
Article:
Gope, P. orcid.org/0000-0003-2786-0273 and Sikdar, B. (2021) A privacy-aware reconfigurable authenticated key exchange scheme for secure communication in smart grids. IEEE Transactions on Smart Grid, 12 (6). pp. 5335-5348. ISSN 1949-3053

https://doi.org/10.1109/tsg.2021.3106105

A Privacy-Aware Reconfigurable Authenticated Key Exchange Scheme for Secure Communication in Smart Grids

Prosanta Gope *Senior Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—The smart metering infrastructure plays an important role in smart grid environments. Such metering networks need to be protected against cyber attacks by using authenticated key exchange protocols, and many relevant schemes have been presented by researchers. In addition, in order to protect against the energy theft problem, it is also important to consider physical security of the smart meter. Recently, PUFs (physical uncloneable functions) have gained popularity as a primitive against physical attacks. In 2019, we proposed the *first* PUF-based authentication scheme for secure smart grid communication with resilience against physical attacks on smart meters. However, recent studies have shown that PUFs are susceptible to modeling attacks. To address this issue, this paper proposes a reconfigureable authenticated key exchange scheme for secure communication in smart grids by using the concept of reconfigureable PUFs. In addition to security, the efficiency evaluation demonstrates that our new scheme has advantages in both the computation and communication costs as compared to the state-of-the-art protocols.**

***Index Terms*—Mutual Authentication, Smart Meter, RPUF, Secure Communication.**

I. Introduction

SMART GRIDS incorporate advanced information, communication, and computing technologies to facilitate improved management and resilience of various aspects of electricity generation, transmission and distribution. One of the key components of smart grids is the advanced metering infrastructure (AMI) that provides two main functions: (i) it measures and collects the electricity consumption data from the consumer, and (ii) it assists with the delivery and use of pricing signals at consumer end for purposes such as demand side management. Thus, AMI plays an important role in the overall management and reliability of the grid, and is of benefit to both the consumers and the utilities.

The importance of AMI for the operation of smart grids also makes it an attractive target for a wide range of adversaries. For example, a breach in the security of the communication between a smart meter and the utility may be exploited to violate the privacy of the consumer. Security breaches may also be exploited for purposes of energy theft, disrupting the balance between supply and demand, increasing the peak usage, and cause outages [15] [19]. Consequently, the security of AMI has received considerable attention and cryptography-based solutions for secure communications have been proposed. One of the fundamental requirements of such cryptographic systems is the establishment of a key distribution and management system. The security of the key distribution system is imperative as any successful compromise can lead to a complete loss of security. The objective of this paper is to develop a key exchange mechanism for use in smart grids. The proposed solutions is specifically designed with the operating conditions and hardware limitations of devices in smart grid AMI. In addition, the proposed solution addresses the recently discovered machine learning attacks that are capable of compromising existing solutions.

A. Related Work and Motivation

Several authenticated key-exchange (AKE) schemes have been proposed in literature for secure smart grid communication, and majority of these are based on public-key crypto systems. For instance, in 2014, Nicanfar et al. [1] proposed an AKE scheme for smart grid communications, in which a key generator was used to update the public and private keys, together with the multi-casting keys. Consequently, this scheme is not suitable in practice, as shown in [2]. Subsequently, Tsai and Lo [3] introduced an anonymous key distribution scheme for smart grid environments by using identity-based cryptography. However, as shown in [4], the scheme presented in [3] cannot ensure security against the ephemeral secret leakage attack or ensure privacy of the secret credential in the smart meter. The authors of [4] also introduced a new authenticated key exchange scheme for smart girds with the assertion that their scheme can ensure better security level than others. However, as pointed out by Chen et al. [5], the scheme presented in [4] has a security weakness, where the private key of each smart meter is generated by a key generation center (KGC), and consequently the corresponding smart meters can be tracked and impersonated by the KGC. Subsequently, Abbasinezhad-Mood and Nikoohgadam [6] introduced an anonymous key distribution scheme for smart grids by using the ECQV [17] mechanism. However, as reported by Braeken et al. in [7], when the long term secret keys held by the smart meter and service provider in the scheme from [6] are revealed to an adversary, then the adversary can derive the session keys established in previous rounds. Apart from [1-6], recently a few more public-key-based solutions [21-25], [34-35] and [36] have been proposed

P. Gope is with Department of Computer Science, University of Sheffield, Regent Court, Sheffield S1 4DP, United Kingdom. (E-mail: prosanta.nitdgp@gmail.com/p.gope@sheffield.ac.uk)

B. Sikdar is with Department of Electrical and Computer Engineering, National University of Singapore, 21 Lower Kent Ridge Rd, Singapore 119077. (Email: bsikdar@nus.edu.sg)

Corresponding author: Dr. Prosanta Gope

Table I
COMPARATIVE ANALYSIS OF THE RELATED SCHEMES

Scheme	Primitive Used	Physical-Security-of-the-Smart-Meter
[1-7], [21-25], [34-39]	Public-key-based such as ECC, ECQV, Chebyshev Chaotic Maps	No
[8-11], [26-28], [40-41]	Non-Public-key-based such as Hash Function, PUF, XOR, etc.	Only [11] and [28]
Proposed Scheme	Non-Public-key-based such as PUF, Hash-Function, XOR, FHD	Yes (with ML-attack-resilience)

in literature. For instance, in [21] the authors proposed a privacy-preserving recording and gateway-assisted authentication protocol using homomorphic encryption and bloom filter. Wazid et al. [22] considered the authentication requirement in smart grids where an individual needs to be authenticated to access a smart meter, and they presented a three-factor user authentication scheme for this scenario. Qi and Chen [23] presented a two-pass privacy preserving AKE scheme for smart grids by using the ECQV implicit certificate mechanism. Mahmood et al. [35] introduced a scheme using ECC and lightweight hash function for secure connection between smart electrical equipment and distributed substations. The scheme includes mutual authentication between a distributer and a consumer via a reliable third party and ultimately a session key is agreed between the parties for further communications. However, according to [36], the proposed scheme cannot ensure session-key security, and resilience against impersonation attack. Subsequently, four more Elliptic curve cryptography (ECC)-based authentication protocols have been proposed in [24], [25], [34], and [36]. Recently, Abbasinezhad-Mood et al. [37] proposed a new key distribution scheme with privacy protection based on ECC by preserving the anonymity of the participants. They evaluated the security of this scheme by considering the Random-Oracle Model and applied the cryptographic protocols on ARM chips. They also proposed two other schemes [38], [39] to read isolated smart meters by using extended Chebyshev Chaotic Maps. One important issue with all the above public–key-based schemes is their computational complexity. Also, none of the above existing works has considered the physical security of smart meters, which is greatly important for resisting inside attackers (e.g., a home user) from compromising and controlling smart meters for their own profit.

On the other hand, since smart meters have limited computational abilities, AKE schemes that focus on efficiency by avoiding the use of public key cryptography have also been proposed [8-11], [26-28], and [40-41]. For instance, the authors of [40] proposed a secure communication scheme based on the Lagrange polynomial, bit-wise XOR, and hash function for secure smart grid communication. In [41], the authors have introduced an ultra-lightweight protocol using bit-wise XOR and hash function for secure communication in smart grids. Among the above non-public-key-based protocols ([8-11], [26-28], and [40-41]), only the protocols in [11] and [28] have considered the physical security of smart meters. In this regard, the authors of [11] introduced an AKE protocol for smart grids by utilizing PUFs without using public key cryptography. Although PUFs are fundamentally based on random physical variations and are consequently supposed to be unclonable, they may be susceptible to attacks that aim to model their behavior using Machine Learning (ML) techniques [12-13]. In fact, by having access to a subset of the challenge-response pairs (CRPs) of a PUF, an adversary may be able to model the PUF, including strong PUFs. Therefore, it is necessary to prevent the interception of challenge-response exchange messages used for authenticating the smart meters. However, the protocols presented in [11] and [28] cannot ensure security against ML-attacks, where an attacker with access to the PUF-enabled smart meter can collect sufficient number of CRPs and create a marionette PUF. After such an attack, if the marionette PUF interacts with the server, then the server will not be able to comprehend this impersonation (since the marionette PUF will be able to generate the desired PUF response for any given challenge). Furthermore, to deal with the noise issue, a mechanism is presented in [11] where a smart meter needs to execute the computationally expensive re-construction algorithm (FE.Rec), which consumes excessive CPU cycles of the smart meter (as shown in Table III of Section V). Table I compares related work to our approach with respect to the primitive used and ability to support physical-security-of-the-smart meter. In this paper, we seek to address these issues by introducing a reconfigureable authentication scheme for smart grids by using the concept of reconfigurable PUFs and fractional Hamming distance (FHD). The contributions of this paper can be summarized as follows:

- This paper presents a *new* PUF-based reconfigurable AKE scheme for securing the smart metering network in the smart grid environments. One of the notable properties of the proposed scheme is that, apart from the physical security of of the smart meter, it can also ensure resilience against modeling or machine learning attacks, which is imperative for any PUF-based authentication scheme.
- We provide a rigorous formal security analysis of our proposed scheme to show that it is secure against some of the imperative attacks.
- We demonstrate that the new AKE scheme for smart metering networks has a better efficiency in terms of both the computation and communication costs when compared with other existing AKE schemes for smart grids.

The rest of the paper has been organized as follows. In Section II, we provide a brief introduction to machine learning attacks on PUFs, RPUF-security, and fractional Hamming distance. In Section III, we present the proposed privacy-preserving, machine-learning resilient, reconfigurable authentication scheme. Security of the proposed scheme is analyzed in Section IV. A discussion on the properties of the proposed scheme is provided in Section V, with concluding remarks in Section VI. All the symbols and cryptographic functions of

Table II
SYMBOLS AND CRYPTOGRAPHIC FUNCTIONS

Symbol	Definition
SID_M^i	Shadow identity of smart-meter M for i-th round
REF_{ID}	Reference identity of the smart-meter M
(α_i)	Challenge for the i-th round
(β_i)	Response to the challenge α_i for the i-th round
$CRP(\alpha_i, \beta_i)$	Challenge-response pair for the i-th round
WPUF_M	Weak PUF attached with the secure NVM of the smart meter M
RPUF_M^i	PUF Re-Configuration for i-th round of authentication
n_x	Nonce/random number generated by the smart meter
n_s	Nonce/random number generated by the server
k_i	Round-key generated by the Weak PUF WPUF_M for the i-th round
SK	Session key
FHD	Fractional Hamming Distance
$h(.)$	Secure One-way hash function
$\oplus$	Exclusive-OR operation
$\|$	Concatenation operation

the proposed scheme are presented in Table II.

II. PRELIMINARIES

A. Machine Learning Attacks on PUFs

Physical unclonable functions are a promising security primitive that can be utilized by lightweight authentication protocols to facilitate high levels of security while simultaneously minimizing the computational resource requirement per device. The operation of a PUF can be expressed as the function: $\beta \leftarrow \mathrm{PUF}(\alpha)$ where the variables α and β serve as the challenge and response pair (CRP). A PUF always returns the same β for a given challenge α, if tested multiple times.

Even since the first use of PUFs as security primitives, machine learning has been known to be a powerful threat to PUFs by enabling the development of modeling attacks. These types of attacks generally involve an adversary collecting a large subset of a PUF's possible CRPs: $\{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \cdots, (\alpha_w, \beta_w)\}$. Using this collected data, a mathematical model, $\hat{m}$, can be derived to serve as an algorithm that can predict an unknown response, β_{w+1}, for a new challenge, α_{w+1} [9]. As a result, most often only strong PUFs are susceptible to modeling attacks, with exceptions for weak PUFs when part of their obfuscation relies on the interaction with a strong PUF [12] [20]. In general, weak PUFs can only support a small number of CRPs. Hence, they have very limited uses (such as PUF-based key storage, etc.). Because of the limited number of CRPs, it is difficult to model the behavior of a weak PUF. This is the reason why adversaries target strong PUFs for modeling attacks. Hence, strong PUF-based security solutions are more vulnerable to modeling attacks.

B. Reconfigurable PUF (RPUF) and its Security Against Machine Learning

A "*reconfigurable PUF* (RPUF)" is a strong PUF with a mechanism to change the PUF configuration or settings after each session of the authentication process. The idea of RPUFs stems from the literature surrounding the resetting of a PUF's configuration [14]. Reconfiguration describes a feature of a PUF that enables a complete change of individual behavior in response to challenges, by updating its state. RPUFs can achieve both *forward- and backward-unpredictability*: the former assures that responses measured before the reconfiguration event are invalid thereafter, while the latter assures that an adversary with access to a reconfigured PUF cannot estimate the PUF behavior before the reconfiguration. Assuming that an adversary needs to collect a large subset of a strong PUF's possible CRPs in order to mount a successful modeling attack, the reconfiguration of an individual PUF would render such attacks useless. An adversary would then need to collect a new subset of CRPs for the new configuration in order to generate a new mathematical model for the attack. Additionally, it is assumed that the outcome of the reconfiguration mechanism is uncontrollable and difficult to revert, even with invasive means. Also, after reconfiguration, the configuration of the PUF does not affect the security properties of the original PUF (such as tamper detection, unclonability, etc.). However, after each new configuration, a RPUF behaves as a new PUF. This makes the attacker's job to create a new mathematical model more difficult. For instance, if we use DRAM PUF [14] as a RPUF in authentication protocol, then the authentication system will utilize a different DRAM block for each round of authentication. An attacker cannot gain any insight into the challenge-response behavior of the new block based on the knowledge of challenge-response pairs of previous blocks. This is due to the unique, random bit-flip entropy across different blocks in a DRAM.

C. Noise in PUFs and Fractional Hamming Distance (FHD)

The output of a PUF may exhibit instability due to variations in operating conditions such as temperature and supply voltage, as well as ageing. The native instability (unstable

bits based on the PUF's raw output without any correction) typically range from 1-6 % when tested over a range of operating conditions [31]. Such instabilities can be easily corrected through the use of error correction codes (ECC) [32] or temporal majority voting (TMV) [33]. However, the use of ECC or TMV comes with additional power consumption by the chip and on-chip real estate. Since PUFs are inherently noisy, for a given challenge α, the PUF output β may differ slightly when it is measured multiple times. This gap can be expressed by the Hamming distance, which is a popular metric used in error correction of noisy outputs from PUFs. For a given string of fixed length, the Hamming distance computes the difference between it and another string of the same length by measuring the number of substitutions required to change the given string into the other. Two identical strings would thus have a Hamming distance of zero. Now, considering binary vectors χ and $\tilde{\chi}$ of the same length, the Hamming weight HW(χ) counts the number of non-zero digits (i.e., 1's) in the vector χ, and the fractional Hamming distance is given by $\mathrm{FHD}(\chi, \tilde{\chi}) = \mathrm{HW}(\chi \oplus \tilde{\chi})/\mathtt{L}(\chi)$, where $\mathtt{L}(\chi)$ denotes the length of χ.

III. Proposed Scheme

This section introduces the proposed scheme. The system model in this paper considers a scenario where each home is equipped with a PUF-enabled, resource-constrained smart meter for tasks related to the management of power consumption, including data collection, data transmission, and reception of pricing and other information from the utility. The smart meters need to send their reading/consumption data to a utility server via a communication gateway (router). This communication needs to be secured through an authenticated key exchange scheme. Here, apart from the communication security (such as confidentiality and integrity of the consumption data), the server is also concerned about any physical tampering of the smart meter. The purpose of the proposed authenticated key exchange scheme is to ensure both communication and physical security with resilience against any modeling or ML-attack. The proposed scheme consists of two phases: setup phase followed by the authentication and key exchange phase.

A. Adversary Model

In the proposed RPUF-based authentication scheme, we consider the following capabilities of the adversary. First, in the proposed scheme, we allow an adversary to eavesdrop on the communication channel between the server and the smart meter. He/she may also change and block some of the messages sent between the two entities. Additionally, we also allow the adversary to mount physical and cloning attacks on the PUF. Next, the proposed scheme considers the added ability of the adversary to attempt machine learning modeling attacks (as described in Section II.A). In this regard, we allow an adversary to repeatedly access the smart meter and try to obtain a considerable numbers of CRPs in order to model the behavior of the strong RPUF, and subsequently manipulate the meter reading (e.g., providing high consumption date) by impersonating as a legitimate device (smart-meter). Our adversary model also considers insider attacks where the adversary (a malicious consumer) who has access to the RPUF attached with the smart meter (installed at his/her home) and after collecting enough CRPs, may try to model the behavior of the RPUF and then manipulate the meter reading to cheat during billing. Details of the adversary model are provided in Section IV.

B. Assumptions

In the proposed scheme, we assume that any action taken within the setup (aka enrollment) phase are inaccessible to the adversaries. Therefore, adversaries attempt their attacks during the authentication phase. During the execution of the authentication phase, we assume that an adversary has obtained physical access to the smart meter with a PUF and can thus access the PUF's interface. The adversary can then brute force query the PUF with arbitrary challenges and monitor the responses. This allows the adversary to compile its own CRP data set in order to train a machine learning model to attempt a modeling attack. Now, we use two PUFs in the proposed scheme: a weak PUF (WPUF) (such as SRAM) attached with the main control circuit and the secure NVM (non volatile memory) of the smart meter, and a RPUF (which is basically a strong PUF with reconfigurable property, e.g., D-PUF [14]) attached with the device's memory, where metering data are stored. The WPUF is used to securely generate the round-key k_i and also to protect the main control circuit and the secure NVM from any physical tampering. Here, we assume that the WPUF and secure NVM are embedded within a system on chip (SoC) and are physically inaccessible to the adversary. However, the adversary has physical access to the RPUF to obtain a considerable number of CRPs. Since a RPUF is a strong PUF, the adversary may collect a considerable number of CRPs and attempt to model its behavior and break the forward- and backward-unpredictability properties of the RPUF. However, any attempt to tamper with the PUFs reflects on the behavior of the device and that will render the device useless. Finally, we also assume that the database of the server is fully secure and inaccessible to any adversary.

C. Setup Phase

Our setup phase consists of the following steps:

Step Set_0**:** The server randomly generates two challenges $\{\alpha_i, \alpha_x\}$ and sends them to the smart meter SM through a secure channel.

Step Set_1**:** After receiving the challenges $\{\alpha_i, \alpha_x\}$, the smart meter generates a unique random key k_{i-1} and then the device uses its weak PUF (WPUF_M) and computes $\beta_x = \mathrm{WPUF}_M(\alpha_x)$, along with key for the *i*-th round of authentication, i.e., $k_i = h(\beta_x || k_{i-1})$. After that, the device uses its strong reconfigurable PUF (RPUF^i_M) and extracts the PUF output $\beta_i = \mathrm{RPUF}^i_M(\alpha_i)$, where β_i can be divided into two-parts, i.e., $\{\beta_i^1 || \beta_i^2\}$. Hereafter, the device composes a message with the parameters $\{ID_{SM}, \alpha_i, k_i, \beta_x\}$ and sends it to the server through the secure channel, where ID_{SM} denotes the real identity of the smart meter.

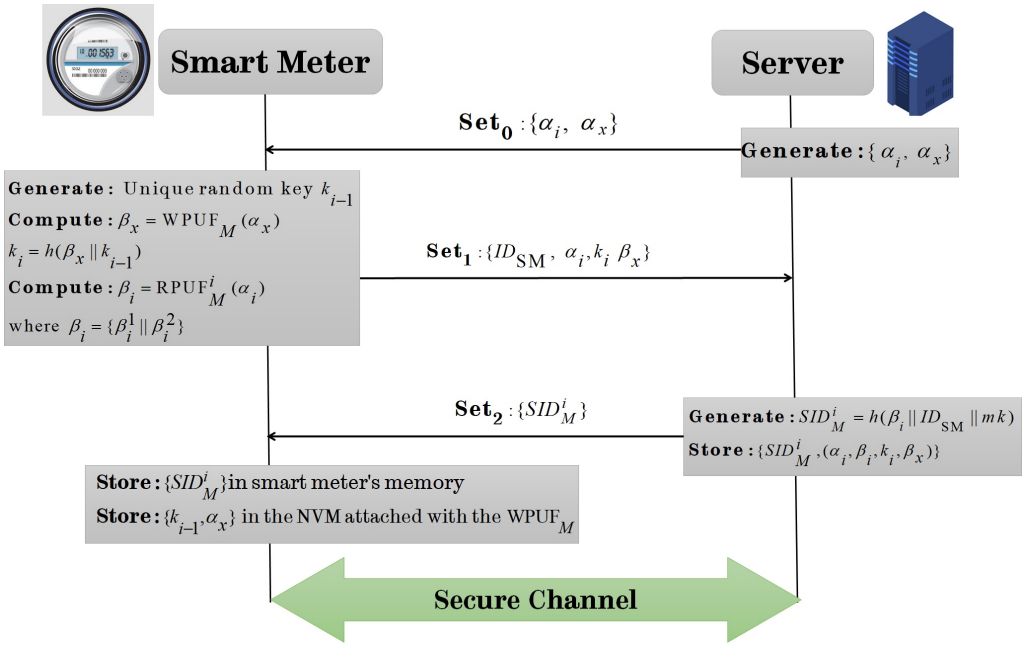

Figure 1. Setup phase of the proposed reconfigurable authentication scheme.

Step Set_2**:** Next, the server uses its master key (mk) and computes the one-time-shadow-identity $SID^i_M = h(\beta_i||ID_{SM}||mk)$ for the i-th round of authentication and sends SID^i_M to the smart meter SM through the secure channel. Then, the server stores $\{SID^i_M, (\alpha_i, \beta_i, k_i, \beta_x)\}$ for authenticating the smart meter in the i-th round.

Step Set_3**:** After receiving the one-time-shadow-identity SID^i_M, the smart meter stores $\{SID^i_M\}$ in its memory and also stores $\{k_{i-1}, \alpha_x\}$ in its secure NVM which is attached with the WPUF_M. Details of the setup phase of the proposed scheme are also depicted in Fig. 1.

D. Authentication and Key Exchange Phase

The i-th round of the authentication phase of the proposed scheme consists of the following steps:

Step 1: The smart meter SM first selects its one-time-shadow-identity SID^i_M and also generates a random number n_x. After that, the smart meter loads $\{k_{i-1}, \alpha_x\}$ into its memory from the NVM and computes $\beta_x = \text{WPUF}_M(C_x)$, $k_i = h(\beta_x||k_{i-1})$, $n^*_x = n_x \oplus k_i$, and $\lambda_0 = h(n^*_x||k_i)$. Finally, the smart meter composes a message $\text{MSG}_1 : \{SID^i_M, n^*_x, \lambda_0\}$ and sends it to the server for verification.

Step 2: Upon receiving the authentication request message MSG_1 from the smart meter, the server first finds SID^i_M in its database. If the server cannot find SID^i_M in its database then it aborts the authentication process. Otherwise, the server reads $(\alpha_i, \beta_i, k_i, \beta_x)$ and computes and verifies the hash response λ_0. If the verification is successful, then the server generates a random number n_s and computes $n_x = n^*_x \oplus k_i$, $n^*_s = k_i \oplus n_s$, $\beta^{1*}_i = \beta^1_i \oplus k_i$, and $\lambda_1 = h(n_s||k_i||\beta^{1*}_i||n_x)$. Finally, the server composes a message $\text{MSG}_2 : \{\alpha_i, \beta^{1*}_i, \lambda_1, n^*_s\}$ and sends it to the smart meter.

Step 3: Upon receiving message MSG_2, the smart meter first computes $n_s = k_i \oplus n^*_s$ and $\beta^1_i = \beta^{1*}_i \oplus k_i$, and verifies the parameter λ_1. If the verification is not successful, then the smart meter aborts the execution of the protocol. Otherwise, the smart meter generates $\{\beta^{1\dagger}_i||\beta^{2\dagger}_i\} = \text{RPUF}^i_M(\alpha_i)$ and also checks whether $\text{FHD}(\beta^{1\dagger}_i, \beta^1_i) > \tau$. If so, the smart meter terminates the execution of the protocol. Otherwise, the smart meter computes $X = \beta^{2\dagger}_i \oplus k_i$ and $\alpha_{i+1} = h(\alpha_i||n_s||n_x)$ and *reconfigures* the strong-PUF for the i+1-th round of authentication. Subsequently, the smart meter extracts the PUF output $\beta_{i+1} = \text{RPUF}^{i+1}_M(\alpha_{i+1})$ and then computes $\beta^*_{i+1} = \beta_{i+1} \oplus k_i$ and $SID^{i+1}_M = h(SID^i_M||\beta_{i+1})$ for the i+1-th round of authentication. Hereafter, the smart meter computes $\lambda_2 = h(k_i||\beta^*_{i+1}||n_s||X)$ and composes a message $\text{MSG}_3 : \{\beta^*_{i+1}, X, \lambda_2\}$ and sends it to the server. Finally, the smart meter generates a session key $SK = h(n_x||n_s||\beta^1_i||\beta^{2\dagger}_i)$, stores SID^{i+1}_M in its memory, and replaces k_i with k_{i+1} in its secure-NVM attached with the WPUF.

Step 4: Next, when the server receives message MSG_3 from the smart meter, it first computes and verifies the parameter λ_2 in order to check the integrity of the other parameters in MSG_3 and also to validate the legitimacy of the smart meter. If the validation is successful, then the server computes $\beta^{2\dagger}_i = X \oplus k_i$ and checks whether $\text{FHD}(\beta^{2\dagger}_i, \beta^2_i) > \tau$. If not, then the server computes $\alpha_{i+1} = h(\alpha_i||n_s||n_x)$, $\beta_{i+1} = \beta^*_{i+1} \oplus k_i$, $SID^{i+1}_M = h(SID^i_T||\beta_{i+1})$ and $k_{i+1} = h(k_i||\beta_x)$ for the i+1-th round of authentication. After that, the server derives the session key $SK = h(n_x||n_s||\beta^1_i||\beta^{2\dagger}_i)$, which will be used for securely communicating with the smart meter. Finally, the server replaces the round-key k_i with k_{i+1}, α_i with α_{i+1}, β_i with β_{i+1}, and SID^i_M with SID^{i+1}_M. Therefore, the server stores $\{SID^{i+1}_M, (\alpha_{i+1}, \beta_{i+1}, k_{i+1}, \beta_x)\}$ for authenticating the smart meter at the i+1-th round. Details of the authentication phase of the proposed scheme are also depicted in Fig. 2.

Remark 1: Here, we consider DoS/desynchronization at-

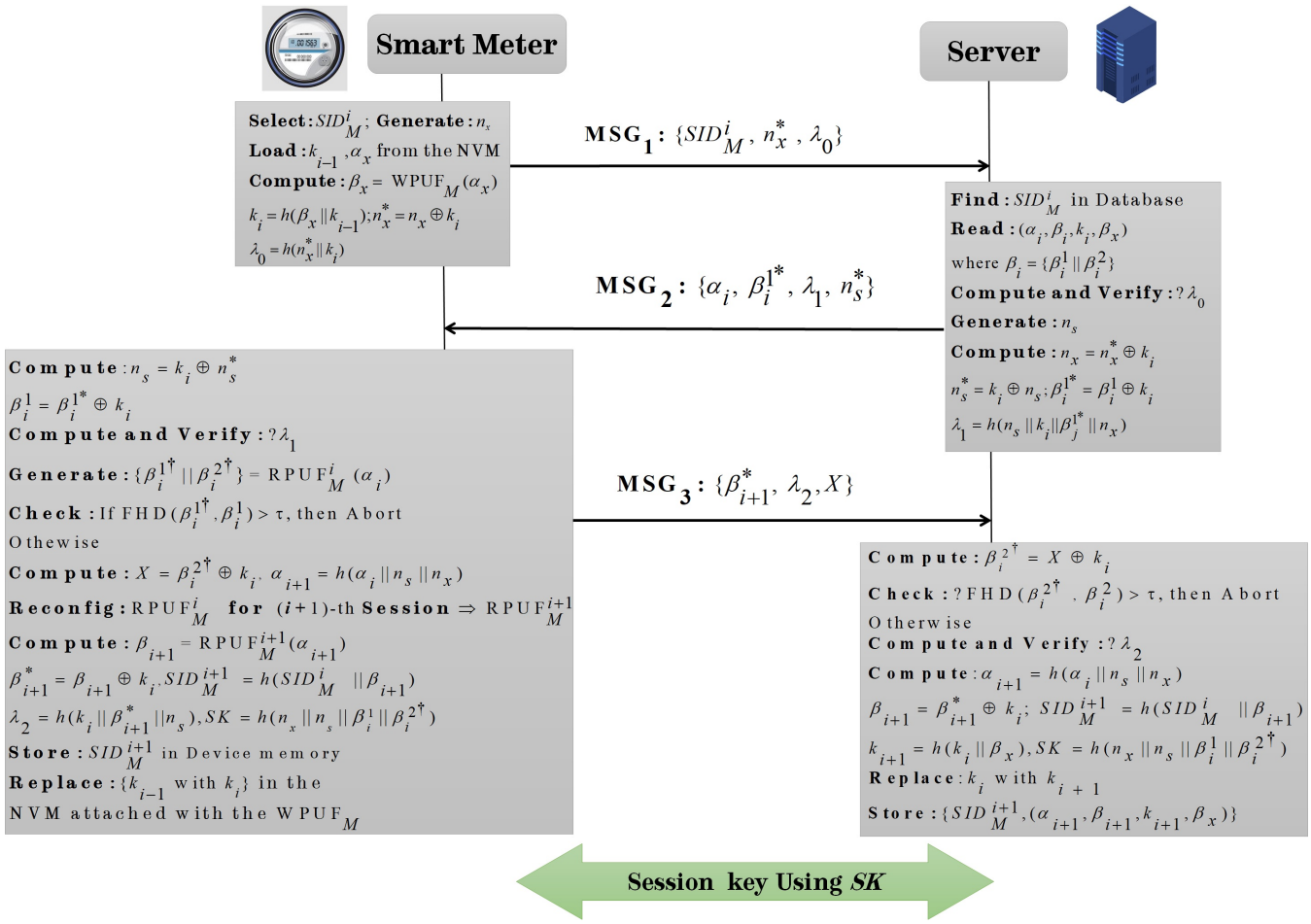

Figure 2. Proposed ML-attack prevention-based authentication scheme.

tacks which may lead to the loss of synchronization between the smart meters and the server. Even though there exist a few solutions in the literature to address this issue, most of these solutions are based on the strategy proposed in [29]. In these approaches based on [29], both the smart meter and the server need to keep previous security credentials (such as the previous round's key k_{i-1} and previous one-time-shadow-identity SID_M^{i-1}). The main problem with this approach is that the attacker will be able to identify the entity (in our case smart meter) because the same security credentials are reused. In order to handle any desynchronization between the smart meter and server without compromising the untraceablity property, we suggest a few enhancements to our previous strategy in [30]. In the proposed strategy, apart from $\{\alpha_i, \alpha_x\}$, the server also needs to generate a set of synchronization (SYN) challenges $\alpha_{SYN} = \{\alpha_{SYN}^1, \cdots, \alpha_{SYN}^n\}$ and send them to the device during the registration phase. After that, in Step Set_1, the smart meter uses the RPUF_M and generates $\beta_{SYN}\{\beta_{SYN}^1, \cdots, B_{SYN}^n\} \leftarrow \text{RPUF}_M(\alpha_{SYN} = \{\alpha_{SYN}^1, \cdots, \alpha_{SYN}^n\})$ and also generates a few pairs of Reference ID and SYN keys, i.e., $(REF_{ID}, K_{SYN}) = \{(REF_{ID}^1, K_{SYN}^1), \cdots, (REF_{ID}^n, K_{SYN}^n)\}$ and then sends $(REF_{ID}, K_{SYN}, \alpha_{SYN}, \beta_{SYN}) = \{(REF_{ID}^1, K_{SYN}^1, \alpha_{SYN}^1, \beta_{SYN}^1), \cdots, (REF_{ID}^n, K_{SYN}^n, \alpha_{SYN}^n, \beta_{SYN}^n)\}$ to the server through the secure channel, also keeps a copy in its secure NVM. Note that for restricting any modeling attack, the device needs to use a new setting of the RPUF_M for generating each $(\alpha_{SYN}^j, \beta_{SYN}^j)$. Now, in case of loss of synchronization, both the server and the smart meter need to use one of the sets $(REF_{ID}^j, K_{SYN}^j, \alpha_{SYN}^j, B_{SYN}^j)$ from $(REF_{ID}, K_{SYN}, \alpha_{SYN}, \beta_{SYN})$.

Remark 2: To ensure security against any key-compromise-impersonation (KCI)-attacks [42], the proposed scheme introduces the concept of round-key instead of using a static long-term secret key. In the proposed scheme, a secure private NVM (such as 2T MTP) stores the round-key of the previous round/state (say k_{i-1} for the (i-1)-th state). In order to form the round-key of a specific round (say i-th round) and prove itself as a legitimate smart meter, an adversary would require the support of the WPUF attached with the main control circuit and the secure NVM. From the assumptions in Section III-B, both the NVM and the WPUF are inaccessible to any adversary. This can be achieved in practice by using programmable fuse technology (such as OTP anti-fuses) to disable all the associated data and control paths. Now, if we assume that an adversary somehow (say by launching physical attack) succeed to get k_{i-1}, then to convert k_{i-1} to k_i, he/she would also require the support of the WPUF, which

is inaccessible to the adversary. Besides, since the WPUF is attached with the main control circuit and the secure NVM, any forcefully attempt to access the NVM will also impact on the PUF-settings of the WPUF. In such cases, the PUF will not be able to generate the desirable output β_x, which is generated on-the-fly (not stored anywhere). On the other hand, like any other secure symmetric-key-based system, we assume that the database of the utility sever, where security credentials (such as the round key k_i and the RPUF output β_i) are stored is secure. To enhance the security level of the proposed protocol, we can also consider that the round key k_i and the RPUF output β_i are stored in two separate databases. In that case, even if one of the database (say the one storing k_i) is compromised, the other can still ensure security, since in our proposed protocol the adversary needs to know both the round key k_i and the RPUF output β_i for *i*-th session to prove its legitimacy.

IV. Formal Security Evaluation

In this section, we formally analyze our proposed reconfigurable anonymous authentication scheme with respect to the major security requirements. In this regard, we first specify the formal adversarial model and some imperative security requirements.

A. Adversarial Model

Consider a server $\mathcal{S}$ that interacts with smart meter $\mathcal{M} = \{M_1, M_2, \cdots, M_n\}$. For initializing each device, the server $\mathcal{S}$ runs a setup algorithm $\mathrm{SetupM}(1^\gamma)$ and generates a public parameter ψ and secret key $\mathcal{K}$. During the execution of the authentication phase, both $\mathcal{S}$ and the devices in $\mathcal{M}$ interact through an insecure channel and try to validate each other. Finally, the parties $(\mathcal{S}, \mathcal{M})$ output 1 (Acceptance) or 0 (Rejection) as the authentication outcome. The sequence of interactions between $\mathcal{S}$ and a device $M \in \mathcal{M}$ can be defined as a session, where a unique session identifier sid is used for distinguishing each session. A session can be claimed as a *matching session* if the messages exchanged between $\mathcal{S}$ and M are honestly transferred until they authenticate each other. For the correctness of the protocol, if the session has a *matching session*, then both the server $\mathcal{S}$ and the device M accept the session. In this section, we consider security against the *man-in-the-middle attack*, which is the canonical security level for any authentication protocol. In this regard, the ability of an attacker is modeled by letting the attacker to control all the communication between a smart meter and the service provider. Here, the attacker is allowed to modify messages between a smart meter and the service provider. The authentication outputs for both parties becomes 1 if and only if the communication messages are honestly transferred. In addition to the canonical security requirement for the *man-in-the-middle attack*, in our model we allow the adversary to obtain the memory contents in the non-volatile memory before and after the session (authentication).

More formally, now we consider the following security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ against a mutual authentication protocol $\mathcal{P}$:

$\mathrm{Expr}^{\mathrm{Sec}}_{\mathcal{P},\mathcal{A}}(\gamma)$:

1) $(\psi, \mathcal{K}) \xleftarrow{Random} \mathrm{Setup}(1^\gamma)$;
2) $(\mathrm{sid}^*, \mathcal{M}) \xleftarrow{Random} \mathcal{A}_1^{\mathrm{Launch}, \mathrm{Send}\mathcal{S}, \mathrm{Send}\mathcal{M}, \mathrm{Outcome}, \mathrm{Reveal}}(\psi, \mathcal{S}, \mathcal{M})$;
3) $\Phi := \mathrm{Outcome}(\mathrm{sid}^*, \varpi)$;
4) Output Φ.

After receiving $(\psi, \mathcal{S}, \mathcal{M})$ the adversary $\mathcal{A}$ can issue the following oracle queries :

-$\mathrm{Launch}(1^\gamma)$: Initiate the server $\mathcal{S}$ to launch a new session.

-$\mathrm{Send}\mathcal{S}(m)$: An arbitrary message m is sent to $\mathcal{S}$.

-$\mathrm{Send}\mathcal{M}(M_j, m)$: An arbitrary message m is sent to the device $M_j \in \mathcal{M}$.

-$\mathrm{Result}(\varpi, \mathrm{sid})$: Output whether the sid of ϖ is accepted or not where $\varpi \in \{\mathcal{S}, \mathcal{M}\}$.

-$\mathrm{Reveal}(M_j)$: Output all the information stored in the memory of the smart meter M_j.

The advantage of the adversary $\mathcal{A}$ against $\mathcal{P}$, $\mathrm{Advr}^{\mathrm{Sec}}_{\mathcal{P},\mathcal{A}}(\gamma)$, can be defined as the probability that $\mathrm{Expr}^{\mathrm{Sec}}_{\mathcal{P},\mathcal{A}}(\gamma)$ outputs 1 when sid^* of ϖ has no matching session.

Definition 1. *An authentication protocol $\mathcal{P}$ is said to be secure against man-in-the-middle attacks and impersonation attacks with key compromise if for any probabilistic polynomial time adversary $\mathcal{A}$, $\mathrm{Advr}^{\mathrm{Sec}}_{\mathcal{P},\mathcal{A}}(\gamma)$ is negligible in γ (for large enough γ).*

B. Unpredictability Property of a RPUF for Defending Against Modeling Attacks

Now we consider a game played between the adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to define the unpredictability behavior of a RPUF, which is a desirable imperative property to ensure security against modeling attacks.

Setup: Challenger $\mathcal{C}$ issues a RPUF to adversary $\mathcal{A}$.

Queries: $\mathcal{A}$ queries the RPUF Φ times using challenges α_i (where $1 \leq i \leq \Phi$) and receives the PUF output β_i ($\beta_i \leftarrow RPUF_M(\alpha_i)$).

Output: At the end of the game, $\mathcal{A}$ outputs a CRP pair (α^*, β^*).

We say $\mathcal{A}$ wins the game if he/she can output a valid PUF response β^*. Otherwise, the behavior of the PUF is unpredictable and no polynomial adversary can predict the PUF output with significant success probability.

Definition 2. *A PUF is said to be (q, ϵ)-unpredictable if there is no ppt (probabilistic polynomial time) adversary $\mathcal{A}$ that issues at most q queries to the RPUF and can win the game with probability greater than ϵ.*

Backward and Forward Unpredictability: Next, we define backward- and forward-unpredictability of a RPUF in terms of a two-stage game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In the first stage, $\mathcal{A}$ is given oracle access (i.e., access to the interface) of the RPUF, from which $\mathcal{A}$ can obtain challenge/response pairs (CRPs) at will. This stage models the ability of $\mathcal{A}$ to obtain challenges and responses (with respect to a fixed internal RPUF state) by passive eavesdropping. We also give $\mathcal{A}$ access to the internal RPUF state in order to model hardware attacks against the RPUF implementation.

Once $\mathcal{A}$ has learned enough CRPs, the challenger performs the reconfiguration operation and finally gives $\mathcal{A}$ oracle access to the reconfigured RPUF such that $\mathcal{A}$ can obtain CRPs of the reconfigured RPUF. At the end of the game, $\mathcal{A}$ needs to output a non-trivial CRP $(\alpha^{\#}, \beta^{\#})$.

More formally, $\mathcal{A} = ((\mathcal{A}_{\S}, \mathcal{A}_{\dagger}))$ consists of two probabilistic polynomial time algorithms, where $\mathcal{A}_{\S}$ interacts with the LR-PUF before reconfiguration and $\mathcal{A}_{\dagger}$ thereafter. A engages in the following experiment:

Setup: The adversary $\mathcal{A} = (\mathcal{A}_{\S}, \mathcal{A}_{\dagger})$ is given an arbitrary state ς of the RPUF by the challenger $\mathcal{C}$ who sets up an RPUF. Then, in Phase 1, $\mathcal{A}_{\S}$ queries the RPUF up to q_x times and at the end of this phase, $\mathcal{A}_{\S}$ stops and outputs to the log file $\mathcal{F}$ that is used as input to $\mathcal{A}_{\dagger}$.

Reconfiguration: Now, the challenger $\mathcal{C}$ resets the RPUF, which updates its internal state to ς^*. Then, in Phase II, $\mathcal{A}_{\dagger}$ is initialized with state ς^* and the log file $\mathcal{F}$ from $\mathcal{A}_{\S}$. Now, $\mathcal{A}_{\dagger}$ is allowed to query the reset RPUF up to q_y times.

Outputs: At the end of the game, $\mathcal{A}_{\dagger}$ outputs a non-trivial CRP $(\alpha^{\#}, \beta^{\#})$ of the RPUF.

We say that $\mathcal{A}$ wins the *forward-unpredictability* game if $\beta^{\#}$ is a valid RPUF response to query $\alpha^{\#}$ that was not included in the q_y queries. Therefore, with this unpredictability, once the RPUF is reset, the adversary cannot output a valid CRP for the reset RPUF. On the other hand, we say that $\mathcal{A}$ wins the *backward-unpredictability* game if $\beta^{\#}$ is a non-trivial (valid) RPUF output to the query $\alpha^{\#}$ that was not part of the q_x queries. This unpredictability implies that an adversary with access to the RPUF will not be able to predict a valid response of the RPUF *before the reset* happened. Accordingly, a RPUF is *backward (or forward) unpredictable*, when there is no PPT adversary $\mathcal{A}$ that can win the game with significant success probability.

Definition 3. *A RPUF is said to be a (q_x, q_y, ε)-secure backward and forward unpredictable PUF if there is no PPT adversary $\mathcal{A}$ who makes at most q_x queries in Phase 1 and at most q_y queries in Phase II, is able to win the above backward and forward unpredictability game with probability greater than ε.*

C. Tamper-Resilience Property of PUF for Defending Against Physical Attacks

One of the main objectives for employing PUFs in a security solution is to ensure resilience against physical attacks with lower cost as compared to other measures like using a Trusted Platform Module (TPM). In these solutions, it is considered that any physical attack against the PUF will not leak any information about the internal structure of the device. To prove this assertion, we now formalize a tamper-resilience game between an adversary $\mathcal{A}$ and a simulator $\mathfrak{S}$. Initially, $\mathfrak{S}$ selects a manufacturing process $\mathcal{MP}$ and initial parameter Ω and sends $(1^{\gamma}, \mathcal{MP}, \Omega)$ to the adversary $\mathcal{A}$. Next, $\mathcal{A}$ can issue $(\mathbf{Create}, \mathbf{Response})$ queries, where **Create** is polynomially bounded in γ and we denote the upper bound by n. Similarly, the **Response** query can be issued by $\mathcal{A}$ to obtain the PUF's response and is also polynomially bounded. In this regard, whenever **Create**(Ω) is launched, $\mathcal{A}$ receives the the produced PUF, $\mathcal{PUF}_i$, and can analyze it physically. Here, $\mathcal{A}$ is allowed to mount arbitrary physical attacks on the PUF (e.g., power analysis, probing attack, etc.). However, the simulator algorithm $\mathfrak{S}$ can only adaptively issue $(\mathbf{Create}, \mathbf{Response})$ queries and does not get physical access to the created PUFs. Next, both the adversary and the simulator output the internal state $\mathfrak{st}$. Here, the main idea is that for any adversary $\mathcal{A}$ who has physical access to a PUF (e.g., WPUF and/or RPUF), there exists a simulator $\mathfrak{S}$ whose behaviour is practically the same but without physical access. As a result, physical access does not provide any advantage. In this case, we say that the PUF is tamper resilient. The advantage of $\mathcal{A}$ in the above experiment is defined by

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A},\mathfrak{S},\mathfrak{D}}{}^{\mathrm{Tamp}}(\gamma) := |Pr\left[\mathfrak{D}(1^{\gamma}, \mathfrak{st}) \rightarrow 1 | \mathfrak{st} \leftarrow \right. \\ \left. \mathcal{A}^{\mathrm{Create,Response}}(1^{\gamma}, \mathcal{MP}, \Omega, \mathcal{PUF}_1, \mathcal{PUF}_2, \cdots)\right]| \\ - \left|Pr\left[\mathfrak{D}(1^{\gamma}, \mathfrak{st}) \rightarrow 1 | \mathfrak{st} \leftarrow \mathcal{A}^{\mathrm{Create,Response}}(1^{\gamma}, \mathcal{MP}, \Omega)\right]\right|\end{aligned}$$

In the equation above, $\mathfrak{D}$ represents a distinguisher who tries to distinguish whether $\mathfrak{st}$ is generated by $\mathcal{A}$ or $\mathfrak{S}$.

Definition 4. A PUF, $\mathcal{PUF}_i(\mathcal{MP}, \Omega, \epsilon)$, is considered as a tamper resilient one-way function if for any PPT algorithm $\mathfrak{S}$ and PPT distinguisher $\mathfrak{D}$, there is no PPT adversary $\mathcal{A}$ that can achieve an advantage $\mathrm{Adv}^{\mathrm{Tamp}}_{\mathcal{A},\mathfrak{S},\mathfrak{D}}(\gamma) > \epsilon(\gamma)$.

As discussed above, the game is structured such that the adversary $\mathcal{A}$ actually receives physical access to the PUFs and can thus conduct different actions on them, such as, observe the structure of the chip and gate-delay, and launch arbitrary side-channel analysis. The tamper-resilience of $\mathcal{PUF}_i$ assures there is no extra information leaked about its parameters by physical attacks. These results can be represented through the output internal state $\mathfrak{st}$. Next, the distinguisher $\mathfrak{D}$ tries to distinguish whether $\mathfrak{st}$ is output from $\mathcal{A}$ or $\mathfrak{S}$. Therefore, if $\mathfrak{D}$ cannot distinguish between $\mathcal{A}$'s output and $\mathfrak{S}$'s output, this means that no additional information which is not trivially derived from challenge-response pairs is extracted by the physical attack (regardless of what they are).

D. Security Analysis

In this section, we provide security and privacy proofs for the proposed authentication scheme by considering the above security and privacy models.

Theorem 1 (Security): *Consider a RPUF instance $\mathcal{RPUF}^* \leftarrow \mathrm{RPUF}$ that is a $(q_x, q_y, \varepsilon_1)$-secure backward and forward unpredictable PUF, and let $h(\cdot)$ be an ε_2-secure collision resistant pseudo random function. Then, the proposed protocol $\mathcal{P}$ is secure against man-in-the-middle attacks under the memory-compromise assumption.*

Proof. The main objective of the adversary is to violate the security game and convince either the server $\mathcal{S}$ or a smart meter $M_j \in \mathcal{M}$ to accept a session without being the matching session. We utilize the security game approach to prove the security of our protocol based on gradually replacing the communications in the protocol with random strings. If the adversary is able to distinguish between the instances of real execution and that of the random string, and modifies the execution such that either the meter or the server accepts the

non-matching session, then the adversary wins the game. Now we consider the following game transformations. Let $\mathcal{G}_i$ be the advantage that the adversary wins the game in Game i.

- **Game 0:** This represents the main game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$, without any modification to the protocol.
- **Game 1:** In this game, we evaluate and alter the parameters of the RPUF in a session between the server $\mathcal{S}$ and a smart meter $M_j \in \mathcal{M}$. The challenger $\mathcal{C}$ evaluates the output of the RPUF in M_j. Since the RPUF is a $(q_x, q_y, \varepsilon_1)$-secure backward and forward unpredictable PUF, it implies that the output from the RPUF satisfies the min-entropy property such that each output is uncorrelated. Based on this assumption, even if an adversary issues the Reveal query and obtains the stored information from the RPUF's memory, the output will not be correlated. This implies that there is no effect on the game transformation from Game 1. Now, if adversary $\mathcal{A}$ is unable to impersonate as M_j to the server, $\mathcal{C}$ aborts the game.
- **Game 2:** Now, assume that the attacker can establish at most l sessions in the game. For $1 \leq m \leq l$, we evaluate or alter the variables related to the session between smart meter M_j and the server $\mathcal{S}$ in various sessions as the following games.
  - **Game 2-*m*-1:** At the l-th session, $\mathcal{C}$ evaluates the output of the RPUF implemented in M_j. $\mathcal{C}$ aborts the game if the output does not have enough entropy or if it is correlated to the other outputs derived from the inputs to the RPUF.
  - **Game 2-*m*-2:** Challenger $\mathcal{C}$ replaces the outputs of the pseudorandom function $h(\cdot)$ that provides entity authentication, with random strings of the same size.
  - **Game 2-*m*-3:** Challenger $\mathcal{C}$ replaces the pseudorandom string of $n_s^* = n_s \oplus k_i$ and $\beta_i^{1*} = k_i \oplus \beta_i^1$ with randomly generated strings of the same size.
  - **Game 2-*m*-4:** Challenger $\mathcal{C}$ replaces the pseudorandom string of $\{\beta_{i+1}^*, X, \lambda_2\}$ with randomly generated strings of the same size. Since the adversary has no access to k_i (derived from the RPUF as a pseudorandom output), the adversary will not be able to distinguish between these strings and truly random strings.

Here, we will modify the messages corresponding to the smart meter M_j. We say that the attacker wins the game if he/she can distinguish the random strings from real messages/outputs. We proceed with the game transformation starting with the first call of meter M_j. After that, we gradually change the communication messages from Game 2-m-1 to Game 2-m-4. Once these transformations are finished, we move to the next section. Through these game transformations, we show that the advantage of the adversary against the authentication protocol can be limited to negligible values as shown in the results of Lemma 1 through 5.

Lemma 1: *If the numbers of smart meters is n, then we can write $\mathcal{G}_0 = n\mathcal{G}_1$.*

Proof. Since there are n devices, $\mathcal{C}$ can correctly guess the related session with probability $1/n$. ■

Lemma 2: *If RPUF$_{\mathcal{M}_j}$ is a $(q_x, q_y, \varepsilon_1)$-secure backward and forward unpredictable PUF, then $\mathcal{G}_1 = \mathcal{G}_{2-m-1}$ and $\mathcal{G}_{2-(m-1)-4} = \mathcal{G}_{2-m-1}$, for any $2 \leq m \leq l$.*

Proof: The RPUF attached with device M_j is a $(q_x, q_y, \varepsilon_1)$-secure backward and forward unpredictable PUF and the min-entropy of the PUF is larger than χ. In addition, the PUF also has the property that even if the input to the PUF is exposed, the output derived from the input maintains sufficient min-entropy property and the outputs are thus uncorrelated. Now, if an adversary issues the Reveal query and obtains the stored information from the RPUF's memory, then, since the games in $\mathcal{G}_1$, $\mathcal{G}_{2-m-1}$ and $\mathcal{G}_{2-(m-1)-4}$ are based on the above condition, the gap between them is bounded by ε_1. Therefore, we can write $|\mathcal{G}_1 - \mathcal{G}_{2-m-1}| \leq \varepsilon_1$ and $\left|\mathcal{G}_{2-(m-1)-4} - \mathcal{G}_{2-m-1}\right| \leq \varepsilon_1$. This means that there is no effect of the game transformations. ■

Lemma 3: *Let $\mathrm{Advr}^{\mathrm{PRF}}_{h(\cdot),\mathcal{B}}(k)$ denote the advantage of $\mathcal{B}$ to break the security of the PRF $h(\cdot)$. Then, $\forall 1 \leq m \leq l$, we have $|\mathcal{G}_{2-m-1} - \mathcal{G}_{2-m-2}| \leq \mathrm{Advr}^{\mathrm{PRF}}_{h(\cdot),\mathcal{B}}(k)$.*

Proof: Now, consider that an algorithm $\mathcal{B}$ is constructed which breaks the security of the PRF $h(\cdot)$. $\mathcal{B}$ sets up all the security credentials and simulates our protocol except for the i-th session (the current session). $\mathcal{B}$ can access the real PRF $h(\cdot)$ or a truly random function. When the adversary invokes the i-th session, $\mathcal{B}$ sends the uniformly distributed random challenge $\{n_s^* \underset{\leftarrow}{\mathrm{U}} \{0,1\}^k\}$ as the output of the server. When $\mathcal{A}$ sends $n_s^{\#}$ to the device, $\mathcal{B}$ continues the computations as per the protocol specification and issues $n_s^{\#}$ to the oracle instead of the normal computation of $h(\cdot)$. After receiving MSG_2, $\mathcal{B}$ outputs $\mathrm{MSG}_3 : \{\beta_{i+1}^*, X, \lambda_2\}$ as the response of the smart meter. When the adversary sends $\{\beta_{i+1}^{\#}, \lambda_2^{\#}. X^{\#}\}$, $\mathcal{B}$ issues $n_s^{\#}$ to the oracle and obtains $\lambda_2^{\#}$, which is used to authenticate the smart meter.

If $\mathcal{B}$ accesses the real PRF, this simulation is equivalent to Game 2-m-1. Otherwise, the oracle query issued by $\mathcal{B}$ is completely random, and its distribution is equivalent to that in Game 2-m-2. Therefore, we can write $|\mathcal{G}_{2-m-1} - \mathcal{G}_{2-m-2}| \leq \mathrm{Advr}^{\mathrm{PRF}}_{h(\cdot),\mathcal{B}}$. ■

Lemma 4: $\forall 1 \leq m \leq l$, $|\mathcal{G}_{2-m-2} - \mathcal{G}_{2-m-3}| \leq \mathrm{Advr}^{\mathrm{PRF}}_{h(\cdot),\mathcal{B}}(k)$.

Proof: The proof for this lemma follows along the lines of the proof for Lemma 3. ■

Lemma 5: $\forall 1 \leq m \leq l$, *we have* $\mathcal{G}_{2-m-2} = \mathcal{G}_{2-m-3} = \mathcal{G}_{2-m-4}$.

Proof: In the three games considered in this lemma, the RPUF and PRF $h(\cdot)$ are changed to a truly random function. Therefore, in the i-th round of authentication, k_i, $n_s^* = k_i \oplus n_s$, $\beta_i^{1*} = \beta_i^1 \oplus k_i$, $\lambda_1 = h(n_s||k_i||\beta_i^{1*}||n_x)$, $n_x^* = n_x \oplus k_i$, $\lambda_0 = h(n_x^*||k_i)$, $\beta_{i+1}^* = \beta_{i+1} \oplus k_i$, and $\lambda_2 = h(k_i||\beta_{i+1}^*||n_s||X)$ are effectively used as one-time pads. Therefore, no adversary can differentiate these parameters from a randomly chosen string. ■

Theorem 2: *Consider an RPUF instance $\mathcal{RPUF}^* \leftarrow \mathrm{RPUF}$ that is a $(q_x, q_y, \varepsilon_1)$-secure backward and forward unpredictable PUF and let $h(\cdot)$ be an ε_2-secure collision resistant PRF. Then, our protocol $\mathcal{P}$ satisfies the*

indistinguishability-based privacy property.

Proof: The proof for this theorem is similar to that of Theorem 1, where we have proved that the proposed authentication protocol holds security against *man-in-the-middle attacks.* Now, based on the game transformation illustrated in the proof of Theorem 1, if we continuously modify the communication messages for meters M_0^* and M_1^*, then the whole transcript will be similar to a random string. Thus, no information that identifies the challenger's coin will be leaked. Recall that all the identity related parameters stored in the memory such as the shadow id SID_M^i, and the reference identities $REF_{ID} = \{(REF_{ID}^1, \cdots, REF_{ID}^n)\}$ are randomly generated and each pair is restricted to be used only once. Hence, the probability that the challenger can identify M_0^* and M_1^* such that the game transformation is finished within a polynomial time is $1/n^2$, where n is the number of devices in the network. In other words, no adversary can distinguish between the messages from the smart meters M_0^* and M_1^* with probability greater than $1/n^2$. Thus, we can claim that the proposed scheme can ensure indistinguishability-based privacy. ■

Theorem 3: *The proposed reconfigureable authentication scheme can ensure forward and backward unpredictability along with security against ML-attacks under the assumption of a semi-honest authentication server $\mathfrak{S}$, if the $RPUP$ is a (Φ, ϵ)-unpredictable PUF and h is a secure one-way collision resistant hash function.*

Proof. In order to prove this theorem, we use the above *backward-and forward unpredictability game*, where an adversary $\mathcal{A}$ is allowed to access of the RPUF attached with the device and to obtain a set of CRPs from that. Assume that $\mathcal{A} = (\mathcal{A}_{\S}, \mathcal{A}_{\dagger})$ breaks the *backward-and forward unpredictability* of the RPUF with non-negligible probability. We now construct an adversary $\mathfrak{B}$ that breaks the unpredictability of the underlying physical RPUF with the same success probability as $\mathcal{A}$. $\mathfrak{B}$ selects an arbitrary state ς of the RPUF, then passes it to $\mathcal{A}_{\S}$ and executes a black-box simulation of the challenger $\mathcal{C}$ of the *backward-and forward unpredictability game* (shown in Theorem 1). Now, for a challenge α_j received from $\mathcal{A}_{\S}$, $\mathfrak{B}$ queries the RPUF and stores (α_j, β_j) in a log file $\mathcal{F}$ and forwards β_j to $\mathcal{A}_{\S}$. At some point, $\mathcal{A}_{\S}$ stops and outputs some log file $\mathcal{F}^*$. After that, $\mathfrak{B}$ changes the RPUF state to ς^* for resetting the configuration of the RPUF. Next, $\mathfrak{B}$ initializes $\mathcal{A}_{\dagger}$ with state ς^* and log file $\mathcal{F}^*$ and continues to simulate $\mathcal{C}$. Now, when $\mathcal{A}_{\dagger}$ sends a challenge α_j, $\mathfrak{B}$ queries the RPUF and stores (α_j, β_j) in a log file $\mathcal{F}$ and forwards β_j to $\mathcal{A}_{\dagger}$. Finally, $\mathcal{A}_{\dagger}$ stops and outputs a CRP $(\alpha^{\#}, \beta^{\#})$ of the RPUF. Since, $\mathfrak{B}$ has never queried $\alpha^{\#}$ to the RPUF, this contradicts the unpredictability property of the RPUF. Hence, the success probability of $\mathfrak{B}$ is similar as $\mathcal{A}$. Now, as mentioned before, the security of the proposed scheme against any ML-attacks is based the unpredictability property of the RPUF, where an adversary should not be able to predict any PUF response for a given challenge. Therefore, no adversary can differentiate the encoded RPUF outputs such as $X = \beta_i^{2\dagger} \oplus k_i$ and $\beta_{i+1}^* = \beta_{i+1} \oplus k_i$ from a randomly chosen string. Hence, our proposed authentication scheme is secure against any modeling attacks. ■

V. Discussion

This section presents a comparison of the security properties of the proposed scheme with other recently proposed relevant schemes in [3], [4], [8], [10], [11], [25], [26], [27], [28] and [41]. From Table III we can see that the schemes presented in [8], [25], [26], [28] and [41] cannot ensure some of the desirable security properties such as privacy of the smart meter, session-key security, etc. For instance, in the protocols presented in [8], [25], [26], [28] and [41], a smart meter reveals its identity during the authentication process (in Step 1). Therefore, these protocols cannot ensure the privacy of the smart meter. On the other hand, even though the schemes presented in [11] and [28] can ensure security against physical tampering of a smart meter, they cannot ensure security against impersonation attacks. In this context, an attacker (including a malicious consumer) with access to the smart meter will be able to model the PUF after collecting sufficient number of CRPs. Then, if the attacker uses the modeled PUF during the authentication process, the server will not be able to detect that. Besides, in Step 1 of the protocol presented in [28], the smart meters reveal their identity. Hence, the protocol cannot ensure privacy of the smart-meter, eventhough it can can guarantee confidentiality of usage data. In our proposed scheme, each smart meter uses a one-time shadow id SID_M^i to ensure privacy. Next, to ensure security against impersonation attacks and session key security, the proposed scheme uses the PUF-generated one-time secrets k_i, β_i^1, and $\beta_i^{2\dagger}$. In addition, the proposed scheme utilizes the reconfigurability property of the RPUF (e.g., refresh-pause behavior of DRAM PUFs), where after each round of the authentication process, the PUF's configuration is updated. Now, if an adversary $\mathcal{A}$ has access to the PUF-enabled smart meter and is provided with a set of CRPs, it may develop a model for the RPUF. However, the RPUF's behavior is changed after each reconfiguration operation. Hence, it will be difficult for $\mathcal{A}$ to perform any modeling or ML attacks (as shown in [14]).

Next, we evaluate the performance of the proposed scheme with respect to others in terms of the computational cost at the resource limited smart meters. Note that apart from the proposed scheme, only the schemes presented in [11] and [28] have considered physical security, which is important in smart grids in order to ensure the integrity, confidentiality and accountability of the data in the AMI. For example, an inside attacker in a home or a business may try to alter the configuration of a smart meter and subject it to physical attacks in order to cheat in billing. Therefore, we first compare the proposed reconfigurable authentication scheme with the schemes in [11] and [28]. Subsequently, we also show the cost of performing the computationally expensive public-key-based operations in [3], [4], [8], [9], and [25].

To evaluate the performance of the proposed scheme with respect to others in terms of the computational overhead (cost) at the smart meter, we consider a APUF (arbiter PUF for [11]), SRAM-PUF (WPUF of the proposed scheme) and DRAM-PUF (RPUF of the proposed scheme) implementations on a SASEBO-GII board consisting of a Xilinx XC5VLX30 FPGA device with system clock of 1.846 MHz and 16 KByte of

Table III
SECURITY PROPERTY COMPARISON

Properties	[3]	[4]	[8]	[9]	[10]	[11]	[25]	[26]	[27]	[28]	[41]	Proposed Scheme
P1	Yes	Yes	No	Yes	No	Yes	No	No	Yes	No	Yes	Yes
P2	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
P3	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes
P4	No	No	No	No	No	Yes	No	No	No	Yes	No	Yes
P5	-	-	-	-	-	No	-	-	-	No	-	Yes

P1: Privacy of the smart meter; **P2:** Session-key security ; **P3:** Impersonation attack resistance;
P4: Protection Against Physical Attacks; **P5:** Protection Against Machine Learning/Modeling Attacks;

Table IV
COMPARISON BASED ON THE COMPUTATIONAL COST, EXECUTION TIME AND NUMBER OF INTERACTION

Schemes	Computation-Cost (at the Smart-Meter)	Execution-Time (in-clock-cycles)	NoI
Protocol of [3]	4×MP + EXP + 6×H	4×987,784 +1296,656 +6×12,145 = 5320,662	4
Protocol of [4]	3×MP + EXP + 5×H	3×987,784 + 1296,656 +5×12,145 = 4320,733	4
Protocol of [8]	4×MP + SD + 2×H	4×987,784 +16,358 + 2×12,145 = 3979,639	5
Protocol of [9]	3×MP + SE + SD + 4×H	3×987,784+13,194 +16,358 +4×12,145= 3041,484	3
Protocol of [10]	SE + 4×H	13,194 + 4×12,145= 61,774	3
Protocol of [11]	FE.Rec + 7×H + 2×PUF (e.g., APUF)	413,615 +7×12,145 +2 ×7,284 = 513,198	4
Protocol of [25]	2×MP + 4×H	2×987,784 + 4×12,145= 2024,148	3
Protocol of [26]	3×H	3×12,145= 36,435	3
Protocol of [27]	8×H	8×12,145= 97,160	3
Protocol of [28]	6×H + 2×PUF (e.g., APUF)	6×12,145 +2 ×7,284 = 87,438	4
Protocol of [41]	4×H	4×12,145 =48,580	2
Proposed Scheme	FHD + 6 ×H + WPUF+ RPUF	11,760 + 6×12,145 + 4,129 +6,154 =94,903	3

P: PUF Operation; **H:** Hash Operation (SHA-256); **FE.Rec:**Reconstruction Algorithm (Fuzzy Extractor);
MP: Multiplication-point Operation; **EXP:** Modular Exponential Operation; **SE:**Symmetric-key Encryption(AES-CBC);
SD: Symmetric-key Decryption(AES-CBC); **FHD:** Fractional Hamming Distance;
NoI: Number-of-Interaction During the Execution of the Authentication Process;

program memory. We implemented this design at a system clock of 1.846 MHz to reflect the constrained platform for the device. We also use a MSP430 micro-controller for interfacing the PUFs and NVM. The communication between the micro-controller and the hardware engine is implemented through a shared-memory. The micro-controller initializes the input arguments for the hardware engine in the shared memory, initiates the protocol computation, and waits for a completion notification from the hardware engine. After completion, the result of the computation is available in the shared memory. In order to evaluate the effect of noise, before execution of each phase of the above protocols, we power cycle the device to reinitialize the APUF and the DRAM-PUF. This gives us a real noise profile. Next, since the protocol presented in [11] uses a fuzzy-extractor, we construct helper data from a (63,16,23)-BCH code [18]. The BCH encoding function expands the randomness of a 16-bit seed into a 63-bit code-word. In this regard, for the error correction part of the FE.Rec, we used a LFSR-based implementation of the BCH encoding. We measured the computational cost for each function in system clock cycles, where it was observed that each hash operation (SHA-256) takes 12,145 clock cycles and each FE.Rec operation takes 413,615 clock cycles. Whereas, exacting each APUF, SRAM-PUF, and DRAM-PUF (G8E DDR2) output takes 7,284, 4,129 and 6,154 clock cycles, respectively. From Table IV, we can see that the protocol presented in [28] incurs lower computational cost as compared to [11] and the proposed reconfigurable authentication scheme. However, it should be noted that the scheme presented in [11] has not considered the noise issue, which is extremely important in PUF-based security solutions. In contrast, the protocol presented in [11] addresses the noise issue by relying on conventional fuzzy extractor algorithms. In comparison, the proposed scheme uses FHD to efficiently handle the noise issue in PUFs, where FHD can ensure significantly lower computational cost than FE-Rec (as shown in Table IV).

From Table IV, we can see that both the computational cost and execution time (in terms of CPU cycles) of the proposed scheme is significantly lower than that of [11]. In addition, in case of the proposed scheme, the number of interactions required during the execution of the authentication process is three, whereas the protocol presented in [11] and [28] require four interactions. Therefore, we can say that communication cost of the proposed scheme is lower than that of [11] and [28]. Next, from Table III and Table IV, it can also be argued that as compared to the solution in [11], the proposed

Table V
ACCURACY RESULT (IN %)

Machine Learning Methods	DRAM RPUF#1 (at Pandaboard-Platform)	DRAM RPUF#2 (at Intel-Galileo-Platform)
Naive Bayes	10.23 %	15.83 %
Logistic Regression	13.56 %	16.78 %
Support Vector Machine	15.39 %	18.65 %

reconfigureable authenticated key-exchange scheme can not only ensure higher security level (since [11] cannot ensure security against machine learning/modeling attacks), but also incurs significantly lower computational and communication cost. Finally, from Table IV we can also see that PUF-based authentication schemes can ensure lower computational overhead on a resource-limited smart meter as compared to public-key-based authentication schemes (such as [3], [4], [8], [9] and [25]). In addition, these schemes cannot ensure physical security of the smart meter. In [26] and [41] two ultra-lightweight and secure communication scheme have been proposed using XOR and one-way hash function operation. However, the protocol in [26] needs to run the Diffie-Hellman key establishment protocol to share and update the cryptographic keys between the smart meters and neighborhood gateways, which incurs additional computational cost at the resource limited smart meters. Besides, in the Step 6 of the protocol, the smart meters reveal their identity. Hence, the protocol cannot ensure privacy of the smart-meter but can guarantee confidentiality of usage data. In addition, both the protocols cannot guarantee hardware protection of the smart meter against physical attacks.

A DRAM PUF implementation utilizes power-up and cell refresh behavior to generate entropy for transforming stored challenge bit-strings to response bit-strings, along with the reconfiguration characteristic of RPUF (**used in designing our proposed authenticated key-exchange scheme**) to ensure backward and forward unpredictability of the response bit-strings. These properties make a DRAM PUF secure against modeling attacks [14]. The proposed scheme considers a DRAM PUF as an RPUF, which is expected to ensure both backward and forward unpredictability. Section IV theoretically showed that how a RPUF ensures such properties. In this section, we conduct another experiment to show the resilience of RPUFs (DRAM-based PUFs) against modeling attacks. For this purpose, we extract PUF instances from the Panda-Board ES Revision B3 and the Intel Galileo Gen 2 platforms. The Panda-Board contains a System-on-Chip (SoC) module (T1 OMAP 4460) that implements 1 GB DDR2 (G9E DDR2) memory (**we denote this as DRAM RPUF #1**). Similarly the Intel Galileo platform contains a SoC (Intel Quark SoC X1000) with 1 GB of DDR3 memory (**we denote this as DRAM RPUF #2**). To access DRAM PUFs during run-time, we implemented a Linux kernel module for each platform. The kernel module modifies the memory controller to disable DRAM refresh. After a time interval (decay time) of 10 sec, the memory refresh behavior is enabled again and then we read out the PUF response. It has been observed that during a PUF query, much of the CPU resources are spent on selective memory refresh, where the security our protocol is inherently based on the number of newly flipped bits that emerge in each selective memory refresh operation. In our experiment, we measured two 32KB logical PUFs on both the Pandaboard and Intel Galileo, where we use three different decay times to measure each of the logical PUFs 250 times. All the measurement were taken at room temperature with DRAM chips operating at around $\mathbf{38^o}$ C.

After measuring the PUF instances from both the DRAM PUFs of the Pandaboard and Intel Galileo, we use the Scikit-learn machine learning library to model the DRAM PUF behavior. In our evaluation, we use three well-known machine learning algorithms (Naive Bayes (NB), Support Vector Machine (SVM) , and the Logistic Regression (LR)) for predicting correct response on a given challenge with accuracy. In order to do that, we first trained each model with m_x number of measurement data and then tried to predict the (m_{x+1})-th measurement data. After that, we compared their accuracy with the original data. From Table V, we can see that the accuracy of the results for these three methods are very low. Therefore, it can be argued that modeling a reconfigureable PUF such as a DRAM PUF is difficult (which is also proved in [14]) and hence they are secure against modeling attacks. In contrast, it has been demonstrated that other PUFs (such as APUFs) are vulnerable to modeling attacks based on supervised learning algorithms (NB, LR, SVM) with significantly higher accuracy rates [12-13].

VI. CONCLUSION

In this paper, we first pointed out a security vulnerability of PUF-based authentication protocols for smart grids. To address the problem, we developed an effective and robust reconfigurable PUF-based authentication protocol for secure smart grid communication. Through the comparative analyses, we have shown that the proposed scheme can ensure better security level as compared to the existing solutions. In addition, the proposed reconfigureable authentication scheme can ensure lower computational cost and execution time as compared to others. Hence, it can be argued that the proposed scheme is a better choice for securing the smart metering network in smart grids.

ACKNOWLEDGEMENTS

This research was supported in part by Singapore Ministry of Education Academic Research Fund Tier 1 (R-263-000-D01-114). The work of Prosanta Gope was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Award EP/V039156/1. Authors would like to thank all the anonymous reviewers for their insightful comments and valuable suggestions.

REFERENCES

[1] H. Nicanfar, P. Jokar, K. Beznosov, and V. C. M. Leung, "Efficient authentication and key management mechanisms for smart grid communications," *IEEE Systems Journal*, vol. 8, no. 2, pp. 629–640, 2014.

[2] A. Mohammadali, M. S. Haghighi, M. H. Tadayon, and A. Mohammadinodooshan, "A novel identity-based key establishment method for advanced metering infrastructure in smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2834–2842, 2018.

[3] 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. 906–914, 2016.

[4] 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. 1900–1910, 2018.

[5] Y. Chen, J.-F. Mart´ınez, P. Castillejo, and L. Lopez, "An anonymous ´ authentication and key establish scheme for smart grid: Fauth," *Energies,* vol. 10, no. 9, p. 1354, 2017.

[6] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics, vol.* 65, no. 10, pp. 7996–8004, 2018.

[7] A. Braeken, P. Kumar, and A. Martin, "Efficient and provably secure key agreement for modern smart metering communications," *Energies*, vol. 11, no. 10, p. 2662, 2018.

[8] N. Saxena, B. J. Choi, and R. Lu, "Authentication and authorization scheme for various user roles and devices in smart grid," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 5, pp. 907–921, 2017.

[9] P. Kumar, A. Gurtov, M. Sain, A. J. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4349–4359, 2019.

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

[11] 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.

[12] U. Ruhrmair, F. Sehnke, J. Solter, G. Dror, S. Devadas, and J. Schmidhuber, "Modeling Attacks on Physical Unclonable Functions," *in Proc. 17th ACM Conference on Computer and Communications Security (CCS)*, 2010, pp. 237–249.

[13] J. Delvaux, Machine-Learning Attacks on PolyPUFs, OB-PUFs, RPUFs, LHS-PUFs, and PUF-FSMs. *IEEE Transactions on Information Forensics and Security*. 14(8), pp.2043-2058, 2019.

[14] S. Sutar, A. Raha, et al. , D-PUF: An Intrinsically Reconfigurable DRAM PUF for Device Authentication and Random Number Generation, *ACM Transactions on Embedded Computing Systems*, vol 17(1), 2017.

[15] P. Gope, and B. Sikdar, "Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart-Grids," *IEEE Transactions on Information Forensics & Security*, Vol. 14(6), pp. 1554 –1566, 2019

[16] G. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Design Automation Conference, 2007, *DAC '07, 44th ACM/IEEE,* 2007, pp. 9–14.

[17] M. Campagna, "Sec 4: Elliptic curve Qu-Vanstone implicit certificate scheme (ECQV)," Certicom Research, Mississauga, ON, Canada, Tech. Rep, 2013.

[18] J. Delvaux, D. Gu, I. Verbauwhede ,M. Hiller, "Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications" *In: Cryptographic Hardware and Embedded Systems (CHES),* LNCS vol. 8913 pp. 412-430, Springer (2016).

[19] R. C. Parks, "Advanced metering infrastructure security considerations," Sandia Report SAND2007-7327, Sandia National Laboratories, 2007.

[20] P. Gope, O. Millwood, and B. Sikdar, " Scalable Protocol Level Approach to Prevent Machine Learning Attacks on PUF-based Authentication Mechanisms for Internet-of-Medical-Things," *IEEE Transactions on Industrial Informatics*, DOI: 10.1109/TII.2021.3096048, 2021.

[21] T. W. Chim, S. Yiu, V. O. K. Li, L. C. K. Hui, and J. Zhong, "Prga: Privacy-preserving recording & gateway-assisted authentication of power usage information for smart grid," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 85-97, 2015.

[22] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3144-3153, 2017.

[23] M. Qi and J. Chen, "Two-pass privacy preserving authenticated key agreement scheme for smart grid," *IEEE Systems Journal*, vol. PP, pp. 1-7, 05 2020.

[24] N. Kumar et al., "Eccauth: A secure authentication protocol for demand response management in a smart grid system," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6572-6582, 2019.

[25] S. Garg et al., "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Trans. Industrial Informatics*, vol. 16, no. 5, pp. 3548-3557, 2020.

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

[27] S. Yu et al., "Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment," *Applied Sciences*, vol. 10, no. 5, p. 1758, 2020.

[28] M. Kaveh and M-R Mosavi, "A Lightweight Mutual Authentication for Smart Grid Neighborhood Area Network Communications Based on Physically Unclonable Function," *IEEE System Journal*, vol. 14, no. 3, pp. 6194-6205, 2020.

[29] C. H. Wang and C. Y. Lin, "An efficient delegation-based roaming payment, protocol against denial of service attacks," *in Proc. Int. Conf. Electron., Commun. Control*, pp. 4136-4140, Sep. 2011.

[30] P. Gope, et al., "Resilience of DoS Attack in Designing Anonymous User Authentication Protocol for Wireless Sensor Networks," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 498-503, 2016.

[31] Q. Zhao et al., "A 1036-F^2/bit high reliability temperature compensated cross-coupled comparator-based PUF," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 6, pp. 1449-1460, June 2020.

[32] S. Devadas and M.-D. Yu, "Secure and robust error correction for physical unclonable functions," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 48–65, Jan./Feb. 2010.

[33] J. Li and M. Seok, "Ultra-compact and robust physically unclonable function based on voltage-compensated proportional-to-absolute temperature voltage generators," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2192-2202, Sep. 2016.

[34] V. Sureshkumar, S. Anandhi, R. Amin, N. Selvarajan, and R. Madhumath, "'Design of robust mutual authentication and key establishment security protocol for cloud-enabled smart grid communication," *IEEE Systems Journal*, Dec., 2020.

[35] K. Mahmood et al., "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication ," *Future Generation Computer Systems*, April 2018.

[36] D. Sadhukhan et al., "A secure and privacy preserving lightweight authentication scheme for smart-grid communication using elliptic curve cryptography," *Journal of Systems Architecture* , Nov., 2020.

[37] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, S.M. Mazinani, "A secure and efficient key establishment scheme for communications of smart meters and service providers in smart grid," *IEEE Trans. Ind. Inf.*, 16 (3) , pp. 1495-1502, 2020.

[38] D. Abbasinezhad-Mood, M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps," *IEEE Trans. Ind. Inf.*, 14 (11), pp. 4815-4828, 2018.

[39] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, " Novel anonymous key establishment protocol for isolated smart meters," *IEEE Trans. Indust. Electron.*, 67 (4) , pp. 2844-2851, 2020

[40] Y. Liu, C. Cheng, T. Gu, T. Jiang, and X. Li, "A lightweight authenticated communication scheme for smart grid," *IEEE Secsors Journal*, vol. 16, no. 3, pp. 836–842, Feb. 2016.

[41] S. Aghapour, M. Kaveh et al., "An Ultra-Lightweight Mutual Authentication Scheme for Smart Grid Two-Way Communications," *IEEE Access*, May 2021.

[42] M. Just and S. Vaudenay, " Authenticated Multi-Party Key Agreement," *Advances in Cryptology-ASIACRYPT* , 1996.

Prosanta Gope (M'18-SM'21) is currently working as an Assistant Professor in the Department of Computer Science (Cyber Security) at the University of Sheffield, UK. Dr. Gope served as a Research Fellow in the Department of Computer Science at National University of Singapore (NUS). Primarily driven by tackling challenging real-world security problems, he has expertise in lightweight anonymous authentication, authenticated encryption, access control, security of mobile communications, healthcare, Internet of Things, Cloud, RFIDs, WSNs, SmartGrid and hardware security of the IoT devices. He has authored more than 75 peer-reviewed articles in several reputable international journals and conferences and has four filed patents. He received the *Distinguished Ph.D. Scholar* Award 2014 by National Cheng Kung University (Taiwan). Several of his papers have been published in high impact journals such as IEEE TIFS, IEEE TDSC, IEEE TIE, IEEE TSG, etc. Dr. Gope has served as TPC member/Chair in several reputable international conferences such as IEEE TrustCom, IEEE GLOBECOM (Security-track), ARES, etc. He currently serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL, IEEE SYSTEMS JOURNAL, IEEE SENSORS JOURNAL, and the *Security and Communication Networks*.

Biplab Sikdar (S'98-M'02-SM'09) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was on the faculty of Rensselaer Polytechnic Institute from 2001 to 2013, first as an Assistant and then as an Associate Professor. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include computer networks, and security for IoT and cyber physical systems. Dr. Sikdar is a member of Eta Kappa Nu and Tau Beta Pi. He served as an Associate Editor for the IEEE Transactions on Communications from 2007 to 2012. He currently serves as an Associate Editor for the IEEE Transactions on Mobile Computing.