electronics

MDPI

Article

Physical-Unclonable-Function-Based Lightweight Anonymous Authentication Protocol for Smart Grid

Yu Guo [1], Lifeng Li [1], Xu Jin [1], Chunyan An [2], Chenyu Wang [2,*] and Hairui Huang [2]

[1] State Grid Jinhua Electric Power Company, Jinhua 321000, China; guo_yu@zj.sgcc.com.cn (Y.G.); li_lifeng@zj.sgcc.com.cn (L.L.); jin_xu@zj.sgcc.com.cn (X.J.)
[2] China Electric Power Research Institute, Haidian, Beijing 100192, China; anchunyan1@epri.sgcc.com.cn (C.A.); 2021212815@stu.cqupt.edu.cn (H.H.)
* Correspondence: wangchenyu@bupt.edu.cn; Tel.: +86-186-1109-8789

Abstract: In the Internet of Everything era of Web 3.0, smart grid (SG) technology is also developing towards intelligent interconnection of terminal devices. However, in the smart grid scenario, security issues are particularly prominent, especially the openness of wireless sensor networks. Sensor nodes are vulnerable to attacks and other security threats, which makes confirming the legitimacy of access identity and ensuring the secure transmission of data an urgent problem to be solved. At present, although a variety of authentication schemes for smart grid nodes have been proposed, most of them have problems. For example, some cannot achieve forward security. Therefore, this paper aims to solve this problem and proposes a lightweight anonymous authentication protocol based on physical unclonable functions (PUFs), which can implement mutual authentication and session key agreement between gateway nodes and sensor nodes. Compared to five state-of-the-art schemes in security and performance, the proposed scheme achieves all eight of the listed security requirements with lightweight calculation overhead, communication overhead, and storage overhead.

Keywords: authentication; key agreement; physical unclonable function (PUF); smart grid (SG)

Academic Editor: Dorin Petreus

Received: 20 December 2024
Revised: 20 January 2025
Accepted: 2 February 2025
Published: 5 February 2025

Citation: Guo, Y.; Li, L.; Jin, X.; An, C.; Wang, C.; Huang, H. Physical-Unclonable-Function-Based Lightweight Anonymous Authentication Protocol for Smart Grid. *Electronics* **2025**, *14*, 623. https://doi.org/10.3390/electronics14030623

1. Introduction

In the Internet of Everything era of Web 3.0, smart grid (SG) technology is also developing towards intelligent interconnection of terminal devices [1]. As vital components of modern power systems, smart grids play a critical role in optimizing energy distribution [2]. This paper employs wireless sensor network (WSN) technology to construct a network framework that integrates the application of renewable energy resources [3]. By combining advanced information and communication technologies with automated control systems, smart grids enable real-time monitoring, analysis, and management of power systems, thereby significantly enhancing their reliability, security, and economic efficiency [4].

Smart grids are characterized by their high levels of informatization, automation, and adaptability to heterogeneous network environments, which allow them to respond effectively to dynamic energy demands and evolving market conditions. Unlike traditional networks and other sensing technologies, WSNs offer the ability to rapidly establish network connections [5]. Their flexible and dynamic topology enables the addition and removal of nodes with ease, while their robust design ensures high resilience against damage or attacks [6,7]. A typical WSN consists of three primary components: sensor nodes, the sensor network, and the terminals of users. Nodes are typically deployed across

a defined area to meet specific monitoring requirements. The sensor network collects data from these nodes via fixed communication channels, performs analysis and computation on the data, and aggregates the results at a base station before transmitting the information to designated users via satellite or other wireless communication methods. This enables wireless sensing capabilities to be realized.

In the context of smart grids, authentication protocols are essential for ensuring the secure interaction of various entities, including grid operators, distributed generation units [8], end consumers, and a range of sensors and smart devices. These participants form a complex and dynamic authentication environment. For instance, grid operators need to verify the authenticity and integrity of remote control commands, distributed generation units must confirm the legitimacy of the entities receiving their generation data, and end consumers require assurances that the electricity service information they receive originates from a trusted provider.

However, the open nature of smart grid systems poses significant security challenges, as adversaries can exploit public communication channels to intercept or impersonate legitimate participants. Given the critical nature of power control information transmitted within smart grids, any unauthorized access or data leakage can result in severe consequences, such as operational disruptions, financial losses, or even threats to national security [9]. A notable example occurred on 12 October 2020, when a cyberattack on Mumbai's power grid led to an unprecedented blackout and the near-paralysis of critical infrastructure [10]. Therefore, ensuring the secure identification of entities and protecting communication data are fundamental prerequisites for smart grid security. To address these concerns, this paper proposes a mutual authentication and session key establishment protocol between sensor nodes and gateway nodes to enhance the integrity and confidentiality of communications within smart grids.

Figure 1 shows the architecture of an SG system in terms of authentication schemes. The protocol authentication system consists of two parties: a sensor node ($Node_s$) and a gateway node ($Node_g$). The two nodes need to generate the device number through the registration and initialization phase, complete the registration and initialization of the device, and then complete the mutual authentication through the authentication and key agreement phase. Since data transfer between $Node_s$ and $Node_g$ takes place through open and vulnerable communication channels, this presents challenges in data protection, authentication, and mutual authentication. Therefore, in order to ensure the data security and the authenticity of the identity of the participants in the smart grid environment, a scheme is urgently needed to realize the mutual authentication between $Node_s$ and $Node_g$ and the establishment of session keys.

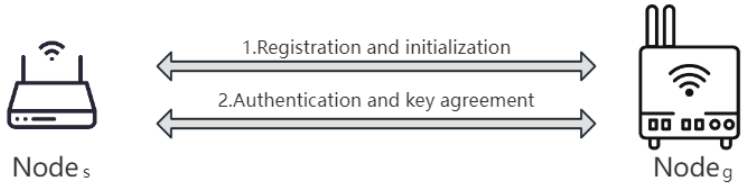

Figure 1. System architecture of anonymous authentication protocol.

In a smart grid environment, it is critical to ensure the security of data transmission and the legitimacy of the identities of participants. However, in an unattended IoT environment, all sensitive data collected by industrial sensor nodes is transmitted over open channels and is vulnerable to various types of attacks. On the other hand, because WSNs use wireless communication, their communication links are not as private and controllable as wired networks. It can be seen that ensuring the authenticity of the entity's identity and the integrity of the communication data is the basic requirement for maintaining security of the

smart grid. In order to enhance the security of the smart grid system, this paper designs a set of lightweight anonymous authentication protocols for smart grid based on physical unclonable functions.

1.1. Motivations and Contributions

In the context of the increasing popularity of smart grid systems, ensuring their safe operation has become even more critical.

Despite significant advancements in smart grid technology, existing authentication protocols still have several challenges to the security and efficiency of smart grids:

(1) Lack of Strong Mutual Authentication:

Many existing protocols focus on one-way authentication, where only the server or the client is authenticated. However, in a smart grid environment, mutual authentication between sensor nodes and gateway nodes is crucial to preventing unauthorized access [9].

(2) Insufficient Protection of Communication Data:

The open nature of smart grid systems makes them vulnerable to eavesdropping and man-in-the-middle (MITM) attacks [11]. Existing protocols often fail to provide robust encryption mechanisms to protect sensitive data transmitted over public channels.

(3) Limited Support for Anonymity and Privacy:

In many existing protocols, user identity information is not adequately protected, leaving users vulnerable to tracking and profiling [12]. This is particularly concerning in smart grids, where end consumers may be hesitant to use services if they feel their privacy is at risk. A lack of anonymity can also make it easier for attackers to target specific users or devices, increasing the risk of targeted attacks.

(4) High Computational and Communication Overheads:

Some existing protocols rely on computationally intensive cryptographic operations, such as asymmetric encryption, which can be resource-intensive for resource-constrained devices like sensor nodes in smart grids [13]. This leads to increased latency, higher energy consumption, and reduced scalability.

(5) Vulnerability to Various Attacks:

Existing protocols often fail to adequately address a wide range of potential attacks, including replay attacks, node capture attacks, node spoofing attacks, online key guessing attacks, desynchronization attacks, physical attacks, and modeling attacks [14]. These vulnerabilities can compromise the integrity and availability of the smart grid system.

These factors sparked the need to develop a more secure and efficient robust anonymous authentication scheme.

Our contributions are as follows:

(1) Using symmetric cryptography and physical unclonable functions (PUFs), we have designed an anonymity protocol that is both efficient and secure for authentication and session key generation between smart meters and service providers for future communication use. Compared to other schemes that require complex asymmetric cryptographic operations (such as [13,15,16]), the computational burden is significantly reduced while maintaining the same level of security.

(2) We have demonstrated through formal verification that the scheme can achieve mutual authentication and ensure the security of the session key. The security of the protocol was verified from BAN logic and non-formal analysis. This solution has good forward confidentiality and user anonymity and can effectively defend against replay

attacks, node capture attacks, node spoofing attacks, online key guessing attacks, desynchronization attacks, physical attacks, and modeling attacks.

(3) In comparison with the five advanced SG protocols, the SG lightweight anonymous authentication protocol proposed in this paper has the highest security on the basis of satisfying the eight security requirements. The results show that our scheme achieves all eight of the listed security requirements with the condition that the calculation overhead, communication overhead, and storage overhead are as small as possible.

1.2. The Structure of This Paper

The structure of this paper is as follows. Section 2 provides a comprehensive review of related works, highlighting their strengths and limitations. Section 3 establishes the necessary preliminaries, defining the adversary model, notational conventions, evaluation criteria used throughout the paper, the properties of hash functions, and the foundational PUF-based AKA protocol. Section 4 details our proposed scheme, outlining the steps involved in the registration and initialization phase, followed by the authentication and key agreement phase. Section 5 presents a rigorous security analysis, including formal verification using ProVerif-2.05, a heuristic analysis to address potential vulnerabilities, a BAN logic analysis for authentication properties, and a proof of security. Section 6 analyzes the performance of our proposed scheme, comparing it to existing approaches. Finally, Section 7 summarizes our contributions, discusses limitations, and suggests directions for future research.

2. Related Works

Smart grid (SG) environments are characterized by devices with limited computing power, storage space, and communication bandwidth. This resource scarcity dictates that security protocols designed for SGs must be highly efficient, minimizing overhead in these areas while maintaining robust security. In the early development of SG security, Wu and Zhou [11] proposed a key management scheme that employed elliptic curve cryptography (ECC). Their work aimed to establish mutual authentication and key agreement between smart sensors and data collectors, which are fundamental components of SG infrastructure. However, Xia and Wang [12] soon identified a critical vulnerability in this approach: it was susceptible to man-in-the-middle (MITM) attacks. To counter this, Xia and Wang [12] enhanced the scheme by introducing trusted privileges specifically between the smart meter and the service provider. This modification was intended to create a more secure channel for session key establishment. Despite this improvement, Park et al. [17] subsequently pointed out that even the revised scheme by Xia and Wang [12] had limitations. They demonstrated that it still did not offer adequate protection against impersonation attacks, where malicious entities could pretend to be legitimate users or devices, nor did it fully guarantee user anonymity, a growing concern in privacy-sensitive applications.

Addressing the specific need for anonymity, Tsai and Lo [18] developed an anonymous key distribution scheme. Their design ingeniously employed two distinct identity-based cryptosystems to simultaneously achieve mutual authentication and ensure the anonymity of smart meters. This was a significant step towards enhancing user privacy in SG communications. However, the evolution of security analysis led Odelu et al. [13] to scrutinize Tsai and Lo's [18] scheme. In their evaluation, Odelu et al. [13] highlighted vulnerabilities related to ephemeral secret leak (ESL) attacks. ESL attacks are a concern when temporary secrets, even if intended to be short-lived, are compromised. Odelu et al. [13] also raised concerns that Tsai and Lo's scheme did not sufficiently guarantee privacy in all operational contexts. As an alternative, they proposed a provably secure protocol grounded in bilinear pairing, a more complex cryptographic primitive. While offering enhanced security

guarantees, Abbasinezhad-Mood et al. [19] critiqued Odelu et al.'s [13] protocol as being too computationally intensive. They argued that the computational demands of bilinear pairing made it potentially unsuitable for the resource-constrained nature of many SG environments, particularly the sensors and meters at the network edge.

In a different approach, Srinivas et al. [15] designed an anonymous authentication and key exchange protocol based on Schnorr signatures [20]. Schnorr signatures are known for their efficiency and security properties. This scheme aimed to strike a balance between security and computational feasibility. Following the publication of Srinivas et al.'s work, Baruah et al. [21] conducted a thorough analysis. Their investigation revealed that despite the use of Schnorr signatures, Srinivas et al.'s protocol [15] still exhibited vulnerabilities to both MITM attacks and impersonation attacks. This finding underscored the persistent challenges in designing truly robust authentication mechanisms for smart grids. Khan et al. [22] explored another avenue for lightweight authentication in SGs, incorporating both hash functions and ECC into a single scheme. Their intention was to create a protocol that was both secure and computationally efficient. However, a detailed examination by Chaudhry [23] uncovered a critical flaw: the ECC operations within Khan et al.'s [22] scheme were not implemented correctly. This misuse of ECC had the serious consequence of compromising the security of the session key, effectively undermining the intended security of the protocol.

To address the emerging threat of physical attacks, research began to shift focus towards technologies like physical unclonable functions (PUFs). PUFs are hardware-based security primitives that leverage inherent variations in physical manufacturing processes to generate unique and unclonable outputs. Gope and Sikdar [24] recognized the potential of PUFs and utilized them in conjunction with symmetric encryption algorithms. Their work aimed to design privacy-preserving schemes specifically for SG environments, offering a way to enhance security at the hardware level. Kaveh et al. [25] also adopted PUFs, developing a lightweight protocol intended for secure communications specifically between smart meters and their neighboring gateways. This focus on meter–gateway communication is critical for data aggregation and control in SGs. However, Liu et al. [26] raised important concerns about Kaveh et al.'s [25] protocol. They pointed out a potential design flaw where gateways could gain direct access to secret challenge-response pairs associated with the PUFs. This direct access, Liu et al. [26] argued, not only compromised anonymity but also opened up the system to potential denial-of-service (DoS) attacks, where malicious actors could exploit this access to disrupt network operations.

In more recent developments, Bagheri Nasour et al. [27], in 2024, presented PPSG, a protocol that represents an evolution in SG security. PPSG strategically combines the strengths of PUFs with ECC modules. This hybrid approach was explicitly designed to address vulnerabilities that had been identified in previous SG authentication methods, aiming for a more robust and secure solution. Concurrently, recognizing the broader needs of SG resilience, Alidaee Bahram et al. [28] made a significant contribution to disaster recovery planning. They proposed a comprehensive framework for optimizing redundant allocation problems within critical SG infrastructure. Their work focused on ensuring that smart grids can maintain essential functions even in the face of disruptions, by strategically placing redundant resources.

In summary, a review of the existing body of work on SG security protocols reveals persistent shortcomings. Many of these protocols, despite their individual contributions, still fall short in providing essential security properties such as perfect forward secrecy and comprehensive user anonymity. Furthermore, a significant number of them do not offer adequate defense against a range of known attack vectors, including MITM attacks, impersonation attacks, ESL attacks, replay attacks, and even privileged internal attacks.

Critically, when considering the practical constraints of smart grid deployments, a major limitation is performance. Most of the currently available schemes are not ideally suited for resource-constrained SG settings. They often impose high computational demands and create significant communication overhead, making them less practical for widespread implementation. Therefore, to address these multifaceted challenges, this paper proposes a novel authentication scheme specifically tailored for the unique environment of smart grids. Inspired by the work of Gope et al. [29], who effectively used fuzzy extractors to mitigate the impact of noise on PUF responses in RFID systems, our proposed protocol integrates both PUFs and fuzzy extractors. This integration is intended to leverage the physical security of PUFs to resist physical attacks while simultaneously using fuzzy extractors to reliably handle the inherent variability and noise associated with PUF responses, leading to a more practical and robust authentication mechanism for smart grids.

3. Preliminaries

3.1. Adversary Model

The Dolev–Yao model is a classic model in the field of security protocol analysis, widely used for analyzing the security of various protocols and recognized by both academia and industry. Smart grid networks demand extremely high levels of security and reliability, as any leakage or tampering of communication data can lead to severe consequences. The communication links in smart grids are often exposed in public networks, allowing attackers to control public channel to launch man-in-the-middle attacks or replay attacks, etc., which aligns closely with the Dolev–Yao model's assumptions that an attacker has complete control over the communication channel, enabling them to intercept, modify, replay, and forge all messages. Additionally, In the practical deployment of smart grids, many security protocols such as IEEE 802.1X [30] and IEC 62351 [31] are designed based on the assumptions of the Dolev–Yao model and have proven effective in defending against network attacks. Therefore, this study designs the protocol based on the Dolev–Yao adversary model. By analyzing under the Dolev–Yao model, we can uncover various potential weaknesses in the protocol, thereby enhancing its robustness and security.

It is assumed that Adversary A in the industrial Internet of Things environment has the following capabilities:

(a) A knows the AKA protocol used;
(b) A can completely control the open communication channel. That is, A can eavesdrop, modify, insert and delete messages transmitted through the open channel;
(c) When the user's mobile device is acquired by A, A can acquire the stored secret parameters through a side channel attack method;
(d) A can physically capture industrial sensor nodes and extract all stored parameters from their memory by power analysis method;
(e) When evaluating forward security, A can obtain the long-term key of the participating entity;
(f) When evaluating the replay attack, A can insert the previously intercepted message in the communication channel;
(g) A may be a legitimate but malicious industrial sensor node.

3.2. Notation

To facilitate understanding of the proposed scheme, Table 1 lists necessary terms and notation.

Table 1. Notation of symbols in our scheme.

Parameters	Notions	Length
Pos	Random position information	64 bits
$MPos$	Location mask	128 bits
$PUF_{s(seed)}$, $PUF_{g(seed)}$, $PUF_{a(seed)}$	PUF fingerprint seed value of sensor/gateway/access node	128 bits
PUF_s, PUF_g, PUF_a	PUF fingerprint of sensor/gateway/access node	256 bits
SID_s, SID_g, SID_a	Sensor/gateway/access node hidden identification	64 bits
ID_s, ID_g, ID_a	Identification of sensor/gateway/access node	48 bits
$ADDR_s$, $ADDR_g$, $ADDR_a$	Address of PUF fingerprint of sensor/gateway/access node	16 bits
$N_{s,g}$, $N_{g,a}$	Synchronous serial numbers of sensor/gateway node and gateway/access node	32 bits
BCH_s, BCH_g, BCH_a	BCH code word of sensor/gateway/access node	256 bits
HD_s, HD_g, HD_a	Sensor/gateway/access node help data	256 bits
$h_1(\cdot)$	One-way hash function	64 bits
$h_2(\cdot)$	One-way hash function	128 bits
LEA/LDA	Lightweight encryption/decryption algorithm	64 bits
M	Key information	128 bits
APS	Encrypted information	64 bits
R	random digit	32 bits
IDF	Identification field	128 bits
K	Encryption/decryption key	64 bits
SK	Session key	64 bits
ACK_1, ACK_2	Response information	8 bits

3.3. Evaluation Criteria

In an unattended industrial IoT environment, all sensitive data collected by industrial sensor nodes are transmitted through open channels and is vulnerable to various types of attacks [6]. At the same time, this data is crucial for safe production, and industrial control is highly dependent on this sensitive data [3,4]. Therefore, there is an urgent need for a robust and efficient authentication scheme to protect the secure transmission of industrial data against unauthorized access [8]. Based on previous research results [15,16,32] and actual environmental requirements, Table 2 shows eight security evaluation criteria that the proposed AKA protocol needs to meet.

The smart grid, as a critical national infrastructure, has unique security requirements. First and foremost, it demands extremely high reliability, as any security breach could lead to widespread power outages, thereby impacting social stability and economic development. Second, the long deployment lifecycle of smart grid devices necessitates their ability to withstand a variety of evolving security threats over extended periods. Furthermore, the geographically dispersed nature of smart grid nodes presents significant

challenges for security protection. Finally, the data generated during smart grid operation includes sensitive information related to user privacy and grid operational status, requiring stringent protection.

Table 2. Frequently used security evaluation criteria (SEC) in SG environments.

Short Term		Definition
SEC1	Forward secrecy	Even if a system's private keys are compromised, past session keys remain secure and cannot be decrypted.
SEC2	Anti-replay attack	A technical means by which data packets in network transmission are not maliciously reused by attackers to bypass the security mechanism.
SEC3	Anti-node capture attack	It protects a network or system from the consequences of a compromised node by ensuring secure protocols and encryption.
SEC4	Anti-impersonation attack	It involves robust authentication mechanisms to prevent an adversary from masquerading as a legitimate node within a network.
SEC5	Anti-online key guessing attack	It aims to thwart attacks where an attacker guesses cryptographic keys by implementing rate limiting and strong password/key management practices.
SEC6	Anti-desynchronization attack	It ensures the robustness of cryptographic protocols against attacks that disrupt synchronization between parties, maintaining consistency and integrity.
SEC7	Anti-physical attack	It safeguards a system or network from attacks that involve physical access or manipulation, such as tampering with hardware components, stealing devices, or conducting side-channel attacks to extract sensitive information.
SEC8	Anti-modeling attack	It defends against attacks where an adversary attempts to create a model or simulation of the system or cryptographic algorithms to gain insights and potentially break the security, often by using techniques like machine learning or statistical analysis to predict or reverse-engineer the system's behavior.

The following sections explain why each of the proposed security criteria is more critical in the smart grid domain than in other areas:

SEC1: Forward Secrecy

Forward secrecy is crucial in the smart grid due to the long deployment lifecycle, fixed locations, and difficult replacement of smart grid devices. If a private key is compromised (e.g., through physical capture or long-standing vulnerabilities), attackers can decrypt all past communications, leading to large-scale data breaches and even the possibility of retroactively forging control commands, severely threatening grid stability. In contrast, general IoT devices have shorter lifecycles and lower data sensitivity, making the impact of key compromise relatively limited.

SEC2: Anti-replay Attack

Control commands in the smart grid are highly time-sensitive and sequential, making anti-replay attacks critical for ensuring its secure operation. Replay attacks can cause incorrect control commands to be executed repeatedly; for example, repeated circuit breaker tripping commands could trigger widespread blackouts. General IoT devices, however, have less stringent requirements for command timing, and the consequences of replay attacks are typically less severe.

SEC3: Anti-node Capture Attack

The smart grid comprises a vast number of geographically distributed nodes, often deployed in unattended environments, making them vulnerable to physical attacks. Anti-node capture attacks are effective in mitigating these risks. Node capture can lead to attackers gaining access to critical keys, control privileges, and even the ability to manipulate the entire grid. In contrast, general IoT devices are typically deployed in relatively secure environments, where the risk of physical capture is lower.

SEC4: Anti-impersonation Attack

In the smart grid, if an attacker can successfully impersonate a legitimate control center or device, they can issue false control commands, causing grid malfunctions or data theft. Therefore, anti-impersonation attacks are of paramount importance. In general IoT devices, even if impersonation occurs, it usually only affects a single device or user experience.

SEC5: Anti-online Key Guessing Attack

Key management is crucial in the smart grid, as a compromised key grants attackers control over the grid. Anti-online key guessing attacks effectively strengthen the security perimeter. Online key guessing attacks can lead to key leakage, resulting in severe consequences. In contrast, general IoT devices often use shorter keys or passwords, and even if they are cracked, the impact is limited.

SEC6: Anti-desynchronization Attack

Many protocols in the smart grid rely on time synchronization, making anti-desynchronization attacks vital for ensuring its reliable operation. Desynchronization attacks can lead to communication interruptions, control failures, and even cascading grid failures. For example, an attack on a clock synchronization protocol could cause protective relays to malfunction. General IoT devices have lower requirements for time synchronization, and the consequences of desynchronization attacks are typically less severe.

SEC7: Anti-physical Attack

Critical infrastructure components of the smart grid (e.g., substations, smart meters) are often deployed outdoors, making them susceptible to physical attacks. Anti-physical attacks are essential for ensuring system security and data integrity. Physical attacks can lead to device damage, data breaches, and even direct control of the grid. In contrast, general IoT devices are typically deployed indoors, where the risk of physical attack is lower.

SEC8: Anti-modeling Attack

If an attacker successfully models the security mechanisms of the smart grid, they can predict system behavior, identify vulnerabilities, and even bypass security controls. Anti-modeling attacks are therefore crucial for maintaining system security. For instance, an attacker might analyze grid operational data to reconstruct the grid topology, thereby launching targeted attacks. General IoT devices, due to their relatively simple functionality, pose a lower risk and consequence of model-based attacks.

The eight security criteria outlined above comprehensively cover the two core objectives of authentication—identification/verification and authorization—and enhance the robustness of the smart grid security model from multiple dimensions, enabling it to effectively resist various potential attacks.

3.4. Hash Function

Hash functions are divided into key-based hash functions and keyless hash functions. Among them, the hash function based on a key is usually called message authentication code (MAC), while the hash function without a key is usually called manipulation detection

code (MDC). MDC can be divided into one-way hash functions (OWHF) and collision-resistant hash functions (CRHF).

3.4.1. One-Way Hash Function (OWHF)

The first formal definition of OWHF was put forward by R.Merkl and M.Rabin, and its description is as follows. The one-way hash function is a function h that satisfies the following conditions:

(a) The description of H must be public and without the aid of any key;
(b) The parameter x can be of any length. $h(x)$ has a fixed length of n bit, where $n \geq 64$;
(c) Given the function h and the parameter x, it is very simple to calculate $h(x)$;
(d) Hash function must be unidirectional—that is, given a parameter y, it is difficult to find the parameter x so that the condition $h(x) = y$ holds, and given the parameter x, it is difficult to find $x' \neq x$, which makes the condition $h(x') \neq h(x)$ hold.

Among them, (a), (b), and (c) describe the generality of the keyless hash function, while (d) gives the characteristic constraints of OWHF, in which the first half gives the intuitive one-way concept. That is, it is difficult to find the original image of a given value within a certain range, also known as the anti-first original image property, which is related to the arrangement or mapping property of OWHF; in the latter part, it provides a stronger anti-second original image property. That is, it is difficult to find a substitute image of the original image for hash operation.

3.4.2. Collision-Resistant Hash Function (CRHF)

The first formal definition of CRHF was put forward by I. Damgdard, and its description is as follows. The anti-collision hash function is a function h that satisfies the following conditions:

(a) The description of H must be public and without the aid of any key;
(b) The parameter x can be of any length. $h(x)$ has a fixed length of n bit, where $n \geq 128$;
(c) Given the function h and the parameter x, it is very simple to calculate $h(x)$;
(d) The hash function must be unidirectional. That is, given a parameter y, it is difficult to find the parameter x so that the condition $h(x) = y$ holds; given the parameter x, it is difficult to find $x' \neq x$, which makes the condition $h(x') \neq h(x)$ hold.
(e) The h function must be conflict-resistant: this means that it is difficult to find two different parameters so that they have the same hash value.

In Sections 3.4.1 and 3.4.2, we detailed the definitions and characteristics of one-way hash functions (OWHF) and collision-resistant hash functions (CRHF). The pre-image resistance property of OWHF means that it is difficult for attackers to derive the original input from a hash value, which increases the complexity of attacks. CRHF, building upon OWHF, further emphasizes collision resistance, making it difficult to find two different inputs that produce the same hash value. Under ideal security conditions, the operational complexity required to generate a second pre-image is $O(2^n)$, while the complexity for finding collisions is $O\left(2^{n/2}\right)$. These characteristics significantly increase the difficulty for attackers to break the hash function or forge data.

In selecting hash functions, we focused on their modeling resistance properties, particularly collision resistance and one-way characteristics. These properties are crucial for preventing data tampering and forgery. Specifically, we chose collision-resistant hash functions (CRHF) because they not only possess the pre-image resistance property of one-way hash functions (OWHF) but also provide stronger collision resistance. The collision complexity of CRHF is $O\left(2^{n/2}\right)$, which provides enhanced security in practical applications, especially in scenarios requiring protection against collision attacks. Moreover, these char-

acteristics of CRHF also help to resist various modeling attacks, such as birthday attacks, thereby ensuring the overall security of the system.

3.4.3. Message Authentication Code Based on Hash Function (HMAC)

Message authentication code has been adopted by banks for a long time and is older than cryptography, which was only opened for research in the mid-1970s. However, a MAC with good cryptographic characteristics based on a hash function which we called HMAC was put forward after the study of open cryptography, and its definition is as follows. HMAC is a function h that satisfies the following conditions:

(a) The description of H must be public, and the key K used therein is secret information;
(b) The parameter x can be of any length, and $h(K, x)$ has a fixed length of n bit, where $32 \leq n \leq 64$;
(c) Given the function h, the parameter x and the key k to calculate $h(K, x)$ simply;
(d) Given the function h and the parameter x, $h(K, x)$ cannot be calculated with a probability higher than $1/2n$. Even if there is a large set of $\{x_i, h(K, x_i)\}$ data pairs, where x_i is the known data selected by the adversary, it is very difficult to guess the value of the key k or $h(K, x')$ for any $x' \neq x_i$. This shows that HMAC should be one-way and anti-collision for adversaries who do not know the key K.

3.5. PUF-Based AKA Protocol

PUF is an effective key generation method that can effectively resist physical attacks. Unlike traditional cryptography, PUFs use the method of stimulus and response pairs to generate keys, and the mapping between stimuli and responses is determined only by certain physical differences between individual devices. These physical differences are created by subtle differences in the IC manufacturing process that cannot be predicted, replicated, and controlled. Due to the highly physically correlated and uncontrollable nature of these physical differences, each IC chip has and only has its own unique physical "fingerprint", which is born with the chip and does not change. Just like human fingerprint recognition, only one instance of fingerprint collection and archiving is required, and each time fingerprint identification is needed in the future, only on-site collection is required. Individuals do not need to save their own fingerprint samples, nor do they have to worry about their fingerprint samples being stolen by others. Therefore, PUF keys are highly physically deterministic, are ready to use, do not need to be kept, are difficult to predict in advance, and are easy to measure, providing a secure, stable, and low-cost security key.

At present, there are few studies on physical unclonable functions (PUFs) in China, and their applications are limited, mainly focusing on the RFID authentication protocol of micro-delay circuits and the design of a small number of arbitrator PUFs, and there are few studies and applications on the implementation methods of other types of PUFs.

In contrast, PUF has a longer history of research abroad. As early as 2001, the Massachusetts Institute of Technology's Pappu S.R. proposed the first optical PUF model, which used a laser beam and a transparent wafer to produce a fixed-length bit value in response. Subsequently, non-electronic PUFs such as CD PUFs and paper PUFs were developed, which can extract physical "fingerprints" from the medium for authentication and unique identification, but their application range is limited.

After non-electronic PUFs, analog circuit PUFs such as coated PUFs appeared, which responded by spraying a special coating on the IC and measuring the random capacitance using a capacitive sensor. In addition, there are two main implementation methods for digital circuit PUF, which is the most studied abroad: one is to use the change of digital signal propagation delay, such as arbiter PUF, ring oscillator PUF and glitch PUF; the

second is to use the steady-state manufacturing changes of the memory cell structure, such as SRAM PUF, flip-flop PUF, latch PUF, and butterfly memory PUF.

Among them, the memory-based PUFs have been widely studied and applied, especially the SRAM-based PUFs. A typical SRAM (static random access memory) logical storage unit can be regarded as consisting of two cross-coupled inverters. Due to some uncontrollable factors in the manufacturing process, such as the uniformity of doping concentration and the slight difference of transistor channel length-width ratio, every memory cell is inherently unique, forming a unique "fingerprint" of SRAM memory cells. Because of the influence of environmental factors such as external temperature and noise in the process of power-on, the SRAM memory matrix will obtain different responses at different times, thus forming a series of excitation-response pairs of power-on-SRAM values. In the existing literature, SRAM features have been used to design FPGA IP protection protocols, chip fingerprint authentication and true random number generation concepts, and chip unique identification code generation on microprocessors. These studies not only promote the development of PUF technology but also provide new solutions for the security of electronic systems and the protection of intellectual property.

4. Proposed Scheme

This section presents our scheme. To facilitate the reader's understanding of the proposed scheme, Figures 2 and 3 illustrate the registration and initialization phase and the authentication and key agreement phase of the protocol, respectively.

	Sensor node ($Node_s$) Store < SID_g, ID_g, $N_{s,g}$, $ADDR_s$, HD_s>	**Gateway node ($Node_g$)** Store < SID_s, ID_s, $N'_{s,g}$, $ADDR_g$, HD_g>
Registration and Initialization phase	**Step1:** Generate $\boldsymbol{SRAM_{s1}}$, $\boldsymbol{SRAM_{s2}}$, $\boldsymbol{ID_s}$ Extract $\boldsymbol{PUF_{s(seed)}}$ from $SRAM_{s1}$ Generate $\boldsymbol{BCH_s}$, $\boldsymbol{HD_s}$ according to $PUF_{s(seed)}$ Extract $\boldsymbol{PUF_s}$ from $SRAM_{s2}$ Extract address information $\boldsymbol{ADDR_s}$ of PUF_s $\boldsymbol{SID_s} = h_1(PUF_s)$ Generate $\boldsymbol{N_{s,g}}$=0 $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ID_s}$, $\boldsymbol{SID_s}$}	**Step2:** Generate $\boldsymbol{SRAM_{g1}}$, $\boldsymbol{SRAM_{g2}}$, $\boldsymbol{ID_g}$ Extract $\boldsymbol{PUF_{g(seed)}}$ from $SRAM_{g1}$ Generate $\boldsymbol{BCH_g}$, $\boldsymbol{HD_g}$ according to $PUF_{g(seed)}$ Extract $\boldsymbol{PUF_g}$ from $SRAM_{g2}$ Extract address information $\boldsymbol{ADDR_g}$ of PUF_g $\boldsymbol{SID_g} = h_1(PUF_g)$ Generate $\boldsymbol{N'_{s,g}}$=0 $\boldsymbol{Node_s \leftarrow Node_g}$:{$\boldsymbol{ID_g}$, $\boldsymbol{SID_g}$}

Figure 2. Registration and initialization phase.

	Sensor node ($Node_s$) Store < SID_g, ID_g, $N_{s,g}$, $ADDR_s$, HD_s>	**Gateway node ($Node_g$)** Store < SID_s, ID_s, $N'_{s,g}$, $ADDR_g$, HD_g>
Authentication and key agreement phase	**Step1:** Generate a random number $\boldsymbol{Pos}$ $\boldsymbol{MPos} = Pos \oplus h_2(SID_s \| \| SID_g \| \| N_{s,g})$ $\boldsymbol{V} = h_2(SID_s \| \| Pos \| \| N_{s,g})$ $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ID_g}$, **MPos, V**} **Step3:** if $V=V'$ $\boldsymbol{R'} = M \oplus h_2(SID_s \| \| SID_g \| \| Pos)$ $\boldsymbol{IDF'} = h_2(SID_s \| \| SID_g \| \| R')$ $\boldsymbol{K} = \text{LEA}(IDF', Pos)$ $\boldsymbol{Pos''} = \text{LDA}(APS, K)$ if $Pos''=Pos$ $SK = K$ $N_{s,g} = N_{s,g} + 1$ $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ACK_2}$, $\boldsymbol{ID_s}$} if $Pos'' \neq Pos$ $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ACK_1}$, $\boldsymbol{ID_s}$} if $V \neq V'$ if $ID'_s = ID_s$ $\boldsymbol{N'_{c,h}} = N''_{s,g}$ $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ACK_1}$, $\boldsymbol{ID_s}$} if $ID'_s \neq ID_s$ $\boldsymbol{Node_s \rightarrow Node_g}$:{$\boldsymbol{ACK_1}$, $\boldsymbol{ID_s}$}	**Step2:** $\boldsymbol{Pos'} = MPos \oplus h_2(SID_s \| \| SID_g \| \| N'_{s,g})$ $\boldsymbol{V'} = h_2(SID_s \| \| Pos' \| \| N'_{s,g})$ if $V=V'$ Generate a random number $\boldsymbol{R}$ $\boldsymbol{IDF} = h_2(SID_s \| \| SID_g \| \| R)$ $\boldsymbol{K'} = \text{LEA}(IDF, Pos')$ $\boldsymbol{M} = R \oplus h_2(SID_s \| \| SID_g \| \| Pos')$ $\boldsymbol{APS} = \text{LEA}(Pos', K')$ $\boldsymbol{Node_s \leftarrow Node_g}$:{$\boldsymbol{ID_g}$, $\boldsymbol{M}$, $\boldsymbol{APS}$} if $V \neq V'$ $\boldsymbol{APS} = \text{LEA}(ID_s \| \| N'_{s,g}, SID_g)$ $\boldsymbol{Node_s \leftarrow Node_g}$:{$\boldsymbol{ID_g}$, $\boldsymbol{APS}$} **Step4:** if received message is ACK_2 $\boldsymbol{N'_{s,g}} = N'_{s,g} + 1$ If not pass

Figure 3. Authentication and key agreement phase.

4.1. Registration and Initialization Phase

In this phase, we need to generate a unique device identifier and complete device registration and initialization. The details are presented in Figure 2.

R1. Generation of equipment number

Firstly, the sensor node and the gateway node respectively extract an initial value of SRAM as seed values $PUF_{s(seed)}$ and $PUF_{g(seed)}$. Secondly, according to the seed value, the sensor node and the gateway node respectively generate respective *BCH* codewords BCH_s and BCH_g and help data HD_s and HD_g according to the seed values. Thirdly, the sensor and the gateway node respectively extract the initial value of another SRAM. Then, the sensor and the gateway node respectively extract the hidden identifiers $SID_s = h_1(PUF_s)$ and $SID_g = h_1(PUF_g)$. Lastly, each node generates the sequence number $N_{s,g}$, corresponding to the initial value of zero.

R2. Registration and initialization of equipment

The sensor node sends ID_s and SID_s to the gateway node, and the gateway node sends ID_g and SID_g to the sensor node. Then, because data such as *SID* and *ID* contain the identity characteristics, each node needs not to be modified frequently and stores {*SID*, *ID*, $N_{s,g}$, *ADDR*, *HD*} in their respective FLASH in a read-only manner. In order to enhance the security of stored data, the data is encrypted for the secret information, users needs to run the PUF to recover the secret. It is especially important to note that this stage needs to be completed under the condition of secure channel.

4.2. Authentication and Key Agreement Phase

As shown in Figure 3. After the registration and initialization phase is completed, the sensor and the gateway node respectively hold parameters {ID_s, ID_g, SID_s, SID_g, $N_{s,g}$, BCH_s, HD_s} and parameters {ID_g, ID_s, SID_g, SID_s, $N_{s,g}$, BCH_g, HD_g}. Based on each parameter, the authentication and key agreement phase is carried out as below.

V1. $Node_s \Rightarrow Node_g$: {$ID_g$, *MPos*, *V*}.

The sensor node $Node_s$ generates the random position information *Pos* and reads the parameter SID_s from $FLASH_c$. Based on received parameter SID_g and $N_{s,g}$, it can calculate $MPos = Pos \oplus h_2\ (SID_s \mid\mid SID_g \mid\mid N_{s,g})$ and $V = h_2\ (SID_s \mid\mid Pos \mid\mid N_{s,g})$. After that, the sensor node sends parameters {ID_g, *MPos*, *V*} to the gateway node. It is worth noting that we use HMAC method to protect the data containing identity information and data to be transmitted to enhance the credibility of identity authentication. After receiving the data, the receiver can calculate the HMAC value through the same HMAC algorithm and compare it to the HMAC value provided by the sender to verify the integrity of the data. The following practice is the same as here.

V2. $Node_g \Rightarrow Node_s$: {$ID_g$, *APS*} or $Node_g \Rightarrow Node_s$: {$ID_g$, *M*, *APS*}.

After the gateway node $Node_g$ receiving the parameters, it verifies whether the ID_g is its own ID. If not, the authentication is terminated; otherwise, it reads the parameter SID_g from FLASH. Based on received parameter SID_s and $N'_{s,g}$, it can calculate $Pos' = MPos \oplus h_2\ (SID_s \mid\mid SID_g \mid\mid N'_{s,g})$ and $V' = h_2(SID_s \mid\mid Pos' \mid\mid N'_{s,g})$. Then, the gateway node verifies whether $V' = V$. If $V' \neq V$, it is required to initiate authentication again and calculate $APS = LEA\ (ID_s \mid\mid N'_{s,g}, SID_g)$. Then, the gateway node sends ID_g and *APS* to the sensor node; if $V' = V$, the gateway node generates a random number *R*, calculates $IDF = h_2\ (SID_s \mid\mid SID_g \mid\mid R)$, $K' = LEA(IDF, Pos')$, $M = R \oplus h_2\ (SID_s \mid\mid SID_g \mid\mid Pos')$ and $APS = LEA(Pos', K')$, and sets the session key $SK = K'$. Lastly, the gateway node sends ID_g, *M* and *APS* to the sensor node.

V3. $Node_s \Rightarrow Node_g$: {$ACK_1$, ID_s} or $Node_s \Rightarrow Node_g$: {$ACK_2$, ID_s}.

After receiving the message, if $V' \neq V$, the sensor node calculates $(ID'_s \| N''s_{,g}) = LDA$ (APS, SID_g). If $ID'_s = ID_s$, the sensor node makes $N'_{s,g} = N''_{s,g}$ and feedbacks ACK_1 and ID_s and then reissues the certification application; if not, the sensor node feedbacks ACK_1 and ID_s directly and then reissues the certification application. If $V' = V$ from the beginning, the sensor node calculates $R' = M \oplus h_2$ $(SID_s \| SID_g \| Pos)$, $IDF' = h_2$ $(SID_s \| SID_g \| R')$, $K = LEA(IDF', Pos)$ and $Pos'' = LDA$ (APS, K). If $Pos'' \neq Pos$, the authentication fails and the sensor node reissues the certification application after sending ACK_1; if $Pos'' = Pos$, the sensor node sets the session key $SK = K$, makes $N_{s,g} = N_{s,g} + 1$, and finally sends an acknowledgement message ACK_2 to the gateway node.

As mentioned in 5.2.1, in order to ensure that the key will not be corrupted by attackers because it lives for too long, we add random location information *Pos* into the key generation algorithm to ensure that the session key is dynamic and unique.

V4. When the gateway node receives ACK_2, let $N_{s,g} = N_{s,g} + 1$.

In conclusion, Node_s and Node_g authenticate each other's identities and make a key agreement to form a session key.

5. Security Analysis

In this section, three typical protocol security analyses are conducted to prove the security of the lightweight anonymous authentication protocol for smart grid. First, we formally analyze the protocol. Then, the behavior detection of the possible attacks of the protocol is analyzed through nine aspects, such as forward security and non-formal analysis. Finally, BAN logic analysis again proves that the protocol can realize bidirectional authentication and session key agreement between the gateway node and sensor node.

5.1. Formal Verification by the ProVerif

ProVerif [33] is a formal automatic verification tool for security protocols. Based on the process algebra and Prolog rules of Pi calculus, it can infer the possibility of an attack event. When using the tool ProVerif to verify the authentication protocol, if the protocol exposes some security problems, ProVerif will generate a corresponding attack sequence. ProVerif can prove the following properties of the authentication protocol:

- Confidentiality (the adversary cannot obtain the secret);
- Equivalence between processes with different terms only, which tests the property of identity verification among entities.

To make the paper more concise, we have omitted the specific code here. Instead, we have uploaded the source code with detailed description to the GitHub platform, and the code details can be found at: https://github.com/2502094307/Security-test-code (accessed on 20 January 2025).

As follows, Figure 4 shows the ProVerif execution process with test results and verification summary. The verification summary indicates that for the first two queries, before the two events occur to each other, both entities complete the identity verification of each other, so the proposed protocol achieves the mutual authentication. For the last two queries, it denotes that the adversary cannot break the secname encrypted by the session key, which means that the secrecy of session key can be preserved in the proposed protocol.

```
Start process
Process
    new SRAM_GW: fingerprint;(*--SRAM of gateway node--*)
    new ADDR_GW: bitstring;(*--Address of PUF fingerprint of gateway node--*)
    new SRAM_SN: fingerprint;(*--SRAM of sensor node--*)
    new ADDR_SN: bitstring;(*--Address of PUF fingerprint of sensor node--*)
    new GID: bitstring;(*--identifier of gateway node--*)
    new SID: bitstring;(*--identifier of sensor node--*)
    new Ser_Num: integer;(*--Synchronous serial numbers of sensor node and gateway node--*)
(
    (*-- Launch an unbounded number of sessions of the GW--*)
    (!processGW(SRAM_GW, ADDR_GW, SID, Ser_Num))
    (*-- Launch an unbounded number of sessions of the SN--*)
    (!processSN(SRAM_SN, ADDR_SN, GID, Ser_Num))
)
```

```
Test results
Completing equations...
Query inj-event(GSend(x)) ==> inj-event(GSbegin(x))
Translating the process into Horn clauses...
Completing...
Starting query inj-event(GSend(x)) ==> inj-event(GSbegin(x))
RESULT inj-event(GSend(x)) ==> inj-event(GSbegin(x)) is true.
-- Query inj-event(SGend(x)) ==> inj-event(SGbegin(x))
Translating the process into Horn clauses...
Completing...
Starting query inj-event(SGend(x)) ==> inj-event(SGbegin(x))
RESULT inj-event(SGend(x)) ==> inj-event(SGbegin(x)) is true.
-- Query not attacker(secnameA[]); not attacker(secnameB[])
Translating the process into Horn clauses...
Completing...
Starting query not attacker(secnameA[])
RESULT not attacker(secnameA[]) is true.
Starting query not attacker(secnameB[])
RESULT not attacker(secnameB[]) is true.
```

```
Verification summary
Query inj-event(GSend(x)) ==> inj-event(GSbegin(x)) is true.
Query inj-event(SGend(x)) ==> inj-event(SGbegin(x)) is true.
Query not attacker(secnameA[]) is true.
Query not attacker(secnameB[]) is true.
```

Figure 4. Code of process, test results, and verification summary in ProVerif.

5.2. Heuristic Analysis

Heuristic analysis is a kind of informal analysis. Informal security analysis can also be called an attack detection test, which is based on tests and experiments. Therefore, informal security analysis methods not only require analysts to have high professional knowledge but also have certain requirements for analysis conditions. Generally speaking, informal security analysis is a tentative verification analysis for one or several known attacks. Through simulation tests, the known attack steps and methods are applied to the checked security protocol. According to the resistance of the security protocol to the known attacks, the analyst finally judges whether the protocol has one or several security performances. The protocol proposed in this paper is analyzed informally from the following perspectives.

5.2.1. Forward Secrecy

In this protocol, there is no shared key stored between the participating entities. In each authentication process, the session key is generated by using random location data *Pos*, so the session key is dynamic and randomly changed in the authentication process. Even if the adversary obtains the session key of the current communication, it is impossible to obtain the previous session key, which ensures that the confidentiality of the previous communication will not be affected. Therefore, the protocol can achieve forward security.

5.2.2. Dynamic Session Key Agreement

In this protocol, the gateway node and the sensor node negotiate a session key SK to protect the secure communication between them. The session key is obtained by the random location data *Pos* acting on the identification field *R* through the extraction algorithm, and no third party can obtain the session key in advance. Therefore, this protocol can realize session key agreement.

5.2.3. Anti-Replay Attacks

In this protocol, serial numbers and random numbers are used to resist replay attacks. In each session, the gateway node and the sensor node generate random numbers *Pos* and *R* respectively, which are used to ensure the freshness and independence of the exchanged messages. In addition, the gateway node and the sensor node have a serial number synchronous detection mechanism, so even if the adversary replays the previous message, it will not pass the authentication. Therefore, the protocol can resist replay attacks.

5.2.4. Anti-Node Capture Attacks

In this protocol, the sensor node only stores the device number, serial number, BCH code word and help data, but not the shared key. Therefore, the adversary cannot obtain these data with read protection, so they cannot forge the message and pass the authentication of the gateway node. Therefore, the protocol can resist node capture attacks.

5.2.5. Anti-Impersonation Attacks

In this protocol, the sensor node only stores the device number, serial number, BCH code word and help data, and the key information in the public channel is encrypted and transmitted. Even if the adversary obtains the message, it is impossible to forge the information and pass the authentication of the gateway node. Therefore, the protocol can resist node impersonation attacks.

5.2.6. Anti-Online Key Guessing Attacks

In this protocol, the temporary session key is generated by a special extraction algorithm based on the random location data *Pos* and the identification field *R*, so the adversary cannot obtain the *Pos*, let alone calculate the session key of the current communication from *R* in a linear time. Therefore, the protocol can resist online key guessing attacks.

Additionally, in our protocol, the generation of session key *SK* needs to obtain both *IDF* and *Pos*. It is worth noting that the length of parameter *Pos* is 64 bits; the length of *IDF* is 128 bits. From a mathematical point of view, because the session key *SK* is determined by *IDF* and *Pos* together, and they are independent of each other in the process of key generation (independence here means that their combination of values is completely random). For an attacker, if he tries to guess the session key *SK*, he needs to traverse all possible key combinations in consideration of the most unfavorable situation. Because there are only 2^{192} possible values of *Pos* and *IDF*, in the worst case, the attacker needs to try 2^{192} possible values to ensure the correct guess of *SK*. Although it is possible to obtain the session key by guessing in theory, in practice—because the number of attempts required is extremely large—and we mentioned earlier that the random location parameter *Pos* of the session key is dynamic, it can be considered that this attack mode is very difficult and almost infeasible in calculation.

5.2.7. Anti-Desynchronization Attacks

In this protocol, the same serial numbers $N_{s,g}$ are stored among the participating entities. When the synchronization occurs or the session is completed, the serial numbers

$N_{s,g}$ and H will be synchronized between the participating entities to ensure the normal operation of the protocol. Therefore, the protocol can resist desynchronization attacks.

5.2.8. Anti-Physical Attacks

In this protocol, PUF uses random differences of physical entities in the manufacturing process to construct the physical fingerprint of the device, which is unpredictable and difficult to copy, so it is difficult for an adversary to copy a device with the same physical fingerprint as the existing device. Therefore, the protocol can effectively resist physical attacks.

5.2.9. Anti-Modeling Attacks

In this protocol, what is transmitted on the channel and stored in the node is not the plaintext form of the response but its hash value, and the adversary cannot obtain the hash value, let alone deduce the response, so the reference of hash function increases the nonlinear relationship between the challenge and the output, making it more difficult to model the attack. Therefore, the protocol can resist modeling attacks.

5.3. BAN Logic Analysis

BAN logic is a formal verification scheme based on trust modal logic proposed by Burrows and others to solve the disadvantages of informal security analysis. Its main idea is to use formal trust mode to represent the protocol content of each step executed by the entity running the security protocol.

In the process of logical verification, it is necessary to establish a set of initial logical assumptions and express each step of protocol execution as an idealized logical formula. Then, the protocol is formally deduced and proved from the initial state by using the established initial assumptions. Finally, the security goal of the protocol is obtained. The content of BAN logic mainly includes two parts: logical semantics and inference rules.

5.3.1. Logical Semantics

The basic terms and symbols are as follows:

- A, B, S: the target subject or entity that needs authentication (A and B are general communication subjects and S is the server);
- K_{ab}, K_{as}, K_{bs}: the shared key between the target subjects to be authenticated (subscript indicates the subjects sharing the key);
- K_a, K_b, K_s: the public key corresponding to the authentication subject (public key for short);
- K_a^{-1}, K_b^{-1}, K_s^{-1}: the private key corresponding to the authentication subject (private key for short);
- N_a, N_b, N_s: temporary variables (or temporary values) generated by the corresponding subjects during the operation of the protocol;
- P, Q, R: the protocol subject that needs authentication.

The basic components are as follows:

- P $|\equiv X$ or P believes X: Subject P believes that (formula or message) X is true;
- P $\triangleleft X$ or P sees X: Subject P receives the message X (or the message contains X) sent by another subject;
- P $|\sim X$ or P said X: Subject P has sent the message X (or the message sent contains X);
- P $|\Rightarrow X$ or P controls X: Subject P has jurisdiction over message X;
- #(X) or fresh X: message X has not been used in the whole or as a part of the message in the previous session. That is, message X is fresh, and X is generally a temporary value;
- P $\overset{K}{\leftrightarrow}$ Q: Subjects P and Q communicate with each other through the shared key K, and no other subjects can obtain Key K except Subjects P and Q and their trusted subjects;

- $\xrightarrow{K}$ P or $\left|\xrightarrow{K}\right.$P: K is the public key of Subject P, and the corresponding private key K^{-1} can not be obtained by other subjects except Subject P; $\xrightarrow{K}$ P or $\left|\xrightarrow{K}\right.$P:
- P $\overset{X}{\rightleftharpoons}$ Q: Subject P and Subject Q share secret information X, and P and Q can authenticate each other's identities through information X;
- $\{X\}_K$: means the ciphertext $\{X\}_K$ obtained by encrypting plaintext X with key K, which is a brief expression of $\{X\}_K$ from P;
- <X>$_Y$: indicates a message composed of messages Y and X. Generally, Y is the secret information by default.

5.3.2. Inference Rules

There are 19 basic inference rules in 7 categories in BAN logic (the improved BAN logic proposed later partially supplements or modifies the inference rules). These reasoning rules are briefly introduced and explained below. This means that a conclusion is derived from a precondition (or a set of conditions).

Information Meaning Rules

The rule for the shared key K is shown in Formula (1):

$$\frac{\mathrm{P}|\equiv \mathrm{Q} \overset{K}{\leftrightarrow} \mathrm{P},\ \mathrm{P} \lhd \{X\}_K}{\mathrm{P}|\equiv \mathrm{Q}|\sim X} \tag{1}$$

This rule indicates that if Subject P believes that the key K is a shared key between itself and Q, and P has received the message $\{X\}_K$ encrypted by K, then Subject P believes or thinks that Subject Q has sent the message X.

Similarly, the rule of the corresponding public key K is shown in Formula (2):

$$\frac{\mathrm{P}|\equiv \xrightarrow{K} \mathrm{Q},\ \mathrm{P} \lhd \{X\}_{K^{-1}}}{\mathrm{P}|\equiv \mathrm{Q}|\sim X} \tag{2}$$

The rule of sharing secret information is shown in Formula (3):

$$\frac{\mathrm{P}|\equiv \mathrm{Q} \overset{Y}{\rightleftharpoons} \mathrm{P},\ \mathrm{P} \lhd \langle X\rangle_Y}{\mathrm{P}|\equiv \mathrm{Q}|\sim X} \tag{3}$$

Among them, the reasoning content and essence expressed by Formula (2) and Formula (3) are the same as Formula (1), but the encryption mode of Message X or the secret information used are different.

Temporary Variable Verification Rules

This rule is shown in Formula (4), which means that if Subject P believes that the message X is fresh, and Subject P believes that Subject Q has sent the message X, Subject P believes that Subject Q believes the message X. It should be noted that the message X here generally represents plaintext.

$$\frac{\mathrm{P}|\equiv \#(X),\ \mathrm{P}|\equiv \mathrm{Q}|\sim X}{\mathrm{P}|\equiv \mathrm{Q}|\equiv X} \tag{4}$$

Jurisdiction Rules

This rule is shown in Formula (5), which means that if Subject P believes that Subject Q has the right to govern X and Subject P believes that Subject Q believes the message X, then Subject P believes the message X.

$$\frac{\mathrm{P}|\equiv \mathrm{Q}|\Rightarrow X,\ \mathrm{P}|\equiv \mathrm{Q}|\equiv X}{\mathrm{P}|\equiv X} \tag{5}$$

Belief Rules

The belief rule mainly means that the premise that the subject P believes in the combined message is that there is and only the subject P believes in each component of the combined message, and vice versa. This set of rules contains four articles in total, as shown in the following formulas.

$$\frac{\mathrm{P}|\equiv X,\ \mathrm{P}|\equiv Y}{\mathrm{P}|\equiv (X,Y)} \tag{6}$$

$$\frac{\mathrm{P}|\equiv (X,Y)}{\mathrm{P}|\equiv X} \tag{7}$$

$$\frac{\mathrm{P}|\equiv \mathrm{Q}|\equiv (X,Y)}{\mathrm{P}|\equiv \mathrm{Q}|\equiv X} \tag{8}$$

$$\frac{\mathrm{P}|\equiv \mathrm{Q}|\sim (X,Y)}{\mathrm{P}|\equiv \mathrm{Q}|\sim X} \tag{9}$$

Message-Receiving Rules

This rule is mainly used to deduce the following conclusion: If Subject P receives a whole message (a combination of messages or an encrypted message), it proves that Subject P has received all the components in the whole message. If the message is encrypted, Subject P needs to have the corresponding encryption key. There are five message-receiving rules in total, as shown in the following formulas.

$$\frac{\mathrm{P} \lhd (X,Y)}{\mathrm{P} \lhd X} \tag{10}$$

$$\frac{\mathrm{P} \lhd \langle X \rangle_Y}{\mathrm{P} \lhd X} \tag{11}$$

$$\frac{\mathrm{P}|\equiv \mathrm{Q} \overset{K}{\leftrightarrow} \mathrm{P},\ \mathrm{P} \lhd \{X\}_K}{\mathrm{P} \lhd X} \tag{12}$$

$$\frac{\mathrm{P}|\equiv \overset{K}{\rightarrow} \mathrm{P},\ \mathrm{P} \lhd \{X\}_K}{\mathrm{P} \lhd X} \tag{13}$$

$$\frac{\mathrm{P}|\equiv \overset{K}{\rightarrow} \mathrm{Q},\ \mathrm{P} \lhd \{X\}_{K^{-1}}}{\mathrm{P} \lhd X} \tag{14}$$

Message Freshness Rule

The rule of message freshness is shown in Formula (15), which means that if a part of a message is fresh, then the whole message is also fresh.

$$\frac{\mathrm{P}|\equiv \#(X)}{\mathrm{P}|\equiv \#(X,Y)} \tag{15}$$

Shared Secret Key Rule

Shared secret key rules are shown in the following formulas:

$$\frac{\mathrm{P}|\equiv \mathrm{R} \overset{K}{\leftrightarrow} \mathrm{R}\prime}{\mathrm{P}|\equiv \mathrm{R}\prime \overset{K}{\leftrightarrow} \mathrm{R}} \tag{16}$$

$$\frac{\mathrm{P}|\equiv \mathrm{Q}|\equiv \mathrm{R} \overset{K}{\leftrightarrow} \mathrm{R}\prime}{\mathrm{P}|\equiv \mathrm{Q}|\equiv \mathrm{R}\prime \overset{K}{\leftrightarrow} \mathrm{R}} \tag{17}$$

$$\frac{\mathrm{P}|\equiv \mathrm{R} \overset{X}{\rightleftharpoons} \mathrm{R}\prime}{\mathrm{P}|\equiv \mathrm{R}\prime \overset{X}{\rightleftharpoons} \mathrm{R}} \tag{18}$$

$$\frac{\mathrm{P}|\equiv \mathrm{Q}|\equiv \mathrm{R} \overset{X}{\rightleftharpoons} \mathrm{R}\prime}{\mathrm{P}|\equiv \mathrm{Q}|\equiv \mathrm{R}\prime \overset{X}{\rightleftharpoons} \mathrm{R}} \tag{19}$$

BAN logic formally verifies the function and security of the security protocol through the above seven types of logic rules. However, due to the dependence of BAN logic on initial assumptions, formal analysis based on BAN logic may still be subjective, so general BAN logic only conducts formal security analysis for security protocols with simple protocol processes.

The protocol proposed in this paper needs to meet the following four objectives:

$$\text{Goal1} : GatewayNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)$$

$$\text{Goal2} : GatewayNode| \equiv SensorNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)$$

$$\text{Goal3} : SensorNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)$$

$$\text{Goal4} : SensorNode| \equiv GatewayNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)$$

First, the messages exchanged in the protocol are converted into an ideal form, as shown below:

$$\text{Msg1} : GatewayNode \rightarrow SensorNode : \left\langle Pos, SID_s, SID_g, N_{s,g} \right\rangle$$

$$\text{Msg2} : SensorNode \rightarrow GatewayNode : \left\langle ID_s, N_{s,g}, R, SID_s, SID_g, Pos' \right\rangle$$

Secondly, some preliminary assumptions of the agreement are listed:

$$\begin{aligned}
&\text{A1} : SensorNode| \equiv \#\left(ID_g, SID_g\right)\\
&\text{A2} : GatewayNode| \equiv \#(ID_s, SID_s)\\
&\text{A3} : GatewayNode| \equiv GatewayNode \leftrightarrow SensorNode\\
&\text{A4} : SensorNode| \equiv GatewayNode \leftrightarrow SensorNode\\
&\text{A5} : SensorNode| \equiv GatewayNode \Rightarrow GatewayNode \overset{SK}{\leftrightarrow} SensorNode
\end{aligned}$$

Thirdly, based on BAN logic rules and the above assumptions, the proof process is as follows:

(1) According to the message Msg1, we can get:

$$\text{S1} : SensorNode \lhd \left\langle Pos, SID_s, SID_g, N_{s,g} \right\rangle$$

(2) According to assumptions A4, S1 and message meaning rules, we can obtain:

$$\text{S2} : SensorNode| \equiv GatewayNode| \sim (Pos, SID_s, SID_g, N_{s,g})$$

(3) According to the assumption A1, we can obtain:

$$\text{S3} : SensorNode| \equiv \#(Pos, SID_s, SID_g, N_{s,g})$$

(4) According to S3, S2 and temporary value verification rules, we can obtain:

$$\text{S4} : SensorNode \models GatewayNode \models (Pos, SID_s, SID_g, N_{s,g})$$

(5) According to the message Msg2, we can obtain:

$$\text{S5} : \text{GatewayNode} \lhd \langle \text{ID}_s, \text{N}_{s,g}, \text{R}, \text{SID}_s, \text{SID}_g, \text{Pos}' \rangle$$

(6) According to assumptions A3, S5 and message meaning rules, we can obtain:

$$\text{S6} : \text{Gateway}Node| \equiv SensorNode| \sim (ID_s, N_{s,g}, R, SID_s, SID_g, Pos')$$

(7) According to the hypothesis A2, we can obtain:

$$\text{S7} : GatewayNode| \equiv \#(ID_s, N_{s,g}, R, SID_s, SID_g, Pos)$$

(8) According to S7, S6 and provisional value verification rules, we can obtain:

$$\text{S8} : GatewayNode| \equiv SensorNode| \equiv (ID_s, N_{s,g}, R, SID_s, SID_g, Pos')$$

(9) According to S8 and session key rules, we can obtain:

$$\text{S9} : GatewayNode| \equiv SensorNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)(Goal2)$$

(10) According to S4 and session key rules, we can obtain:

$$\text{S10} : SensorNode| \equiv \text{Gateway}Node| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)(Goal4)$$

(11) According to S9, A5 and arbitration rules, we can obtain:

$$\text{S11} : GatewayNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)(Goal1)$$

(12) According to S10, A6 and arbitration rules, we can obtain:

$$\text{S12} : SensorNode| \equiv \left(GatewayNode \overset{SK}{\leftrightarrow} SensorNode\right)(Goal3)$$

5.4. Provable Security

This section uses a random oracle model (ROM) to carry out provable security analysis on the proposed scheme [34]. The participants are categorized into two types: Node_s and Node_g. Their corresponding instances t_1 and t_2 are denoted as $P_{N_s}^{t_1}$ and $P_{N_g}^{t_2}$, respectively. These instances are also referred to as oracles. For any participant, whether it is a Node_s or Node_g, the instance t is defined as P^t. Table 3 details a series of queries that an attacker

can utilize to launch attacks. Once instances t_1 and t_2 have successfully completed the processes of mutual authentication and key agreement, they are considered partners. If an attacker is unable to acquire the session key through the reveal (P^t) query, we call $P_{N_s}^{t_1}$ and $P_{N_g}^{t_2}$ fresh.

Table 3. Queries in the ROM Model.

Query	Function
Execute ($P_{N_s}^{t_1}$, $P_{N_g}^{t_2}$)	Through this query, the attacker can obtain the information over the channel between Node_s and Node_g.
CorruptNs ($P_y^{t_1}$)	Through this query, the attacker can extract the data stored in the memory of a compromised meter.
Reveal (P^t)	Through this query, the attacker can obtain session key SK negotiated between P^t and its partner.
Test (P^t)	In this query, the attacker asks P^t for the session key and gets a flipped "unbiased coin c" with probability.
Send (P^t, m)	Through this query, the attacker can launch an active attack by sending message m to P^t. P^t will process and send result to the attacker.

The attacker A has the ability to carry out several types of queries within polynomial time t_p against protocol P. These queries include Execute ($P_{N_s}^{t_1}$, $P_{N_g}^{t_2}$), CorruptNs ($P_y^{t_1}$), Reveal (P^t) and Send (P^t, m). Through these operations, A can obtain fresh instances. Additionally, A can attempt to guess the plaintext data c which may be obtained by Test(P^t) query. If A manages to guess data $c' = c$ successfully, the advantage that A can break the semantic security of protocol P's session key correctly in polynomial time t_p. Furthermore, the collision-avoidance hash operation $h(\cdot)$ is represented as a random oracle hash, PUF is expressed as HPUF, $\text{Adv}_\text{A}^{\text{ECDLP}}(t_p)$ represents the advantage in polynomial time t_p to compute t_z from the formula $\text{PUB}_z = t_z \cdot \text{P}$, and ECDLP stands for "elliptic curve discrete logarithm problem".

Theorem 1. $\text{Adv}_\text{A}^{\text{SG}}(t_p)$ *is defined as the probability of success for an attacker A to break the security of the we proposed SG and obtains the session key in polynomial time* t_p*. The advantages of the attacker are as follows*

$$Adv_A^{SG}(t_p) \leq \frac{q_h^2}{|Hash|} + \frac{q_p^2}{|HPUF|} + 2Adv_A^{ECDLP}(t_p). \tag{20}$$

where q_h^2 *represents the number of hash queries,* q_p^2 *represents the number of PUF queries,* $|Hash|$ *represents the Hash output space,* $|HPUF|$ *represents the HPUF output space and* $Adv_A^{ECDLP}(tp)$ *represents A's advantage in breaking ECDLP in time* t_p*.*

Proof of Theorem 1. Assuming that there is an attacker A who can break the security of the protocol in polynomial time t_p, after several games, the accumulated winning advantage of the attacker is calculated. We simulate five games, namely Game_j, (j = 0, 1, 2, 3, 4) to prevent the proof. Let Win_j be the event in where A can successfully guess the random bit c in Game_j and the advantage of A in Game_j can be defined as

$$\text{Adv}_\text{A}^{\text{SG}}(t_p) \;=\; \Pr\left[\text{Win}_j\right]. \tag{21}$$

The details of the Game_j, (j = 0, 1, 2, 3, 4) are provided as follows.

Game_0: This game simulates A's initialization attack. In this game, we can obtain:

$$\text{Adv}_{\text{A}}^{\text{SG}}(t_p) = \left|2\text{Pr}\left[\text{Win}_0\right] - 1\right|. \tag{22}$$

Game_1: This game simulates eavesdropping attacks, and the attacker can obtain all the messages of the public channel. The query Test (P^t) makes the attacker able to determine whether the session key is a real key or not. In fact, the session key SK = LEA (IDF', Pos) generated in the protocol where $IDF' = h_2$ ($SID_c \,||\, SID_h \,||\, R$). SID_c and SID_h may be obtained by attackers from public channels. However, for R and Pos, because R is a 32-bit random number and Pos is a 64-bit random number, if and only if A successfully guesses $R' = R$ and $Pos' = Pos$, the attacker can correctly obtain the session key SK, but they need to calculate it at most $2^{64} \cdot 2^{32}$ times within polynomial time t_p. On the other hand, the session key used in the designed protocol is dynamic, and because of ECDLP, the attacker cannot increase the probability of winning Game_1 through an eavesdropping attack. Therefore, the probability of Game_1 and Game_0 is equal. Therefore, we can obtain

$$\text{Pr}\left[\text{Win}_0\right] = \text{Pr}\left[\text{Win}_1\right]. \tag{23}$$

Game_2: This game simulates the attacker's active attack. This game maintains a hash list H_m. When receiving the attacker's h (m) query, if H = h (m) $\in H_m$, output H; on the contrary, randomly select a value and record it in H_m. In this game, we consider A launching an active attack through the Hash query. The messages {ID_g, $MPos$, V} and {ID_g, M, APS} are related to random numbers Pos and R. As a result, A can execute the hash query without causing a collision, and Game_1 and Game_2 are indistinguishable. According to the birthday paradox, the sum of the probability of hash collision and the probability of generating a message consistent with the original message is $\frac{q_h^2}{2|\text{Hash}|}$. We can obtain:

$$\left|\text{Pr}\left[\text{Win}_2\right] - \text{Pr}\left[\text{Win}_1\right]\right| \leq \frac{q_h^2}{2|\text{Hash}|} \tag{24}$$

Game_3: A can capture Node_s by executing CorruptNs and obtain {SID_g, ID_g, $N_{s,g}$, $ADDR_s$, HD_s} stored in memory. Since the data are encrypted for the secret information, A needs to perform PUF to recover the secret. However, once the context related to the PUF changes, the output will be different from the previous one even for the same challenge. Therefore, A cannot encounter any collision when executing the PUF query. We can obtain:

$$\left|\text{Pr}\left[\text{Win}_3\right] - \text{Pr}\left[\text{Win}_2\right]\right| \leq \frac{q_{\text{P}}^2}{2|\text{HPUF}|}. \tag{25}$$

Game_4: Since A can obtain {SID_g, ID_g, $N_{s,g}$, $ADDR_s$, HD_s} in Node_s in Game_3 and A can intercept the messages {ID_g, $MPos$, V} and {ID_g, M, APS} in the public channel in Game_1, in order to calculate SK = LEA (IDF', Pos), A needs to derive h_2 ($SID_c \,||\, SID_h \,||\, R$). Due to the computational complexity of ECDLP within polynomial time t_p, this is a costly operation. Therefore, we can obtain:

$$\left|\text{Pr}\left[\text{Win}_4\right] - \text{Pr}\left[\text{Win}_3\right]\right| \leq \text{Adv}_{\text{A}}^{\text{ECDLP}}(tp). \tag{26}$$

After the execution of the above-mentioned games, the session key contains random numbers, and the generation process is independent and random. It follows that:

$$\text{Pr}\left[\text{Win}_4\right] = \frac{1}{2}. \tag{27}$$

From Equations (22) and (23), we can obtain:

$$\frac{1}{2}\mathrm{Adv}_{\mathrm{A}}^{\mathrm{SG}}(t_p) = |\Pr[\mathrm{Win}_0] - \frac{1}{2}|. \tag{28}$$

From Equations (27) and (28), we can obtain:

$$\frac{1}{2}\mathrm{Adv}_{\mathrm{A}}^{\mathrm{SG}}(t_p) = |\Pr[\mathrm{Win}_4] - \Pr[\mathrm{Win}_3]|. \tag{29}$$

On the other hand, through the triangle inequality, we know that:

$$\begin{aligned} &|\Pr[\mathrm{Win}_1] - \Pr[\mathrm{Win}_4]| \le |\Pr[\mathrm{Win}_1] - \Pr[\mathrm{Win}_2]| + |\Pr[\mathrm{Win}_2] - \Pr[\mathrm{Win}_4]| \le \\ &|\Pr[\mathrm{Win}_1] - \Pr[\mathrm{Win}_2]| + |\Pr[\mathrm{Win}_2] - \Pr[\mathrm{Win}_3]| + |\Pr[\mathrm{Win}_3] - \Pr[\mathrm{Win}_4]|. \end{aligned} \tag{30}$$

Combining the above formulas, we can obtain:

$$\mathrm{Adv}_{\mathrm{A}}^{\mathrm{SG}}(t_p) \le \frac{q_h^2}{|\mathrm{Hash}|} + \frac{q_{\mathrm{P}}^2}{|\mathrm{HPUF}|} + 2\mathrm{Adv}_{\mathrm{A}}^{\mathrm{ECDLP}}(t_p).$$

The above proof shows that the protocol is secure, and it can successfully realize mutual authentication and session key agreement between sensor nodes ($\mathrm{Node_s}$) and gateway nodes ($\mathrm{Node_g}$). □

6. Performance Analysis

In this section, we perform a detailed comparison on security and costs in compute, communication and storage among our scheme and other five relevant authentication protocols, including the schemes of Yang et al. [35], Amin et al. [36], Sharif et al. [37], Das et al. [38], and Li et al. [39]. We stipulate that the smart meter platform is based on HiPerSmart card, whereas the service provider's infrastructure is powered by Pentium IV processors (Manufacturer: Intel Corporation; City: Hillsboro; Country: USA). Let T_P, T_B, T_H, and T_S denote the operation time for elliptic curve point multiplication, fuzzy extracting biometric data, hash, and symmetric encryption, respectively. According to [40], $T_H \approx 0.01$ ms, $T_S \approx 0.01$ ms, $T_P \approx 1.17$ ms, and these results are tested on an Arduino device. Moreover, according to the test time of [15], it can be assumed that $T_B \approx T_P$. Additionally, according to [39,41], the identities, the tolerance error value, and the public reproduction parameters are set to 128 bits. ECC point, random numbers, and hash function outputs are set to 160 bits. The timestamp and the counter n_0 are 32 bits. For the length of other related parameters, please refer to Table 1. Given that our schemes are mutual schemes between two nodes, regardless of other participation links such as cloud centers and users, the comparison in Table 3 is only at two ends—the sensor nodes and the gateway nodes. Three kinds of costs among relevant authentication schemes are recorded in Table 4, and we will provide a detailed analysis of these costs in the subsequent sections.

From a security point of view, as shown in Table 5, our scheme is the only one that meets all eight evaluation criteria. All of the compared schemes cannot resist node capture attacks, and only the schemes of Li et al. [39] and ours provide forward secrecy.

From a computation cost point of view, as shown in Figure 5, our scheme is competitive. The computation cost on the sensor node is the smallest (0.07 ms), and its cost on the gateway/access node (0.07 ms) is a little more than that of Sharif et al.'s (0.05 ms [37]). In fact, the difference in time spent by these schemes is imperceptible to the user. Thus, the computation cost on the gateway/access node is acceptable and has little impact on the user experience.

Table 4. Three kinds of costs among relevant authentication schemes.

Scheme	Ref.	R.	Computational Cost (ms)		Communication Cost (bits)		Storage Cost (bits)	
			$Node_c$	$Node_h$	$Node_c$	$Node_h$	$Node_c$	$Node_h$
Yang et al.	[35]	6	$8T_H \approx 0.08$	$7T_H \approx 0.07$	960	480	480	288
Amin et al.	[36]	4	$9T_H \approx 0.09$	$10T_H \approx 0.1$	672	1504	640	320
Sharif et al.	[37]	6	$11T_H + T_B \approx 1.28$	$5T_H \approx 0.05$	1024	352	608	288
Das et al.	[38]	3	$18T_H + T_B \approx 1.35$	$9T_H \approx 0.09$	672	352	1184	704
Li et al.	[39]	4	$3T_P + T_B + 7T_H \approx 4.75$	$2T_P + 4T_H \approx 2.38$	640	480	768	288
ours	-	5	$2T_S + 5T_H \approx 0.07$	$2T_S + 5T_H \approx 0.07$	472	352	432	432

Table 5. Security comparison among relevant authentication schemes.

Scheme	Ref.	Security Evaluation Criteria							
		SEC1	SEC2	SEC3	SEC4	SEC5	SEC6	SEC7	SEC8
Yang et al.	[35]	×	√	×	√	√	×	×	√
Amin et al.	[36]	×	√	×	√	√	√	×	√
Sharif et al.	[37]	×	×	×	×	×	√	×	×
Das et al.	[38]	×	×	×	×	×	√	×	×
Li et al.	[39]	√	√	×	√	√	√	×	√
ours	-	√	√	√	√	√	√	√	√

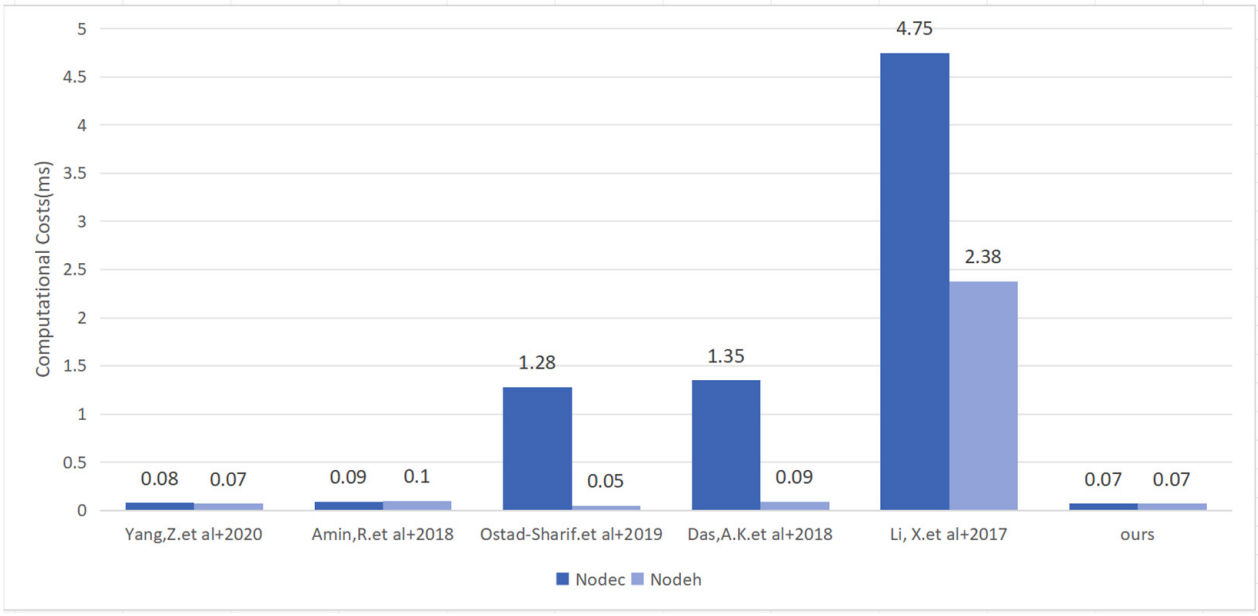

Figure 5. Computational costs comparison in all related schemes. The results are shown for various approaches, including Yang, Z. et al. (2020) [35], Amin, R. et al. (2018) [36], Ostad-Sharif et al. (2019) [37], Das, A.K. et al. (2018) [38], Li, X. et al. (2017) [39], and our proposed method.

As shown in Figure 6, from the communication cost point of view, we have the best performance in the comparison schemes. The communication cost on sensor node and gateway/access node is the smallest.

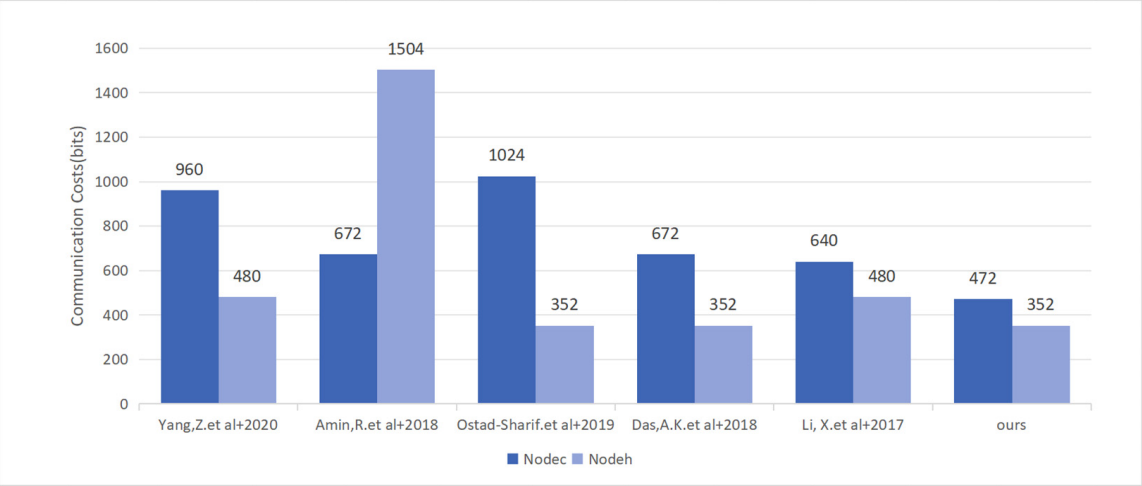

Figure 6. Communication costs comparison in all related schemes. The results are shown for various approaches, including Yang, Z. et al. (2020) [35], Amin, R. et al. (2018) [36], Ostad-Sharif et al. (2019) [37], Das, A.K. et al. (2018) [38], Li, X. et al. (2017) [39], and our proposed method.

As shown in Figure 7, from storage cost point of view, our scheme performs best on the sensor node (432-bit). On the gateway/access node (432-bit), it performs much better than that in Das et al.'s scheme [38] (704 bit) but a little worse than that of others (288-bit [35], 320-bit [36], 288-bit [37], 288-bit [39]). In summary, the computation cost of our scheme on the IoT device side is reasonable and competitive, achieving a better balance between security and performance.

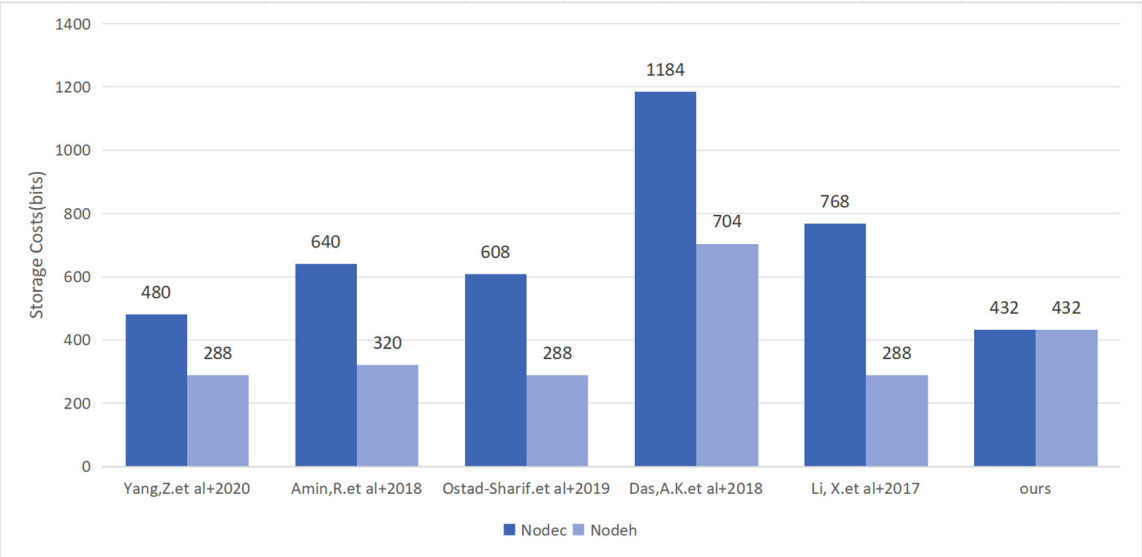

Figure 7. Storage costs comparison in all related schemes. The results are shown for various approaches, including Yang, Z. et al. (2020) [35], Amin, R. et al. (2018) [36], Ostad-Sharif et al. (2019) [37], Das, A.K. et al. (2018) [38], Li, X. et al. (2017) [39], and our proposed method.

In conclusion, the proposed scheme satisfies all eight evaluation criteria. Additionally, our scheme achieves relatively small computational costs due to the lightweight encryption and decryption method and the minimum communication costs on each node. Therefore, we can think that our protocol scheme achieves the same or even higher security than the reference scheme, and at the same time, it achieves less computational overhead. Thus, the proposed scheme is especially suitable for a smart grid with relatively high requirements on safety and performance.

7. Conclusions

In smart grid (SG) networks, the transmission of power control and consumption data predominantly occurs over public channels. In order to safeguard the confidentiality of power control information and the privacy of data exchanged between smart grids and service providers across communication channels, we designed a lightweight authentication protocol based on symmetric encryption/decryption and physical unclonable functions (PUFs). This protocol can ensure mutual authentication and facilitates session key agreement between each node in the grid. A comprehensive informal analysis demonstrates that the proposed protocol we designed can resist nine common attacks. Furthermore, through rigorous BAN logic analysis, we reconfirm the security of our protocol, validating its capacity to effectively achieve bidirectional authentication and session key agreement between gateway nodes and sensor nodes. Additionally, we establish the semantic security of the protocol by utilizing the ProVerif tool to formally verify that an adversary is unable to compute or trace the session key. In comparison to five pertinent authentication protocols, our results indicate that this protocol requires fewer resources while providing improved security functions. Additionally, given that the Dolev–Yao adversary model utilized in ProVerif cannot describe many other adversary capabilities, in the future work, we plan to focus on ProVerif studies to enhance its ability to characterize more attacker capabilities and enable to have more comprehensive validation functions.

Author Contributions: Conceptualization, L.L.; Methodology, Y.G.; Validation, C.W.; Formal analysis, H.H.; Writing—original draft, C.A.; Project administration, X.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Project of State Grid Zhejiang Electric Power Co., Ltd. "Research and Application of Load-side Security Technology for Sensing Equipment and Networking" grant number [5211JH240008].

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Yu Guo, Lifeng Li and Xu Jin were employed by the State Grid Jinhua Electric Power Company, Chunyan An, Chenyu Wang and Hairui Huang were employed by the China Electric Power Research Institute. The authors declare that this study received funding from State Grid Zhejiang Electric Power Co., Ltd. The funder had the following involvement with the study: Security requirements analysis.

References

1. Fadel, E.; Gungor, V.C.; Nassef, L.; Akkari, N.; Malik, M.A.; Almasri, S.; Akyildiz, I.F. A survey on wireless sensor networks for smart grid. *Comput. Commun.* **2015**, *71*, 22–33. [CrossRef]
2. Fang, X.; Misra, S.; Xue, G.; Yang, D. Smart Grid—The New and Improved Power Grid: A Survey. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 944–980. [CrossRef]
3. El Brak, M.; Essaaidi, M. Wireless sensor network in smart grid technology: Challenges and opportunities. *Int. Conf. Sci. Electron.* **2012**, *6*, 578–583.
4. El Brak, M.; El Brak, S.; Essaaidi, M.; Benhaddou, D. Wireless Sensor Network applications in smart grid. In Proceedings of the 2014 International Renewable and Sustainable Energy Conference, Ouarzazate, Morocco, 17–19 October 2014; pp. 587–592.
5. Tashtarian, F.; Montazerolghaem, A.; Abbasi, M. Distributed Lifetime Optimization of Wireless Sensor Networks in Smart Grid. In Proceedings of the 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–15 August 2018; pp. 240–243.
6. Monshi, M.M.; Mohammed, O.A. A study on the efficient wireless sensor networks for operation monitoring and control in smart grid applications. In *2013 Proceedings of IEEE Southeastcon, Jacksonville, FL, USA, 4–7 April 2013*; Institute of Electrical and Electronics Engineers: New York, NY, USA, 2013; pp. 1–5.
7. Houssaini, D.E.; Khriji, S.; Besbes, K.; Kanoun, O. Performance Analysis of Received Signal Strength and Link Quality in Wireless Sensor Networks. In Proceedings of the 2018 15th International Multi-Conference on Systems, Signals & Devices (SSD), Yasmine Hammamet, Tunisia, 19–22 March 2018; pp. 173–178.

8. Kazmi, S.A.A.; Hasan, S.F.; Shin, D.R. Analyzing the integration of Distributed Generation into smart grids. *IEEE Conf. Ind. Electron. Appl.* **2015**, *10*, 762–766.
9. Kumar, S.; Pandey, A.; Goswami, P.; Pentayya, P.; Kazi, F. Analysis of Mumbai Grid Failure Restoration on Oct 12, 2020: Challenges and Lessons Learnt. *IEEE Trans. Power Syst.* **2022**, *37*, 4555–4567. [CrossRef]
10. 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]
11. Wu, D.; Zhou, C. Fault-tolerant and scalable key management for smart grid. *IEEE Trans. Smart Grid* **2011**, *2*, 375–381. [CrossRef]
12. Xia, J.; Wang, Y. Secure Key Distribution for the Smart Grid. *IEEE Trans. Smart Grid* **2012**, *3*, 1437–1443. [CrossRef]
13. Odelu, V.; Das, A.K.; Wazid, M.; Conti, M. Provably Secure Authenticated Key Agreement Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2018**, *9*, 1900–1910. [CrossRef]
14. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S. LAKAF: Lightweight Authentication and Key Agreement Framework for Smart Grid Network. *J. Syst. Archit.* **2021**, *116*, 102053. [CrossRef]
15. 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.* **2021**, *17*, 4425–4436. [CrossRef]
16. 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]
17. Park, J.H.; Kim, M.; Kwon, D. Security Weakness in the Smart Grid Key Distribution Scheme Proposed by Xia and Wang. *IEEE Trans. Smart Grid* **2013**, *4*, 1613–1614. [CrossRef]
18. Tsai, J.-L.; Lo, N.-W. Secure Anonymous Key Distribution Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
19. 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]
20. Schnorr, C.-P. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology—CRYPTO '89 Proceedings*; Springer: Berlin/Heidelberg, Germany, 1990; pp. 239–252.
21. Baruah, B.; Dhal, S. An Authenticated Key Agreement Scheme for Secure Communication in Smart Grid. In Proceedings of the 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 5–9 January 2021; pp. 447–455.
22. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S.; Mishra, D. Palk: Password-based Anonymous Lightweight Key Agreement Frame-work for Smart Grid. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106121. [CrossRef]
23. Chaudhry, S.A. Correcting PALK: Password-based Anonymous Lightweight Key Agreement Framework for Smart Grid. *Int. J. Elect. Power Energy Syst.* **2021**, *125*, 106529. [CrossRef]
24. Gope, P.; Sikdar, B. Privacy-aware Authenticated Key Agreement Scheme for Secure Smart Grid Communication. *IEEE Trans. Smart Grid* **2019**, *10*, 3953–3962. [CrossRef]
25. 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]
26. Liu, F.; Yan, Y.; Sun, Y.; Liu, J.; Li, D.; Guan, Z. Extremely Lightweight PUF-based Batch Authentication Protocol for End-edge-cloud Hierarchical Smart Grid. *Secur. Commun. Netw.* **2022**, *2022*, 9774853. [CrossRef]
27. Bagheri Nasour, B.; Bendavid, Y.; Safkhani, M.; Rostampour Samad, S. Smart Grid Security: A PUF-Based Authentication and Key Agreement Protocol. *Future Internet* **2024**, *16*, 9. [CrossRef]
28. Alidaee, B.; Wang, H.B.; Huang, J.S.; Lutfu, S. Integrating Statistical Simulation and Optimization for Redundancy Allocation in Smart Grid Infrastructure. *Energies* **2024**, *17*, 225. [CrossRef]
29. Gope, P.; Lee, J.; Quek, T.Q. Lightweight and Practical Anonymous Au thentication Protocol for RFID Systems Using Physically Unclonable Functions. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2831–2843. [CrossRef]
30. *IEEE Std 802.1X-2020 (Revision of IEEE Std 802.1X-2010 Incorporating IEEE Std 802.1Xbx-2014 and IEEE Std 802.1Xck-2018)*; IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access Control. IEEE: Piscataway, NJ, USA, 2020; pp. 1–289.
31. *IEC 62351-3:2023*; IEC 62351-3: Power Systems Management and Associated Information Exchange—Data and Communications Security—Part 3: Communication Network and System Security. Profiles including TCP/IP. International Electrotechnical Commission (IEC): Geneva, Switzerland, 2023.
32. 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]
33. Blanchet, B.; Smyth, B.; Cheval, V.; Sylvestre, M. *ProVerif2.03: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*; Tech. Rep. EP/D076625/2; INRIA: Paris, France, 2020.

34. Bellare, M.; Rogaway, P. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, VA, USA, 3–5 November 1993; ACM: New York, NY, USA, 1993; pp. 62–73.
35. Yang, Z.; He, J.; Tian, Y.; Zhou, J. Faster Authenticated Key Agreement with Perfect Forward Secrecy for Industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6584–6596. [CrossRef]
36. Amin, R.; Kumar, N.; Biswas, G.; Iqbal, R.; Chang, V. A Light-weight Authentication Protocol for IoT–enabled Devices in Distributed Cloud-computing Environment. *Future Gener. Comput. Syst.* **2018**, *78*, 1005–1019. [CrossRef]
37. Ostad-Sharif, A.; Arshad, H.; Nikooghadam, M.; Abbasinezhad-Mood, D. Three-party Secure Data Transmission in IoT Networks through Design of a Lightweight Authenticated Key Agreement Scheme. *Future Gener. Comput. Syst.* **2019**, *100*, 882–892. [CrossRef]
38. Das, A.K.; Wazid, M.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.P.C. Biometrics-based Privacy-preserving User Authentication Scheme for Cloud-based Industrial Internet of Things Deployment. *IEEE Internet Things J.* **2018**, *5*, 4900–4913. [CrossRef]
39. Li, X.; Niu, J.; Bhuiyan, M.Z.A.; Wu, F.; Karuppiah, M.; Kumari, S. A Robust ECC-based Provable Secure Authentication Protocol with Privacy Preserving for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3599–3609. [CrossRef]
40. Zahoor, A.; Mahmood, K.; Saleem, M.A.; Badar, H.M.S.; Le, T.-V.; Das, A.K. Lightweight Authenticated Key Agreement Protocol for Smart Power Grid Systems Using PUF. *IEEE Open J. Commun. Soc.* **2024**, *5*, 3568–3580. [CrossRef]
41. Wang, D.; Wang, P. Two Birds with One Stone: Two-factor Authentication with Security Beyond Conventional Bound. *IEEE Trans. Dependable Secur. Comput.* **2018**, *15*, 708–722. [CrossRef]