mathematics

MDPI

Article

ESCI-AKA: Enabling Secure Communication in an IoT-Enabled Smart Home Environment Using Authenticated Key Agreement Framework

Hisham Alasmary [1,*] and Muhammad Tanveer [2]

[1] Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[2] Department of Computer Science, University of Management and Technology, Lahore 54770, Pakistan; muhammad_tanveer@umt.edu.pk
* Correspondence: alasmary@kku.edu.sa

Abstract: Smart home environments are a vital component of the larger ecosystem within smart cities, aiming to revolutionize residential living through the integration of Internet of Things (IoT) devices and advanced technologies. However, ensuring robust security and preserving privacy in these interconnected ecosystems present significant challenges. During the monitoring and controlling tasks in the smart home environment, diverse commands are exchanged between the IoT device and the user over the public Internet. The public Internet is open and vulnerable to various security attacks, which can corrode the monitoring and controlling operation of the smart home. In addition, conventional security algorithms are inappropriate for IoT devices deployed in the smart home. However, various pernicious security attacks are equally efficacious in the resource-limited smart home environment. Thus, various authenticated encryption schemes are proposed to enable security services in resource-constricted smart home environments. This paper presents a lightweight and efficient authentication framework for a smart home environment by leveraging the features of an authenticated encryption scheme and the hash function called "ESCI-AKA". ESCI-AKA checks the authenticity of the user at the local device and exchanges three messages among the user, gateway, and smart embedded device for establishing a secure channel for indecipherable communication by setting a session key. In addition, we corroborate the security of the established session key through the random oracle model and informal security analysis. Moreover, the Scyther tool is employed for the security validation of ESCI-AKA. Finally, the performance comparison of ESCI-AKA and other eminent security frameworks explicates that ESCI-AKA requires low computational and communication costs while providing robust security features.

Keywords: smart home; smart city; Internet of things; authentication; smart device

MSC: 94A60; 68P25

Citation: Alasmary, H.; Tanveer, M. ESCI-AKA: Enabling Secure Communication in an IoT-Enabled Smart Home Environment Using Authenticated Key Agreement Framework. *Mathematics* **2023**, *11*, 3450. https://doi.org/10.3390/math11163450

Received: 20 July 2023
Revised: 4 August 2023
Accepted: 7 August 2023
Published: 9 August 2023

1. Introduction

The Internet of Things (IoT) has emerged as a crucial enabler in the advancement of smart cities. These cities strive to leverage technological innovations to improve urban living standards, promote sustainability, and enhance overall efficiency. IoT devices and sensors deployed throughout a city enable the collection, analysis, and utilization of vast amounts of data to improve various aspects of urban life [1,2]. In a smart home, IoT devices such as thermostats, lighting systems, security cameras, door locks, appliances, and entertainment systems are interconnected and can be controlled remotely through a central hub or a mobile application. These devices can communicate with each other, share data, and respond to user commands or environmental conditions, creating an interconnected and intelligent living environment [3].

The implementation of smart home technologies necessitates significant attention to security and privacy concerns, while smart homes endeavor to improve urban living through the employment of state-of-the-art technologies, they also introduce potential risks related to data security and privacy breaches [4]. Smart homes yield extensive amounts of data via sensors, cameras, and other connected IoT devices. To safeguard against unauthorized access, data breaches, and the misuse of sensitive information, it is essential to employ robust data security measures. This includes implementing strong encryption, authentication mechanisms, and access controls.

In Figure 1, we can observe a smart home setup, enabling remote communication between the user and resource-constrained IoT devices present within the premises. The user has the capability to send diverse command and control instructions to accomplish various tasks within the smart home environment. However, it is essential to address potential security risks. The command and control information transmitted through a public communication channel is exposed to potential security vulnerabilities. Consequently, it becomes crucial to implement authenticated key agreement (AKA) mechanisms to safeguard the confidentiality of information [5]. By employing such mechanisms, the communicated command and control information can be protected, ensuring that only authorized individuals or entities can access sensitive data.

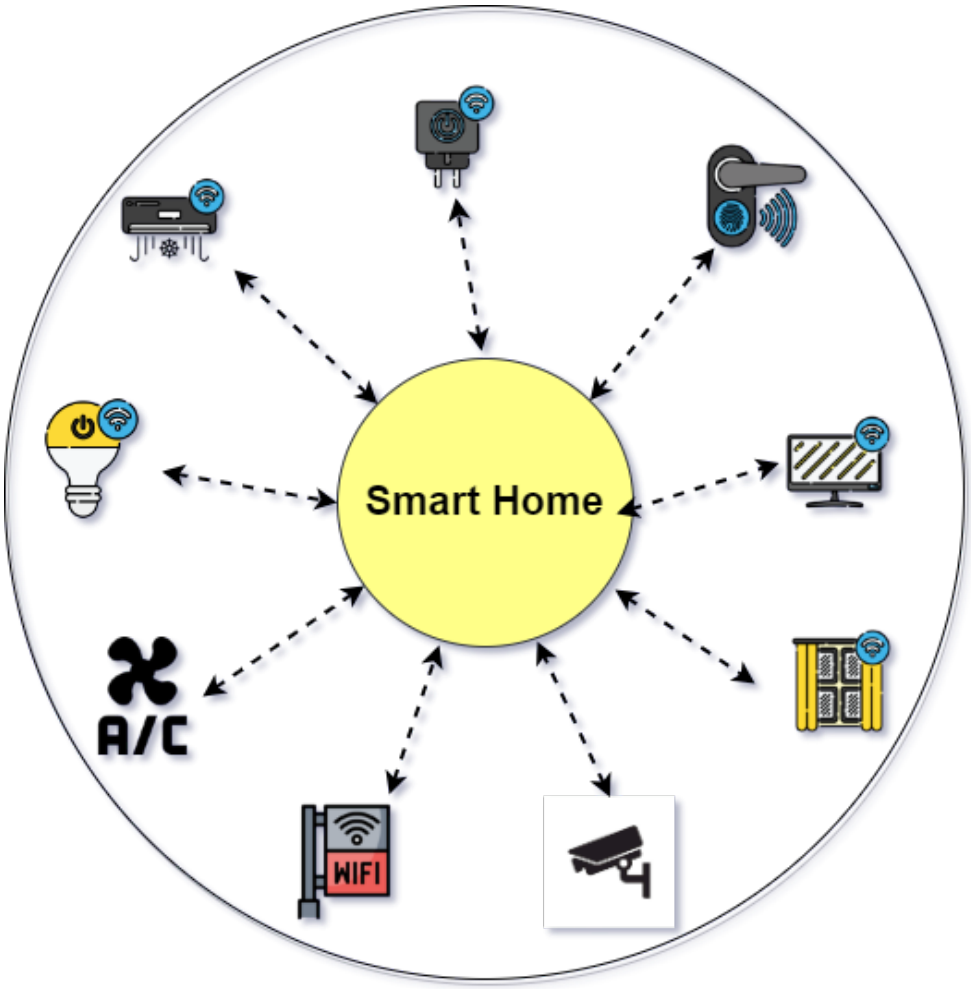

Figure 1. Smart home environment.

Several AKA frameworks have been proposed to establish secure and encrypted communication within the smart home environment. Despite the presence of various existing AKA frameworks, several significant security concerns remain unresolved. These include the prevention of impersonation, mitigation of denial of service (DoS) attacks,

protection against man-in-the-middle (MITM) attacks, and the need to ensure resource efficiency. Recently, various authenticated encryption (AE) schemes have been developed to enable security services in the resource-limited IIoT/IoT environment [6–8]. AEAD schemes are more lightweight regarding computational resources than symmetric and asymmetric encryption schemes [9]. Thus, a cost-effective AKA framework requiring low computational and communication delays can be realized using an AEAD encryption scheme. Therefore, this paper proposes tackling the aforementioned security challenges, and we have introduced a resource-efficient AKA framework for the smart home called "ESCI-AKA".

1.1. Research Contribution

1. ESCI-AKA is designed with the integration of resource-efficient cryptographic primitives, such as the ASCON encryption scheme and a "hash function". Its primary goal is to enable the establishment of a secure channel (session key) between users and devices within the smart home environment, utilizing the gateway node. By setting up this secure channel, ESCI-AKA allows users and devices to securely communicate sensitive information in an encrypted format, thereby ensuring protection against unauthorized access. Moreover, ESCI-AKA introduces an innovative mechanism for the login and password change processes, which rely on a single encryption operation. This simplifies the authentication procedures involved, streamlining the overall user experience and enhancing system efficiency.
2. The security of ESCI-AKA is validated using the widely recognized formal mechanism known as ROM. This ensures the credibility of the security claims associated with ESCI-AKA. In addition to its security features, ESCI-AKA prioritizes user anonymity, safeguarding the privacy of users within the smart home environment. To further ensure the security of ESCI-AKA, Scyther, a security verification tool, is employed. Scyther aids in validating and confirming that ESCI-AKA is indeed secure and meets the intended security objectives.
3. The efficiency of ESCI-AKA is evaluated based on its computational and communication costs. A comparison is made between ESCI-AKA and several relevant user authentication frameworks, including references [10–16]. ESCI-AKA demonstrates superior efficiency, achieving (65.71%, 66.18%, 84.87%, 66.18%, 79.65%, 61.01%, 86.27%) lower computational costs compared to references [10–16], respectively. Furthermore, ESCI-AKA outperforms these reference frameworks in terms of communication costs, achieving (44.71%, 57.66%, 40.51%, 57.66%, 54.29%, 47.78%, 65.94%) lower communication costs, respectively. In addition to its improved efficiency, ESCI-AKA also provides enhanced security features compared to the relevant security frameworks.

1.2. Paper Organization

The paper's remaining structure is outlined as follows: In Section 2, we provide an overview of the related AKA frameworks. Section 3 offers a comprehensive explanation of the authentication and system models employed in the study. The design procedure of the proposed ESCI-AKA is presented in Section 4, emphasizing the key aspects of its development. Section 5 conducts both formal and informal analyses of ESCI-AKA to evaluate its security capabilities. The performance of ESCI-AKA is demonstrated in Section 6, showcasing its efficiency and effectiveness in real-world scenarios. Finally, in Section 7, the paper concludes with a summary and key findings.

2. Related Work

In the smart home environment, the devices deployed face resource limitations, including restricted computational power, communication capabilities, and storage capacity. Despite these constraints, it is paramount to guarantee the security of information exchange among IIoT devices over the public Internet. Several authors, as referenced in [17–19], have conducted surveys on information protection necessities in both IoT and smart home envi-

ronments, shedding light on the diverse problems that require handling. In the context of a 6LoWPAN-based IoT environment, the study in [20] offers a user authentication technique. This technique operates an AEAD scheme and a hash function, and undergoes security confirmation operating the random oracle model (ROM) and Scyther. Another article [21] demonstrates an authentication technique, which also undergoes security verification via ROM and Scyther. However, the security technique presented in [22] is discovered to be weak against identity guessing, impersonation, and "man-in-the-middle" (MITM) attacks. To handle these problems, an authentication technique operating ECC and a hash function is suggested in [23]. The security of this technique is corroborated by operating the ROM and BAN logic.

The authentication technique suggested in [24] operates a hash function but is encountered to be powerless to "privilege insider, password guessing, and temporary secret number leakage attack", as evidenced in [25]. Moreover, user anonymity remains unprotected within the security technique presented in [24]. In contrast, the authors of [11] designed a user authentication technique utilizing ECC and a hash function. However, this technique is incapable of withstanding "password guessing, impersonation, MITM, stolen smart card, and device capture" attacks. Likewise, it does not guarantee user anonymity. Similarly, the authentication technique suggested in [26] for the IoT environment, which operates symmetric encryption and a hash function, fails to prevent device capture and desynchronization attacks. Additionally, the user authentication framework presented in [26] does not guarantee user anonymity. Another three-factor authentication scheme using ECC and a hash function is proposed in [10], but it is found to be weak against device capture and impersonation attacks. Additionally, user anonymity is not ensured by the user authentication technique in [10]. For an IoT-enabled healthcare system, an authentication technique based on a hash function is suggested in [27], while resource-efficient, this technique is susceptible to diverse attacks, and it does not cover user anonymity. An AEAD and hash-function-based authentication technique is suggested in [28]. The security of this technique is corroborated by operating ROM and Scyther-based formal security analysis. Lastly, an authentication technique for a 6LoWPAN-enabled IoT environment is demonstrated in [29], which mandates fewer computational and communication resources for the authentication phase.

The authors of [30] developed a user authentication technique that lacks mutual authentication capability. In [31], an authentication technique is suggested particularly for the smart home environment. This technique utilizes XOR, concatenation, and hash function operations. For real-time data retrieval from IIoT devices, a cloud-assisted authentication technique is suggested in [32]. This technique employs the chaotic map and hash function. The security of the technique suggested in [33] is validated via ROM. This work demonstrates a strong and efficient authentication technique established on AEAD and hash function, and its security is substantiated by employing ROM and Scyther. In the context of smart grids, an authentication technique is familiarized in [34]. This technique depends on the hash function "Esch256", authenticated encryption, and XOR operations. Further, the security of this technique is corroborated by operating the ROM and Scyther. An exhaustive examination of diverse user authentication techniques is demonstrated in Table 1 [35,36].

Table 1. Overview of Existing User AKA Frameworks.

Reference	Environment	Cryptographic Primitive	Validation + Implementation	Attack Model	Strength/Weakness	Analyzed by	NP	ME
[37]	WSN	HF + XOR	ROM	DY + CK	Vulnerable to privilege insider, device capture, and impersonation attacks.	[38]	3	3
[39]	IoT	HF + XOR	ProVerify + NS2	DY + CK	Vulnerable to password guessing, DoS, replay, and impersonation attacks.	[38]	3	3
[40]	IoD	HF + ECC + XOR	ROM	DY + CK	Vulnerable to drone impersonation attacks.	[41]	3	3
[42]	IoT	HF + XOR	BAN logic	DY + CK	The devised authentication scheme is vulnerable to replay and DoS attacks.	[43]	3	3
[29]	IoT	HF + AEAD + XOR	BAN logic	DY + CK	The devised authentication scheme is secure and resource-efficient.	–	2	2
[28]	IIoT	HF + ECC + AEAD + XOR	ROM + Scyther	DY + CK	The devised authentication framework is lightweight and reliable.	–	3	3
[21]	IIoT	HF + ECC + AEAD + XOR	ROM + Scyther	DY + CK	The propounded user authentication framework is resilient against various security attacks.	–	3	3
[23]	IIoT	HF + ECC + XOR	ROM + BAN logic	DY + CK	Secure against various security threats.	–	3	3
[11]	WSN	HF + ECC + XOR	BAN logic + AVISPA	DY + CK	Incapacitated against off-line guessing, impersonation, and MITM attacks.	[44]	3	4
[27]	IoT	HF + XOR	AVISPA	DY + CK	Cannot prevent password guessing and impersonation attacks.	[45]	3	4
[14]	IIoT	HF + ECC + XOR	BAN logic + AVISPA	DY + CK	Weak against impersonation, replay, device capture attacks.	[46]	3	3
[47]	WSN	HF + ECC + XOR	ROM + Scyther	DY + CK	Cannot resist privilege insider attacks.		3	4
[48]	FC	HF + ECC + XOR	ROM + AVISPA	DY + CK	Cannot prevent DoS, privilege insider, stolen smart card attacks.	[47]	3	3
[16]	IIoT	HF + ECC + XOR	BAN logic	DY + CK	Weak against privileged insider and temporary secret leakage attacks.	–	3	3
[49]	IIoT	HF + ECC + XOR	–	DY	Weak against identity guessing and stolen smart card attacks.	–	3	3
[15]	IoT	HF + ECC + XOR	ROM	DY + CK	Weak against forgery, session key disclosure, and temporary parameters leakage attacks.	[50]	3	4
[51]	SG	HF + PUF + XOR	BAN logic	DY + CK	Cannot resist impersonation and desynchronization attacks	[52]	2	3
ESCI-AKA	IIoT	HF + ECC + AEAD + XOR	ROM + Scyther	DY + CK	Secure against various security attacks.	–	3	3

Note: HF: hash function; NP: number of participants; ME: message exchange; DY: Dolev–Yao; PUF: physical unclonable function; CK: Canetti–Krawczyk; AVISPA: automated validation of Internet security protocols and applications; FC: fog computing; WSN: wireless sensor network; BAN: Burrows–Abadi–Needham; SG: smart grid.

3. System Models

This subsection focuses on the explication of the authentication and attack models, which play a crucial role in the design of the proposed scheme. These models are employed to ensure robust security measures within the system.

3.1. Authentication Model

The authentication model employed for the proposed ESCI-AKA consists of the following components, as illustrated in Figure 2:

- **Gateway:** The trusted authority (TA) assumes the responsibility of deploying gateway nodes (GW_k) within the smart home environment. These gateway nodes provide internet connectivity to the IIoT-enabled devices deployed in the environment. Additionally, GW_k stores the sensitive parameters associated with the remote user and smart embedded devices. It possesses the capability to establish connections between IIoT-enabled devices and the Internet, using cellular or other Internet connectivity options. Furthermore, all IIoT-enabled devices deployed in the environment are connected to GW_k through communication protocols such as WiFi, 6LoWPAN, or Zigbee.
- **Smart Embedded Device:** Smart embedded devices (SEDs) refer to resource-limited devices deployed within the smart home environment. Each SED, denoted as SED_j, is equipped with communication, storage, and computational resources. These devices can establish communication with GW_k using communication protocols like WiFi, 6LoWPAN, or Zigbee. Additionally, SEDs are equipped with sensing modules, allowing them to collect sensitive information from their surrounding environment. This collected data can be transmitted to a central location for further analysis.
- **Remote User:** The user possesses smart devices (SD_i) equipped with biometric sensors. Communication between U_i and U_i occurs through the gateway node (GW_k). Furthermore, U_i can communicate with GW_k utilizing cellular or internet technology. In order to access real-time information from the deployed SED_j in the smart home environment, it is crucial to ensure that only authorized U_i can obtain such information. To aid in comprehending the proposed scheme, Table 2 provides an elucidation of the various symbols employed.

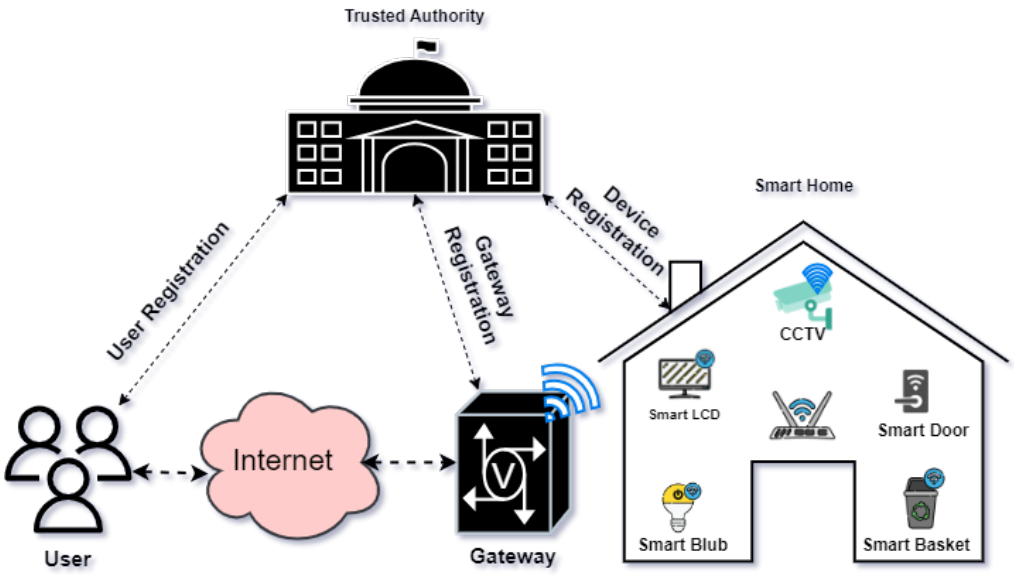

Figure 2. Authentication model for the smart home environment.

Table 2. Notations Utilized in ESCI-AKA.

Notation	Description
U_i	The remote user
SD_{U_i}	IoT-enabled smart device
GID_k	The identity of the gateway
ID_{U_i}, PW_{U_i}	The identity and password of U_i, respectively
PID^c	Current pseudo-identities
PID^o	Old pseudo-identities
TiS_1, TiS_2, TiS_3	Timestamps utilized in ESCI-AKA's authentication phase
ADL, TR	Allowed time delay limit and received time of a message
AD_x	Associative data, where $x = 1, 2, 3, \ldots, n$
$E_k(P)$, $D_k(C)$	Secret-key-based encryption of string "P" and decryption "C" using ASCON
RNU_y	Random numbers utilized while accomplishing the AKA phase
P_z	Plaintext $z = 1, 2, 3, 4, 5, 6, 7$
C_a	Ciphertext $a = 1, 2, 3, \ldots, 7$
Bio_{U_i}, σ, $FE.Gen(\cdot)$, hd, $FE.Rep(\cdot)$	User biometric information and key, respectively
$FE.Gen(\cdot)$, hd, $FE.Rep(\cdot)$	User key generation algorithm, reproduction parameter, and key reproduction algorithm, respectively
$\oplus$, $\|\|$, $H(\cdot)$	XOR, concatenation, and hash function, respectively

3.2. Attack Model

In order to evaluate the security of the ESCI-AKA framework, we utilize the widely acknowledged Dolev–Yao (DY) model [53,54]. The DY model effectively simulates the smart home environment. In the simulation, we consider that all parties in the smart home environment establish their trust in a trusted authority (TA) and communicate information securely via a dedicated communication channel with the TA. The TA handles the system initialization, user registration, and cancellation. Apart from exchanges with the TA, all parties intercommunicate with other parties via public channels. In the DY model, an attacker has the capability to both eavesdrop and modify messages if they are communicated over open channels. Given the complex nature of the smart home environment, an attacker may also resort to physically seizing devices to obtain private parameters and information.

Furthermore, there is a potential risk that the user's SD_i may be stolen or lost, which can result in the adversary attaining access to the user's confidential information via the compromised device. This emphasizes the importance of addressing security concerns related to users' SD_is. In addition, we consider the CK adversary attack model, which builds upon the DY model and enhances the capabilities of the adversary. This model grants the adversary the ability to obtain secret public parameters during the authentication session, thereby enabling them to acquire a short-lived partial key. This further amplifies the adversary's capabilities and underscores the need for robust security measures.

4. The Proposed ESCI-AKA Framework

The ESCI-AKA system includes several phases: smart embedded registration, remote user registration, AKA, and password and biometric change. Each of these phases will be discussed in detail in the subsequent subsections.

4.1. Gateway Registration Phase

The TA is responsible for registration and deployment in the smart home environment. For deploying GW_k, TA selects a distinct identity, GID_k, and generates a long-term gateway key, LGK, and stores the parameters $\{LGK, GID_k\}$ in the database of the GW_k.

4.2. Smart Embedded Device Registration Phase

The TA is responsible for deploying SED_j in the smart home environment after its registration. The following steps are imperative for the registration of SED_j.

Step SEDRP-1

The TA selects a unique identity for SID_{SED_j} and computes $DSK = H(SID_{SED_j} \parallel LGK)$. Finally, TA stores the parameters $\{SID_{SED_j}, DSK\}$ in the memory of SED_j and stores the parameter SID_{SED_j} in the database of GW_k .

4.3. Remote User Registration Phase

TA executes the following steps to register U_i using a secure channel.

4.3.1. Step RMURP-1

U_i selects its own secret parameter, such as distinct identity, ID_{U_i}, and password, PW_{U_i}. In addition, U_i wakes its own biometric information (Bio_{U_i}) on the biometric sensor deployed on the IoT-enabled smart device (SD_{U_i}), and SD_{U_i} uses $FE.Gen(\cdot)$ to generate the biometric key by computing $(\sigma_1, hd) = FE.Gen(Bio_{U_i})$ after taking Bio_{U_i} as the input. In addition, SD_{U_i} picks a random number, RNU_1, and, by using hashing algorithm, computes $K_1 = H(ID_{U_i} \parallel PW_{U_i} \parallel \sigma_1)$ and, by using the ASCON encryption algorithm, computes $((C_1, C_2, C_3), MAC_1) = E_{K_1}\{AD_1, P_1, P_2, P_3\}$, where C_1, C_2, and C_3 are the ciphertext, $P_1 = ID_{U_i}$, $P_2 = PW_{U_i}$, and $P_3 = \sigma_1$ are the plaintext, and $AD_1 = RNU_1$ is the associative data. Moreover, SD_{U_i} computes $CID_i = C_1 \oplus C_2$ and constructs the message with parameters $\{CID_i, C_3\}$ and sends it to TA via secure channel.

4.3.2. Step RMURP-2

TA, after obtaining $\{CID_i, C_3\}$, picks RNU_2 and computes $PID_i = (CID_i \oplus RNU_2)$. In addition, TA selects the list of registered SID_{SED_j} for U_i, from where U_i can access real-time information. TA stores the parameters $\{PID_i, C_3\}$ in the database of GW_k. Finally, TA sends the parameters $\{RNU_2, SID_{SED_j}, PID_i, GID_k\}$ to U_i/SD_{U_i} through a secure channel.

4.3.3. Step RMURP-3

U_i/SD_{U_i}, after sending the parameters $\{RNU_2, SID_{SED_j}, PID_i, GID_k\}$ to U_i/SD_{U_i} from TA, computes the following:

$$Q_1 = (SID_{SED_j} \parallel C_3) \oplus H(C_2^* \oplus C_3^* \oplus C_1^*), \tag{1}$$

$$GID_k^* = GID_k \oplus H(C_2^* \oplus C_3^*), \tag{2}$$

$$RNU_2^* = H(C_2^* \oplus C_3^* \oplus C_1^*) \oplus RNU_2. \tag{3}$$

Finally, U_i/SD_{U_i} stores the parameters $\{RNU_2^*, GID_k^*, AD_1, Q_1, MAC_1, hd, FE.Gen(\cdot), FE.Rep(\cdot)\}$ in its own memory.

Remark 1. *In the proposed ESCI-AKA, we utilize ASCON [6] as the encryption/decryption algorithm. ASCON is an AEAD scheme that ensures the simultaneous provision of integrity, authenticity, and confidentiality of information. The encryption operation of ASCON can be represented as* $((C, MAC) = E_K\{AD, P\})$*, where C, MAC, AD, P, and K represent the ciphertext, authentication code (Tag), associated data, plaintext, and encryption key, respectively.*

Similarly, the decryption operation can be expressed as $((P, MAC1) = E_K\{AD, C\})$*. In this case, the generated MAC1 during the decryption operation will be valid if it matches the original MAC value. If the condition,* $MAC = MAC1$*, holds, the plaintext will be considered valid. Otherwise, if the condition is not satisfied, the plaintext will be deemed invalid.*

Definition 1. *An AEAD scheme is reflected as protected if $\mathcal{A}$'s ultimate OCCA3 advantage is insignificant. The OCCA3 advantage of $\mathcal{A}$ on an AEAD is the cumulative advantage of $\mathcal{A}$ for performing a chosen plaintext attack and compromising the integrity of an AEAD scheme [55,56].*

$$\begin{aligned} Adv^{OCCA3}_{AEAD,\mathcal{A}}(polt) \leq\ & Adv^{OPRP-CPA}_{AEAD}(Qr, len, polt) \\ & + Adv^{INT-CTXT}_{AEAD}(Qr, len, polt), \end{aligned} \tag{4}$$

Here, Adv, Qr, len, $polt$, $OPRP-CPA$, and $INT-CTXT$ denote the advantage, number of queries performed by $\mathcal{A}$, length, polynomial time, online permutations, and ciphertext integrity, respectively.

Remark 2. *In the proposed ESCI-AKA, the fuzzy extractor (FE) technique is utilized to derive a reliable biometric key from the user's biometric information. The FE consists of two main functions: $FE.Gen(\cdot)$ and $FE.Rep(\cdot)$. The $FE.Gen(\cdot)$ function takes the biometric information of the user as input and generates both the biometric key and the corresponding helper data. On the other hand, the $FE.Rep(\cdot)$ function takes the helper data and the biometric information as input and reconstructs the biometric key. To reconstruct the biometric key, a condition is imposed: $HD(Bio_{U_i}, Bio^*_{U_i}) \leq et$. Here, HD denotes the Hamming distance, and et represents the allowable difference between Bio_{U_i} and $Bio^*_{U_i}$ (the login biometric template and the current biometric sample, respectively). If the Hamming distance between these two values falls within the specified threshold, et, the biometric key can be successfully reconstructed.*

4.4. AKA Phase

During this stage, U_i/SD_{U_i} and SED_j work together to establish a secure channel, also known as a session key, for secure information exchange. The secure channel is established through mutual authentication and session key exchange. The subsequent algorithms are performed by using U_i/SD_{U_i} and SED_j to initiate the setup of the session key or secure channel.

4.4.1. Local Authentication and Generation of MG_1

Algorithm 1 accomplishes the task of local authentication and generates the authentication message, MG_1. The algorithm starts by taking the input parameters as ID_{U_i}, PW_{U_i}, $Bio^*_{U_i}$, MAC_1, and hd. It then generates the biometric key (σ_1) using the reproduction function of FE. This reproduction function takes the biometric information of the user and helper data as input and produces the biometric key. Furthermore, the algorithm derives the encryption key and utilizes the ASCON encryption algorithm for encryption. In this encryption process, $P^*_1 = IDU_i$, $P^*_2 = PWU_i$, and $P^*_3 = \sigma^*_1$ serve as the plaintext, while C^*_1, C^*_2, and C^*_3 represent the associated ciphertext. The integrity of the secret credentials and the local authentication of the user is verified by checking if $MAC^*_1 \stackrel{?}{=} MAC_1$. Upon successful local authentication, the algorithm retrieves the values SID_{SED_j}, GID_k, and C_3 for further processing.

In addition, the algorithm proceeds to derive a temporary pseudo-identity and compute associative data. ASCON encryption is employed to encrypt certain parameters, such as SID_{SED_j} and RNU_3. It is worth noting that the nonce used in the ASCON encryption and decryption algorithm is computed by XORing associative data and a secret encryption key. Finally, SD_{U_i} constructs the message, MG_1, which includes TiS_1 (timestamp), PID_i (pseudo-identity), C_4, C_5, and MAC_2. This message is then transmitted to GW_k using an open channel of communication.

4.4.2. Validates MG_1 and Generates MG_2

Algorithm 2 facilitates the validation of the received message, MG_1, by GW_k and generates MG_2. Upon receiving the message, MG_1, GW_k performs several checks to ensure its validity. Firstly, it examines whether the message MG_1 is a replay by comparing the condition $T_{ADL} \geq |TiS_1 - RTM|$. If the message is determined to be a replay, GW_k discards MG_1. Otherwise, GW_k proceeds with further checks.

Algorithm 1 Performs Local Authentication and Generates MG_1

Input: { ID_{U_i}, PW_{U_i}, $Bio^*_{U_i}$, MAC_1, hd}
Output: {TiS_1, PID_i, C_4, C_5, MAC_2}
1: **procedure** ALGO-1({ID_{U_i}, PW_{U_i}, $Bio^*_{U_i}$, hd, MAC_1})
2: $\sigma^*_1 \leftarrow FE.Rep(Bio^*_{U_i}, hd)$
3: $K^*_1 \leftarrow H(ID_{U_i} \parallel PW_{U_i} \parallel \sigma^*_1)$
4: $((C^*_1, C^*_2, C^*_3), MAC^*_1) \leftarrow E_{K^*_1}\{AD_1, P^*_1, P^*_2, P^*_3\}$
5: $AD_1 \leftarrow RNU_1$
6: **if** $MAC^*_1 \stackrel{?}{=} MAC_1$ **then**
7: $(SID_{SED_j} \parallel C_3) \leftarrow (Q_1 \oplus H(C^*_2 \oplus C^*_3))$
8: $GID_k \leftarrow GID^*_k \oplus H(C^*_2 \oplus C^*_3 \oplus C^*_1)$
9: $RNU_2 \leftarrow H(C^*_2 \oplus C^*_3 \oplus C^*_1) \oplus RNU^*_2$
10: selects RNU_3 and TiS_1
11: $PID_i \leftarrow (C^*_1 \oplus C^*_2 \oplus RNU_2)$
12: $AD_2 \leftarrow (PID_i \oplus TiS_1 \oplus GID_k)$
13: $N_1 \leftarrow (PID_i \oplus TiS_1 \oplus GID_k) \oplus C_3$
14: $((C_4, C_5), MAC_2) \leftarrow E_{(C_3 \| N_1)}\{AD_2, SID_{SED_j}, RNU_3\}$
15: **else**
16: terminates execution
17: **end if**
18: **end procedure**

GW_k verifies if the condition $PID_i \stackrel{?}{=} PID^c_i$ is satisfied. If it holds true, GW_k retrieves C_3 and RNU_4 from its own database. On the other hand, if the condition $PID_i \stackrel{?}{=} PID^o i$ is met, GW_k retrieves C_3 and RNU_2 from its database. Here, PID_i refers to the received pseudo-identity with the message $MG1$, PID^c_i represents the current pseudo-identity, and PID^o_i corresponds to the old pseudo-identity. If no match is found in either case, GW_k terminates the AKA process.

Upon obtaining C_3 and RNU_2, GW_k calculates the associative data, AD_3, and nonce, N_2. Furthermore, after performing the decryption using C_3 as the secret key, GW_k obtains $((SID_{SED_j}, RNU_3), MAC_3)$. The decryption operation is carried out utilizing the ASCON decryption algorithm. Moreover, GW_k verifies the integrity of the received message by checking if $(MAC_2 \stackrel{?}{=} MAC_3)$. Finally, GW_k retrieves the values, $SIDSED_j$ and RNU_3, for further processing.

GW_k performs the computation of plaintexts P_4, P_5, and P_6, along with the generation of associative data, AD_4. By utilizing the ASCON encryption algorithm and the encryption key, K_2, the encryption process encrypts P_4, P_5, and P_6, resulting in the generation of $((C_6, C_7, C_8), MAC_4)$. Finally, GW_k constructs the message, MG_2: {TiS_2, C_6, C_7, C_8, MAC_4}, and sends MG_2 to SED_j using the open communication channel.

Remark 3. *To avoid identity desynchronization, GW_k computes a new pseudo-identity, $PID^n_i = (PID_i \oplus RNU_2 \oplus RNU_4)$, and updates PID^c_i with PID^n_i. In addition, GW_k also keeps C_3 and updates RNU_2 with RNU_4. Furthermore, GW_k also updates PID^o_i with PID^c_i and keeps C_3 and RNU_2 in its own database.*

Algorithm 2 Validates MG_1 and Generates MG_2

Input: $\{TiS_1,\ PID_i,\ C_4,\ C_5,\ MAC_2\}$
Output: $\{TiS_2,\ C_6,\ C_7,\ C_8,\ MAC_4\}$
procedure ALGO-2($\{TiS_1,\ PID_i,\ C_4,\ C_5,\ MAC_2, LGK, SID_{SED_j}\}$)
  if $T_{ADL} \geq |TiS_1 - RTM|$ **then**
    if $(PID_i \stackrel{?}{=} PID_i^c)$ **then**
      retrieves C_3 and RNU_4
      if $(PID_i \stackrel{?}{=} PID_i^o)$ **then**
        retrieves C_3 and RNU_2
        $AD_3 \leftarrow (PID_i \oplus TiS_1 \oplus GID_k)$
        $N_2 \leftarrow (PID_i \oplus TiS_1 \oplus GID_k) \oplus C_3$
        $((SID_{SED_j}, RNU_3), MAC_3) \leftarrow D_{(C_3 \| N_2)}\{AD_3, C_4, C_5\}$
        if $(MAC_2 \stackrel{?}{=} MAC_3)$ **then**
          generates TiS_2, RNU_4 and RNU_5
          $P_4 \leftarrow C_3 \oplus RNU_3$
          $P_5 \leftarrow RNU_4$
          $P_6 \leftarrow RNU_5$
          $K_2 \leftarrow H(LGK \parallel SID_{SED_j})$
          $AD_4 \leftarrow (SID_{SED_j} \oplus TiS_2)$
          $((C_6, C_7, C_8), MAC_4) \leftarrow E_{K_2}\{AD_4, P_4, P_5, P_6\}$
          $PID_i^n \leftarrow (PID_i \oplus RNU_2 \oplus RNU_4)$
          updates PID_i^c with PID_i^n
          keeps C_3 and updates RNU_2 with RNU_4
          updates PID_i^o with PID_i^c
          keeps C_3 and RNU_2
        else
          terminates execution
        end if
      else
        terminates execution
      end if
    else
      terminates execution
    end if
  else
    terminates execution
  end if
end procedure

4.4.3. Validates MG_2 and Generates MG_3

Algorithm 3 is designed to enable the validation of the received message, MG_2, by SED_j and subsequently generate MG_3. The freshness of the message is validated by SED_j using the condition $T_{ADL} \geq |TiS_2 - RTM|$. If the received message is determined to not be fresh, SED_j terminates the authentication process. Conversely, if the message is fresh, SED_j proceeds to compute the associative data, AD_5, and the decryption key, K_3. The ASCON decryption algorithm utilizes K_3 to decrypt the ciphertext and generate the plaintext along with the authentication parameter, i.e., $((P_4, P_5, P_6), MAC_5)$. The integrity of these returned parameters is verified by comparing MAC_4 with MAC_5. If they match, the returned (P_4, P_5, P_6) is considered valid, where $P_4 = C_3 \oplus RNU_3$, P_5 is equal to RNU_4, and P_6 is equal to RNU_5. In the case of a mismatch, the authentication process is terminated by SED_j.

Algorithm 3 Validates MG_2 and Generates MG_3

Input: $\{TiS_2, C_6, C_7, C_8, MAC_4\}$
Output: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$
procedure ALGO-3($\{TiS_2, C_6, C_7, C_8, MAC_4\}$)
  if $T_{ADL} \geq |TiS_2 - RTM|$ **then**
    $AD_5 \leftarrow (SID_{SED_j} \oplus TiS_2)$
    $K_3 \leftarrow DSK$
    $((P_4, P_5, P_6), MAC_5) \leftarrow D_{K_3}\{AD_5, C_6, C_7, C_8\}$
    if $(MAC_4 \stackrel{?}{=} MAC_5)$ **then**
      selects TiS_3 and RNU_6
      $SK_{SED_j} \leftarrow H(RNU_6 \parallel (C_3 \oplus RNU_3) \parallel (SID_{SED_j} \oplus RNU_6 \oplus RNU_5) \parallel TiS_3)$
      $AD_6 \leftarrow (SK^a_{SED_j} \oplus SK^b_{SED_j})$
      $K_4 \leftarrow (SID_{SED_j} \oplus (C_3 \oplus RNU_3))$
      $N_3 \leftarrow (SID_{SED_j} \oplus (C_3 \oplus RNU_3) \oplus AD_6)$
      $P_7 \leftarrow RNU_4$
      $P_8 \leftarrow (SID_{SED_j} \oplus RNU_6 \oplus RNU_5)$
      $((C_9, C_{10}), MAC_6) \leftarrow E_{(K_4 \| N_3)}\{AD_6, P_7, P_8\}$
    else
      terminates execution
    end if
  else
    terminates execution
  end if
end procedure

Moreover, SED_j proceeds to select TiS_3 and RNU_6, followed by the computation of the session key, SK_{SED_j}, for future secure communication with the user. Once SK_{SED_j} is computed, SED_j proceeds to calculate the associative data, AD_6; nonce, N_3; encryption key, K_4; and plaintexts, P_7 and P_8. By utilizing the ASCON encryption algorithm, SED_j encrypts P_7 and P_8 with the encryption key, K_4, generating $((C_7, C_8), MAC_6)$. Subsequently, SED_j constructs the message, MG_3, as $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$ and transmits it to SD_i or the user via the public communication channel.

4.4.4. Validates MG_3 and Session Key

Algorithm 4 is designed to enable the validation of the received message MG_3 by SD_i and subsequently generate the session key. If SD_i or the user verifies the freshness of the message MG_3, they need to check the condition $T_{ADL} \geq |TiS_3 - RTM|$. If the message is determined to be valid, SD_i calculates the decryption key, K_5, and the nonce, N_4. By utilizing the ASCON decryption algorithm with K_5 and N_4, (P_7, P_8), and MAC_7 are generated.

In addition, the integrity of the message, MG_3, is checked by comparing MAC_6 with MAC_7. To ensure secure encrypted communication between SD_i and SED_j, a session key, SK_{U_i}, is established. In order to verify the session key, SD_i computes AD_7 and checks whether AD_6 matches AD_7. If they match, this indicates that the session keys derived at SD_i and SED_j, which are identical.

Finally, RNU_2^* is computed, and SD_i updates RNU_2^* with the new value, RNU_2.

4.5. Secret Credentials Change Phase

The ESCI-AKA mechanism offers a user-friendly way to modify the secret credentials (e.g., passwords and biometrics) of U_i using Algorithm 5. To initiate this process, U_i provides its old secret credentials, namely, ID_{U_i} and PW^oU_i, and biometric information $Bio^o_{U_i}$ to SD_i. Upon receiving these secret credentials, SD_i utilizes the FE to generate the biometric key, σ^o_1, from $Bio^o_{U_i}$ and hd^o as input parameters. Subsequently, SD_i derives the encryption key, K^o_1, and employs the ASCON encryption algorithm to encrypt P^o_1, P^o_2, P^o_3, resulting in $((C^o_1, C^o_2, C^o_3), MAC^o_1)$ using the encryption key, K^o_1, and $AD^o_1 = RNU_1$.

Algorithm 4 Validates MG_3 and Generates Session Key

Input: $\{TiS_3,\ C_9,\ C_{10},\ AD_6,\ MAC_6\}$
Output: {Session key SK_{U_i} Generation and Mutual Authentication}
procedure ALGO-4($\{TiS_3,\ C_9,\ C_{10},\ AD_6,\ MAC_6\}$)
  if $T_{ADL} \geq |TiS_3 - RTM|$ **then**
    $K_5 \leftarrow (SID_{SED_j} \oplus (C_3 \oplus RNU_3))$
    $N_4 \leftarrow K_5 \oplus AD_6$
    $((P_7, P_8), MAC_7) \leftarrow D_{(K_5 \| N_4)}\{AD_6, C_9, C_{10}\}$
    if $MAC_6 \stackrel{?}{=} MAC_7$ **then**
      $P_7 \leftarrow RNU_4$
      $P_8 \leftarrow (SID_{SED_j} \oplus RNU_6 \oplus RNU_5)$
      $SK_{U_i} \leftarrow H(RNU_6 \parallel (C_3 \oplus RNU_3) \parallel (SID_{SED_j} \oplus RNU_6 \oplus RNU_5) \parallel TiS_3)$
      $AD_7 \leftarrow (SK^a_{U_i} \oplus SK^b_{U_i})$
      if $AD_6 \stackrel{?}{=} AD_7$ **then**
        $RNU^*_2 \leftarrow H(C^*_2 \oplus C^*_3 \oplus C^*_1) \oplus RNU_4$
        updates RNU_2 with RNU^*_2
        Mutual authentication is achieved
      else
        terminates execution
      end if
    else
      terminates execution
    end if
  else
    terminates execution
  end if
end procedure

Algorithm 5 Performs Password Change and Biometric Update

Input: $\{ID_{U_i}, Bio^o_{U_i}, PW^o_{U_i}, RNU_2, AD_1, Q_1, MAC_1, hd, GID^*_k, FE.Gen(\cdot), FE.Rep(\cdot)\}$
Output: $\{RNU^n_2, AD^n_1, Q^n_1, MAC^n_1, hd^n, GID^n_k, FE.Gen(\cdot), FE.Rep(\cdot)\}$
procedure ALGO-1($\{RNU_2, AD_1, Q_1, MAC_1, hd, GID^*_k, FE.Gen(\cdot), FE.Rep(\cdot)\}$)
  $\sigma^o_1 \leftarrow FE.Rep(Bio^o_{U_i}, hd)$
  $K^o_1 \leftarrow H(ID_{U_i} \parallel PW^o_{U_i} \parallel \sigma^o_1)$
  $((C^o_1, C^o_2, C^o_3), MAC^o_1) \leftarrow E_{K^o_1}\{AD_1, P^o_1, P^o_2, P^o_3\}$
  $AD^o_1 \leftarrow RNU_1$
  if $MAC^o_1 \stackrel{?}{=} MAC_1$ **then**
    $(SID_{SED_j} \parallel C_3) \leftarrow (Q_1 \oplus H(C^o_2 \oplus C^o_3))$
    $GID_k \leftarrow GID^o_k \oplus H(C^o_2 \oplus C^o_3 \oplus C^o_1)$
    $RNU_2 \leftarrow H(C^o_2 \oplus C^o_3 \oplus C^o_1) \oplus RNU^o_2$
    Enters the new or fresh secret parameters
    $(\sigma^n_1, hd^n) \leftarrow FE.Gen(Bio^n_{U_i})$
    $K^n_1 \leftarrow H(ID_{U_i} \parallel PW^n_{U_i} \parallel \sigma^n_1)$
    $AD^n_1 \leftarrow RNU^n_1$
    $((C^n_1, C^n_2, C^n_3), MAC^n_1) \leftarrow E_{K^n_1}\{AD^n_1, P^n_1, P^n_2, P^n_3\}$
    $Q^n_1 \leftarrow ((SID_{SED_j} \parallel C_3) \oplus H(C^n_2 \oplus C^n_3))$
    $GID^n_k \leftarrow GID_k \oplus H(C^n_2 \oplus C^n_3 \oplus C^n_1)$
    $RNU^n_2 \leftarrow H(C^n_2 \oplus C^n_3 \oplus C^n_1) \oplus RNU_2$
  else
    terminates execution
  end if
end procedure

To ensure the authenticity of the secret credentials and perform local authentication, SD_i verifies the condition, $MAC_1^o \stackrel{?}{=} MAC_1$. If the condition holds true, SD_i derives the parameters SID_{SED_j}, GID_k, and RNU_2. Moreover, SD_i notifies U_i to input a new $PW_{U_i}^n$ and update the biometric information to $Bio_{U_i}^n$ to complete the process. SD_i picks a new random number, RNU_1^n, and computes $\{RNU_2^n, AD_1^n, Q_1^n, MAC_1^n, hd^n, GID_k^n, FE.Gen(\cdot), FE.Rep(\cdot)\}$. Finally, it replaces $\{RNU_2, AD_1, Q_1, MAC_1, hd, GID_k^*, FE.Gen(\cdot), FE.Rep(\cdot)\}$ with $\{RNU_2^n, AD_1^n, Q_1^n, MAC_1^n, hd^n, GID_k^n, FE.Gen(\cdot), FE.Rep(\cdot)\}$ in SD_i's memory.

5. Security Validation

We provide a security analysis of ESCI-AKA, formally and informally, in this section.

5.1. Informal Security Analysis

In this subsection, we provide an informal security analysis of the proposed ESCI-AKA framework.

5.1.1. Secret Credential Change Attack

$\mathcal{A}$, after capturing SD_i, can obtain the sensitive parameters, such as $\{RNU_2^*, AD_1, Q_1, MAC_1, hd, GID_k^*, FE.Gen(\cdot), FE.Rep(\cdot)\}$, which are stored in the memory of SD_i at the time of registration. $\mathcal{A}$ cannot update the secret credentials, such as PW_{U_i}, Bio_{U_i}, and ID_{U_i}, because $\mathcal{A}$ needs to compute:

$$(\sigma_1^{\mathcal{A}}) = FE.Rep(Bio_{U_i}^{\mathcal{A}}, hd), \tag{5}$$

$$K_1^{\mathcal{A}} = H(ID_{U_i}^{\mathcal{A}} \parallel PW_{U_i}^{\mathcal{A}} \parallel \sigma_1^{\mathcal{A}}), \tag{6}$$

$$((C_1^{\mathcal{A}}, C_2^{\mathcal{A}}, C_3^{\mathcal{A}}), MAC_1^{\mathcal{A}}) = E_{K_1^{\mathcal{A}}}\{AD_1, P_1^{\mathcal{A}}, P_2^{\mathcal{A}}, P_3^{\mathcal{A}}\}, \tag{7}$$

$$MAC_1^{\mathcal{A}} \stackrel{?}{=} MAC_1. \tag{8}$$

$\mathcal{A}$ can update the secret credentials, such as PW_{U_i}, Bio_{U_i} if Condition (8) holds. However, $\mathcal{A}$ cannot generate the biometric key. Thus, without knowing the secret credentials, it is hard for $\mathcal{A}$ to update the password and biometric key. Hence, the proposed ESCI-AKA is resistant to password and biometric key update attacks.

5.1.2. Replay Attack

To prevent a replay attack, in ESCI-AKA, timestamps are incorporated in all the communicated messages. The freshness of the messages, MG_1, MG_2, and MG_3, are checked through the conditions $T_{ADL} \geq |TiS_1 - RTM|$, $T_{ADL} \geq |TiS_2 - RTM|$, and $T_{ADL} \geq |TiS_3 - RTM|$, respectively. If any of the conditions fail, the associated message is considered to be invalid or replayed. Thus, the proposed ESCI-AKA demonstrates resistance against replay attacks.

5.1.3. DoS Attack

The proposed scheme prevents a DoS attack through the local authentication of U_i secret credentials. To achieve local authentication, the SD_i of U_i needs to perform the following computations:

$$(\sigma_1^*) = FE.Rep(Bio_{U_i}^*, hd), \tag{9}$$

$$K_1^* = H(ID_{U_i} \parallel PW_{U_i} \parallel \sigma_1^*), \tag{10}$$

$$((C_1^*, C_2^*, C_3^*), MAC_1^*) = E_{K_1^*}\{AD_1, P_1, P_2, P_3\}, \tag{11}$$

$$where\ AD_1 = RNU_1, \tag{12}$$

$$MAC_1^* \stackrel{?}{=} MAC_1, \tag{13}$$

If the condition in (13) holds, then SD_i sends the authentication and AKA request to GW_k. Otherwise, SD_i stops the further execution of the AKA phase. In this way, the proposed ESCI-AKA prevents DoS attacks.

5.1.4. MITM Attack

An authentication and key agreement mechanism must be capable of resisting an MITM attack. In the proposed ESCI-AKA, there are three messages, such as MG_1: $\{TiS_1, PID_i, C_4, C_5, MAC_2\}$, MG_2: $\{TiS_2, C_6, C_7, C_8, MAC_4\}$, and MG_3: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$, which are disseminated by SD_i, GW_k, and SED_j, respectively, to accomplish a secure channel. To effectuate an MITM attack, $\mathcal{A}$ requires knowing the secret parameters, which are used in the construction of all the messages communicated during the AKA phase, such as RNU_3, RNU_4, RNU_5, RNU_6, C_3, GID_k, C_1, and C_2. A lack of knowledge of these parameters makes it hard for $\mathcal{A}$ to generate an MITM attack. This way, ESCI-AKA exhibits resistance to MITM attacks.

5.1.5. U_i Impersonation Attack

In ESCI-AKA, when the message MG_1: $\{TiS_1, PID_i, C_4, C_5, MAC_2\}$ is transmitted to GW_k for authentication and the AKA process, an impersonation attack requires $\mathcal{A}$ to generate fabricated or modified messages using random parameters, like $C_3^{\mathcal{A}}$, $GID_k^{\mathcal{A}}$, and $RNU_3^{\mathcal{A}}$. However, since $\mathcal{A}$ does not possess the knowledge of the actual valid parameters (C_3, GID_k, and RNU_3), it is unable to generate a valid message, especially the parameter MAC_2. Consequently, $\mathcal{A}$ cannot impersonate a valid user in the smart home environment. Therefore, ESCI-AKA effectively prevents user impersonation attacks.

5.1.6. SED_j Impersonation Attack

In ESCI-AKA, the message, MG_3: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$, is transmitted to U_i. Therefore, in order to impersonate a valid SED_j, $\mathcal{A}$ would need to generate a fabricated message, $MG_3^{\mathcal{A}}$. However, for the generation of $MG_3^{\mathcal{A}}$, $\mathcal{A}$ must possess knowledge of the parameters RNU_3, RNU_4, RNU_5, RNU_6, C_3, and SID_{SED_j}. Without knowing these parameters, $\mathcal{A}$ cannot generate a valid MG_3. As a result, the proposed ESCI-AKA effectively safeguards against impersonation attacks targeting SED_j.

5.1.7. Temporary Parameter Leakage Attack

In ESCI-AKA, the session key is computed as $SK_{SED_j} (= SK_{U_i}) = H(RNU_6 \parallel (C_3 \oplus RNU_3) \parallel (SID_{SED_j} \oplus RNU_6 \oplus RNU_5) \parallel TiS_3)$, which incorporates a combination of ephemeral and long-term parameters to enhance its security measure. $\mathcal{A}$ needs to obtain ephemeral parameters, such as RNU_3, RNU_4, RNU_5, and RNU_6, and long-term parameters, such as C_3, SID_{SED_j}, GID_k, C_1, and C_2, to breach the security of the session key. Hence, the proposed ESCI-AKA is resistant to temporary parameter leakage attacks.

5.1.8. Anonymity and Untraceability

During the secure channel establishment (AKA) phase, three messages are exchanged: MG_1: $\{TiS_1, PID_i, C_4, C_5, MAC_2\}$, $MG2$: $\{TiS_2, C_6, C_7, C_8, MAC_4\}$, and MG_3: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$. After capturing these messages, $\mathcal{A}$ is unable to determine the user's identity, which prevents user tracking. Additionally, all the messages generated during the current and previous AKA phases are different and random, making it impossible for $\mathcal{A}$ to establish any correlation between captured messages from different sessions. Furthermore, even if $\mathcal{A}$ obtains parameters such as $\{RNU_2^*, AD_1, Q_1, MAC_1, hd, GID_k^*, FE.Gen(\cdot), FE.Rep(\cdot)\}$, it cannot extract the real identity of the user. Thus, the proposed ESCI-AKA ensures anonymous communication.

5.1.9. Desynchronization

In the proposed authentication scheme, the pseudo-identity is updated at the gateway. However, this updating of pseudo-identities introduces vulnerability to desynchronization attacks. To mitigate the risk of desynchronization attacks, we have implemented a solution by retaining both the old and current pseudo-identities at the gateway. In the event of eavesdropping or jamming attacks, an attacker can capture messages and drop them at any point during the execution of the proposed scheme. Nevertheless, even if the messages are dropped, users can still utilize the old identities to successfully complete the AKA phase of the scheme. This safeguard ensures protection against desynchronization attacks.

5.2. ROM-Based Validation

A formal analysis of the security of the session key generated during the AKA phase of ESCI-AKA is conducted using the random oracle model (ROM). The essential components of the ROM are outlined below:

- **Participants**: In ESCI-AKA, there are three main participants: U_i, GW_k, and SED_j. We represent the instances, $p1$, $p2$, and $p3$, of U_i, GW_k, and SED_j of these participants as $\Pi_{U_i}^{p1}$, $\Pi_{GW_k}^{p2}$, and $\Pi_{SED_j}^{p3}$, which serve as oracles in the system.
- **Partnership**: Upon reaching the acceptance state, the instances, $\Pi_{U_i}^{p1}$ and $\Pi_{SED_j}^{p3}$, establish a partnership if they possess a shared session key.
- **Freshness**: $\mathcal{A}$ is incapable of disclosing the session key that is established between $\Pi_{U_i}^{p1}$ and $\Pi_{SED_j}^{p3}$ during the AKA phase.

The capabilities of $\mathcal{A}$ are analyzed in Section 3.2. Furthermore, $\mathcal{A}$ can influence different queries to execute various attacks on ESCI-AKA.

- **Execute** $(\Pi_{U_i}^{p1}, \Pi_{GW_k}^{p2}, \Pi_{SED_j}^{p3})$: A passive attack can be simulated employing this query, authorizing $\mathcal{A}$ to model and observe the passive attack. With this query, $\mathcal{A}$ can acquire all the messages exchanged during the AKA process of ESCI-AKA.
- **Test** (Π^{p1}): $\mathcal{A}$ operates this query to demonstrate whether the imagined session key is definitely the correct session key or merely a random guess.
- **Reveal** (Π^{p1}): This query is effectuated by $\mathcal{A}$ to acquire the session key maintained by the oracle, Π^{p1}.
- **Send** (Π^{p1}, MG): This query is effectuated to establish an active attack. Additionally, Π^{p1} can transmit a message, MG, to Π^{p1} and acquire an affiliated response.
- **CorruptSD** (Π^{p1}): This query is effectuated by $\mathcal{A}$ to acquire the long-term parameters held in the memory of SD_i.

Theorem 1. *We consider that $\mathcal{A}$ is a polynomial-time (polt) adversary endeavoring to crack the security of the session key established between the user and SED_j in ESCI-AKA. Hence, the advantage of $\mathcal{A}$ in successfully cracking the security of the session key can be derived as follows:*

$$Adv_{\mathcal{A}}^{ESCI-AKA}(polt) \leq \frac{H_q^2}{|HSP|} + \frac{S_q}{2^{bkl-1} \cdot |PSP|} + 2 \cdot Adv_{AEAD}^{OCCA3}. \tag{14}$$

In the above equation, H_q^2, S_q, $|PSP|$, and 2^{bkl}, $|HSP|$represent the number of queries for the hash function, send queries, password space, biometric key space/length, and hash function space, respectively. Furthermore, $Adv_{\mathcal{A}}^{OCCA3}(polt)$ [55] denotes the advantage of $\mathcal{A}$ in breaking the security of ASCON within a polynomial-time constraint.

Proof. The proof of Theorem 1 is conducted in the same way as in [54,57,58]. Here, we consider the four games $(G_q|q = 0, 1, 2, 3)$, where the winning probability of $\mathcal{A}$ to determine

the correct bit "b" is denoted by Adv^{G}. All the games are explicated in the following sections in detail.

*G*0: This is considered an initial attack from $\mathcal{A}$ against ESCI-AKA in the ROM model. As "b" must be decided before *G*0, it is obvious that

$$Adv_{\mathcal{A}}^{ESCI-AKA}(polt) = |2 \cdot Adv^{G0} - 1|. \tag{15}$$

*G*1: This game enables $\mathcal{A}$ to perform $Execute(\Pi_{U_i}^{p1}, \Pi_{GW_k}^{p2}, \Pi_{SED_j}^{p3})$. By using this query, $\mathcal{A}$ can capture all the messages, such as MG_1: $\{TiS_1, PID_i, C_4, C_5, MAC_2\}$, MG_2: $\{TiS_2, C_6, C_7, C_8, MAC_4\}$, and MG_3: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$, exchanged between network participants, such as SD_i, GW_k, and SED_j, respectively. Now, the primary objective of $\mathcal{A}$ is to construct the session key $SK_{SED_j} (= SK_{U_i}) = H(RNU_6 \parallel (C_3 \oplus RNU_3) \parallel (SID_{SED_j} \oplus RNU_6 \oplus RNU_5) \parallel TiS_3)$, which is constructed during the execution of the AKA phase. As the session key in ESCI-AKA is generated using a combination of long-term and ephemeral parameters, such as RNU_3, RNU_4, RNU_5, RNU_6, C_3, GID_k, C_1, and C_2, the *Reveal* query is employed at the end of *G*1 to reveal the session key, and the *Test* query is used to verify whether the constructed session key is a valid output or a random one. However, the probability of $\mathcal{A}$ winning without knowledge of RNU_3, RNU_4, RNU_5, RNU_6, C_3, GID_k, C_1, and C_2 is extremely low. Therefore, *G*0 and *G*1 can be considered equivalent. Thus, we can conclude that

$$Adv^{G1} = Adv^{G0} \tag{16}$$

*G*2: An active attack is conducted by executing H_q^2 queries. In ESCI-AKA, the hash function generates the session key (SK) for U_i and SED_j. $\mathcal{A}$ attempts to find a collision by constructing HSP queries to compromise the security of the SK. However, the likelihood of a collision is minimal, as indicated by the birthday paradox.

$$Adv^{G2} - Adv^{G1} \leq \frac{H_q^2}{2|HSP|}. \tag{17}$$

*G*3: In this scenario, $\mathcal{A}$ captures the SD_i of a user and extracts sensitive parameters, including $\{RNU_2, AD_1, Q_1, MAC_1, hd, GID_k, FE.Gen(\cdot), FE.Rep(\cdot)\}$, from the memory of SD_i. To accomplish this, $\mathcal{A}$ executes $CorruptSD(\Pi^{p1})$. The objective of $\mathcal{A}$ is to modify or change the user's password and biometrics. However, generating or guessing biometric keys is challenging, and the length of the biometric key, denoted as *bkl*, makes the probability of guessing the biometric key ($\frac{1}{2^{bkl}}$) negligible. Furthermore, considering the limited number of permissible incorrect password attempts, we can deduce that the probability of successfully guessing or modifying the user's password and biometrics within the allowed number of attempts is extremely low. Therefore, the security of ESCI-AKA is maintained, as it effectively protects against unauthorized access to and manipulation of user credentials. Hence, we have the following:

$$Adv^{G3} - Adv^{G2} \leq \frac{S_q}{2^{bkl} \cdot |PSP|}. \tag{18}$$

*G*4: In this game, $\mathcal{A}$ conducts a real attack by capturing all the exchanged messages, including MG_1, MG_2, and MG_3, using the $Execute(\Pi_{U_i}^{p1}, \Pi_{GW_k}^{p2}, \Pi_{SEDj}^{p3})$ query. It is important to note that all the messages transmitted during the AKA phase are encrypted using the ASCON encryption algorithm. Based on the security definition of ASCON (refer to Definition 1), ASCON is deemed safe for usage. Consequently, the advantage of $\mathcal{A}$ in cracking the security of the AEAD scheme is nominal. Thus, we can extrapolate that

$$Adv^{G4} - Adv^{G3} \leq Adv_{AEAD,\mathcal{A}}^{OCCA3}(polt). \tag{19}$$

$\mathcal{A}$, after finishing games, such as $(G_q|q \in [0,3])$, receives no considerable advantage to gain the correct bit "b". Thus, we arrive at

$$Adv^{G4} = 1/2 \tag{20}$$

From (15) and (16), we obtain

$$Adv_{\mathcal{A}}^{ESCI-AKA}(polt) = |2 \cdot Adv^{G0} - \frac{1}{2}|. \tag{21}$$

From (21), we obtain

$$\frac{1}{2} \cdot Adv_{\mathcal{A}}^{ESCI-AKA}(polt) = |Adv^{G0} - Adv^{G4}|. \tag{22}$$

By using (20) and (22), we obtain

$$\frac{1}{2} \cdot Adv_{\mathcal{A}}^{ESCI-AKA}(polt) = |Adv^{G1} - Adv^{G4}| \tag{23}$$

Upon considering the triangular inequality, we have

$$\begin{aligned} |Adv^{G1} - Adv^{G4}| &\leq |Adv^{G1} - Adv^{G2}| \\ &+|Adv^{G2} - Adv^{G4}| \\ \leq |Adv^{G1} - Adv^{G2}| &+ |Adv^{G2} - Adv^{G3}| \\ &+|Adv^{G3} - Adv^{G4}|. \end{aligned} \tag{24}$$

By using (17), (19), and (24), we obtain

$$\begin{aligned} Adv_{\mathcal{A}}^{ESCI-AKA}(polt) \leq \frac{H_q^2}{|HSP|} + \frac{S_q}{2^{bkl-1} \cdot |PSP|} \\ +2.Adv_{AEAD,\mathcal{A}}^{OCCA3}(polt). \end{aligned} \tag{25}$$

□

5.3. Formal Validation Using Scyther

Scyther is employed to analyze potential vulnerabilities in security frameworks. Developed in Python, Scyther ensures that all cryptographic operations/functions are impenetrable. This implies that unless an attacker manages to seize the decryption key, the encrypted transmission remains incomprehensible to them. In this article, we employ Scyther to examine the security characteristics of ESCI-AKA. ESCI-AKA is implemented using Security Protocol Description Language (SPDL), which defines three roles: (i) GWK (gateway role), (ii) $SEDJ$ (smart embedded device role), and (iii) RMU/SDI (user role). The SPDL script contains both manually established claims and automatically generated ones, all of which are verified by Scyther, as depicted in Figure 3. Consequently, we can affirm that ESCI-AKA is well protected and secure, as illustrated in Figure 3.

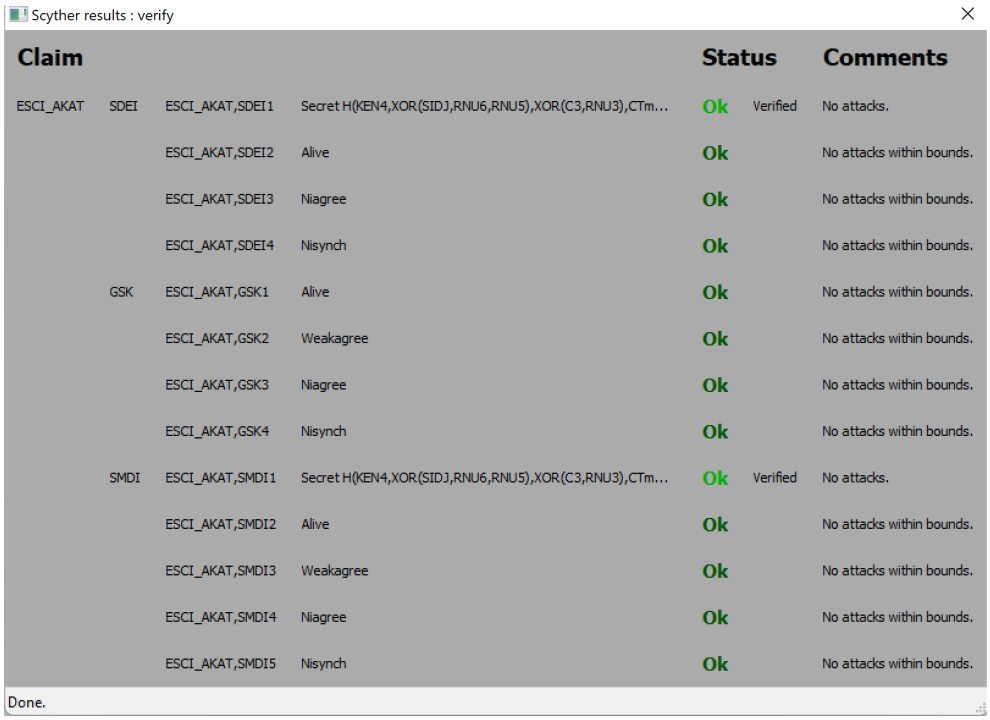

Figure 3. Results generated through Scyther.

6. Performance Comparison

The proposed ESCI-AKA was compared with several references, including [10–16], in terms of their security characteristics and communication and computational costs. To evaluate the computational performance of various cryptographic primitives, we conducted tests on "Raspberry-Pi-3 with a CPU clocked at 1.2 GHz and 1 GB of RAM", running the Ubuntu operating system. Each cryptographic primitive was executed 100 times, and the average time taken by each primitive is presented in Table 3.

Table 3. Execution Time.

Cryptographic Function	Notation	Raspberry Pi-3	Size in Bits
ECC scalar multiplication	T_{ECC}	3.47 ms	ECC size (320 bits)
Encryption/decryption	T_{ENC}	0.664 ms	ID size (128 bits)
Hash function (SHA-256)	T_{HF}	0.382 ms	HASH output size (256 bits)
ASCON encryption/decryption	T_{AEAD}	0.401 ms	MAC size (128 bits)
Biometric key generation	$T_B \approx T_{ECC}$	3.47 ms	Timestamps (32 bits)

6.1. Security Comparison

In this subsection, we correlate the security characteristics of ESCI-AKA with other references: [10–16]. The security mechanism offered in [10] lacks protection against impersonation and device capture attacks, and it does not guarantee mutual authentication. The authentication mechanism offered in [11] is defenseless against password guessing, impersonation, and MITM attacks. Moreover, it is unable to accomplish mutual authentication and achieve user anonymity. The authentication mechanism offered in [12] does not guarantee protection against privileged insider, user anonymity, stolen smart card, and password guessing attacks. The security mechanism offered in [13] is ineffective against password guessing and temporary secret leakage, and it also lacks anonymity and untraceability characteristics. The scheme offered in [14] is defenseless against privileged insider, impersonation, replay, stolen smart card, identity guessing, and password guessing attacks. The security protocol proposed in [16] is unable to resist privileged insider and temporary parameter leakage attacks. In contrast, the proposed ESCI-AKA is protected against diverse security attacks. A comparison of the security characteristics is represented in Table 4.

Table 4. Security Properties Comparison.

Features/Attacks	[10]	[11]	[12]	[13]	[14]	[15]	[16]	ESCI-AKA
"Anonymity/Untraceability"	✓	×	×	×	✓	✓	✓	✓
"Password Guessing Attack"	✓	×	×	×	×	✓	✓	✓
"Impersonation Attack"	×	×	✓	✓	×	✓	✓	✓
"MITM"	✓	×	✓	✓	✓	✓	✓	✓
"TSL Attack"	✓	✓	✓	×	✓	✓	×	✓
"Replay Attack"	✓	✓	✓	✓	×	✓	✓	✓
"SSC Attack"	✓	✓	✓	✓	✓	✓	✓	✓
"Identity Guessing"	✓	✓	✓	✓	×	✓	✓	✓
"Desynchronization"	✓	✓	✓	✓	✓	✓	✓	✓

Note: SSC: stolen smart card; TSL: temporary secret leakage; ✓ denotes the availability of features; × indicates the feature not available.

6.2. Memory Cost

In the proposed scheme, the memory requirements at different nodes are as follows:

- Smart device node: SID_{SED_j}, DSK = {128 + 256} = 384 bits;
- Gateway node: 2 × (PID_i RNU) = {256 × 2} = 512 bits;
- User device: {RNU_2^*, GID_k^*, AD_1, Q_1, MAC_1, hd, $FE.Gen(\cdot)$, $FE.Rep(\cdot)$} = {128 + 128 + 128 + 256 + 160} = 800 bits. In the proposed scheme, the total memory cost required is 1696 bits, while the works referenced as [10–16] require 1888 bits, 2048 bits, 1928 bits, 1024 bits, 992 bits, 1024 bits, and 992 bits, respectively. The proposed scheme requires more memory compared to certain security schemes but still demands less memory compared to other relevant security schemes. The proposed scheme, in comparison to related security schemes, incurs lower computational and communication costs while offering more significant security features.

6.3. Computational Cost

The computational cost of the proposed ESCI-AKA and other relevant security frameworks are computed using the computational time given in Table 3. The aggregated computational cost of ESCI-AKA is 8.569 ms, which is 65.71%, 66.18%, 84.87%, 66.18%, 79.65%, 61.01%, and 86.27% less than the security frameworks presented in [10–16], respectively. In addition, a comparison of the total computational cost is given in Figure 4, which indicates that the proposed ESCI-AKA requires low computational time to accomplish the AKA phase. The computational costs at U_i, GW_k, and SED_j are 6.201 ms, 1.184 ms, and 1.184 ms, respectively. A comparison of computational cost at U_i, GW_k, and SED_j is provided in Figure 5, which shows that ESCI-AKA requires low computation resources at U_i, GW_k, and SED_j.

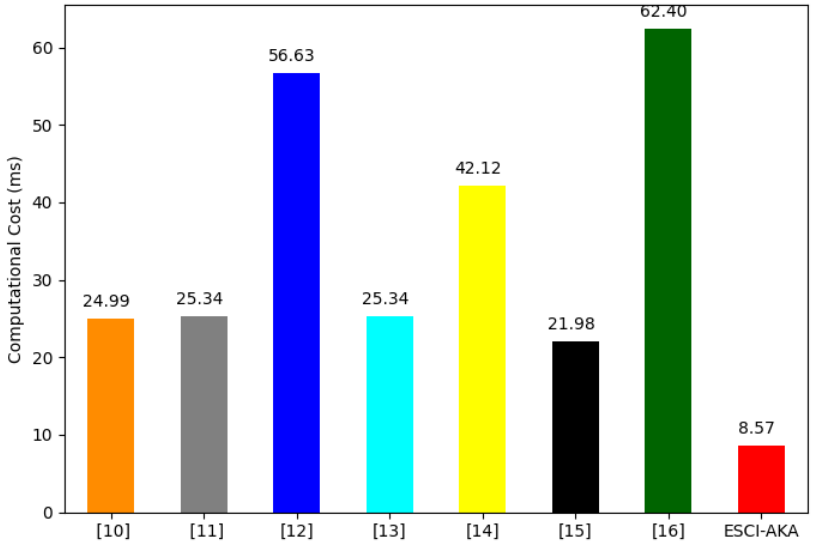

Figure 4. Computation cost to complete the secret channel establishment phase.

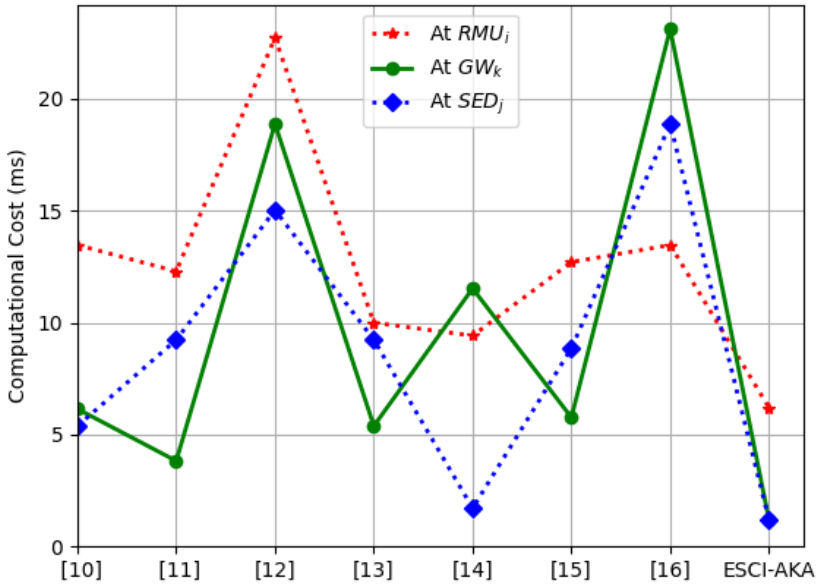

Figure 5. Computational cost at U_i, GW_k, and SED_j side during the accomplishment of the AKA phase.

In addition, the proposed ESCI-AKA requires fewer computational resources when many users are sending the security channel establishment or authentication request to GW_k. A comparison of computation cost when increasing the number of users at GW_k is provided in Figure 6 and Table 5.

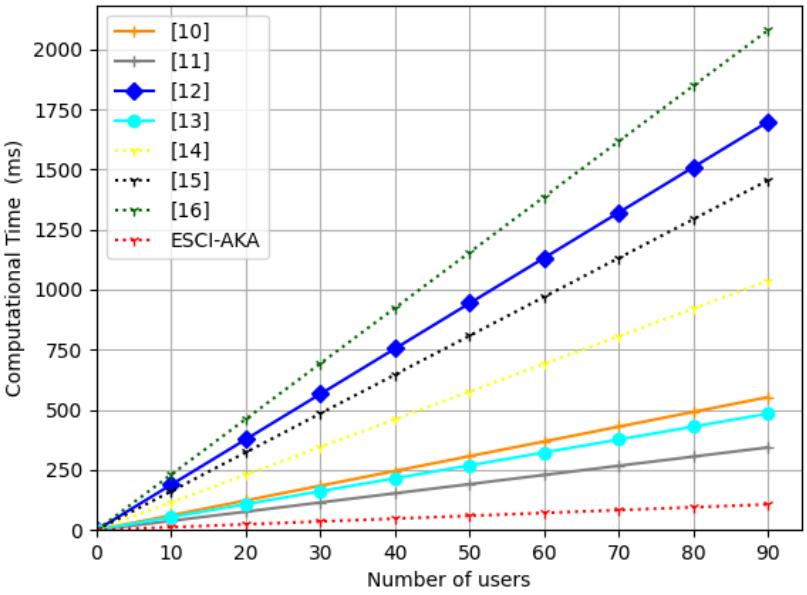

Figure 6. Computational cost at server increasing the number of U_i.

Table 5. Computational Cost.

Scheme	U_i Side	GW_k/TA Side	SED_j Side	Total Time (ms)
[10]	$8T_{HF} + 3T_{ECC} \approx$ **13.46**	$7T_{HF} + T_{ECC} \approx$ **6.144**	$5T_{HF} + T_{ECC} \approx$ **5.38** ms	$20T_{HF} + 5T_{ECC} \approx$ **24.99** ms
[11]	$14T_{HF} + 2T_{ECC} \approx$ **12.288**	$10T_{HF} \approx$ **3.82**	$6T_{HF} + 2T_{ECC} \approx$ **9.232** ms	$30T_{HF} + 4T_{ECC} \approx$ **25.34** ms
[12]	$5T_{HF} + 5T_{ECC} + T_B \approx$ **22.73**	$4T_{HF} + 5T_{ECC} \approx$ **18.87**	$3T_{HF} + 4T_{ECC} \approx$ **15.026** ms	$12T_{HF} + 14T_{ECC} + T_B \approx$ **56.634** ms
[13]	$8T_{HF} + 2T_{ECC} \approx$ **9.96**	$5T_{HF} + T_{ECC} \approx$ **5.38**	$6T_{HF} + 2T_{ECC} \approx$ **9.232** ms	$30T_{HF} + 4T_{ECC} \approx$ **25.34** ms
[14]	$3T_{HF} + 2T_{ECC} + 2T_{ENC} \approx$ **9.41**	$12T_{HF} + 2T_{ECC} \approx$ **1.71**	$T_{HF} + 2T_{ENC} \approx$ **19.24** ms	$16T_{HF} + 3T_{ENC} + 4T_{ECC} \approx$ **42.11** ms
[15]	$6T_{HF} + 3T_{ECC} \approx$ **12.70**	$6T_{HF} + T_{ECC} \approx$ **5.76**	$5T_{HF} + 2T_{ECC} \approx$ **8.85** ms	$17T_{HF} + 6T_{ECC} \approx$ **21.98** ms
[16]	$8T_{HF} + 3T_{ECC} \approx$ **13.46**	$6T_{HF} + 6T_{ECC} \approx$ **23.11**	$4T_{HF} + 5T_{ECC} \approx$ **18.87** ms	$18T_{HF} + 16T_{ECC} \approx$ **62.39** ms
ESCI-AKA	$4T_{HF} + 3T_{AEAD} + T_B \approx$ **6.201**	$T_{HF} + 2T_{AEAD} \approx$ **1.184**	$T_{HF} + 2T_{AEAD} \approx$ **1.184** ms	$6T_{HF} + 7T_{AEAD} + T_B \approx$ **8.569** ms

6.4. Communication Cost

To estimate the communication cost of the proposed ESCI-AKA during the AKA phase or authentication phase, we consider the sizes of the different parameters given in Table 3. In ESCI-AKA, three messages are exchanged among various network entities: MG_1: $\{TiS_1, PID_i, C_4, C_5, MAC_2\}$, MG_2: $\{TiS_2, C_6, C_7, C_8, MAC_4\}$, and MG_3: $\{TiS_3, C_9, C_{10}, AD_6, MAC_6\}$. The sizes of MG_1, MG_2, and MG_3 are 544 bits, 544 bits, and 416 bits, respectively. ESCI-AKA requires 544 + 544 + 416 = 1540 bits to complete the AKA phase. In comparison, the communication costs required by [10–16] (shown in Figure 7 and Table 6) are 2720 bits, 3550 bits, 2528 bits, 3552 bits, 3290 bits, 2880 bits, and 4416 bits, respectively. Figure 7 and Table 6 provide a comparison of the communication costs between ESCI-AKA and other related security mechanisms.

Table 6. Communication Cost.

Frameworks	Factor	Communicated Messages
[10]	3F	2720 bits
[11]	2F	3552 bits
[12]	3F	2528 bits
[13]	2F	3552 bits
[14]	3F	3290 bits
[15]	2F	2880 bits
[16]	3F	4416 bits
ESCI-AKA	3F	1504 bits

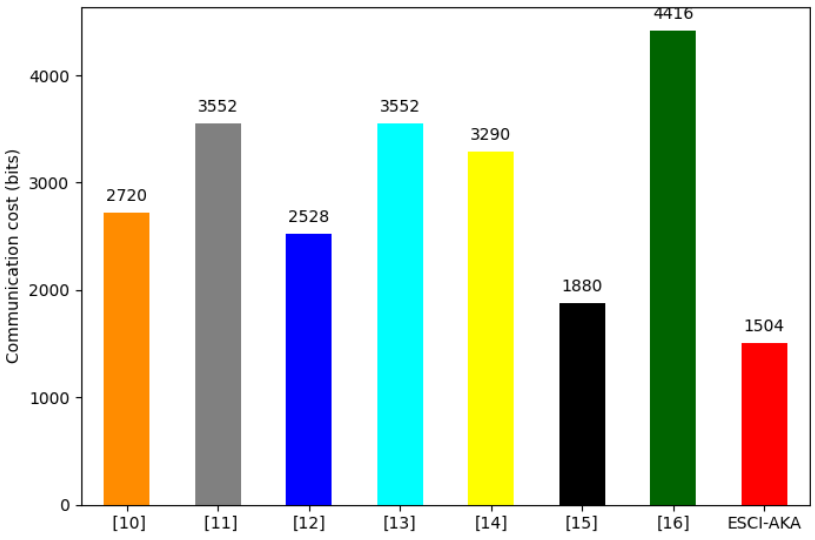

Figure 7. Communication cost required to accomplish the AKA phase.

7. Conclusions

In this paper, we introduced ESCI-AKA, an innovative secure authentication framework designed specifically for the smart home environment. The primary goal of ESCI-AKA is to establish a secure channel between a user's device and the smart home, ensuring secure communication over the public Internet. To achieve optimal resource efficiency, ESCI-AKA makes use of the lightweight cryptographic authenticated encryption scheme called "ASCON" and incorporates a hash function. The security of the session key established by ESCI-AKA is verified through a thorough ROM-based analysis. Furthermore, extensive informal analysis has demonstrated ESCI-AKA's robustness against various types of attacks, including replay attacks, MITM attacks, and desynchronization attacks. This ensures that the communication between the device and the smart home remains secure even in the presence of potential threats. The security claims of ESCI-AKA are further supported by a Scyther-based implementation, which reinforces its reliability in real-world scenarios. This implementation has proven its effectiveness in practical applications, en-

hancing the overall security of the framework. Moreover, in performance evaluations, ESCI-AKA has shown significant improvements in computational and communication costs. The framework achieves a significant computational cost reduction, ranging from 61.01% to 86.27%, and a communication cost reduction, ranging from 40.51% to 65.94%. These improvements not only enhance the efficiency of the system but also ensure that the security features remain intact. In the future, we plan to utilize lightweight cryptographic primitives, such as ASCON and Esch256, to design data sharing and access control mechanisms using blockchain technology.

Author Contributions: Conceptualization, H.A. and M.T.; Methodology, H.A.; Software, H.A.; Formal analysis, H.A. and M.T.; Investigation, H.A.; Data curation, H.A.; Writing—original draft, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Deanship of Scientific Research at King Khalid University through Large Group Research Project under grant number RGP.2/312/44.

Data Availability Statement: This study did not utilize any external datasets in its analysis.

Acknowledgments: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through the large group research project under grant number RGP2/312/44.

Conflicts of Interest: The authors state that there are no conflict of interest to disclose.

References

1. Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Comput. Netw.* **2019**, *148*, 283–294.
2. Kaur, B.; Dadkhah, S.; Shoeleh, F.; Neto, E.C.P.; Xiong, P.; Iqbal, S.; Lamontagne, P.; Ray, S.; Ghorbani, A.A. Internet of things (IoT) security dataset evolution: Challenges and future directions. *Internet Things* **2023**, *22*, 100780. [CrossRef]
3. Toh, C. Security for Smart Cities. *IET Smart Cities* **2020**, *2*, 95–104. [CrossRef]
4. Fabrègue, B.F.; Bogoni, A. Privacy and Security Concerns in the Smart City. *Smart Cities* **2023**, *6*, 586–613. [CrossRef]
5. Ragothaman, K.; Wang, Y.; Rimal, B.; Lawrence, M. Access control for IoT: A survey of existing research, dynamic policies and future directions. *Sensors* **2023**, *23*, 1805. [CrossRef]
6. Dobraunig, C.; Eichlseder, M.; Mendel, F.; Schläffer, M. Ascon v1.2: Lightweight authenticated encryption and hashing. *J. Cryptol.* **2021**, *34*, 33. [CrossRef]
7. Wu, H.; Preneel, B. AEGIS: A fast authenticated encryption algorithm. In Proceedings of the Selected Areas in Cryptography–SAC 2013: 20th International Conference, Burnaby, BC, Canada, 14–16 August 2013; Revised Selected Papers 20; Springer: Berlin/Heidelberg, Germany , 2014; pp. 185–201.
8. Aagaard, M.; AlTawy, R.; Gong, G.; Mandal, K.; Rohit, R. ACE: An authenticated encryption and hash algorithm. *LWC* **2019**, *in submission*.
9. Tanveer, M.; Bhutta, M.N.M.; Alzahrani, B.A.; Albeshri, A.; Alsubhi, K.; Chaudhry, S.A. CMAP-IoT: Chaotic Map-Based Authentication Protocol for Crowdsourcing Internet of Things. *Arab. J. Sci. Eng.* **2023**, 1–14. [CrossRef]
10. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, M.K.; Chen, C. A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems. *IEEE Syst. J.* **2019**, *14*, 39–50. [CrossRef]
11. Yuanbing, W.; Wanrong, L.; Bin, L. An Improved Authentication Protocol for Smart Healthcare System Using Wireless Medical Sensor Network. *IEEE Access* **2021**, *9*, 105101–105117. [CrossRef]
12. Challa, S.; Wazid, M.; Das, A.K.; Kumar, N.; Reddy, A.G.; Yoon, E.J.; Yoo, K.Y. Secure signature-based authenticated key establishment scheme for future IoT applications. *IEEE Access* **2017**, *5*, 3028–3043. [CrossRef]
13. Choi, Y.; Lee, D.; Kim, J.; Jung, J.; Nam, J.; Won, D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2014**, *14*, 10081–10106. [CrossRef]
14. Butt, T.M.; Riaz, R.; Chakraborty, C.; Rizvi, S.S.; Paul, A. Cogent and energy efficient authentication protocol for wsn in iot. *Comput. Mater. Contin.* **2021**, *68*, 1877–1898.
15. Zou, S.; Cao, Q.; Wang, C.; Huang, Z.; Xu, G. A Robust Two-Factor User Authentication Scheme-Based ECC for Smart Home in IoT. *IEEE Syst. J.* **2022**, *16*, 4938–4949. [CrossRef]
16. Sureshkumar, V.; Amin, R.; Vijaykumar, V.; Sekar, S.R. Robust secure communication protocol for smart healthcare system with FPGA implementation. *Future Gener. Comput. Syst.* **2019**, *100*, 938–951. [CrossRef]
17. Liu, Y.; Wang, J.; Yan, Z.; Wan, Z.; Jäntti, R. A Survey on Blockchain-based Trust Management for Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 5898–5922. [CrossRef]
18. Corallo, A.; Lazoi, M.; Lezzi, M.; Luperto, A. Cybersecurity awareness in the context of the Industrial Internet of Things: A systematic literature review. *Comput. Ind.* **2022**, *137*, 103614. [CrossRef]

19. Hussain, S.; Ullah, S.S.; Ali, I.; Xie, J.; Inukollu, V.N. Certificateless signature schemes in Industrial Internet of Things: A comparative survey. *Comput. Commun.* **2022**, *181*, 116–131. [CrossRef]
20. 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.* **2022**, *9*, 2578–2591. [CrossRef]
21. Tanveer, M.; Khan, A.U.; Kumar, N.; Hassan, M.M. RAMP-IoD: A Robust Authenticated Key Management Protocol for the Internet of Drones. *IEEE Internet Things J.* **2022**, *9*, 1339–1353. [CrossRef]
22. Srinivas, J.; Das, A.K.; Wazid, M.; Vasilakos, A.V. Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system. *IEEE Internet Things J.* **2020**, *8*, 7727–7744. [CrossRef]
23. Xu, H.; Hsu, C.; Harn, L.; Cui, J.; Zhao, Z.; Zhang, Z. Three-factor anonymous authentication and key agreement based on fuzzy biological extraction for Industrial Internet of Things. *IEEE Trans. Serv. Comput.* **2023**. [CrossRef]
24. Kwon, D.K.; Yu, S.J.; Lee, J.Y.; Son, S.H.; Park, Y.H. WSN-SLAP: Secure and lightweight mutual authentication protocol for wireless sensor networks. *Sensors* **2021**, *21*, 936. [CrossRef] [PubMed]
25. Kumar, D. Cryptanalysis and improvement of an authentication protocol for wireless sensor networks. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4747.
26. Ali, R.; Pal, A.K.; Kumari, S.; Sangaiah, A.K.; Li, X.; Wu, F. An enhanced three factor based authentication protocol using wireless medical sensor networks for healthcare monitoring. *J. Ambient. Intell. Humaniz. Comput.* **2018**, 1–22. [CrossRef]
27. Masud, M.; Gaba, G.S.; Choudhary, K.; Hossain, M.S.; Alhamid, M.F.; Muhammad, G. Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare. *IEEE Internet Things J.* **2021**, *9*, 2649–2656. [CrossRef]
28. Tanveer, M.; Alkhayyat, A.; Khan, A.U.; Kumar, N.; Alharbi, A.G. REAP-IIoT: Resource-Efficient Authentication Protocol for the Industrial Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 24453–24465. [CrossRef]
29. Ashrif, F.F.; Sundarajan, E.A.; Ahmed, R.; Hasan, M.K. SLAE6: Secure and Lightweight Authenticated Encryption Scheme for 6LoWPAN Networks. In Proceedings of the 12th International Conference on Sensor Networks-SENSORNETS, Online, 23–24 February 2023.
30. Fotouhi, M.; Bayat, M.; Das, A.K.; Far, H.A.N.; Pournaghi, S.M.; Doostari, M.A. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Comput. Netw.* **2020**, *177*, 107333. [CrossRef]
31. Fan, Q.; Chen, J.; Shojafar, M.; Kumari, S.; He, D. SAKE*: A Symmetric Authenticated Key Exchange Protocol with Perfect Forward Secrecy for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6424–6434. [CrossRef]
32. Hu, H.; Liao, L.; Zhao, J. Secure Authentication and Key Agreement Protocol for Cloud-Assisted Industrial Internet of Things. *Electronics* **2022**, *11*, 1652. [CrossRef]
33. Tanveer, M.; Alkhayyat, A.; Chaudhry, S.A.; Zikria, Y.B.; Kim, S.W. REAS-TMIS: Resource-efficient authentication scheme for telecare medical information system. *IEEE Access* **2022**, *10*, 23008–23021. [CrossRef]
34. Tanveer, M.; Alasmary, H. LACP-SG: Lightweight Authentication Protocol for Smart Grids. *Sensors* **2023**, *23*, 2309. [CrossRef] [PubMed]
35. Liu, Z.; Guo, J.; Huang, F.; Cai, D.; Wu, Y.; Chen, X.; Igorevich, K.K. Lightweight trustworthy message exchange in unmanned aerial vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 2144–2157. [CrossRef]
36. Guo, J.; Xiong, L.; Li, J.; Liu, J.; Tian, S.; Li, H. An incentive mechanism for horizontal federated learning based on principle of compound interest. *Phys. Commun.* **2023**, *60*, 102128. [CrossRef]
37. Meshram, C.; Obaidat, M.S.; Lee, C.C.; Meshram, S.G. An Efficient, Robust, and Lightweight Subtree-Based Three-Factor Authentication Procedure for Large-Scale DWSN in Random Oracle. *IEEE Syst. J.* **2021**, *15*, 4927–4938. [CrossRef]
38. Li, Y.; Tian, Y. A Lightweight and Secure Three-Factor Authentication Protocol with Adaptive Privacy-Preserving Property for Wireless Sensor Networks. *IEEE Syst. J.* **2022**, *16*, 6197–6208. [CrossRef]
39. Wu, F.; Li, X.; Xu, L.; Vijayakumar, P.; Kumar, N. A novel three-factor authentication protocol for wireless sensor networks with IoT notion. *IEEE Syst. J.* **2020**, *15*, 1120–1129. [CrossRef]
40. 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]
41. Zhang, M.; Xu, C.; Li, S.; Jiang, C. On the Security of an ECC-Based Authentication Scheme for Internet of Drones. *IEEE Syst. J.* **2022**, *16*, 6425–6428. [CrossRef]
42. Aman, M.N.; Basheer, M.H.; Sikdar, B. A Lightweight Protocol for Secure Data Provenance in the Internet of Things Using Wireless Fingerprints. *IEEE Syst. J.* **2021**, *15*, 2948–2958. [CrossRef]
43. Aminian Modarres, A.M.; Sarbishaei, G. An Improved Lightweight Two-Factor Authentication Protocol for IoT Applications. *IEEE Trans. Ind. Inform.* **2022**, 19, 6588–6598. [CrossRef]
44. Lee, J.; Oh, J.; Park, Y. A Secure and Anonymous Authentication Protocol Based on Three-Factor Wireless Medical Sensor Networks. *Electronics* **2023**, *12*, 1368. [CrossRef]
45. Kwon, D.; Park, Y.; Park, Y. Provably secure three-factor-based mutual authentication scheme with PUF for wireless medical sensor networks. *Sensors* **2021**, *21*, 6039. [CrossRef] [PubMed]
46. Ding, Z.; Xie, Q. Provably Secure Dynamic Anonymous Authentication Protocol for Wireless Sensor Networks in Internet of Things. *Sustainability* **2023**, *15*, 5734. [CrossRef]
47. Rangwani, D.; Om, H. A secure user authentication protocol based on ECC for cloud computing environment. *Arab. J. Sci. Eng.* **2021**, *46*, 3865–3888. [CrossRef]

48. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V. Design of secure key management and user authentication scheme for fog computing services. *Future Gener. Comput. Syst.* **2019**, *91*, 475–492. [CrossRef]
49. Yeh, H.L.; Chen, T.H.; Liu, P.C.; Kim, T.H.; Wei, H.W. A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors* **2011**, *11*, 4767–4779. [CrossRef]
50. Cho, Y.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y. A Secure and Anonymous User Authentication Scheme for IoT-Enabled Smart Home Environments Using PUF. *IEEE Access* **2022**, *10*, 101330–101346. [CrossRef]
51. Kaveh, M.; Mosavi, M.R. A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function. *IEEE Syst. J.* **2020**, *14*, 4535–4544. [CrossRef]
52. Safkhani, M.; Bagheri, N.; Ali, S.; Hussain Malik, M.; Hassan Ahmed, O.; Hosseinzadeh, M.; Mosavi, A.H. Improvement and Cryptanalysis of a Physically Unclonable Functions Based Authentication Scheme for Smart Grids. *Mathematics* **2022**, *11*, 48. [CrossRef]
53. Tanveer, M.; Ahmad, M.; Nguyen, T.N.; Abd El-Latif, A.A. Resource-Efficient Authenticated Data Sharing Mechanism for Smart Wearable Systems. *IEEE Trans. Netw. Sci. Eng.* **2022**. [CrossRef]
54. Tanveer, M.; Ahmad, M.; Khalifa, H.S.; Alkhayyat, A.; Abd El-Latif, A.A. A new anonymous authentication framework for secure smart grids applications. *J. Inf. Secur. Appl.* **2022**, *71*, 103336. [CrossRef]
55. Abed, F.; Forler, C.; Lucks, S. General classification of the authenticated encryption schemes for the CAESAR competition. *Comput. Sci. Rev.* **2016**, *22*, 13–26. [CrossRef]
56. Tanveer, M.; Bashir, A.K.; Alzahrani, B.A.; Albeshrir, A.; Alsubhi, K.; Chaudhry, S.A. CADF-CSE: Chaotic map-based authenticated data access/sharing framework for IoT-enabled cloud storage environment. *Phys. Commun.* **2023**, *59*, 102087. [CrossRef]
57. Tanveer, M.; Alkhayyat, A.; Naushad, A.; Khan, A.U.; Kumar, N.; Alharbi, A.G. RUAM-IoD: A Robust User Authentication Mechanism for the Internet of Drones. *IEEE Access* **2022**, *10*, 19836–19851. [CrossRef]
58. Tanveer, M.; Khan, A.U.; Kumar, N.; Naushad, A.; Chaudhry, S.A. A Robust Access Control Protocol for the Smart Grid Systems. *IEEE Internet Things J.* **2022**, *9*, 6855–6865. [CrossRef]