sensors

MDPI

Article

LACP-SG: Lightweight Authentication Protocol for Smart Grids

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

[1] Department of Computer Science, University of Management and Technology, Lahore 54770, Pakistan
[2] Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
* Correspondence: tanveer123giki@gmail.com

Abstract: Smart grid (SG) recently acquired considerable attention due to their utilization in sustaining demand response management in power systems. Smart meters (SMs) deployed in SG systems collect and transmit data to the server. Since all communications between SM and the server occur through a public communication channel, the transmitted data are exposed to adversary attacks. Therefore, security and privacy are essential requirements in the SG system for ensuring reliable communication. Additionally, an AuthentiCation (AC) protocol designed for secure communication should be lightweight so it can be applied in a resource-constrained environment. In this article, we devise a lightweight AC protocol for SG named LACP-SG. LACP-SG employs the hash function, "Esch256", and "authenticated encryption" to accomplish the AC phase. The proposed LACP-SG assures secure data exchange between SM and server by validating the authenticity of SM. For encrypted communication, LACP-SG enables SM and the server to establish a session key (SEK). We use the random oracle model to substantiate the security of the established SEK. Moreover, we ascertain that LACP-SG is guarded against different security vulnerabilities through Scyther-based security validation and informal security analysis. Furthermore, comparing LACP-SG with other related AC protocols demonstrates that LACP-SG is less resource-intensive while rendering better security characteristics.

Keywords: authentication; smart grid; AEAD; privacy; session key; ROM

1. Introduction

The Industrial Internet of Things (IIoTs) promises to elevate many communication paradigm innovations, focusing on industrial applications. Particularly, IIoT-based smart grid (SG) technology is envisioned to be a vital part of the next-generation power grid system. An SG mainly comprises four elements: sensing, control, actuation, and communication systems. The sensing and communication processes are performed by smart meters (SMs), which are the significant components of an SG, while service providers perform actuation and communication (SPs) [1].

The rapid utilization of SMs has recently been witnessed in smart homes under the SG environment to observe energy utilization in real time. To this end, the SMs communicate with SP on public communication channels. The communication between SMs and SP mandates security and privacy, as the channel used for this communication is prone to various security risks. For instance, an adversary can modify, eavesdrop, and disrupt the communication with consequent degradation in the performance of the SG system [2]. These concerns necessitate the designing of a secure, lightweight, and robust authentication (AC) protocol to guarantee information communication among the honest participants in the SG system while preserving the privacy of the entities.

Citation: Tanveer, M.; Alasmary, H. LACP-SG: Lightweight Authentication Protocol for Smart Grids. *Sensors* **2023**, *23*, 2309. https://doi.org/10.3390/s23042309

Received: 19 January 2023
Revised: 2 February 2023
Accepted: 16 February 2023
Published: 19 February 2023

1.1. Security Requirements in SG Systems

An SM transmits electricity usage information periodically to SP via the public internet. Therefore, the following security requirements are imperative for the smooth working of the SG system [3,4].

1.1.1. Security

Firstly, the SG system contains a large number of SMs. Thus, an SP must check the authenticity of the SM before commencing the information exchange process. It is worth noticing that, by authentication, the authenticity of the deployed SMs in the SG system can be verified. Therefore, the authentication protocol should be able to resist various security attacks, such as denial-of-service (DoS), SM capture, ephemeral secret leakage (EPSL), device impersonation (DIMP), man-in-the-middle (MIDM), de-synchronization (DeS), privilege-insider (PrI), replay, and SP impersonation (SPI) attacks [5]. After accomplishing the authentication process, SM and SP need to create a common session key (SEK) to protect the communicated information. Secondly, the authentication protocol needs to guarantee the authenticity of the SM and SP, verify the data's integrity, and ensure non-repudiation. Thirdly, by capturing an SM by an adversary, the procured sensitive information from the memory of the captured SM should not breach the security of the communication between other SMs and SP [6,7].

1.1.2. Efficiency

In general, an SP has sufficient computational resources and can process a specific volume of information. However, many SMs communicate with SP concurrently in the SG system, requiring significant computational resources. Moreover, SMs are resource-limited devices with limited computational, communication, and energy resources. Thus, it is imperative to devise a resource-efficient authentication protocol that requires the least computational resources of SP and SM during the authentication process [4,8].

2. Related Work

Security and privacy are the critical parameters of concern for the SG systems. Various security schemes have been proposed to cope with the security challenges in the SG system [9,10]. Li et al. [4] proposed an AC mechanism, which is in-efficacious in thwarting replay, MIDM, and EPSL attacks. In addition, the proposed scheme is incapable of rendering MA and anonymity features. Kumar et al. [11] proposed an AC mechanism for the SG environment employing elliptic curve cryptography (ECC) and SHA. However, the scheme of Kumar et al. is incapable of restraining MIDM device impersonation. DIMP and EPSL attacks are unable to ensure mutual authentication (MA) and the security of SEK. An authentication protocol for the SG environment is presented in [12], using PUF and SHA. Similarly, a secure communication protocol for the SG environment is presented in [13], which is unable to withstand DoS and EPSL attacks. An ECC, XOR, and SHA-based lightweight AC protocol for the SG environment is presented in [14], which cannot withstand various security attacks. An authentication and SEK establishment scheme is propounded in [15], utilizing ECC, XOR, and SHA. The authors in [16] propounded a reliable AC protocol using ECC for the SG infrastructure that can hinder different security threats. In this paper, we propose a physical unclonable function (PUF)-based AC mechanism for the SG system. Li et al. [4] devised a pairing-based message AC protocol for the SG environment, unable to withstand the MIDM, DoS, EPSL, and impersonation attacks and incapable of providing security for SEK. Chen et al. [3] propounded a BP-based AC protocol for SG environments, incapable of resisting EPSL and impersonation attacks and incapable of ensuring the security of SEK. The security framework proposed in [17] cannot resist the DeS attack. An AE-based security framework is presented in [18], and its security is proved through the AVISPA. A detailed summary of various AC protocols or schemes propounded for the SG environment is presented in Table 1.

Table 1. Summary of various AC protocols.

AC/AKE Protocol	Shortcomings/Security Vulnerabilities
Wu et al. [19]	Unable to thwart MIDM and EPSL attacks. Incapable of rendering anonymity and PFS features.
Mahmood et al. [20]	In-efficacious in preventing DoS, impersonation, PrI, replay, MIDM, and EPSL attacks.
Dariush et al. [21]	In-efficacious in resisting DoS attack. Incapable of rendering SM's anonymity and SEK security.
Banerjee et al. [22]	Unable to render identity protection and traceability.
Wazid et al. [23]	Exposed to DeS attack. Incapable of rendering revocability and formal validation.
Odelu et al. [24]	In-efficacious in preventing DoS, MIDM, and impersonation attacks. Unable to assure SM's anonymity.
Xie et al. [25]	In-efficacious in resisting replay and impersonation attacks. Incapable of rendering forward secrecy.
Li et al. [4]	In-efficacious in thwarting replay, MIDM, EPSL attacks. Incapable of rendering MA and anonymity features.
LACP-SG	Specialized hardware is required to accomplish the PUF-based AC process. In the future, we will use the AEAD schemes for designing the blockchain-enabled authentication frameworks.

Authenticated encryption with associative data (AEAD), lightweight cryptography (LWC), advance encryption standard (AES), mutual authentication (MA), perfect forward secrecy (PFS), exclusive-OR (XOR), bi-linear paring (BP), elliptic curve cryptography (ECC), authentication and key exchange (AKE), physical unclonable function (PUF), secure hash algorithm (SHA).

2.1. Motivation

Most of the AC protocols in the existing literature are devised using standardized symmetric encryption, such as AES, and public-key cryptography, such as ECC. These standardized cryptographic primitives are computationally expensive for resource-limited devices [14,26]. Moreover, most AC protocols are susceptible to various security risks, including DeS, replay, impersonation attacks, etc., as summarized in Section 2. Therefore, it is imperative to devise a secure and lightweight AC protocol for the SG systems.

Various AEAD schemes are devised to enable encryption and decryption services in resource-limited IoT devices. The main features of AEAD schemes are given to clarify why adopting the LWC primitives is essential when devising an AC protocol. This property of AEAD schemes makes them efficacious in reducing the encryption/decryption operations required to perform the AC process. (i) LWC-based AEAD schemes achieve message authenticity, integrity, and confidentiality simultaneously with a single encryption/decryption operation. (ii) AEAD schemes demand less computational and energy resources with reduced message overhead. (iii) The LWC-based hash function (Esch256) demands fewer computational resources than the existing hash functions while proffering the same security level.

Figure 1 presents the high-level working of an AEAD scheme, which is the base mechanism of the proposed AC protocol. Here, the AEAD scheme at the source node accepts the key along with associative data (AD), initialization vector/nonce, and plaintext as inputs to return output in the form of ciphertext (CT) and authentication parameters (AP). Moreover, the source generates a message with credentials $\{AD, CT, AP\}$ and sends this message to the destination to accomplish MA. In the proposed protocol, AD comprises the temporary identity of the source node, i.e., $AD = \{temporary\ identity, IP\ header, etc.\}$. SP uses the temporary identity to find the record associated with the source from its memory. CT is obtained after encrypting the random numbers and other parameters used in the construction of SEK. At the destination, decryption is performed by using the AEAD scheme. The AEAD scheme generates the PT and AP_d after taking the same input parameters as taken at the source node. To authenticate the validity of the obtained message, the destination node checks the condition $AP = AP_d$. If it holds, the received message is valid. We adopt the same methodology to propose a secure and lightweight AC protocol for the SG environment.

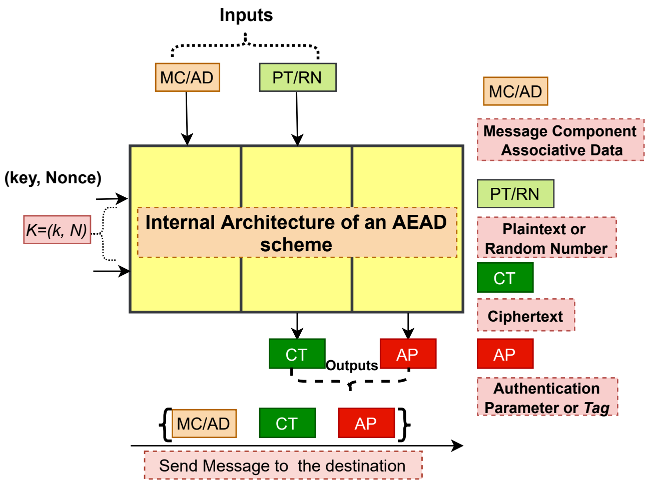

Figure 1. Message generation at source node using AEAD scheme.

2.2. Research Contributions

The paper comprises the subsequent contributions.

1. This paper proffers a new lightweight AC protocol for SGs, called LACP-SG, which utilizes "Counter Mode Encryption with authentication Tag" (COMET) [27] along with a lightweight hash function "Esch256". LACP-SG enables SP to check the authenticity of SM installed in the SG system before commencing the information exchange process. In addition, LACP-SG enables both the SM and SP to generate a shared SEK for future indecipherable communications.
2. The random oracle model (ROM) is utilized to corroborate the security of the established shared SEK. Moreover, security analysis utilizing the Scyther tool is executed to demonstrate that LACP-SG is resilient against MIDM, DeS, and replay attacks. Informal security is performed to illustrate that LACP-SG is resistant to SM capture and impersonation attacks. Moreover, LACP-SG allows the sensitive credentials associated with SM to be stored in ciphertext form in the database of SP, thereby restraining the PrI attack.
3. The meticulous comparative analysis is conducted to illustrate that LACP-SG renders enhanced security features while requiring low communication, storage, and computational overheads, respectively, than the related eminent AC protocols.

The subsequent paper is formed as follows. The system models, such as the network and attack model for LACP-SG, are illustrated in Section 3. Section 4 explicates the preliminary knowledge used in designing LACP-SG. The propounded LACP-SG is explicated in Section 5. The resiliency of LACP-SG against various attacks is furnished in Section 6. The significance of the LACP-SG is studied in Section 7. The paper concludes with concluding statements in Section 8.

3. System Model

3.1. Network Model

For the authentication process, we contemplate the SG network model as depicted in Figure 2, which constitutes registration authority (RA), smart meter $(SM_i|i = 1, 2, \cdots, n)$, where "$n$" symbolizes the installed SMs and $(SP_k|k = 1, 2, \cdots, N)$, where "$N$" symbolizes the number of SPs installed by RA. RA is liable for the registration of SP_k. SP_k stores the data or information sent by SM_i. SP_k pre-loads the confidential credentials into $SM_i's$ memory before its deployment in the SG environment. SM_i collects the sensitive information and transmits the accumulated information to SP_k via an openly available wireless channel, which is imperiled by different security vulnerabilities. Thus, ensuring the transmitted information's integrity and confidentiality is inevitable. In the subsequent sections, the propounded secure AC protocol is elaborated, which validates the authenticity of the deployed SM_i. For encrypted communications, it sets up a secret key between SP_k and SM_i.

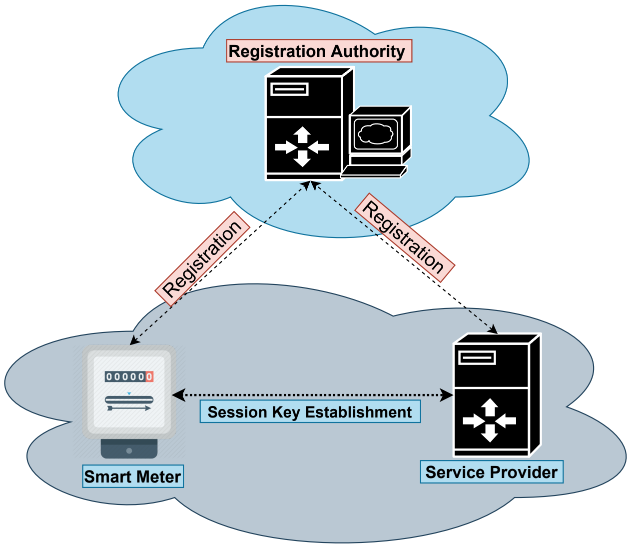

Figure 2. SG network.

3.2. Threat Model

We are considering the broadly utilized Dolev–Yao (DY) model for the proposed LACP-SG for the SG system [16,28]. The adversary $\mathcal{A}$ is able to alter and remove the content of the captured message. Furthermore, after updating the content of the captured message with malicious code, $\mathcal{A}$ can generate a malicious message. Network entities such as SM_i can be physically compromised by $\mathcal{A}$. Moreover, $\mathcal{A}$ can obtain sensitive data loaded in the memory of SM_i. In addition to this, $\mathcal{A}$ can use the procured information to carry out various attacks. In addition, SP_k is contemplated as the trusted entity of the SG system. As in the DY model, in the CK-adversary model, $\mathcal{A}$ can not only intercept communications in the SG environment, but the secret parameters, such as session keys and state and private keys, can also be compromised by $\mathcal{A}$.

4. Preliminaries

4.1. COMET

We use CHAM-based block cipher COMET-128 as the encryption/decryption scheme in the proposed LACP-SG. COMET is an AEAD scheme [27]. We express the encryption and decryption of COMET by $(CTx,\ AP_{tag}) = \mathcal{E}_K\ \{(N, AD), PTx\}$ and $(PTx,\ AP^{'}_{tag}) = \mathcal{D}_K\ \{(N, AD), CTx\}$, respectively, where K, N, AD, CTx, AP_{tag}, and PTx signifies "secret key", "nonce", "associative data", "ciphertext", "authentication parameter", and "plaintext", respectively. COMET decryption process will retrieve the plaintext if the condition $AP_{tag} = AP^{'}_{tag}$ holds.

4.2. Esch256

We use the hash function "Esch256" in designing LACP-SG, which is faster than SHA-160/256 and requires fewer computational resources. In addition, Esch256 renders the same functionality as provided by SHA-160/256 with an output size of 256 bits. Moreover, Esch256 renders enhanced security features.

4.3. Physical Unclonable Function

(PUF) is a one-way function. PUF produces a unique output (response) after taking the challenge as the input parameter. The operation of PUF can be represented as $R = PUF(CH)$.

4.4. Fuzzy Extractor

(FE) comprises two algorithms, namely, Generator $Gen(\cdot)$ and Reproducer $Rep(\cdot)$. The probabilistic algorithm $Gen(\cdot)$ produces key K_{SM_i} and Helper Data (HD) by taking bio-metric R of user, i.e., $(K_{SM_i}, HD) = Gen(R)$. $Rep(\cdot)$ is a deterministic algorithm that reproduce K_{SM} by considering the inputs R and HD, if the condition $HM(R, R') \leq et$ holds, where HM is the hamming distance between R and R' and et is the error tolerance.

5. The Proposed LACP-SG Protocol

The proposed LACP-SG protocol comprises four phases: (1) SM deployment phase; (2) SP Deployment Phase; (3) AC Phase; and (4) New SM Deployment. The subsequent subsections explain the details of the designed LACP-SG protocol. It is assumed that all the participants in the SG environment are time-synchronized to cope with replay attacks. Table 2 lists the notations utilized in devising LACP-SG.

Table 2. Notations used in LACP-SG.

Notation	Description
SM_i, SP_k	Smart meter (SM) and Service Provider (SP), respectively
PUF, CH, R	Physically unclonable function, challenge, and response, respectively
CP_{SP_k}	Common parameter of SP, which is known only to SP
TID_{SM_i}	Temporary-Identity of smart meter (SM)
ID_{SM_i}, ID_{SP_k}, K_{SP_k}	Real-Identity SM, SP, and secret key of SP
CT and AP_{tag}	Ciphertext and authentication parameter (*Tag*)
PT and AP'_{tag}	Plaintext and authentication parameter (*Tag*)
TS_1, TS_2	Timestamps in LACP-SG's AC phase
T_{mrc}, T_{dly}	Received and maximum delay time of a message
AD_1, AD_2	designates the associative data
N_1, N_2, N_3	Signifies the nonce or initialization vector
$\mathcal{E}_{\mathcal{K}}(msge)$, $\mathcal{D}_{\mathcal{K}}(msge)$	designates COMET based encryption/decryption of message "*msge*" employing secret *key*
$Gen(\cdot)$, HD, $Rep(\cdot)$	Signifies *FE* based key production, helper data, and key re-production function, respectively
RN_1, RN_2, RN_3	designates the random numbers
$\mathcal{A}$, $\|$, $H(.)$, $\oplus$,	Signifies attacker/adversary, concatenation, hash-function, and XOR, respectively
Adv, $INT-CTXT$	"Advantage of $\mathcal{A}$ and ciphertext integrity"
$OPRP-CPA$	"Online pseudo-random permutation chosen-plaintext attack"

5.1. SP Deployment Phase

The SP deployment phase is accomplished by RA to deploy SP_k. For this, RA picks a unique identity ID_{SP_k} and computes the secret key for the SP_k deployed in SG environment as $K_{SP_k} = H(K_{RA} \| ID_{SP_k})$, where K_{RA} is the private key of RA. In addition, RA stores the list of credentials $\{ID_{SP_k}, K_{SP_k}\}$ in the temper-resistance database of SP_k. RA also stores the credentials $\{ID_{SP_k}, K_{SP_k}\}$ in its own database.

5.2. SM Deployment Phase

SM_i deployment phase (SDP) is executed by RA. RA stores the secret credentials before SM_i deployment in the SG environment by performing the trailing necessary steps.

5.2.1. Step SDP-1

SM_i picks a real identity ID_{SM_i} of size 128 bits and a random number RN_r of size 128 bits. SM_i fabricates a message with parameters $\{ID_{SM_i}, RN_r\}$ and sends it to RA through a secure channel. RA picks a challenge parameter CH_{SM_i} and computes temporary identity $TID_{SM_i} = (ID_{SM_i} \| RN_{SM_i}) \oplus CP_{SP_k}$, where $CP_{SP_k} = H(ID_{SP_k} \| K_{SP_k})$. In addition to this, RA computes $U = H(ID_{SM_i})$ and determines $SID_i = (U_1 \oplus U_2)$, where U_1 and U_2 are derived by splitting U into two same-sized chunks, each with the size of 128 bits. RA sends the credentials $\{CH_{SM_i}, TID_{SM_i}\}$ to SM_i via the secure channel.

5.2.2. Step SDP-2

After receiving the parameters $\{CH_{SM_i}, TID_{SM_i}\}$ from RA, SM_i generates a response by using PUF function as $R_i = PUF(CH_{SM_i})$. In addition, SM_i by using FE computes $(K_{SM_i}, HD) = Gen(R_i)$ and sends K_{SM_i} to SP_k through a protected channel. Finally, SM_i keeps the credentials $\{TID_{SM_i}, CH_{SM_i}, RN_r, HD\}$ in its own memory.

5.2.3. Step SDP-3

Upon obtaining K_{SM_i} from SM_i, RA computes $B_i = (K_{SM_i} \| RN_r) \oplus CP_{SP_k}$. Finally, RA stores the parameters $\{SID_i, B_i\}$ in the database of SP_k.

5.3. AC Phase

In AC phase (ACP), SM_i achieves MA with SP_k. Moreover, SM_i establishes a secret SEK with SP_k to achieve encrypted communication. The trailing steps provide a detailed explanation of the AC phase.

5.3.1. Step ACP-1

SM_i retrieves CH_{SM_i} from its memory, stored in the SM_i memory during its deployment phase and computes $R_i = PUF(CH_{SM_i})$. SM_i regenerates K_{SM_i} by using FE as $K_{SM_i} = Rep(R_i, HD)$, where the size of K_{SM_i} is 128 bits. In addition, SM_i selects the current timestamps TS_1 with size 32 bits, the random number RN_1 with size 128 bits, and computes $A = H(TS_1 \| RN_r)$ and nonce $N_1 = A_1 \oplus A_2$, where A_1 and A_2 are procured by splitting A into two same-sized chunks, each with the size of 128 bits. In addition, SM_i computes the associative data $AD_1 = X_1 \oplus X_2$, where X_1 and X_2 are two equal parts of TID_{SM_i}. The size of N_1 and AD_1 is 128 bits. SM_i by using COMET computes $(CT_1, AP_{tag1}) = \mathcal{E}_{K_{SM_i}} \{(N_1, AD_1), RN_1\}$, where CT_1, AP_{tag1}, and RN_1 denote ciphertext, authentication parameter (Tag), and plaintext, respectively. Finally, SM_i constructs a message M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and sends M_1 to SP_k through a public communication channel.

5.3.2. Step ACP-2

Upon procuring M_1 form SM_i, SP_k checks the condition $T_{dly} \geq |T_{mrc} - TS_1|$ to validate the M_1 freshness, where T_{dly} is the allowed time delay, T_{mr} is the M_1 received time, and TS_1 designates the M_1 generation time. If the condition holds, SP_k considers M_1 as the authentic message and proceeds with the AC process. Otherwise, SP_k discards M_1 and obstructs the AC process. SP_k determines the common parameter CP_{SP_k} as $CP_{SP_k} = H(ID_{SP_k} \| K_{SP_k})$. Moreover, SP_k retrieves ID_{SM_i} and RN_{SM_i} by computing $TID_{SM_i} \oplus CP_{SP_k} = (ID_{SM_i} \| RN_{SM_i})$, where TID_{SM_i} is received with M_1 and CP_{SP_k} is computed at SP_k. Additionally, SP_k picks the retrieved ID_{SM_i} and computes $Q = H(ID_{SM_i})$ and $SID_i = Q_1 \oplus Q_2$, where Q_1 and Q_2 are two chunks of Q each of 128 bits. In addition, SP_k checks if SID_i is located in its database (memory). If SID_i is found, SP_k retrieves the credential $\{B_i\}$ corresponding to SID_i, stored in the database (memory) of SP_k. In addition to this, SP_k computes $CP_{SP_k} \oplus B_i = (RN_r \| K_{SM_i})$. Additionally, SP_k determines $AA = H(TS_1 \| RN_r)$ and nonce $N_2 = AA_1 \oplus AA_2$, where AA_1 and AA_2 are procured by splitting AA into two same-sized chunks, each with the size of 128 bits. Furthermore, SM_i computes $AD_2 = X_1^a \oplus X_2^a$, where X_1^a and X_2^a are two equal parts of TID_{SM_i}. Finally, SP_k by using COMET computes $(RN_1, AP_{tag2}) = \mathcal{D}_{K_{SM_i}}\ \{(N_2, AD_2), CT_1\}$, where AD_2, N_2, CT_1, AP_{tag2}, and RN_1 denote associative data, nonce, ciphertext, authentication parameter (Tag), and plaintext, respectively. To validate the authenticity of M_1, SP_k checks the condition $AP_{tag1} = AP_{tag2}$. If it holds, SP_k considers M_1 as the authentic message, which is received from a valid SM_i. Otherwise, SP_k discards M_1 and aborts the AC process.

5.3.3. Step ACP-3

After substantiating the authenticity of M_1, SP_k picks timestamp TS_2, RN_2, $RN^n_{SM_i}$, and computes the new temporary identity $TID^{new}_{SM_i}$ as $(ID_{SM_i} \parallel RN^n_{SM_i}) \oplus CP_{SP_k} = TID^{new}_{SM_i}$, where ID_{SM_i} is real identity of SM_i and $RN^n_{SM_i}$ is a new random number. Moreover, SP_k computes $K^1_{SM_i} = (K_{SM_i} \oplus RN_1)$, which is used in the encryption process. For encrypted communication in future, SP_k computes SEK as $SK_{SP_k} = H(TID_{SM_i} \parallel RN_1 \parallel RN_2 \oplus ID_{SM_i} \parallel TS_2 \parallel TID^{new}_{SM_i})$ and calculates $SK_{v1} = SK^a_{SP_k} \oplus SK^b_{SP_k}$. Furthermore, SP_k determines $N_3 = (RN_r \oplus RN_1)$, and $PT_1 = (TID^{new}_{SM_i} \parallel (RN_2 \oplus ID_{SM_i}) \parallel SK_{v1})$. In addition to this, by using COMET, SP_k computes $(CT_2, AP_{tag3}) = \mathcal{E}_{K^1_{SM_i}}\ \{(N_3, AD_2), PT_1\}$, where AD_2, N_3, CT_2, AP_{tag3}, and PT_1 denote associative data, nonce, ciphertext, authentication parameter, and plaintext, respectively. Finally, SP_k contrives a message M_2: $\{TS_2, CT_2, AP_{tag3}\}$ and dispatches M_2 to SM_i via an open/wireless channel.

5.3.4. Step ACP-4

After acquiring M_2 from SP_k, SM_i checks the condition $T_{dly} \geq |T_{mrc} - TS_2|$ to validate the freshness of M_2. If M_2 is fresh, SM_i determines $N_4 = (RN_r \oplus RN_1)$, $K^2_{SM_i}=(K_{SM_i} \oplus RN_1)$, and by using COMET computes $(PT_1, AP_{tag4})= \mathcal{D}_{K^2_{SM_i}}\ \{(N_4, AD_1), CT_2\}$, where AD_1, N_4, CT_2, and AP_{tag4} denote associative data, nonce, ciphertext, authentication parameter (Tag), and plaintext, respectively. Moreover, SM_i checks the condition $AP_{tag3} = AP_{tag4}$. If it holds, SM_i procures the plaintext $PT_1 = (TID^{new}_{SM_i} \parallel (RN_2 \oplus ID_{SM_i}) \parallel SK_{v1})$ from the decryption process. For indecipherable communication, SM_i computes the SEK as $SK_{SM_i} = H(TID_{SM_i} \| RN_1 \| RN_2 \oplus ID_{SM_i} \parallel TS_2 \| TID^{new}_{SM_i})$. In addition to this, SM_i calculates $SK_{v2} = SK^a_{SM_i} \oplus SK^b_{SM_i}$ and checks the condition $SK_{v1} = SK_{v2}$. If it holds, both SK_{SM_i} and SK_{SP_k} are equal. Otherwise, it terminates the AC process. Finally, SM_i updates TID_{SM_i} with $TID^{new}_{SM_i}$ in its own memory. Figure 3 summarizes the LACP-SG AC phase.

Smart Meter SM_i	**Service Provider** SP_k
$\{TID_{SM_i}, CH_{SM_i}, RN_r, HD\}$	$\{SID_i, B_i\}$
picks CH_{SM_i} from its memory, computes $R_i = PUF(CH_{SM_i})$, $K_{SM_i} = Rep(R_i, HD)$, picks TS_1, RN_1, and computes $A = H(TS_1 \parallel RN_r)$, $N_1 = (A_1 \oplus A_2)$, A_1 and A_2 are derived from A, $AD_1 = (X_1 \oplus X_2)$, X_1 and X_2 are derived from TID_{SM_i}, $(CT_1, AP_{tag1})=\mathcal{E}_{K_{SM_i}}\ \{(N_1, AD_1), RN_1\}$, $\xrightarrow[SM_i \rightarrow SP_k]{M_1:\{TS_1,\ TID_{SM_i},\ CT_1,\ AP_{tag1}\}}$.	checks $T_{dly} \geq \|T_{mrc} - TS_1\|$, if holds, computes $CP_{SP_k} = H(ID_{SP_k} \parallel K_{SP_k})$, extracts ID_{SM_i} and RN_{SM_i} as $TID_{SM_i} \oplus CP_{SP_k} = (ID_{SM_i} \parallel RN_{SM_i})$, $Q = H(ID_{SM_i})$, $SID_i = Q_1 \oplus Q_2$, checks if SID_i exists, if so retrieves $\{B_i\}$ and computes $CP_{SP_k} \oplus B_i = (RN_r \parallel K_{SM_i})$, $AA = H(TS_1 \parallel RN_r)$, $N_2 = (AA_1 \oplus AA_2)$, AA_1 and AA_2 are derived from AA, computes $AD_2 = X^a_1 \oplus X^a_2$, where X^a_1 and X^a_2 are derived from TID_{SM_i}, computes $(RN_1, AP_{tag2}) = \mathcal{D}_{K_{SM_i}}\ \{(N_2, AD_2), CT_1\}$, checks $AP_{tag1} = AP_{tag2}$, if holds, picks TS_2, RN_2, $RN^n_{SM_i}$, and computes $(ID_{SM_i} \parallel RN^n_{SM_i}) \oplus CP_{SP_k} = TID^{new}_{SM_i}$, $K^1_{SM_i}= (K_{SM_i} \oplus RN_1)$, $N_3 = (RN_r \oplus RN_1)$, computes $SK_{SP_k}=H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$, $SK_{v1} = SK^a_{SP_k} \oplus SK^b_{SP_k}$, $PT_1 = (TID^{new}_{SM_i} \parallel (RN_2 \oplus ID_{SM_i}) \parallel SK_{v1})$, $(CT_2, AP_{tag3}) = \mathcal{E}_{K^1_{SM_i}}\ \{(N_3, AD_2), PT_1\}$, $\xleftarrow[SP_k \rightarrow SM_i]{M_2:\{TS_2,\ CT_2,\ AP_{tag3}\}}$.
checks $T_{dly} \geq \|T_{mrc} - TS_2\|$, if holds, computes $N_4 = (RN_r \oplus RN_1)$, $K^2_{SM_i}=(K_{SM_i} \oplus RN_1)$, $(PT_1, AP_{tag4}) = \mathcal{D}_{K^2_{SM_i}}\ \{(N_4, AD_1), CT_2\}$, checks condition $AP_{tag3} = AP_{tag4}$, if holds, retrieves $PT_1 = (TID^{new}_{SM_i} \parallel (RN_2 \oplus ID_{SM_i}) \parallel SK_{v1})$, updates TID_{SM_i} with $TID^{new}_{SM_i}$, computes $SK_{SM_i}=H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$, $SK_{v2} = SK^a_{SM_i} \oplus SK^b_{SM_i}$ and checks $SK_{v1} = SK_{v2}$, if holds, both SK_{SM_i} and SK_{SP_k} are equal. Otherwise, it terminates the AC process.	
$SK_{SM_i}(= SK_{SP_k}) = H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$	

Figure 3. LACP-SG authentication phase.

5.4. New SM Deployment Phase

RA performs the subsequent steps to deploy a new SM_i^n.

5.4.1. Step SDP-1

SM_i^n picks a real identity $ID_{SM_i}^n$ and RN_r^n and sends $\{ID_{SM_i}^n, RN_r^n\}$ to RA through a protected channel. RA picks a new challenge $CH_{SM_i}^n$ and computes the new temporary identity $TID_{SM_i}^n = (ID_{SM_i}^n \parallel RN_{SM_i}^n) \oplus CP_{SP_k}$. Moreover, RA computes $U^n = H(ID_{SM_i}^n)$ and derives $SID_i^n = (U_1^n \oplus U_2^n)$, where U_1^n and U_2^n are derived by splitting U^n into two same-sized chunks, each with the size 128 bits. RA sends the credentials $\{CH_{SM_i}^n, TID_{SM_i}^n\}$ to SM_i^n via a secure channel.

5.4.2. Step SDP-2

After receiving a challenge $CH_{SM_i}^n$ from RA, SM_i^n generates a response by using the PUF function as $R_i^n = PUF(CH_{SM_i}^n)$. In addition, SM_i^n by using FE computes $(K_{SM_i}^n, HD^n)$ $= Gen(R_i^n)$ and sends $K_{SM_i}^n$ to RA via secure channel. Furthermore, SM_i stores $\{TID_{SM_i}^n, CH_{SM_i}^n, RN_r^n\}$ in its own memory. Upon receiving $K_{SM_i}^n$ from SM_i^n, RA computes. In addition, SP_k computes $B_i^n = (K_{SM_i}^n \parallel RN_r^n) \oplus CP_{SP_k}$. Finally, RA stores the parameters $\{SID_i^n, B_i^n\}$ in the SP_k database.

6. Security Analysis

6.1. Informal Security Analysis

6.1.1. Anonymity and Untraceability

Assume $\mathcal{A}$ eavesdrops the communicated messages, such as M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and M_2: $\{TS_2, CT_2, AP_{tag3}\}$, which are exchanged during the AC phase of the proposed LACP-SG. $\mathcal{A}$ cannot determine the real identity of SM of SP, which are ID_{SM_i} and ID_{SP_k}, respectively, from the captured M_1 and M_2. $\mathcal{A}$ by capturing M_1 and M_2 cannot procure the real identities of SM and SP.

6.1.2. Replay Attack

$\mathcal{A}$ after expropriating all the messages, such as M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and M_2: $\{TS_2, CT_2, AP_{tag3}\}$ tries to regenerate the captured messages to obtain helpful information from the participants of the AC phase. However, we assume the system is time-synchronized, and each message bears the newest timestamp and random numbers. $\mathcal{A}$ cannot frame the replay attack because the entities SM_i and SP_k verify the newness/oldness of the obtained message by confirming the condition $T_{dly} \geq |T_{mrc} - TS_1|$ and $T_{dly} \geq |T_{mrc} - TS_2|$, respectively. If the obtained transmission is delayed, the entity of the receiving will dump the obtained message. In this way, the proposed LACP-SG detects the replayed messages and discards such received messages. Hence, LACP-SG is protected against replay attacks.

6.1.3. DeS Attack

The proposed LACP-SG renders resistance against DeS attack. For anonymous communication, SM_i uses TID_{SM_i}, which is updated by SP_k during the accomplishment of every new AC session. SP_k constructs TID_{SM_i} by concatenating ID_{SM_i} and a fresh random number RN_{SM_i}, i.e., $(ID_{SM_i} \parallel RN_{SM_i}) \oplus CP_{SP_k}$, where ID_{SM_i} remains constant and RN_{SM_i} is updated to $RN_{SM_i}^n$. Suppose $\mathcal{A}$ drops M_2 during the execution of the AC phase. This action of $\mathcal{A}$ cannot affect the execution of the new AC session because ID_{SM_i} is constant, which is extracted by SP_k to compute the SID_i. SID_i is used to find the record at SP_k related to SM_i. So, LACP-SG is capable of resisting the DeS attack.

6.1.4. Privilege Insider Attack

To accomplish the authentication phase in the proposed LACP-SG scheme, SP_k stores the parameters $\{SID_i, B_i\}$ in the database. Thus, to fabricate a valid messages, such as M_1:

$\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and M_2: $\{TS_2, CT_2, AP_{tag3}\}$, it is imperative for $\mathcal{A}$ to compute $CP_{SP_k} \oplus B_i = (RN_r \parallel K_{SM_i})$. However, without knowing the secret key of SP_k, it is hard for $\mathcal{A}$ to extract RN_r and K_{SM_i}, which are required to construct M_1 and M_2. Hence, LACP-SG can resist the PrI attack.

6.1.5. MIDM Attack

Assume that $\mathcal{A}$ expropriates all the exchanged messages M_1 and M_2 between the entities during the AC phase over the wireless/open communication channel. Now, $\mathcal{A}$ may attempt to reconstruct the seized messages to make the participants of the system believe that the received messages are generated by licit entities. To simulate a licit message M_1 on behalf of SM_i, $\mathcal{A}$ requires to have all the confidential/secret credentials of SM_i, i.e., $\{ID_{SM_i}, CH_i, K_{SM_i}\}$. Similarly, $\mathcal{A}$ needs to extricate all the secret/confidential parameters of SP_k to construct a valid response message on behalf of SP_k. However, without having all the confidential credentials of SM_i and SP_k, it is impractical for $\mathcal{A}$ to construct a valid message. Therefore, LACP-SG can restrain MIDM attacks.

6.1.6. Impersonation/Modification/Injection Attack

To impersonate as SP_k, A has to regenerate the message M_2 on behalf of SP_k to make SM_i believe that the message is licit and obtained from an honest SP_k. Now, suppose $\mathcal{A}$ attempts to generate M_1 with valid credentials. However, to generate M_2, $\mathcal{A}$ requires knowing the confidential credentials of SP_k. However, $\mathcal{A}$ cannot produce a valid message M_2 in polynomial time without knowing the secret credentials to emulate as legitimate SP_k. Similarly, $\mathcal{A}$ requires knowing the confidential credentials of SM_i. Therefore, LACP-SG is protected against SM_i and SP_k impersonation attacks.

6.1.7. Key Compromise Impersonation Attack

In this attack, $\mathcal{A}$ tries to impersonate as a valid SM_i by compromising the long-term secret key of SP_k. However, to construct a valid message M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$, it is necessary for $\mathcal{A}$ to obtain the secret parameters, such as RN_r and K_{SM_i}. Thus, without having these confidential parameters, it is hard for $\mathcal{A}$ to impersonate a valid SM_i. Similarly, without having the confidential parameters of SP_k, $\mathcal{A}$ cannot impersonate a licit SP_k. In this way, LACP-SG can resist key compromise impersonation attacks.

6.1.8. Known Session-Specific Temporary Information Leakage/EPSL Attack

According to the CK-adversary model, $\mathcal{A}$ can compromise the secret credentials (Long Term Secrets (LTS), Ephemeral Secrets (ES)), and session states aside from all the actions allowed under the DY model. In LACP-SG, the session key is created using both LTS and ES, i.e., $SK_{SM_i} (= SK_{SP_k}) = H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$. Therefore, it is imperative for $\mathcal{A}$ to guess that both LTS and ES construct the session key.

6.1.9. SM Capture/Memory Modification Attack

According to the DY threat model, $\mathcal{A}$ can seize some of the SMs from in the SG environment. $\mathcal{A}$ can extricate the secret credentials by using a power analysis attack kept in the memory of SM. However, the parameters CH_i, RN_r, and TID_{SM_i} are unlike for all SMs installed in the SG environment. Therefore, by capturing some of the installed SMs, $\mathcal{A}$ cannot compromise the security of the whole SG environment. Hence, LACP-SG is resilient against SM capture attacks.

6.2. ROM-Based Formal Security Analysis

This section provides a ROM-based analysis of the SEK security between SM_i and SP_k during the execution of the AC phase of LACP-SG. The subsequent components are described in the ROM model.

Participants: Suppose that Ψ^{t1}_{RA}, $\Psi^{t2}_{SM_i}$, and $\Psi^{t3}_{SP_k}$ represent instances $t1$, $t2$, and $t3$ of the participants RA, SM_i, and SP_k, denoted as oracles.

Accepted state: When an instance Ψ^t acquires the last message, it will be in the accepted state. The session identification (Sid) of Ψ^t for the current session prescribes the ordered sequence of all exchanged messages (i.e., messages sent/received by Ψ^t).

Partnering: Two instances Ψ^{t2} and Ψ^{t2} are partners only if both are in an acceptable state and share similar session keys.

Freshness: $\mathcal{A}$ is unable to obtain the SEK established between SM_i and SP_k by running the *Reveal* query presented in Table 3.

Adversary: $\mathcal{A}$ can fully control and seize all the messages and alter, falsify, and infiltrate messages by employing the queries expressed in Table 3. $\mathcal{A}$ can execute the hash function $H(.)$, referred to as random oracle $ESHah$.

Table 3. ROM-based queries.

Query	Purpose
$Execute(\Psi^{t2}_{SM_i}, \Psi^{t3}_{SP_k})$	Perpetration of this query enables $\mathcal{A}$ to seize all the transmitted messages between SM_i and SP_k.
$Send(\Psi^t, Msg)$	Perpetration of this query enables $\mathcal{A}$ to yield an active attack by dispatching a message Msg to Ψ^{t2} and Ψ^{t1} also respond to Msg accordingly.
$Reveal(\Psi^t)$	Perpetration of this query enables $\mathcal{A}$ to get the shared SEK, utilized to guarantee the secure transmission between Ψ^{t1} and its interrelated entity.
$CorruptSM(\Psi^{t2}_{SM_i})$	Perpetration of this query helps $\mathcal{A}$ to acquire the secret/private parameters loaded in the storage of SM_i by operating PA attack.
$Test(\Psi^t)$	Perpetration of this query enables $\mathcal{A}$ to ascertain whether the guessed SEK is licit or random output, just like the outcome of a flipped coin, say C.

Definition 1. *Online chosen ciphertext attack (OCCA3) advantage of $\mathcal{A}$, which is executing against an AEAD scheme in polynomial-time (pt), can be defined as follows.*

$$\begin{aligned} Adv_{\varphi}^{OCCA3}(\mathcal{A}) \leq\ & Adv_{\varphi}^{OPRP-CPA}(que, len, pt) \\ & + Adv_{\varphi}^{INT-CTXT}(que, len, pt), \end{aligned} \tag{1}$$

Theorem 1. *Let $\mathcal{A}$ run against LACP-SG in pt to derive the established SEK between SM_i and SP_k during the AC phase. Let H_{que} signify Esch256 queries, $|ESHah|$ designates the range space of Esch256 output, H_{puf} represents PUF quires, $|PUF|$ designates the range space of PUF output, and $Adv^{OCCA3}_{COMET,\mathcal{A}}(que, len, pt)$ is the advantage in compromising the security of an online AEAD scheme (COMET) (Definition 1). The maximum advantage of $\mathcal{A}$ for compromising the security of SEK, established between SM_i and SP_k, can be described as follows:*

$$\begin{aligned} Adv_{\mathcal{A}}^{LACP-SG}(pt) \leq\ & \frac{H^2_{que}}{|ESHah|} + \frac{H^2_{puf}}{|PUF|} \\ & + 2.Adv^{OCCA3}_{COMET,\mathcal{A}}(que, len, pt). \end{aligned} \tag{2}$$

Proof. The succeeding five games $(GM_z|z = 0, 1, 2, 3, 4)$ are executed to prove Theorem 1. We heed the identical means to establish the proof of Theorem 1 as followed in [29–33]. In addition to this, we characterize the $\mathcal{A}$ advantage in compromising the security of SEK by $Adv_{\mathcal{A}}^{LACP-SG}(pt) = |2 \cdot Pr[SuS] - 1|$, where "$Pr[SuS]$" indicates the possibility of a circumstance where $\mathcal{A}$ can achieve/win the game. LACP-SG is defended if $Adv_{\mathcal{A}}^{LACP-SG}$ (pt) is insignificant.

GAM_0: In this game, $\mathcal{A}$ performs an active attack against LACP-SG under ROM. $\mathcal{A}$ at the commencement of GM_0 guesses the bit C' randomly. Then, trailing can be achieved

$$Adv_{\mathcal{A}}^{LACP-SG}(pt) = |2.Pr[SuS0] - 1|. \tag{3}$$

GAM_1: In GAM_1, $\mathcal{A}$ makes the *execute* query to effectuate the eavesdrop attack. By effectuating eavesdrop attack during the execution of AC phase, $\mathcal{A}$ can intercept all the exchanged messages, such as M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and M_2: $\{TS_2, CT_2, AP_{tag3}\}$. $\mathcal{A}$ effectuates *Test* at the end of this game and validates whether the outcome of the *Test* query is a random number or a real session key, i.e., $SK_{SM_i}(= SK_{SP_k}) = H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$, where $TID^{new}_{SM_i} = (ID_{SM_i} \parallel RN) \oplus CP_{SP_k}$. The session key is produced in the proposed LACP-SG using the LTS and ES. Therefore, to reveal the session key established between SM_i and SP_k, it is imperative for $\mathcal{A}$ to guess both the ES and LTS simultaneously. However, it is impractical for $\mathcal{A}$ to procure all the secret parameters by capturing M_1 and M_2. So, the winning chance of this game for $\mathcal{A}$ will not increase by effectuating the eavesdrop attack:

$$Pr[SuS0] = Pr[SuS1]. \tag{4}$$

GAM_2: In this game, the aim of $\mathcal{A}$ is to deceive an entity to receive a mutated message. $\mathcal{A}$ is authorized to make various *ESHah* queries to check the presence of the hash collisions. All the exchanged messages, such as M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$ and M_2: $\{TS_2, CT_2, AP_{tag3}\}$ during the AC phase indirectly include the associative data and nonce, and temporary identities, which are protected by the collision-resistant Esch256 hash function. Therefore, there will be no collision when $\mathcal{A}$ performs *Send* queries. The consequences of the birthday paradox confer

$$|Pr[SuS1] - Pr[SuS2]| \leq \frac{H^2_{que}}{2|ESHah|}. \tag{5}$$

GAM_3: This game is considered a continuation of GAM_2 that simulates PUF queries. According to GAM_2, it follows that

$$|Pr[SuS3] - Pr[SuS2]| \leq \frac{H^2_{puf}}{2|PUF|}. \tag{6}$$

GAM_4: In this game, $\mathcal{A}$ attempts to construct the session key by capturing M_1 and M_2, which are protected by AEAD scheme. In LACP-SG the session key in constructed as $SK_{SM_i}(= SK_{SP_k}) = H(TID_{SM_i} \parallel RN_1 \parallel (RN_2 \oplus ID_{SM_i}) \parallel TS_2 \parallel TID^{new}_{SM_i})$. Therefore, $\mathcal{A}$ has to procure RN_1 and RN_2, which are encrypted using AEAD scheme (COMET). Moreover, the associative data and the initialization vector used in the encryption process are random. In addition, secret keys are required to decrypt CT_1 and CT_2. It is computationally impractical to perform the decryption process in polynomial time. Due to OCCA3 property (Definition 1), it then follows that

$$|Pr[SuS3] - Pr[SuS4]| \leq Adv^{OCCA3}_{COMET,\mathcal{A}}(que, len, pt). \tag{7}$$

As all the queries are performed, $\mathcal{A}$ executes the *Test* queries to presume bit $C^{'}$ for winning the game. Thus, we obtain

$$Pr[SuS4] = 1/2. \tag{8}$$

From (3) and (4), we obtain

$$Adv^{LACP-SG}_{\mathcal{A}}(pt) = |2.Pr[SuS0] - \frac{1}{2}|. \tag{9}$$

From (9), we obtain

$$\frac{1}{2}.Adv^{LACP-SG}_{\mathcal{A}}(pt) = |Pr[SuS0] - \frac{1}{2}|. \tag{10}$$

By using (8) and (10), we obtain

$$\frac{1}{2}.Adv_{\mathcal{A}}^{LACP-SG}(pt) = |Pr[SuS1] - Pr[SuS4]| \tag{11}$$

Through triangular inequality, we obtain

$$\begin{aligned} |Pr[SuS1] - Pr[SuS4]| \leq |Pr[SuS1] - Pr[SuS2]| \\ + |Pr[SuS2] - Pr[SuS4]| \\ \leq |Pr[SuS1] - Pr[SuS2]| + |Pr[SuS2] - Pr[SuS3]| \\ + |Pr[SuS3] - Pr[SuS4]|. \end{aligned} \tag{12}$$

By utilizing (5), (6), (7) and (12), we obtain

$$\begin{aligned} Adv_{\mathcal{A}}^{LACP-SG}(pt) \leq \frac{H_{que}^2}{|ESHah|} + \frac{H_{puf}^2}{|PUF|} \\ +2.Adv_{COMET,\mathcal{A}}^{OCCA3}(que, len, pt). \end{aligned} \tag{13}$$

□

6.3. Scyther Based Formal Security Verification

We investigated the formal security of LACP-SG by utilizing the widely adopted validation tools, i.e., Scyther. Scyther is a Python-based software designed to formally analyze the security of the authentication schemes, their security claims, and potential vulnerabilities. Scyther employs the Security Protocol Description Language (SPDL) for describing a devised security scheme and is also utilized to determine the weaknesses of a security scheme by demonstrating any potential threats or risks. In the proposed LACP-SG, two roles are defined, such as SM_i and SP_k. There are two manually specified claims, such as $claim(SM, Secret, SK)$ and $claim(SP, Secret, SK)$, which are validated by Scyther, as shown in Figure 4. In addition, Scyther also generates the claims, such as $claim(SM, Alive)$, $claim(SM, Nisynch)$, and $claim(SM, Niagree)$, which are validated as demonstrated in Figure 4.

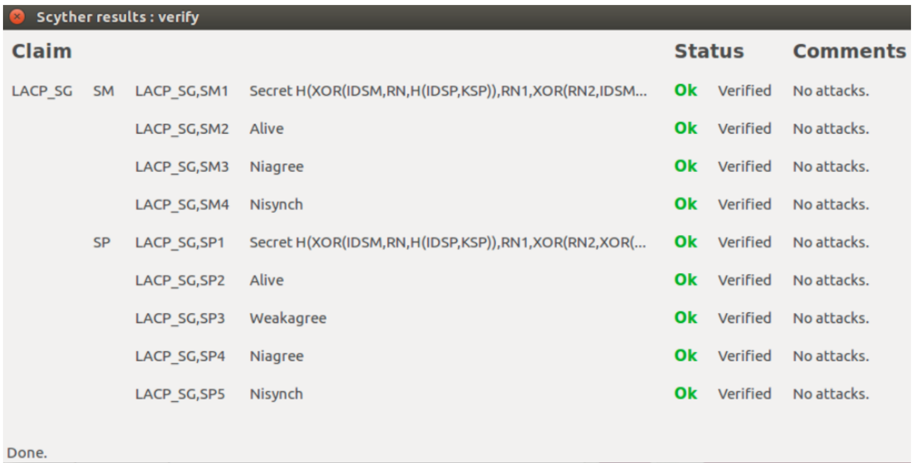

Figure 4. Security analysis of LACP-SG using Scyther.

7. Performance Evaluation

LACP-SG is contrasted with other protocols, such as in Bera et al. [29], Chaudhry et al. [30], Bera et al. [34], Kumar et al. [11], Chaudhry et al. [35], and Mehmood et al. [20]. We use the Python-based library "PyCrypto" along with COMET code to acquire the time complexity of cryptographic primitives and COMET. Table 4 depicts the time complexities of different cryptographic operations.

Table 4. Time complexity of different cryptographic operations.

Notations	Operation	Time on R-Pi3	Time on SP_k
T_{ecc}	ECC-based point multiplication	2.70 ms	0.705 ms
T_{en}	Symmetric key encryption	0.41 ms	0.015 ms
T_{eca}	ECC-based point addition	0.134 ms	0.007 ms
T_H	One-way hash function (16 bytes)	0.345 ms	0.039 ms
T_{HE}	Esch256 one-way hash function (32 bytes)	0.330 ms	0.032 ms
T_{pu}	Physical-unclonable-function	0.49 µs	-
T_{CO}	COMET	0.349 ms	0.041 ms
$T_{rep} \approx T_{ecc}$	Bio-metric key generation and reproduction	2.70 ms	0.705 ms

Time complexities are computed on Quad-core Raspberry Pi-3 (R-Pi3) with CPU @1.2 GHz, and 1GB of RAM"and " Core(TM) i7-6700 system with CPU @3.40 GHz, and RAM 8 GB" to simulate SM_i SP_k, respectively.

7.1. Security Comparison

A comparison of the security properties of LACP-SG and other related AC schemes is demonstrated in Table 5. That of Bera et al. [29] cannot restrain the DeS attack, that of Bera et al. [34] is unprotected against the DeS attack, and that of Mehmood et al. [20] is insecure against the DoS, MIDM, PrI, EPSL, RA attacks and does not provide the SEK security. The scheme of Kumar et al. [11] is against DIMP, MIDM, and EPSL attacks and does not provide SEK security. In addition to this, the scheme of Chaudhry et al. [35] is incapable of resisting EPSL, SIMP, DIMP, device capture, and SEK disclosure attacks. Moreover, Chaudhry et al. [30] provide insecure certificate computation, which causes various attacks, such as device capture and DIMP attacks. However, the proposed LACP-SG is secure and protected against various pernicious attacks, such as MIDM and DeS attacks.

Table 5. Security comparison.

Features	Chaudhry et al. [30]	Bera et al. [29]	Bera et al. [34]	Mehmood et al. [20]	Kumar et al. [11]	Chaudhry et al. [35]	LACP-SG
PrI	✓	✓	✓	×	✓	✓	✓
DIMP	×	✓	✓	✓	×	×	✓
SPI	✓	✓	✓	✓	✓	×	✓
DCA	×	✓	✓	✓	✓	×	✓
MIDM	✓	✓	✓	×	×	✓	✓
DeS	✓	×	×	✓	✓	✓	✓
DoS	✓	✓	✓	×	✓	✓	✓
RA	✓	✓	✓	×	✓	✓	✓
SEKS	✓	✓	✓	×	×	×	✓
EPSL	✓	✓	×	×	×	✓	✓
ROM	✓	✓	✓	×	✓	✓	✓
MA	✓	✓	✓	✓	×	✓	✓
SCER	×	✓	✓	✓	✓	×	-

SCER: secure certificate computation; DCA: device capture attack; ✓: indicates the supported functionality; ×: represents the functionality is not available.

7.2. Communication Overhead Comparison

For analyzing the communication overhead that occurred during the AC phase, we suppose that the length of the ECC point, identity, hash function output, initialization vector/random number/nonce, and timestamp are 320, 128, 256, 128, and 32 bits, respectively. There are two messages required to accomplish the AC phase of LACP-SG, i.e., M_1: $\{TS_1, TID_{SM_i}, CT_1, AP_{tag1}\}$, M_2: $\{TS_2, CT_2, AP_{tag3}\}$. The sizes of M_1 and M_2 are {32 + 256 + 128 + 128} = 544 bits and {32 + 512 + 128} = 662 bits. Hence, the communication cost of LACP-SG is {662 + 544} = 1206 bits, which is 56.68%, 10.27%, 49.07%, 12.35%, 27.52%, and 10.27% lesser than the scheme of Bera et al. [29], Chaudhry et al. [30], Bera et al. [34], Kumar et al. [11], Chaudhry et al. [35], and Mehmood et al. [20], respectively. The comparison between

LACP-SG and the related AC protocol communication overhead is given in Table 6 and Figure 5.

Table 6. Communication overhead comparison.

AC Protocol	Disseminated Messages During AC Phase	Total (bits)
Bera et al. [29]	$SM_i \xrightarrow{1120} SP_k/GS \xrightarrow{1376} D_k/SM_i \xrightarrow{288} SP_k$	2784
Chaudhry et al. [30]	$SM_i \xrightarrow{832} SP_k \xrightarrow{512} SM_i$	1344
Bera et al. [34]	$D_k/SM_i \xrightarrow{864} SP_k/GS \xrightarrow{1216} D_k/SM_i \xrightarrow{288} SP_k$	2368
Kumar et al. [11]	$SM_i \xrightarrow{512} SP_k \xrightarrow{672} SM_i \xrightarrow{192} SP_k$	1376
Chaudhry et al. [35]	$SM_i \xrightarrow{832} SP_k \xrightarrow{832} SM_i$	1664
Mehmood et al. [20]	$SM_i \xrightarrow{672} SP_k \xrightarrow{672} SM_i$	1344
LACP-SG	$SM_i \xrightarrow{544} SP_k \xrightarrow{662} SM_i$	1206

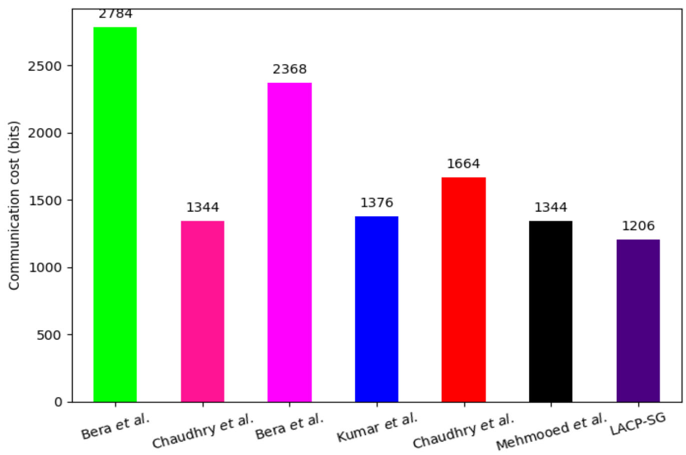

Figure 5. Communication cost needed to perform the AC phase (single SM_i) [11,20,29,30,34,35].

7.3. Computational Overhead Comparison

We employ the time complexity of different cryptographic operations, shown in Table 4, to estimate the computational overhead of LACP-SG and relevant AC protocol. LACP-SG requires the computational overhead of $7T_{HE} + 4T_{co} + T_{rep} + T_{pu} \approx 4.34$ ms in the AC phase. The schemes of Bera et al. [29], Chaudhry et al. [30], Bera et al. [34], Mehmood et al. [20], Kumar et al. [11], and Chaudhry et al. [35] require $22T_H + 8T_{ecc} + 2T_{eca} \approx 17.82$ ms, $8T_H + 9T_{ecc} + 2T_{eca} \approx 17.93$ ms, $18T_H + 4T_{en} + 4T_{ecc} + 2T_{eca} \approx 11.12$ ms, $12T_H + 4T_{ecc} \approx 14.42$ ms, $8T_H + 10T_{ecc} + 4T_{eca} \approx 18.79$ ms, and $8T_H + 9T_{ecc} + 5T_{eca} \approx 18.18$ ms, respectively, which are 75.14%, 75.29%, 60.16%, 69.28%, 76.42%, and 75.63% higher than the proposed LACP-SG, respectively, as shown in Table 7. Moreover, the computational cost needed at the SP_k and SM_i side is shown in Figure 6, where it is obvious that LACP-SG incurs lesser computational cost than the related AC protocols. Furthermore, Figure 7 illustrates the comparison of the computational cost at SP_k with increasing the authentication requests, which are generated by SM_i in the SG environment.

Table 7. Computational overhead comparison.

Protocol/Scheme	SM_i Side	SP_k Side	Total Time
Bera et al. [29]	$11T_H + 4T_{ecc} + T_{eca}$	$11T_H + 4T_{ecc} + T_{eca}$	$22T_H + 8T_{ecc} + 2T_{eca} \approx 17.82$ ms
Chaudhry et al. [30]	$4T_H + 5T_{ecc} + T_{eca}$	$4T_H + 4T_{ecc} + T_{eca}$	$8T_H + 9T_{ecc} + 2T_{eca} \approx 17.93$ ms
Bera et al. [34]	$9T_H + 2T_{en} + 2T_{ecc} + T_{eca}$	$9T_H + 2T_{en} + 2T_{ecc} + T_{eca}$	$18T_H + 4T_{en} + 4T_{ecc} + 2T_{eca} \approx 11.12$ ms
Kumar et al. [11]	$6T_H + 2T_{ecc}$	$6T_H + 2T_{ecc}$	$12T_H + 4T_{ecc} \approx 14.42$ ms
Chaudhry et al. [35]	$4T_H + 5T_{ecc} + 2T_{eca}$	$4T_H + 5T_{ecc} + 3T_{eca}$	$8T_H + 9T_{ecc} + 5T_{eca} \approx 18.79$ ms
Mehmood et al. [20]	$4T_H + 5T_{ecc} + 2T_{eca}$	$4T_H + 5T_{ecc} + 2T_{eca}$	$8T_H + 10T_{ecc} + 4T_{eca} \approx 18.18$ ms
LACP-SG	$2T_{HE} + 2T_{co} + T_{rep} + T_{pu}$	$5T_{HE} + 2T_{co}$	$7T_{HE} + 4T_{co} + T_{rep} + T_{pu} \approx 4.34$ ms

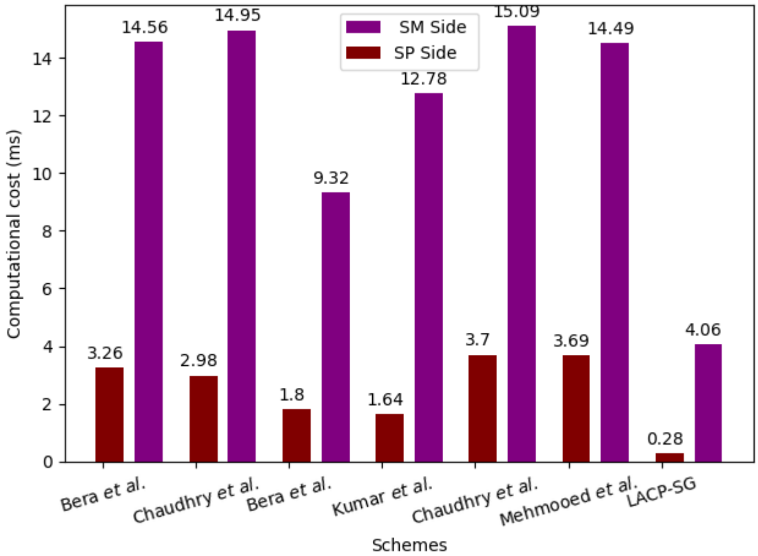

Figure 6. Computational cost at SM_i and SP_k side [11,20,29,30,34,35].

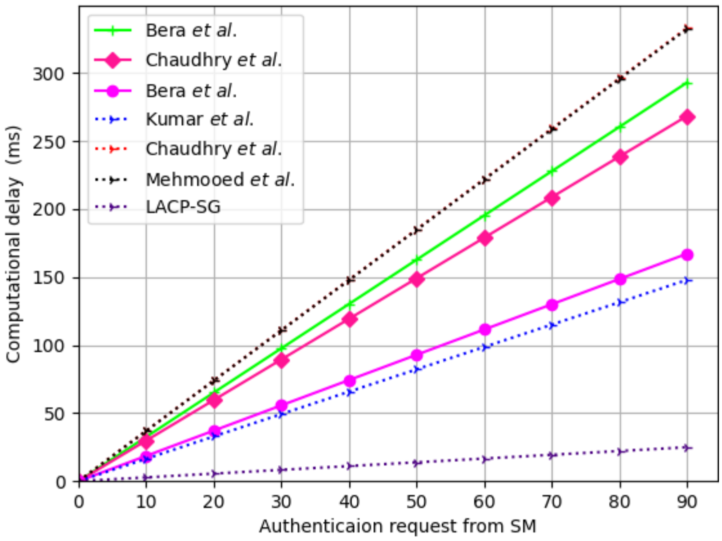

Figure 7. The computational cost increases with the number of authentication requests [11,20,29,30,34,35].

7.4. Storage Overhead Comparison

In LACP-SG, the smart meter SM_i and SP_k requires storing $\{CH_{SM_i}, TID_{SM_i}, RN_r, HD\}$ and $\{SID_i, B_i, RN_r\}$ size of { 256 + 256 + 160} = 672 bits and {128 + 256 } = 384 bits. To execute the AC phase, the aggregated storage overhead of LACP-SG is {672 + 384} = 1056 bits. The schemes of Bera et al. [29], Chaudhry et al. [30], Bera et al. [34], Mehmood et al. [20], Kumar et al. [11], and Chaudhry et al. [35] require storing 3008 bits, 1280 bits, 2752 bits, 1120 bits, 1240 bits, and 2400 bits, respectively, which are 64.89%, 17.5%, 61.63%, 5.71%, 14.84%, 56%, 37.26% higher than the proposed LACP-SG, respectively. The comparison of LACP-SG and the related AC protocols' storage overhead is given in Figure 8.

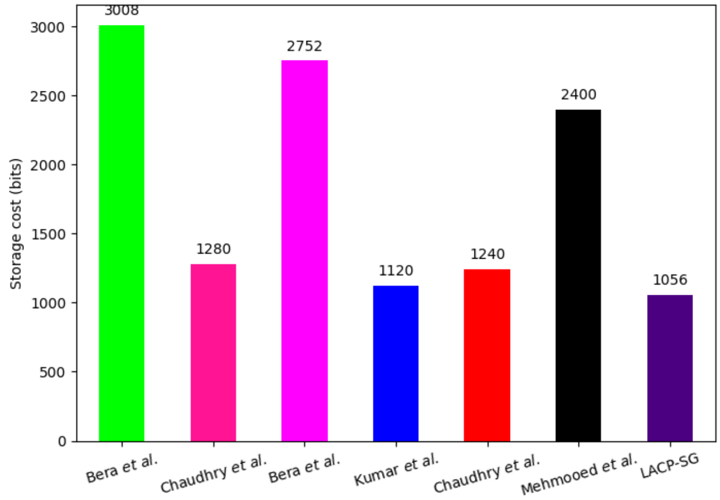

Figure 8. Total storage cost comparison [11,20,29,30,34,35].

8. Conclusions

This paper presents an AC protocol called LACP-SG, which enables secure communication in the resource-constrained SG environment. To this end, LACP-SG validates the authenticity of the deployed SM and establishes a SEK between the SM and server to accomplish secure communications. The security of the established SEK is validated through ROM-based analysis. Moreover, through Scyther-based analysis, LACP-SG is found to be secure against MIDM and replay attacks. Informal security analysis reveals that the protocol is protected against de-synchronization and SM capture attacks. Finally, a rigorous comparative analysis shows that LACP-SG renders superior security and requires lower computational, storage, and communication cost than the related AC protocols, thereby advocating the feasibility of LACP-SG for SG applications.

Author Contributions: Both authors contributed equally to preparing the article. The authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups Project under grant number RGP.2/201/43.

Data Availability Statement: Not applicable.

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

References

1. Salem, F.M.; Ibrahim, E.; Elghandour, O. A Lightweight Authenticated Key Establishment Scheme for Secure Smart Grid Communications. *Int. J. Saf. Secur. Eng.* **2020**, *10*, 549–558. [CrossRef]
2. Kumar, P.; Lin, Y.; Bai, G.; Paverd, A.; Dong, J.S.; Martin, A. Smart Grid Metering Networks: A Survey on Security, Privacy and Open Research Issues. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2886–2927. [CrossRef]
3. Chen, Y.; Martínez, J.F.; Castillejo, P.; López, L. A bilinear map pairing based authentication scheme for smart grid communications: Pauth. *IEEE Access* **2019**, *7*, 22633–22643. [CrossRef]
4. Li, X.; Wu, F.; Kumari, S.; Xu, L.; Sangaiah, A.K.; Choo, K.K.R. A provably secure and anonymous message authentication scheme for smart grids. *J. Parallel Distrib. Comput.* **2019**, *132*, 242–249. [CrossRef]
5. Huseinović, A.; Mrdović, S.; Bicakci, K.; Uludag, S. A Survey of Denial-of-Service Attacks and Solutions in the Smart Grid. *IEEE Access* **2020**, *8*, 177447–177470. [CrossRef]
6. Gope, P.; Sikdar, B. Privacy-aware authenticated key agreement scheme for secure smart grid communication. *IEEE Trans. Smart Grid* **2018**, *10*, 3953–3962. [CrossRef]
7. 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]
8. Tanveer, M.; Abbas, G.; Abbas, Z.H.; Bilal, M.; Mukherjee, A.; Kwak, K.S. LAKE-6SH: Lightweight User Authenticated Key Exchange for 6LoWPAN-Based Smart Homes. *IEEE Internet Things J.* **2021**, *9*, 2578–2591. [CrossRef]
9. Tanveer, M.; Khan, A.U.; Shah, H.; Alkhayyat, A.; Chaudhry, S.A.; Ahmad, M. ARAP-SG: Anonymous and Reliable Authentication Protocol for Smart Grids. *IEEE Access* **2021**, *9*, 143366–143377. [CrossRef]

10. 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.* **2021**, *9*, 6855–6865. [CrossRef]
11. Kumar, N.; Aujla, G.S.; Das, A.K.; Conti, M. ECCAuth: A Secure Authentication Protocol for Demand Response Management in a Smart Grid System. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6572–6582. [CrossRef]
12. 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]
13. Kim, S.; Kwon, E.Y.; Kim, M.; Cheon, J.H.; Ju, S.H.; Lim, Y.H.; Choi, M.S. A secure smart-metering protocol over power-line communication. *IEEE Trans. Power Deliv.* **2011**, *26*, 2370–2379. [CrossRef]
14. Abbasinezhad-Mood, D.; Nikooghadam, M. Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. *Future Gener. Comput. Syst.* **2018**, *84*, 47–57. [CrossRef]
15. Ostad-Sharif, A.; Abbasinezhad-Mood, D.; Nikooghadam, M. A robust and efficient ECC-based mutual authentication and session key generation scheme for healthcare applications. *J. Med. Syst.* **2019**, *43*, 10. [CrossRef]
16. Chaudhry, S.A.; Nebhen, J.; Yahya, K.; Al-Turjman, F. A Privacy Enhanced Authentication Scheme for Securing Smart Grid Infrastructure. *IEEE Trans. Ind. Inform.* **2021**, *18*, 5000–5006. [CrossRef]
17. Far, H.A.N.; Bayat, M.; Das, A.K.; Fotouhi, M.; Pournaghi, S.M.; Doostari, M. LAPTAS: Lightweight anonymous privacy-preserving three-factor authentication scheme for WSN-based IIoT. *Wirel. Netw.* **2021**, *27*, 1389–1412.
18. Tanveer, M.; Abbas, G.; Abbas, Z.H.; Waqas, M.; Muhammad, F.; Kim, S. S6AE: Securing 6LoWPAN using authenticated encryption scheme. *Sensors* **2020**, *20*, 2707. [CrossRef]
19. Wu, D.; Zhou, C. Fault-tolerant and scalable key management for smart grid. *IEEE Trans. Smart Grid* **2011**, *2*, 375–381. [CrossRef]
20. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Li, X.; Sangaiah, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Gener. Comput. Syst.* **2018**, *81*, 557–565. [CrossRef]
21. Abbasinezhad-Mood, D.; Nikooghadam, M. An anonymous ECC-based self-certified key distribution scheme for the smart grid. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7996–8004. [CrossRef]
22. 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]
23. Wazid, M.; Das, A.K.; Kumar, N.; Alazab, M.; Designing Authenticated Key Management Scheme in 6G-enabled Network in a Box Deployed for Industrial Applications. *IEEE Trans. Ind. Inform.* **2020**, *17*, 7174–7184. [CrossRef]
24. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably secure authenticated key agreement scheme for smart grid. *IEEE Trans. Smart Grid* **2016**, *9*, 1900–1910. [CrossRef]
25. Xie, S.; Zhang, F.; Lin, H.; Tian, Y. A New Secure and Anonymous Metering Scheme for Smart Grid Communications. *Energies* **2019**, *12*, 4751. [CrossRef]
26. Abbasinezhad-Mood, D.; Nikooghadam, M. An ultra-lightweight and secure scheme for communications of smart meters and neighborhood gateways by utilization of an ARM Cortex-M microcontroller. *IEEE Trans. Smart Grid* **2017**, *9*, 6194–6205. [CrossRef]
27. Gueron, S.; Jha, A.; Nandi, M. COMET: COunter Mode Encryption with authentication Tag. Second Round Candidate of the NIST LWC Competition, 2019. Available online: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/comet-spec.pdf (accessed on 16 February 2023).
28. Nicanfar, H.; Jokar, P.; Beznosov, K.; Leung, V.C. Efficient authentication and key management mechanisms for smart grid communications. *IEEE Syst. J.* **2013**, *8*, 629–640. [CrossRef]
29. Bera, B.; Saha, S.; Das, A.K.; Vasilakos, A.V. Designing Blockchain-Based Access Control Protocol in IoT-Enabled Smart-Grid System. *IEEE Internet Things J.* **2020**, *8*, 5744–5761. [CrossRef]
30. Chaudhry, S.A.; Alhakami, H.; Baz, A.; Al-Turjman, F. Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. *IEEE Access* **2020**, *8*, 101235–101243. [CrossRef]
31. 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]
32. Srinivas, J.; Das, A.K.; Li, X.; Khan, M.K.; Jo, M. Designing anonymous signature-based authenticated key exchange scheme for Internet of Things-enabled smart grid systems. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4425–4436. [CrossRef]
33. Irshad, A.; Chaudhry, S.A.; Alazab, M.; Kanwal, A.; Zia, M.S.; Zikria, Y.B. A secure demand response management authentication scheme for smart grid. *Sustain. Energy Technol. Assess.* **2021**, *48*, 101571. [CrossRef]
34. Bera, B.; Das, A.K.; Sutrala, A.K. Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of Drones environment. *Comput. Commun.* **2020**, *166*, 91–109. [CrossRef]
35. Chaudhry, S.A.; Yahya, K.; Karuppiah, M.; Kharel, R.; Bashir, A.K.; Zikria, Y.B. GCACS-IoD: A certificate based generic access control scheme for Internet of Drones. *Comput. Netw.* **2021**, *191*, 107999. [CrossRef]