This results in gaining a considerable amount of interest from both industry and academic fields alike [9].

A smart-home environment is used to assist individuals in various tasks by appending automation to them, like a smart meter, which can help users by showing the real-time consumption of power, voltage, and load [10,11] along with other interconnected devices. These devices can be various sensors, like a sensor to measure humidity, temperature, occupancy, and even to monitor the equipment. Multiple services can be provided by employing these services, such as turning on the air conditioner (AC) when an individual enters the room or turning on the lights when a person opens the door, and carbon dioxide (CO2) sensors can be employed to monitor the environment and to trigger automatic responses to energy saving [12,13]. In addition, these various smart gadgets can be deployed to oversee the health of the elderly [14]. Smart-home devices can also be accessed through a user’s device, such as a smartphone, without the restriction of place or time.
An innovative irrigation system may assist in conserving tens of dollars each year in addition to thousands of gallons of water. Cyber Rain [15], for example, allows observing and controlling the system from any Internet-connected device and can be customized to fit any size yard. By analyzing the weather prediction for you and modifying the watering plan automatically when rain is noticed, systems like Cyber Rain save the Earth’s limited supply of freshwater. According to the company’s website, users of Cyber Rain reportedly consume 35% less water on average.

Linked feeders provide automatic pet care. Using linked timers, you may water your yard and inside plants. There are many different types of kitchen equipment useable, such as smart coffee machines that can mix a fresh cup automatically at a set time, smart refrigerators that keep track of expiration dates, generate grocery lists, or even create ingredients using ingredients already in the house, toasters, and delayed cookers, and washing machines and dryers for the laundry room [16].

A wide range of applications, from business to government and military uses, may be realized with secure and continuous user access to designated smart home appliances [17]. Due to the sensitive nature of the operations and inherited vulnerabilities of the public channel, such as real-time data manipulation, clogging, replay, jamming, etc., the use of these smart home appliances for such applications is otherwise regarded as problematic [18,19]. Although various security schemes for the smart-home environment have recently been developed, many of these schemes have turned out to be insecure or unworkable. Consequently, the current demand is for an authentication system that can guarantee both security and privacy. To support the automatic security proof of the suggested method, we used a well-known automated program, AVISPA. The research showed that the suggested system could withstand known assaults and offered excellent balance between security and effectiveness [20–23].

A typical architecture for a smart home is presented in Figure 1. The network model comprises four components that are (i) the end user (\( U_i \)), (ii) a trusted authority (TA), (iii) a gateway node (GWN), and (iv) smart devices (SD). The SDs have limited resources and can be used for various tasks such as controlling the temperature, turning on lights [20], etc. The GWN is used as an access point and has enough computation power; it acts as a gateway for \( U_i \) and SD. The TA is trusted by all the parties involved in the communication. The responsibilities he TA include (i) system parameters’ generation and assignment, (ii) smart device and user registration, and (iii) key maintenance and management.

![Smart home environment architecture diagram](adapted from [12]).
This study is arranged as follows: Section 2 presents the motivation and contribution of our research. Section 3 shortly investigates the existing literature. Section 4 presents our proposed protocol. Moreover, Section 5 discusses the security analysis. Section 6 presents the performance analysis. In Section 7, the whole work is summed up.

2. Motivations and Contributions
This section concisely presents the primary contributions of this study:

1. We suggested an authentication protocol based on a symmetric key to protect the user–smart device connection. The system was created using a mobile device, biometrics, and a password.
2. The security of the proposed work was analyzed by using an automated security tool AVISPA.
3. The analysis showed that the suggested system could withstand known attacks and offered an excellent balance between security and effectiveness.
4. The computation cost of the proposed protocol was less than that of the protocols with the smart devices as they were resource-constrained and had limited computational power.
5. The proposed protocol evaded the clock synchronization issue in timestamp-based two-way authentication protocols.
6. Our presented protocol was compared to others in a similar field, showing that the proposed protocol was better in terms of security and performance analysis.

3. Related Work
Recently, various protocols have been introduced to provide security in smart home environments [24,25]. Every protocol has its strengths and weaknesses. Sciancalepore et al. [26] used ECQV and ECDH certificates to accomplish authentication. However, two distinct keys are required to run the scheme, and the efficiency of the key creation function relies on the key derivation function (KDF). Any problem in the KDF may result in a connection interruption between entities. All the keys are generated by using the master key; hence, disclosure of the master key may cause forward secrecy [27]. In [28], the authors present an authentication protocol for an IoT-enabled smart home. The authors state that their protocol is secure and lightweight against numerous attacks. However, Ref. [27] found that their protocol lacks confidentiality and resistance against a Denial-of-Service attack (DoS) and some known key attacks. Additionally, the communication and computation costs are very high, which is not recommended for a resource-limited environment such as a smart-home environment. Dey and Hossain [29] presented a protocol using a new security model for smart homes. Gaba et al. [27] said that their protocol is efficient in computation and communication cost lacks message freshness and cannot resist a known-key attack.

Kumar et al. [30] introduced a new authentication mechanism for an SH environment. In this scheme, the SK is created by utilizing a small token to achieve authentication and the session key. This protocol is secure against various well-known attacks and lightweight in terms of computation and communication costs. However, after analysis, it is evident that their protocol is vulnerable to time synchronization attacks, replay attacks, does not provide anonymity, and has unlink ability issues. Kumar et al. [31] presented a new improved protocol in which the session key is renewed continuously to resist replay attacks. However, their protocol’s computation and communication costs are still very high. Gope and Sikdar [32] introduced a new protocol and claim that it is secure against impersonation, traceability, and physical attacks. However, due to enormous use of hash operations and high communication complexities, their protocol is not recommended for smart-home environments.

Wazid et al. [33] presented a protocol for smart homes. In this study, the use of an XOR operation one-way hash function makes it accessible for smart sensors. In contrast, their scheme is susceptible to a stolen verifier attack because verifier table is stored on the
GWN and is vulnerable to a synchronization attack because a time stamp is employed to stop a replay attack. Shuai et al. [34] presented a protocol using ECC for a smart-home environment. They eliminated the need of a verifier table for authentication. However, their protocol incurs high communication and computation costs, which is not recommended in a smart-home environment. Banerjee et al. [23] also proposed a scheme to secure smart-home environments. They stated that their scheme can bear well-known attacks but, in this study, it is shown that their scheme is prone to a stolen gateway node attack, a stolen verifier attack, a gateway node impersonation attack towards a smart device, a broadcast issue which causes inefficient resource utilization, a smart device impersonation and a parallel session attack, and a gateway node impersonation attack towards the user. In this study, a three-factor-based enhanced scheme is introduced to overcome the shortfalls.

4. Proposed Protocol

A lightweight and robust security protocol for smart-home environments (LR-AKAP) is presented, which not only withstands well-known active and passive attacks, but also attains the required functional qualities. The proposed protocol includes six processes: (i) the initialization process, (ii) the smart device enrolment process, (iii) the gateway node enrolment process, (iv) the user enrolment process, (v) the login and authentication process, and (vi) the password and biometric update process. Various notations used in this paper are listed in Table 1. The proposed architecture of the scheme is depicted in Figure 2.

Table 1. Notations guide.

| Symbols                          | Representations                                      |
|---------------------------------|------------------------------------------------------|
| $ID_u, PW_u, Bio_u$             | $i^{th}$ users biometric and password, user’s identity|
| $PID_u$                         | Temporary identity of a user                         |
| $SD, SID_d$                     | Smart sensor and its identity                        |
| $GWN, GID_w$                    | Gateway node and its identity                        |
| $R_u, R_d$                      | Random numbers                                       |
| $K$                             | 1024 bit – long secret key of $TA$                   |
| $SK_{du} (= SK_{ud})$           | Session key between $U$ and $SD$                     |
| $h(.)$                          | Cryptographic one-way hash function                  |
| $T_{iu}, T_{iD}, T_u, T_d$      | Current timestamps                                   |
| $Gen(\cdot), Rep(\cdot)$       | Fuzzy extractor probabilistic generation and deterministic reproduction function |
| $\Delta T$                     | Maximum allowable transmission delay                 |
| $i \equiv j$                    | Checks if $i$ equals to $j$                          |
| $||, \oplus$                    | Concatenation and bitwise XOR operators              |
| $A, I, U_A$                     | An adversary, intruder, and privileged insider        |

![Proposed architecture for a smart-home environment.](image-url)
4.1. Assumptions

The following are the assumptions which are taken into consideration to design the introduced protocol:

1. The GWN is trusted and there is no energy restriction. Nevertheless, the sensor nodes (smart devices) are powered by batteries, and they have very limited resources [35].
2. An $\mathcal{A}$ may inaugurate only external attacks by employing powerful devices [36].
3. The GWN is under the user’s possession, and any wicked intruder cannot control it [37].

4.2. Adversarial Model

1. A common adversarial model [17–19], Refs. [20–22] is considered in this article, where the adversary $\mathcal{A}$ has the subsequent capabilities:
2. Public/open channel communication fully controlled by the TA.
3. $\mathcal{A}$ can detain, retransmit, and modify the old message. $\mathcal{A}$ can also cease or transmit a forged message.
4. $\mathcal{A}$ can get the data saved in a smart card through power analysis [17,22].
5. Any insider/privileged user or outsider can attempt to violate the privacy and security of the system.
6. The private key of the TA cannot be compromised.

4.3. Entities Involved in the Proposed Protocol

The network model is made up of four entities: smart device (SD) in the house, gateway node (GWN), end user (U), and trusted authority (TA). The smart devices are typically diverse and resource-constrained, and they may be utilized to allow a wide range of use cases including lighting control, surveillance systems, temperature management, and so on. The master node as a GWN installed in the house acts as a link between the user and smart gadgets. The users may remotely operate the various smart home gadgets based on their requirements. Others totally trust the TA’s processing and communication capabilities.

4.4. Proposed Protocol Processes

The proposed protocol is based on six phases as presented in the following subsections:

4.4.1. Initialization Process

In the initialization phase, the Trusted Authority (TA) selects a private key $K$.

4.4.2. Smart Device Enrollment Process

$SD$ picks an identity $SID_d$ and forwards to the TA. TA computes $S_{GS} = h(SID_d||K)$ and store $SID_d$ in its own database describe in Table 2.

Table 2. Proposed smart device enrollment process.

| Smart Device (SD) | Trusted Authority (TA) |
|-------------------|------------------------|
| Selects an identity $SID_d$ | $MSG_1 = (SID_d)$ |
| $TA$ computes $S_{GS} = h(SID_d||K)$ | $MSG_2 = (S_{GS})$ |
| Store $SID_d$, $S_{GS}$ into memory. | |

4.4.3. Gateway Node Enrollment Process

Given blow Table 3 GWN selects an identity $GID_w$ and sends it to TA. TA computes $N_w = h(GID_w||K)$ and stores $GID_w$ in its own database and sends $N_w$ to the GWN, which it stores in its own memory.
Table 3. Proposed gateway node enrollment process.

| Gateway Node (GWN)          | Trusted Authority (TA) |
|----------------------------|------------------------|
| Selects an identity $GID_w$| $MSG_1 = (GID_w)$      |
| $TA$ computes              | $MSG_2 = (N_w)$        |
| $N_w = h(GID_w||K)$         | Stores $GID_w$ in own database. |
| $MSG_2 = (N_w)$            | Store $N_w$ into memory. |

4.4.4. User Enrollment Process

The subsequent procedure is adopted to register a user with the system in Table 4:

Step 1. $U$ picks $ID_u$ and $PW_u$ and computes $HID_u = h(ID_u)$ and sends $HID_u$ to the $TA$.

Step 2. $TA$ generates $R_{1,T}$, $PID_u \in Z_p$ and calculates $A_u = h(HID_u||R_{1,T}||K)$. Save $\{PID_u, \{A_u\}_{E_1}\}$ tuple in own database. Send $A_u, PID_u$ to $U$ through private channel.

Step 3. On getting $MSG_2$, $U$ computes $(\sigma_u, \tau_u) = Gen(Bio_u)$, $A^*_u = A_u \oplus \sigma_u \oplus PW_u$, $V_u = h(ID_u||PW_u||\sigma_u) \oplus \sigma_u, PID^*_u = PID_u \oplus \sigma_u$. SC replaces $A_u, PID_u$ with $A^*_u, V_u, h(\cdot), Gen(\cdot), Rep(\cdot), \tau_u, t$ in SC.

Table 4. Proposed user enrollment process.

| User ($U$) | Trusted Authority (TA) |
|------------|------------------------|
| choose $ID_u$ and $PW_u$. $HID_u = h(ID_u)$ | $MSG_1 = (HID_u)$ |
| $MSG_1 = (HID_u)$ | $MSG_2 = (A_u, PID_u)$ |
| $N_u = h(ID_u||K)$ | $TA$ computes $N_u$ in own database. |
| $MSG_2 = (N_u)$ | Store $N_u$ into memory. |

4.4.5. Login and Authentication Process

The login and authentication phase describe in Table 5 includes the following steps:

Step 1. Insert SC and input $ID_u, PW_u, Bio_u$. Calculate $\sigma_u = Rep(Bio_u, \tau_u)$ and check whether $V_u \oplus \sigma_u = h(ID_u||PW_u||\sigma_u)$; if true, further compute $A_u = A^*_u \oplus \sigma_u \oplus PW_u$. SC chooses $R_{\delta,T} \in Z_p$ and $T_u$ and computes $A^{**}_u = h(A_u||T_u)$, $PID_u = PID^*_u \oplus \sigma_u$. $SID^*_u = SID_u \oplus A_u \oplus R_u$, $R^*_u = R_u \oplus A_u$. Sends $MSG_1 = (R_u, SID^*_u, PID_u, A^{**}_u, T_u)$ to $TA$ by public channel.

Step 2. First of all, $TA$ verifies the timeliness of the timestamp by inspecting the condition $|T_u - T_c \leq \delta T|$, if so, searches $PID_u$ in database, if it exists, then fetches related $\{A_u\}_{E_1}$ and computes $A_u = \{A_u\}_{E_1}$ and checks if $A^{**}_u = h(A_u||T_u)$, if true, then selects $PID^{new}_u \in Z_p$ and $T_u$, replaces the $PID_u$ with $PID^{new}_u$, and computes $W_t = h(h(SID^*_u||K)||PID^{new}_u||T_u), V_t = h(N_u||GID_w||PID^{new}_u||T_u), R_u = R^*_u \oplus A_u, R^{**}_u = R_u \oplus W_t, X_t = W_t \oplus N_w$. 


Sends message $MSG_2 = (R_u^*, V_t, X_t, PID_u^{new}, SID_d, GID_w, T_{Ta})$ to GWN through public channel. Also sends $MSG_3 = (PID_u^{new})$ back to $U$ via open channel.

Step 3. GWN gets the $MSG_2$ from $TA$ and checks if $|T_{Ta} - T_c ≤ δT|$, if true, computes $V_t = h(N_w || GID_w || PID_u^{new} || T_{Ta})$, created timestamp $T_w$, and computed $W_t = X_t ⊕ N_w$, $V_w = h(SID_d || W_t || T_w)$. At the end, send message to the SD message $MSG_4 = (R_u^*, V_w, PID_u^{new}, SID_d, T_{Ta}, T_w)$ to SD via open/insecure channel.

Step 4. SD verifies the freshness of $MSG_4$ by examining the condition $|T_w - T_c ≤ δT|$. If true, it selects $R_d ∈ Z_p$ and $T_d$, checks if $V_w = h(SID_d || h(S_{GS} || PID_u^{new} || T_{Ta} || T_w)$, if true, then computes $R_d = R_u^* ⊕ h(S_{GS} || T_{Ta})$, $SK_{ud} = h(R_u || R_d || PID_u^{new} || SID_d || T_d)$, $R_d^* = R_u ⊕ R_d$, $V_d = h(SK_{ud} || T_d)$, $X_d = h(W_t || T_d)$. At the end, sends the message $MSG_5 = (V_d, X_d, R_d^*, T_d)$ to GWN by public/insecure channel.

Step 5. GWN examines the freshness of the message by inspecting the condition $|T_d - T_c ≤ δT|$, and if the condition is true, GWN selects timestamp $T_u$ and sends the $MSG_6 = (V_d, R_d^*, T_d, T_u)$ to $U$ via open channel.

Step 6. Upon getting the $MSG_6$, $U$ validates message freshness through condition $|T_w - T_c ≤ δT|$. If true, compute $PID_u^* = PID_u^* ⊕ PID_u$ and $PID_u^{new}$, replace $PID_u$ with $PID_u^*$, $R_d = R_u ⊕ R_d^*$, $SK_{ud} = h(R_u || R_d || PID_u^{new} || SID_d || T_d)$, $V_d = h(SK_{ud} || T_d)$, if true, the session key is saved for secure communication.

| Table 5. Proposed login and authentication process. |

| User (I) | Trusted Authority (TA) | Gateway (GWN) | Smart Device (SD) |
|----------|------------------------|---------------|-------------------|
| Insert SC, Input ID_u, PW_u, Bio_u, Calculate $σ_u = Rbp(Bio_u, τ_u)$, $V_u = A_u ⊕ h(ID_u || PW_u || σ_u)$, $A_u = A_u ⊕ σ_u ⊕ PW_u$, Choose $R_u ∈ Z_p^*$ and $T_u$, $A_u^* = h(A_u || T_u)$, $PID_u = PID_u^* ⊕ σ_u$, $SID_u = SID_u^* ⊕ σ_u ⊕ R_u$, $R_u^* = R_u ⊕ A_u$. |
| Check if $|T_u - T_c ≤ δT|$ | Search $PID_u$ in database. If exists, then fetch related $\{A_u\}_D_u$, $A_u = \{A_u\}_D_u$ | If $A_u^* = h(A_u || T_u)$ true, continue. | Select $PID_u^{new} ∈ Z_p$ and $T_u$, Replace $PID_u$ with $PID_u^{new}$ and compute $W_t = h(h(SID_d || K) || PID_u^{new} || T_{Ta})$, $V_t = h(N_w || GID_u || PID_u^{new} || T_{Ta})$, $R_u = R_u^* ⊕ A_u$, $R_u^* = R_u ⊕ W_t$, $X_t = W_t ⊕ N_w$. |

$MSG_1 = (R_u^*, SID_d, PID_u, A_u^*, T_u)$

$MSG_2 = (R_u^*, V_t, X_t, PID_u^{new}, SID_d, GID_w, T_{Ta})$

$MSG_3 = (PID_u^{new})$

$MSG_4 = (R_u^*, V_w, PID_u^{new}, SID_d, T_{Ta}, T_w)$

$MSG_5 = (V_d, X_d, R_d^*, T_d)$

$MSG_6 = (V_d, R_d^*, T_d, T_u)$

$MSG_7 = (PID_u^{new})$
Table 5. Cont.

| User (UI) | Trusted Authority (TA) | Gateway (GWN) | Smart Device (SD) |
|-----------|------------------------|---------------|-------------------|
| Check if $|T_u - T_i| \leq \delta T$ | \( V_i = h(N_u||GID_u||PID_u^{old}||T_u) \) | \( W_i = X_i \oplus N_u \) | \( V_u = h(SID_u||W_i||T_u) \) |
| \( MSG_u = (K_u^{new}, V_u^{new}, PID_u^{new}, SID_u, T_u, T_i) \) | \( MSC_u = (K_u^{new}, V_u^{new}, SID_u, T_u, T_i) \) | \( W_u = h(SID_u||W_i||T_u) \) | \( V_d = h(SK_{ud}||T_d) \) |
| Check if $|T_d - T_i| \leq \delta T$, | \( X_d = h(W_i||T_d) \) if true. | Select timestamp $T_{u,v}^{new}$ | \( MSC_u = (V_{u,v}^{new}, T_{u,v}, T_i) \) |
| \( PIV^{new} = PIV^{old} \oplus PID_u \oplus PID_{new} \) | \( R_u = R_u \oplus R_{d} \) | \( SK_{ud} = h(R_u||R_d||PID_u^{new}||SID_u||T_d) \) | \( V_d = h(SK_{ud}||T_d) \) |
| Replace $PID_u^{old}$ with $PID_u^{new}$ | \( S_{ud} = h(R_u||R_d)\) | \( V_{ud} = h(SK_{ud}||T_d) \) | \( X_d = h(W_i||T_d) \) |

UI and SD both Save the session – key $SK_{ud} = SK_{ud}$

4.4.6. Biometric and Password Update Process

A registered user can update a biometric/password. To do so, the user needs to login first as described in the “Login and authentication process”. After a successful login, the user will adopt the subsequent procedure to update his/her biometric/password:

Step 1. User will be prompted to enter a new password $PW_{u}^{new}$ biometric $Bio_{u}^{new}$.

Step 2. SC will compute \((\sigma_u^{new}, \tau_u^{new}) = Gen(Bio_{u}^{new}), A_u^{new} = (A_u \oplus \sigma_u^{old} \oplus PW_{u}^{old}) \oplus (\sigma_u^{new} \oplus PW_{u}^{new}) = A_u \oplus \sigma_u^{new} \oplus PW_{u}^{new}, V_u^{new} = V_u \oplus h(ID_u||PW_{u}^{new}||\sigma_u^{new} \oplus PW_{u}^{new}||\sigma_u^{new} \oplus PW_{u}^{new}) = V_u \oplus h(ID_u||PW_{u}^{new}||\sigma_u^{new} \oplus PW_{u}^{new}||\sigma_u^{new} \oplus PW_{u}^{new})\) \( PIV_{u}^{new} = PIV_u^{old} \oplus \sigma_u^{new}\).

Step 3. Finally, SC will replace $A_u^{new}, V_u^{new}$ with $A_u^{new}, V_u^{new}, PIV_{u}^{new}, \tau_u^{new}$.

5. Security Analysis

In the subsequent subsection, formal and informal security analyses for our proposed protocol are performed.

5.1. Informal Security Analysis

The introduced protocol is secure against well-known attacks as presented in the successive subsections:

5.1.1. Replay Attack

In the introduced protocol, timestamps $T_u, T_{i,u}, T_{i,d}, T_d$ and random numbers $R_u, R_d$ are employed to ensure that the introduced protocol is secure from replay attack. Hence, the replay message will be detected if a timestamp is expired, or a random/arbitrary number is inconsistent. So, a replay attack cannot be launched upon the introduced scheme.
5.1.2. Session Key Freshness Property

In the introduced protocol, a session key is constructed by some distinct identities, random numbers, and timestamps, and for each session, random numbers $R_u, R_d$ are unique. Hence, a novel key for every session ensures the key’s freshness.

5.1.3. User Anonymity and Untraceability

In the introduced protocol, the user identity is not shared even with the TA. Additionally, if an attacker tries to capture messages $MSG_1, MSG_2, MSG_3, MSG_4, MSG_5, MSG_6$, none of these messages utter any information about the user. Furthermore, the attacker will not be able to extract from the smart card, because he/she will need a password and a biometric to extract the password, and it is also wrapped by hash function. In addition, in each session, all the values are unique as they are composed of random numbers, and consequently, they are novel for each authentication session. Hence, an adversary will not be able to recognize any specific user or the location of any specific user. Hence, the introduced protocol ensures the anonymity and untraceability of valid user.

5.1.4. Smart Card Stolen Attack

Assume an attacker steals an authorized user’s smart card or it is mishandled and discovered by an attacker, allowing him/her to get any sensitive data from the SC. An attacker can extract information $A_u^*, V_u, PID^*_u, \tau_u, t$ from the SC. But to attain any confidential information from the SC, an attacker needs the information of $ID_u, PW_u$, and $e_u$, and these values are unknown to $A$. Hence, in the introduced protocol, a stolen SC attack is not conceivable.

5.1.5. Impersonation Attack

If an attacker tries to imitate a $U, GWN, or SD$ towards any other legal participant, to do so, a valid request messages is required. To impersonate as a $U$, $A$ tries to create a login request message $MSG_1 = R_u^*, SID^*_u, PID_u, A_u^*, T_u$. However, the attacker cannot generate a true login message because it necessitates the information of $R_u, A_u$. Therefore, the introduced protocol can resist an impersonation attack.

5.1.6. Man-in-the-Middle Attack

Assume an attacker seizes the message and tries to modify a login request or any other message transferring over the public channel. To do this, the attacker requires the knowledge of secret parameters as described in Section 5.1.5. Therefore, the introduced scheme can withstand this attack.

5.1.7. Perfect Forward Secrecy

The shared session key in the introduced protocol includes random nonces included by both $U$ and $SD$. So, if the private key $K$ is in the knowledge of the adversary, he/she still cannot produce previously shared session keys. Hence, the introduced protocol has the feature of perfect forward secrecy.

5.2. AVISPA Tool Based Automated Formal Security Analysis

AVISPA, a tool relying on the Dolev–Yao threat model, was introduced by Armando et al. [38]. In this situation, the attacker can edit, transmit, and alter messages. This tool is well-accepted for assessing various security protocols. AVISPA uses a “high level protocol specification language (HLPSL)” to describe and design security protocols. The HLPSL requirements are broken down into roles. During the execution of a protocol, several roles are utilized to express the activities of a single agent. Each agent performs a specific role throughout the execution of a protocol. HLPSL’s main objective is to verify security properties including message, agent secrecy and authentication. The security protocol is considered to determine its level of security considering the predetermined objectives.
5.2.1. AVISPA Model Checkers

1. **On-the-fly model checker (OFMC):** OFMC uses lazy data types to develop an efficient on-the-fly model for security protocols with limitless state spaces.

2. **Constraint-logic-based attack searcher (CL-AtSe):** The input of (CL-AtSe) is a protocol stated as a set of restrictions that help identify security protocol assaults in the form of a collection of rewriting rules (IF format).

3. **SAT-based model checker (SATMC):** Depending on the transitional state of the IF specification, creates a propositional formula. According to the propositional formula, every violation of security that might result in an attack is considered.

4. **“Tree automata-based on automatic approximations for the analysis of security protocols” (TA4SP) model checker:** By accurately estimating the attacker’s capabilities, it exposes the protocol’s weakness and predicts its accuracy.

5.2.2. AVISPA Simulation Steps

Automated validation of the protocol was carried out to demonstrate that the introduced protocol could resist reply and man-in-the-middle attacks. The subsequent steps were executed to simulate our protocol in AVISPA:

1. Firstly, the scheme was implemented through High-Level Protocols Specification Language (HLPSL) [33], next, the HLPSL2IF translator was employed to interpret HLPSL into Intermediate Format (IF).

2. Additionally, to declare that whether protocol was safe or not, the interpreted IF was provided in Output Format (OF). The HLPSL role specification for the user, smart device, trusted authority and gateway node are depicted in Figures 3–6, respectively.

5.2.3. Simulation Details

Setting simulated security goals is the first step in building the HLPSL script for our scheme. Our major goal was to keep certain values, such as \{SK_{\text{util}}, K, R_u, R_d, R_t\}, secret. Furthermore, we defined the six roles, which are: (1) the trusted authority’s role, (2) the user’s role, (3) the gateway node’s role, (4) the smart device’s role, (5) the role session that combines the basic roles (role_TA, role_USERS, role_GWN, and role_SD), (6) the role environment that incorporates numerous sessions and consists of functions and global variables and outlines the security goals of the protocol.
In Figure 3, the requirement for the role_User that is executed by the user is illustrated. In this role, the $U_u$ knows every agent ($GWN$, $SD$, $TA$) and the symmetric-key $SK_{uita}$ that is joint among the $U_u$, $TA$, the hash function $H(.)$, and the Dolev–Yao dy model-based send/receive channels ($SND, RCV$). To start the protocol, $U_u$ receives a start message ($RCV(start)$) as an indication at the first state (state 0). To register, $U_u$ creates the identity $ID_u$ and retransmits $HID_u'$ after being encrypted with $SK_{uita}$ to the $TA$. $U_u$ gets $H(H(HID_u||R'_t||K))$ and $PID'_d$ encrypted with $SK_{uita}$ coming from the $TA$, and the registration process is completed. Next, $U_u$ generates some fresh nonces $\{R'_u,T'_u\}$ and computes $\{A'_u, A'_u2, SID'_d\}$. Next, $U_u$ transmits $\{R'_u, SID'_d, PID'_d, A'_u2, T'_u\}$ to $TA$ and receives a message from the $TA$ containing $PID'_{d_{new}}$. $U_u$ receives the message $\{H(R_u,R'_d,PID'_{d_{new}},SID'_d,T'_d).R'_u.xor(R_u,R'_d).PID'_{d_{new}}.T'_d.T'_w1\}$ coming from the $SD$ at the next transition. Figure 4 demonstrates the specification for the role_SD which is played by the $SD$. In this role, the $SD$ recognizes all the agents ($U_i$, $TA$, $GWN$), the
symmetric-key $S_{GS}$ which is shared among the $SD$ and the $TA$, the hash function $H(.)$, and the Dolev–Yao-model-based send/receive channels ($SND$, $RSV$).

![Figure 4. Role specification for a smart device.](image)

Figure 4 explains the requirement for role $TA$, which is executed by the $TA$. The $TA$ knows every single agent ($U_u$, $SD$, $GWN$), the symmetric-key $SK_{uita}$, $S_{GS}$, and $N_{iu}$, which are common among the $U_u$ and the $TA$, $SD$, and $TA$, and among $GWN$ and $TA$, respectively. Furthermore, $TA$ has the knowledge of the $H(.)$ and Dolev–Yao-model-based send/receive channels ($SND$, $RSV$). The remaining part of the requirement defines the various states of the protocol implemented by the $TA$. Figure 6 displays the specification for role $GWN$ that is played by the $GWN$. $GWN$ identifies all the agents ($U_u$, $SD$, $TA$), the symmetric-key $N_{iu}$ that is shared amongst the $GWN$ and the $TA$, the $H(.)$, and Dolev–Yao-model-based send/receive channels ($SND$, $RSV$).
Figure 5. Role specification for TA.
Session, environment, and goal roles’ specifications are depicted in Figure 7. In the session role, all roles (role_TA, role_USERS, role_SD, role_GWN) are combined. More than one session is started from the environment role. The constants (ta, users, gwn, sd) represents the agents (TA, USERS, GWN, SD), respectively. The symmetric key shared among the TA and U_u is presented by SK_{uita}. The hash function H is presented by the constant h. The applicable parameters that are supposed to be known by the intruder are defined in the intruder knowledge part. We undertake that the intruder has the knowledge of all the agents (USERS, TA, GWN, SD). The simulation objectives are declared with the help of goal keywords: We are concerned about verifying the secrecy of the subsequent parameters: R_u, R_t, R_d, K.
5.2.4. Simulation Result

Simulation outcomes depicted in Figures 8 and 9 revealed that the introduced protocol was according to the design specifications and could hold out against replay and man-in-the-middle attacks. Based on the AVISPA backend model checker, OFMC and CL-AtSe simulation results are presented. With depth 8 piles in 0.72 s, 198 nodes were scrutinized in the OFMC backend. The CL-AtSe backend inspected 0 states, the computation and interpretation consumed for the backend were 0.08 s and 0.00 s, respectively.

Figure 7. Role session, environment, and goal.

Figure 8. OFMC backend analysis.
6. Comparative Analysis

This section provides the comparison of our proposed protocol with the existing protocols, which includes those of Banerjee et al. [23], Shuai et al. [34], Yu and Li [39], Naoui et al. [40], Fakroon et al. [41], and Dey and Hossain [29].

6.1. Functionality Comparison

The functionality analysis was performed with respect to the existing and proposed protocols as presented Table 6. As per comparison, the proposed protocol rendered superior security along with enhanced security features in contrast to those of [23] and other compared protocols. Here, $F_n$ is the $n^{th}$ compared feature/security requirement, ✓ shows that the specified parameter exists in this study, and the protocol can resist an attack. Moreover, × presents whether the protocol is not able to resist an attack or lacks a specific feature, whereas the − sign shows that a certain security/feature requirement is not applicable.

6.2. Comparison of Communication Overhead

In Table 7, the communication expenses estimate is presented. In order to compare, the sizes of various parameters taken were as follows: identities were considered as 128 bits, a timestamp was 32 bits, random nonces were 128 bits, a hash digest was 160 bits (if SHA-1 is employed [42]), cost for ECC point was $(160 + 160) = 320$ bits, and size of symmetric enc/decryption block was 128 bits [43,44], respectively.
Table 6. Comparison of functionality features.

|       | [23] | [29] | [34] | [39] | [40] | [41] | Our   |
|-------|------|------|------|------|------|------|-------|
| Fua   | ✓    | ×    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fsna  | ✓    | ×    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fu    | ✓    | ×    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fpara | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fsas  | ✓    | ×    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fpar  | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fopga | ✓    | −    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fsia  | ✓    | −    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fepd  | ×    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Frpca | ✓    | −    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fjsa  | ×    | ×    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fsia  | ×    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Efpa  | ×    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fsva  | ×    | ×    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fgwn  | ×    | ×    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fsdi  | ✓    | ×    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fsdi  | ✓    | ×    | ×    | ✓    | ✓    | ✓    | ✓     |
| Fgwna | ×    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fgwna | ×    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |
| Fenv  | ✓    | ✓    | ✓    | ✓    | ✓    | ✓    | ✓     |

Note: Fua: user anonymity; Fsna: sensor node anonymity; Fu: untraceability; Fpara: protection against reply attack; Fsas: secure against stolen smart-card attack; Fra: resilience against ESL attack under the CK-adversary model; Fopga: offline password guessing attack; Fsia: smart-card impersonation attack; Fepd: efficient protocol design; Frpca: resist password change attack; F3fa: three-factor authentication; Fpska: parallel session key attack; Fsva: stolen GvVN attack; Fspa: stolen verifier attack; Fuia: user impersonation attack; Fgwn: GWN impersonation attack towards smart device; Fsdi: smart device impersonation; Fgwnia: GWN impersonation attack towards user; Ffasv: formal automated security verification.

Table 7. Communication cost comparison.

| Protocol | # of Messages | # of Bytes |
|----------|---------------|------------|
| [23]     | 4             | (68 + 40 + 56 + 72) = 236 |
| [29]     | 5             | (132 + 132 + 52 + 52 + 52) = 420 |
| [34]     | 4             | (132 + 64 + 40 + 68) = 304 |
| [39]     | 8             | (84 + 124 + 164 + 164) × 2 = 1072 |
| [40]     | 8             | (104 + 52 + 56) = 212 |
| [41]     | 4             | (100 + 52 + 56 + 84) = 288 |
| Our      | 6             | (76 + 112 + 16 + 72 + 60 + 44) = 380 |

The communication cost of $M_{sg1} = \langle R_u^*, SID_d, PID_u, A_u^*, T_u \rangle$ was $\langle 128 + 128 + 128 + 160 + 32 \rangle = 608$ bits, $M_{sg2} = \langle R_u^*, V_d, X_t, PID_u^{new}, SID_d, GID_w, T_{ia} \rangle$ was $\langle 160 + 160 + 160 + 128 + 128 + 128 + 32 \rangle = 896$ bits, $M_{sg3} = \langle PID_u^{new} \rangle$ was 832 bits, $M_{sg4} = \langle R_u^*, V_d, PID_u^{new}, SID_d, T_{via}, T_{iw} \rangle$ was $\langle 128 + 160 + 128 + 128 + 32 + 32 \rangle = 576$ bits, $M_{sg5} = \langle V_d, X_d, R_d^*, T_d \rangle$ is $\langle 160 + 160 + 128 + 32 \rangle = 480$ bits, and cost for $M_{sg6} = \langle V_d, R_d^*, T_d, T_{w}^* \rangle$ was $\langle 160 + 128 + 32 + 32 \rangle = 352$ bits. Summing all these, the total cost of communication of the introduced protocol through the login and authentication phase becomes 380 bytes. In addition, Table 7 shows that the communication cost of the introduced protocol was less than that of those in comparison [39,42] and slightly higher than that of [23,34,40,41], but this is justifiable as the introduced protocol provides better security than these other protocols do. Figure 10 also depicts the communication cost of our proposed protocol.
6.3. Computation Overhead Comparison

This section compares the computation cost of different protocols. Table 8 shows the approximate computation times expected for different cryptographic procedures and their notations. Based on a real-time environment, an experiment was conducted over a smartphone using MIRACL Library. The smartphone Redmi Note 8 presented as a mobile/user device with the specification of Octa-core Max 2.01 GHz processor, 4 GB RAM, MIUI version 11.0.7, and the underlying Android version was 9. For GSS, over the Ubuntu 16.0 LTS operating system, an HP EliteBook 8460P with 4 GB RAM and an Intel Core i7-2620M 2.7 GHz processor was used. Additionally, to replicate the smart devices, a Pi3 B+ with Cortex-A53(ARMv8) 64-bit SoC @ 1.4 GHz processor and 1 GB LPDDR2 SDRAM RAM was used. The simulation results on each device are given in Table 8, which shows the approximate computation times expected for different cryptographic procedures in ms and their notations [45]. We assumed here that $T_{fe} \approx T_{ecm}$.

Table 8. Approximate computation time of various operations.

| Notation | Operation             | Mobile Device | Gateway/TA | Smart Device |
|----------|-----------------------|---------------|------------|--------------|
| $T_h$    | Hash function         | 0.009         | 0.004      | 0.006        |
| $T_{ecm}$| ECC multiplication    | 5.116         | 0.926      | 4.107        |
| $T_{sym}$| Symmetric enc/dec     | 0.017         | 0.008      | 0.013        |
| $T_{bp}$ | bilinear pairing      | 17.36         | 4.038      | 12.52        |

As represented in Table 9, the computation cost of the proposed protocol is a bit high as compared to those of [23,41]. However, it is justified when compared to the security provided by the introduced protocol, as it is shown in Table 6 that these protocols lack some security features. Figure 11 also depicts the computation cost of the introduced protocol.
Table 9. Computation cost comparison.

| Protocol | $U$ | $TA/RA$ | GWN | SD | Total Cost |
|----------|-----|---------|-----|----|------------|
| [25]     | $10T_h + 1T_{fe}$ | –      | $10T_h$ | $4T_h$ | $24T_h + 1T_{fe}$ |
|          | $5.206$ ms  | –      | $0.04$ ms | $0.024$ ms | $5.27$ ms |
| [34]     | $6T_h + 1T_{ecm}$ | –      | $7T_h + 1T_{ecm}$ | $3T_h$ | $16T_h + 3T_{ecm}$ |
|          | $5.17$ ms    | –      | $5.114$ ms | $0.018$ ms | $10.302$ ms |
| [39]     | $7T_h + 14T_{ecm}$ | –      | $12T_h + 19T_{ecm} + 4T_{bp}$ | $7T_h + 14T_{ecm}$ | $26T_h + 47T_{ecm} + 4T_{bp}$ |
|          | $71.687$ ms  | –      | $33.794$ ms | $57.54$ ms | $163.021$ ms |
| [40]     | $12T_h + 3T_{sym} + 2T_{ecm}$ | –      | $13T_h + 4T_{sym} + 2T_{ecm}$ | $1T_h + 1T_{sym}$ | $26T_h + 7T_{sym} + 4T_{ecm}$ |
|          | $10.391$ ms  | –      | $1.936$ ms | $0.019$ ms | $12.346$ ms |
| [41]     | $4T_h$ | –      | $5T_h$ | $24T_h$ | $33T_h$ |
|          | $0.036$ ms  | –      | $0.02$ ms | $0.144$ ms | $0.2$ ms |
| [29]     | $4T_h + 2T_{ecm} + 3T_{sym}$ | –      | – | $3T_h + 2T_{ecm} + 3T_{sym}$ | $7T_h + 4T_{ecm} + 6T_{sym}$ |
|          | $10.319$ ms  | –      | – | $8.271$ ms | $18.59$ ms |
| Our      | $1T_{fe} + 4T_h$ | $1T_{sym} + 4T_h$ | $3T_h$ | $6T_h$ | $1T_{fe} + 15T_h + 1T_{sym}$ |
|          | $5.152$ ms  | $0.024$ ms | $0.012$ ms | $0.036$ ms | $5.224$ ms |

Figure 11. Computation cost comparison.

7. Conclusions

The Internet of Things (IoT) is turning into the mainstream with each passing day. Applications of the IoT are not limited to only military or industrial use; nowadays, the IoT is employed in homes, agriculture, and even in the medical field. IoT privacy and
security issues gradually increase. To subdue the privacy and security issues related to smart homes, various schemes have been introduced and claimed as robust and with anonymous authentication protocols for a smart-home environment. In this paper, we introduced a protocol to overcome the vulnerabilities found in a previous protocol. In the introduced protocol, no parameters are stored in the verifier table whose disclosure can lead to system compromise, a TA is also included to authenticate the users, and a broadcasting issue is also mitigated to improve resource utilization. Security and comparative analyses of the introduced protocol were performed to show that the introduced protocol was secure and provided better security features with low communication and computation costs. Authentication schemes for a smart-home environment should be lightweight and secure. Many schemes proposed earlier are lightweight but not secure because in achieving lightweight features they neglect security. Similarly, some schemes are more secure but not lightweight, which is undesirable for a smart-home environment. Hence, there is a need to work on both factors at the same time.

Author Contributions: R.M.A.H.-u.-r., conceptualization, methodology, M.L. review; A.H.M.A., supervision; A.A.A., computation cost; M.K.H., formal security analysis; Z.A., draft formatting and draft preparation; R.L.A., literature review. All authors have read and agreed to the published version of the manuscript.

Funding: Under grant FRGS/1/2019/ICT03/UKM/02/1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported by the Faculty of Information Science and Technology, University Kebangsaan Malaysia, Bangi, Selangor, 43600, Malaysia under grant FRGS/1/2019/ICT03/UKM/02/1. The authors, therefore, acknowledge with thanks for the technical and financial support.

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

References
1. Hong, A.; Nam, C.; Kim, S. What will be the possible barriers to consumers’ adoption of smart home services? *Telecommun. Policy* 2020, 44, 101867. [CrossRef]
2. Aman, A.H.M.; Hassan, W.H.; Sameen, S.; Attarbaship, Z.S.; Alizadeh, M.; Latiff, L.A. Iomt amid COVID-19 pandemic: Application, architecture, technology, and security. *J. Netw. Comput. Appl.* 2020, 174, 102886. [CrossRef] [PubMed]
3. Aman, A.H.M.; Yadegaridehkordi, E.; Attarbaship, Z.S.; Hassan, R.; Park, Y.-J. A Survey on Trend and Classification of Internet of Things Reviews. *IEEE Access* 2020, 8, 111763–111782. [CrossRef]
4. Hassan, R.; Qamar, F.; Hasan, M.K.; Aman, A.H.M.; Ahmed, A.S. Internet of Things and Its Applications: A Comprehensive Survey. *Symmetry* 2020, 12, 1674. [CrossRef]
5. Sadeq, A.S.; Hassan, R.; Aman, A.H.M.; Sallehudin, H.; Allehaibi, K.; Albogami, N.; Prabuwono, A.S. MAC protocol with grouping awareness GMAC for large scale Internet-of-Things network. *Peerj Comput. Sci.* 2021, 7, e733. [CrossRef]
6. Aman, A.H.M.; Shaari, N.; Ibrahim, R. Internet of things energy system: Smart applications, technology advancement, and open issues. *Int. J. Energy Res.* 2021, 45, 8389–8419. [CrossRef]
7. Bringhenti, D.; Valenza, F.; Basile, C. Toward Cybersecurity Personalization in Smart Homes. *IEEE Secur. Priv.* 2022, 20, 45–53. [CrossRef]
8. Reig, S.; Fong, T.; Forlizzi, J.; Steinfeld, A. Theory and Design Considerations for the User Experience of Smart Environments. *IEEE Trans. Hum.-Mach. Syst.* 2022, 52, 522–535. [CrossRef]
9. Pathy, B.; Sujatha, R. IoT Enabled Smart Connected Homes and Its Social Graces. In *Internet of Things and Its Applications*; Springer: Cham, Switzerland, 2022; pp. 477–496. [CrossRef]
10. Zhang, S.; Rong, J.; Wang, B. A privacy protection scheme of smart meter for decentralized smart home environment based on consortium blockchain. *Int. J. Electr. Power Energy Syst.* 2020, 121, 106140. [CrossRef]
11. Aubel, P.V.; Poll, E. Smart metering in the networks: What, how, and why. *Int. J. Electr. Power Energy Syst.* 2019, 109, 719–725. [CrossRef]
12. Jin, M.; Jia, R.; Spanos, C.J. Virtual Occupancy Sensing: Using Smart Meters to Indicate Your Presence. *IEEE Trans. Mob. Comput.* 2017, 16, 3264–3277. [CrossRef]
13. Jin, M.; Bekiaris-Liberis, N.; Weekly, K.; Spanos, C.J.; Bayen, A.M. Occupancy detection via environmental sensing. *IEEE Trans. Autom. Sci. Eng.* 2018, 15, 443–455. [CrossRef]

14. Taccardi, B.; Rametta, P.; Carcagni, P.; Leo, M.; Distante, C.; Patrono, L. An innovative AAL system based on neural networks and IoT-aware technologies to improve the quality of life in elderly people. *Int. J. Intel. Syst. Technol. Appl.* 2019, 19, 589–617. [CrossRef]

15. Smys, S. A Survey on Internet of Things (IoT) based Smart Systems. *J. ISMAC* 2020, 2, 181–189. [CrossRef]

16. Oliveira, R.N.; Roth, V.; Henzen, A.F.; Simao, J.M.; Nobama, P.; Wille, E.C.G. Notification Oriented Paradigm Applied to Ambient Assisted Living Tool. *IEEE Lat. Am. Trans.* 2016, 16, 647–653. [CrossRef]

17. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* 1983, 29, 198–208. [CrossRef]

18. Eisenbarth, T.; Kasper, T.; Moradi, A.; Paar, C.; Salmasizadeh, M.; Shalmani, M.T.M. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In *Advances in Cryptology—CRYPTO 2008*; Lecture Notes in Computer Science; Wagner, D., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 203–220. [CrossRef]

19. Yang, W.-H.; Shieh, S.-P. Password authentication schemes with smart cards. *Comput. Secur.* 1999, 18, 727–733. [CrossRef]

20. Hölbl, M.; Welzer, T.; Brumen, B. An improved two-party identity-based authenticated key agreement protocol using pairings. *J. Comput. Syst. Sci.* 2012, 78, 142–150. [CrossRef]

21. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology—CRYPTO’99*; Wiener, M., Ed.; Springer: Berlin/Heidelberg, Germany, 1999; pp. 388–397.

22. Messerges, T.; Dabbish, E.; Sloan, R. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* 2002, 51, 541–552. [CrossRef]

23. 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]

24. Farayez, A.; Reaz, M.B.I.; Arsad, N. SPADE: Activity Prediction in Smart Homes Using Prefix Tree Based Context Generation. *IEEE Access* 2018, 7, 5492–5501. [CrossRef]

25. Shakeri, M.; Shayestegan, M.; Abunima, H.; Reza, S.S.; Akhtaruzzaman, M.; Alamoud, A.; Sopian, K.; Amin, N. An intelligent system architecture in home energy management systems (HEMS) for efficient demand response in smart grid. *Energy Build.* 2017, 138, 154–164. [CrossRef]

26. Sciancalepore, S.; Piro, G.; Boggia, G.; Bianchi, G. Public Key Authentication and Key Agreement in IoT Devices With Minimal Airtime Consumption. *IEEE Embed. Syst. Lett.* 2016, 9, 1–4. [CrossRef]

27. Hasan, M.K.; Shafiq, M.; Islam, S.; Pandey, B.; El-Ebiary, Y.A.B.; Nafi, N.S.; Rodriguez, R.C.; Vargas, D.E. Lightweight Cryptographic Algorithms for Guessing Attack Protection in Complex Internet of Things Applications. *Complexity* 2021, 2021, 5540296. [CrossRef]

28. Alshahrani, M.; Traore, I. Secure mutual authentication and cumulative key-based access control for IoT smart home using cumulative Keyed-hash chain. *J. Inf. Secur. Appl.* 2019, 45, 156–175. [CrossRef]

29. Dey, S.; Hossain, A. Session-Key Establishment and Authentication in a Smart Home Networking Using Public Key Cryptography. *IEEE Sens. Lett.* 2019, 3, 7500204. [CrossRef]

30. Kumar, P.; Gurtov, A.; Janati, J.; Yianitli, M.; Sain, M. Lightweight and Secure Session-Key Establishment Scheme in Smart Home Environments. *IEEE Sens. J.* 2015, 16, 254–264. [CrossRef]

31. Kumar, P.; Braeken, A.; Gurtov, A.; Janati, J.; Ha, P.H. Anonymous Secure Framework in Connected Smart Home Environments. *IEEE Trans. Inf. Forensics Secur.* 2017, 12, 968–979. [CrossRef]

32. Gope, P.; Siddar, B. Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices. *IEEE Internet Things J.* 2018, 6, 580–589. [CrossRef]

33. 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]

34. 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]

35. Lee, Y.S.; Lee, H.J.; Alasaarela, E. Mutual authentication in wireless body sensor networks (WBSN) based on Physical Unclonable Function (PUF). In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 1314–1318. [CrossRef]

36. Khernane, N.; Potop-Butucaru, M.; Chaudet, C. BANZKP: A secure authentication scheme using zero knowledge proof for WBANs. In Proceedings of the 2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Brasilia, Brazil, 10–13 October 2016.

37. Koya, A.M.; Deepthi, P.P. Anonymous hybrid mutual authentication and key agreement scheme for wireless body area network. *Comput. Netw.* 2018, 140, 138–151. [CrossRef]

38. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.; Drielsma, P.H.; Heám, P.C.; Kouchkarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285. [CrossRef]

39. Yu, B.; Li, H. Anonymous authentication key agreement scheme with pairing-based cryptography for home-based multi-sensor Internet of Things. *Int. J. Distrib. Sens. Netw.* 2019, 15, 1550147719879379. [CrossRef]

40. Naoui, S.; Elhdhili, M.H.; Saidane, L.A. Novel smart home authentication protocol LRP-SHAP. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 15–18 April 2019.
41. Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things* 2020, 9, 100158. [CrossRef]
42. Eastlake, D., 3rd; Jones, P. US secure hash algorithm 1 (SHA1). *RFC* 2001, 3174, 1–22. [CrossRef]
43. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Netw.* 2014, 20, 96–112. [CrossRef]
44. Alotaibi, M. An Enhanced Symmetric Cryptosystem and Biometric-Based Anonymous User Authentication and Session Key Establishment Scheme for WSN. *IEEE Access* 2018, 6, 70072–70087. [CrossRef]
45. Hussain, S.; Chaudhry, S.A.; Alomari, O.A.; Alsharif, M.H.; Khan, M.K.; Kumar, N. Amassing the Security: An ECC-Based Authentication Scheme for Internet of Drones. *IEEE Syst. J.* 2021, 15, 4431–4438. [CrossRef]