Open access • Journal Article • DOI:10.1109/TDSC.2020.2992801

Safe is the new Smart: PUF-based Authentication for Load Modification-Resistant Smart Meters — Source link

Harishma Boyapally, Paulson Mathew, Sikhar Patranabis, Urbi Chatterjee ...+4 more authors

Institutions: Indian Institute of Technology Kharagpur

Published on: 06 May 2020 - IEEE Transactions on Dependable and Secure Computing (IEEE)

Topics: Smart meter, Smart grid, Mutual authentication, Secure communication and Authentication

Safe is the new Smart: PUF-based Authentication for Load Modification-Resistant Smart Meters

Author(s):
Boyapally, Harishma; Mathew, Paulson; Patranabis, Sikhar; Chatterjee, Urbi; Agarwal, Umang; Maheshwari, Manu; Dey, Soumyajit; Mukhopadhyay, Debdeep

Publication date:
2022-01

Permanent link:
https://doi.org/10.3929/ethz-b-000408208

Originally published in:
IEEE Transactions on Dependable and Secure Computing 19(1), https://doi.org/10.1109/TDSC.2020.2992801

Safe is the new Smart: PUF-based Authentication for Load Modification-Resistant Smart Meters

Boyapally Harishma, Paulson Mathew, Sikhar Patranabis, Urbi Chatterjee, Umang Agarwal, Manu Maheshwari, Soumyajit Dey, and Debdeep Mukhopadhyay

Abstract—In the energy sector, IoT manifests in the form of next-generation power grids that provide enhanced electrical stability, efficient power distribution, and utilization. The primary feature of a Smart Grid is the presence of an advanced bi-directional communication network between the Smart meters at the consumer end and the servers at the Utility Operators. Smart meters are broadly vulnerable to attacks on communication and physical systems. We propose a secure and operationally asymmetric mutual authentication and key-exchange protocol for secure communication. Our protocol balances security and efficiency, delegates complex cryptographic operations to the resource-equipped servers, and carefully manages the workload on the resource-constrained Smart meter nodes using unconventional lightweight primitives such as Physically Unclonable Functions. We prove the security of the protocol using well-established cryptographic assumptions. We implement the proposed scheme end-to-end in a Smart meter prototype using commercial-off-the-shelf products, a Utility server, and a credential generator as the trusted third party. Additionally, we demonstrate a physics-based attack named load modification attack on the Smart meter to demonstrate that merely securing the communication channel using authentication does not secure the meter, but requires further protections to ensure the correctness of the reported consumption. Hence, we propose a countermeasure to such an attack that goes side-by-side with our protocol implementation.

Index Terms—Smart Grid, Smart Meter, PUF, Authentication, Key-Exchange, Physics-based Attacks.

1 Introduction

The Smart grid is a classical example of complex Cyber Physical Systems (CPS) with distributed generation, transmission and user-end electricity consumption monitored by Smart metering capabilities. The functionalities of such a system can be decomposed into two broad domains, 1) the measurement and information exchange plane, 2) the power system dynamics control plane [1], [2]. The first component is useful for gathering system-level measurements of physical quantities like voltage, current (magnitude as well as phase) using costly equipment like Phasor Measurement Units (PMUs). These measurements are useful for implementing the lower level control loops associated with Automated Generation Control (AGC), Load Frequency Control (LFC), etc. As part of the grid control plane, such loops constitute the backbone of a three-level hierarchical control of Smart grids following well-known grid control standards like [3], [4], [5]. The secondary control loop in such systems are responsible for deciding switching on and off operations of power generation units, both renewable as well as non-renewable, thus performing economic dispatch of power within a closely coupled grid system. The third and highest level of grid control loops decides on the level of cooperation and coordination among loosely coupled grid subsections with independent generation and control, thus deciding upon high-level, complex decisions like real-time pricing.

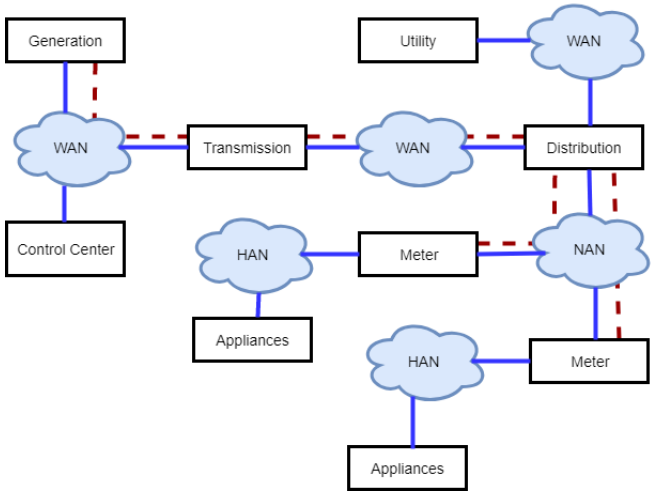


Fig. 1: Smart Grid Architecture

An overview of a Smart grid architecture is given in Fig. 1. The dotted lines represent the flow of electrical power,

- *Boyapally Harishma, Paulson Mathew, Urbi Chatterjee, Soumyajit Dey, and Debdeep Mukhopadhyay are with the Department of Computer Science and Engineering, IIT Kharagpur, India.*
  E-mail: {harishma,paulsonmathew21,sikhar.patranabis,}@iitkgp.ac.in, {urbi.chatterjee,soumya, debdeep}@cse.iitkgp.ernet.in
- *Sikhar Patranabis is with the Department of Computer Science, ETH Zurich, Switzerland. This work was done while he was with the Department of Computer Science and Engineering, IIT Kharagpur, India.*
  E-mail: sikhar.patranabis@inf.ethz.ch
- *Umang Agarwal is with the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, India.*
  E-mail: umangagr56@iitkgp.ac.in
- *Manu Maheshwari is with the Department of Electrical Engineering, IIT Kharagpur, India*
  E-mail: manumaheshwari78@iitkgp.ac.in

and the solid lines indicate information flow associated with grid control and operation. The Wide Area Networks (WANs) are used to connect control centers associated with Generation, Transmission, and Distribution systems of power gird. The Neighbourhood Area Networks (NAN) interconnect Smart meters to the Utility to expand its visibility of the grid till the consumer end. The smart meter is the inter-mediator between NAN and consumer end load devices. In recent times, home appliances are also given the ability to connect and communicate among themselves and with Smart meters using Home Area Networks (HAN) [6]. The consumer's power consumption data is aggregated from the Smart meters belonging to different HANs by a NAN and is sent to the Utility server. In our setup, we assume that a Smart meter directly communicates with its Utility server to exchange information.

Smart meter-based energy monitoring systems form an integral part of power systems dynamics and control in distributed Smart grids. In such a complex scheme of operations, Smart meters perform the useful task of 1) communicating real-time electricity price from the grid operator to the consumer, 2) reporting the consumer's day/hour ahead demand to the grid operator [7]. One of the biggest security challenges in the Smart grid is to protect these embedded devices from security breaches that could lead to disastrous consequences. For example, an attacker can act as an authentic Smart meter and change the power consumption data in transit. A legitimate consumer can also act as an adversary to alter the power consumption data to reduce the billing cost, without being detected by the server.

In this work, we propose a lightweight heterogeneous authenticated key-exchange protocol for secure communication that can mitigate such attacks. We assume that the Smart meter is embedded with a Physically Unclonable Function (PUF), hence alleviating the requirement of storing any long-term cryptographic key. The feasibility of such a Smart meter setup has been studied and validated in the existing literature [8]. We also gently point to the recent studies on FPGA-enabled IoT devices [9], that support our assumption. A salient feature of the protocol is its operational asymmetry, such that the computation at the Smart meter is less resource-intensive than the computation at the servers. Additionally, the protocol requires little secure storage capacity on the participating devices. In terms of security, our protocol achieves forward secrecy and resistance to key-impersonation attacks under well-established cryptographic assumptions, even in the presence of adversaries that can corrupt one or more parties of the protocol.

While it is an independent technical challenge to design PUF-based secure Smart meters, deployment for such meters do not serve the actual purpose of securing the energy management control loops. Nevertheless, another class of attacks is still possible, where the attacker leverages the knowledge about the *underlying physics* of the meter. The attacker can tamper the meter readings on a large scale, causing a region-wise cascaded power outage. In this paper, we discuss an attack that is derived from the basic AC power control technique used to reduce the power consumption of a load. Since meter reading is generated inside the meter before being communicated to the Utility, the authentication scheme itself cannot mitigate this attack. In works like [7], it has been reported that orchestrating such variations in load measurements in a predefined manner may potentially lead to instabilities for lower-level control loops leading to grid sections getting disconnected by protective relays. Hence, in conjunction with the protocol, we also propose a mitigation technique to this kind of physics-based attack to realize end-to-end secured metering equipment.

1.1 Our Contributions

To summarize, our main contributions in this paper are:

- We propose a secure, operationally asymmetric, mutually authenticated key-exchange protocol. The protocol preserves the authenticity, integrity, and non-repudiation of the communicating parties in the presence of passive as well as an active adversary.
- We provide formal proof of the protocol using well-established cryptographic assumptions.
- We develop a simple yet end-to-end Smart meter test-bed from commercial-off-the-shelf products and integrate it with the proposed scheme.
- We implement the Load Modification Attack (or False Load Attack [10]) on our Smart meter test-bed using low-cost additional circuitry.
- Finally, we propose an efficient countermeasure to this attack with minimum hardware overhead.

The rest of the paper is organized as follows. In Section 2, we provide the threat model, and in Section 3 we provide the background of the work and security definition. We then introduce the proposed protocol, security proof, performance, and efficiency in Section 4. We also present a detailed comparison of the proposed protocol with the existing literature in this Section. In Section 5, we provide implementation details of our test-bed, the attacking device, and the countermeasure. We provide the software implementation details, overheads incurred by the proposed techniques, comparison of the Smart meter test-bed with existing designs, and the timing results by integrating our metering setup with our protocol in Section 6. Finally, we conclude the paper in Section 7.

2 Threat Model

System Model: The setting assumed is that the Smart meter communicates directly with the Utility back-end server. In this paper, we specifically assume a Smart meter setup that is capable of embodying an FPGA-based PUF instance. Each Smart meter has the capability to perform symmetric key encryption and group operations. On the other hand, each server is associated with a unique identity (for e.g., serial number), has the capability to perform pairing operations, and stores the secret key in a side-channel resistant, tamper proof non-volatile memory (NVM).

Threat Assumptions: The adversary is assumed to have control over the communication channel either actively or passively. We assume that the server is in a trusted environment, and a legitimate server will not impersonate as a meter to any other server. From the server, the adversary can obtain databases stored but not the secret key, which is assumed to be stored in side-channel resistant, tamper-proof NVMs. The adversary can either try to authenticate to the

server as a legitimate meter or the other way round without possession of the later. The adversary can also perform man-in-the-middle and replay attacks on the protocol. The adversary cannot clone the PUF embedded in the meter, due to its unclonability property. Instead, she can attempt to acquire the PUF embedded in the meter only by tampering the PUF circuit, making it obsolete to the adversary. On the other hand, an adversary can try to alter the power consumption data on a large scale using additional circuitry on the meter to send malicious data to the server.

3 PRELIMINARIES

In this section, we give an overview of the Smart meter functioning, and the PUF instance embedded in the meter.

3.1 Smart Meter

A Smart meter serves as a mediator between consumer and Utility operator. They empower consumers with load-level energy consumption status and Utility with real-time power consumption status, creating a closed demand response control loop. The instantaneous power at time t is computed as $p(t) = v(t) \times i(t)$ where, $v(t)$ and $i(t)$ are the instantaneous voltage and current respectively. The energy W consumed over a time interval $[0, \tau K]$ is given by:

$$W = \int_0^{\tau K} p(t)dt$$

where τ is the sampling period and, K is the number of samples. In a meter, the total energy consumption value is calculated as a summation of the product of maximum power consumed and sampling period over K samples. It is given by:

$$W \approx \tau \sum_{j=1}^{K} v(\tau j) \times i(\tau j)$$

The current and voltage signals are sampled at a frequency more than twice the power supply frequency, following the Nyquist criterion. The maximum power consumed over the sampling period is measured as the product of the highest values of current and voltage. The Utility uses this real-time power measurement value not only for billing but also for control and economic dispatch mechanisms that are used by the Smart grid services. Any manipulation to current or voltage measurements lead to incorrect power value, affecting the control operations of Utility. Smart meters being the end-point communication device, require high-level security considerations during design and deployment.

3.2 Physically Unclonable Functions

A physically unclonable function $\mathsf{PUF} : \mathscr{C} \longrightarrow \mathscr{R}$ is an injective mapping from a challenge space $\mathscr{C}$ to a response space $\mathscr{R}$. A PUF instance is deemed cryptographically useful if it satisfies the following properties:

- **Evaluatable:** Given a PUF and a challenge $\mathcal{C} \in \mathscr{C}$, it should be easy to find the response $\mathcal{R} \in \mathscr{R}$.
- **Unpredictability:** A PUF instance is said to be *sufficiently unpredictable* if an adversary that is not allowed to evaluate the PUF, is able to predict its response $\mathcal{R} \in \mathscr{R}$ to a challenge $\mathcal{C} \in \mathscr{C}$, with only negligibly small probability.
- **Uniqueness:** Given two PUF instances PUF_1 and PUF_2 over the same challenge space $\mathscr{C}$ and response space $\mathscr{R}$, the probability that $\mathsf{PUF}_1(\mathcal{C}) = \mathsf{PUF}_2(\mathcal{C})$ for any $\mathcal{C} \in \mathscr{C}$, is only negligibly small.
- **Reliability:** A PUF instance is said to be *sufficiently reliable* if its response $\mathcal{R}$ to any challenge $\mathcal{C}$ remains unaltered over time with overwhelmingly high probability.
- **One-Wayness.** A PUF instance is said to be one-way if there exists no algorithm $\mathsf{invertPUF} : \mathscr{R} \rightarrow \mathscr{C}$ which is allowed to evaluate PUF a feasible number of times and for which it holds that $Pr(\mathsf{dist}[\mathcal{R} \leftarrow \mathsf{PUF}(\mathcal{C}) \; ; \; \mathcal{R}' \leftarrow \mathsf{PUF}(\mathsf{invertPUF}(\mathcal{R}))] > \mathsf{intra}_{\mathcal{C}})$ is low for $\mathcal{C} \in \mathscr{C}$ where, $\mathsf{intra}_{\mathcal{C}}$ is defined as the distance between two responses from the same PUF instance, for challenge $\mathcal{C} \in \mathscr{C}$.

One of the main challenges inherent to any PUF-based protocol is to realize a cryptographically useful PUF in practice. A number of different PUF architectures have been proposed, targeting both hardware and software platforms. One of the foremost architectures to be proposed for hardware realizations was that of an Arbiter PUF (APUF) [11]. An APUF is a delay-based silicon PUF with a design comprising of two symmetric parallel delay lines, made up of two-port path-swapping switches connected serially. The delay difference between these paths is used to extract some instance-specific random noise, which in turn determines the final response bit corresponding to an n-bit input challenge sequence. However, a major disadvantage of any APUF architecture is its poor uniqueness, resulting in a compromise of its unpredictability property and makes it unsuitable for adoption in the context of our protocol.

Fortunately, the basic APUF may be extended to realize a Double Arbiter PUF (DAPUF) [12], that is much more suited to our requirements. A major advantage of this architecture is that it improves the uniqueness property approximately to the ideal value (which is 50%), hence is sufficiently unpredictable to be cryptographically useful. In this paper we adopted the 5-4 DAPUF design presented in [13], which is an extension of double arbiter PUFs to improve the uniqueness of delay based PUFs in FPGA. It takes as input a 64-bit challenge and outputs a 4-bit response. The 5-4 DAPUF is extensively characterized in that work and shown to be a good candidate for PUF-based authentication protocols. The detailed characterization for 5-4 DAPUF and comparison with other FPGA-based PUF candidates is presented in Section 5.1.2.

3.3 Security Definitions for Authenticated Key-Exchange Protocols

Design Goals: Informally; our authenticated key-exchange protocol must satisfy the following requirements:

- *Known Session Keys.* It must retain session key secrecy even against an adversary that may have gained some past session keys.

- *Forward Secrecy.* If secret credentials (long-term secrets) of one or more entities are compromised, the secrecy of previous session keys should not be affected.
- *Key-Compromise Impersonation.* Suppose an adversary gains access to the credentials of a given party. While this loss allows the adversary to impersonate the compromised party to all other parties, it should not allow the adversary to impersonate other parties to the compromised party.
- *Known Ephemeral Keys.* Compromise of only ephemeral (short-term) keys during a key-exchange session should not reveal the session key.

We now present a formal definition that captures all of the aforementioned design goals:

Definition 3.1. Secure Authenticated Key-Exchange Protocol Our security definitions for an authenticated key-exchange protocol involve n parties - $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_n$, and an adversary $\mathcal{B}$, all of which are probabilistic polynomial-time algorithms. Each party A_i for $i \in [1, n]$ is assumed to be in possession of its own secret credential generated by the trusted credential generator. The public counterpart of these secret credentials is the associated data, again generated by the credential generator at the beginning of the protocol. The adversary $\mathcal{B}$ runs an experiment in which it activates different $(\mathcal{A}_i, \mathcal{A}_j)$ pairs of its choice to run multiple instances of the protocol. Each such instance is referred to as a *session*, and is identified uniquely by a $\mathsf{nonce} \in \mathbb{N}$. $\mathcal{B}$ is not allowed to assign the same nonce to multiple sessions. It can, however, eavesdrop on all communications between the parties participating in any session.

$\mathcal{B}$ forces the establishment of a second session with the same session key K as the target session.

Oracle Queries. $\mathcal{B}$ is allowed to make the following oracle queries during the experiment:

- $\mathsf{Key\text{-}Reveal}(\mathsf{nonce})$: $\mathcal{B}$ can reveal the session key corresponding to a session with identifier nonce.
- $\mathsf{Credential\text{-}Reveal}(i \in [1, n])$: $\mathcal{B}$ can reveal the secret credential of any party $\mathcal{A}_i$. Any party whose credential is revealed by $\mathcal{B}$ is said to be *corrupt*.
- $\mathsf{Ephemeral\text{-}Key\text{-}Reveal}(\mathsf{nonce})$: $\mathcal{B}$ can reveal any ephemeral keys corresponding to a session with identifier nonce.
- $\mathsf{Test}(\mathsf{nonce})$: Depending on a bit $b \xleftarrow{R} \{0, 1\}$, $\mathcal{B}$ receives either an actual session key corresponding to the session with identifier nonce, or a uniformly random key. $\mathcal{B}$ is limited to only one such query, which can be made at any time during the experiment.

Fresh Sessions. A session between two parties $\mathcal{A}_i$ and $\mathcal{A}_j$ is said to be *fresh* if the following conditions are satisfied:

- $\mathcal{B}$ makes no $\mathsf{Key\text{-}Reveal}$ queries corresponding to this session.
- If $\mathcal{B}$ makes $\mathsf{Ephemeral\text{-}Key\text{-}Reveal}$ queries on this session, then it *does not make* $\mathsf{Credential\text{-}Reveal}$ queries on either $\mathcal{A}_i$ or $\mathcal{A}_j$.
- If $\mathcal{B}$ had made $\mathsf{Credential\text{-}Reveal}$ queries on $\mathcal{A}_i$ and/or $\mathcal{A}_j$, it does not make $\mathsf{Ephemeral\text{-}Key\text{-}Reveal}$ queries on this session.

$\mathcal{B}$'s goal in the experiment is to run a Test query on a fresh session and guess the corresponding bit b chosen by the oracle. At the end of the experiment, $\mathcal{B}$ outputs a guess b' for b, and wins the experiment if $b' = b$. A two-party authenticated key-exchange protocol is said to be secure if $\mathcal{B}$'s success probability in the aforementioned experiment is only negligibly smaller or greater than $1/2$. Informally, in order to achieve a success probability that is non-negligibly smaller or greater than $1/2$, the adversary B has to perform one of the following two tasks:

- It can either force the establishment of a second session with the same key as the target session key and then issue a key-reveal query on that session. This is the Key-Replication Attack.
- It is able to directly recover sufficient information about the target session key to distinguish it from the random key. This is called a Forging Attack.

4 Proposed Authenticated Key-Exchange Protocol

We now present our authenticated key-exchange protocol. As already mentioned, the protocol assumes a simplistic Smart grid communication network comprising of two device types - *Smart meters* deployed in the consumer premises and *servers* in the Utility. Smart meters are typically resource-constrained embedded devices, while servers are more computationally equipped. The Smart meter is embedded with a physically unclonable function $\mathsf{PUF} : \mathscr{C} \longrightarrow \mathscr{R}$, while each server is associated with a unique identity id (such as a serial number). The protocol proceeds through two phases - the first phase is a one-time enrollment phase performed in a trusted environment where the protocol participants - the Smart meter and the server interact with a trusted credential generator maintained by the Utility. The output of this phase is secret information, which is known only to the participants and associated data, which is publicly known. The second phase is the authenticated key-exchange phase. For each protocol run, the meter-server pair use their secret credentials along with publicly known associated data to generate an entity that will mutually authenticate them. This phase also allows us to securely exchange a session key for encrypting subsequently exchanged messages between the two communicating parties.

4.1 Identity-Based Encryption

An identity-based encryption scheme IBE [14] is an ensemble of the following probabilistic polynomial-time algorithms:

- $\mathsf{IBE.Setup}(1^\lambda)$: Takes as input the security parameter λ and outputs the master secret key msk and the master public key mpk.
- $\mathsf{IBE.KeyGen}(\mathsf{msk}, \mathsf{id})$: Takes as input the master secret key msk and an identity id, and outputs a secret key $\mathsf{sk}_{\mathsf{id}}$.
- $\mathsf{IBE.Encrypt}(\mathsf{mpk}, \mathsf{id}, M)$: Takes as input the master public key mpk, an identity id and a message M, and outputs a ciphertext C.
- $\mathsf{IBE.Decrypt}(C, \mathsf{sk}_{\mathsf{id}})$: Takes as input a ciphertext C and a secret key $\mathsf{sk}_{\mathsf{id}}$, and outputs either a message M or $\perp$.

An IBE scheme is said to be functionally correct if for all security parameters $\lambda \in \mathbb{N}$ and all pairs of identities $(\mathsf{id}, \mathsf{id'})$,

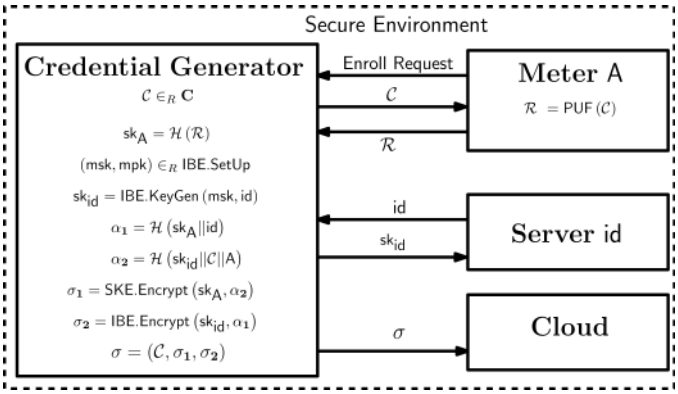


Fig. 2: Enrollment Phase

decrypting a ciphertext $C = \mathsf{IBE.Encrypt}(\mathsf{mpk}, \mathsf{id}, M)$ using a secret key $\mathsf{sk_{id'}} = \mathsf{IBE.KeyGen}(\mathsf{msk}, \mathsf{id'})$ returns M whenever $\mathsf{id'} = \mathsf{id}$, and $\perp$ with overwhelmingly large probability otherwise (where the probability is taken over the internal randomness of the aforementioned algorithms). An IBE scheme is said to be anonymously indistinguishability-secure against chosen plaintext attacks if any PPT algorithm $\mathcal{A}$ with access to a secret key generation oracle, is unable to distinguish if a ciphertext C corresponds to one of its two chosen pairs (id_0, M_0) and (id_1, M_1). In this paper, we employ IBE to avoid certificate management for the Smart metering network with growing population.

4.2 Symmetric-Key Encryption

A symmetric-key encryption scheme SKE is an ensemble of the following algorithms:

- $\mathsf{SKE.KeyGen}(1^\lambda)$: Takes as input the security parameter λ and outputs the secret key sk.
- $\mathsf{SKE.Encrypt}(\mathsf{sk}, M)$: Takes as input the secret key sk, and a message M, and outputs a ciphertext C.
- $\mathsf{SKE.Decrypt}(C, \mathsf{sk})$: Takes as input a ciphertext C and a secret key sk, and outputs the message M.

In this paper, we assume the notion of PCPA (pseudo-randomness against chosen plaintext attacks) security of SKE, which guarantees that the ciphertexts are indistinguishable from the outputs of a random function. We note that common symmetric-key encryption schemes such as AES in counter mode satisfy this security notion.

4.3 Proposed Protocol

In this section, we now describe the enrollment, authentication, and key-exchange phases in details.

Setup Phase. Let IBE be an anonymous indistinguishability-secure identity-based encryption scheme and let SKE be a PCPA-secure symmetric-key encryption scheme, both defined over a security parameter λ. Also, let $\mathcal{H} : \{0,1\}^* \longrightarrow \{0,1\}^\lambda$ and $\mathcal{H}' : \{0,1\}^* \longrightarrow \mathbb{G}$ be two collision-resistant hash functions where, $\mathbb{G}$ is a group of prime order q. The credential generator sets up the IBE scheme by sampling $(\mathsf{msk}, \mathsf{mpk}) \xleftarrow{R} \mathsf{IBE.SetUp}$.

Enrollment Phase. In this phase, the credential generator generates the following:

- **Credential for a Smart Meter:** When a new meter with identity A is added to the system, the credential generator samples $\mathcal{C} \xleftarrow{R} \mathscr{C}$, where $\mathscr{C}$ is the challenge space for the PUF instance embedded in the meter. It then evaluates $\mathcal{R} = \mathsf{PUF}(\mathcal{C})$, and sets the secret credential for the Smart meter as:

$$\mathsf{sk_A} = \mathcal{H}(\mathcal{R}) \tag{1}$$

- **Credential for a Server:** When a new server is added to the system, the credential generator sets the secret credential for the server as:

$$\mathsf{sk_{id}} = \mathsf{IBE.KeyGen}(\mathsf{msk}, \mathsf{id})$$

  where id is the unique identity associated with the server.
- **Meter-Server Associated Data:** To assign a meter to a Utility server and deploy in the grid, the credential generator calculates some associated data corresponding to this particular meter-server pair. To generate the associated data, the credential generator first computes the following:

$$\begin{aligned} \alpha_1 &= \mathcal{H}(\mathsf{sk_A}||\mathsf{id}) \\ \alpha_2 &= \mathcal{H}(\mathsf{sk_{id}}||\mathcal{C}||\mathsf{A}) \\ \sigma_1 &= \mathsf{SKE.Encrypt}(\mathsf{sk_A}, \alpha_2) \\ \sigma_2 &= \mathsf{IBE.Encrypt}(\mathsf{mpk}, \mathsf{id}, \alpha_1) \end{aligned}$$

  It then outputs the associated data as:

$$\sigma = (\mathcal{C}, \sigma_1, \sigma_2)\,. \tag{2}$$

The associated data denoted by σ is stored publicly in a readily accessible storage location (such as a cloud) and is used by the meter-server pair for all subsequent authentications.

Note that conceptually, the associated data σ is a collection of injective trapdoor one-way function outputs, evaluated on the credentials of the meter-server pair. The meter and the server can meaningfully invert the one-way function outputs using the knowledge of their own credentials, which serve as the trapdoors in this case. On the other hand, an external adversary cannot perform this inversion without gaining access to at least one of the credentials. Finally, note that while the server needs to store its credentials securely, the meter can dynamically generate its credentials at any time using its PUF instance and the associated data. The enrollment phase is illustrated in Fig. 2.

Mutual Authentication. Authentication between a meter with identity A and a server with identity id proceeds through the following round operations:

- **Round-1.** The meter sends an authentication request to the server. The server accepts the request and responds with a nonce[1] $\in \mathbb{N}$.
- **Round-2.** The server accesses the associated data entry generated during the enrollment phase, sends it to the meter along with a nonce and proceed as follows:

1. Note that we use the term nonce interchangeably to denote an integer as well as its string representation

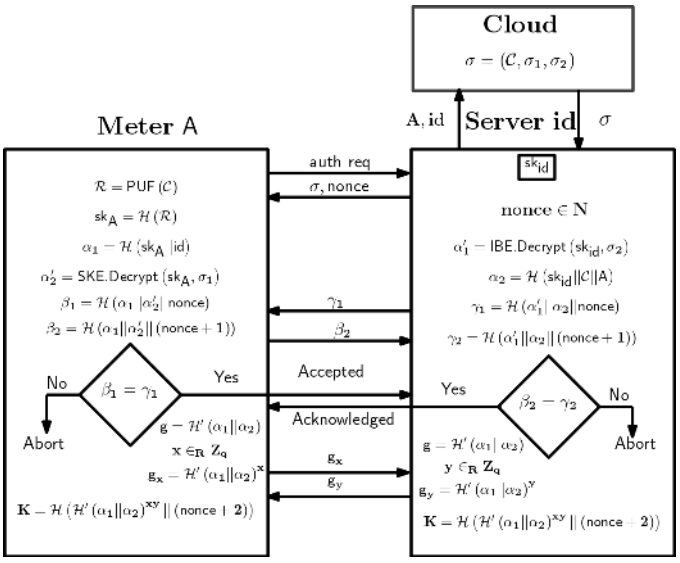


Fig. 3: Authentication and Key Exchange Phase

– The meter parses the associated data as $\sigma = (\mathcal{C}, \sigma_1, \sigma_2)$. It evaluates $\mathcal{R} = \mathsf{PUF}(\mathcal{C})$ and computes its own credential $\mathsf{sk_A}$ as in Equation 1. It then sets the following:

$$\begin{aligned} \alpha_1 &= \mathcal{H}(\mathsf{sk_A}||\mathsf{id}) \\ \alpha_2' &= \mathsf{SKE.Decrypt}(\mathsf{sk_A}, \sigma_1) \\ \beta_1 &= \mathcal{H}(\alpha_1||\alpha_2'||\mathsf{nonce}) \\ \beta_2 &= \mathcal{H}(\alpha_1||\alpha_2'||(\mathsf{nonce}+1)) \end{aligned}$$

– The server parses the associated data as $\sigma = (\mathcal{C}, \sigma_1, \sigma_2)$. It then retrieves its credential $\mathsf{sk_{id}}$ from the secure storage, and sets the following:

$$\begin{aligned} \alpha_1' &= \mathsf{IBE.Decrypt}(\mathsf{sk_{id}}, \sigma_2) \\ \alpha_2 &= \mathcal{H}(\mathsf{sk_{id}}||\mathcal{C}||\mathsf{A}) \\ \gamma_1 &= \mathcal{H}(\alpha_1'||\alpha_2||\mathsf{nonce}) \\ \gamma_2 &= \mathcal{H}(\alpha_1'||\alpha_2||(\mathsf{nonce}+1)) \end{aligned}$$

- The meter sends β_2 to the server, and the server sends γ_1 to the meter. If $\beta_2 = \gamma_2$, the server accepts the authentication request of the meter. Similarly, if $\beta_1 = \gamma_1$, the meter acknowledges the acceptance. This concludes the mutual authentication step.

Session Key Exchange. Finally, the meter and the server exchange a session key via the following round operations:

- **Round-1.** The meter uniformly samples $x \xleftarrow{R} \mathbb{Z}_q$ and sends $\mathcal{H}'(\alpha_1||\alpha_2)^x$ to the server. The server, at the same time, samples $y \xleftarrow{R} \mathbb{Z}_q$ and sends $\mathcal{H}'(\alpha_1||\alpha_2)^y$ to the meter.
- **Round-2.** The final session key K agreed upon by the meter and the server is:

$$K = \mathcal{H}(\mathcal{H}'(\alpha_1||\alpha_2)^{xy}||(\mathsf{nonce}+2))$$

This concludes the session key-exchange phase. The authentication and key-exchange phases are illustrated in Fig. 3. Note that x and y act as the ephemeral (short-term) key pair in the key-exchange step. It is to be noted that, though the key-exchange phase is based on ECDLP, it is not vulnerable to Man-in-the-Middle attacks.

Theorem 4.1. Our authenticated key-exchange protocol is secure as per Definition 1, under the assumptions that: (a) the PUF instance is sufficiently unpredictable, (b) both SKE and IBE are CPA-secure, (c) the decisional Diffie-Hellman (DDH) assumption holds in the group $\mathbb{G}$, and (d) $\mathcal{H}$ and $\mathcal{H}'$ are modeled as random oracles [15].

Proof. Let $\mathcal{B}$ be a probabilistic polynomial-time adversary as per Definition 1 against the authenticated key-exchange protocol Π. Assuming that the hash functions $\mathcal{H}$ and $\mathcal{H}'$ are modeled as random oracles, there are only two ways for $\mathcal{B}$ to distinguish an actual session key $K = \mathcal{H}(\mathcal{H}'(\alpha_1||\alpha_2)^{xy}||(\mathsf{nonce}+2))$ for a *fresh* session, from a uniformly random string in $\{0,1\}^\lambda$:

- **Key-Replication Attack.** $\mathcal{B}$ forces the establishment of a second session with the same session key K as the target session without querying the random oracle $\mathcal{H}$ on $\mathcal{H}'(\alpha_1||\alpha_2)^{xy}||(\mathsf{nonce}+2)$.
- **Forging Attack.** At some point during the experiment, $\mathcal{B}$ queries the random oracle $\mathcal{H}$ on $(\mathcal{H}'(\alpha_1||\alpha_2)^{xy}||(\mathsf{nonce}+2))$.

We first analyze the probability that $\mathcal{B}$ forces the establishment of a second session with the same session key K as the target session, without querying $\mathcal{H}$ on $(g^{xy}||(\mathsf{nonce}+2))$, where $g = \mathcal{H}'(\alpha_1||\alpha_2)$. Recall that $\mathcal{B}$ is restricted by definition from generating multiple sessions with the same nonce value. Hence, this attack amounts to forcing a collision on the random oracle $\mathcal{H}$. The probability that a probabilistic polynomial-time algorithm $\mathcal{B}$ can produce such a $\mathcal{H}$-collision in Q many queries may be formulated as:

$$\begin{aligned} \rho &= 1 - \prod_{j=1}^{Q}(1 - \mathsf{Pr}[H(q_j) = H(g^{xy}||(\mathsf{nonce}+2)) \\ &\qquad\qquad |q_j \neq (g^{xy}||(\mathsf{nonce}+2))]) \\ &= 1 - \left(1 - 2^{-\lambda}\right)^Q \end{aligned}$$

which is negligible in the security parameter λ whenever Q is polynomially large in λ. Therefore, $\mathcal{B}$ must perform a forging attack. We consider two possible forging attack scenarios depending on the nature of the queries issued by $\mathcal{B}$, and argue that the probability that it performs a forging attack is negligible in either scenario. The proof of Theorem 1 immediately follows from these arguments.

- Suppose that $\mathcal{B}$ makes Ephemeral-Key-Reveal queries on the target session. In particular, suppose that $\mathcal{B}$ reveals both the ephemeral keys x and y used during this session. In order to complete the forgery, it must recover (with non-negligible probability) *both* $\alpha_1 = \mathcal{H}(\mathsf{sk_A}||\mathsf{id})$ and $\alpha_2 = \mathcal{H}(\mathsf{sk_{id}}||\mathcal{C}||\mathsf{A})$, without querying for either $\mathsf{sk_A}$ or $\mathsf{sk_{id}}$ (see Definition 1), which are the secret keys for SKE and IBE, respectively. The only additional information available to $\mathcal{B}$ is the associated data σ, which is essentially a PUF challenge, and ciphertexts corresponding to the SKE and IBE schemes (see Equation 2). Recall that $\mathcal{B}$ does not

have physical access to the PUF instance. Given that the PUF instance is sufficiently unpredictable, the probability that $\mathcal{B}$ guesses the PUF responses corresponding to the challenge instances in the associated data within a polynomially bounded time frame, is negligible in the security parameter λ. It now remains to argue that $\mathcal{B}$ cannot recover α_2 and α_1 from the SKE and IBE ciphertexts, namely σ_1 and σ_2.

Case 1: Suppose $\mathcal{B}$ recovers α_2 from the SKE ciphertext σ_1 without the knowledge of the corresponding secret credential $\mathsf{sk_A}$ with non-negligible probability ϵ_1. Then one can trivially construct a probabilistic polynomial-time adversary $\mathcal{S}_1$ that can break the CPA-security of SKE with the same advantage. $\mathcal{S}_1$ sets up the key-exchange protocol and interacts with $\mathcal{B}$ exactly as in the real protocol, except when generating the SKE ciphertext σ_1. It forwards α_2 and a randomly chosen message α_2' to the CPA-security challenger for the SKE scheme, and receives in response a challenge ciphertext σ_2^*, which it uses instead of σ_2 in the protocol. If $\mathcal{B}$ successfully forges the session key by recovering α_2, $\mathcal{S}_1$ immediately infers that the challenge ciphertext σ_2^* must be an encryption of α_1. On the other hand, if $\mathcal{B}$ fails to forge the session key, σ_2^* is inferred to be an encryption of the randomly chosen message α_2'. This constitutes a break of CPA-security for the SKE scheme; hence, ϵ_1 must be negligible in the security parameter λ.

Case 2: Suppose $\mathcal{B}$ recovers α_1 from the IBE ciphertext σ_2 without knowing the corresponding secret credential $\mathsf{sk_{id}}$ with non-negligible probability ϵ_2. Then one can similarly construct a probabilistic polynomial-time adversary $\mathcal{S}_2$ that can break the CPA-security of IBE with the same advantage. Hence, ϵ_2 must be negligible in the security parameter λ. The argument for this inference is identical to that presented above for the SKE scheme, and is hence avoided.

In summary, under the assumption that the PUF instance is sufficiently unpredictable, and both SKE and IBE are CPA-secure, issuing Ephemeral-Key-Reveal queries on the target session allows $\mathcal{B}$ to successfully forge the session key with only negligible probability.

- Alternatively, suppose that $\mathcal{B}$ makes only Credential-Reveal queries on the target session. In particular, suppose that $\mathcal{B}$ reveals both $\mathsf{sk_A}$ and $\mathsf{sk_{id}}$. This allows $\mathcal{B}$ to compute α_1, α_2, and $g = \mathcal{H}'(\alpha_1||\alpha_2)$. In order to complete the forgery, it must successfully distinguish (g^x, g^y, K) from $(g^x, g^y, \mathcal{H}(\mathcal{H}'(\alpha_1||\alpha_2)^z||(\mathsf{nonce}+2))$ (where $x, y, z \xleftarrow{R} \mathbb{Z}_q$) *without querying for either x or y* (see Definition 1) during the fresh session. Let this experiment be $\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\Pi}$ and outputs 1 when the adversary $\mathcal{B}$ distinguishes successfully. If such an adversary exists, then we show how to build a simulator $\mathcal{S}$, which can solve the DDH problem. Simulator $\mathcal{S}$ be a PPT algorithm, which is given $(\mathbb{G}, q, g, g^x, g^y, g^w)$ as input for a target session, where w is either xy or z (where $x, y, z \xleftarrow{R} \mathbb{Z}_q$), whose goal is to determine which is the case. Adversary $\mathcal{B}$ will make all the oracle queries to the simulator $\mathcal{S}$ till the target session and only Credential-Reveal query for the target session. The simulator responds by sending $\mathcal{H}(\mathcal{H}'(\alpha_1||\alpha_2)^{x'y'}||(\mathsf{nonce}'+2))$, $(\mathsf{sk_A}, \mathsf{sk_{id}})$ and (x', y') corresponding to a session with identifier nonce'. The adversary $\mathcal{B}$ having computed α_1 and α_2, receives g when queried the random oracle $\mathcal{H}'$ for $\mathcal{H}'(\alpha_1||\alpha_2)$. The simulator reveals $K = \mathcal{H}(g^w||(\mathsf{nonce}+2))$ as the key for the fresh session and (g^x, g^y) that has been exchanged openly(public). Assume $\mathcal{B}$ distinguishes g^{xy} from g^z, then we will show that there is a negligible function negl,

$$\mathsf{Pr}[\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\Pi} = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Consider the modified authenticated key-exchange protocol $\tilde{\Pi}$ which generates the session key as $K = \mathcal{H}(\mathcal{H}'(\alpha_1||\alpha_2)^z||(\mathsf{nonce}+2))$ by choosing a random $z \xleftarrow{R} \mathbb{Z}_q$. Although $\tilde{\Pi}$ is not an actual key generation scheme, the experiment $\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\tilde{\Pi}}$ is still defined. Since g^z is a uniform group element in $\mathbb{G}$ when z is chosen uniformly over $\mathbb{Z}_q$, it follows that

$$\mathsf{Pr}[\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\tilde{\Pi}} = 1] = \frac{1}{2}.$$

Analyzing the behavior of $\mathcal{S}$, we have two cases two consider.

Case 1: If the input to $\mathcal{S}$ is generated by choosing random $x, y, z \in \mathbb{Z}_q$, then the view of $\mathcal{B}$ when run as subroutine by $\mathcal{S}$ is distributed identically to $\mathcal{B}'s$ view in experiment $\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{A},\tilde{\Pi}}$. Since $\mathcal{S}$ solves the DDH problem exactly when $\mathcal{B}$ outputs 1, we have that

$$\mathsf{Pr}[\mathcal{D}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] = \mathsf{Pr}[\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{A},\tilde{\Pi}} = 1] = \frac{1}{2}.$$

Case 2: If the input to $\mathcal{S}$ is generated by choosing random $x, y \in \mathbb{Z}_q$ and computing g^{xy}, then the view of $\mathcal{B}$ when run as subroutine by $\mathcal{S}$ is distributed identically to $\mathcal{B}'s$ view in experiment $\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\Pi}$. Define

$$\epsilon(\lambda) = \mathsf{Pr}[\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\Pi} = 1]^{q(n)c_2}$$

where $q(n)$ is the polynomial number of queries $\mathcal{B}$ makes to the oracle. Since $\mathcal{S}$ solves the DDH problem exactly when $\mathcal{B}$ outputs 1, we have that

$$\mathsf{Pr}[\mathcal{S}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1] = \mathsf{Pr}[\mathsf{AuthK}^{\mathsf{eav}}_{\mathcal{B},\Pi} = 1]^{q(n)c_2}.$$

Since DDH problem is hard relative to $\mathbb{G}$, there must be a negligible function negl such that

$$\begin{aligned}\mathsf{negl}(\lambda) &\geq \Big|\mathsf{Pr}[\mathcal{S}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] - \\ &\qquad \mathsf{Pr}[\mathcal{S}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1]\Big| \\ &= \Big|\frac{1}{2} - \epsilon(\lambda)\Big|\end{aligned}$$

This implies $\epsilon(\lambda) \leq \frac{1}{2} + \mathsf{negl}(\lambda)$ implying that the proposed authenticated key-exchange protocol is secure under the DDH assumption.

□

4.4 Security against Various Attacks

In this section, we discuss the security of the protocol against the man-in-the-middle, replay, and impersonation attacks. We also describe the security of the protocol against modeling attacks on the chosen 5-4 DAPUF.

Note that the ephemeral keys x and y in the key-exchange step of our protocol are essentially Diffie-Hellman exponents. In a naïve Diffie-Hellman key exchange protocol, an adversary $\mathcal{B}$ could potentially launch a *man-in-the-middle* attack by intercepting the communications between the two parties, and forcing them to exchange separate keys with itself individually. The vulnerability occurs due to a lack of prior authentication between the participants of the key-exchange protocol. We argue that such a vulnerability *does not occur* in our protocol. In particular, during the authentication step, the meter and the server use the associated data and their own credentials to securely compute α_1 and α_2, which in turn allows them to compute $g = \mathcal{H}'(\alpha_1||\alpha_2)$. An external adversary, on the other hand, can compute neither α_1 nor α_2, (and hence g) without knowing at least one of the secret credentials. Without the knowledge of g, it cannot launch a man-in-the-middle attack on our protocol. Observe that our authentication step acts as a lightweight alternative to the traditionally used digital signature-based mechanism for thwarting such attacks. Most importantly, our protocol avoids the need to store secret signing-keys at each of the resource-constrained devices explicitly.

The man-in-the-middle attacks [16] concerning PUF-based protocols are performed by an adversary who can intercept the challenge-response pairs exchanged in *plaintext* during authentication to perform replay attacks. However, the proposed protocol *does not exchange the challenge-response pair in plaintext*. Consequently, man-in-the-middle attacks will not work against our protocol since: (a) the raw response of the PUF (and even its hash) is not exposed to the adversary in cleartext, and (b) the response of the PUF is further re-randomized every time via hashing with an additional nonce. The usage of the nonce for re-randomization plays a key role in resisting man-in-the-middle attacks because the attacker cannot reuse a transcript of messages exchanged during a previous session to authenticate in a fresh session.

On the other hand, the single long term key $\mathsf{sk}_{\mathsf{id}}$ stored in the server, is generated during enrollment phase which is considered to be executed in a secure and trusted environment. This key is assumed to be stored in a secure tamper-proof environment, such that it is not revealed to an attacker. Once the server is deployed in the network, the secret is neither revealed to any communicating party nor sent in plaintext during protocol runs. Instead, we hash and encrypt this secret to store as associated data. During the authentication phase, the Smart meter further hashes the hashed key along with a never-repeating nonce. In this way, we protect the long term key from man-in-the-middle attacks mounted by an eavesdropping adversary. In summary, we use a combination of secure storage and transmission of long-term secrets with the appropriate usage of short-term secrets to mitigate the possibility of man-in-the-middle attacks.

We now discuss the security against modeling attacks on the PUF instance itself. As shown in [13], the modelling accuracy of the entire 4-bit response of the 5-4 DAPUF is 39%. But, the response size used in the protocol is 128 bits. So any polynomial-time adversary using the model can guess the correct 128-bit response with probability $(0.39)^{32} = 8.2 \times 10^{-12}\% \approx 2^{-43}\%$, which is negligible. In addition, we gently point to the fact that all known PUF-modelling techniques assume that the adversary has explicit access to CRP databases in plaintext for training the ML algorithms. In our protocol, we computationally hide all the responses via hashing and encryption. Even the server has access to only the hashed responses. Hence, extracting the responses in plaintext requires breaking the security guarantees of the underlying cryptographic primitives, which is computationally infeasible.

4.5 Performance and Efficiency

We discuss the performance and efficiency of our authenticated key-exchange protocol in terms of operations performed, resource, and storage requirements at the meter and the server.

Resource Requirements at the Meter. Our protocol ensures that the operations executed at the meter are lightweight and resource thrifty. The PUF instance may be efficiently implemented either in software (using SRAM cells) or in hardware (using a dedicated ASIC/FPGA chip), and embedded in the meter. The cryptographic modules required at the meter for authentication are a symmetric-key cipher SKE, and the hash functions $\mathcal{H}$ and $\mathcal{H}'$. The former may be realized efficiently using AES-128 [17], or lightweight alternatives such as PRESENT [18]. For realizing the hash function, viable options include the SHA-3 family [19] of hash functions, or lightweight alternatives such as the PHOTON family [20] of hash functions. The group $\mathbb{G}$ in the key-exchange protocol may be instantiated using a prime-order elliptic curve, with efficient implementations for the point addition and point doubling operations. The ephemeral key x may be generated using a PRNG (pseudo-random number generator) module. Each of the aforementioned modules also has efficient hardware implementations reported in the literature [21], [22], [23], with low area/power requirements and reasonable latencies. It is, in fact, a viable option to embed each meter with low-cost ASIC/FPGA-based accelerators, in addition to the PUF module, to aid the various protocol operations.

Resource Requirements at the Server. As compared to the meter, a server is equipped to handle more resource-intensive operations, such as identity-based encryption (IBE). Existing IBE schemes with short secret keys and ciphertexts typically use bilinear maps, which may be efficiently realized via Tate pairings [24]. Once again, pairings may either be implemented in software or accelerated using dedicated ASIC/FPGA-based hardware accelerators. Other cryptographic modules such as hash functions and PRNGs may be implemented as in the meter.

Key-Storage Requirements. A salient feature of our protocol is that it requires *no long-term key-storage* at the meter end. The meter may generate its credential at any time by applying the challenge to its PUF instance, and hashing the response (see Equation 1). This is especially beneficial in

Smart metering networks, where the endpoint devices typically are resource-constrained. Avoiding key-storage at the meter also inherently resists *side-channel attack* possibilities on the key storage, and avoids the need for costly side-channel countermeasures. The server only requires constant secure storage for its credential (typically around 512 bits for a single elliptic curve point), which may be readily provisioned using tamper-proof side-channel protected NVMs.
Associated Data Storage. Each associated data entry for a meter-server pair requires constant storage ξ (typically less than 1 KB). For a network with n servers and m meters, where each server i communicates with m_i $(\leq m)$ meters, the overall associated data storage requirement is $\mathcal{O}(m \cdot n \cdot \xi)$. For example, a network comprising of a million meters and $10,000$ servers, the storage requirement is less 10TB, which can be easily provisioned on public cloud-based infrastructure. Note that, the associated data need not be protected, and can be stored as-is on the cloud.

4.6 Comparison with Existing PUF-Based Authentication Protocols

In IoT, authentication and key management have been the major security concerns. Several state-of-the-art PUF-based protocols targeting heterogeneous multi-party applications such as smart cards, RFID tags, and wireless sensor networks are present in the literature with more focus on the construction of a new PUF design rather than the security of the protocol. A detailed survey on such PUF-based protocols was presented in [25], which discusses several proposed active and passive attacks such as denial-of-service (DoS) attack, synchronization problem, replay attack, token/server impersonation, PUF modeling attack, etc., making them impossible to use for IoT.

In [26], Konstantinou et al. focus on security and privacy concerns of CPS and present solutions for ensuring the same. They discuss the lack of authentication and intrusion detection mechanisms in many IoT devices and suggest the use of public-key cryptosystems and digital signature schemes to achieve the same. The PUF-based protocols presented in [27], [28] were proposed for Internet of Things where the PUF response is used as the public key for each device along with certificate-less identity-based encryption. While they are not designed with complete focus on the Smart Gird environment, but they can be adopted in any heterogeneous multi-party communication network. A lightweight two-step mutual authentication protocol using a message authentication scheme is proposed in [29], and an authentication scheme using a one-time signature for Smart Grid environments has bee proposed in [30]. All these protocols have the limitation of using heavy-weight pairing computation at the resource-constrained device end. In [31], a PUF and identity-based symmetric key-exchange framework for the advanced metering system has been proposed. The architecture has three layers;

- To generate the secret key of each Smart meter, the authors have used scalar multiplication on Elliptic Curve, exploiting the Elliptic Curve Diffie-Hellman Problem (ECDLP).
- PUF response has been used as the key to encrypt and securely store the secret key in the Smart meter.
- The scheme has used bilinear pairing to generate a symmetric key for message exchange between two such devices.

Two major limitations of the proposed system in [31] are a) For two specific communicating parties, key freshness is not maintained, as the value of the key is dependent on the identity of the device. This can lead to a replay attack of the data frames. b) As the Smart meters use computationally heavy bilinear pairing, the proposed scheme is not lightweight. In [32], Mustapa et al. mainly focuses on how to extend the idea of RO-PUF for fault-tolerant Smart grid authentication. But the drawback in this scheme is that the exchange of challenge and parity bits between the Smart meter and the authenticator is done through a secure channel. But there is no mention in the paper of how to achieve this goal.

Techniques like digital signatures, asymmetric key-based authentication, or Kerberos scheme have their own limitations of certificate management and scalability issues for resource-constrained devices. In [33], the authors pointed out that symmetric key-based encryption or similar approaches should be used for authentication purposes due to their lightweight nature. The work reported in [8] is a zero-knowledge based authentication protocol for smart meters embedded with a PUF. The protocol assumes huge secure storage on the utility server. Also, [8] does not discuss how the Smart meter authenticates the utility. In [34], a PUF design using the frequency synthesizer chain composed of voltage control oscillator (VCO) and dynamic divider has been proposed. It exploits the variation in oscillation frequency in these two components to generate a silicon fingerprint. But the authors did not analyze the PUF quality in terms of uniqueness, uniformity, reliability, and mathematical unclonability. A naïve protocol to exchange the challenge-response between the prover and the verifier has been shown in the paper. But the protocol is prone to replay and man-in-the-middle attack, and no security proof has been provided. A new lightweight PUF-based authentication protocol namely, Lockdown Technique presented in [35], is proposed to be secure against machine learning attacks.

In this paper, we try to address the issues as mentioned above in the context of PUF-enabled IoT devices. In particular, all PUF-enabled Smart meters in our protocol *do not require any secure on-chip storage*. All computations at these Smart meter nodes are run-time, resource-thrifty, and low-latency by design. We propose a mutual authentication scheme, without the requirement of secure storage on the server to save the associated data related to the Smart meter for authentication. A significant part of the storage requirement is offloaded securely to a public repository (such as a cloud), in the form of *associated data*. The associated data is protected via cost-efficient cryptographic techniques that do not demand large challenge-response spaces for the PUF instances. The security of our protocol is formally proved using well-established cryptographic assumptions. Our protocol has three advantages over the Lockdown Technique: (a) constant-overhead associated data for each meter-server pair; (b) re-usability of the same challenge-response pair for multiple authentications; (c) non-requirement of specific protocol-oriented PUF properties. A preliminary version of

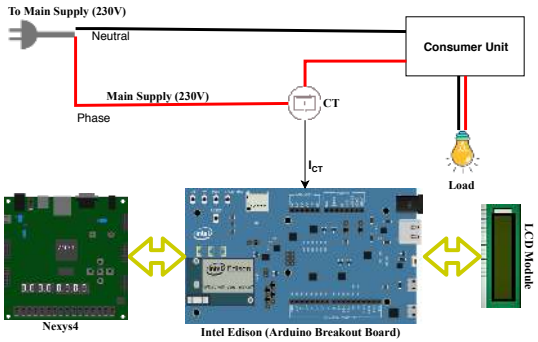


Fig. 4: Basic Meter Design

this protocol has been presented in [36] without formal security proofs. The version presented in this manuscript is significantly more detailed.

5 Smart Meter Implementation with Countermeasure

In this section, we present the design of our in-house metering device integrated with a PUF instance. As discussed earlier, the metering device and its connected load serve as additional exploitable attack surfaces that can be used by an attacker to mount load modification attacks even in the presence of secure communication as guaranteed by the PUF-based protocol. We provide the implementation details of a low-cost circuit capable of mounting such load modification attacks. We also propose a simple mitigation scheme for the same.

5.1 Smart Meter Design

In any metering setup, the current flowing through the meter to load is monitored using a current sensor. The power consumption measurement is obtained by integrating the product of the current and voltage sample over a power cycle. In our meter design shown in Fig. 4, we use a current transformer (ZMCT103C) labelled as CT1. The current through the supply line is stepped down using CT1, and the corresponding voltage is taken across a resistor. The voltage is converted to equivalent current using Ohm's law, and the corresponding Root Mean Square (I'_{RMS}) value is calculated as:

$$I'_{RMS} = \frac{I_{CT}}{\sqrt{2}} \tag{3}$$

where I_{CT} is the peak-to-peak value of the current flowing through the current transformer. The current flowing through the wire to load is calculated as:

$$I_{RMS} = I'_{RMS} \times Ratio_{CT} \tag{4}$$

where $Ratio_{CT}$ is the current transformer turn ratio. We multiply I_{RMS} with RMS voltage value to calculate the power consumption. A Liquid Crystal Display (LCD) module is used to display the measured quantities to the consumer. We use the inbuilt WiFi connectivity of Intel Edison to transfer the measured power to the Utility server by means of the secured communication channel provided by the PUF-based authenticated key-exchange protocol.

Fig. 5: In-house Built Experimental Setup

5.1.1 Embedding the PUF Instance

Next we integrate the PUF with Smart meter. We chose 5-4 DAPUF [13] for this purpose, and implement it on Digilent Nexys-4 board containing Xilinx Artix-7 FPGA. The typical input challenge to the 5-4 DAPUF is 64 bits, and the output response is of 4 bits.

As mentioned in Section 4.3, the PUF response needs to be fed to a hash function. Hence the response space should be large. Accordingly, we choose to generate a 128-bit response for each 64-bit challenge $\mathcal{C}$. To do so, we use a single 64-bit challenge as a seed to a linear feedback shift register (LFSR) module and change the state of the LFSR 32 times to generate intermediate challenge values that are fed to the PUF circuit. Finally we cascade 32, 4-bit PUF responses to generate the final 128-bit output $\mathcal{R}$.

In the proposed setup, the credential generator sends the seed or primary input challenge to the Edison board, which is communicating with the FPGA through a serial port. Upon receiving the seed, the Edison board forwards it to the FPGA that executes the LFSR+5-4 DAPUF module and returns the 128-bit output. It serves as the credential for that particular meter. The purpose of using the LFSR is to minimize communication overhead compared to giving 32 different challenges to the FPGA over serial communication.

The 5-4 DAPUF inherently provides comparable uniqueness and unpredictability, yet suffers from reliability issues due to ambient factor variations. We use majority voting and BCH error correction codes [46] to rectify the noisy responses. In particular, we use a $(7, 4, 1)$ variant of this code to rectify a single bit-flip per 4 bits of response output. The in-house built PUF integrated Smart metering setup is shown in Fig. 5.

5.1.2 Amplifying Reliability via Majority Voting

To improve the reliability of our PUF-based authentication protocol, we employ a majority voting technique. The idea

	Uniqueness	Reliability	Temperature variation	Platform	Hardware overhead	Power	Other Information
5-4 DAPUF [13]	44.16	79-88 †	$[-20, 80]$°C	Artix 7	283 FFs, 891 LUTs, 451 Slices for 4-bits	0.044W	10×10^3 CRPs on 8 FPGAs.
Loop PUF [37]	47.5	98.7	NEI ‡	Cyclone II	NEI	NEI	Outputs 16 reliable bits in 20ms (using single parity bit).
HELP PUF [38]	50.0019	99.99	0°C, 25°C, 70°C	Virtex2Pro	1749 FFs 7098 LUTs 3986 Slices	NEI	-
Interpose PUF [39]	39.0 - 59.0	98.6 - 97.9	$[0, 70]$°C	Artix 7	NEI	NEI	-
FOA PUF [40]	49.33	94.22-100	$[45, 95]$°C	Virtex 5	558 LUTs as Logic 138 LUTs as memory	1.05W	Clock Period : 5.209ns. Simulated 40 FPGAs in one Board.
TERO PUF [41]	48.5 & 47.6	97.4	$[-15, 65]$°C	Spartan 6 & Cyclone V	2 AND, 14 Inverters & 2 AND, 2 Inverters, 12 LCELL	NEI	30 Spartan 6, 18 Cyclone V 30 Clock cycles at 50 MHz Requires ECC 3 bits out of 16 bits are usable.
XRBR [42]	40.67	98.22	$[0, 70]$°C	Spartan 6	8LUTs (8 XOR gates)	NEI	450 CRPs
XRRO [42]	48.76	97.72	$[0, 70]$°C	Spartan 6	8 LUTs (1 AND, 7 XOR gates)	NEI	1-bit Response450 CRPs
ROPUF [43]	49.83	99.69	$[-5, 75]$°C (Step 10°)	Artix 7	NEI	NEI	43ms to produce 255-bits (excluding transmission time) 34 FPGAs (24 Nexys 4, 10 Basys 3) 255-bit Response
ROPUF [44]	NEI	99.33 (No ECC)	30°	Kintex 7	1217 LUTs, 365 Slices	NEI	625,000 bit/sec, Each measurement in 16 Clock cycles.
SR latch PUF [45]	49.32	96.1	$[0, 65]$°C	Spartan 3	4 LUTs, 4FFs, 2 MUX in 2 Slices of 1 CLB	NEI	2 bits per CLB 28 FPGAs

† Reliability corresponding to the raw responses without applying any error correction mechanism.
‡ Not Enough Information

TABLE 1: List of PUF designs with uniqueness, reliability, hardware overhead and other details.

is to apply a fixed challenge to the PUF instance n times, and take a majority vote over the n response samples for each bit position. The value of n may be chosen such that the probability that the correct response bit gets the majority vote is higher than some threshold probability p_{th}.

To estimate an appropriate value of n statistically, we resort to the additive form of the Chernoff-Hoeffding theorem [47]. Let $X_1, \ldots, X_n$ be independent Boolean random variables such that for each $i \in [1, n]$, we have,

$$X_i = \begin{cases} 1 & \text{if the } i^{\text{th}} \text{ response is accurate} \\ 0 & \text{if the } i^{\text{th}} \text{ response is faulty} \end{cases}$$

Let $0 < r < 1$ be the reliability of a given PUF instance. Then, for each random variable X_i, we have $\Pr[X_i = 1] = r$. Now, define the random variable $X = \sum_{i=1}^{n} X_i$. By linearity of expectation, we have $E[X] = \sum_{i=1}^{n} E[X_i] = n \cdot r$. It is easy to see that the outcome of majority voting is correct if and only if X takes a value greater than $n/2$. Also, a typical good PUF will have reliability $r > \frac{1}{2}$. Thus, majority voting fails for a good PUF instance when $X < E[X]$.

By the additive variant of the Chernoff-Hoeffding theorem, for any $\delta > 0$,we have

$$\Pr[|X - E[X]| \geq \sqrt{n} \cdot \delta] \leq 2e^{-2\delta^2}.$$

With respect to our application, the region of interest is $E[X] - X > 0$. Under this assumption, we have

$$\Pr[E[X] - X \geq \sqrt{n} \cdot \delta] \leq e^{-2\delta^2}.$$

Further, when $X \leq n/2$, we must have

$$E[X] - X \geq n \cdot \left(r - \frac{1}{2}\right) = \sqrt{n} \cdot \left(\left(r - \frac{1}{2}\right) \cdot \sqrt{n}\right).$$

Now, setting $\delta = \left(r - \frac{1}{2}\right) \cdot \sqrt{n}$, we get

$$\begin{aligned} p &= \Pr[X \leq n/2] \\ &\leq \Pr[|X - E[X]|] \geq n \cdot \left(r - \frac{1}{2}\right) \\ &\leq e^{-2\left((r-\frac{1}{2}) \cdot \sqrt{n}\right)^2} \end{aligned}$$

In other words, the probability that majority voting fails decreases exponentially as n increases. The appropriate choice of n such that $p < p_{\text{th}}$ will depend on the exact reliability parameter r of the PUF instance. Fig. 6 shows the variation of failure probability with n for reliability $r = 0.85$. Quite evidently, as n increases, the probability of authentication failure becomes negligible.

In this paper, we propose a PUF-based mutual authentication protocol and develop a prototype implementation for the same using a suitable PUF instance. The 5-4 DAPUF design we used for our implementation is a strong PUF as the challenge set size is exponential. However, our protocol can be securely instantiated using a weak PUF as the protocol does not expose the raw responses to the adversary. In the literature, many superior PUF designs exist for FPGAs compared to the 5-4 DAPUF, which we have used only for prototyping our protocol and illustrating its suitability in practice. In order to emphasize this point, we have added Table. 1 comprising some recent PUF designs along with the corresponding quality and resource-cost estimates as per reported literature. As mentioned in the majority voting

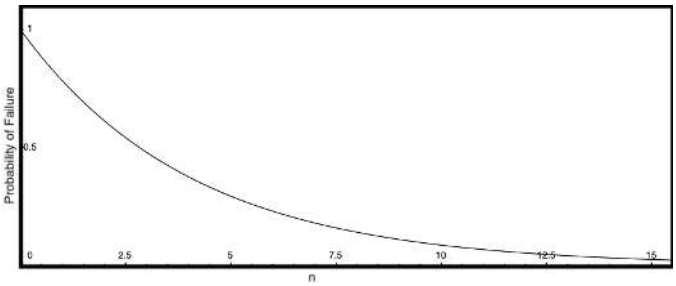


Fig. 6: Variation of failure probability of majority voting for DAPUF instance with number of samples n.

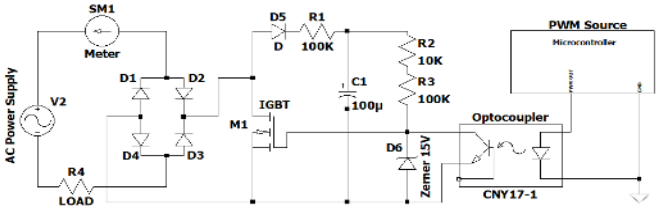


Fig. 7: Circuit Diagram of the Attacking Device

technique, the number of iterations can be set depending on the reliability estimate of the underlying PUF instance, thereby providing an opportunity for further reduction of latency. Hence, the current choice of PUF can be suitably updated depending on the decision that we make.

5.2 Load Modification Attack

As discussed earlier, while the integrated PUF instance provides security in the communication surface, physics based attacks leading to widespread instability may still be possible. We present the working methodology of such a physics-based attack on the PUF integrated Smart meter. We also present a countermeasure to mitigate it.

5.2.1 Attack Methodology

The False Load attack is derived from elemental AC power control techniques to reduce the power consumption of a load. The objective is to subvert the energy metering scheme by switching the load off when the meter tries to sample the line current. If a load device such as incandescent lamp is turned on and off rapidly using such a scheme, it *appears* to the human eye to be working continuously. Please note that this attack is feasible for any appliance where the effect of fast switching is imperceptible.

The method exploits the sampling-based current measurement as performed by any metering scheme. For this purpose, a Pulse Wave Modulation (PWM) signal of a particular duty cycle is generated using a microcontroller.

The PWM signal is used by the attacker to control any fast power switching device such as IGBT gate or MOSFET, which is inserted in series with the load. The higher current carrying capacity of IGBT makes it the first choice for designing such attacking devices.

The strategy of the attacker is to accurately synchronize the PWM signal such that a controlled IGBT switch turns the load off during the sampling period of the meter.

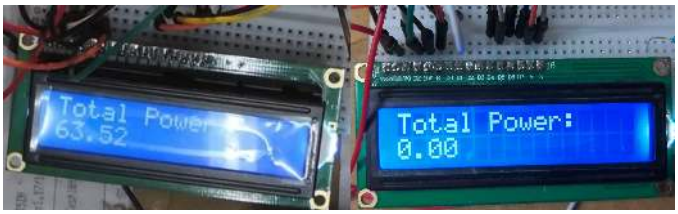


Fig. 8: Meter Under Normal Operation and Under Attack

As discussed earlier, we consider the metering equation again.

$$W = \tau \sum_{j=1}^{K} v(\tau j) \times i(\tau j)$$

where K is the number of samples considered. If the attacker succeeds in switching off the load continuously during every sampling instance of the meter over the entire period of τK, the total power consumption, as reported by the meter is 'zero'. The implementations of currently available commercial meters use continuous and zero-crossing measurement schemes, as mentioned in [10]. A zero-crossing detector circuit allows proper synchronization in zero-crossing based meter design. For continuous metering, the attacker has to synchronize by adjusting the PWM phase to get zero real-time power in the meter display unit manually.

It is to be noted that the attack does not affect the functionality of the meter, but fools the meter into not measuring the actual energy metric when the load is active. Attack on a smaller scale might not affect the stability of the grid. But a coordinated attack initiated from multiple grid regions can cause cascaded outage due to the resulting consumer demand misinterpretation and continuous fluctuations in the grid control loop [7].

5.2.2 Circuit Design of the Load Modification Attack

In this section, we propose a circuit design for executing the load modification attack. Unlike the circuit presented in [10], we propose a low-cost and easy-to-reproduce construction of the attacking circuit. This confirms that an attacker can use this kind of circuits to onset a coordinated attack initiated from multiple regions on a much larger scale.

The attack-circuit is derived from PWM based AC power control with a minimum number of components. Our attack setup consists of an AC source, an energy meter, the attacking device, and a load, as shown in Fig. 7. We placed the attacking device in-between meter and load, allowing the current from the meter to pass through the attacking device, followed by a load. The attacking device consists of an IGBT (Infineon K40EF5, Fairchild FGH60N60SFD) as a switching device, optocoupler (PC817, CYN65), and a bridge rectifier.

We have performed the attack with two different PWM sources: Arduino UNO and Mbed LPC1768 boards. Arduino is used when low-frequency PWM signals are required, whereas Mbed is used when high-frequency PWM signals are needed. The microcontroller used for generating the

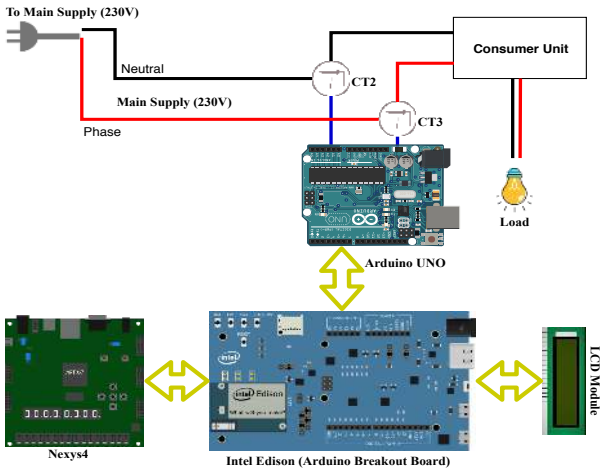


Fig. 9: Modified Meter Architecture.

PWM signal is DC grounded while the power control circuitry is AC grounded, and proper isolation is required between both. Whenever the gate voltage is high ($\approx$ 15volts), the IGBT will allow high current to flow through it to the output, which then flows through the load. We regulate the gate voltage to 15 volts using resistors (R1, R2, R3) and Zener diode.

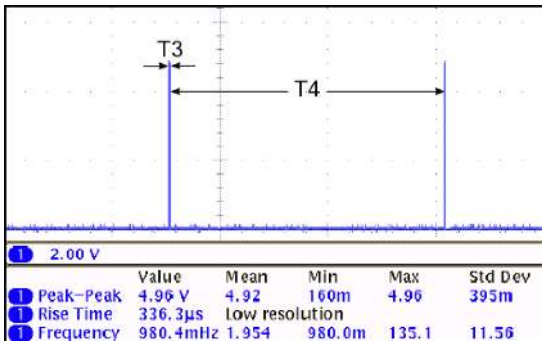
The PWM generated by the microcontroller is used to control the Optocoupler (PC817), which ensures electrical isolation between PWM source and Power control circuitry. Note that when PWM output is high, i.e., it is driving the optocoupler, whose output is shorted and the IGBT turns off. When IGBT is off, current cannot flow through the load. When the PWM output is low, the Zener holds the optocoupler output as high and drives a high current to the IGBT gate, turning it on. On the other hand, when IGBT is on, current flows through the load and the meter. Hence, if the PWM on period exactly matches with meter sampling interval in consecutive samples, the current measured by meter would be zero. Essentially this points to controlling the PWM suitably for synchronizing with meter on periods. The PWM duty cycle is selected based on the rise/fall time of the IGBT gate and the sampling duration of the meter. Consider the IGBT as an ideal switching device i.e. the rise time (T_{d_ON}) and the fall time (T_{d_OFF}) are null values. Let τ be the sampling period of the meter. If the PWM duty cycle is 50% then

$$T_{ON} = T_{OFF} = \frac{\tau}{2}$$

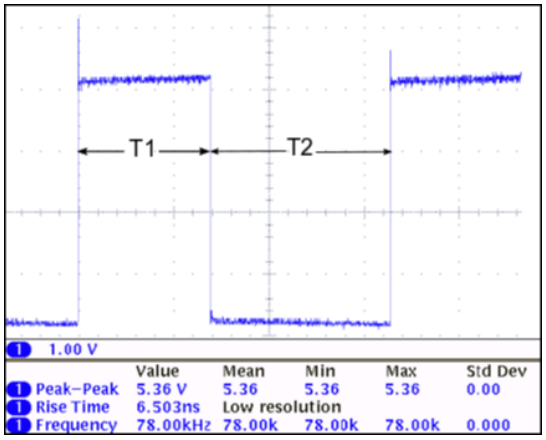
along with the condition that any meter sampling is suitably synchronized with the T_{ON} period shall ensure that the meter readings start coinciding with the load off periods. Ideally, the IGBT gate on(or off) duration is given as:

$$T_{Gate\_ON} = T_{ON}$$

But, in practice, every switching device exhibits a delay in rising and falling. Hence, in order to succeed, an attacker should consider these switching delays and adjust the T_{Gate_OFF} and T_{Gate_ON} as:



(a) Time taken by for Power Calculation is T3 + T4.



(b) Time taken for Sensor Sampling T1 + T2.

Fig. 10: Time taken by Meter for Power Calculation and for Sensor Sampling.

$$\begin{aligned} T_{Gate\_OFF} &= T_{ON} + (T_{d\_OFF} - T_{d\_ON}) \\ T_{Gate\_ON} &= T_{ON} + T_{OFF} - T_{Gate\_OFF} \end{aligned} \quad (5)$$

We consider the T_{Gate_ON} and T_{Gate_OFF} as the PWM on and off duration for the attack. The attacker needs to adjust the PWM duty cycles suitably and thus control the on/off duration of IGBT (ref. Equation 5).

The initial phase of the attack was tested against a Smart meter designed in-house, and the corresponding meter reading during normal functioning and under attack are shown in Fig. 8. Synchronization of PWM with meter sampling periods was done by online modification of the PWM duty cycle based on observed meter readings. As a load, we considered a 60 Watts incandescent lamp. We have done the initial simulation of the attacking device in LT-Spice and then designed the circuit in general-purpose Printed Circuit Board (PCB). We would like to mention that the current version of the attacking device works perfectly with resistive loads.

5.3 Mitigating Load Modification Attack

In this section, we present our proposed mitigation technique to counter the load modification attack on Smart meters. The only other mitigation approach reported to date [10] is based on randomized sampling, where the efficacy of the countermeasure varies with the randomized timing generation. The disadvantage is reduced accuracy

of measurement, which is necessarily inherent to any randomized sampling-based scheme. Our aim is to propose a more effective countermeasure strategy with high efficiency against a general class of current theft attacks [10], [48].

An overview of our proposed meter design is presented in Fig. 9. The setup consists of an Arduino UNO board that measures *both* neutral and phase samples using current transformers CT2 and CT3, respectively. The key idea is to increase the sampling rate of the ADC in the Arduino UNO using the pre-scaler option in microcontroller (Atmega328P). As a proof-of-concept (POC), we use a sampling rate of 180KHz in our implementation. This is because the ADC conversion of one current sample (phase or neutral) in our setup takes approximately $5.564\mu S$, which in turn implies that the maximum possible frequency for data sampling can be $1000/5.564 = 180KHz$. We program the microcontroller in Arduino to sequentially sample each of the neutral and phase currents at 90 KHz by switching between two ADC channels. For each instance of consecutive current samples, i.e., one neutral line reading followed by one phase line reading, the microcontroller considers the highest reading and computes instantaneous power value. The whole process of measuring and comparing the neutral and phase samples takes a constant amount of time.

It is now easy to see why our strategy thwarts current-theft attacks: our design ensures that power readings are calculated using both lines at a rate of 90 KHz, while it is known that IGBT or any other power switching device cannot switch faster than 37 KHz. In other words, no IGBT based attack device can subvert a metering scheme that is fortified using our strategy.

Fig. 10a is a timing diagram for the power measurement in our countermeasure implementation. We use an AC supply with frequency 50Hz and voltage 230V. The Arduino compares the phase and neutral samples measured during positive (marked as T1) and negative (marked as T2) cycles respectively and finalizes the instantaneous power reading. Note that the Arduino does additional computation, including calculation of instantaneous power and summation of power over an AC cycle in the negative cycles, which explains why T2 is greater than T1 in Fig. 10b. The corresponding waveform is captured by suitably instrumenting the Arduino code such that an output pin of the microcontroller is set/reset at the measurement and computation boundaries.

It is to be noted that the total time taken for the positive and negative cycles is only $14\mu s$ (T1+T2), while the AC supply waveform has a time period of $20ms$. Hence, we can obtain close to 1400 measurement samples per time period. The high number of measurement samples make our countermeasure strategy very accurate. Additionally, the total time taken by Edison for requesting and receiving instantaneous power readings from Arduino is as low as $2ms$. The Arduino sends the summation (W) of the instantaneous power value to the Edison board in every second. The Edison board displays this value through the LCD and calculates the total energy consumed at intervals (T3+T4), as shown in Fig. 10a.

To test our mitigation circuit, we introduce the attacking device into the POC implementation. The PWM signal frequency of the attacking device is increased to $78KHz$, similar to the sampling period of the meter. The signal is fed to the optocoupler as input, and the corresponding output is fed to the IGBT gate. It is observed that the IGBT in the attacking device could not switch at a higher rate, thus allowing the steady power supply to the load. Moreover, the high sampling rate, along with the continuous measurement of neutral and phase currents, result in highly granular power consumption data. This validates that our countermeasure strategy successfully thwarts load modification attacks, without compromising on the accuracy of power measurement.

6 Experimental Validation of the Proposed Protocol and Countermeasure in the Metering Infrastructure

In Section 5.1, we have illustrated the details of hardware implementation of the modified Smart meter design.

In this section, we first present the software implementation details of the proposed protocol. We follow this by presenting the overhead and security associated with the implementation of the proposed techniques. Then, we present the comparison of our Smart meter design equipped with the protocol and countermeasure with existing commercial Smart meter designs. We finally discuss the timing results on our Smart meter test-bed.

6.1 Details of the Software Implementation of Proposed Protocol

- The credential generator and the server in our metering infrastructure are simulated using PCs equipped with a Quadcore Intel i5-4570 @3.20GHz CPU, 11.6 Gb RAM, and 500 Gb storage.
- We implement the well-known IBE scheme of Boneh and Franklin [14] using the publicly available Pairing-Based Cryptography (PBC) library on the server-side. The PBC library provides APIs to securely instantiate all bilinear pairing-related operations on the Barreto-Naehrig family of elliptic curves [49], with embedding degree 12 and a security level of 160 bits.
- Once the meter is enrolled with the credential generator in a trusted environment, it is deployed in the consumer premises. Both the SKE scheme and the hash algorithms in the Smart meter are executed in the Edison board.
- We instantiate the SKE scheme using AES-128, the hash function $\mathcal{H}$ using SHA-256 and the hash function $\mathcal{H}'$ using the `element_from_hash` API of the PBC library. The AES-128 and the SHA-256 cores are adopted from *Libgcrypt* - a general purpose cryptographic library based on code from GnuPG.

6.2 Software and Hardware Overhead Incurred by the Proposed Techniques

Storage Overhead. The proposed authenticated key-exchange scheme enables a legitimate server-Smart meter pair to authenticate each other and securely communicate in the presence of an eavesdropping adversary. Since in

Cloud Storage Overhead $\sigma = (C, \sigma_1, \sigma_2)$	No. of Smart meters	AES-128	AES-256
1 entry per Smart meter	1	(64+128+2880)bits =0.4KB	(64+256+2880)bits =0.432KB
100 entries per Smart meter	1	40KB	43.2KB
	10,000	400MB	432MB
	1 Million	40GB	43.2GB
Communication Overhead		**SHA-256**	**SHA-512**
During Authentication Per Smart meter-Server Pair		256·2 =512 bits	512·2 =1024bits

TABLE 2: Overhead related to the cryptographic countermeasure for Smart meter.

Mode	Power Consumption	Operation
I	1.0994 Watts	Insecure Mode: Meter operating in Edison Board
II	2.00925 Watts	**I** + FPGA Board with PUF (inactive)
III	2.048125 Watts	**II** + PUF (active) and Protocol running

TABLE 3: Meter power consumption during various operating modes.

a Smart metering infrastructure, an adversary can impersonate as a Smart meter as well as a server, mutual authentication scheme has been proposed for this purpose. The proposed scheme ensures that it is infeasible for any adversary to succeed in authenticating itself as a legitimate entity. Hence, an adversary will neither gain information regarding power consumption, nor would it be able to inject faulty meter readings without being detected by the legitimate server. In Table. 2, we present the overheads associated with the data storage(associated data, Equation 2) in cloud and communication during authentication (γ_1, β_2). It should be noted that the Smart meter does not require any secure storage, since it embodies a PUF, and the server just needs a tamper-proof memory to store one secret key to perform IBE operations. The communication overhead during the key-exchange phase depends on the bit length of the prime used for the group $\mathbb{G}$. The length of the session key is λ (the security parameter).

Hardware Overhead. For power consumption overhead comparison, we consider the different meter operating modes as given in Table. 3.

- Basic metering functionality running in Edison board (insecure, Mode **I**).
- The basic metering setup connected with an FPGA development board having the PUF instance in inactive state (Mode **II**).
- The basic metering setup connected with an FPGA development board having the PUF instance in inactive state and authentication protocol running (Mode **III**).

Note that, in mode **II** we have more power consumption w.r.t. mode **I** due to the Digilent Nexys4 DDR FPGA development board drawing power for peripheral activity although the PUF is inactive. Thus the power consumed by only the PUF and the protocol operations are (2.04812 - 2.00925) Watts = 0.03887 Watts while discounting the FPGA peripherals. We consider this as representative of secure mode incremental power consumption in case we integrate the PUF circuit with the microcontroller in a custom build PCB. It may be observed that the extra power consumption is very small when compared with that by the microcontroller-based insecure meter operation (Mode **I**).

6.3 Comparison with Existing Meter Designs

We present a comparison of our secure Smart meter design with some of the existing secure smart meter designs in Table. 4. It is visible from the comparison table that our design outperforms others in terms of the authentication mechanism and key storage requirements. We use a PUF-based authentication, which is lightweight compared to the existing certificate-based schemes. Moreover, our design completely precludes the key storage requirement for Encryption and Authentication. In the next section, we discuss the timing results obtained from the Smart metering test-bed.

6.4 Timing Results on Our Test-Bed

Table. 5 presents timing results for an end-to-end execution of the authenticated key-exchange protocol on our test-bed:

- **Enrollment Phase.** In this phase, we generate credentials for both meter and server. The most time-consuming operation in this phase is the generation of the 128-bit golden response for a given challenge by taking majority voting over 999 responses, which serves as the credential for the meter. This may be attributed due to the serial communication delay between Edison and FPGA. The overall timing delay of the enrollment phase for each Smart meter is around **one min**. This delay will not determine the efficiency of the protocol since it is a one-time operation performed offline.
- **Authentication Phase.** The most time-consuming operation in this phase is the re-computation of credential ($\mathsf{sk_A}$) at the meter. Note that, the re-computing ability relieves the meter from any storage requirements, which is an essential feature of our protocol. The credential re-generation includes generation of a *correct* PUF response as mentioned in Section (5.1). The time taken for re-generating the credential by the software is 250 milliseconds. But this delay can be significantly decreased by implementing the majority voting technique using hardware.
- **Key-Exchange Phase.** As expected, computation of the ephemeral Diffie-Hellman exponent $\mathcal{H}'(\alpha_1||\alpha_2)^x$ at the meter end is the most time-consuming operation in the key-exchange phase. In our implementation, the exponentiation has been performed in software using the PBC library. Hence, the low processing capacity of the Edison board contributes to the relatively higher delay. The overall time consumed by the key-exchange phase is nearly 90 milliseconds in our current implementation.

However we stress, the asymmetric nature of our protocol ensures that the most resource-intensive operations, particularly those involving bilinear pairing computations are not performed on the resource-constrained meters. The aim

Meter	Encryption	Authentication Type	False Load Mitigation	Hardware overhead	Power Consumption	Key Storage Requirement
NES G4 MTR 1000	Symmetric (AES)	Available	NEI[1]	NEI	1.2W	Required
NES G4 MTR 3000	NEI	NEI	NEI	NEI	<2W	NEI
Liberty 100	Symmetric (AES)	NEI	NEI	NEI	NEI	Required
Honeywell Alpha4R	Symmetric (AES)	NEI	NEI	NEI	NEI	Required
Renesas	PKI (In-system), Symmetric	Certificate Based	NEI	Secure Coprocessor	NEI	Required
Our Design	**Symmetric (AES)**	**PUF-based**	**Available**	**PUF Circuitry**	**<=2W**	**Not Required**

[1] Not Enough Information

TABLE 4: Comparison with existing meter designs.

Protocol Phase	Execution Time (secs)	
	Meter	Server
Authentication	0.436	0.273
Key Exchange	0.089	0.087
Total Time	0.525	0.360

TABLE 5: Timing results of Authenticated Key-exchange protocol on the Smart Meter Setup

of the experimental results presented in this paper is to demonstrate the practical feasibility of the protocol on a Smart metering test-bed. We believe that more optimized implementations of our protocol, supported by dedicated accelerators and architectural nuances, would significantly reduce the timing overheads of our protocol.

7 Conclusion

In this paper, we have proposed a secure, operationally asymmetric, mutual authenticated key-exchange scheme for secure communication between Smart meters and Utility servers. We have considered the meter to be resource-constrained and having an embedded PUF instance. The protocol is designed in such a way that the meter does not need to store any credential for authentication and can generate the same on the fly at the time of protocol run. The proposed protocol also combines the benefits of both identity-based cryptography and symmetric key cryptography to effectively balance the computational overhead between the server and the meter. The protocol is proved to be forward-privacy preserving and secure against impersonation, man-in-the-middle, and replay attacks.

We have presented an end-to-end Smart metering test-bed using commercial-off-the-shelf products and integrated it with the proposed scheme. Our experimental results demonstrate that despite the huge gap in the resources, the inherent asymmetric property of our protocol ensures secure communication that can be established between the meter and the server. We have implemented a circuit to launch a specific class of physics-based attacks named load modification attacks. We proposed a mitigation technique to resist the same and show that the countermeasure can be successfully incorporated with the proposed authenticated key-exchange protocol. Finally, we concluded by presenting an experimental validation of the proposed techniques.

Acknowledgments

This work was partially funded by the Department of Science and Technology (DST), Government of India, [Sanction No: DST/SJF/ETA01/2015-16]. Debdeep Mukhopadhyay is supported by a Swarnajayanti Fellowship from DST.

References

[1] J. Qi, Y. Kim, C. Chen, X. Lu, and J. Wang, "Demand response and smart buildings: A survey of control, communication, and cyber-physical security," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 4, pp. 18:1–18:25, Oct. 2017.

[2] N. I. of Standards and T. (U.S.)., *NIST framework and roadmap for Smart Grid interoperability standards, release 1.0 [electronic resource]*. U.S. Dept. of Commerce, National Institute of Standards and Technology, Office of the National Coordinator for Smart Grid Interoperability [Gaithersburg., MD], 2010.

[3] "Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pp. 1–138, 2018.

[4] F. Leccese, "An overwiev on ieee std 2030," *2012 11th International Conference on Environment and Electrical Engineering, EEEIC 2012 - Conference Proceedings*, 05 2012.

[5] "Ieee standard for the specification of microgrid controllers," *IEEE Std 2030.7-2017*, pp. 1–43, 2018.

[6] M. N. O. Sadiku, M. Tembely, and S. M. Musa, "Home area networks: A primer," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, pp. 634–635, 05 2017.

[7] C. Su and D. Kirschen, "Quantifying the effect of demand response on electricity markets," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1199–1207, 2009.

[8] M. Nabeel, S. Kerr, X. Ding, and E. Bertino, "Authentication and key management for advanced metering infrastructures utilizing physically unclonable functions," in *IEEE Third International Conference on Smart Grid Communications, SmartGridComm 2012, Tainan, Taiwan, November 5-8, 2012*. IEEE, 2012, pp. 324–329. [Online]. Available: https://doi.org/10.1109/SmartGridComm.2012.6486004

[9] S. Wang, Y. Hou, F. Gao, and X. Ji, "A reconfigurable smart interface based on IEEE 1451 and field programmable gate array for multiple internet of things devices," *IJDSN*, vol. 13, no. 2, 2017. [Online]. Available: https://doi.org/10.1177/1550147717693848

[10] Y. Wu, B. Chen, J. Weng, Z. Wei, X. Li, B. Qiu, and N. Liu, "False load attack to smart meters by synchronously switching power circuits," *IEEE Transactions on Smart Grid*, pp. 1–1, 02 2018.

[11] R. Pappu, *Physical One-way Functions*. Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2001. [Online]. Available: https://books.google.co.in/books?id=ChD-NwAACAAJ

[12] T. Machida, D. Yamamoto, M. Iwamoto, and K. Sakiyama, "A new mode of operation for arbiter PUF to improve uniqueness on FPGA," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, September 7-10, 2014.*, 2014, pp. 871–878.

[13] U. Chatterjee, D. P. Sahoo, D. Mukhopadhyay, and R. S. Chakraborty, "Trustworthy proofs for sensor data using FPGA based physically unclonable functions," in *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018*, J. Madsen and A. K. Coskun, Eds. IEEE, 2018, pp. 1504–1507. [Online]. Available: https://doi.org/10.23919/DATE.2018.8342252

[14] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.

[15] J. Katz and Y. Lindell, *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[16] A. Babaei and G. Schiele, "Physical unclonable functions in the internet of things: State of the art and open challenges," *Sensors*, vol. 19, no. 14, p. 3208, 2019. [Online]. Available: https://doi.org/10.3390/s19143208

[17] V. Rijmen and J. Daemen, "Advanced encryption standard," *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pp. 19–22, 2001.

[18] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *CHES*, vol. 4727. Springer, 2007, pp. 450–466.

[19] Q. Dang, "Changes in federal information processing standard fips 180-4, secure hash standard," *Cryptologia*, vol. 37, pp. 69–73, 01 2013.

[20] J. Guo, T. Peyrin, and A. Poschmann, "The photon family of lightweight hash functions," *Advances in Cryptology–CRYPTO 2011*, pp. 222–239, 2011.

[21] R. P. McEvoy, F. M. Crowe, C. C. Murphy, and W. P. Marnane, "Optimisation of the sha-2 family of hash functions on fpgas," in *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on*. IEEE, 2006, pp. 6–pp.

[22] N. Sklavos and O. Koufopavlou, "On the hardware implementations of the sha-2 (256, 384, 512) hash functions," in *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, vol. 5. IEEE, 2003, pp. V–V.

[23] K. Gaj, E. Homsirikamol, and M. Rogawski, "Fair and comprehensive methodology for comparing hardware performance of fourteen round two sha-3 candidates using fpgas." in *CHES*, vol. 6225. Springer, 2010, pp. 264–278.

[24] M. Scott, N. Benger, M. Charlemagne, L. J. D. Perez, and E. J. Kachisa, "On the final exponentiation for calculating pairings on ordinary elliptic curves." *Pairing*, vol. 9, pp. 78–88, 2009.

[25] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong pufs," *ACM Comput. Surv.*, vol. 48, no. 2, pp. 26:1–26:42, Oct. 2015.

[26] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin, "Cyber-physical systems: A security perspective," in *20th IEEE European Test Symposium, ETS 2015, Cluj-Napoca, Romania, 25-29 May, 2015*. IEEE, 2015, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ETS.2015.7138763

[27] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A puf-based secure communication protocol for iot," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, pp. 67:1–67:25, 2017.

[28] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building puf based authentication and key exchange protocol for iot without explicit crps in verifier database," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2018.

[29] M. Fouda, Z. M. Fadlullah, N. Kato, R. Lu, and X. Shen, "A lightweight message authentication scheme for smart grid communications," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 675–685, 2011. [Online]. Available: https://doi.org/10.1109/TSG.2011.2160661

[30] Q. Li and G. Cao, "Multicast authentication in the smart grid with one-time signature," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 686–696, 2011. [Online]. Available: https://doi.org/10.1109/TSG.2011.2138172

[31] V. Seferian, R. Kanj, A. Chehab, and A. Kayssi, "Puf and id-based key distribution security framework for advanced metering infrastructures," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 933–938.

[32] M. Mustapa, M. Y. Niamat, A. P. D. Nath, and M. Alam, "Hardware-oriented authentication for advanced metering infrastructure," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1261–1270, 2018.

[33] K. Sha, W. Wei, T. A. Yang, Z. Wang, and W. Shi, "On security challenges and open issues in internet of things," *Future Generation Comp. Syst.*, vol. 83, pp. 326–337, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2018.01.059

[34] S. Ryu, "Puf based smart meter security with sx chain," *International Journal of Control and Automation*, vol. 9, pp. 407–414, 09 2016.

[35] M. M. Yu, M. Hiller, J. Delvaux, R. Sowell, S. Devadas, and I. Verbauwhede, "A lockdown technique to prevent machine learning on pufs for lightweight authentication," *IEEE Trans. Multi-Scale Computing Systems*, vol. 2, no. 3, pp. 146–159, 2016.

[36] B. Harishma, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay, "Poster: Authenticated key-exchange protocol for heterogeneous cps," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ser. ASIACCS '18, 2018, pp. 849–851.

[37] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet, "An easy-to-design PUF based on a single oscillator: The loop PUF," in *15th Euromicro Conference on Digital System Design, DSD 2012, Cesme, Izmir, Turkey, September 5-8, 2012*. IEEE Computer Society, 2012, pp. 156–162. [Online]. Available: https://doi.org/10.1109/DSD.2012.22

[38] J. Aarestad, P. Ortiz, D. Acharyya, and J. Plusquellic, "HELP: A hardware-embedded delay PUF," *IEEE Design & Test*, vol. 30, no. 2, pp. 17–25, 2013. [Online]. Available: https://doi.org/10.1109/MDT.2013.2247459

[39] P. H. Nguyen, D. P. Sahoo, C. Jin, K. Mahmood, U. Rührmair, and M. van Dijk, "The interpose PUF: secure PUF design against state-of-the-art machine learning attacks," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 4, pp. 243–290, 2019. [Online]. Available: https://doi.org/10.13154/tches.v2019.i4.243-290

[40] J. Zhang, X. Tan, Y. Zhang, W. Wang, and Z. Qin, "Frequency offset-based ring oscillator physical unclonable function," *IEEE Trans. Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 711–721, 2018. [Online]. Available: https://doi.org/10.1109/TMSCS.2018.2877737

[41] C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer, "Implementation and characterization of a physical unclonable function for iot: A case study with the TERO-PUF," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 97–109, 2018. [Online]. Available: https://doi.org/10.1109/TCAD.2017.2702607

[42] W. Liu, L. Zhang, Z. Zhang, C. Gu, C. Wang, M. O'Neill, and F. Lombardi, "Xor-based low-cost reconfigurable pufs for iot security," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 3, pp. 25:1–25:21, 2019. [Online]. Available: https://doi.org/10.1145/3274666

[43] A. S. Chauhan, V. Sahula, and A. S. Mandal, "Novel randomized & biased placement for FPGA based robust random number generator with enhanced uniqueness," in *32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems, VLSID 2019, Delhi, India, January 5-9, 2019*. IEEE, 2019, pp. 353–358. [Online]. Available: https://doi.org/10.1109/VLSID.2019.00079

[44] W. Yan, C. Jin, F. Tehranipoor, and J. A. Chandy, "Phase calibrated ring oscillator PUF design and implementation on fpgas," in *27th International Conference on Field Programmable Logic and Applications, FPL 2017, Ghent, Belgium, September 4-8, 2017*, M. D. Santambrogio, D. Göhringer, D. Stroobandt, N. Mentens, and J. Nurmi, Eds. IEEE, 2017, pp. 1–8. [Online]. Available: https://doi.org/10.23919/FPL.2017.8056859

[45] A. Ardakani, S. B. Shokouhi, and A. Reyhani-Masoleh, "Improving performance of fpga-based sr-latch PUF using transient effect ring oscillator and programmable delay lines," *Integration*, vol. 62, pp. 371–381, 2018. [Online]. Available: https://doi.org/10.1016/j.vlsi.2018.04.017

[46] W. Peterson and E. Weldon, *Error-correcting Codes*. MIT Press, 1972.

[47] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

[48] S. Sardar and S. Ahmad, "Detecting and minimizing electricity theft: A review," *JOURNAL OF EMERGING TRENDS IN APPLIED ENGINEERING*, vol. 1, pp. 9–12, 11 2015.

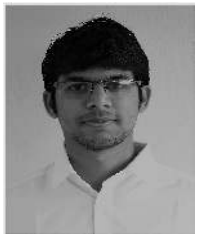
[49] P. S. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *International Workshop on Selected Areas in Cryptography*. Springer, 2005, pp. 319–331.

Boyapally Harishma has been pursuing her Ph.D. in Dept. of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, since 2017. She received B.Tech. and M.Tech. in Computer Science and Engineering from Indian Institute of Technology Kharagpur. Her research interests include authentication and secure communication protocols for smart grid applications.

Paulson Mathew has been pursuing his Ph.D. in Dept. of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, since 2018. He received B.Tech. in Electronics and Communication Engineering from MG University, Kerala, India. He is currently working as a research fellow in SEAL Lab, IIT Kharagpur, India. His research interests include the field of smart grid security.

Sikhar Patranabis is a post doctoral researcher at the Department of Computer Science, ETH Zurich. He received his Ph.D. from the Dept. of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, in 2019. His research interests span all aspects of cryptography, with special focus on cryptographic complexity, encrypted computing, and the design and implementation of real world cryptographic protocols.

Urbi Chatterjee has been pursuing Ph.D. in Dept. of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, since 2015. Before that, she worked as Assistant Systems Engineer in TATA Consultancy Services Limited, Kolkata. Her research interests are Design of PUF based Lightweight Authentication and secure communication protocols, cryptanalysis and security evaluation of PUFs.



Umang Agarwal is a third year undergraduate student pursuing his bachelors from the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology Kharagpur, India. His research interest lies in the area of VLSI circuits.

Manu Maheshwari is a third year undergraduate student pursuing his bachelors from the Department of Electrical Engineering, Indian Institute of Technology Kharagpur, India. His research interest lies in the field of digital circuit design.

Soumyajit Dey joined the dept. of CSE, IIT Kharagpur in 2013, where he is currently an Assistant Professor. He received an M.S. followed by PhD degree in Computer Science from Indian Institute of Technology, Kharagpur in 2007 and 2011 respectively. His research interests include Synthesis and Verification of Safe, Secure and Intelligent Cyber Physical Systems and Runtime Systems for Heterogeneous Platforms .

Debdeep Mukhopadhyay joined the dept. of CSE, IIT Kharagpur in 2008. He received his Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India, in 2007, where he is currently a Professor. His research interests include cryptography, VLSI of cryptographic algorithms, hardware security, and side channel analysis.