Security Challenges in Smart Grid and Suitable Countermeasures

Soumyadyuti Ghosh, Urbi Chatterjee, Durba Chatterjee, Rumia Masburah, Debdeep Mukhopadhyay, and Soumyajit Dey

Secure Embedded Architecture Laboratory (SEAL),
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur,721302.

Abstract. In recent years, the conventional power grid system has been streamlined towards Smart grid infrastructure that empowers two-way communication between the consumers and the utility providers. This however also makes the grid more susceptible towards faults as well as physical and cyber attacks. In this work, we propose a Physically Unclonable Function (PUF) and Blockchain based detection and prevention mechanism to secure the Smart grid system against such faults and adversarial threats. In this context, we discuss a recently proposed Manipulation of demand via IoT (MadIoT) attacks, False Data Injection Attacks (FDIA) via Smart meters and Electric Fault Attacks (EFA) on Smart grid which can lead to localized blackout, falsified load forecasting, imbalance in demand-response, generator tripping, frequency instability and loss of equipment. To *detect* these threats and to trace back to the source of such attacks, we inspect the potential of the promising blockchain technology which gives a mechanism to authenticate and ensure the integrity of real power consumption information. However, the blockchain needs to be augmented with a root-of-trust, to bind the Smart meter with a unique hardware fingerprint. This can be achieved through Physically Unclonable Functions (PUFs) which is considered to be an unconventional cryptographic primitive and used for keyless authentication. The proposed PUF based authentication scheme would further *prevent* the system from injection of any false data by an illegitimate Smart meter that aids to false power estimation. The novelty of the proposed work is to blend these two technologies in developing a robust and secure framework which detects and prevents all of the above mentioned security vulnerabilities and can be easily integrated with the Smart grid infrastructure. Finally an end-to-end demonstration of the attack has been presented using MATLAB and Power World simulator and the proposed framework has been prototyped using commercial off-the-shelf products such as Raspberry Pi and Artix 7 FPGA along with an in-house blockchain simulator.

Keywords: MadIoT attacks · Electric Fault attack · False Data Injection Attack · Physically Unclonable Functions · Blockchain · Smart Meters · Cryptographic Protocols

1 Introduction

The rapid changes in consumer participation, urge for improved reliability and efficiency and incorporation of renewable energy sources have made the modernisation of power grid imperative and accelerated. Since the inception of Smart grid, the sophisticated control system has been undergoing significant adjustments due to the discovery of several security exploitation. In this work, we mainly focus on recently proposed three categories of attacks that can pose a serious threat against the Smart grid system. These attacks are illustrated as follows:

1. **MadIoT Attacks:** Rapid growth of Internet-of-Things (IoT) devices and its integration with electrical appliances have opened up a new dimension for the attackers to affect the transmission as well as cyber plane of the Smart grid with multitude of penetration techniques. A large-scale compromise of such devices under Mirai Botnet [1] can elucidate a Distributed Denial of Service (DDoS) attack. Its impact can be accentuated beyond network infrastructure and can lead to localised or complete blackout in the Smart grid if not handled carefully. Soltan *et al.* proposed a novel attack called Manipulation of demand via IoT (MadIoT) where the attacker can collude with thousands of high-energy electrical appliances through IoT devices. It can disrupt a black start, significant drop/rise in the frequency, introduce cascading faults and increase operational cost. Though there are some power system protections [2] mechanisms such as automatic disconnection of the generators, under frequency load shedding, over current protection, over/under voltage protection already present in the framework, it can still lead to partition of a bulk power system and even to localised blackout. *Additionally, this protection mechanisms can not lead to the source of such Botnet attacks.*
2. **Electric Fault Attacks:** Similarly, cascading line failures can spring from the electric fault attack (EFA) [3] introduced in the transmission line physically by the adversary. The failure in one network eventuates in the disruption of another network, which consecutively perturbs the former network. Set-Valued Observers (SVOs) has been used for distributed fault detection system in the literature. Silvestre *et al.* proposed to utilise this technique [4] for both centralized detection system as well as a fully decentralized system where various detector nodes are distributed over the network and share a subset of measurements. Several machine learning (ML) and deep learning (DL) based schemes [5] have been proposed to provide a reasonable accuracy for detecting electric faults. *However, such schemes work with original voltage/current consumption signals, thus violating the privacy of the consumers.* Keeping this in mind, we address the issue of developing a detection mechanism for EFA which does not compromise with the user privacy.
3. **False Data Injection Attacks:** On the other hand, load forecasting [6] is one of the most important processes for system monitoring and involves an in-depth analysis of Smart meter measurements and evaluates power system models. Liu *et al.* [7] have proposed a new class of attacks, called false data injection attacks (FDIA), against load forecasting in electric power grids. In such attack, it is

assumed that the adversary can access the information regarding the system setup and skillfully modify the Smart meter data to bring out random errors into specific state variables without being noticed by existing algorithms. Though several works have been proposed in the literature [8, 9] to detect such bad measurements in the control system, *no preventive mechanism has been taken so far to address this issue. Hence, we try to build up an authentication framework for the Smart meters to detect falsified power profile information before it gets inserted in the cyber plane by the adversary.*

In this work, we propose an end-to-end architectural solution that blends the advantages of Physical Unclonable Functions (PUFs) and Blockchain technology to detect the source of such physical disruption as well as prevent the fraudulent data entry that can cause wrong decision making in load forecasting. Physically Unclonable Functions (PUFs) has been proposed as a promising unconventional cryptographic primitive for certificate-less identity based authentication [10,11]. A silicon PUF is an input-output mapping $\gamma : \{0,1\}^n \rightarrow \{0,1\}^m$, where the output m-bit output *response* words are unambiguously identified by both the n-bit input *challenge* words, and the unclonable, unpredictable (but repeatable) instance-specific system behavior. In the context of Smart grid, this tool can be used to authenticate the Smart meters [12], maintain the integrity of power profile that is sent by the legitimate Smart meters for the dedicated electrical appliances and prevent the adversary from injecting falsified data into the control system. Accountability of nodes also matters when it deals with tracing back the root cause of grid disruption. The existing grid technologies do not have sufficient security in place to deal with accountability and non-repudiation. Hence we integrate Smart Grid components with blockchain network. Blockchain is a distributed ledger or database that maintains a continuously growing chain of blocks. The chain is ever growing where only new records (blocks) can only be added, subject to the *consensus protocol* of the network. Owing to its unique design, it provides *immutability, integrity, transparency* and *provenance* to the blockchain data. In the state-of-the-art literature, blockchain has been used in Smart grid application for key management [13], energy trading [14], grid monitoring [15], trustworthy data aggregation [16], group signature and covert channel authorization [17] etc. But to best of our knowledge, no previous work has been proposed to use this technology to trace back and detect the source of Botnet attacks or electric fault attacks.

Hence, to summarize our work, we have the following contributions in this paper:

- First, we demonstrate an undesirable MADIoT and EFA attack and explain the intuition behind FDIA attack.
- Next, we propose a lightweight countermeasure to integrate PUF based power profile verification process with the Smart meter setup to prevent the injection of false data.
- We also propose a strategical solution to integrate blockchain network with Non intrusive Load Monitoring (NILM) process of the Smart grid to trace back to the physical attack and electrical fault source.

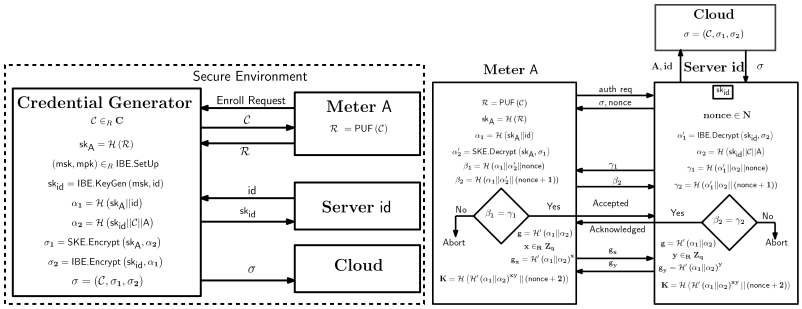


Fig. 1: Enrolment Phase (Left), Authentication and Key Exchange Phase (Right) [12]

– Finally a prototype has been implemented to demonstrate the proposed solution.

The rest of the paper is organised as follows. In Sec. 2, we provide a brief introduction of PUF based authentication and key exchange protocol. In Sec. 3, we demonstrate MadIoT attack and EFA. We also sketch the idea behind the FDIAs. In Sec. 4, we describe the proposed architecture, threat model and proposed scheme for demand modification attack detection and prevention. Finally we present the experimental results in Sec 5 and conclude our work in Sec. 6.

2 Background: PUF based Authentication and Key exchange Protocol [12]

Next we briefly describe the PUF based authenticated key exchange protocol [12] which has been referred in our work. The scheme consists of three parties, credential generator (CG) which acts as a trusted third party (TTP), server and the meter. The protocol executes in four phases as mentioned below:

Setup Phase: It is assumed that IBE is an identity-based encryption scheme and SKE is a symmetric-key encryption scheme. $\mathcal{H} : \{0,1\}^* \longrightarrow \{0,1\}^\lambda$ and $\mathcal{H}' : \{0,1\}^* \longrightarrow \mathbb{G}$ be two collision-resistant hash functions where λ is the security parameter and $\mathbb{G}$ be a group of prime order q. CG sets up its private/public key pair using IBE scheme as: $(msk, mpk) \xleftarrow{R} IBE.SetUp$.

Enrollment Phase. In this phase, the credential for the meter is generated by characterising the embedded PUF instance for challenge $\mathcal{C}$ and collecting the response $\mathcal{R}$. Next the secret key sk_A is generated by applying $\mathcal{R}$ to $\mathcal{H}$. Next the credentials for the server sk_{id} is generated by using the $IBE.KeyGen$ applying the msk and the server's identity id as input. The server then stores sk_{id} in its non-volatile memory (NVM). Finally associations between (id,sk_A) and $(sk_{id}, \mathcal{C}, A)$ are created using $\mathcal{H}$. It is denoted as α_1 and α_2. These two entities are encrypted with respective secret keys of meter A and server id using the encryption scheme of SKE and IBE. The encrypted association data σ_1 and σ_2 along with the challenge are stored in the cloud as σ.

Mutual Authentication. For each authentication request from the meter, the server responds with σ and a nonce value. The node then characterises its em-

bedded PUF instance with $\mathcal{C}$ and generates the response $\mathcal{R}$ and re-generates sk_A and α_1 as described in enrollment phase. Further it decrypts σ_1 to retrieve α_2'. α_2' ideally should be equal to α_2. Similarly, server also decrypts σ_2 to retrieve α_1' using its secret key sk_{id}. It also generates α_2 by hashing sk_{id}, $\mathcal{C}$ and A. Next the meter and the server generate:

$$\beta_1 = \mathcal{H}\left(\alpha_1||\alpha_2'||\mathsf{nonce}\right), \beta_2 = \mathcal{H}\left(\alpha_1||\alpha_2'||\left(\mathsf{nonce}+1\right)\right)$$
$$\gamma_1 = \mathcal{H}\left(\alpha_1'||\alpha_2||\mathsf{nonce}\right), \gamma_2 = \mathcal{H}\left(\alpha_1'||\alpha_2||\left(\mathsf{nonce}+1\right)\right)$$

Session Key Exchange. Finally, the meter and the server randomly choose x and y from $\mathbb{Z}_q$ respectively. They next exchange $\mathcal{H}'\left(\alpha_1||\alpha_2\right)^x$ and $\mathcal{H}'\left(\alpha_1||\alpha_2\right)^y$. The final session key is: $K = \mathcal{H}\left(\mathcal{H}'\left(\alpha_1||\alpha_2\right)^{xy}||\left(nonce+2\right)\right)$. If $\beta_1 == \gamma_1$ and $\beta_2 == \gamma_2$, then both parties successfully authenticate each other.

3 Attack Demonstration

The main objective of this work is to propose a attack source detection methodology for the very popular demand manipulation attacks on Smart grid as discussed above. But to fully realise the effectivity of the countermeasure, we first need to recreate the attack scenario. In this section, we have used the concepts of MadIoT and EFA from the state-of-the-art literature, but created our own construction to replicate it. We also give a basic intuition of how FDI attack can hamper the system monitoring given the knowledge of the underline system model.

3.1 MADIoT Demonstration

Smart grid demand-response management systems continuously strive to minimize the imbalance of power consumed and power generated, ramping up or

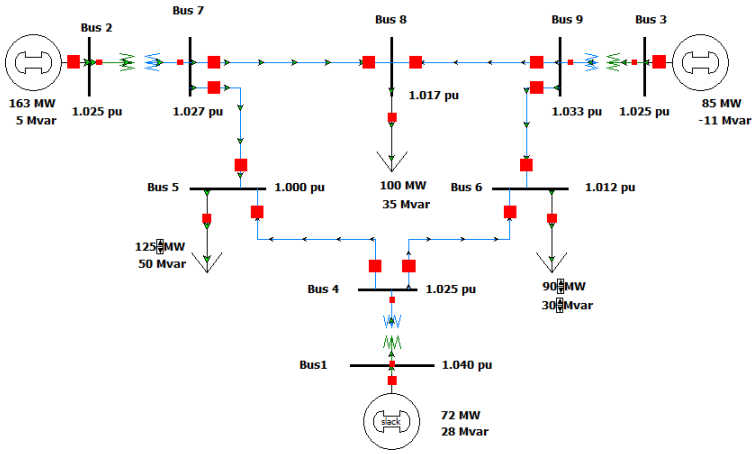
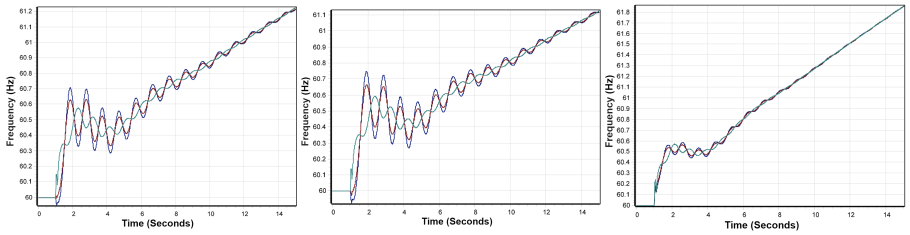
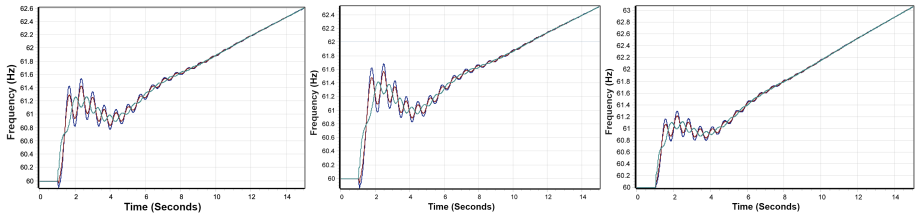


Fig. 2: WSCC 9-bus system



(a) (b) (c)

Fig. 3: Frequency deviation in load bus 5(blue), 6(red) and 8(green) for low Inertia constant (15s) of the generators due to unpredicted demand alteration in all the loads (a) Decrease of 15MW demand , (b) Decrease of 20MW demand, (c) Increase of 30MW demand



(a) (b) (c)

Fig. 4: Frequency deviation in load bus 5(blue), 6(red) and 8(green) for low Inertia constant (5s) of the generators due to unpredicted demand alteration in all the loads (a) Decrease of 15MW demand, (b) Decrease of 20MW demand, (c) Increase of 30MW demand

ramping down the power generation based on the real-time demand. It is extremely important to balance the load demand and generated power for maintaining the stability of the grid. Here, we show how power demand can be manipulated through IoT devices to disrupt the frequency of the grid [1]. We run our demonstration in a Power World simulator with the Western System Coordinating Council (WSCC) 9-bus model having a total of 315MW of initial demand as shown in Fig. 2. We consider the same model that has been used to demonstrate the MADIoT attack [1]. The bus number 5, 6 and 8 are the load buses of the corresponding model. The nominal frequency of the system is 60 Hz and it is assumed that the frequency should be within 58.2 and 61.3 Hz approximately for normal operation of the grid. Fig. 3 and Fig. 4 show the frequency disturbances of the load buses under high (15s) and low (5s) inertia constants of the generators when the demands of all the loads are unexpectedly alerted. We know that the frequency of the system is directly proportionally to the difference between supply and demand generated. So any deviation in demand causes fluctuation in the grid frequency. A large-scale coordinated MADIoT attack can lead to further deviation of the grid frequency. This attack may also result in the

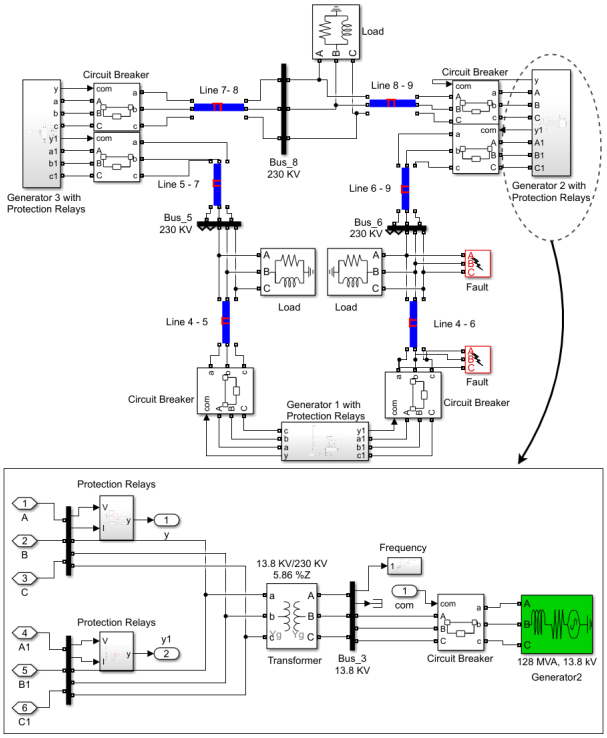
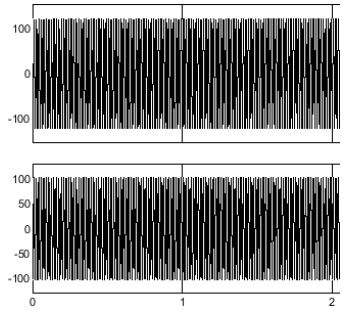


Fig. 5: An IEEE 9-Bus Model augmented with Fault Injection.

activation of generator protection relays, loss of generators and grid blackout. In case of a blackout, the grid operator needs to restart the system at the earliest. This process is known as *black start*. To avoid frequency instability due to unknown demand at this point, the operator divides the grid into small regions (also called as islands). As a result, the inertia of each region becomes very low, which makes the system vulnerable to any such demand modification. Hence, the adversary can easily hamper the black start process by suddenly increasing the demand once an island is up.

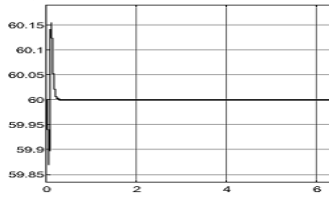
3.2 Electric Fault Attack Demonstration

In this demonstration, we show how electrical faults may cause a localized blackout of a system by disconnecting faulty transmission lines from the grid. The attack setup as shown in Fig. 5, consists of an IEEE 9-Bus power system model with three generators, nine buses, circuit breakers, transmission lines, three loads and three protection relay modules along with multiple faults. Below part of Fig. 5 shows the internal components of the block named "Generator 2 with Protection Relays". The "Protection Relays" block inside the "Generator 2 with Protection Relays" block consists of overcurrent, overvoltage and undervoltage protection systems to check whether the current and voltage of a bus are maintained within



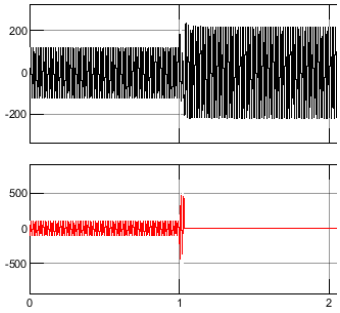
(a) Current of two buses of inside block "Generator 2 with Protection Relays" as of Fig. 5

(b) Frequency of Generator 2

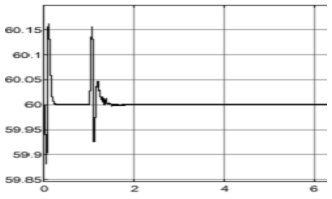


(c) Frequency of Generator 1

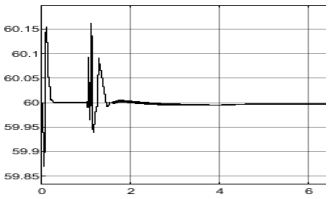
Fig. 6: Signals under Normal Operation



(a) Current of two buses of inside block "Generator 2 with Protection Relays" as of Fig. 5



(b) Frequency of Generator 2



(c) Frequency of Generator 1

Fig. 7: Signals after Fault is Induced

predefined threshold values. A circuit breaker trip signal is generated if the corresponding voltage and current values violate any of these predefined thresholds. We simulate our model using Simulink in Matlab.

Fig. 6 shows the current through two of the buses and the frequency of both generator 1 and generator 2 under normal condition. On the other hand, Fig. 7 shows the behaviour of the corresponding buses and generators when the fault is induced after one second. Due to the fault, the current through one of the buses starts increasing, resulting in a frequency disturbance leads to destabilization of the speed of generators which may permanently damage the generators. As a countermeasure, the overcurrent relay module generates an undesired breaker trigger that cuts the line 6-9 and line 4-6 in the circuit resulted in a load-shedding, as shown in Fig. 7a by using the red colour.

3.3 False Data Injection Attack via Smart Meters

The System monitoring process generates pertinent information on the current state of the power grid system. It collects all the meter readings and submits it to the control centre for thorough analysis using the power flow model. It finally estimates unknown state variables for contingency analysis. In general, a linearized power flow model, *DC Power Flow Model* is used for this purpose. Here, the relationship between measurements $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ and state variables $\mathbf{z} = (z_1, z_2, ..., z_m)^T$ can be represented as:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{e} \tag{1}$$

where $\mathbf{H}$ is an $m \times n$ full rank matrix called system Jacobian to allow estimating $\mathbf{x}$ from $\mathbf{z}$, $\mathbf{e}$ is the error vector, n is the number of state variables, m is the number of meter measurements. Assuming that error is normally distributed with zero mean, the minimum mean squared error (MMSE) estimator leads to the following matrix solution:

$$\hat{\mathbf{x}} = (\mathbf{H^T W H})^{-1}\mathbf{H^T W z} \tag{2}$$

where $\mathbf{W}$ represents diagonal matrix whose elements are reciprocals of the variances of meter errors. Now, to detect bad measurements, the state-of-the-art literature proposed to calculate the measurement residue $\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}$ and compare its 2-norm $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$ with a threshold τ. There exists a bad measurement of $||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}|| > \tau$.

Now, to achieve her goal, the adversary injects malicious data by adding it with original data. Let us assume $\mathbf{a} = (a_1, a_2, ..., a_m)^T$ is the malicious data that has been added to $\mathbf{z}$ and calculate the spurious measurement $\mathbf{z_a} = \mathbf{z} + \mathbf{a}$. This new measurement can bypass the the detection approach if $\mathbf{a}$ is the linear combination of $\mathbf{H}$ i.e., for any mismatch c, $\mathbf{a} = \mathbf{Hc}$ [7].

$$||\mathbf{z_a} - \mathbf{H}\hat{\mathbf{x}}_\mathbf{a}|| = ||\mathbf{z} + \mathbf{a} - \mathbf{H}(\hat{\mathbf{x}} + \mathbf{c})|| = ||\mathbf{z} + \mathbf{a} - \mathbf{H}\hat{\mathbf{x}} - \mathbf{Hc}|| = ||\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}||$$

Further, Anwar *et al.* proposed an improved version of blind FDI attack [18] where the stealthy attack-vector is prepared solely from the measurement matrix, without the knowledge of underline system Jacobian and the number of system states. From the above discussion, it is comprehensible that FDIA can introduce any random faults in the system without getting detected. This stealthy FDIA can disrupt system state estimation, as well as the energy distribution of the grid, resulting in destabilization of grid infrastructure. In the next section, we will describe the proposed Smart grid architecture to integrate PUFs and blockchain within its components to resist the above mentioned attacks.

4 Proposed Architecture of the Smart grid System

4.1 System Model

The setting assumed here is that the components of the grid are divided into two planes: a) Transmission Plane, b) Data Plane. Both the planes are monitored and

managed by a supervisory control and data acquisition (SCADA) system which provides various grid functionalities by maintaining power and information flow between the grid entities. The information from the power system passes through the Remote Terminal Units (RTUs) which is a fundamental part of SCADA, to the Load Dispatch Centre (LDC). LDC also plays a crucial role in the reliable and efficient operation of the grid infrastructure. It is mainly responsible for the real-time grid monitoring, operation and control of the system.

The transmission plane consists of generation, transmission, distribution networks and the consumers. It connects the communication between the power generation module, substations and consumers within a vast geographical region. This plane controls the power lines that are responsible for supplying the electricity to the household as per the demand request. Local substations regulate the power distribution between generation systems and the loads along with sending operational data to SCADA systems. The data plane maintains the information flow between the Smart meters and the trusted utility server. The Smart meters periodically collect consumers' energy profile before sending the reading streams to the trusted utility through Data Concentrator Units (DCUs). SCADA consists of a collection of this information from meters distributed throughout the area through RTUs, before selectively sending them to the LDC. While power flow from power systems through SCADA to LDC is unidirectional, the information flow maintains a bidirectional interface between power systems and LDC through SCADA. The permissioned blockchain network of the Smart grid system is constructed by these power generation module, the substations, the LDCs (local and centralized), the DCUs and the trusted utility server. We assume that the blockchain network maintains the basic security properties such as consistency (i.e. each node has the same view of the blockchain) and immutability (i.e. blockchain data once committed cannot be changed).

4.2 Adversarial Model

As discussed earlier in Sec. 3, we assume the threat model considering three potential scenarios through which the grid can be compromised. First, the attacker has physical access to the appliances or can control those appliances from a remote location. The attacker can manipulate the demand of these appliances, which can lead to disrupting the grid frequency as a result of the supply and demand imbalance. Secondly, we assume that the attacker can introduce electric fault attacks into the system, which can result in a disconnection of the transmission lines for preventing generation tripping. Moreover, in the last scenario, we assume the attacker with network access can perform eavesdropping, false data injection and replay attack to insert falsified power profile information in the Smart metering setup which may further cause undesirable control operations in the grid. Also, from a privacy aspect, we consider the grid nodes to be honest but curious entities, who want to gain the power consumption information of individual consumers to sell it to the marketing companies or obtain additional information of consumers' daily life patterns.

4.3 Working Flow of the Blockchain Network

The blockchain network consisting of the power generation modules, substations, DCU(s) and LDC(s) and a trusted utility server, is entrusted with the role of logging the events of the Smart grid. The blockchain allows grid functionalities such as link break scenario between a generator and substation, change of utility request from a consumer, or sending power consumption data of a consumer to be transparent and accessible to all participants of the blockchain network. The above-mentioned functionalities of the grid operations are realized with the help of a grid topology encoded in the form of a graph and is updated during every transaction. For instance, if an electrical link is broken between a generator module and a substation in the Smart grid, the graph is updated by removing the corresponding edge to reflect the change and the generator informs the other members of the network by posting a new transaction. The PUF enabled Smart meters, being resource thrifty, are not a part of the blockchain network. The power profile of the consumer is sent to the DCU by the respective Smart meter after its PUF instance is authenticated by the DCU. The DCU then stores the data in Inter Planetary File System (IPFS) [19] and adds the hashed value of the data (denoted by *IPFS Hash*) in a blockchain transaction (refer to Section 4.4 for details). The trusted utility server initiates the blockchain by creating the genesis block and sharing the initial grid topology. The various transactions used in our blockchain network are given as follows:

1. **InitialTemplateTxn** This transaction creates the genesis block and shares the initial grid topology with all the nodes who are part of the blockchain network. It is posted by the utility server and consists of a template file (in xml schema) which is used to generate the graph. All the blockchain nodes fetch the template file to build the initial grid view.
2. **LinkDownTxn** The transaction is posted by a node to inform other nodes about a link break down in the grid. The nodes in the network, on receiving this transaction, update their local graph to reflect the changes in the power grid.
3. **LinkUpTxn** This transaction informs about the restoration of a broken link and is posted by the same node which has posted the **LinkDownTxn** previously after the link is again available.
4. **DownReroutingTxn** This transaction initiates the rerouting process after a link breakdown to identify an alternate path to deliver electricity.
5. **ChangeUtilityTxn** This transaction is posted by a substation informing the other nodes in the network regarding the change in utility of a consumer.
6. **ConsumptionTxn** This transaction posts the power profile of a consumer. It consists of the Consumer Id and the IPFS hash of the power profile.
7. **RegisterTxn** This transaction logs the registration data of a consumer with a substation.
8. **CredGenTxn** This transaction shares the association data of a PUF enabled Smart meter and a DCU binding the root-of-trust of Smart meter with blockchain permanently and immutably.

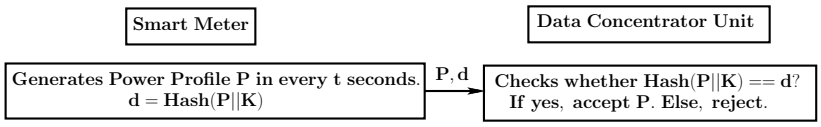


Fig. 8: Power Profile Verification Protocol

9. **AuthStatusTxn** This transaction broadcasts the status after authentication and key-exchange between a Smart meter and a DCU.
10. **NotifyLDCTxn** This transaction is posted by a generator which notifies the LDC whenever the frequency difference becomes more than the tolerance threshold τ. It consists of the identity of the generator and the timestamp when the disturbance has happened.
11. **TriggerSubstationsTxn** This transaction is posted by the LDC to trigger the substations to initiate the backtrack process to identify the source of the frequency disruption. It consists of the identity of LDC, the list of substations needed to be triggered and timestamp of the attack.

4.4 Proposed Power Profile Verification Process

Here, we first map the PUF based authentication and key exchange protocol explained in Section 2 in our usecase. We consider the utility server as a *TTP*. If there is a demand to install a Smart meter in a household, an association between the Smart meter and the DCU is made by the utility server in a secure and trusted environment. Hence the DCU is analogous to the *Server* as shown in Fig. 1 and assumed to have a secret key stored in its NVM. The Smart meter is enabled with a embedded PUF instance. The utility server stores the authentication credentials between the Smart meter-DCU pair in the IPFS) and adds a **CredGenTxn** in the blockchain to log this event. This is considered as the *enrolment phase*. For every sampling period, the Smart meter and DCU generates a session key K by following the *authentication and key exchange protocol* and a **AuthStatusTxn** is posted by the DCU in case the PUF instance of the Smart meter is successfully authenticated by the DCU. As this procedure is not part of the main contributions of the paper, we are not going into further details. Rather, we concentrate on the power profile verification procedure.
For every new sampling period, the Smart meter generates the power profile. It is to be noted that the power profile can not be encrypted with the session key as it is not possible to aggregate the encrypted data collected from multiple Smart meters at the DCU side. Moreover, encryption of such condensed data also might incur considerable execution time. Hence, a privacy preservation metering scheme is applied on it to camouflage the usage pattern of the household(refer to Section 5.2 for details). Now, we denote the privacy preserved power signature as P (refer to Fig. 8) for a sampling period of t seconds. It then calculates a hash of P appended with session key K, i.e., $d = Hash(P||K)$ and sends (P, d) to the DCU. As the DCU has also calculated the session key K, it can immediately verify the authenticity of the source of P. If the hash value matches, it saves the power profile in IPFS and posts **ConsumptionTxn** along with the Smart

meter ID. The aggregated power profile from multiple sources are then sent to the utility server for further analysis. Now, for every sampling period, a new session key is used to generate the hash value. Hence, the freshness of the session is maintained and replay attack is resisted. Moreover, if the adversary wants to inject or modify the actual power profile, she would fail to calculate a valid hash value corresponding to the modified data as she has no knowledge about the session key K. Hence, using the simple verification procedure as proposed in this work, the FDIA attack via Smart meter can be thwarted. Next, we propose a novel tracing algorithm to detect the source of any physical disturbance induced by the adversary using EFA or MadIoT attack.

4.5 Proposed Tracing Algorithm for Attack Source Detection

In this section, we propose a novel approach to detect the source of physical attack when an adversary compromises thousands of electric appliances and creates a power surge in the grid. As mentioned in Section 4, the software component in a LDC processes collected Smart metering data to maintain the security and stability of the system in real-time. However, since the number of components that need to be monitored is vast, sophisticated digital processing of the data is required. In this context, we propose our demand manipulation based attack detection algorithm to identify compromised consumers in the grid infrastructure as described in Algo. 1.

The centralized LDCs (located in state capitals) are usually connected to multiple area/sub LDCs, which are individually connected to major substations and power generation stations. If there is significant change in the frequency of any generator, the LDC gets notified. As mentioned in Algo. 1, it always looks for deviation in the frequency of a particular generator v_i (denoted as $f_{v_i}^{TS}$) at timestamp TS from the ideal frequency (denoted as f^*) by a tolerable limit τ, as shown in line 5. For a unwanted scenario, the generator v_i notifies the associated LDC about the disturbance (line 6). It also adds a **NotifyLDCTxn** in the blockchain to log this event. The LDC immediately triggers the corresponding substations under generator v_i through **TriggerSubstationsTxn** to check the consumers under its jurisdiction for any unexpected demand alteration. Every substation v_s is individually configured to identify the compromised consumer by executing $TriggerNILM$ function (Lines 27-33). For every consumer v_c, the $TriggerNILM$ function collects the average power profile of v_c. The average value is calculated using the power data till the day before the abnormality is suspected at the timestamp TS (line 29). This estimated power profile P_{ideal} is then compared with the current power profile $P_{current}$ of the consumer (line 30). The P_{ideal} and $P_{current}$ are then disaggregated by the NILM algorithm to find power profile for each appliance a_c (lines 36- 37). The algorithm then uses $Distance$ function, *Student's t-test* to find any difference between the means of these two power profiles. If the t-value resulted from the test is higher than of appliance disturbance threshold ϕ, then the algorithm considered the consumer v_c to be compromised.

Additionally, the frequency deviation can also be caused by electric faults in the transmission lines. Hence, along with triggering the NILM functionality in

Algorithm 1 Algorithm for Detection of Attack Source

Inputs: V_G : Set of Generator nodes;
V_S : Set of Substation nodes;
V_C : Set of Consumer nodes;
Grid Network G;
Blockchain Ledger $T = \{t_1, \cdot, t_i, \cdot, t_n\}$; ▷ t_i is the t^{th} block in the ledger;
Frequency Tolerance Threshold τ;
Ideal Frequency f^*;
Appliance Disturbance Threshold ϕ;
Output: A set of consumers $Consumer\_Id$, set of broken links $Broken\_Links$;
Initialize: $flag \leftarrow false$, $Consumer\_Id \leftarrow \{\emptyset\}$, $Broken\_Links \leftarrow \{\emptyset\}$;

for all $v_i \in V_G$ **do**
  if $|f_{v_i}^{TS} - f^*| > \tau$ **then**
    Consumer_Id$_{v_i}$ $\leftarrow$ NotifyLDC(v_i, TS);
    $flag \leftarrow true$;
  Consumer_Id $\leftarrow$ Consumer_Id $\cup$ Consumer_Id$_{v_i}$;
if $flag = true$ **then**
  $Broken\_Links \leftarrow$ isLinkBroken(T, V_G, V_S, V_C, G);
return $\langle Consumer\_Id, Broken\_Links \rangle$;

function isLinkBroken(T, V_G, V_S, V_C, G)
  for $t_i \in T$ **do**
    if $t_i.type =$ "$linkdown$" & $t_i.timestamp \geq TS$ **then**
      $Broken\_Links \leftarrow Broken\_Link \cup t_i.link$;
  return $Broken\_Links$;

function NotifyLDC(v_i, TS)
  $C\_id \leftarrow \{\}$;
  V_S^* is set of substations in TriggerSubstationsTxn t_i where $t_i.timestamp \geq TS$.
  for all $v_s \in V_S^*$ **do**
    $C\_id_{v_s} \leftarrow \{\}$;
    V_C^* is the set of all consumers under v_s.
    for all $v_c \in V_C^*$ **do**
      $C\_id_{v_s} \leftarrow C\_id_{v_s} \cup$ TriggerNILM(v_c , TS);
    $C\_id \leftarrow C\_id \cup C\_id_{v_s}$;
  return C_id ;

function TriggerNILM(v_c, TS)
  $flag \leftarrow false$;
  $P_{ideal} \leftarrow getAveragePowerProfile(v_c, TS)$;
    ▷ Returns average power profile of v_c till last date
  $P_{current} \leftarrow getPowerProfile(v_c, TS)$; ▷ Power profile of v_c at TS
  $flag \leftarrow$ Compare(v_c, P_{ideal}, $P_{current}$);
  if flag **then**
    return v_c;

function Compare(v_c, P_{ideal}, $P_{current}$)
  for all $a_c \in v_c$ **do**
    $P_{ideal}(a_c) \leftarrow NILM(P_{ideal})$;
    $P_{current}(a_c) \leftarrow NILM(P_{current})$;
    if $Distance(P_{ideal}(a_c), P_{current}(a_c)) \geq \phi$ **then**
      return $true$;

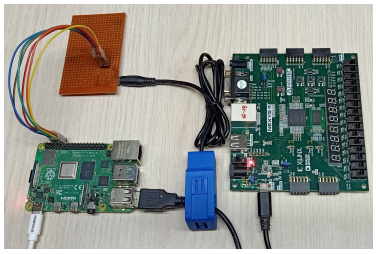


Fig. 9: PUF Enabled Smart Meter Prototype

ID	BUS	ID	BUS
0	('Gen1', 'Sub1')	10	('Sub2', 'Consumer5')
1	('Gen1', 'Sub2')	11	('Sub2', 'Consumer6')
2	('Gen2', 'Sub2')	12	('Sub2', 'Consumer7')
3	('Gen3', 'Sub2')	13	('Sub3', 'Consumer8')
4	('Gen3', 'Sub4')	14	('Sub3', 'Consumer9')
5	('Gen4', 'Sub3')	15	('Sub3', 'Consumer10')
6	('Sub1', 'Consumer1')	16	('Sub4', 'Consumer11')
7	('Sub1', 'Consumer2')	17	('Sub4', 'Consumer12')
8	('Sub1', 'Consumer3')	18	('Sub4', 'Consumer13')
9	('Sub2', 'Consumer4')	19	('Sub4', 'Consumer14')

Fig. 10: Transmission Links in the Grid

the substations, the LDC also calls for $IsLinkBroken$ function (Lines 12- 16) to check whether there is any link that has been brought down recently. The $IsLinkBroken$ function traverses in the blockchain and retrieves all **LinkDownTxn** transactions whose timestamp is same or more than TS and returns the links which are down. If there is any such fault, it triggers **DownReroutingTxn** transaction to deliver the power supply through alternative link.

4.6 Overall Summary

On a summary, we first propose our power system model consisting of the SCADA, RTU, LDC, DCU, power generation and distribution units, substations and the consumers. Except the consumers, all the components of the power system are considered to be a part of the blockchain network and assumed to log their activities as a transaction in the blockchain ledger. The consumers are enabled with a PUF instance and the DCU(s) accept the Smart meter data after verifying the authenticity of the same, thus resisting the FDIAs at the time of state monitoring. Finally, in case of any demand manipulation or fault attack in the system, the LDC or subsequent nodes are triggered to execute the tracing algorithm (refer to Algo. 1) as described above to detect the source of physical disruption using the power profile stored in the blockchain or identify the broken links in the network from blockchain transaction history. Next, we discuss the proof-of-concept implementation of the proposed model and the feasibility of the countermeasures.

5 Experimental Setup and Results

In this section, we describe our experimental setup comprising of PUF enabled Smart meter, blockchain network and the attack source detection procedure.

5.1 Setup for PUF enabled Smart Meter, Blockchain and Authentication and Key-Exchange Protocol

As discussed in Sec. 4.4, a Smart meter embedded with a PUF instance can thwart FDIA attacks (refer to Sec. 3.3). To realise this, we have made a PUF enabled Smart meter prototype using Raspberry Pi, a non-invasive split core current transformer (SCT-013-030) and a Digilent Nexys 4 FPGA board, as

shown in Fig. 9. In any metering setup, current flowing through the meter to load is monitored using a current sensor. Current measured by the sensor is captured by Raspberry Pi and sent to blockchain node (utility server). We choose a 5-4 Double Arbiter PUF [20], and deploy it on Digilent Nexys-4 board containing Xilinx Artix-7 FPGA. Raspberry Pi communicates with the PUF instance over USB to send challenges and receive the corresponding response. Now, the blockchain framework (refer to Sec. 4.3) is implemented in Golang and Python. Each blockchain node is enabled with a REST API built using Gorilla MUX that is used to post transactions to the blockchain, along with providing the ability to view the blockchain data in a web browser. Creation of point-to-point (P2P) network and handling the P2P connections is implemented using go-libp2p library. The web-server returns the chain of transactions in JSON format for simplicity. File sharing in the blockchain is enabled by IPFS. In order to share a file, the node posting the transaction adds the IPFS hash of the file to be shared in the transaction block. The transactions used in the blockchain and their corresponding actions are explained in Section 4.3. We have also shown the JSON structure of four transactions, namely **InitialTemplateTxn**, **LinkDownTxn**, **ConsumptionTxn** and **NotifyLDCTxn** in Fig. 11.
As mentioned in Sec. 4.1, the functionalities of utility server, LDC, DCU and substation are executed using machine equipped with a Quadcore Intel i5-4570 @3.20GHz CPU. The authentication and key-exchange protocol (refer to sec: 2) is implemented in C language. The IBE scheme used in the protocol is implemented using Pairing-Based Cryptography (PBC) library, which provides APIs to securely instantiate all bilinear pairing-related operations on the Barreto-Naehrig family of elliptic curves with, embedding degree 12 and a security level of 160 bits of finite field. The SKE scheme is realised using AES-128, hash function $\mathcal{H}$ using SHA-256 and hash function $\mathcal{H}'$ using the `element_from_hash` API of PBC library. The AES-128 and SHA-256 are implemented using Libgcrypt and executed in Raspberry Pi which replicates the meter setup in our proposed work. In order to execute the protocol, the smart meter needs to store executable files of size around 89.8kB. The time taken to generate the association data binding a smart meter and a substation is 0.124 seconds and the time taken to authenticate and perform key-exchange is 0.885 seconds. Besides, setting up the initial grid topology and posting the transaction takes 0.124 seconds and events such as link breakdown or rerouting takes around 0.009 seconds.

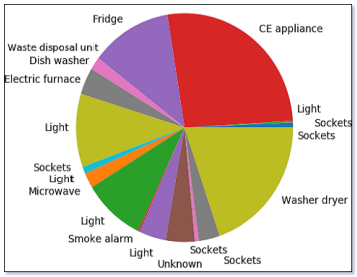
5.2 Attack source detection

To realise the attack source detection methodology as proposed in Sec. 4.5, we use the REDD data set [21] provided by Kolter and Johnson and Non-Intrusive Load Monitoring Toolkit (nilmtk) [22] to desegregate the power profiles to acquire the consumption of each appliance during a sampling period. However, the disaggregation capabilities of the NILM algorithms raise severe concerns for user privacy by leaking the power consumption status of individual appliance. Hence, in this work, we use the privacy preserved Smart metering scheme proposed by Barbosa *et al.* [23] that hides the usage patterns by providing differential

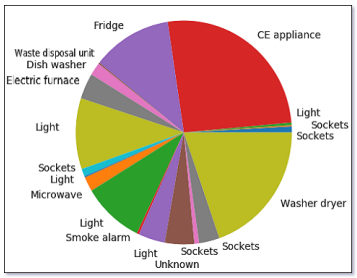
Fig. 11: Blockchain transactions given in Sec. 4.3 (a) InitTxn, (b) LinkDownTxn, (c) ConsumptionTxn, (d) NotifyLDCTxn

privacy guarantees for the appliances. Here, the Smart meters transmit masked measurements by adding noise generated by a distribution model. The maximum allowed error between the original and the masked power consumption over a billing period is bounded by a desired percentage, denoted by privacy level ϵ. Based on this, the variance of the normal distribution is calculated in such a way that the probability of obtaining the error within the desired value ϵ is very high, for e.g. 98%. If Laplace distribution model is used for the privacy, the magnitude of the Laplacian noise is determined by the scale parameter of the original distribution which can be calculated using the obtained variance and the total number of measurements. With the increasing value of ϵ, the privacy of resulted stream w.r.t the original power profile increases.

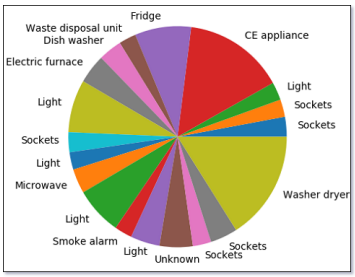
For the experimental results, We show the feasibility of our approach with different privacy levels (such as 0-5%) for the sampling period of 15 minutes. In our attack preparation, we consider the building number 3 of the REDD dataset. We have added $30W$ of extra demand for each of the channels presented in the building assuming that the adversary is synchronously controlling multiple loads at the same time. Fig. 12 shows the fraction of energy consumption of each appliance of building 3 with privacy level of 0%, 1% and 5%. Simultaneously, Fig. 13 demonstrates the fraction of energy consumption for the same set of appliances under the attack scenario mentioned earlier. It shows a significant deviation in appliance power consumption estimation compared with the normal condition. Fig. 14a shows the estimated power profile of the fridge on a particular date, 23rd April 2011, when 5% privacy level is applied. On the other hand, Fig. 14b shows the power profile of the fridge under the attacked scenario if the demand is manipulated (as discussed in Sec.3.1) throughout the whole day. It is clear from the graph that the magnitude of the power profile has got a shift of approximately 30 Watts. Without loss of generality, we consider that the tracing algorithm got triggered during the first reading stream on 23rd April 2011. Hence, we run the t-test between the received power profile and the estimated power profile



(a) No privacy

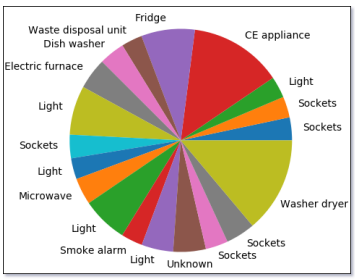


(b) 1% privacy level

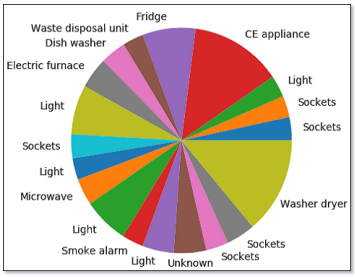


(c) 5% privacy level

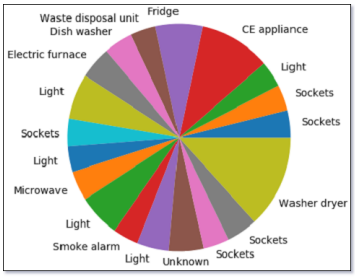
Fig. 12: Fraction of energy consumption of each appliance under normal operation



(a) No privacy



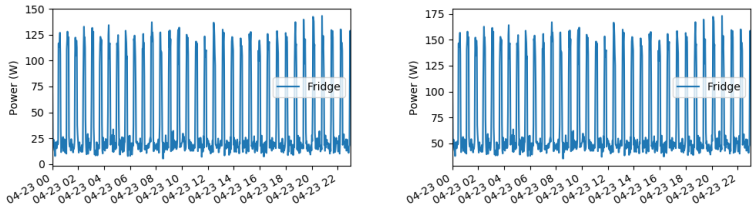
(b) 1% privacy level



(c) 5% privacy level

Fig. 13: Fraction of energy consumption after demand manipulation

of the fridge during that period. Tab. 1 shows calculated t-values for different privacy levels if $30W$ and $20W$ demand alteration is done during this sampling period. The expected value in the *T-Distribution Table* for 95% confidence level is around $1.960 - 1.980$. Tab. 1 displays that all the resulted t-values are very much higher compared to the expected value. Hence, the received power profile can be considered as manipulated for all of the above cases. Also, It reveals that the privacy level of the metering infrastructure does not significantly affect the outcome of our scheme.



(a) (b)

Fig. 14: Disaggregated power profile of fridge with 5% privacy on 23^{rd} April,2011 (a) in estimated scenario (b) after demand manipulation

Privacy Level (ϵ)	t-value	
	20W increase	30W increase
No privacy	6.04991	9.07487
1%	6.07545	9.11317
2%	6.11155	9.16732
3%	6.01353	9.02029
4%	5.81902	8.72853
5%	5.90373	8.85561

Table 1: T-test result between the fridge's original and attacked power profile for different privacy levels due to 30W and 20W demand alteration in every channel

6 Conclusion

In this paper we have demonstrated demand manipulation attacks through MadIoT, electric faults and false data injection that can inflict undesired load-shedding. Next we have proposed a PUF based power profile verification and attack source detection methodology without compromising the differential privacy guarantees. We have also addressed the issues related to transparency and accountability in Smart grid operations using a permissioned blockchain network. Finally, we have prototyped end-to-end security solution using Raspberry Pi, Artix 7 FPGA and blockchain based simulator. The experimental results show that the scheme takes significantly less execution time and memory footprint and makes it suitable for detecting any physical disturbance in the grid and preventing injection of any falsified data.

References

1. S. Soltan, P. Mittal, and H. V. Poor, "Blackiot: Iot botnet of high wattage devices can disrupt the power grid," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, W. Enck and A. P. Felt, Eds. USENIX Association, 2018, pp. 15–32.
2. B. Huang, A. A. Cárdenas, and R. Baldick, "Not everything is dark and gloomy: Power grid protections against iot demand attacks," in *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, N. Heninger and P. Traynor, Eds. USENIX Association, 2019, pp. 1115–1132.

3. S. Ruj and A. Pal, "Analyzing cascading failures in smart grids under random and targeted attacks," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, 2014, pp. 226–233.
4. D. Silvestre, J. P. Hespanha, and C. Silvestre, "Fault detection for cyber-physical systems: Smart grid case," in *23rd International Symposium on Mathematical Theory of Networks and Systems (MTNS)*. IEEE, 2018, pp. 475–481.
5. F. Mohammadi, G.-A. Nazri, and M. Saif, "A fast fault detection and identification approach in power distribution systems," in *2019 International Conference on Power Generation Systems and Renewable Energy Technologies (PGSRET)*. IEEE, 2019, pp. 1–4.
6. S. Ali, H. Mansoor, N. Arshad, and I. Khan, "Short term load forecasting using smart meter data," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 419–421.
7. Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09, 2009, p. 21–32.
8. J. Zhao, G. Zhang, M. La Scala, Z. Y. Dong, C. Chen, and J. Wang, "Short-term state forecasting-aided method for detection of smart grid general false data injection attacks," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1580–1590, 2017.
9. H. Wang, J. Ruan, G. Wang, B. Zhou, Y. Liu, X. Fu, and J. Peng, "Deep learning-based interval state estimation of ac smart grids against sparse cyber attacks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4766–4778, 2018.
10. 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 Trans. Dependable Secur. Comput.*, vol. 16, no. 3, pp. 424–437, 2019.
11. U. Chatterjee, R. Sadhukhan, V. Govindan, D. Mukhopadhyay, R. S. Chakraborty, S. Pati, D. Mahata, and M. M. Prabhu, "PUFSSL: an openssl extension for PUF based authentication," in *23rd IEEE International Conference on Digital Signal Processing, DSP 2018, Shanghai, China, November 19-21, 2018*. IEEE, 2018, pp. 1–5.
12. H. Boyapally, P. Mathew, S. Patranabis, U. Chatterjee, U. Agarwal, M. Maheshwari, S. Dey, and D. Mukhopadhyay, "Safe is the new smart: Puf-based authentication for load modification-resistant smart meters," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020.
13. H. Zhang, J. Wang, and Y. Ding, "Blockchain-based decentralized and secure keyless signature scheme for smart grid," *Energy*, vol. 180, pp. 955–967, 2019.
14. F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, and V. Sassone, "A blockchain-based infrastructure for reliable and cost-effective iot-aided smart grids," in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, 2018, pp. 1–6.
15. J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, and G. Dong, "Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid," *IEEE Access*, 2018.
16. Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

17. K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7992–8004, 2019.
18. A. Anwar, A. N. Mahmood, and M. Pickering, "Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 58 – 72, 2017.
19. "Inter Planetary File System," https://ipfs.io/, Nov 2019, [Online].
20. 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*, 2018, pp. 1504–1507.
21. J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, vol. 25, no. Citeseer, 2011, pp. 59–62.
22. N. Batra, J. Kelly, O. Parson, H. Dutta, W. Knottenbelt, A. Rogers, A. Singh, and M. Srivastava, "Nilmtk: an open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265–276.
23. P. Barbosa, A. Brito, and H. Almeida, "A technique to provide differential privacy for appliance usage in smart metering," *Information Sciences*, vol. 370, pp. 355–367, 2016.