Hindawi
Security and Communication Networks
Volume 2022, Article ID 9774853, 14 pages
https://doi.org/10.1155/2022/9774853

Research Article

Extremely Lightweight PUF-based Batch Authentication Protocol for End-Edge-Cloud Hierarchical Smart Grid

Feifei Liu,[1,2] **Yu Yan**,[1] **Yu Sun**,[1] **Jianwei Liu**,[1] **Dawei Li**,[1] **and Zhenyu Guan**[1]

[1]*School of Cyber Science and Technology, Beihang University, Beijing, China*
[2]*Henan Key Laboratory of Network Cryptography Technology, Henan, China*

Correspondence should be addressed to Yu Sun; sunyv@buaa.edu.cn and Jianwei Liu; liujianwei@buaa.edu.cn

Received 8 August 2022; Revised 18 October 2022; Accepted 20 October 2022; Published 29 November 2022

Academic Editor: Kuo-Hui Yeh

Smart grid is gradually replacing traditional grid with two-way communication and improved management. Besides the efficiency and reliability it brings, the smart grid is inevitably fraught with rampant physical and cyber-attack. Although several physical unclonable function (PUF)-based schemes have been proposed, they are unsuited to the end-edge-cloud hierarchical smart grid. This paper proposes a PUF-based batch authentication and key agreement protocol, which protects both meters and gateways and provides end-to-end authentication between meters and the server. By offloading heavy operations from field devices to the server, the computation overhead is reduced substantially. Moreover, we innovatively devolve batch authentication and access control to the gateway, which additionally decreases downlink communication and signaling cost, and is superior to most recent schemes. Our protocol is proved by Tamarin under extended Dolev–Yao adversary and the Real-or-Random model and is evaluated to be secure against various attacks. Using extremely lightweight operations, our protocol is implemented on the MSP430FR5969 microcontroller.

1. Introduction

1.1. Background. Featuring with bidirectional communication, real-time monitoring, and intelligent control, the smart grid is gradually replacing traditional grid due to the contradiction between diminishing fossil fuels and growing electricity production cost [1]. By utilizing the advanced metering infrastructure (AMI), the smart grid reduces power wastage and brings much convenience.

Smart grid is a complicated system with tight integration of power plane and data plane, as illustrated in Figure 1. In power plane, electricity generated from thermal heat, wind energy, and solar radiation is transmitted, distributed, and finally for consumption. Data plane forms a typical end-edge-cloud hierarchical architecture that deals with measurement and information exchange, and mainly consists of meters, gateways, and the server [2]. As the main metering field units for electricity consumers, meters are located in consumers' home, responsible for the electricity consumption, and are embedded equipment with limited computation and storage capabilities. Gateway is deployed outside to coordinate the communication among massive meters and the server. To mitigate the burden at the server side, a gateway is mainly in charge of batch authentication and access control authority against malicious meters. Server is the electricity provider, which collects users' power consumption data and distributes command to achieve real-time billing and mutual benefit.

As a promising technology to support massive connections, Narrow-Band Internet of Things (NB-IoT) [3] has been widely explored in smart grid [4, 5] for low-cost and large capacity. In practice, the intermediate gateway statically wires meters through bus protocol at one side, and the other side connects the server wirelessly through its NB-IoT air-interface. Thus, smart grid forms a Bus-NB-IoT hierarchical network. As can be seen in the inset of Figure 1, the typical Commercial Off-The-Shelf (COTS) gateway supports both the NB-IoT module and some industrial bus interfaces. Once powered on, the gateway will request for an authentication of m meters that it connects within a fixed period of

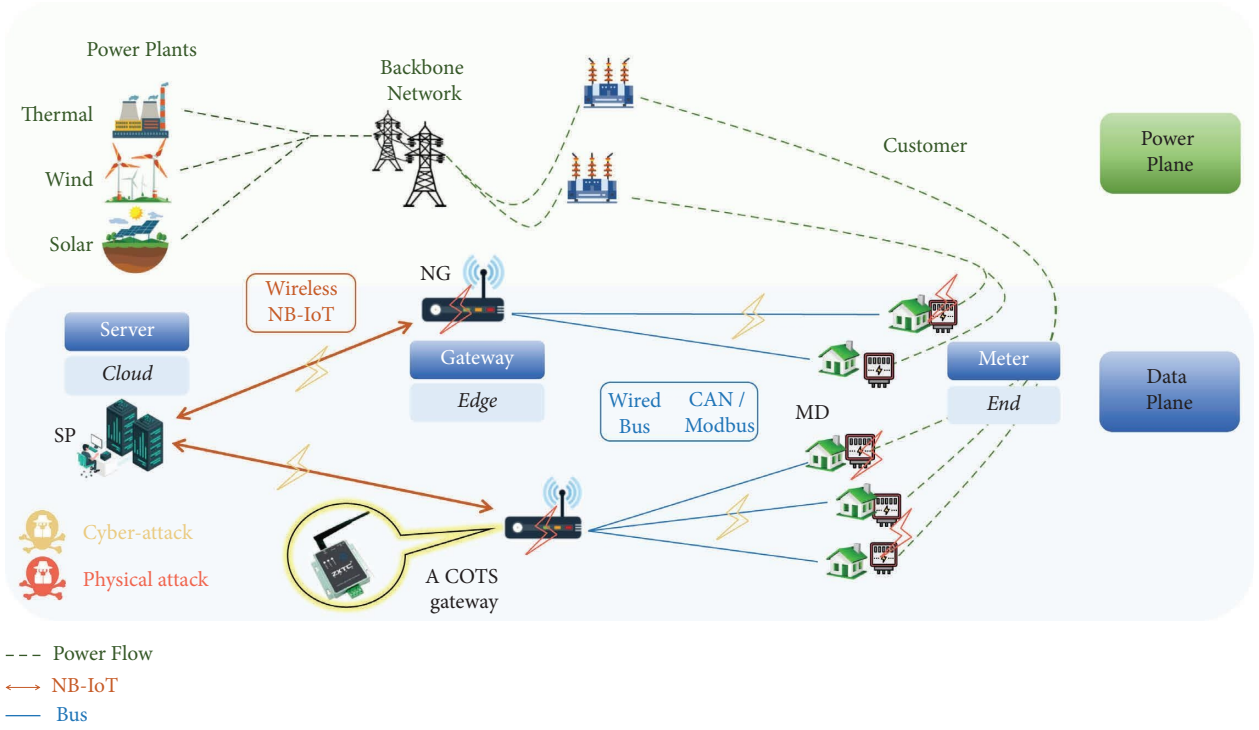

Figure 1: Smart grid forms an *end-edge-cloud* architecture that is connected by the Bus-NB-IoT hierarchical network. Power plane and data plane are integrated tightly for mutual communication and management. A COTS gateway is equipped with both NB-IoT module and industrial bus interfaces.

time to increase efficiency. The gateway also plays an important role in authority filtering to reduce the attacks on the server. Nevertheless, there are few secure authentication schemes among meters, gateway, and the server.

The downlink transmission in NB-IoT is far more expensive than the uplink transmission since it should establish a new nonaccess stratum connection patterned on a completed access authentication process [6]. However, recent schemes seldom take signaling optimization into consideration to minimize the expensive downlink transmission.

Although the smart grid provides numerous benefits, there are still a variety of security concerns to this complicated cyber-physical system. Smart grid is under furious threats from both wireless NB-IoT [7, 8] and the wired bus [9, 10]. Connecting meters and gateways, the wired bus is intrinsically vulnerable to cyber-attacks [11] owing to unguaranteed confidentiality, integrity, and access control. In the smart grid networks, external adversaries might eavesdrop, manipulate, replay messages transmitted among meters, gateways, and the server. Furthermore, internal adversaries may launch impersonation attack and deduce confidential information from intermediate steps. Several works have confirmed that instant electricity consumption data will reveal the activities of family members, house occupancy, and economic status. Authentication and Key Agreement (AKA) is an essential technique to provide security for the smart grid. After a successful AKA protocol, the server can confirm the validity of the meter who claimed to be. A fresh session key can guarantee the confidentiality of messages transmitted in the smart grid. However, there is still no integrated secure AKA protocol for the Bus-NB-IoT hierarchical network. We will give a detailed explanation of the limitation of recent AKA schemes targeting smart grid in the next section.

Physical attack is another security challenge where the confidential information stored in the non-volatile memory (NVM), such as long-term secret key, is stole, copied, or replaced by the attacker. One alternative remedy is to embed a tamper resistant hardware in each device, which results in huge commercial cost due to millions of field devices in the smart grid.

1.2. Related Work. Although as a hierarchal network, the smart grid in vast majority of authentication protocols is partially modeled as the communication between meters and the server [12–19] or meters and gateways [20–23]. The former ignores the actual architecture and the authority filtering function of gateway, while the latter exaggerates the security assumption of gateway. And only several schemes [6, 13, 14, 17, 20, 24–26] take physical attack into consideration. A comparison of recent schemes is shown in Table 1. The limitation of recent AKA schemes for smart grid lies in three aspects:

1.2.1. These Schemes Cannot Resist Physical Attack Since the Security Relies on the Confidentiality of Long-Term Secret Key. Li et al. proposed an authentication scheme between the meter gateway of home area network and of

Table 1: Comparisons of recent schemes.

Property	[13]	[14]	[15]	[24]	[20]	[27]	[16]	[17]	[18]	[19]	[21]	[22]	[23]	[28]	[29]	[25]	[30]	[31]
*P*1	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
*P*2	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
*P*3	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
*P*4	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
*P*5	✓	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
*P*6	—	—	—	✗	—	✓	—	—	—	—	—	—	—	✗	✗	✗	—	—

*P*1 : PUF-based; *P*2 : hierarchical network among three entities; *P*3 : batch authentication mechanism; *P*4 : gateway with authentication and access control ability; *P*5 : no key storage at MD side; *P*6 : semitrusted gateway. '✓' means the scheme can satisfy the property. '✗' means the scheme does not support the property. '—' means the property is not considered in this scheme (in Tables 1 and 3) or the entity is not involved in this operation (in Tables 4–8).

neighborhood area network [30]. Later, Wu et al. pointed out Li's vulnerability in the impersonation attack and DoS attack in some realistic scenarios. To overcome the weakness, Wu et al. proposed an improved anonymous message authentication scheme [31], where meter identities are hidden by Diffie–Hellman parameters. To achieve mutual authentication and anonymous key distribution, Tsai and Lo utilized identity-based signature and encryption scheme without the help of the trusted anchor [15]. Later, Odelu et al. presented an enhanced one to overcome Tsai's vulnerability [16], but the identity-based cryptography also brought heavy bilinear pairing operation on both meter and server side. These above schemes can only merely secure the communication channel, but the smart grid requires further protections at physical degree. Table 1 shows that these schemes do not meet condition P1.

1.2.2. The System Architecture is not Practical for Hierarchal Smart Grid with End-Edge-Cloud Structure. As a protocol that aims at the secure communication between meters and the gateway, Kaveh and Mosavi proposed a lightweight mutual authentication and message reporting protocol based on physical unclonable function (PUF) [20]. However, PUF response is literally used as encryption key, and it needs more expensive strong PUF to support frequent update of Challenge-Response-Pair (CRP). The PUF was introduced in Boyapally's work [17] to protect meters against physical attacks. Considering the constrained computation capabilities of meters, the protocol is operationally asymmetric. A physics-based attack named load modification attack is also implemented to demonstrate the strength of the scheme. The security assumption of the vulnerable gateway is exaggerated in these schemes, where the fully-trusted gateway is given direct access to the CRP database. However, this is infeasible in reality (violates condition P2 in Table 1) since gateways residing at open environment are also prone to various attacks. Following the concept of computational asymmetry, Wang et al. proposed a PUF-based lightweight AKA protocol for edge IoT nodes [26]. To acclimatize environmental noise, the reverse fuzzy extractor is employed, which also mitigated the computation burden at end devices. This scheme is claimed to resist desynchronization attack. However, when massive meters connected to the server at the same time, one-by-one authentication will lead to network congestion, which means condition P3 is not satisfied. Also, these schemes ignored the important authentication and aggregation function of the gateway, as illustrated as condition P4.

1.2.3. Session Key is Leaked to the Curious Intermediate Entity. There are also some research studies focusing on authentication among three entities [24, 27, 29]. Uludag et al. [27] described the first hierarchical data collection scheme with the curious-but-honest gateway. In this scheme, all entities stored a pair of public and private keys, which may introduce public key infrastructure and additional key management. In Wazid's protocol [29], the gateway maintains all the values that made up of the session key, thus can deduce information between meters and the server. Using PUF, Badar et al. proposed an identity-based authentication protocol for power supply-line surveillance in a smart grid environment [24] yet still stores long-term keys in meters for authentication. This violates the original intention of using PUF, as well as property P5 in Table 1. Still, the session key was known among the three entities including the gateway, which do not meet condition P6.

1.3. Motivation and Contribution

1.3.1. Motivation. Our paper aims to provide a secure and lightweight batch authentication between the meter and server in end-edge-cloud hierarchal smart grid. Despite of numerous authentication schemes in smart grid, only a few of them are immune to physical attack. Besides, almost all recent schemes only involve the authentication between the meter and gateway or the meter and server, and is not secure against the honest-but-curious gateway. Heavy cost on communication, signaling, and especially computation are also urgent problems that need to be solved. As a result, targeting at the Bus-NB-IoT hierarchical smart grid, this paper proposes an integrated secure batch AKA protocol using PUF.

1.3.2. Contribution. In view of the above limitations, our contributions are outlined as follows:

(1) We propose an extremely lightweight batch AKA protocol in the hierarchical smart grid based on PUF which is secure against an honest-but-curious gateway. Extracting PUF as fingerprint, neither meters nor gateway are required to store long-term key, so all field devices are protected from physical

attack. Many security properties are also provided against cyber-attacks. The strength of our scheme is demonstrated by Tamarin and formal verification. In addition, we implemented our protocol on resource-constrained MSP430FR5969.

(2) This paper is one of the few that is aimed at the more pragmatic end-edge-cloud architecture of smart grid. Different from the common two-party authentication scheme, our protocol is designed for meters, gateway, and the server that are connected by the bus-NB-IoT hierarchical network. During authentication, the gateway is given the batch authentication capacity after receivan ing aggregated credential from the server. Furthermore, by filtering unverified meters, a gateway is capable of access control, which simultaneously mitigates DoS attack targeting at the server.

(3) By utilizing a reverse fuzzy extractor and extremely lightweight operations, our scheme has outstanding performance in computation, and can be easily adopted on a resource-constrained microcontroller. Our protocol also has promising performances in terms of communication, storage, and signaling overhead. To decrease the expensive downlink communication in NB-IoT from $O(m)$ to $O(1)$, batch authentication credentials are aggregated, which can also reduce the signaling cost significantly.

1.3.3. Outline. The remainder of this paper is organized as follows. We review the introduction of PUF and reverse fuzzy extractor in Section 2. The system model and security model are expressed in Section 3. Section 4 describes a specific construction for our proposed scheme. The security demonstration and performance analysis are conducted in Sections 5 and 6, respectively. Finally, we draw a conclusion in Section 7.

2. Preliminary

PUF is a mechanism when applying a challenge to a device, a unique response will be generated due to the physical structure diversity. Due to the manufacturing differences in hardware, response varies a lot from different devices even of the same type. Featuring this device-specific response, PUF can be extracted as fingerprints and have been widely used for authentication and key derivation. A qualified PUF should be unclonable, unidirectional, unique, and unpredictable.

PUF can be divided into strong PUF and weak PUF according the number of challenge-response pairs CRPs that it can generate. Although as a weak PUF with limited CRPs, the intrinsic SRAM PUF does not need additional hardware structure, e.g., FPGA, arbiter, to be implemented, thus is more friendly to resource-constrained devices like meters. Once powered on, the state of the SRAM cell will converge at a stable 1 or 0 bit, as an amplified feedback of tiny manufacturing variation. Taken cell address as a challenge, SRAM PUF will output the bit state of these address as response.

The response of a PUF may exhibit instability due to variations in environmental conditions such as temperature, voltage, thermal noise, as well as ageing. To overcome the instability caused by the external environment and achieve identical response for the same challenge, the reverse fuzzy extractor is introduced as an error correction scheme. The reverse fuzzy extractor is composed of two algorithms: FE.Gen and FE.Rec. FE.Gen is a probabilistic algorithm which takes the real-time PUF response r as input and outputs helper data hd, i.e., $hd \leftarrow \text{FE.Gen}(r)$. On the contrary, the key reproduction algorithm FE.Rec takes a noisy response $\hat{r}$ and helper data hd as input, and output the same cryptographic key r, i.e., $r \leftarrow \text{FE.Rec}(\hat{r}, hd)$. These two algorithms are found extremely different in terms of computation cost. FE.Rec usually takes far more execution time than FE.Gen. We can assume the response $\hat{r}$ during the enrollment phase to be a noisy one, so meters only need to execute FE.Gen for an accurate r in AKA phase. Then the server needs to recover the real-time response r from $\hat{r}$ using FE.Rec. Taking advantage of this property, the computation cost of the protocol can be substantially reduced.

3. System and Security Model

3.1. System Architecture. In this paper, we consider a Bus-NB-IoT hierarchical network for cloud-edge-end architecture, which is widely adopted in smart grid. Figure 1 shows that Meter Device (MD), Neighborhood Gateway (NG), and Service Provider (SP) are involved in our protocol. We illustrate their security abilities as follows.

SP : Since SP is located in the utility company, it can be considered as a fully trusted entity who holds a global secret key K_S for initial pseudoidentity generation. It also securely stores the secret responses of field devices as credentials.

NG : Since NG is deployed remotely, it is perceived as an honest-but-curious entity which is vulnerable to both cyber and physical attacks. Once jeopardized, a compromised NG will divulge any intermediate result it computes. As a result, NG is not allowed to access to neither other's secret credentials nor session keys.

MD : To avoid potential physical attack by external attackers, no secret key is stored in meters' NVM and PUF is used for secret key deviation. Similar to NG, curious MDs also pry about the confidential information of others.

3.2. Security Model. Besides the extensively used Dolev Yao (DY) model, we strengthen the attacker's ability to launch a physical attack. The abilities of extended DY attacker $\mathcal{A}$ are shown as follows:

(i) The communication channel among SP, NG, and MDs during authentication is completely controlled by the DY attacker $\mathcal{A}$, where $\mathcal{A}$ can eavesdrop,

inject, modify, and reorder the messages exchanged among these entities.

(ii) Targeting at field devices (NG and MD), $\mathscr{A}$ can capture them physically and extract the sensitive information, i.e., long-term secret key, that is stored in the memory by using side channel attack.

(iii) NG and MD are both assumed to be honest-but curious. They execute the protocol as ordered, but still intends to infer secret credential of others, and eavesdrops user privacy from metering messages.

4. Proposed Scheme

4.1. Protocol Overview. To overcome the computation inefficiency, the proposed scheme *offloads* the computation cost from meters/gateway to the powerful server. This section explains how to achieve it using reverse fuzzy extractor. Definitions of parameters are shown in Table 2.

4.2. Enrollment Phase. Before being deployed outside, MD_is and NG should register themselves to the SP as shown in Figure 2. Initially, MD_is and NG send their identities to SP. SP generates corresponding challenges c_*, pseudo identities $TID_* = H(K_s||ID_*)$, and sends them back. On receiving the challenges, MD_is and NG extract the registration responses as $\hat{r}_* = \mathrm{PUF}_*(c_*)$ and forwards them to SP. MD_is and NG store the $\langle ID_*, TID_*, c_*\rangle$ and SP maintains $\langle ID_*, \hat{r}_*\rangle$ of all the devices securely.

4.3. AKA Phase. During this phase, NG first gathers the helper data from MDs and forwards them to SP. Then it acquires aggregated credential to authenticate MDs in a batch. We describe this phase in Figure 3. Since meters and gateways are employed statically, it is unnecessary for the smart grid to execute the authentication protocol frequently, and SRAM PUF is enough for CRP update. Even if the same challenge is reused, credentials for authentication are generated with fresh nonce N_G.

STEP 1. NG chooses a nonce N_G and sends authentication request $M_1 = \langle \mathrm{TAG}_{\mathrm{Req}}, N_G\rangle$ to MDs through wired bus. On receiving the request from NG, each MD_i derives the real-time response $r_i = \mathrm{PUF}_i(c_i)$ and generates $hd_i \leftarrow \mathrm{FE.Gen}(r_i)$. Here, we assume the response in the AKA phase is the accurate one, which SP needs to run a time-consuming FE.Rec to recover from registration one. Then, the end-to-end session key for this round can be computed as $SK_i = H(r_i||N_G)$. Since there is only one pair of CRP being stored during the enrollment phase, CRP has to be updated in time. MD_i chooses another challenge c_i' for new response $r_i' = \mathrm{PUF}_i(c_i')$, and encrypts it using the session key to get $X_i = E_{SK_i}(r_i')$. To prevent entropy leakage caused by transmitting helper data in plaintext, hd_i is concealed in $hd_i' = h(ID_i\|c_i)\oplus hd_i$. Then, MD_i responds message $M_{2i} = \langle TID_i, hd_i', X_i\rangle$ back to NG.

STEP 2. NG also generates the real-time response r_G, computes the session key SK_G, derives new response r_G', and the encryption form $X_G = E_{SK_G}(r_G')$, $hd_G' = h(ID_G\|c_G)\oplus hd_G$ as MD does. Then, it sends message $M_3 = \langle\{M_{2i}\}, TID_G, hd_G', N_G, X_G, Q_1\rangle$ to SP, where $Q_1 = H(SK_{S-G}\| \{M_{2i}\}\|TID_G\|hd_G\|X_G)$ is for integrity verification.

STEP 3. Receiving M_3 from NG, SP finds all registration response $\hat{r}_*$ and true identity ID_* according to TID_* and recovers the helper data $hd_* = h(ID_*\|c_*)\oplus hd_*'$ and real-time response $r_* \leftarrow \mathrm{FE.}Rec(\hat{r}_*, hd_*)$. Then, SP can compute the session key and authenticate NG through verifying Q_1. The credential for each MD, i.e., $\mathrm{Auth}_i = H(SK_i\|N_G)$ are aggregated as $\mathrm{Auth} = \mathrm{Auth}_1\oplus\cdots\oplus\mathrm{Auth}_m$, which is then encrypted for $\mathrm{Auth}E = E_{SK_G}(\mathrm{Auth})$. The aggregated credential is sent to NG through $M_4 = \langle \mathrm{Auth}E, Q_2\rangle$, where $Q_2 = H(\mathrm{Auth}E\|N_G\|SK_G)$.

STEP 4. When NG receives the authentication response, it first verifies Q_2 to authenticate SP. Then, NG decrypts $\mathrm{Auth}E$ to get the aggregated credential. To hide it from adversaries, NG the computes the hashed credential $\mathrm{Auth}H = H(\mathrm{Auth})$ and broadcast $M_5 = \langle \mathrm{TAG}_{\mathrm{Auth}}, \mathrm{Auth}H\rangle$ on the bus.

STEP 5. On the arrival of M_5, each MD_i computes its Auth_i, $Q_{3i} = H(\mathrm{Auth}_i\|N_G)$, and also broadcast $M_{6i} = \langle TID_i, \mathrm{Auth}_i, Q_{3i}\rangle$ on the bus, which is also sent to the NG. While sending M_{6i}, each MD_i could obtain other m-1 Auth_i sent on the bus at the same time. Therefore, each MD_i could check whether $\mathrm{Auth}H$ equals $H(\mathrm{Auth}_1\oplus\cdots\oplus\mathrm{Auth}_m)$. If they are equal, MD_i authenticates SP and believes that it shares the same session key SK_i with SP. Then the new pseudoidentity for next round authentication is updated as $TID_i^{\mathrm{new}} = r_i'\oplus TID_i$. Otherwise, MD_i will wait for reauthentication.

STEP 6. Getting all Auth_i from MD_i, NG has the ability of batch authentication and access control. After verifying Q_{3i}, NG checks whether *Auth* equals $\mathrm{Auth}_1\oplus\cdots\oplus\mathrm{Auth}_m$. If it does, all m meters are successfully authenticated, and the result $M_{\mathrm{Done}} = \langle \mathrm{TAG}_{\mathrm{Done}}, H(\mathrm{TAG}_{\mathrm{Done}}\|SK_G)\rangle$ is sent to SP. NG also updates its new pseudoidentity as MD does.

STEP 7. SP accepts the authentication result after checking the correctness of M_{Done}. Then, it decrypts X_* in M_3, to get new response r_*', and updates $\langle TID_*^{\mathrm{new}}, r_*'\rangle$ of all devices.

5. Security Evaluation

5.1. Formal Verification by Tamarin. In this section, the powerful automatic verification tool Tamarin is employed to elucidate the strengths of our protocol. Tamarin is an analytical tool for security properties by symbolic modeling. Multilevel rewriting rules are used to model the behavior of participants, depending on which, the protocol is executed through state transition triggered by rules. In this interactive system, messages are output to the DY channel, from which DY attackers can acquire any intermediate states, deduce some values, and interact with participants. Tamarin

Table 2: Definitions of parameters.

Notation	Definition
ID_G, ID_i	True identity of the gateway and meter i
TID_G, TID_i	Pseudoidentity of the gateway and meter i
$\mathrm{PUF}_G, \mathrm{PUF}_i$	Physical unclonable function embedded in the gateway and meter i
c_G, c_i	Challenge for PUF in gateway and meter i
$\hat{r}_G, \hat{r}_i$	Response generated in enrollment
r_G, r_i	Real-time response during authentication
FE.Gen FE.Rec	The helper data generation algorithm and response reproduction algorithm for the reverse fuzzy extractor
E_K, D_K	Encryption and decryption algorithm using the key K
$H(\ )$	Secure hash function
N_G	Nonce generated by gateway
SK_{S-G}, SK_i	Session key between SP-NG, and SP-MD_i
$Auth_i$	Authentication credential for MD_i

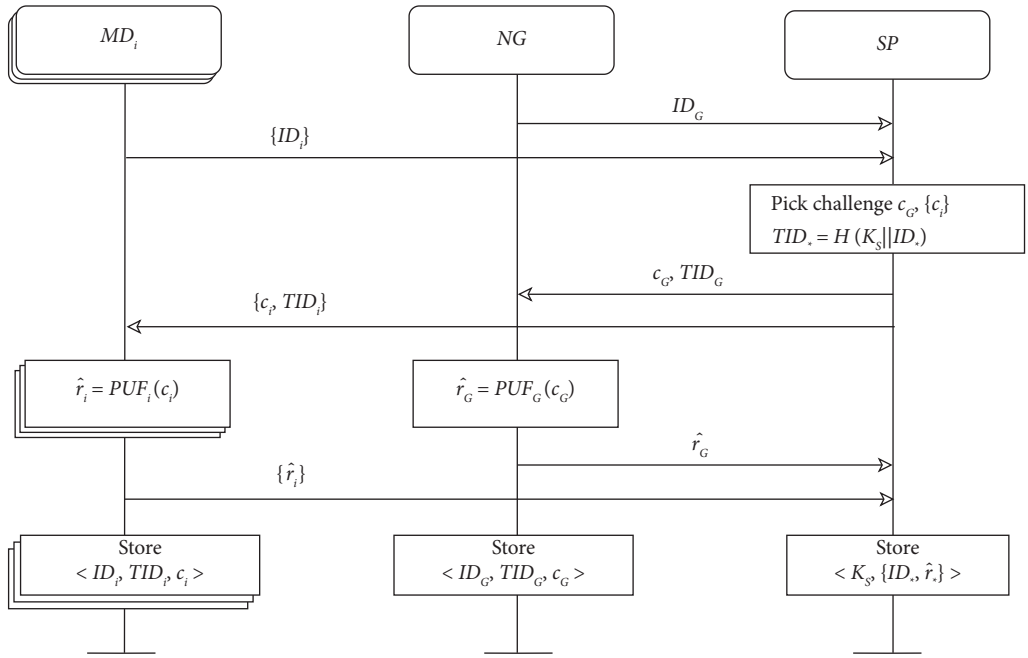

Figure 2: Enrollment phase.

supports numerous cryptographic operations, e.g., *hashing*, *symmetric-encryption signing*. Here in ours, PUF in an authentication phase and during enrollment is modelled as aprivate function PUF_r/1 and PUF_e/1 respectively. These functions also satisfy an abstract equation Rec (PUF_e(c), Gen (PUF_r(c))) = PUF_r(c), which means that once SP obtains the enrolled response and helper data, it can recover the same response as MD or NG does.

Rules are usually attached with action concerning some security properties that are intended to be proved. Secrecy and authentication are the most two significant security properties. If a rule for SP is tagged with an action secret ('SP', key), it means the adversary does not know key, and there is no case that some entities are revealed even if they are honest. As for authentication, it can be specified as four levels, aliveness, weak agreement, noninjective agreement, and injective agreement. Here, we only focus on injective agreement with the most security level. The lemma SP_injective_agreement_On_ SK_NG implies that SP injectively agrees with NG on key if, whenever SP completes a run of the protocol with NG, labelled by action Commit (SP, NG, <'SP', 'NG, key>), there must has been a NG who previously run the protocol with SP, labelled by action Running (NG,SP,<'NG', 'SP', key>), and they both agreed on the same key during the same session. Otherwise, there must be an adversary who has previously performed a session key reveal on either of them.

Figure 4 shows that SK_∗_Secrecy implies that session keys are secret against adversaries. And the rest lemmas indicate that SP and NG, SP and MDA/MDB achieve mutual authentication with the same session key. Our code is available at https://github.com/BUAA-CST/Extremely-Lightweight-SmartGrid.

5.2. Formal Security Proof with the RoR Model. In this section, session key security is proved under the Real-Or-Random model (ROR) [32], which basically contains the

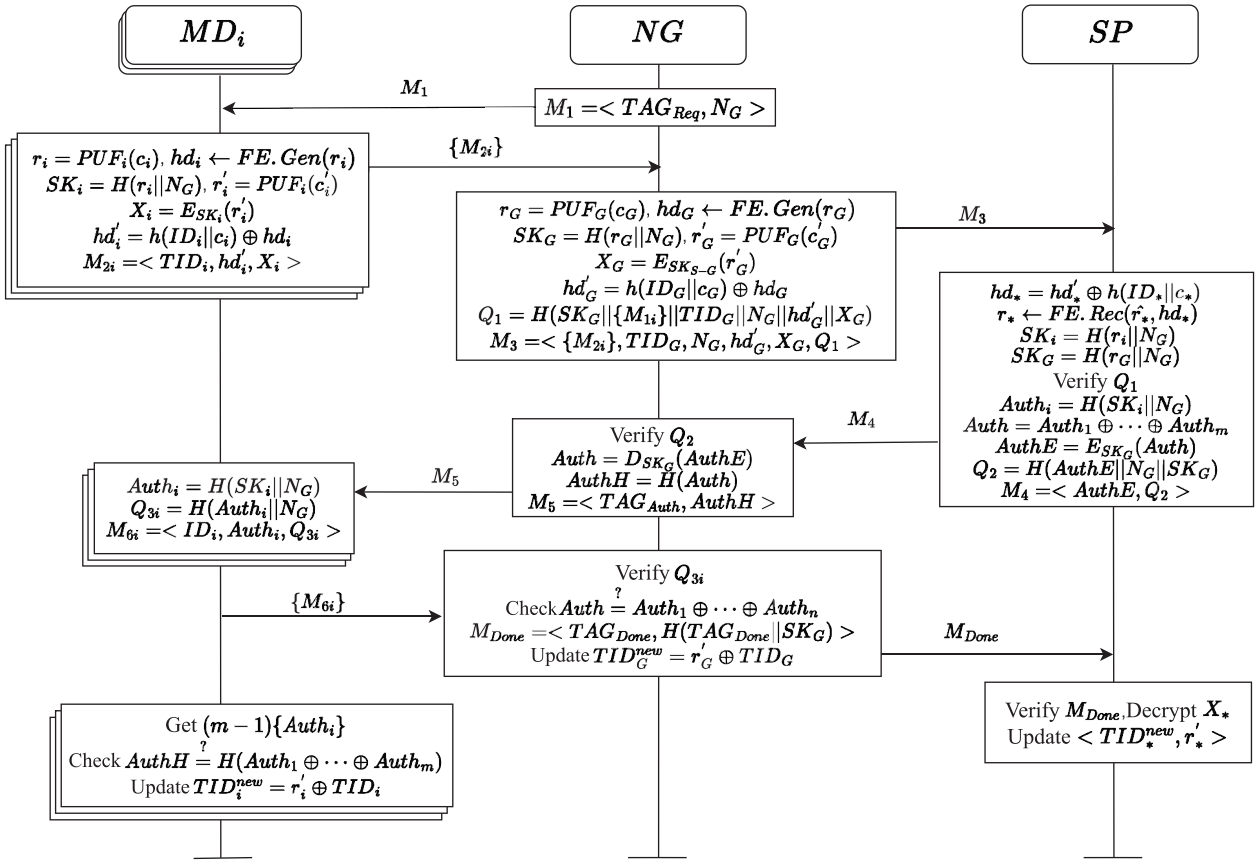

Figure 3: Batch AKA phase with computation offload and NG access control.

```
lemma SP_injective_agreement_On_SK_NG:
"
    All SP NG SK_NG #i. Commit(SP, NG, <'SP',' NG',SKA >) @i
    ==> (Ex #j. Running(NG, SP, < 'NG','SP', SKA >) @j  & j < i
    & not (Ex NG2 SP2 SK #i2. Commit(SP2, MDA2, <'SP','MDA',SKA >)
    @i2 & not(#i2 = #i)))
    | (Ex  device #r. Reveal(device) @r & Honest(device) @i  & r < i)
"
 ==============================================================================
 summary of summaries:
 analyzed: MD-NG-SP.spthy

   trace_exists (exists-trace): verified (16 steps)
   SK_A_Secrecy (all-traces): verified (17 steps)
   SK_B_Secrecy (all-traces): verified (17 steps)
   SK_NG_Secrecy (all-traces): verified (177 steps)
   SP_injective_agreement_On_SK_A (all-traces): verified (1107 steps)
   MDA_injective_agreement_On_SK_A (all-traces): verified (44 steps)
   SP_injective_agreement_On_SK_B (all-traces): verified (1120 steps)
   MDB_injective_agreement_On_SK_B (all-traces): verified (44 steps)
   SP_injective_agreement_On_SK_NG (all-traces): verified (88 steps)
   NG_injective_agreement_On_SK_NG (all-traces): verified (27 steps)
 ==============================================================================
```

Figure 4: Tamarin lemma and verification for the authentication scheme.

concept of participants, adversaries, and queries. Let oracle $\pi_{MD}^{t_1}, \pi_{NG}^{t_2}, \pi_{SP}^{t_3}$ denote the t_k th instance of MD, NG, and SP, respectively. Adversary $\mathcal{A}$ having full-control of the channel can launch passive attack and various active attacks, and deliberately strive to break the protocol through visiting queries *Execute*, *Corrupt*, *Reveal*, *Send,* and *Test*.

(i) $Execute(\pi_{MD}^{t_1}, \pi_{NG}^{t_2}, \pi_{SP}^{t_3})$: this query enables $\mathcal{A}$ to launch passive attacks like eavesdropping.

(ii) $Corrupt(\pi_{MD}^{t_1}, \pi_{NG}^{t_2})$: by this query, $\mathcal{A}$ can get extract sensitive information from meters and gateway.

(iii) $Reveal(\pi^t)$: $\mathcal{A}$ can reveal the current session key.

(iv) $Send(\pi^t, u)$: $\mathcal{A}$ can visits this query to send message u to any entity to launch active attacks.

(v) $Test(\pi^t)$: at the beginning of the game, an unbiased coin b is determined. If $b = 1$, The *Test* query will return the real session key, otherwise it will return a random key. After this query, $\mathcal{A}$ will guess the value of b; that is, $\mathcal{A}$ needs to distinguish real session key from a random key. This query determines the semantic security of the established session key SK.

Theorem 1. *In the ROR model, an adversary $\mathscr{A}$ tries to calculate the session key of the proposed scheme in polynomial time. Hash and PUF function are modeled as random oracle h(•) and PUF(•). Let $Adv_{\mathscr{A}}(t)$ denotes the advantage that $\mathscr{A}$ breaks the semantic security of the session key. We define q_h, q_p, and q_s as the number of Hash, PUF, and Send queries, |Hash| and |PUF| as the range space of h(•) and PUF(•), respectively. C′ and s' are the Zipf's parameters [33].*

Proof. We use four games Game_k to prove it, where S_{Game_k} is the probability that $\mathscr{A}$ wins the game, and $Pr(S_{\text{Game}_k})$ is the winning advantage.

(1) Game_0: in Game_0, $\mathscr{A}$ is given the ability as in real world to guess b, thus we have

$$\text{Adv}_{\mathscr{A}}(t) = \left|2Pr\left[S_{\text{Game}_0}\right] - 1\right|. \tag{1}$$

(2) Game_1: $\mathscr{A}$ can eavesdrop on the channel through an *Execute* query, which returns information propagated among MD, NG, and SP. Then $\mathscr{A}$ visits *Test* and *Reveal* queries to distinguish between the session key and random key. Lacking r_*, $\mathscr{A}$ cannot compute $SK_* = H(H(r_*)\|N_S)$, thus its winning advantage is the same as in Game_0.

$$Pr\left[S_{\text{Game}_1}\right] = Pr\left[S_{\text{Game}_0}\right]. \tag{2}$$

(3) Game_2: compared with Game_1, the ability of $\mathscr{A}$ is extended to perform active attacks through visiting $Send(\pi^t, u)$ and *Hash* queries. Even though $\mathscr{A}$ tries to forge messages that it eavesdrops, these messages in different session with fresh nonce and integrity authentication protection cannot be fabricated without the knowledge of r_*. As a result, the probability for $\mathscr{A}$ to guess the session key is up to the collision-resistance of hash function. According to birthday paradox, we have the following advantage:

$$\left|Pr\left[S_{\text{Game}_2}\right] - Pr\left[S_{\text{Game}_1}\right]\right| \le \frac{q_h^2}{2|\text{Hash}|}. \tag{3}$$

(4) Game_3: in Game_3, $\mathscr{A}$ are given additional access to $Corrupt(\pi_{MD}^{t_1}, \pi_{NG}^{t_2})$ to extract secret information stored in meters and gateway compared to Game_2. Even $\mathscr{A}$ extracts challenge, it still cannot generate correct response to derive the session key due to the nonclonability of PUF. By applying password dictionary attack, $\mathscr{A}$'s advantage rests with the number of $Send(\pi^t, u)$ query and the space range of PUF(•).

$$\left|Pr\left[S_{\text{Game}_3}\right] - Pr\left[S_{\text{Game}_2}\right]\right| \le \frac{q_s}{2^n|PUF|}. \tag{4}$$

Finally, after executing all the oracles and querying the $Test(\pi^t)$ query for only one time, it is clear that the probabilistic of $\mathscr{A}$ to presume the bit b is

$$Pr\left[S_{\text{Game}_3}\right] = \frac{1}{2}. \tag{5}$$

According to equations (1), (2), and (5), we can obtain

$$\begin{aligned}\frac{1}{2}\text{Adv}_{\mathscr{A}}(t) &= \left|Pr\left[S_{\text{Game}_0}\right] - \frac{1}{2}\right| \\ &= \left|Pr\left[S_{\text{Game}_0}\right] - Pr\left[S_{\text{Game}_3}\right]\right| \\ &= \left|Pr\left[S_{\text{Game}_1}\right] - Pr\left[S_{\text{Game}_3}\right]\right|.\end{aligned} \tag{6}$$

Considering (3) and (4), we apply the triangular inequality to conclude that

$$\begin{aligned}&\left|Pr\left[S_{\text{Game}_1}\right] - Pr\left[S_{\text{Game}_3}\right]\right| \le \left|Pr\left[S_{\text{Game}_1}\right] - Pr\left[S_{\text{Game}_2}\right]\right| \\ &+\left|Pr\left[S_{\text{Game}_2}\right] - Pr\left[S_{\text{Game}_3}\right]\right| \le \frac{q_h^2}{2|\text{Hash}|} + \frac{q_s}{2^n|PUF|}.\end{aligned} \tag{7}$$

Finally, by solving (6) and (7), we obtain the required result as follows:

$$\frac{1}{2}\text{Adv}_{\mathscr{A}}(t) \le \frac{q_h^2}{2|\text{Hash}|} + \frac{q_s}{2^n|PUF|}. \tag{8}$$

□

5.3. Informal Security Analysis. In this section, an informal security analysis of the proposed protocol is given to show the security of our protocol against various attacks. A comparison on security properties with others is demonstrated in Table 3.

5.3.1. Mutual Authentication against NG. By using challenge-response mechanism, we achieve mutual authentication between MD and SP and NG and SP. Although NG participates in the AKA procedure, it cannot derive the session key SK_i, thus cannot decrypt the reporting messages. As a result, our protocol supports end-to-end AKA and message confidentiality against curious-but-hones NG (*S*1).

5.3.2. MD Anonymity (S2). Pseudonyms TID_* for MD and NG are initially generated by SP, and is indistinguishable for attackers. During each execution for the batch AKA phase, TID_* is updated and cannot be traced due to the unknowability of the former one and new session key. Since different pseudonyms are used for different authentication rounds, it is infeasible for the attackers to identify the real identity for a specific MD or NG.

5.3.3. Multiple Attacks Resistance. The protocol is able to resist various passive and active attacks.

(1) Physical attack resistance (S3). The root of trust is established on PUF due to its uniqueness and the nonclonability. Secret response is only generated during enrollment, and the interface is closed ever since. Therefore, an attacker cannot extract response from the memory.

Table 3: Comparisons of properties on security.

Security	[12]	[13]	[14]	[15]	[24]	[20]	[27]	Ours
$S1$	—	—	—	—	✗	—	✓	✓
$S2$	✗	✓	✓	✓	✓	✗	✗	✓
$S3$	✗	✓	✓	✗	✗	✓	✗	✓
$S4$	✗	✓	✓	✗	✓	✓	✓	✓
$S5$	✗	✓	✗	✗	✗	✗	✗	✓
$S6$	✗	✓	✓	✗	✗	✗	✗	✓

$S1$: curious-but-honest intermediate entity; $S2$: anonymity; $S3$: resist physical attack; $S4$: resist replay attack; $S5$: resist DoS attack; $S6$: resist desynchronization attack.

Similarly, an attacker cannot clone a MD/NG due to the nonclonability of PUF. The protocols of Xia [12], Tsai [15], Badar [24] and Uludag [27] do not offer prevention against the physical attack.

(2) Impersonation attack/MitM attack resistance. Adding integrity checking, an adversary is forbidden from effectuating impersonation attacks no matter whom it masquerades as. If it impersonates a SP, it cannot generate a valid Q_2 that contains correct SK_G, so M_4 will be neglected by NG. If it impersonates a NG deceiving SP, its message M_{Done} will not be authenticated due to the incorrectness encryption. If it aims to deceive MD_is, MD_is will be aware of in STEP 5. If it impersonates a MD by sending false $Auth_i$, NG will easily find it through batch authentication. Furthermore, it is not sufficient for an attacker to impersonate as a MD, even if it acquires $AuthH$ and m-1 $Auth_i$s.

(3) Replay attack resistance (S4). The fresh random number N_G is embedded in $Auth_*$ to prevent credential replaying. Also, the one-time identifier TID_i is updated per round authentication. The replayed message with old nonce cannot be verified in the new session.

(4) DoS attack resistance (S5). Contrary to the scheme [12, 14, 15, 20, 24, 27], NG in our protocol can figure out the malicious MD after batch authentication, then it will filter out these MDs, which alleviates the DoS attack to SP.

(5) Desynchronization attack resistance (S6). For all entities, TID_*s is updated only after a successful verification. If some steps fail ahead, they will unable to verify these messages successfully; hence, TID_i is always synchronous. While schemes [12, 15, 20, 24, 27] cannot resist the desynchronization attack.

6. Performance Analysis

In this section, we discuss the performance and efficiency of our proposed AKA scheme in terms of computation, communication, storage overhead, and signaling cost. To be fair, only PUF-based authentication schemes, i.e., Gope and Sikdar [13], Tanveer et al. [14], Badar et al. [24], and Kaveh and Mosavi [20] are introduced for comparison.

6.1. Computation Cost. To evaluate the computation cost, we build an in-house prototype system for the smart grid, as shown in Figure 5. Our implementation of SRAM PUF employs 8 bits challenge for 64 bits response. Different platforms are chosen to simulate the entities with different computational capacities. SP is simulated using a computer equipped with Intel Core i9-10885H CPU @ 2.40 GHz processer. NG is an embedded platform equipped with Quad core Cortex-A72 (ARM v8) 64 bit SoC @ 1.5 GHz and NB-IoT module. We choose the extremely resource-constrained MSP430FR5969 microcontroller to perform the operations of MD.

We use the notations $T_{E/D}$, T_H, T_{PUF}, $T_{\text{GEN/REC}}$, T_{ECC} to denote the time needed for basic operations such as the symmetric encryption/decryption, hash function, PUF operation, key generation/reproduction algorithm of the fuzzy extractor, and point multiplication. Without loss of generality, we initialize the symmetric encryption as AES-128, the fuzzy extractor algorithm with (31, 16) BCH code. The hash function is instantiated as HWAES-CMAC on the fed message of average of 16 bytes, which is benefited from the hardware accelerator for MSP430FR5969. Furthermore, PUF is extracted from the 2 KB SRAM memory embedded in. All computation overhead for MD is measured in terms of CPU clock cycles with frequency of 8 MHz. Then we compile and run *C*/*C*++ programs with the MIRACL library and test the computation cost of one single cryptography operation employed in these platforms. To improve the accuracy, we run 100 times for each cryptographic operation to get the average execution time. T_{PUF}, $T_{E/D}$, T_H, T_{GEN}, and T_{REC} takes 180, 670, 2,096, 71,303, and 263,135 cycles, respectively, which is equivalent to 22.5, 83.75, 262, 8,912.8, and 32,891.8 us. The experimental data verify the efficiency of the reverse fuzzy extractor again. Concrete execution time for the above cryptographic operations in three different platforms is illustrated in Table 4.

We can observe an overall computation overhead comparison with related schemes in Table 5 and Figure 6(a), where our scheme has a great advantage over others. As can be seen, the [13, 14, 20] present an end-to-end structure only between two entities. Scheme [24] is the only one that features an *end-edge-cloud* structure but with fully trusted NG. And the lack of security properties described in Table 3 makes them uncompetitive. Scheme [14] introduces heavy point multiplication at the MD side. Since our resource-constrained microcontroller cannot provide point multiplication with a competitive security level, the execution time T_{ECC} at the MD side is quoted from the original paper. To be fair, we add T_{REC} to [20] to provide equal security. We also compare our scheme with forward fuzzy extractor version. The replacement from FE.Rec to FE.Gen makes our scheme more computational efficiency.

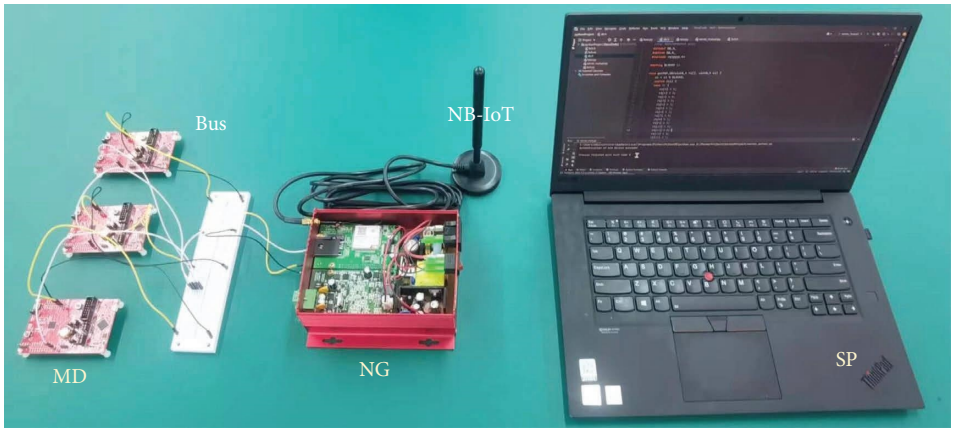

Figure 5: In-house built the experimental setup.

Table 4: Computation cost of cryptographic operations.

Operations	Execution time (us)		
	MD	NG	SP
$T_{E/D}$	83.75	16.9	1.8
T_H	262	25.1	1.1
T_{PUF}	22.5	0.5	—
T_{GEN}	8,912.8	1,968	6.7
T_{REC}	32,891.8	8,806	24.8
T_{ECC}	—	3,297.5	359.6

6.2. Communication Cost. We compare the communication cost during AKA phase with the recent schemes in Table 6. In our protocol, both real and pseudo identities are 128 bits, which is the same for random number/nonce and hash value since the usage of HWAES-CMAC. In other schemes, the use of one-by-one authentication makes SP need to send $O(m)$ message for m MDs. This brings a huge burden on expensive NB-IoT downlink transmission. While in ours, SP only has to send one message $M_3 = \langle \text{Auth}E, Q_2 \rangle$, which is $128 * 2 = 256$ bits. Besides, MD_i sends $M_{2i} = \langle TID_i, hd_i, X_{1i} \rangle$ and $M_{6i} = \langle TID_i, \text{Auth}_i, Q_{3i} \rangle$ which costs $128 + 128 + 128 = 384$ bits and $128 + 128 + 128 = 384$ bits, and that amounts to 768 m bits for m MDs. NG sends messages $M_1 = \langle \text{TAG}_{\text{Req}}, N_G \rangle$, $M_5 = \langle \text{TAG}_{\text{Auth}}, \text{Auth}H \rangle$ to MD, $M_3 = \langle \{M_{2i}\}, TID_G, N_G, hd_G, X_2, Q_1 \rangle$, $M_{\text{Done}} = \langle \text{TAG}_{\text{Done}}, H(\text{TAG}_{\text{Done}} \| SK_{S-G}) \rangle$ to SP. These messages costs $4 + 64 = 68$ bits, $4 + 128 = 132$ bits, $384m + 576$ bits and $4 + 128 = 132$ bits, which amount to $384m + 908$ bits. As a result, the total communication cost of the proposed scheme is $1512m + 1164$ bits. The batch authentication mechanism optimizes the NB-IoT downlink transmission, and makes the communication cost of our scheme far more less than that of others. The comparison of total communication cost versus MD density is shown in Figure 6(b).

6.3. Signaling Cost. Table 7 shows that we evaluate our scheme by comparing with others in terms of the number of signaling messages. To be fair, we add NG as a transmitter in those schemes that only evolve MD and SP. The signaling cost of ours mainly comes from MDs, since m MDs need to forward m messages containing different hd_i, and m messages that contain different credential $Auth_i$. Due to the batch mechanism, although SP targets at building session keys with m MDs, it only needs to send one downlink message to NG to devolve authentication ability, which greatly reduces signaling cost. More MDs can be accessed to the smart grid without increasing the number of messages sent from SP. As a transmitter between MD and SP, NG has to forward at least 4 messages. From Figure 6(c), the signaling cost of ours is much less than those of others when the number of MDs is increasing.

6.4. Storage Burden. We compare the storage burden on MD, NG, and SP, respectively, with those using PUF in Table 8. Here, n denotes the number of CRPs that the entity stores, and helper data hd is set to be 128 bits. We also compare the storage overhead at the MD side when $n = 128$ in Figure 7. We assume that there is only one NG under an SP. Protocol [13] has the largest storage overhead since MD stores all the challenge and helper data. Despite that [14] is superior to us at the SP side, the usage of elliptic curve Diffie–Hellman key exchange makes it underperforming in computation efficiency. As the only protocol featuring end-edge-cloud structure, [24] presets symmetric keys among three entities, which is not as secure as ours in physical attack resistance. Scheme [20] has the lowest storage overhead at the MD side, since it only stores true identities, which is not privacy preserving. In our protocol, NG stores $\langle ID_G, TID_G, C_G \rangle$, MD stores $\langle ID_i, TID_i, C_i \rangle$, which both

Table 5: Comparisons of computation cost.

Protocol	MD side (ms)	NG side (ms)	SP side (ms)	Total (ms)
[13]	$(4T_H + T_{\text{REC}} + 2T_{PUF})m \approx 33.9m$	—	$(5T_H + T_{\text{GEN}})m \approx 0.01m$	$33.91\,m$
[14]	$(4T_H + 2T_E + 2T_{ECC} + T_{\text{REC}} + T_{PUF})m \approx 39.1m$	—	$(4T_H + 3T_E + T_{ECC})m \approx 0.37m$	$39.47\,m$
[24]	$(3T_H + T_{PUF} + T_{\text{REC}})m \approx 33.7m$	$3mT_H \approx 0.08m$	$6mT_H \approx 0.007m$	$33.79\,m$
[20]	$(2T_H + 2T_{PUF} + T_{\text{REC}})m \approx 33.4m$	$3mT_H \approx 0.08m$	—	$33.48\,m$
Forward FE	$(4T_H + 2T_{PUF} + T_{\text{REC}} + T_E)m \approx 34.1m$	$(m+5)T_H + 2T_{PUF} + T_{\text{REC}} + 2T_{E/D} \approx 0.02m + 8.9$	$(2m+4)T_H + (m+2)T_{E/D} + (m+1)T_{\text{GEN}} \approx 0.01m$	$34.13\,m + 8.9$
Ours	$(4T_H + 2T_{PUF} + T_{\text{GEN}} + T_E)m \approx 10.1m$	$(m+5)T_H + 2T_{PUF} + T_{\text{GEN}} + 2T_{E/D} \approx 0.02m + 2.1$	$(2m+4)T_H + (m+2)T_{E/D} + (m+1)T_{\text{REC}} \approx 0.03m$	$19.05\,m + 2.1$

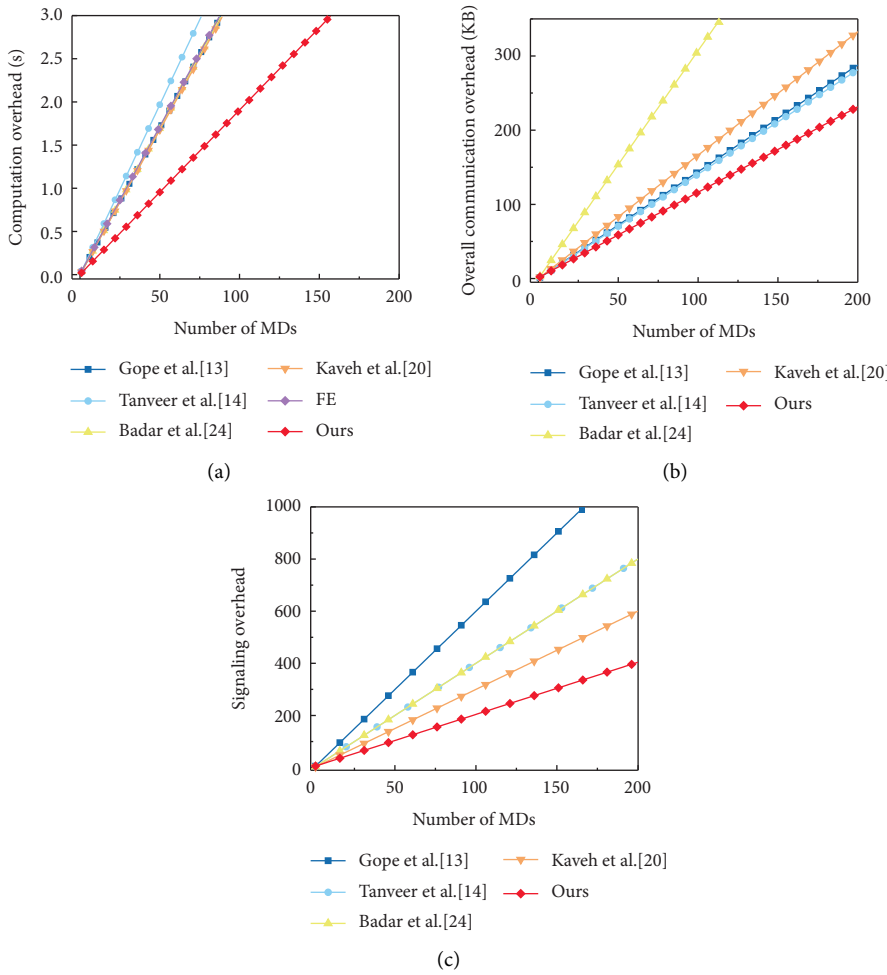

Figure 6: Comparison of (a) computation, (b) communication, and (c) signaling cost versus MD density.

Table 6: Comparisons of communication cost.

Protocol	MD side (m)	NG side	SP side	Total
[13]	704	—	736 m	1440 m
[14]	864	—	544 m	1408 m
[24]	576	1664 m	832m	3072 m
[20]	896	768 m	—	1664 m
Ours	768	384 m + 908	256	1152 m + 1164

Table 7: Comparisons of signaling burden.

Protocol	MD side	NG side	SP side	Total
[13]	m	3m	2m	6m
[14]	m	2m	m	4m
[24]	m	2m	m	4m
[20]	2m	m	—	3m
Ours	2m	4	1	2m + 5

Table 8: Comparisons of storage burden.

Protocol	MD side	NG side	SP side
[13]	160n + 160	—	(208n + 208)m
[14]	544	—	128 m + 704
[24]	480	320	560 m + 320
[20]	160	(128 + (256 + 64)n)m	—
Ours	264	264	256 m + 384

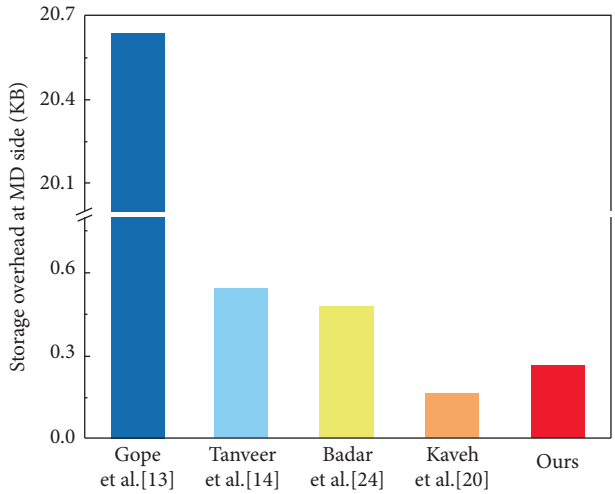

FIGURE 7: Comparison of storage overhead at MD side when $n = 128$.

need $128 + 128 + 8 = 264$ bits. SP only stores $\langle K_S, \{ID_*, r^*\}\rangle$, that needs $128 + (128 + 128)\ (m+1) = 256\,m + 384$ bits of storage space.

7. Conclusion

In this paper, we propose a PUF-based batch AKA protocol to protect all devices in the open environment for an end-edge-cloud smart grid connected by Bus-NB-IoT hierarchical network. Using intrinsic SRAM PUF, both MD and NG are protected from physical attack. Receiving the aggregated credential from SP, NG is allowed to authenticate a batch of MDs, then is endowed with the access control authority. More importantly, we provide a mutual end-to-end AKA only between MD and SP, which is secure against honest-but-curious NG. Our scheme is proved by Tamarin-based and formal security verification. Through informal analysis, we show that our protocol satisfies rich security properties and is capable to resist various attacks. The analysis of performance among others proves that our solution is outstanding in computation, and is competitive in terms of communication, storage, and signaling overhead.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (U21B2021, 32071755, 62002006), Henan Key Laboratory of Network Cryptography Technology (LNCT2021-A05), CCF-NSFOCUS Kun-Peng Scientific Research Fund (CCF-NSFOCUS 2021011), and the Defense Industrial Technology Development Program (JCKY2021211B017).

References

[1] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: motivations, requirements and challenges," *IEEE communications surveys & tutorials*, vol. 15, no. 1, pp. 5–20, 2013.

[2] A. Ghosal and M. Conti, "Key management systems for smart grid advanced metering infrastructure: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2831–2848, 2019.

[3] 3Gpp Ts 36.300, "Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); overall description; stage 2," 3gpp, France, R. G. P. Project, 2020.

[4] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: narrowband internet of things (NB-IoT)," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1505–1515, 2018.

[5] S. K. Routray, D. Gopal, A. Javali, and A. Sahoo, "Narrowband IoT (NBIoT) assisted smart grids," in *Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems*, pp. 1454–1458, Coimbatore, India, March 2021.

[6] X. Ren, J. Cao, M. Ma, H. Li, and Y. Zhang, "A novel PUF-based group authentication and data transmission scheme for NB-IoT in 3GPP 5G networks," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3642–3656, 2022.

[7] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security: a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[8] A. Triantafyllou, J. A. P. Jimenez, A. D. R. Torres, T. Lagkas, K. Rantos, and P. Sarigiannidis, "The challenges of privacy and access control as key perspectives for the future electric smart grid," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 1934–1960, 2020.

[9] S. Nie, L. Liu, and Y. Du, "Free-fall: hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[10] C. Parian, T. Guldimann, and S. Bhatia, "Fooling the master: exploiting weaknesses in the modbus protocol," *Procedia Computer Science*, vol. 171, pp. 2453–2458, 2020.

[11] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the Modbus protocols," *International Journal of Critical Infrastructure Protection*, vol. 1, pp. 37–44, 2008.

[12] J. Xia and Y. Wang, "Secure key distribution for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1437–1443, 2012.

[13] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2019.

[14] M. Tanveer, A. U. Khan, N. Kumar, A. Naushad, and S. A. Chaudhry, "A robust access control protocol for the smart grid systems," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6855–6865, 2022.

[15] J. L. Tsai and N. W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 1–914, 2015.

[16] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1–1910, 2016.

[17] H. Boyapally, P. Mathew, S. Patranabis et al., "Safe is the new smart: PUF-based authentication for load modification-resistant smart meters," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, pp. 663–680, 2022.

[18] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2830–2838, 2019.

[19] K. Mahmood, X. Li, S. A. Chaudhry et al., "Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure," *Future Generation Computer Systems*, vol. 88, pp. 491–500, 2018/11/01/2018.

[20] M. Kaveh and M. R. Mosavi, "A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4535–4544, 2020.

[21] D. Abbasinezhad-Mood and M. Nikooghadam, "An ultra-lightweight and secure scheme for communications of smart meters and neighborhood gateways by utilization of an ARM cortex-M microcontroller," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6194–6205, 2018.

[22] S. Aghapour, M. Kaveh, M. R. Mosavi, and D. Martin, "An ultra-lightweight mutual authentication scheme for smart grid two-way communications," *IEEE Access*, vol. 9, Article ID 74562, 2021.

[23] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3548–3557, 2020.

[24] H. M. S. Badar, S. Qadri, S. Shamshad, M. F. Ayub, K. Mahmood, and N. Kumar, "An identity based authentication protocol for smart grid environment using physical uncloneable function," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4426–4434, 2021.

[25] R. De Smet, T. Vandervelden, K. Steenhaut, and A. Braeken, "Lightweight PUF based authentication scheme for fog architecture," *Wireless Networks*, vol. 27, no. 2, pp. 947–959, 2021/02/01 2021.

[26] H. Wang, J. Meng, X. Du, T. Cao, and Y. Xie, "Lightweight and anonymous mutual authentication protocol for edge IoT nodes with physical unclonable function," *Security and Communication Networks*, vol. 2022, Article ID 1203691, 11 pages, 2022.

[27] S. Uludag, K. S. Lui, W. Ren, and K. Nahrstedt, "Secure and scalable data collection with time minimization in the smart grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 43–54, 2016.

[28] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.

[29] M. Wazid, A. K. Das, S. Shetty, J. J P C Rodrigues, and Y. J. S. Park, "LDAKM-EIoT: lightweight device authentication and key management mechanism for edge-based IoT deployment," *Sensors*, vol. 19, no. 24, p. 5539, 2019.

[30] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2014.

[31] L. Wu, J. Wang, S. Zeadally, and D. He, "Anonymous and efficient message authentication scheme for smart grid," *Security and Communication Networks*, vol. 2019, Article ID 4836016, 12 pages, 2019.

[32] M. Abdalla, P.-A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *International Workshop on Public Key Cryptography*, pp. 65–84, Springer, Singapore, 2005.

[33] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.