sensors

MDPI

Article

Lightweight Hash-Based Authentication Protocol for Smart Grids

Sangjin Kook [1], Keunok Kim [1], Jihyeon Ryu [2], Youngsook Lee [3] and Dongho Won [1,*]

[1] Department of Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si 16419, Republic of Korea; sangjinkook@gmail.com (S.K.); kimkeunok@gmail.com (K.K.)
[2] School of Computer and Information Engineering, Kwangwoon University, Seoul-si 01897, Republic of Korea; jhryu@kw.ac.kr
[3] Department of Computer Information Security, Howon University, 64 Impi-myeon, Howondae 3-gil, Gunsan-si 54058, Republic of Korea; ysooklee@howon.ac.kr
* Correspondence: dhwon@security.re.kr

Abstract: Smart grids integrate information and communications technology into the processes of electricity production, transportation, and consumption, thereby enabling interactions between power suppliers and consumers to increase the efficiency of the power grid. To achieve this, smart meters (SMs) are installed in households or buildings to measure electricity usage and allow power suppliers or consumers to monitor and manage it in real time. However, SMs require a secure service to address malicious attacks during memory protection and communication processes and a lightweight communication protocol suitable for devices with computational and communication constraints. This paper proposes an authentication protocol based on a one-way hash function to address these issues. This protocol includes message authentication functions to address message tampering and uses a changing encryption key for secure communication during each transmission. The security and performance analysis of this protocol shows that it can address existing attacks and provides 105,281.67% better computational efficiency than previous methods.

Keywords: smart grid authentication; lightweight user authentication; hash-based authentication

Citation: Kook, S.; Kim, K.; Ryu, J.; Lee, Y.; Won, D. Lightweight Hash-Based Authentication Protocol for Smart Grids. *Sensors* **2024**, *24*, 3085. https://doi.org/10.3390/s24103085

Academic Editor: Hossam A. Gabbar

Received: 15 April 2024
Revised: 4 May 2024
Accepted: 9 May 2024
Published: 13 May 2024

1. Introduction

A smart grid (SG) is an advanced power-grid system that integrates information and communications technologies to enhance the efficiency and reliability of electricity production, transportation, and consumption [1]. These systems enable intelligent demand management, the linkage of new and renewable energies, and electric vehicle charging through real-time information exchange between suppliers and consumers [2]. As the sales of electric vehicles and power consumption increase significantly every year, SGs and related security issues have become more important [3]. One of the key components of the SG is the deployment of smart meters (SMs) in households and buildings [4–10], which enable the real-time monitoring and management of electricity usage by both power suppliers and consumers.

Information monitored in real time is important for security [11]. For example, if electricity usage is leaked outside, an attacker can determine whether a house is empty, and by analyzing this information, they can also determine the living patterns of the individual. This is an important personal privacy issue, as individuals may become involved in crimes or undesirable events against their will. In another example, problems may occur if electricity usage is falsified. Attackers may attempt to make financial gains by reducing their own usage; conversely, attackers may increase their usage and cause inconvenience to neighbors with whom they do not get along.

However, the security of SMs and their communication protocols is of paramount importance for preventing malicious attacks and ensuring the integrity and confidentiality of data. To address these security concerns, this paper introduces a hash-based lightweight

authentication scheme specifically designed for SG environments. The proposed authentication scheme aims to provide a secure and efficient method for authenticating communication between SMs and power suppliers while considering the computational and communication constraints of these devices.

The primary objective of the authentication scheme is to ensure the following:

- *Secure memory protection:* The scheme addresses the need for secure memory protection in SMs to safeguard against the unauthorized access and tampering of sensitive data stored within the devices.
- *Robust communication security:* By employing a lightweight communication protocol, the scheme ensures secure communication between SMs and power suppliers, protecting against eavesdropping, message tampering, and replay attacks.
- *Efficient computational requirements:* Recognizing the resource limitations of SMs, the proposed scheme aims to minimize the computational overhead, ensuring efficient authentication without compromising security.

Recently, researchers [4–10] have conducted studies on the security of SMs and their communication protocols; however, several of these studies [4–10] have failed to satisfy the various security requirements outlined earlier. In 2021, Aghapour et al. [10] published a study on lightweight cryptography. However, our study demonstrates that Aghapour et al. [10]'s study has vulnerabilities, such as inferred data reports, extracted keys, and the potential for message recovery. Therefore, a new authentication protocol is required for SGs.

We propose a scheme that satisfies these requirements. Our scheme is designed to provide secure memory protection and has been verified to satisfy ten security requirements, ensuring robust communication security. Our scheme is based on a one-way hash function and utilizes message authentication functions and changing encryption keys to satisfy efficient computational requirements. Through a comprehensive security and performance analysis, the proposed scheme demonstrates its effectiveness in addressing existing attacks and achieving better computational efficiency than previous studies.

The remainder of this paper is organized as follows: In Section 3, we present the hash functions of the system and attack models. The target scheme is introduced in Section 4. Section 5 describes the limitations of the proposed scheme. The proposed scheme is presented in Section 6. In Section 7, we provide formal and informal security analyses. In Section 8, we present a performance analysis of the proposed scheme, and in Section 9, we discuss the results. Finally, we conclude this paper in Section 10.

2. Related Work

In the field of SG security, several studies have proposed lightweight authentication schemes that address the unique challenges and requirements of SG environments.

In 2018, Mahomood et al. [4] proposed an authentication scheme based on elliptic curve cryptography (ECC) to satisfy the complex security requirements of SGs. In 2021, Sadhukhan et al. [6] introduced an ECC-based SG communication authentication scheme comprising a trusted authority, an SM, and a service provider. Sadhukhan et al. [6]'s scheme defends against impersonation attacks, which Mahomood et al. [4]'s scheme fails to protect against, and additionally satisfies, SM anonymity and data confidentiality. In 2021, Sureshkumar et al. [7] designed a scheme for the communication between service providers and SMs. However, Sureshkumar's method is vulnerable because it does not use a one-time pad key. Furthermore, in 2023, Hu et al. [5] pointed out that Mahomood et al. [4]'s scheme does not ensure user anonymity and is vulnerable to ephemeral secret leakage attacks, and hence proposed an authentication and key agreement scheme for SGs with enhanced security based on ECC.

Recently, several authentication schemes for SG environments that do not use ECC have been proposed. In 2020, Kaveh and Mosavi [8] introduced an authentication scheme for SG environments using a physically unclonable function to counteract attacks involving physical replication or damage. Recently, Tanveer and Alasmary [9] proposed an

authentication scheme for SG environments using the new hash function "Esch256". In 2021, Aghapour et al. [10] proposed a fully lightweight two-way communication scheme for SG environments. Aghapour et al. [10] utilized only one-way hash functions and XOR operations for authentication between the participants, making their scheme the most lightweight one. However, in this study, we identified a critical vulnerability in Aghapour et al. [10]'s scheme. Their scheme enables the extraction of keys when data reports are inferred, and messages can be recovered based on the extracted key.

3. Preliminaries

In this section, the hash function, system model, and attack model are described. The details are as follows:

3.1. Hash Function

In this study, we adopt a hash function as an algorithm for verifying messages or for generating keys [12–14]. Hash functions are widely known to have the following four main characteristics:

- *Compute a hash function efficiently:* The calculation of the hash value by the hash function must be fast, regardless of the size of the input data.
- *Preimage resistance:* For the hash function $h(\cdot)$, given $y = h(x)$, it should be computationally infeasible to find x.
- *Second preimage resistance:* For the hash function $h(\cdot)$, given x, it should be computationally infeasible to find another $x_2 \neq x$ such that $h(x) = h(x_2)$.
- *Collision resistance:* For the hash function $h(\cdot)$, it should be computationally infeasible to find x_1 and x_2, where $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$.

Furthermore, recent studies have shown that widely used hash functions, such as MD4, MD5, SHA1, RIPEMD-160, SHA2-256, and SHA-512, are prone to issues, such as collision resistance, second preimage resistance, and no length extension, owing to advances in computational speed [15]. Therefore, we assume that the hash function used in our scheme is the most recently developed and has yet to be found to be vulnerable: SHA3-256.

3.2. System Model

We proposed a scheme for communication between SMs and power supplier servers in an SG environment [16,17]. The two nodes that participate in the communication possess a hierarchical communication model as illustrated in Figure 1.

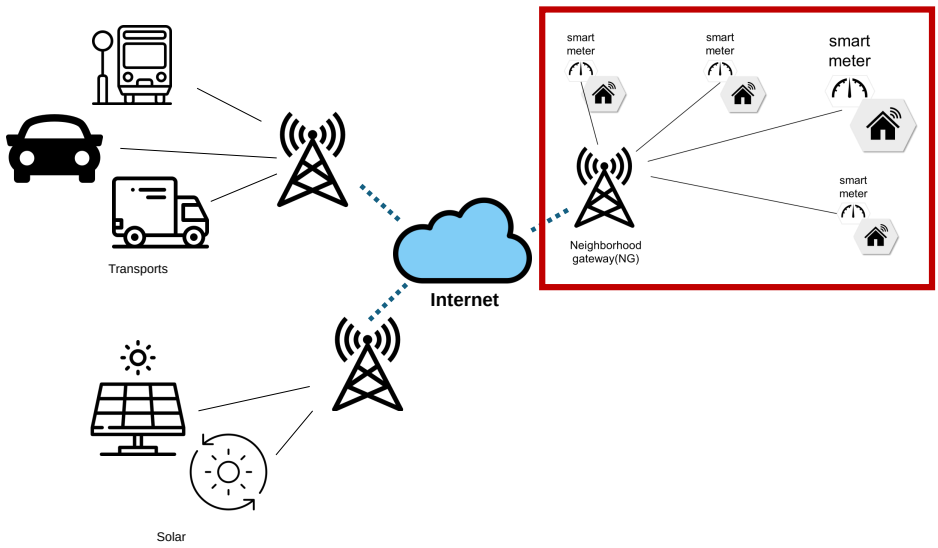

Figure 1. A system model where the smart meter and neighborhood gateway communicate with other neighborhoods' edge nodes over the internet.

Smart grids provide bidirectional services; thus, automated communication occurs over public channels. If certain nodes provide incorrect status and situational information, the microgrid controlled by these nodes is at risk of being compromised [18]. Furthermore, while current smart grids are easily deployable and modifiable, they must be carefully designed due to the various existing cyber threats they face [19].

Smart grids have long been subject to attacks worldwide. In 2009, a senior analyst at the US CID reported that Russian and Chinese cyber spies had penetrated the US power grid [20]. In December 2016, Russia attacked Ukraine's energy grid, which resulted in opening the circuit breakers of Ukraine's energy grid and caused a power outage for about an hour [21].

Attacks on smart grids typically originate from the information sent from endpoint devices to common nodes such as neighborhood gateways. Attackers who infiltrate the smart grid network through these devices can then exploit vulnerabilities in the central control system to take over the smart grid. Subsequently, attackers may attempt attacks such as power shutdowns and personal data breaches through the control system, causing damage. To defend against such attacks, the FERC uses emergency orders and sanctions related to the cyber security of the power infrastructure [22], while NIST sets standards to ensure all systems in the smart grid are interoperable [23].

The details regarding the participating smart meters (SMs) and neighborhood gateways (NG) are as follows:

- *Smart meter (SM):* An electronic device that measures the consumption of utilities, such as electricity, gas, and water, collecting data in real time. It communicates with the neighborhood gateway to transmit data reports. Users utilize SMs to monitor their energy usage.
- *Neighborhood gateway (NG):* A neighborhood gateway is configured within a neighborhood area network and communicates regularly with dozens to hundreds of smart meters. For example, it could be installed in a commercial building's technical room, where it serves the role of transmitting data to a central energy management system, or it might be placed within a home to monitor the household's energy consumption. In the case of a residential gateway, it could be connected via Bluetooth, Zigbee, or Wi-Fi, and typically supports a capacity of 128 MB or more [24,25]. At a minimum, the gateway must store the information from the smart meter until it can be sent to the cloud or the company. The neighborhood gateway enables smart meters to exchange information with the cloud or the company. It requests data from each SM and collects their data. The neighborhood gateway checks the confidentiality and integrity of the data collected from the SMs.

3.3. Attack Model

We propose a scheme based on the threat model suggested by Dolev–Yao [26,27]. The main characteristics of the Dolev–Yao model [26] are as follows:

- The attacker eavesdrops on all the transmission packets used in the public channel.
- The attacker attempts to decrypt the eavesdropped transmission packets to obtain the values (data report, message, etc.) intended for transmission through communication.
- The attacker attempts to alter the messages used in communication by performing a man-in-the-middle attack.
- The attacker attempts a replay attack.

In this paper, we propose a scheme that defends against these attacks and demonstrate its resistance to them.

4. Review of Aghapour et al.'s Scheme [10]

In this section, we introduce the target scheme suggested by Aghapour et al. [10]. Their scheme consists of an initialization phase and a secure communication phase.

4.1. Initialization Phase

In Aghapour et al. [10]'s scheme, at this stage, each j-th SM_j registers its identity ID_j with a neighborhood gateway (NG). NG then transmits an initial secret key value K_0^j to each SM over a secure channel. Subsequently, NG stores the pair of the SM identity and secret key (ID_j, K_0^j) in its database, and each SM SM_j stores the initial secret key value K_j^0 in its memory.

4.2. Secure Communication Phase

In the stage proposed by Aghapour et al. [10] , message authentication between the j-th SM SM_j and NG occurs over a public channel. The details are as follows.

4.2.1. First Authentication

1. NG generates the random number r_i^j for SM_j. NG computes $A_i^j = ((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus K_i^j$, $V_i^j = H(m_i^j \parallel r_i^j \parallel ID_j \parallel T_{NG} \parallel K_i^j)$, where m_i^j is the i-th message for SM_j, T_{NG} is a timestamp of NG, and $H(\cdot)$ is a one-way hash function. NG sends a message $M_1 = \{A_i^j, V_i^j, T_{NG}, ID_j\}$ to SM_j in the public channel.
2. SM_j receives the message $M_1 = \{A_i^j, V_i^j, T_{NG}, ID_j\}$ from NG, and computes $(m_i^j \oplus r_i^j) \parallel r_i^j = A_i^j \oplus K_i^j$ to obtain r_i^j and m_i^j. SM_j verifies $V_i^j = h(m_i^j \parallel r_i^j \parallel ID_j \parallel T_{NG} \parallel K_i^j)$. If it fails to verify the message, SM_j stops the protocol. If its verification succeeds, the authenticity of NG is verified by SM_j, and the first authentication phase ends.

4.2.2. Second Authentication

1. SM_j computes $E_i^j = (h(r_i^j) \parallel D_i^j) \oplus K_i^j$, where D_i^j is the data report from the corresponding SM, and $h(\cdot)$ is a different hash function with $H(\cdot)$. SM_j creates the new key $K_{i+1}^j = H(r_i^j \parallel ID_j \parallel T_j \parallel K_i^j)$, where T_j is a timestamp of SM_j. It replaces the old key K_i^j with K_{i+1}^j. SM_j makes the verification $V_i^{'j} = H(D_i^j \parallel r_i^j \parallel ID_j \parallel T_j \parallel K_{i+1}^j)$ and sends a message $M_2 = \{E_i^j, V_i^{'j}, T_j\}$ to NG.
2. NG receives the message $M_2 = \{E_i^j, V_i^{'j}, T_j\}$ from SM_j and computes $(h(r_i^j) \parallel D_i^j) = E_i^j \oplus K_i^j$. NG computes $K_{i+1}^j = H(r_i^j \parallel ID_j \parallel T_j \parallel K_i^j)$. NG verifies $V_i^{'j} = H(D_i^j \parallel r_i^j \parallel ID_j \parallel T_j \parallel K_{i+1}^j)$, and if its verification succeeds, NG compares D_i^j with the existing format and stores K_{i+1}^j in its database.

5. Limitations of Aghapour et al.'s Scheme [10]

We identified a critical vulnerability in the scheme proposed by Aghapour et al. [10] as previously described. In this section, we discuss the vulnerabilities identified in Aghapour et al. [10]'s scheme. The details are as follows:

5.1. Inferrability of the Data Report

We assume that the data report D_i^j can be inferred because it has a similar format. This is likely because the data report D_i^j, such as electricity usage, tends to be within a certain range of the actual values.

5.2. Inferrability of the Message

We can obtain the values of A_i^j and E_i^j using the values in M_1 and M_2 transmitted over the public channel. Using the obtained A_i^j and E_i^j values, we derive the following equation:

$$A_i^j \oplus E_i^j \tag{1}$$

$$= (((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus K_i^j) \oplus ((h(r_i^j) \parallel D_i^j) \oplus K_i^j) \tag{2}$$

$$= ((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus (h(r_i^j) \parallel D_i^j) \quad (3)$$

Here, we assume that we can estimate D_i^j according to Section 5.1; thus, we obtain the value of r_i^j. In addition, we obtain $h(r_i^j)$ using r_i^j. Finally, we can derive the message m_i^j using the previously obtained r_i^j, $h(r_i^j)$, and D_i^j.

5.3. Extraction of the Secret Key

In Section 5.2, we obtained r_i^j, m_i^j, and D_i^j. Using these variables, we derived the secret key value K_i^j using A_i^j. This is derived as follows:

$$A_i^j = ((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus K_i^j \quad (4)$$

$$K_i^j = ((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus A_i^j \quad (5)$$

6. Proposed Scheme

In this section, we propose enhanced hash-based authentication in SGs to address the vulnerabilities identified in Section 5. The notations used in this paper are explained in Table 1. The details are as follows:

Table 1. Notations used in this paper.

Notations	Description
SM_j	j-th smart meter
NG	Neighborhood gateway
ID_j	SM_j's identification
m_i^j	i-th message for SM_j
D_i^j	Data report of i-th SM_j
V_i^j, $V_i'^j$	Verification
K_i^j	i-th secret key for SM_j
r_i^j	i-th random number for SM_j
$h(\cdot)$, $H(\cdot)$	One-way hash function
$X \parallel Y$	Concatenation operator
$\oplus$	Bitwise XOR operator
T_{NG}, T_j	Timestamp for NG and SM_j

6.1. Initialization Phase

In this phase, NG verifies the identity of each SM and assigns an initial secret key individually. The details are shown in Figure 2.

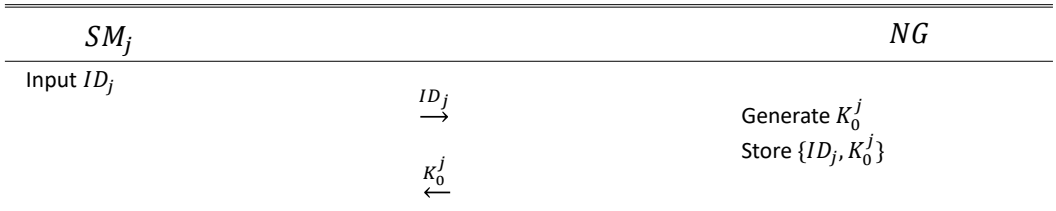

Figure 2. The phase of registering the identity ID_j of the smart meter SM_j with the neighborhood gateway NG proposed in this study.

1. We denote the j-th SM as SM_j. At this time, SM_j selects its own identity information. When the identity chosen by SM_j is denoted as ID_j, SM_j transmits the ID_j information to NG through a secure channel.
2. NG receives the identity information of each SM through a secure channel. Assuming that it receives the identity ID_j of the j-th SM, NG generates an initial secret key K_0^j

for communication with SM_j. NG then stores the pair ID_j, K_0^j in its database. NG transmits the generated K_0^j to SM_j through a secret channel, and SM_j receives and stores the secret key K_0^j.

6.2. First Secure Communication Phase

In this phase, NG sends information to the j-th SM SM_j through a public channel, protecting it from external leakage using hashing and concatenation operations. SM_j checks the message received from NG and verifies its integrity. The details are presented in Figure 3.

SM_j		NG
		Generate r_i^j Compute $A_i^j = \left(\left(m_i^j \oplus r_i^j\right) \| r_i^j\right) \oplus K_i^j$ $V_i^j = H\left(m_i^j \| r_i^j \| ID_j \| T_{NG} \| K_i^j\right)$
Compute $\left(m_i^j \oplus r_i^j\right) \| r_i^j = A_i^j \oplus K_i^j$ Verify $V_i^j = H\left(m_i^j \| r_i^j \| ID_j \| T_{NG} \| K_i^j\right)$	$\xleftarrow{\{A_i^j, V_i^j, T_{NG}, ID_j\}}$	

Figure 3. The first authentication phase between smart meter SM_j and neighborhood gateway NG proposed in this study.

1. To securely send a message to SM_j, NG generates a random number r_i^j and a timestamp T_{NG}. To protect the message m_i^j from external leakage, NG performs the following operations: $A_i^j = ((m_i^j \oplus r_i^j) \| r_i^j) \oplus K_i^j$, $V_i^j = H(m_i^j \| r_i^j \| ID_j \| T_{NG} \| K_i^j)$. NG then transmits $M_1 = \{A_i^j, V_i^j, T_{NG}, ID_j\}$ to SM_j through a public channel.
2. Upon receiving $M_1 = \{A_i^j, V_i^j, T_{NG}, ID_j\}$ from NG, SM_j checks if the timestamp T_{NG} is within an appropriate range and performs the following operations to verify the message: $(m_i^j \oplus r_i^j) \| r_i^j = A_i^j \oplus K_i^j$. SM_j computes m_i^j using the extracted r_i^j: $m_i^j = (m_i^j \oplus r_i^j) \oplus r_i^j$. Then, it computes $V_i^j = H(m_i^j \| r_i^j \| ID_j \| T_{NG} \| K_i^j)$ to verify the integrity of the message. If the verification fails, the protocol is immediately halted. If the verification succeeds, the next phase proceeds.

6.3. Second Secure Communication Phase

In this phase, SM_j protects and transmits its data report via a public channel to prevent external leakage. NG verifies the data report received from SM_j and checks its integrity. The details are presented in Figure 4.

1. To securely send the data report D_i^j to NG, SM_j generates a timestamp T_j and performs the following operations: $E_i^j = (h(r_i^j) \| h(K_i^j) \oplus D_i^j) \oplus K_i^j$. It then computes the new key value $K_{i+1}^j = H(r_i^j \| ID_j \| T_{NG} \| K_i^j)$ and performs the verification $V_i^{'j} = H(m_i^j \| r_i^j \| ID_j \| T_j \| K_{i+1}^j)$. Then, SM_j transmits $M_2 = \{E_i^j, V_i^{'j}, T_j\}$ to NG through a public channel.
2. Upon receiving $M_2 = \{E_i^j, V_i^{'j}, T_j\}$ from SM_j, NG checks if the timestamp T_j is within an appropriate range and performs the following operations for verification D_i^j: $(h(r_i^j) \| h(K_i^j) \oplus D_i^j) = E_i^j \oplus K_i^j$, $D_i^j = (h(K_i^j) \oplus D_i^j) \oplus h(K_i^j)$. NG compares D_i^j with existing reports, and if it matches the established format, it is accepted. When NG computes $K_{i+1}^j = H(r_i^j \| ID_j \| T_{NG} \| K_i^j)$ and checks the verification $V_i^{'j} = H(m_i^j \| r_i^j \| ID_j \| T_j \| K_{i+1}^j)$, if the verification is successful, K_{i+1}^j replaces the existing K_i^j.

SM_j		NG
Compute $E_i^j = \left(h\left(r_i^j\right) \| h\left(K_i^j\right) \oplus D_i^j\right) \oplus K_i^j$ $K_{i+1}^j = H\left(r_i^j \| ID_j \| T_{NG} \| K_i^j\right)$ $V_i'^j = H\left(m_i^j \| r_i^j \| ID_j \| T_j \| K_{i+1}^j\right)$	$\xrightarrow{\{E_i^j, V_i'^j, T_j\}}$	Compute $\left(h\left(r_i^j\right) \| h\left(K_i^j\right) \oplus D_i^j\right) = E_i^j \oplus K_i^j$ $K_{i+1}^j = H\left(r_i^j \| ID_j \| T_{NG} \| K_i^j\right)$ Verify $V_i'^j = H\left(m_i^j \| r_i^j \| ID_j \| T_j \| K_{i+1}^j\right)$ Store $\left(ID_j, K_{i+1}^j\right)$

Figure 4. The second authentication phase between smart meter SM_j and neighborhood gateway NG proposed in this study.

7. Security Analysis of the Proposed Scheme

In this section, we describe the formal and informal security analyses of the proposed scheme. The formal security analysis is conducted using ProVerif 2.05 [28], whereas the informal security analysis includes ten different analyses, including providing mutual authentication and resisting replay attacks.

7.1. Formal Security Analysis

In this section, we discuss the results of a formal analysis of our scheme conducted using ProVerif. The analysis using ProVerif demonstrates the results of verifying and analyzing the security of the proposed scheme as in several recent studies [29–32].

We define two types of channels: privateChannel and publicChannel. The reason for setting the publicChannel as private is discussed later when explaining the SM_j and NG processes. The constants are set with the SM_j ID and the NG unique value as N. Functions define XOR, concatenate, and two hash operations, and events for SM_j and NG are defined for both the first and second authentication phases. The detailed information is provided in Table 2.

The initial and authentication phases of SM_j and NG are listed in Tables 3 and 4. The initial phases of SM_j and NG are transmitted through the privateChannel. Subsequently, the first authentication begins. However, the process of omitting the part where r is concatenated cannot be implemented using ProVerif. Therefore, to modify it such that NG sends r to SM_j, the publicChannel is set to private to verify the formality.

We verify the results in Table 5 using the queries listed in Table 6. The results are as follows:

- Query inj-event(EVENT) ==> inj-event(EVENT) is true.
- Query not attacker(K) is true.

"Query inj-event(EVENT) ==> inj-event(EVENT) is true" indicates that the event has been verified, and the authentication is successful. This indicates that the event occurred as expected, and under the specified conditions, the authentication mechanism functioned correctly. "Query not attacker(K) is true" indicates that the result of this query is true, which indicates that the attacker could not discover the keys within the array.

Table 2. ProVerif code for defining values and functions.

```
(*—-channels—-*)
free privateChannel:channel [private].
free publicChannel:channel [private].

(*—-constants—-*)
free ID:bitstring [private].
free N:bitstring [private].

(*—-shared key—-*)
free K:bitstring [private].

(*—-functions—-*)
fun xor(bitstring, bitstring):bitstring.
fun concat(bitstring, bitstring):bitstring.
fun h(bitstring):bitstring.
fun H(bitstring):bitstring.
equation forall a:bitstring, b:bitstring; xor(xor(a, b), b) = a.

(*—-events—-*)
event startfstS(bitstring).
event endfstS(bitstring).
event startfstN(bitstring).
event endfstN(bitstring).
event start2ndS(bitstring).
event end2ndS(bitstring).
event start2ndN(bitstring).
event end2ndN(bitstring).
```

Table 3. ProVerif code for the SM.

```
(*—-SMj process—-*)
let SMj =
out(privateChannel, (ID));
in(privateChannel, (XK:bitstring));
event startfstS(ID);
in(publicChannel, (XA:bitstring, XV:bitstring, XT:bitstring, XXID:bitstring, Xr:bitstring));
let P = xor(xor(XA, XK), XA) in
let Xm = xor(P, Xr) in
let XXV = H(concat(concat(Xm, Xr), concat(concat(XXID, XT), XK))) in
event endfstS(ID);
event start2ndS(ID);
if XV = XXV then
new Tj:bitstring;
new D:bitstring;
let E = xor(xor(concat(h(Xr), h(XK)), D), XK) in
let newK = H(concat(concat(Xr, XXID), concat(XT, XK))) in
let Vp = H(concat(concat(Xm, Xr), concat(concat(XXID, Tj), newK))) in
out(publicChannel,(E, Vp, Tj));
event end2ndS(ID).
```

Table 4. ProVerif code for the neighborhood gateway.

```
(*—-NG process—-*)
let NG =
in(privateChannel, (XID:bitstring));
out(privateChannel, (K));
event startfstN(N);
new r:bitstring;
```

Table 4. *Cont.*

```
new m:bitstring;
new T:bitstring;
let A = xor(xor(m, r), K) in
let V = H(concat(concat(m, r), concat(concat(XID, T), K))) in
out(publicChannel,(A, V, T, XID, r));
event endfstN(N);
event start2ndN(N);
in(publicChannel,(XE:bitstring, XVp:bitstring, XTj:bitstring));
let PP = xor(XE, K) in
let XD = xor(PP, concat(h(r), h(K))) in
let XnewK = H(concat(concat(r, XID), concat(T, K))) in
let XXVp = H(concat(concat(m, r), concat(concat(XID, XTj), XnewK))) in
if XVp = XXVp then
event end2ndN(N).
```

Table 5. ProVerif query results.

```
Query inj-event(endfstS(IDj)) ==> inj-event(startfstS(IDj)) is true.
Query inj-event(end2ndS(IDj)) ==> inj-event(start2ndS(IDj)) is true.
Query inj-event(endfstN(IDj)) ==> inj-event(startfstN(IDj)) is true.
Query inj-event(end2ndN(IDj)) ==> inj-event(start2ndN(IDj)) is true.
Query not attacker(K[]) is true.
```

Table 6. ProVerif code for queries.

```
(*—-queries—-*)
query IDj:bitstring; inj-event(endfstS(IDj)) ==> inj-event(startfstS(IDj)).
query IDj:bitstring; inj-event(end2ndS(IDj)) ==> inj-event(start2ndS(IDj)).
query IDj:bitstring; inj-event(endfstN(IDj)) ==> inj-event(startfstN(IDj)).
query IDj:bitstring; inj-event(end2ndN(IDj)) ==> inj-event(start2ndN(IDj)).
query attacker(K).

(*—-process—-*)
process
((!SMj) | (!NG))
```

7.2. Informal Security Analysis

In this section, we present an informal verification of the proposed scheme. Table 7 shows a comparison with previous studies [5,7,10,33]. We conducted ten informal verifications, and the details are as follows.

Table 7. Comparison of security features.

Security Features	Sureshkumar et al. [7]	Garg et al. [33]	Hu et al. [5]	Aghapour et al. [10]	Ours
Provide Mutual Authentication	O	O	O	O	O
Resist Replay Attack	O	O	O	O	O
Resist Smart Meter Impersonation Attack	O	O	O	O	O
Resist Extraction of the Secret Key	O	O	O	O	O
Resist Inferrability of the Message	O	O	O	X	O
Resist Message Altering	O	O	O	X	O
Resist Injection Attack	O	O	O	O	O
Provide Forward Secrecy	O	O	O	O	O
Provide One-time Pad Key	X	O	O	O	O
Resist Man-in-the-Middle Attack	O	O	O	X	O

7.2.1. Provide Mutual Authentication

The proposed scheme verifies the integrity of the message received by SM_j from NG during the first authentication phase and the integrity of the message received by NG from SM_j during the second authentication phase. Therefore, the proposed scheme provides mutual authentication.

7.2.2. Resist Replay Attack

In the proposed scheme, the decision to proceed with the subsequent operations is based on verifying the timestamps T_{NG} and T_j transmitted during the first and second authentication phases, respectively. Therefore, the proposed scheme is resistant to replay attacks.

7.2.3. Resist Smart Meter Impersonation Attack

For an attacker to impersonate SM_j, they must be able to deceive NG into passing the $V_i^{'j}$ verification during the second authentication phase. To do this, the attacker must obtain the information necessary to generate $V_i^{'j}$, which includes m_i^j, r_i^j, and K_{i+1}^j. The information required to generate K_{i+1}^j includes r_i^j and K_i^j. As the attacker cannot calculate these values from the information A_i^j and V_i^j available through the public channel, the attacker cannot impersonate SM_j.

7.2.4. Resist Extraction of the Secret Key

The only way for an attacker to obtain K_i^j is by already knowing m_i^j and r_i^j, and then performing the operation $((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus A_i^j$ or by intercepting it from the private channel. Assuming that interception from the private channel is not possible and because m_i^j and r_i^j are neither directly disclosed nor calculated, an attacker cannot obtain K_i^j in our scheme.

7.2.5. Resist Inferrability of the Message

The message m_i^j is extracted by performing an XOR operation between A_i^j and K_i^j. However, as there is no way for an attacker to obtain K_i^j, messages cannot be inferred in our scheme.

7.2.6. Resist Message Altering

In our scheme, message m_i^j and data report D_i^j are included in the information contained in A_i^j and E_i^j, respectively. To verify the integrity of each message m_i^j and data report D_i^j, ensuring they have not been altered, V_i^j and $V_i^{'j}$ are used for verification. Therefore, if an attacker arbitrarily changes the message to create A_i^j and E_i^j and attempts to extract the message, it will not pass the verification. Each message and data report can only be verified with the encryption key K_i^j; however, as K_i^j cannot be extracted by the attacker, the attacker cannot verify the message and data report. Therefore, the proposed scheme resists message alterations.

7.2.7. Resist Injection Attack

In the authentication phases, as message m_i^j and data report D_i^j to be transmitted contain the verification variables V_i^j and $V_i^{'j}$, it is impossible to perform a data injection attack on the original message and data report. This prevents SQL injections, cross-site scripting, code injections, and other related attacks from becoming feasible.

7.2.8. Provide forward Secrecy

Our scheme employs a method for hashing values that include K_i^j to generate K_{i+1}^j. Even if the future key K_{i+1}^j is compromised, it is computed as $K_{i+1}^j = H(r \parallel ID_j \parallel T_{NG} \parallel K_i^j)$, which makes it impossible to deduce the value of K_i^j because of the one-way nature of the hash function. Thus, the proposed scheme provides forward secrecy.

7.2.9. Provide One-Time Pad Key

Our scheme employs a method for hashing values that include K_i^j to generate the new key K_{i+1}^j. Thus, the proposed scheme provides a one-time pad key.

7.2.10. Resist Man-in-the-Middle Attack

In the scenario where an attacker accesses the public channel used during the first and second authentication phases of our scheme to carry out a man-in-the-middle attack, the only information they can obtain are $M_1 = \{A_i^j, V_i^j, T_{NG}, ID_j\}$ and $M_2 = \{E_i^j, V_i^{'j}, T_j\}$. These values include the smart meter's identity information and timestamps T_{NG} and T_j, but among the $A_i^j = ((m_i^j \oplus r_i^j) \parallel r_i^j) \oplus K_i^j$, $V_i^j = H(m_i^j \parallel r_i^j \parallel ID_j \parallel T_{NG} \parallel K_i^j)$, $V_i^{'j} = H(m_i^j \parallel r_i^j \parallel ID_j \parallel T_j \parallel K_{i+1}^j)$, and $E_i^j = (h(r_i^j) \parallel h(K_i^j) \oplus D_i^j) \oplus K_i^j$ information, the V_i^j and $V_i^{'j}$ values are hashed and therefore unusable. Even if the attacker can see the A_i^j or E_i^j values, without knowing the session key, which changes with each session, they cannot recreate these values. Therefore, a man-in-the-middle attack is not feasible.

8. Performance Analysis of the Proposed Scheme

In this section, we compare the performance of our paper with related studies. Performance analysis was conducted in the environment of Table 8. The time taken for a hash algorithm was measured as 0.012 ms for symmetric key encryption, decryption was 0.19 ms, and for scalar multiplication in the field, it was 28.03 ms. The computational overhead of the authentication phases for our scheme and related studies [5,7,10,33] is presented in Table 9.

Table 8. Development environment.

Item	Value
CPU	Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz 1.99 GHz (Intel, Santa Clara, CA, USA)
RAM	16.0 GB
OS	Windows 10 Home
Software	JDK 17
Security level	secp521r1 ECC

Table 9. Comparisons of computational costs (ms).

Schemes	Hu et al. [5]	Garg et al. [33]	Sureshkumar et al. [7]	Aghapour et al. [10]	Ours
NG, SP	$4T_m + 5T_h$ $= 112.18$	$3T_m + 4T_h + 1T_e$ $= 84.328$	$3T_m + 6T_h$ $= 84.162$	$4T_h$ $= 0.048$	$5T_h$ $= 0.06$
Smart Meter(SM)	$4T_m + 5T_h$ $= 112.18$	$3T_m + 4T_h + 1T_e$ $= 84.328$	$1T_m + 4T_h$ $= 28.078$	$4T_h$ $= 0.048$	$5T_h$ $= 0.06$
Total	$8T_m + 10T_h$ $= 224.36$	$6T_m + 8T_h + 2T_e$ $= 168.656$	$4T_m + 10T_h$ $= 112.24$	$8T_h$ $= 0.096$	$10T_h$ $= 0.12$

We compute the performance of our scheme in the environment of Table 8 using five hash functions, resulting in a total computational load of $5T_h$ for the neighborhood gateway and $5T_h$ for the smart meter, totaling $10T_h$ = 0.12 ms. According to our find-

ings, Hu et al. [5]'s scheme requires the neighborhood gateway to perform four field multiplications ($4T_m$) and use $5T_h$. The smart meter operates at $4T_m + 5T_h$, totaling $8T_m + 10T_h = 224.36$ ms. In Garg et al. [33]'s scheme, the neighborhood gateway performs three field multiplications (T_m), four hash function operations (T_h), and one symmetric key encryption (T_e). Additionally, Garg et al.'s smart meter computes at $3T_m + 4T_h + 1T_e$, totaling $6T_m + 8T_h + 2T_e = 168.656$ ms. Similarly, Sureshkumar et al. [7]'s scheme calculates the neighborhood gateway at $3T_m + 6T_h$, and the smart meter at $1T_m + 4T_h$, totaling $4T_m + 10T_h = 112.24$ ms. Furthermore, we confirmed that the vulnerable scheme by Aghapour et al. [10] involves $4T_h$ for both the neighborhood gateway and the smart meter, resulting in a total of $8T_h = 0.096$ ms.

9. Discussion of Performance

Based on Section 8, we quantify and compare how much better our performance is. The formula we use is as follows:

$$(t_1 - t_2)/t_2 \tag{6}$$

According to Formula (6), our scheme demonstrates superior performance by 186,966.67%, 140,546.67%, 93,533.33% and 80.00% compared to Hu et al. [5]'s, Garg et al. [33]'s scheme, Sureshkumar et al. [7]'s scheme and Aghapour et al. [10] scheme. In contrast to other studies [5,7,10,33] which primarily utilize public key or symmetric key cryptography, our scheme mainly uses hash operations to construct lightweight protocols.

According to Table 7, which compares the security aspects of our scheme against others, we found that our scheme performs about 20% worse than Aghapour et al. [10]'s scheme in terms of efficiency. However, our scheme is significantly safer than the proposal by Aghapour et al. [10]. We have developed a scheme that provides a one-time pad key, which Sureshkumar et al. [7]'s scheme failed to do. Moreover, our scheme outperforms the average of the four schemes, including those by Garg et al. [33] and Hu et al. [5], by approximately 105,281.67%.

10. Conclusions

In this paper, we proposed a lightweight authentication scheme for SG environments. Our scheme minimizes computational requirements by using only hash functions and XOR operations, and provides security against ten protocol vulnerabilities that previous studies failed to defend, including the extraction of secret keys and the inferrability of the message. We demonstrate that our scheme satisfies the security requirements using ProVerif, a formal verification tool. Moreover, in terms of performance, our scheme shows a superior computational speed of 105,281.67% compared with other schemes.

Author Contributions: Conceptualization, S.K.; Methodology, K.K., S.K. and Y.L.; Software, K.K. and J.R.; Validation, S.K. and J.R.; Formal analysis, K.K. and Y.L.; Supervision, D.W.; Funding acquisition, D.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00239728).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Jumayev, B.A.; Nazarov, S. Smart Calculation of Heat Energy Supplied by Hot Water. *IEIE Trans. Smart Process. Comput.* **2023**, *12*, 155–161. [CrossRef]
2. Barman, P.; Dutta, L.; Bordoloi, S.; Kalita, A.; Buragohain, P.; Bharali, S.; Azzopardi, B. Renewable energy integration with electric vehicle technology: A review of the existing smart charging approaches. *Renew. Sustain. Energy Rev.* **2023**, *183*, 113518. [CrossRef]
3. Hasan, M.K.; Habib, A.A.; Shukur, Z.; Ibrahim, F.; Islam, S.; Razzaque, M.A. Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. *J. Netw. Comput. Appl.* **2023**, *209*, 103540. [CrossRef]
4. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Xiong, L.; Sangaiah, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. *Future Gener. Comput. Syst.* **2018**, *81*, 557–565. [CrossRef]
5. Hu, S.; Chen, Y.; Zheng, Y.; Xing, B.; Li, Y.; Zhang, L.; Chen, L. Provably secure ECC-based authentication and key agreement scheme for advanced metering infrastructure in the smart grid. *IEEE Trans. Ind. Inform.* **2023**, *19*, 5985–5994. [CrossRef]
6. Sadhukhan, D.; Ray, S.; Obaidat, M.S.; Dasgupta, M. A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography. *J. Syst. Archit.* **2021**, *114*, 101938. [CrossRef]
7. Sureshkumar, V.; An hi, S.; Amin, R.; Selvarajan, N.; Madhumathi, R. Design of robust mutual authentication and key establishment security protocol for cloud-enabled smart grid communication. *IEEE Syst. J.* **2020**, *15*, 3565–3572. [CrossRef]
8. 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]
9. Tanveer, M.; Alasmary, H. LACP-SG: Lightweight authentication protocol for smart grids. *Sensors* **2023**, *23*, 2309. [CrossRef]
10. Aghapour, S.; Kaveh, M.; Mosavi, M.R.; Martín, D. An ultra-lightweight mutual authentication scheme for smart grid two-way communications. *IEEE Access* **2021**, 9, 74562–74573. [CrossRef]
11. Shim, S.; Kim, J.Y.; Hwang, S.W.; Oh, J.M.; Kim, B.K.; Park, J.H.; Hyun, D.J.; Lee, H. A Comprehensive Review of Cyber-physical System (CPS)-based Approaches to Robot Services. *IEIE Trans. Smart Process. Comput.* **2024**, *13*, 69–80. [CrossRef]
12. Ryu, J.; Lee, H.; Lee, Y.; Won, D. SMASG: Secure mobile authentication scheme for global mobility network. *IEEE Access* **2022**, *10*, 26907–26919. [CrossRef]
13. Degefa, F.; Ryu, J.; Kim, H.; Won, D. MES-FPMIPv6: MIH-Enabled and enhanced secure Fast Proxy Mobile IPv6 handover protocol for 5G networks. *PLOS ONE* **2022**, *17*, e0262696. [CrossRef]
14. Lee, H.; Ryu, J.; Won, D. Secure and Anonymous Authentication Scheme for Mobile Edge Computing Environments. *IEEE Int. Things J.* **2024**, *11*, 5798–5815. [CrossRef]
15. Cheval, V.; Cremers, C.; Dax, A.; Hirschi, L.; Jacomme, C.; Kremer, S. Hash Gone Bad: Automated discovery of protocol attacks that exploit hash function weaknesses. In Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23), Anaheim, CA, USA, 9–11 August 2023; pp. 5899–5916.
16. 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]
17. Ye, F.; Qian, Y.; Hu, R.Q. Energy efficient self-sustaining wireless neighborhood area network design for smart grid. *IEEE Trans. Smart Grid* **2014**, *6*, 220–229. [CrossRef]
18. Khurana, H.; Hadley, M.; Lu, N.; Frincke, D.A. Smart-grid security issues. *IEEE Secur. Priv.* **2010**, *8*, 81–85. [CrossRef]
19. Aloul, F.; Al-Ali, A.R.; Al-Dalky, R.; Al-Mardini, M.; El-Hajj, W. Smart grid security: Threats, vulnerabilities and solutions. *Int. J. Smart Grid Clean Energy* **2012**, *1*, 1–6. [CrossRef]
20. Gorman, S. Electricity grid in US penetrated by spies. *Wall Str. J.* **2009**, *8*.
21. Gjesvik, L.; Szulecki, K. Interpreting cyber-energy-security events: Experts, social imaginaries, and policy discourses around the 2016 Ukraine blackout. *Eur. Secur.* **2023**, *32*, 104–124. [CrossRef]
22. Eisen, J.B. Who Regulates the Smart Grid: FERC's Authority over Demand Response Compensation in Wholesale Electricity Markets. *San Diego J. Clim. Energy L.* **2012**, *4*, 69.
23. Gopstein, A.; Nguyen, C.; O'Fallon, C.; Hastings, N.; Wollman, D. *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2021.
24. Casa. EMH Metering. 4 May 2024. Available online: https://emh-metering.com/en/products/smart-meter-gateway/casa/ (accessed on 4 May 2024).
25. Xiaomi. Smart-Home-Hub-2-Xiaomi UK. 4 May 2024. Available online: https://www.mi.com/uk/product/xiaomi-smart-home-hub-2/ (accessed on 4 May 2024).
26. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
27. Park, B.; Kim, J.; McNair, J. ISAS: AAA Protocol-based Handover and Improved Security Methodology through the Integration Security Authentication System Constitute. *IEIE Trans. Smart Process. Comput.* **2023**, *12*, 358–367. [CrossRef]
28. Blanchet, B.; Smyth, B.; Cheval, V.; Sylvestre, M. ProVerif 2.05: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2023. Available online: https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf (accessed on 4 May 2024).
29. Kim, K.; Ryu, J.; Lee, H.; Lee, Y.; Won, D. Distributed and Federated Authentication Schemes Based on Updatable Smart Contracts. *Electronics* **2023**, *12*, 1217. [CrossRef]
30. Kang, T.; Woo, N.; Ryu, J. Enhanced Lightweight Medical Sensor Networks Authentication Scheme Based on Blockchain. *IEEE Access* **2024**, *12*, 35612–35629. [CrossRef]

31. Kim, K.; Ryu, J.; Lee, Y.; Won, D. An improved lightweight user authentication scheme for the internet of medical things. *Sensors* **2023**, *23*, 1122. [CrossRef]
32. Liu, Y.; Cheng, C.; Gu, T.; Jiang, T.; Li, X. A lightweight authenticated communication scheme for smart grid. *IEEE Sens. J.* **2015**, *16*, 836–842. [CrossRef]
33. Garg, S.; Kaur, K.; Kaddoum, G.; Rodrigues, J.J.; Guizani, M. Secure and lightweight authentication scheme for smart metering infrastructure in smart grid. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3548–3557. [CrossRef]