future internet

MDPI

Article

Smart Grid Security: A PUF-Based Authentication and Key Agreement Protocol

Nasour Bagheri [1,†], Ygal Bendavid [2,*,†], Masoumeh Safkhani [3,†] and Samad Rostampour [4,†]

[1] Electrical Engineering Department, Shahid Rajaee Teacher Training University (SRTTU), Tehran 16788-15811, Iran; nbagheri@sru.ac.ir
[2] AOTI Department, School of Management, Université du Québec à Montréal (UQAM), Montreal, QC H2X 1L7, Canada
[3] Computer Engineering Department, Shahid Rajaee Teacher Training University (SRTTU), Tehran 16788-15811, Iran; safkhani@sru.ac.ir
[4] Computer Science Department, Vanier College, Montreal, QC H4L3X9, Canada; rostamps@vaniercollege.qc.ca
* Correspondence: bendavid.ygal@uqam.ca
† These authors contributed equally to this work.

Abstract: A smart grid is an electricity network that uses advanced technologies to facilitate the exchange of information and electricity between utility companies and customers. Although most of the technologies involved in such grids have reached maturity, smart meters—as connected devices—introduce new security challenges. To overcome this significant obstacle to grid modernization, safeguarding privacy has emerged as a paramount concern. In this paper, we begin by evaluating the security levels of recently proposed authentication methods for smart meters. Subsequently, we introduce an enhanced protocol named PPSG, designed for smart grids, which incorporates physical unclonable functions (PUF) and an elliptic curve cryptography (ECC) module to address the vulnerabilities identified in previous approaches. Our security analysis, utilizing a real-or-random (RoR) model, demonstrates that PPSG effectively mitigates the weaknesses found in prior methods. To assess the practicality of PPSG, we conduct simulations using an Arduino UNO board, measuring computation, communication, and energy costs. Our results, including a processing time of 153 ms, a communication cost of 1376 bits, and an energy consumption of 13.468 mJ, align with the requirements of resource-constrained devices within smart grids.

Keywords: smart grid; smart meter; authentication; Internet of Things; security; elliptic curve cryptography

Citation: Bagheri, N.; Bendavid, Y.; Safkhani, M.; Rostampour, S. Smart Grid Security: A PUF-Based Authentication and Key Agreement Protocol. *Future Internet* **2024**, *16*, 9. https://doi.org/10.3390/fi16010009

Academic Editor: Paolo Bellavista

Received: 8 November 2023
Revised: 17 December 2023
Accepted: 26 December 2023
Published: 28 December 2023

1. Introduction

As COP28 concluded in the UAE to accelerate climate action and gradually transition to a decarbonized energy system, the evolution and modernization of existing grids with new technologies are positioned as key enablers to this increasingly urgent transition [1]. Aging power infrastructures are being modernized to meet growing demand for electricity and efficiently distribute both traditional and renewable energy while also meeting the environmental imperative to reduce greenhouse gas emissions and ensure sustainable growth. In this context, along with other low-less carbon technologies (e.g., nuclear power, thermal and hydro-energy, solar photovoltaic power, and wind energy), utilities are responding to this challenge by investing in smart grids (SG) to ensure a safe and reliable supply of electricity.

Smart grids are electricity networks that use digital technologies, sensors, and software to better match the supply and demand of electricity in real time while minimizing costs and maintaining the stability and reliability of the grid [2]. This network integrates advanced metering devices, information and communication technologies (ICT), demand response mechanisms, and real-time control systems. This shift towards smart grids is part of a broader trend observed in the electricity sector, integrating with remote sensing, cloud

computing, and the Internet of Things (IoT). Within this context, IoT technologies are instrumental in transforming the sector from a centralized to a distributed, smart, and integrated energy system [3]. Indeed, in the age of digital transformation, IoT-enabled smart grids empower utilities to sense, analyze, control, and optimize their grids and explore new revenue streams by building innovative business models.

Among the components shaping this transformation in the electricity sector, grid sensors such as smart meters (SMs) not only monitor and transmit electricity consumption data to utility offices but also eliminate the need for manual meter readings. Indeed, SMs, as connected devices, automatically communicate with gateways such as neighborhood area networks (NANs) or other base stations using various communication technologies, such as LoraWAN, cellular, or satellite. SMs capture the date and time of electricity consumption, allowing the implementation of effective energy management systems. For instance, dynamic pricing strategies, where utilities charge higher rates during peak demand periods, enable consumers and businesses to manage their energy consumption remotely, shifting it to off-peak hours when costs are lower. Additionally, smart meters enhance utility services by detecting tampering and theft, offering faster responses to meter failures and power outages. The interest in efficient and accurate metering solutions is reflected by the smart meters market size (gas, water, and electricity), which surpassed USD 22 billion in 2021 and is expected to grow at a CAGR of 13 percent between 2022 and 2028. [4]. Electricity metering has the largest share of this market, with over 60 percent of the global market value [5].

While SMs play a vital role in efficient energy management, their intricate design introduces new security challenges. The ability of smart meters to communicate with nearby IoT devices raises concerns about grid resilience in the face of potential disruptions leading to increased operational costs loss of productivity and loss of sales as well as [6] user security and privacy. Hackers could potentially access personal data and exploit the system for financial gains. Consequently, ensuring privacy has become a top priority in the realm of smart metering.

From a professional perspective, one recent benchmark study from IANS and Artico cited in VentureBeat found that utilities spent an average of 8% of their IT budgets on cybersecurity in 2022, highlighting the importance of this critical domain [7].

From an academic perspective, in recent years, extensive research has been conducted to secure the infrastructure of smart grids. In a review of cyber attacks and defense mechanisms for improving security in smart grid energy systems, Ghiasi et al. point out the expanding use of multiple sensors, controllers, meters, and wireless networks to control and transmit data, suggesting that the issues caused by cyber attacks on these heterogeneous types of infrastructures should receive more attention. They also call for researchers to keep pace with different methods of detecting cyber attacks and develop up-to-date countermeasures—among which they suggest updating or creating new protocols to prevent the access of attackers to the grid [8]. Similarly, Kamrul Hasan et al. conducted a review on cyber-physical and cyber-security systems in smart grid. The authors focus on the increased complexity of managing the security aspects of the grids due to the challenging combination of communication technology, standards, protocols, and applications. To reduce security threats and increase the system's reliability, they also propose to look at security requirements from a defense life cycle (pre-attack, under attack, and post-attack) where, at each phase, different techniques can be used [9].

To address these concerns, the community of researchers proposes various cyber-security protocols. However, many existing designs suffer from security vulnerabilities. In this paper, we propose a new protocol for smart metering systems to address issues found in existing protocols and enhance the efficiency of smart grids.

1.1. Our Contributions

This paper has two main contributions, as follows:

1. We contribute to the literature on smart meter security by proposing a new protocol based on PUFs and ECC for smart grids named PPSG.
2. We provide an in-depth security analysis (with a real-or-random model) for the proposed protocol and also the communication and communication overheads analysis (with an Arduino UNO R3 board) to show that it is among the lightest protocols, compared to the recent related proposals.

1.2. Paper Organization

In the rest of the paper, first, we analyze the existing protocols in Section 2; in Section 3, the required preliminaries are provided. Next, we propose PPSG as a secure protocol in Section 4. The security evaluation of PPSG is detailed in Section 5, and the comprehensive cost analysis is provided in Section 6. The concluding remarks and summary of the paper can be found in Section 7.

2. Related Work

In this section, some recent protocols are described. In Table 1, every protocol is evaluated considering communication overhead, time-consuming processes, encryption techniques, and vulnerability. The assessment is represented by triangles (▼ for low and ▲ for high). A check mark (✓) indicates a documented successful attack, whereas a multiplication symbol (×) signifies an attack not yet published. Interest in the subject is reflected in the number of publications, which have increased exponentially over the last decade, particularly in the field of cyber security, which remains a major concern [10]. Recently substantial efforts have been directed towards establishing a reliable and secure communication infrastructure for smart metering. For example, Kumar et al. introduced the LAKA system, a lightweight authentication and key agreement scheme, aimed at ensuring an acceptable level of security and integrity in smart energy networks [11]. To ensure the confidentiality of messages, LAKA incorporates both hash functions and an ECC module. Moreover, it employs a message authentication code (MAC) function to maintain message integrity. This integration of multiple functions contributes to the heightened complexity of the smart meter (SM). Furthermore, as observed by Baghestani et al. [12], this protocol exhibits susceptibility to traceability attacks.

Kumar et al. introduced ECCAuth, a recent authentication protocol for smart grid applications that relies on ECC cryptography [13]. According to the authors, ECCAuth's primary objective is to establish a secure connection between a smart grid (SG) device and a utility center (UC), ensuring user privacy and the confidentiality of data. Nevertheless, ECCAuth, while maintaining acceptable communication costs, exhibits time-consuming authentication procedures and computational operations. Furthermore, ECCAuth's vulnerability has been exposed by Yu et al., who identified security weaknesses, including session key disclosure, stolen devices, and masquerade, attacks [14]. In response, they proposed a new lightweight protocol that incorporates XOR and hash functions to rectify these shortcomings. Wu et al. also suggested an ECC-based authentication protocol [15]. Although it secures message confidentiality through an ECC module and data encryption, the protocol's functionality falls short of an acceptable level. Notably, it exhibits high communication and computation overhead, leading to extended authentication processing times. Garg et al. recently introduced another authentication protocol for SMs based on the ECC method [16]. Despite their claim of robustness against diverse attacks and reasonable computational expenses, our assessment highlights susceptibility to traceability and impersonation attacks. Additionally, the authors inaccurately calculated and underestimated computational costs.

By utilizing the ECC technique, He et al. [17], Abbasinezhad-Mood [18], and PALK [19] also proposed some protocols, aiming to establish a secure infrastructure. However, they face efficiency challenges, which will be discussed in detail in Section 6.

Tanveer et al. introduced an innovative access control protocol for smart grids, known as RACP-SG [20]. This protocol utilizes lightweight-cryptography-based authenticated encryption with associative data (AEAD) techniques, hash functions, and elliptic curve cryptography (ECC) to successfully execute the authentication process. Moreover, RACP-SG enables mutual authentication between a service provider and an SM, allowing them to establish a session key during communication over the public channel.

Chaudhry et al. recently introduced a new ECC-based protocol called LAS-SG, emphasizing its lightweight nature to ensure satisfactory security and privacy levels [21]. Their approach involves optimizing communication by utilizing only two transferred messages, containing 192 bytes. Although the computation cost of LAS-SG is deemed acceptable, it remains higher than the protocol proposed in this paper.

PUF-based techniques present an intriguing approach utilized in smart metering and grid applications [22]. Numerous PUF-based protocols have been developed for SG/SM, such as the Gope and Sikdar scheme—a key agreement method with privacy-aware authentication protocol to improve security in these kinds of applications [23]. Recognizing the potential impact of cyber attacks on electrical networks, such as real-time decision-making in demand and supply management as well as data manipulation, they created a method to boost the confidentiality of communication channels between UCs and SMs, guaranteeing physical security. Nevertheless, Baeken et al. discovered weaknesses in this approach, suggesting that it does not fulfill all the necessary security criteria [24]. Moreover, the Gope–Sikdar protocol relies on a hash function as its main security measure, rendering it susceptible to key compromise impersonation attacks due to its symmetric nature. Additionally, their protocol includes XORing a temporal value with the secret key before transmitting it on the public channel (i.e., $np* = n_p \oplus K$), where K becomes the main source of authentication afterward This vulnerability exposes the protocol to known session-specific temporary information attacks, allowing various malicious actions, including impersonation and de-synchronization. Rostampour et al. introduced an authentication protocol, EPSG, for the smart grid in the IoT infrastructure [25]. This protocol, combining PUF functions and ECC encryption, establishes a secure environment, ensuring message confidentiality and integrity simultaneously. The authors conducted simulations of EPSG on an Arduino, assessing communication cost and energy consumption in a practical setting. Although EPSG's performance is acceptable, its ability to resist machine learning attacks is limited.

Mustapa et al. introduced a security scheme based on a ring oscillator physically unclonable function [26] to enhance information security in advanced metering infrastructures. Their primary goal was to create a robust and secure authentication approach for smart grid infrastructure. In the architecture named ROPUF, they created a secure connection between the utility center and the smart meter (SM) to transfer data. However, they did not provide details about the workings of this channel and how the SM utilizes it for communication. This channel imposes an additional burden on the SM, generating ciphertext, which was not factored into the protocol's computational cost. Moreover, the proposed protocol exhibited vulnerabilities to impersonation and tracing attacks when data exchanges occurred over a public channel. Furthermore, due to the absence of cryptographic primitives, the protocol was susceptible to advanced attacks, including insider attacks.

In a recent development, Harishma et al. presented a scheme to secure key exchange mutually [27]. Their scheme utilized advanced encryption techniques, such as identity-based encryption (IBE), SHA-2, and the advanced encryption standard (AES), with the possibility of employing ECC encryption and physically unclonable function (PUF) functions. The authors implemented and tested this scheme in a practical environment to provide experimental results. However, the scheme's use of identity-based encryption (IBE) for credential management raised concerns. Currently, the most efficient IBE schemes rely on bilinear pairings on elliptic curves, such as Weil or Tate pairings, while previously published non-pairing-based schemes tend to be inefficient in encryption, decryption, key generation, ciphertext size, or key size [28,29].

Considering that the scheme is designed for resource-constrained devices and involves various encryption methods in each authentication process, it exhibits high complexity and is time-consuming. As an illustration, the authenticated key-exchange protocol on the smart meter setup takes 525 ms for the meter and 360 ms for the server. Additionally, as highlighted in a study by Lounis [30], the protocol is vulnerable to spoofing attacks, where an attacker can impersonate the server and deceive the meter, compromising both the authentication and key-establishment claims of the protocol. Furthermore, the meter does not contribute to the protocol's freshness during the authentication phase, potentially allowing impersonation of the server using the GUMAP attack [31].

LAKE-BSG, a lightweight key exchange scheme empowered by blockchain, was devised by Badshah et al. specifically for smart grids [32]. Leveraging the inherent security of a blockchain system, the authors aimed to establish a secure authentication method for smart meters (SM) while safeguarding user privacy. The proposed technique boasts comparable transmission and computation costs to existing authentication protocols. Furthermore, the integration of blockchain technology is asserted to enhance security by ensuring data storage in a secure, decentralized, and immutable ledger.

Table 1. Comparison of related work.

Reference	Communication Cost	Time-Consuming	Method	Approved Attack
[11]	▲	▲	ECC	✓
[13]	▼	▲	ECC + MAC	✓
[15]	▲	▲	ECC	✓
[16]	▼	▲	ECC	✓
[17]	▲	▼	ECC	✓
[18]	▼	▼	ECC	✓
[19]	▲	▲	ECC	✓
[20]	▼	▼	ECC + AEAD	✓
[21]	▲	▼	ECC	×
[23]	▼	▼	PUF	✓
[25]	▼	▼	PUF	×
[26]	▲	▼	PUF	×
[27]	▼	▲	AES + IBE(ECC) + PUF	✓
[32]	▲	▼	Blockchain	×

3. System Model

The infrastructure of a smart metering system is illustrated in Figure 1. As shown, an SM, a NAN, and a certificate authority (CA) server are the key components, and the communication between the CA and other parties is established via a secure channel. On the other hand, the SM and the NAN are connected over a public channel, which can be the weak point of this structure. Through this paper, we use the list of notations listed in Table 2.

The proposed protocol adopts Canetti and Krawczyk's adversary model (CK-adversary model) [33], which is more robust than the commonly used Dolev–Yao (DY) adversary model [34] in many designs. In the DY-adversary model, the adversary possesses complete control over message transmission through a public channel. It can eavesdrop, delete, insert, or modify fake messages in different instances. Under the CK-adversary model, the adversary possesses all the powers of the DY model and more, enabling them to infiltrate session states and secret information, encompassing secret keys. Consequently, if these session states and secret details are exposed during a particular session, this revelation must not jeopardize the confidentiality of other involved parties, as emphasized in [35]. The CK-adversary model proves advantageous over the DY model, especially in contexts where forward secrecy is a vital protocol requirement.

Table 2. List of used notations.

Symbol	Description
P	Generator point of a large group G
q	A large prime number
N_i	ith IoT node
CA	A trusted server
ID_i	The unique identifier of N_i
d_{SM}	The ECC based private key of the smart meter (SM)
d_{NAN}	The ECC based private key of neighborhood area network gateway (NAN)
$Q_{SM/NAN}$	The ECC based public key of SM/NAN
$r_{SM/NAN}$	A random number generated by SM/NAN
$Auth_{SM/NAN}$	Authentication token generated by SM/NAN
$H(.)$	One-way hash function
$T_{SM/NAN}$	Timestamp of SM/NAN
$I_{SM/NAN}$	Identifier of SM/NAN
$a.P$	Multiplying a point P on the elliptic curve E by natural number (scalar) a, results another point on the curve
$\|\|$	Concatenation
ΔT	An acceptable threshold for time
SK	The shared session key between SM and NAN gateway
$\|X\|$	Cardinality of the set X

To thwart potential attacks stemming from the exposure of secret information, we presume that every smart meter (SM) is equipped with a robust Physical unclonable function ($PUF(.)$). This measure is essential considering the attacker's capability to compromise a SM and extract its confidential data. Given challenges $\mathcal{C} \neq \mathcal{C}'$, $PUF(\mathcal{C})$ and $PUF(\mathcal{C}')$ are expected to be completely different. On the other hand, given the same challenge $\mathcal{C}$ to $PUF(.)$, it is expected to have the same response. However, different PUFs should return completely different responses for the same challenge with a high probability. It should be noted attempting to design such a PUF function is an active research area but out of the scope of this paper, although many proposed schemes are vulnerable to modeling attacks or machine learning attacks [36]. An example of such attempts is the proposed scheme by [37] Zalivaka et al., which is claimed to be reliable and secure against modeling attacks.

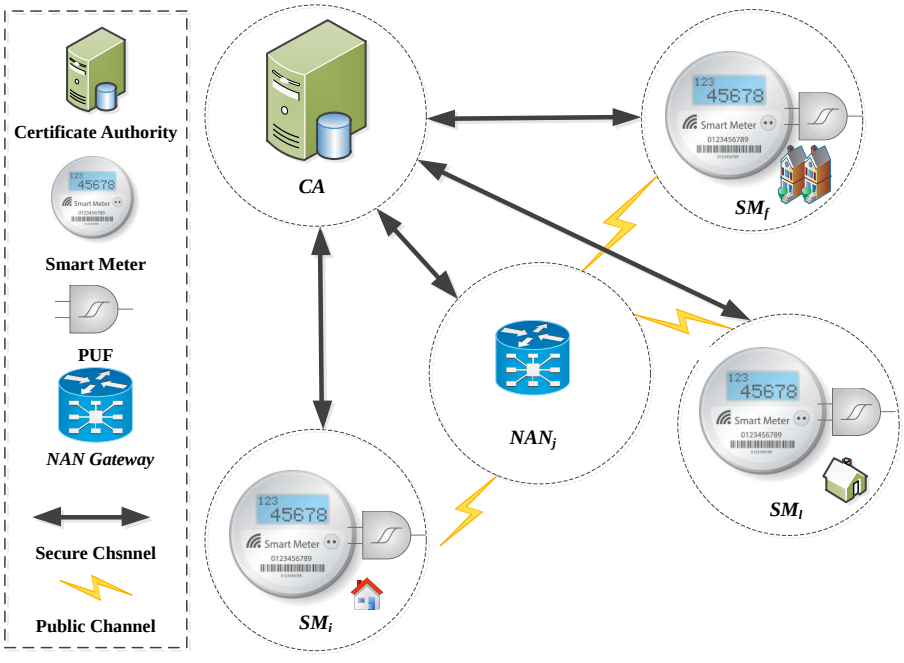

Figure 1. Infrastructure of a smart metering system.

We assume that the public information is stored in the smart metering infrastructure (SMI), which is accessible by all protocol parties (including the adversary), but its integrity is guaranteed and the adversary cannot modify its content.

4. Proposed Protocol (PPSG)

To overcome the security pitfalls of existing protocols, following our system model, we propose a secure protocol that is named PUF-based protocol for the smart grid—in the shortened form, PPSG. In the initialization phase, the CA selects and discloses the protocol's parameters publicly in the smart metering infrastructure (SMI). We take into consideration that each smart meter is outfitted with a $PUF(.)$. As a result, during this stage, the certificate authority (CA) discloses the system parameters, i.e., $\{q, P, h(.), E_q(c, d)\}$, and they are stored in the SMI.

The registration phase of the protocol is used for the SMs and the NAN gateways enrollment to the CA over a secure channel. In this phase, to register a SM, it generates an identity I_{SM} for itself and transmits it to the CA. It will be accepted by the CA if it is unique, i.e., has not been used by another SM already. When SM chooses a unique identity I_{SM}, the CA assigns it to the SM and generates a pair (d_{SM}, $Q_{SM} = d_{SM}.P$) as the SM's private and public keys, respectively. The CA then sends the token $< d_{SM}, Q_{SM} = d_{SM}.P >$ to the SM through a secure channel and deletes d_{SM} from its database. Once the message is received, the SM stores $(I_{SM}, sd_{SM} = PUF(I_{SM}) \oplus d_{SM}, Q_{SM})$ in its memory. To register a NAN gateway, the same process will be run, and it chooses its unique identifier I_{NAN} and the CA computes $< d_{NAN}, Q_{NAN} = d_{NAN}.P >$ as its private and public keys, respectively, and shares with the NAN. The set I_{SM}, Q_{SM} is also stored in the smart metering infrastructure (SMI), similarly (I_{NAN}, Q_{NAN}).

Assume that the ith SM, which is denoted by SM_i, wants to communicate with a nearby jth NAN gateway, which is denoted by NAN_j. The mutual authentication and key agreement phase of the protocol process is as follows, as also depicted in Figure 2:

1. The SM_i obtains I_{NAN} and Q_{NAN} from SMI, generates a random number $r_{SM} \in Z_q^*$ and the timestamp T_{SM}, computes $R1_{SM} = r_{SM}.Q_{SM}$, $R2_{SM} = r_{SM}.(PUF(I_{SM}) \oplus sd_{SM}).Q_{NAN}$, and $Auth1_{SM} = H(R2_{SM}, I_{SM}, T_{SM})$ and sends the message $M_1 =< (I_{SM}, Auth1_{SM}) \oplus R2_{SM}, R1_{SM}, T_{SM} >$ to the NAN.
2. Once the NAN_j received M_1, it validates T_{SM}, calculates $R2^*_{SM} = d_{NAN}.R1_{SM}$, and extracts I^*_{SM} and $Auth1^*_{SM}$. Next, it verifies whether $Auth1^*_{SM} \stackrel{?}{=} H(R2^*_{SM}, I^*_{SM}, T_{SM})$ to accept the login request. Assuming the request has been accepted, using I^*_{SM}, NAN_j obtains Q^*_{SM} from SMI, generates a random number $r_{NAN} \in Z_q^*$ and its timestamp T_{NAN} and computes $R1_{NAN} = r_{NAN}.Q_{NAN}$, $R2_{NAN} = r_{NAN}.R2^*_{SM}$ and $Auth_{NAN} = H(R2_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN})$, and sends the message $M_2 =< Auth_{NAN}, R1_{NAN}, T_{NAN} >$ to the SM_i.
3. Once the SM_i receives M_2, it validates T_{NAN}, calculates $R2^*_{NAN} = (PUF(I_{SM}) \oplus sd_{SM}).r_{sm}.R1_{NAN}$, and verifies whether $Auth_{NAN} \stackrel{?}{=} H(R2^*_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN})$ to authenticate the NAN_j. Next, it extracts its current timestamp T'_{SM} and computes the shared key $SK = H(I_{NAN} \| I_{SM} \| R2^*_{NAN} \| T'_{SM} \| T_{NAN})$ and $Auth2_{SM} = H(SK \| T'_{SM})$ and sends $M_3 =< Auth2_{SM}, T'_{SM} >$ to the NAN_j.
4. Once the NAN_j receives M_3, it verifies T'_{SM}, calculates $SK^* = H(I_{NAN} \| I^*_{SM} \| R2_{NAN} \| T'_{SM} \| T_{NAN})$, and verifies whether $Auth2_{SM} \stackrel{?}{=} H(SK^* \| T'_{SM})$ to authenticate the SM_i.
5. Once the legitimacy of both SM_i and NAN_j has been verified and they have been successfully authenticated, the mutual authentication and key agreement process concludes, and the shared key will be $SK = H(I_{NAN} \| I_{SM} \| r_{NAN}.r_{SM}.d_{SM}.d_{NAN}.P \| T'_{SM} \| T_{NAN})$.

SM_i	NAN_j
$(I_{SM}, sd_{SM} = PUF(I_{SM}) \oplus d_{SM}, Q_{SM})$	$< d_{NAN}, Q_{NAN} = d_{NAN}.P >$
Obtains I_{NAN} and Q_{NAN} from SMI, generates $r_{SM} \in Z_q^*$ and T_{SM}, computes $R1_{SM} = r_{SM}.Q_{SM}$, $R2_{SM} = r_{SM}.(PUF(I_{SM}) \oplus sd_{SM}).Q_{NAN}$ and $Auth1_{SM} = H(R2_{SM}, I_{SM}, T_{SM})$ $\xrightarrow{M_1=<(I_{SM}, Auth1_{SM}) \oplus R2_{SM}, R1_{SM}, T_{SM}>}$	
	Validates T_{SM}, calculates $R2^*_{SM} = d_{NAN}.R1_{SM}$ and extracts I^*_{SM} and $Auth1^*_{SM}$, verifies $Auth1^*_{SM} \stackrel{?}{=} H(R2^*_{SM}, I^*_{SM}, T_{SM})$ to accept the login request. Assuming the request has been accepted, using I^*_{SM}, NAN_j obtains Q^*_{SM} from SMI, generates $r_{NAN} \in Z_q^*$ and T_{NAN}, and computes $R1_{NAN} = r_{NAN}.Q_{NAN}$, $R2_{NAN} = r_{NAN}.R2^*_{SM}$ and $Auth_{NAN} = H(R2_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN})$ $\xleftarrow{M_2=<Auth_{NAN}, R1_{NAN}, T_{NAN}>}$.
Validates T_{NAN}, calculates $R2^*_{NAN} = (PUF(I_{SM}) \oplus sd_{SM}).r_{sm}.R1_{NAN}$, verifies $Auth_{NAN} \stackrel{?}{=} H(R2^*_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN})$ to authenticate the NAN_j, extracts T'_{SM}, and computes $SK = H(I_{NAN} \| I_{SM} \| R2^*_{NAN} \| T'_{SM} \| T_{NAN})$ and $Auth2_{SM} = H(SK \| T'_{SM})$ $\xrightarrow{M_3=<Auth2_{SM}, T'_{SM}>}$	
	Verifies T'_{SM}, calculates $SK^* = H(I_{NAN} \| I^*_{SM} \| R2_{NAN} \| T'_{SM} \| T_{NAN})$, and verifies $Auth2_{SM} \stackrel{?}{=} H(SK^* \| T'_{SM})$ to authenticate the SM_i $\xleftarrow{M_4=<Auth2_{SM}>}$.

Figure 2. Mutual authentication and key agreement phase of PPSG.

5. Security Analysis of PPSG

In this section, we embark on a comprehensive security evaluation of the proposed authentication and key agreement protocol for smart grid PPSG. The primary objective of this evaluation is to thoroughly assess the security aspects of PPSG from multiple perspectives.

To begin with, we employ a heuristic evaluation approach to scrutinize the security of PPSG. This method involves a systematic examination of the protocol's components, algorithms, and implementation details to identify any potential vulnerabilities or weaknesses. Through this heuristic analysis, we leverage our expertise and knowledge in the field to identify possible security risks and provide valuable insights into the overall security posture of PPSG.

Furthermore, we conduct a formal security evaluation of PPSG within the real-or-random model. By adopting this formal model, we can rigorously assess the security guarantees provided by the protocol.

In order to further validate and reinforce the security claims of PPSG, we employ an automated security protocol verification tool named Scyther [38]. This tool plays a crucial role in validating the security properties of the protocol by subjecting it to rigorous analysis. By utilizing Scyther's advanced algorithms and formal methods, we can exhaustively examine PPSG for any potential security flaws, design vulnerabilities, or weaknesses. The

utilization of Scyther ensures a comprehensive assessment of the security of PPSG and offers additional confidence in its effectiveness.

By combining these three evaluation approaches—heuristic evaluation, formal analysis in the real-or-random model, and security validation using Scyther—we aim to provide a robust and multi-dimensional assessment of the security of the proposed PUF-based authentication and key agreement protocol for smart grid PPSG. This comprehensive evaluation approach enhances the reliability of our findings and strengthens the confidence in the security claims made for PPSG.

5.1. Heuristic Security Evaluation

In the PPSG protocol, the secret key of the smart meter, denoted as d_{SM}, is protected through the process of masking with the $PUF(.)$ function. As a result, if an adversary manages to compromise the smart meter, they are unable to directly extract the original secret key d_{SM}. Instead, the adversary can only obtain a modified version of the secret key, denoted as sd_{SM}, which is computed as the XOR operation between d_{SM} and the output of the $PUF(.)$ function applied to a unique identifier I_{SM} associated with the smart meter. Mathematically, this can be expressed as $sd_{SM} = d_{SM} \oplus PUF(I_{SM})$. Assuming that the employed $PUF(.)$ function is secure enough, the adversary faces significant difficulties in extracting the actual secret key d_{SM} or conducting related attacks, such as impersonation. The security of the PPSG protocol relies on the assumption that the $PUF(.)$ function effectively masks the secret key and prevents its direct extraction. By leveraging the security properties of the $PUF(.)$ function, the protocol ensures that even if the adversary compromises the smart meter, they cannot obtain the original secret key and are limited to accessing the modified version sd_{SM}.

Additionally, the session key used in the PPSG protocol is randomized using a combination of parameters: r_{NAN}, r_{SM}, d_{SM}, d_{NAN}, and P. The adversary, however, only has access to $r_{NAN} \cdot Q_{NAN}$ and $r_{SM} \cdot Q_{SM}$, where Q_{NAN} and Q_{SM} are the public keys corresponding to d_{NAN} and d_{SM}, respectively. This means that even if the adversary possesses knowledge of d_{NAN} and d_{SM}, they are unable to extract the session key without solving the elliptic curve discrete logarithm problem (ECDLP) or the elliptic curve computational Diffie–Hellman problem (EC-CDHP). Therefore, the proposed PPSG protocol provides forward secrecy, ensuring that even with compromised long-term secret keys, the adversary cannot retroactively derive the session key. In the proposed protocol, the integrity of the messages is guaranteed by the following equations:

$$\begin{aligned} Auth1_{SM} &= H(R2_{SM}, I_{SM}, T_{SM}) \\ Auth_{NAN} &= H(R2_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN}) \\ Auth2_{SM} &= H(SK \| T'_{SM}) \end{aligned}$$

where $SK = H(I_{NAN} \| I_{SM} \| R2^*_{NAN} \| T'_{SM} \| T_{NAN})$. Given that the timestamp has been used in all messages, the adversary cannot use these messages in a later session to apply a replay attack, thus demonstrating the security of PPSA against this attack.

In conclusion, the PPSG protocol strengthens the security of the smart grid system by masking the secret key using the $PUF(\cdot)$ function, preventing direct extraction. Furthermore, the randomized session key construction and the computational hardness of the ECDLP or EC-CDHP problems ensure forward secrecy, protecting the confidentiality of past sessions even in the presence of compromised long-term secret keys.

5.2. Formal Security Evaluation—RoR

Throughout the remaining part of this section, we conduct a comprehensive security evaluation of PPSG within the framework of the real-or-random (RoR) model. In this model, an initial random selection is made where a bit b is uniformly chosen; when $b = 0$, it represents the random world (RW), and when $b = 1$, it signifies the real world (target

protocol). The adversary's objective is to accurately distinguish the value of b in this scenario. To do this, the adversary $\mathcal{A}$ can run the following query types [39]:

- Execute: it models a passive adversary $\mathcal{A}$, which eavesdrops transferred messages over public channel;
- Send: it models an active adversary on the public channel;
- Reveal (N_i): its output is the session key that is held by the instance N_i;
- Test (N_i): it returns the session key for instance N_i if $b = 1$ or a random value of the same size if $b = 0$.

Consider protocol $\mathcal{P}$, in which $\mathcal{A}$ is given access to the Execute , Send, Reveal (N_i) and Test (N_i) oracles, and outputs a guess bit b_0. The adversary wins the semantic security game in the RoR sense if $b_0 = b$ and its advantage to win this game, $Adv^{RoR}_{\mathcal{D},\mathcal{P}}(t, R)$, is defined as follows:

$$Adv^{RoR}_{\mathcal{D},\mathcal{P}}(t, R) \quad = \quad \left((Pr(\mathcal{A} \rightarrow b_0 = 1 : b = 1) - (Pr(\mathcal{A} \rightarrow b_0 = 1 : b = 0))\right)$$

$\mathcal{P}$ offers RoR semantic security if:

$$Adv^{RoR}_{\mathcal{D},\mathcal{P}(t,R)} < \varepsilon(.)$$

with $\varepsilon(.)$ being some negligible function.

In this section, as outlined in [39], we conduct a formal assessment of PPSG's security within the RoR model. This evaluation involves gauging the adversary's advantage in differentiating PPSG from the random world (*RW*).

Theorem 1. *Let q_{exe}, q_{send}, q_{Reveal}, and q_{test}, respectively, represent the number of queries to* Execute, Send, Reveal, *and* Test *oracles on PPSG/RW, then:*

$$\begin{aligned} Adv^{RoR}_{\mathcal{D},PPSG}(t, q_{exe}; q_{send}; q_{Reveal}; q_{test}) \quad &- \\ Adv^{RoR}_{\mathcal{D},RW}(t, q_{exe}; q_{send}; q_{Reveal}; q_{test}) \quad &\leq \\ 3.q.\varepsilon_{ECC} + 4.q.\varepsilon_{H} + q.\varepsilon_{PUF} \end{aligned}$$

In the given context, ε_{ECC} represents the utmost advantage an adversary can gain in solving ECDLP or EC-CDHP with each query. Additionally, ε_H signifies the maximum advantage in challenging the collision resistance property of $H(.)$, while ε_{PUF} denotes the maximum advantage in distinguishing the output of $PUF(.)$ from a random sequence. Here, q is calculated as the sum of q_{exe}, q_{send}, q_{Reveal}, and q_{test}.

Proof. Consider the scenario where SM_i and NAN_j engage in communication to establish a session key SK. Let $\mathcal{A}$ denote an adversary aiming to challenge the semantic security of PPSG within the real-or-random (RoR) model.

To establish the theorem, a game-based methodology is employed. This approach involves defining a sequence of games denoted as $\mathcal{G}$, initiating from the random world *RW*, and concluding in the real-world PPSG. Each game, represented as $\mathcal{G}_n$, introduces an event $Adv^{RoR-\mathcal{G}_n}_{\mathcal{D},\mathcal{P}}(t, R)$. This event signifies the adversary's advantage in accurately determining the hidden bit b involved in the Test queries. It should be noted the structure of the transferred messages is identical in *RW* and PPSG, including plain values such as timestamps; otherwise distinguishing them is trivial.

Game $\mathcal{G}_0$. It is corresponding to *RW* and $Adv^{RoR-\mathcal{G}0}_{\mathcal{D},RW}(t, R) = 0$.

Game $\mathcal{G}_1$. In this game, any instance follows the structure of the generated and transferred messages in PPSG, e.g., $(SK, R1_{SM}, \ldots)$. However, all computed messages, excluding timestamps, are selected completely randomly. It is clear $Adv^{RoR-\mathcal{G}0}_{\mathcal{D},RW}(t, R) - Adv^{RoR-\mathcal{G}1}_{\mathcal{D},RW}(t, R) = 0$.

Game $\mathcal{G}_2$. In this game, $Auth1_{SM} = H(R2_{SM}, I_{SM}, T_{SM})$, $Auth2_{SM} = H(SK \| T'_{SM})$, and $Auth_{NAN} = H(R2_{NAN}, I_{NAN} \oplus I_{SM}, T_{SM} \oplus T_{NAN})$. Given that $R2_{SM}$, $R2_{SM}$, and SK

are session-dependent random values, this modification has no impact on the adversary's advantage as long as $H(.)$ is not distinguishable from a random function. Hence:

$$Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_2}(t,R) \leq Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_1}(t,R) + 3.q.\varepsilon_H$$

where $q = q_{exe} + q_{send} + q_{test}$.

Game $\mathcal{G}_3$. In this game, $R1_{SM}$ and $R2_{NAN}$ are calculated using ECC point multiplication. Given that r_{SM} and r_{NAN} are fresh random numbers, the adversary's advantage to distinguish $\mathcal{G}_3$ from $\mathcal{G}_2$ is as follows:

$$Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_3}(t,R) \leq Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_2}(t,R) + 2.q.\varepsilon_{ECC}.$$

Game $\mathcal{G}_4$. In this game, as a part of the transferred messages, the values of $(I_{SM}, Auth1_{SM}) \oplus R2_{SM}$ is used in M_1, where $R2_{SM} = r_{SM}.(PUF(I_{SM}) \oplus sd_{SM}).Q_{NAN}$. It is clear this modification does not affect the adversary's advantage as long as it cannot solve ECDLP or EC-CDHP. Hence,

$$Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_4}(t,R) \leq Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_3}(t,R) + q.\varepsilon_{ECC}.$$

Game $\mathcal{G}_5$. In this game, I_{SM}, Q_{SM}, I_{NAN}, and Q_{NAN} are, respectively, replaced by their real values and are taken from SMI. However, all these parameters are already masked by ECC or $H(.)$ and we have considered the adversary's advantages of those masking in the previous games. Hence, this modification does not give a new advantage to $\mathcal{A}$, and $Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_5}(t,R) = Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_4}(t,R)$.

Game $\mathcal{G}_6$. This game is identical to $\mathcal{G}_5$, excluding that d_{SM} is computed as $PUF(I_{SM}) \oplus sd_{SM}$. Hence,

$$Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_6}(t,R) \leq Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_5}(t,R) + q.\varepsilon_{PUF}.$$

Game $\mathcal{G}_7$. In this game, the session key is computed using the hash function as $SK = H(I_{NAN} \| I_{SM} \| R2_{NAN} \| T'_{SM} \| T_{NAN})$. Given that the input value for SK_{ij} is randomized by nonce and the timestamps therefore,

$$Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_7}(t,R) \leq Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_6}(t,R) + q.\varepsilon_H.$$

It is clear that $\mathcal{G}_7$ represents the implementation of PPSG. Hence,

$$\begin{aligned} Adv_{\mathcal{D},PPSG}^{RoR}(t,R) - Adv_{\mathcal{D},RW}^{RoR}(t,R) &\leq \\ Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_7}(t,R) - Adv_{\mathcal{D},RW}^{RoR-\mathcal{G}_0}(t,R) &\leq \\ 3.q.\varepsilon_{ECC} + 4.q.\varepsilon_H + q.\varepsilon_{PUF} & \end{aligned}$$

which completes the proof. □

5.3. Formal Security Validation—Scyther

In this section, we validate the security of the proposed protocol using the Scyther tool. Scyther is a powerful tool that is widely used for security analysis and verification of cryptographic protocols. One of the main advantages of Scyther is its ability to detect vulnerabilities in protocols that are not easily detected by other tools. It uses a formal language called SPDL (Scyther protocol description language) to specify the protocol being analyzed. SPDL allows Scyther to model the protocol's behavior and identify potential weaknesses or flaws in its design. Scyther makes several security claims, including the ability to detect all possible attacks on a protocol, provide a complete analysis of its security properties, and offer automated proof of security properties. Some of the specific security claims made by Scyther include:

- Alive: Scyther claims to be able to detect liveness violations, which occur when a protocol becomes stuck or deadlocked;

- Secret: Scyther claims to be able to detect confidentiality violations, which occur when an attacker gains unauthorized access to sensitive information;
- Weakagree: Scyther claims to be able to detect weaknesses in agreement protocols, which are used to establish shared secrets between parties;
- Niagree: Scyther claims to be able to detect non-injective agreement protocols, which can allow an attacker to impersonate one of the parties involved;
- Nisynch: Scyther claims to be able to detect non-injective synchronization protocols, which can allow an attacker to manipulate the order of messages between parties.

The security analysis results of the proposed protocol, i.e., PPSG, are depicted in Figure 3.

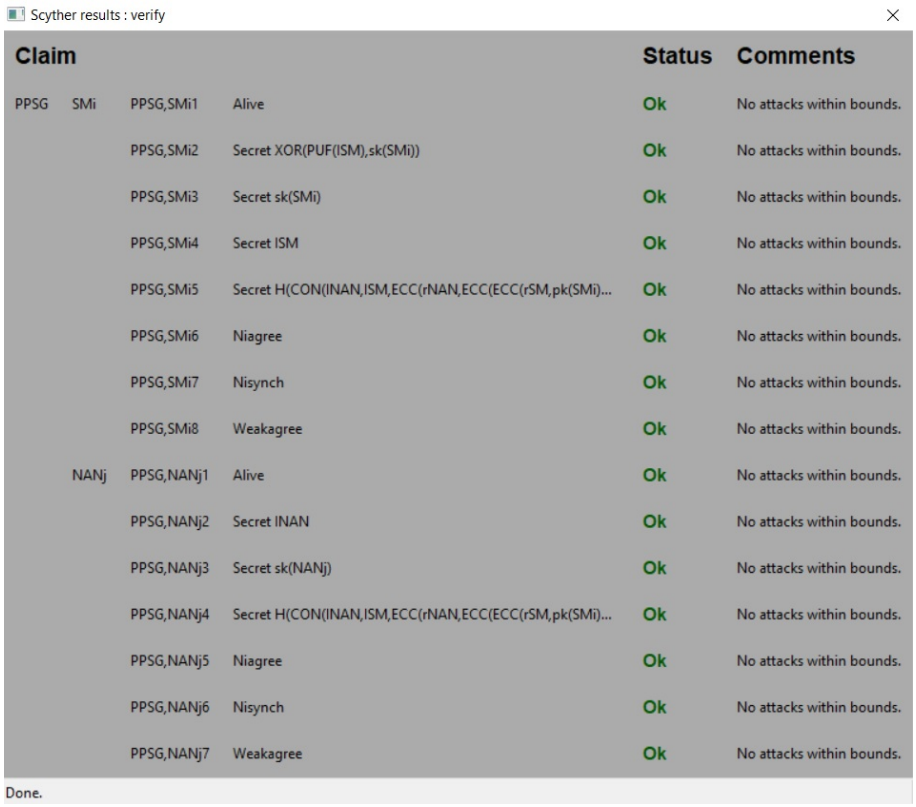

Figure 3. Security validation of PPSG using Scyther tool.

6. Cost Analysis

To set up our experiments and obtain practical results, we designed a simulation of a smart home network, as shown in Figure 4. This simulation included crucial components like a microcontroller, photoresistor sensor, humidity sensor, temperature sensor, and a relay for controlling AC power. To perform cryptographic operations on each smart meter client, we employed an Arduino UNO board. This board is equipped with an ATmega328P microcontroller boasting 32-kB flash memory, 2-kB SRAM, and operates at a clock speed of 16 MHz. It is worth mentioning that we verified the reliability of physical unclonable functions (PUF) in a similar microcontroller, as stated in [40].

In our evaluation, we examined the power-up values of SRAM in 20 microcontrollers, collected 100 times at room temperature. This assessment aimed to determine the quality of these values for implementing an SRAM PUF. The results were promising: the mean bias across all devices (indicating uniformity) stood at 48.38%, a figure remarkably close to the ideal 50%. Moreover, the intra-distance between different acquisitions (which measures reliability) was 97.58%, indicating strong consistency. Additionally, the inter-distance between distinct devices (indicative of uniqueness) was 38.62%, aligning well with findings

in similar microcontroller studies documented in the existing literature. These results affirm the robustness of our experimental setup and validate the viability of our approach.

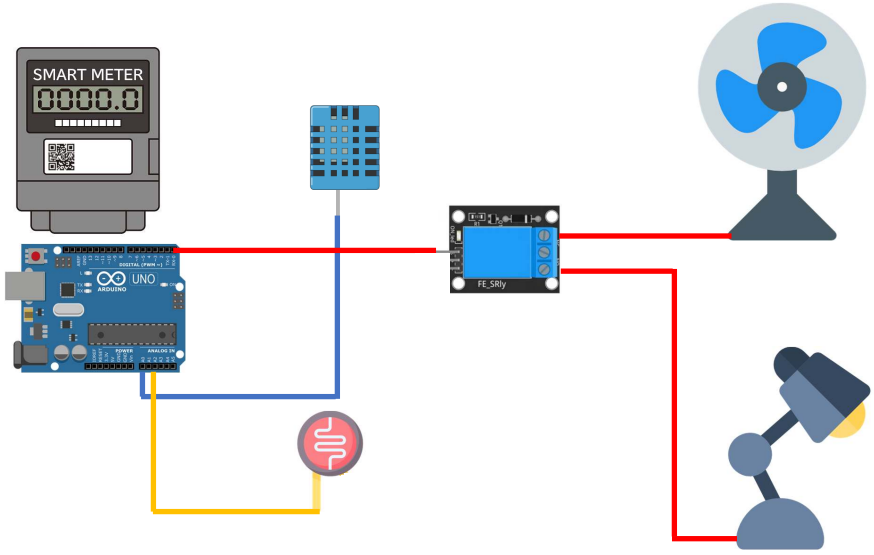

Figure 4. Simulation of a smart home system.

Using this setup, we achieved timings of approximately 21 ms for elliptic curve cryptography (T_{ECC}), 26 ms for double elliptic curve cryptography (T_{2ECC}), 6 ms for symmetric encryption (T_{SE}), 3 ms for SHA-256 hashing (T_h), and 3.7 ms for error syndrome calculation (T_{Es}). It is worth mentioning that SHA-256 might be replaced by SHA-3 based on system performance requirements or if SHA-256 is deemed insecure. We also considered the time of a PUF invocation (T_{PUFn}) as equal to T_h. This equivalence was established under the assumption of utilizing a key management module capable of generating multiple keys from a single root key. To ensure cryptographic separation between these derived keys, a secure key derivation function (KDF) utilizing cryptographic primitives like SHA-256 is employed. In a comparable research effort, functions $FE.GEN$ and $FE.REC$ utilize fuzzy extractors and helper data, among other algorithms. According to the information outlined in [23], the times for $T_{FE.REC}$ and $T_{FE.GEN}$ can be estimated as $30 \times T_{PUF}$ and $10 \times T_{PUF}$, respectively.

Garg et al. estimated the computational complexity of their protocol ([16] Section 5.2.1, Table 4) and claimed the computational complexity of the SM and the NAN are same and equal to $2.T_{em} + 4.T_H$. Based on this claim, they have shown that their protocol outperforms related protocols, e.g., [11,41]. However, after comparing Garg et al.'s protocol computational cost with PPSG, we understood that they underestimated the protocol's complexity.

By summing up all the ECC point-multiplication in the SM side, we come up with $3T_{em} + T_{2em}$ which is 150% more than the reported value by Garg et al. ([16], Section 5.2.1, Table 4), which was $2T_{em}$. The same argument can be expressed for the NAN gateway's computations. On the other hand, in PPSG, the SM's computations costs $3.T_{em} + 4.T_H + 2T_{PUF}$ and the NAN gateway's computations costs $3.T_{em} + 4.T_H$.

Table 3 displays a cost comparison between PPSG and other protocols discussed in Section 2. To compare communication overhead, we examined the bit lengths of various components: a timestamp, an identifier, a random number, a hash value, and an ECC point, which were set at 32, 64, 128, 160, and 320 bits, respectively. It is important to note that we used SHA-256 but truncated its output to 160 bits to address recent security vulnerabilities in SHA-1 [42]. Following these parameters, the communication overhead of PPSG includes $M1$ at 512 bits, $M2$ at 512 bits, $M3$ at 192 bits, and $M4$ at 160 bits, totaling 1376 bits.

Table 3. Cost comparison of the related protocols and PPSG.

Protocol	Computations	Time (ms)	Communications (Bit)	Energy (mJ)
[15]	$2 \times T_{2ECC} + 6 \times T_{ECC} + 11 \times T_h$	211	1600	18.568
[16]	$2 \times T_{2ECC} + 6 \times T_{ECC} + 8 \times T_h$	202	1344	17.776
[17]	$2 \times T_{2ECC} + 6 \times T_{ECC} + 5 \times T_h$	193	1632	16.984
[18]	$8 \times T_{ECC} + 10 \times T_h +$	198	1440	17.424
[19]	$8 \times T_{ECC} + 4 \times T_{Es} + 19 \times T_h$	240	2912	21.12
[20]	$8 \times T_h + T_{PUF} + T_{FE.REC} + 3 \times T_{ECC} + 4 \times T_{Es}$	198	1408	17.414
[21]	$10 \times T_h + 4 \times T_{SE} + 7 \times T_{ECC}$	205	1536	18.034
[23]	$11 \times T_h + T_{PUF} + T_{FE.GEN} + T_{FE.REC}$	156	896	13.728
[25]	$9 \times T_h + T_{PUF} + 6 \times T_{ECC}$	156	1408	13.728
[32]	$16 \times T_h + 1 \times T_{SE} + 3 \times T_{ECC} + T_{PUF} + 2 \times T_{FE.GEN}$	180	1664	15.835
PPSG	$8 \times T_h + 6 \times T_{ECC} + 1 \times T_{PUF}$	153	1376	13.468

In contrast, the study by [19] documented a communication cost of 1184 bits for identical parameters. However, our analysis uncovered a possible typographical error in their report, leading to an underestimation of the communication cost. This discrepancy might have originated from the mismatched bit lengths used for values calculated via symmetric encryption, a critical factor in accurate cost estimation. Examining the findings in Figure 5 (time and byte), our comprehensive evaluation clearly showcases PPSG's superiority, as it imposes significantly lower communication overhead compared to its counterparts. This discrepancy underscores the importance of precise calculations when assessing the efficiency of communication protocols.

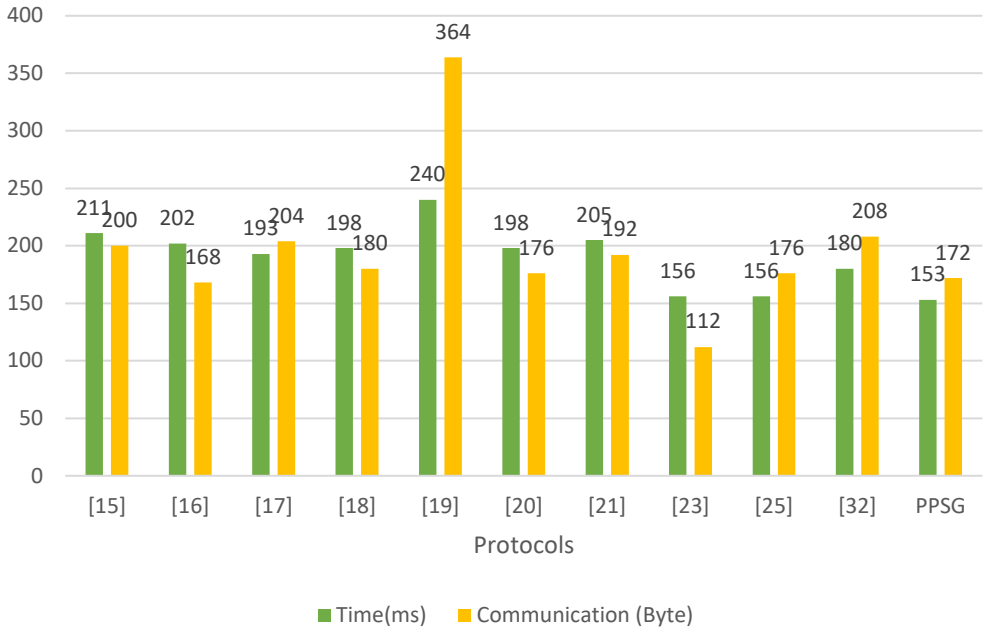

Figure 5. Computation and communication comparison of PPSG versus related protocols [15–21,23,25,32].

Regarding computational complexity, SM_i involves four hash function calls (T_h), three ECC scalar multiplications (T_{ECC}), and one PUF invocation (T_{PUF}) during its operation. On the other hand, NAN_j performs four hash function calls and three ECC scalar multiplications (T_{ECC}). Consequently, the total computational cost for the login and key agreement phase in PPSG amounts to $6 \times T_{ECC} + 1 \times T_{PUF} + 8 \times T_h$. As per our analysis, the key agreement session within PPSG demonstrates remarkable efficiency, completing in a mere 153 ms, establishing its position as the fastest protocol among those under comparison.

Energy consumption can be limited by the formula $Ec = V_{max}.I_{max}.T_c$, where Ec represents energy consumption, I_{max} stands for maximum consumed current, V_{max} represents the upper limit of working voltage, and Tc signifies the cumulative computational time essential for session key sharing. Based on the specifications outlined in the ATmega328P datasheet [43], the maximum operational power, denoted as $(V.I)$, for the ATmega328P stands at 14 mA $\times$ 5.5 V = 77 mW under active mode with a clock speed of 16 MHz. The energy measurement of PPSG on the Arduino board is depicted in Figure 6. In addition, the energy efficiency of PPSG was compared with other schemes, as illustrated in Figure 7. These findings reveal that the energy consumption for a PPSG session is notably lower compared to other schemes.

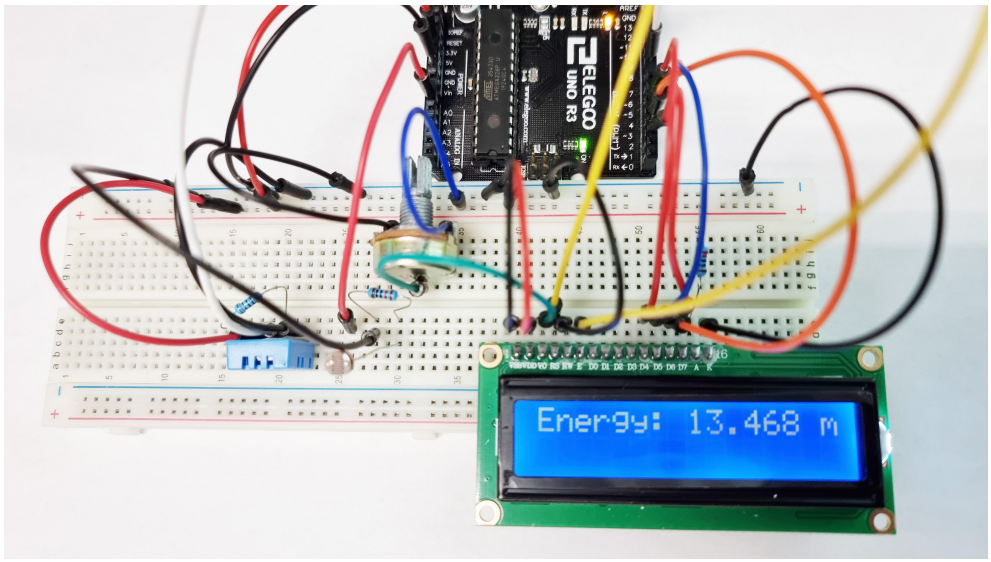

Figure 6. Energy measurement on an Arduino UNO board.

LAS-SG [21], operating as a lightweight protocol, involves the transmission of only two messages totaling 205 bits. The authentication process, taking 205 ms and consuming 18.034 mJ, indicates a resource-intensive nature compared to PPSG. Lake-BGS [32], designed for constrained smart meters (SMs), prioritizes lightweight functionality. However, being a blockchain-based protocol, it exhibits higher computation costs, with an authentication process requiring 180 ms—still more than PPSG. Consequently, with higher power consumption and overall, PPSG demonstrates superior performance.

In the comparison with [25], while both the proposed PPSG protocol and the one proposed by [25] incorporate PUF functions, PPSG outperforms it when considering computation and communication costs, resulting in lower energy consumption. This makes PPSG a more appropriate choice for smart meter security. In addressing the vulnerability of PUF functions to machine learning attacks, a novel aspect of PPSG remains in its combination of ECC and PUF. This integration safeguards transferred messages, offering an enhanced and more reliable technique compared to existing PUF-based solutions.

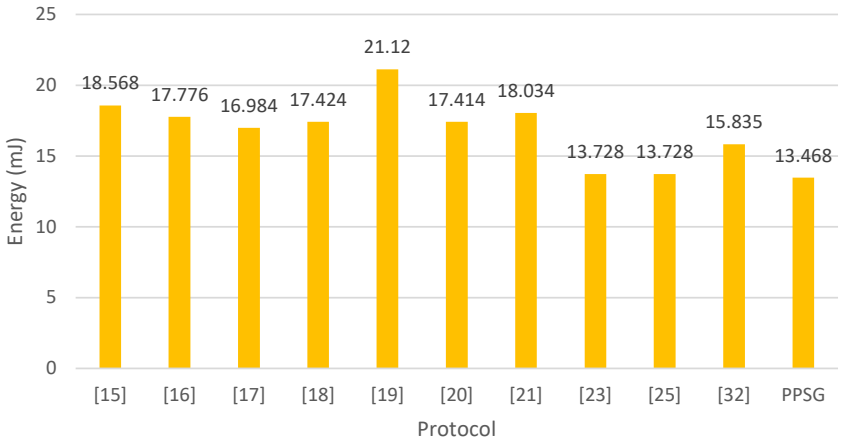

Figure 7. Energy consumption of PPSG versus related protocols [15–21,23,25,32].

7. Conclusions and Future Work

When considering the phases of grid modernization, key requirements such as data security, reliability, and accuracy are crucial for realizing the grid's full potential [44]. A breach in any of these aspects can jeopardize the entire system. This concern is even more critical in emerging new grids, such as water and gas grids, often fully working on battery, therefore operating in resource-limited environments with constraints on processing power, memory, and energy consumption.

Initially, we examined the security of various research protocols using different models, highlighting their vulnerabilities and potential solutions. Subsequently, to address these issues, we introduced a PUF-based protocol for smart grids (SG) named PPSG, utilizing PUF and ECC methods. Our formal security analysis of PPSG within the RoR model demonstrated that it offers robust security against adversaries constrained by polynomial time. Furthermore, our cost analysis revealed that PPSG stands out as one of the most lightweight protocols compared to recent related work.

To validate PPSG's performance in a real-world scenario, we simulated a smart meter and connected it to various electrical components representing smart home devices. The results showcased PPSG's superiority in terms of computation and communication costs, as well as energy consumption, making it a promising choice for smart grid applications.

Finally, although PUF-based protocols offer an interesting approach to SMs security they have certain limitations, such as vulnerability to machine learning attacks, environmental variability, and aging. In this paper, we aimed to tackle the issue of machine learning attacks by developing an efficient protocol in terms of energy use and computation speed. This area of research is a call for researchers to explore further options to overcome additional challenges. For instance, they could introduce features like temperature compensation to lessen the effect of environmental changes on PUF responses. Another approach could involve continuous monitoring to spot any alterations in PUF behavior over time. These strategies offer potential ways to enhance the overall performance of PUFs and contribute to enhancing the reliability, security, and longevity of PUF functions in smart meters.

Author Contributions: N.B., Y.B., M.S. and S.R. contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Acknowledgments: Masoumeh Safkhani was supported by Shahid Rajaee Teacher Training University under grant number 4899.

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

Reference

1. Agency, I.E. Electricity Grids and Secure Energy Transitions Report. 2023. Available online: https://www.iea.org/reports/electricity-grids-and-secure-energy-transitions (accessed on 25 December 2023).
2. What Are Smart Grids? Available online: https://www.iea.org/energy-system/electricity/smart-grids (accessed on 31 October 2023).
3. Hossein Motlagh, N.; Mohammadrezaei, M.; Hunt, J.; Zakeri, B. Internet of Things (IoT) and the Energy Sector. *Energies* **2020**, *13*, 494. [CrossRef]
4. Insights, G.M. Mart Meters Market—By Application (Residential, Commercial, Utility), By Technology (AMI, AMR), by Product (Smart Gas Meter). 2022. Available online: https://www.gminsights.com/industry-analysis/smart-metering-systems-market (accessed on 25 December 2023).
5. Analytics, I. Smart Meter Market Report 2019–2024. 2019. Available online: https://iot-analytics.com/product/smart-meter-market-report-2019-2024 (accessed on 25 December 2023).
6. Thomson, J.; Motyka, M.; Hardin, K.; Nagdeo, J. Electric Power Supply Chains: Achieving Security, Sustainability, and Resilience. 2022. Available online: https://www2.deloitte.com/us/en/insights/industry/power-and-utilities/supply-chain-resilience-electric-power-sector.html (accessed on 31 October 2023).
7. Columbus, L. Benchmarking Your Cybersecurity Budget in 2023. 2023. Available online: https://venturebeat.com/security/benchmarking-your-cybersecurity-budget-in-2023/ (accessed on 31 October 2023).
8. Ghiasi, M.; Niknam, T.; Wang, Z.; Mehrandezh, M.; Dehghani, M.; Ghadimi, N. A comprehensive review of cyber-attacks and defense mechanisms for improving security in smart grid energy systems: Past, present and future. *Electr. Power Syst. Res.* **2023**, *215*, 108975. [CrossRef]
9. Hasan, M.K.; Habib, A.A.; Shukur, Z.; Ibrahim, F.; Islam, S.; Razzaque, M.A. Review on cyber-physical and cyber-security system in smart grid: Standards, protocols, constraints, and recommendations. *J. Netw. Comput. Appl.* **2023**, *209*, 103540. [CrossRef]
10. Mazhar, T.; Irfan, H.M.; Khan, S.; Haq, I.; Ullah, I.; Iqbal, M.; Hamam, H. Analysis of Cyber Security Attacks and Its Solutions for the Smart Grid Using Machine Learning and Blockchain Methods. *Future Internet* **2023**, *15*, 83. [CrossRef]
11. Kumar, P.; Gurtov, A.; Sain, M.; Martin, A.; Ha, P.H. Lightweight authentication and key agreement for smart metering in smart energy networks. *IEEE Trans. Smart Grid* **2018**, *10*, 4349–4359. [CrossRef]
12. Baghestani, S.H.; Moazami, F.; Tahavori, M. Lightweight Authenticated Key Agreement for Smart Metering in Smart Grid. *IEEE Syst. J.* **2022**, *16*, 4983–4991. [CrossRef]
13. Kumar, N.; Aujla, G.S.; Das, A.K.; Conti, M. ECCAuth: A Secure Authentication Protocol for Demand Response Management in a Smart Grid System. *IEEE Trans. Ind. Inform.* **2019**, *15*, 6572–6582. [CrossRef]
14. Yu, S.; Park, K.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment. *Appl. Sci.* **2020**, *10*, 1758. [CrossRef]
15. Wu, F.; Xu, L.; Li, X.; Kumari, S.; Karuppiah, M.; Obaidat, M. S. A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography. *IEEE Syst. J.* **2018**, *13*, 2830–2838. [CrossRef]
16. Garg, S.; Kaur, K.; Kaddoum, G.; Rodrigues, J.J.P.C.; Guizani, M. Secure and Lightweight Authentication Scheme for Smart Metering Infrastructure in Smart Grid. *IEEE Trans. Ind. Inform.* **2020**, *16*, 3548–3557. [CrossRef]
17. He, D.; Wang, H.; Khan, M.K.; Wang, L. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. *IET Commun.* **2016**, *10*, 1795–1802. [CrossRef]
18. 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]
19. Khan, A.A.; Kumar, V.; Ahmad, M.; Rana, S.; Mishra, D. PALK: Password-based anonymous lightweight key agreement framework for smart grid Author links open overlay panel. *Int. J. Electr. Power Energy Syst.* **2020**, *121*, 106121. [CrossRef]
20. Tanveer, M.; Kumar, N.; Naushad, A.; Chaudhry, S.A. A robust access control protocol for the smart grid systems. *IEEE Internet Things J.* **2021**, *9*, 6855–6865. [CrossRef]
21. Chaudhry, S.A.; Yahya, K.; Garg, S.; Kaddoum, G.; Hassan, M.M.; Zikria, Y.B. LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments. *IEEE Trans. Ind. Inform.* **2022**, *19*, 1504–1511. [CrossRef]
22. Rincón, A.E.R.; Melo, W.S.; de Farias, C.M.; Carmo, L.F.R.C. Securing Smart Meters Through Physical Properties of Their Components. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–11. [CrossRef]
23. Gope, P.; Sikdar, B. Privacy-aware authenticated key agreement scheme for secure smart grid communication. *IEEE Trans. Smart Grid* **2018**, *10*, 3953–3962. [CrossRef]
24. Braeken, A.; Kumar, P.; Martin, A. Efficient and provably secure key agreement for modern smart metering communications. *Energies* **2018**, *11*, 2662. [CrossRef]
25. Rostampour, S.; Bagheri, N.; Ghavami, B.; Bendavid, Y.; Kumari, S.; Martin, H.; Camara, C. Using a Privacy-Enhanced Authentication Process to Secure IoT-based Smart Grid Infrastructures. Available online: https://www.researchsquare.com/article/rs-2802756/v1 (accessed on 25 December 2023).
26. Mustapa, M.; Niamat, M.Y.; Nath, A.P.D.; Alam, M. Hardware-Oriented Authentication for Advanced Metering Infrastructure. *IEEE Trans. Smart Grid* **2018**, *9*, 1261–1270. [CrossRef]
27. Harishma, B.; Mathew, P.; Patranabis, S.; Chatterjee, U.; Agarwal, U.; Maheshwari, M.; Dey, S.; Mukhopadhyay, D. Safe is the New Smart: PUF-Based Authentication for Load Modification-Resistant Smart Meters. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 663–680. [CrossRef]

28. Liu, J.; Ke, L. New efficient identity based encryption without pairings. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 1561–1570. [CrossRef]
29. Salimi, M. A New Efficient Identity-Based Encryption Without Pairing. *Cryptol. Eprint Arch.* **2021**, *10*, 1561–1570.
30. Lounis, K. PUF Security: Reviewing The Validity of Spoofing Attack Against Safe is the New Smart. Available online: https://eprint.iacr.org/2021/985 (accessed on 25 December 2023).
31. Safkhani, M.; Rostampour, S.; Bendavid, Y.; Sadeghi, S.; Bagheri, N. Improving RFID/IoT-based generalized ultra-lightweight mutual authentication protocols. *J. Inf. Secur. Appl.* **2022**, *67*, 103194. [CrossRef]
32. Badshah, A.; Waqas, M.; Abbas, G.; Muhammad, F.; Abbas, Z.H.; Vimal, S.; Bilal, M. LAKE-BSG: Lightweight authenticated key exchange scheme for blockchain-enabled smart grids. *Sustain. Energy Technol. Assess.* **2022**, *52*, 102248. [CrossRef]
33. Canetti, R.; Krawczyk, H. Universally Composable Notions of Key Exchange and Secure Channels. In *Lecture Notes in Computer Science, Proceedings of the Advances in Cryptology—EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, 28 April–2 May 2002*; Knudsen, L.R., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2332, pp. 337–351. [CrossRef]
34. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
35. Jangirala, S.; Das, A.K.; Vasilakos, A.V. Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7081–7093. [CrossRef]
36. Khalafalla, M.; Gebotys, C.H. PUFs Deep Attacks: Enhanced modeling attacks using deep learning techniques to break the security of double arbiter PUFs. In Proceedings of the Design, Automation & Test in Europe Conference & Exhibition, Florence, Italy, 25–29 March 2019; pp. 204–209.
37. Zalivaka, S.S.; Ivaniuk, A.A.; Chang, C. Reliable and Modeling Attack Resistant Authentication of Arbiter PUF in FPGA Implementation With Trinary Quadruple Response. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1109–1123. [CrossRef]
38. Cremers, C.J.F. The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In Proceedings of the Computer Aided Verification, Princeton, NJ, USA, 7–14 July 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 414–418.
39. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-Based Authenticated Key Exchange in the Three-Party Setting. In *Lecture Notes in Computer Science, Proceedings of the Public Key Cryptography—PKC 2005, 8th International Workshop on Theory and Practice in Public Key Cryptography, Les Diablerets, Switzerland, 23–26 January 2005*; Vaudenay, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3386, pp. 65–84.
40. Wang, R.; Selimis, G.; Maes, R.; Goossens, S. Long-term Continuous Assessment of SRAM PUF and Source of Random Numbers. In Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2020; pp. 7–12. [CrossRef]
41. Tsai, J.; Lo, N. Secure Anonymous Key Distribution Scheme for Smart Grid. *IEEE Trans. Smart Grid* **2016**, *7*, 906–914. [CrossRef]
42. Leurent, G.; Peyrin, T. From Collisions to Chosen-Prefix Collisions Application to Full SHA-1. In *Lecture Notes in Computer Science, Proceedings of the Advances in Cryptology—EUROCRYPT 2019—38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, 19–23 May 2019*; Part III; Ishai, Y., Rijmen, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11478, pp. 527–555.
43. Atmel. 8-Bit AVR Microcontroller with 32K Bytes In-System Programmable Flash. Microchip. Available online: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf (accessed on 10 June 2020).
44. Young, R.; McCue, J.; Grant, C. The Power Is On: How IoT Technology Is Driving Energy Innovation. 2016. Available online: https://www2.deloitte.com/us/en/insights/focus/internet-of-things/iot-in-electric-power-industry.html (accessed on 25 December 2023).