Hardware Assisted Smart Grid Authentication

Mohammad Ebrahimabadi, Mohamed Younis, Naghmeh Karimi
Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County (UMBC)
Email:{ebrahimabadi, younis, nkarimi}@umbc.edu

ICC 2021 - IEEE International Conference on Communications | 978-1-7281-7122-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICC42927.2021.9500247

***Abstract*— A Cyber-Physical System (CPS) refers to the interconnection of control (actuation), computational nodes and sensors, in order to manage physical processes. In recent years, the CPS design methodology has been adopted in several large-scale infrastructures such as smart power grids. Given the application criticality, sustaining the security of these systems is of utmost importance. One of the major security goals is to protect CPS against impersonation, where an adversary intends to manipulate the system state by sending erroneous data that appears to be reported by one of the system nodes, e.g. PMUs of a power grid. This paper proposes a novel hardware-assisted authentication scheme to counter such a threat, by exploiting imperfections that occur in the manufacturing process of integrated circuits. In essence, the proposed scheme associates a fingerprint for each system node so that the authenticity of the data source could be verified. In addition, the paper tackles the threat of message replay where the adversary re-transmits a legitimate message so that the system factors in outdated rather than fresh sensor measurements. This paper thwarts such a replay attack by leveraging the synchronized clocks across the CPS nodes, e.g., based on GPS; the idea is to employ a combination of time-stamp signatures and hardware fingerprints. Our proposed schemes can also detect and prevent data forgery, and Sybil attacks. The viability and performance of the proposed schemes are validated through analysis and prototype implementation.**

I. Introduction

Cyber-attacks that disrupt the operation of the power grid could result in major economic losses and could constitute a national security threat. An adversary may target the communication network or the physical infrastructure of power grids to maliciously interrupt its operation [1]–[3]. Fundamentally, the design paradigm of a CPS and in particular smart grids is to conduct sensing and actuation in a distributed manner where tasks are split among multiple modules and communication among the various modules is used to ensure coordination. Therefore, cyber attackers could be targeting: (i) the computing platform of the individual modules to disrupt their operation or corrupt their results, and/or (ii) the communication links to hinder or degrade the inter-module coordination. In this research, we opt to tackle these cyber attacks, particularly, impersonation attacks where an adversary intends to alter the system state by sending erroneous data that appears to be reported by one of the system nodes. Moreover, we tackle the message replay attacks in which the adversary re-transmits a legitimate message so that the system factors in outdated rather than fresh sensor measurements.

In smart grids, Phasor Measurement Units (PMUs) represent sensing modules while control stations play the role of actuators. PMUs monitor the load in the various parts of the grid to detect any overload conditions. The control stations reconfigure the grid to sustain load stability and avoid service interruption. Basically, control stations receive periodic assessment from the individual PMUs, and adjust the power flow accordingly. Applying the right adjustment requires a current and consistent status of load in the various parts of the grid. Therefore, all PMUs are equipped with Global Positioning System (GPS) receivers in order to be synchronized to a global time reference. PMUs are often associated with the transmission lines from the various power generators. However, given the advantages of synchronized PMUs, smaller units, namely μPMU, are being deployed in power distribution lines in order to enable fine-grained control and to provide detailed system-wide view of the grid status. The increased PMU count also has motivated the introduction of concentrators that collect and aggregate the PMU readings. In practice, the concentrators, which often referred to as Data Aggregation Units (DAUs), receive data from multiple PMUs and align such data based on the measurement time tags, on route to the control station.

To ensure data integrity and the authenticity of its source, applying symmetric cryptographic algorithms using shared keys has been common in current designs [4]. While encrypting data payloads using a shared key is invaluable and also supports confidentiality, it does not prevent data manipulation at the source module or revelation of the key through cryptanalysis. Countering that through frequent key update requires the implementation of a key management protocol which imposes more communication and processing overhead. Meanwhile, the use of asymmetric cryptographic certificates has been widely accepted as a robust mechanism to authenticate the identity of data sources and to guard the inter-connected modules against data forgery attacks [5]. However, such an authentication approach is computationally demanding. In addition, key management is cumbersome and requires access to a remote trusted authority, something the PMUs are not conventionally designed to handle.

To address the aforementioned shortcomings, this paper proposes a Hardware-based mechanism for Authentication and data Integrity in smart power Grids (HAIG). HAIG employs Physically Unclonable Functions (PUFs) [6] to associate unique hardware-based identifiers to the participating PMUs in order to enable effective protection against contemporary security threats such as impersonation, data manipulation, and message replay. In practice, PUFs operate based on unintentional variations that occur in the fabrication process of the integrated circuits. For example, in delay-PUFs, these variations cause signals which follow similar paths in the design to experience slightly different propagation delays in the different chips that realize such a design. Thereby, the response of each PUF to the same input (referred to as challenge) varies among similar chips [7]. These unique signatures will be used to prevent data forgery, impersonation, Sybil, and replay attacks in power grids. We also benefit from the combination of PUF-based hardware fingerprints and timestamp-based (GPS enabled) signatures to counter message replay attacks. The contributions of this paper are as follows:

- Developing a hardware-assisted authentication mecha-

978-1-7281-7122-7/21/$31.00 ©2021 IEEE

nism for mitigating data forgery, impersonation and Sybil attacks on grid nodes, e.g., PMUs. Our solution enables message recipients to validate the identity of the source of the wireless transmission;
- Developing a protocol for thwarting message replay attacks by combining the hardware fingerprints with timestamp-based signatures;
- Developing a moving target defense strategy that prevents an attacker from uncovering the security primitives by eavesdropping on transmissions for extended duration;
- Validating the proposed countermeasures using prototype implementation and testing.

II. Threat Model

In this paper, the adversary opts to mislead the DAU to provide inaccurate/wrong aggregated data. Provision of such data can cause the controller to make wrong decisions in regulating electrical loads, and consequently can result in failures in the power grid network. To achieve this objective, the adversary opts to alter the PMU measurement data maliciously. Our assumption is that the DAU is trustworthy, and that the PMUs cannot be tampered. We consider two threat scenarios: In the first one, the adversary deploys malicious nodes to replicate the functionality of PMUs, and impersonate existing nodes in the grid network. The second scenario occurs when the adversary eavesdrops on the communication link between PMU and DAU, aiming at intercepting transmissions and launching a Man-In-The-Middle attack. This paper targets these attacks:

- **Impersonation attack:** it occurs when a malicious PMU tries to identify itself as a legitimate unit in the grid to be able to send erroneous measurements to DAU;
- **Sybil attack:** it reflects the case when the adversary impersonates multiple PMUs. It can be considered as a variant of the impersonation attack for which the adversary not only gets the DAU to suspect data manipulation, e.g., due to failed consistency checks, but also confuses the DAU about the specific PMU to suspect.
- **Data forgery attack:** this is a means for causing grid instability where wrong data is pushed to the DAU. Such an attack can be considered as an instantiation of man-in-the-middle or replicated PMU scenarios.
- **Message replay attack:** it is launched by re-transmission of a valid data message from PMU at a later time. Message replay can also be pursued by the adversary to authenticate a malicious node by re-transmitting the authentication message of a legitimate PMU.
- **Selective forwarding attack:** the adversary deprives the DAU from some of the data collected by PMUs so that the grid does not respond to controllable events in a timely manner and hence becomes unstable.

III. Literature Review and Preliminaries

Related Work: Traditional cryptographic schemes based on Public Key Infrastructures (PKI) have been widely used in recent years to provide authentication services for various applications [8]. However, the inflexibility as well as time and computational complexity of these methods make them unsuitable for cyber physical systems like the smart power grid. To alleviate the computational load of PKI, in multiple messages are combined and authentication is conducted once for all of them. However, message buffering not only increases the storage demand but also could lead to violating the data delivery latency requirement, which is very critical for systems involving real-time control such as the smart grid.

To avoid the complexity of asymmetric cryptography used in PKI, lightweight symmetric methodologies is pursued, e.g., the One-Time Signature (OTS) scheme employs cryptographic one-way functions, e.g., SHA-1 [9]. Despite the computationally efficient process, OTS-based schemes suffer from either high signature sizeor large key sizes. Moreover, in these methods the public key needs to be distributed periodically. TESLA pursues symmetric cryptography for authentication, which results in reduced computation and communication overhead [10]. TESLA uses one-way hash functions to generate keys; each message is encrypted with a key that is revealed in the subsequent message. Like HAIG, the key validity is restricted in time [10]. However, the latency in revealing the key risks meeting the timeliness requirement for CPS and would not make TESLA suitable for power grid applications.

Hardware-based authentication schemes can be categorized into implicit and explicit [11]. The Trusted Platform Module (TPM) is a prominent example of explicit authentication schemes. TPM is a crypto-processor that validates the authenticity of the software and hardware of a device during boot-up [12]. The main deficiency of TPM and other explicit schemes is the lack of ability for preventing runtime attacks [13]. Such shortcoming makes explicit schemes unfit for smart grid applications. PUF-based authentication techniques, on the other hand, are representatives of the implicit category [14]–[16]. However, existing PUF-based schemes suffer from security vulnerabilities, implementation complexity, or large challenge bit pattern requirement [17]. HAIG fills the technical gap and enables effective and efficient authentication and data integrity services for cyber-physical systems in general and the smart power grid in particular.

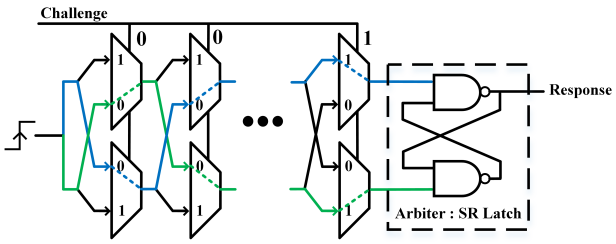

Fig. 1: Illustrating the design of an arbiter-PUF.

Preliminaries: PUFs are classified as weak or strong based on whether their challenge response space is small or large, respectively. Strong PUFs are often used for authentication protocols, while weak PUFs are deployed for key generation in cryptographic chips to avoid the key storage in insecure nonvolatile memories [18]. An arbiter-PUF is one of the popular strong PUFs and is used in this paper. Consisting of a pair of delay chains per challenge; when queried, it generates one response bit for the related challenge [6]. This PUF operates based on the process-variation that induces race between two identical paths (shown in green and blue in Fig. 1). The race corresponds to the difference in signal propagation delay on these two paths; such difference affects the value that is latched by the arbiter. In fact, only the sign of this difference (and

not the exact value) is important. This sign (extracted by the arbiter) presents the PUF identifier (response). The arbiter can be realized as a simple SR latch implemented by two NAND gates. *In this paper, without loss of generality, we deploy arbiter-PUFs. However, the proposed schemes are applicable to other strong PUFs*. As mentioned, weak PUFs are not used for authentication and are more suitable for key generation.

IV. Detailed HAIG Approach

Our proposed security protection mechanism benefits from the unclonablity of PUFs, where the response of a PUF to each challenge is device specific and varies from one PUF to another. Accordingly, a PUF response to a certain challenge bit pattern constitutes a unique signature for the circuitry the PUF is embedded in. Hence, we incorporate a strong PUF, e.g., an arbiter PUF, in each PMU module and leverage the PUF unique signature in authenticating the PMU modules as well as securing data transmission from the PMUs to DAU. The details are discussed below. In this study, an arbiter-PUF is used. However, any other strong PUFs can be deployed too.

A. Hardware Fingerprinting

Providing incorrect measurements to the DAU in a power grid may result in taking inappropriate decisions that can lead to catastrophic consequences. To protect the power grid, we incorporate security primitives to ensure the authenticity of the data sources, namely the PMU modules, and guard the integrity of the transmitted data. Basically, we embed a PUF in each PMU module during design. At the time of system integration, each PUF is characterized through the generation of a subset of its Challenge Response Pair (CRP) combinations. Such a subset is associated with the corresponding PMU and stored at the DAU. To do so, a set of challenge bit patterns will be applied to the PUF and the corresponding responses will be tabulated. Such a table will serve as a placeholder of valid PMU signatures. We note that the table size, i.e., the number of used CRPs, is subject to trade-off as discussed below.

In HAIG, the PMUs are authenticated periodically by the DAU. The frequency is a system level parameter that naturally depends on the criticality of the CPS application, the threat that it is subject to, and the performance requirements. We use Δ to denote the period between authentication requests; HAIG leaves it up to the grid administrator to decide appropriate setting for Δ. In each authentication round, the DAU sends a distinct challenge to each PMU P_i. The latter applies the challenge to its embedded PUF and replies the DAU by providing the PUF response. P_i will be successfully authenticated if its response to the challenge bits matches the corresponding entry in the CRP table of P_i that is pre-stored at the DAU. The handshaking between each PMU and DAU to exchange challenge/response, size and number of the challenges used in each authentication round, and size of the PUF itself, affect the level of security and associated overhead, as analyzed below.

To prevent modeling the PUF via exhaustive enumeration of its CRPs, HAIG employs a PUF with a large bit pattern. In this paper, we embed a 64-bit PUF in each PMU module (i.e., the size of the challenge bit pattern is 64 bits); thereby the huge number of CRPs, specifically, 2^{64}, makes it impractical to predict the PUF response for a certain challenge. Moreover, as will be discussed in Section IV-B, modeling the deployed PUF through a subset of its challenge response pairs by using machine learning schemes (e.g., [19]) isn't possible since we do not exchange the full challenge bits. To enhance the security, we recommend using large PUF response; in our implementation the PUF has 64 bits of response. Obviously, a larger PUF yields more distinct responses to challenges (lower collision probability) and consequently increases the robustness of the authentication process. On the other hand, a larger PUF requires more overhead in terms of logic and area. We will discuss the imposed overhead in Section V. Alternatively, one could increase the frequency of authentication rounds or use multiple challenges per round; however, this will increase the security-related communication and processing overhead.

To reduce the overhead imposed on the DAU for storing CRPs for each PMU, all possible CRPs are not stored, instead a subset of those pairs will be used. The size of the subset is subject to trade-off. If the cardinality of the selected subset is small, an adversary who can intercept the transmissions between PMU and DAU can launch an impersonation attack by replaying the eavesdropped response the next time a challenge is reused for authentication. To prevent such a threat, HAIG exploits the timestamp provided by GPS receivers on the PMUs and DAU. A GPS receiver is usually incorporated in order to synchronize the grid nodes to a global time reference and enable the correlation of the collected measurements. Specifically, HAIG considers a combination of PUF-based hardware fingerprints and timestamp-based (GPS enabled) signatures in authenticating PMUs as we explain next. Note that GPS spoofing attack is out of the scope of this paper.

B. Hardware- and Time-based Authentication

In HAIG, a PMU P_i is authenticated through the response of its PUF to a challenge. However, instead of feeding the PUF with only the bit string C_k received from DAU, HAIG appends a timestamp to C_k before applying to P_i's PUF. In other words, the input to the PUF will be the challenge bits sent by DAU to P_i concatenated with a timestamp that is derived from a real-time clock at P_i. The idea is to have a part of the input of the PMU's PUF to be implicitly agreed upon between the DAU and PMU rather than being sent within the authentication request from the DAU. In HAIG such implicit part is derived from the real-time clock; obviously there has to be a common (agreed upon) time reference since the DAU should be able to validate the response. Since in smart grids all modules are time-synchronized using GPS, the timestamp will be consistent at all PMUs and DAU; hence the DAU will be able to validate P_i's response. Fig. 2 shows a high level overview of the proposed authentication protocol. In this figure, C_k, T_j, and $(R_{k,j})_{PUF}$ denote the challenge bitstream k sent by DAU, the time stamp extracted from the GPS when C_k was received by PMU, and the PMU's PUF response to $C_k \| \mathrm{T}_j$ (i.e., concatenation of C_k and T_j), respectively. Upon receiving $(R_{k,j})_{PUF}$ from P_i, the DAU will check its database to ensure that the received response $(R_{k,j})_{PUF}$ matches the stored response for $C_k \| \mathrm{T}_j$, (i.e., $R_{k,j}$) in its database for P_i.

The incorporation of timestamps strengthens the resilience of HAIG to security attacks. In essence, varying the challenge bits over time makes the PUF's response changes even when a similar challenge C_k is received from DAU. Thus, the adversary will not be able to infer the CRP even if C_k and the response are intercepted. In essence, C_k could be mapped

to multiple responses, thereby modeling the PUF will fail. Moreover, a replay attack can be thwarted since the response in the replayed message will not correspond to the current time, as we further discuss later in this section. The interesting question is how to decide on the size of C_k, i.e., how many bits sent from DAU, and the resolution of the timestamp which would affect the number of bits for T_j. In fact, the PUF configuration for a PMU, i.e., the size of the challenge PUF_C and response PUF_R, and the corresponding CRP table $Table_{CR}$ at the DAU, are all subject to trade-off between security and overhead as we analyze below.

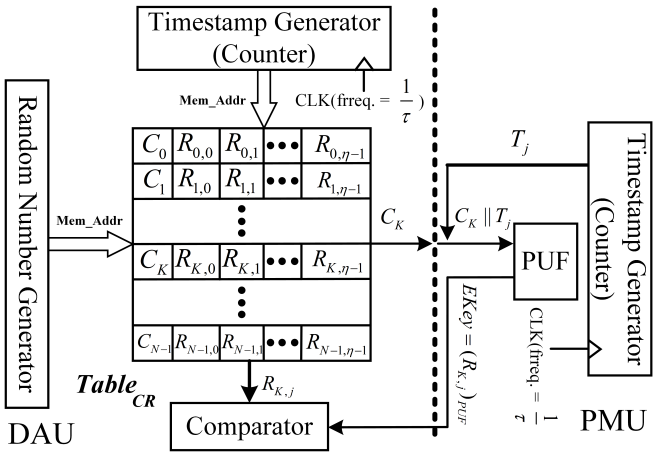

Fig. 2: PUF Based Authentication

Let L_{PUF_C}, and L_{PUF_R} be the size of the PUF challenge and response in bits. Assume also that L_C is the size of challenge bitstream sent by DAU (which is smaller than L_{PUF_C}). Thus, the size of the timestamp T is L_T and equals to (L_{PUF_C}-L_C). As shown in Fig. 2, the DAU includes $N \times (L_C + \eta L_{PUF_R})$ CRP bits in the $Table_{CR}$, in which the first element of each row is L_C bit and each of the other elements in the row is L_{PUF_R} bits long. In essence, N denotes the cardinality of the challenge subset that is going to be used during the PMU lifetime for its authentication. In other words, $N \leq 2^{L_C}$. In addition, η is the number of different timestamps we use for authentication purposes. When the PMU P_i receives a challenge C_k, it appends the current timestamp (T_j) to it, feeds the PUF with this combination, extracts the PUF response $(R_{k,j})_{PUF}$ and sends this response to DAU. Upon receiving P_i's reply, the DAU compares the received bit string with the pre-stored value in the row of C_k and column of T_j in $Table_{CR}(P_i)$. Note that T_j is the current time stamp that DAU itself extracts based on its GPS which is synchronized with the time stamp generated by the counter circuitry embedded in the PMU (Fig. 2).

Obviously there could be a difference between the time at which P_i has generated the PUF response and the time when the PMU message arrives at the DAU. If significant, such a difference could cause the DAU to match the response to the wrong entry in the row of C_k in $Table_{CR}(P_i)$. Therefore, the DAU may need to match with few entries in the row of C_k, depending on the time stamp resolution (i.e., measurement period of τ) and the data transfer latency between the PMU and DAU as will be discussed further in Section V-C. For example, if the resolution of the counter is one second, with communication latency in the order of milliseconds, the time difference between the PMU transmission and DAU reception could be negligible at the level of seconds. In this case, we only need to check two entries of the DAU table as will be discussed in Section V-C. Generally, the resolution of a timestamp T will be subject to trade-off. GPS-based synchronization is at least accurate to the millisecond level while the accuracy achieved by message-exchange clock synchronization protocols depends on the proximity, and the quality and bandwidth of the communication links. On the other hand, using high-resolution timestamps will grow the size of the CRPs stored in DAU. In Section V we analyze the storage overhead and the frequency of the authentication process and provide guidelines on how to set up the various parameters.

C. Securing Data Transfer

Secure transfer of PMU measurements and protection against data forgery and replay attacks are very important requirements in smart grids. In this section, we discuss how the proposed HAIG approach for PMU authentication can also be employed to ensure data integrity and freshness. Figure 3 provides an overview of the payload formation of a data packet in HAIG. The PUF block shown in this figure refers to the PUF circuity shown in Fig. 2, which is to reside in each PMU.

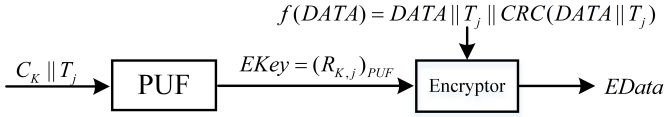

Fig. 3: PUF Based Key Generation and Payload Encryption.

Basically, the data is encrypted prior to being transmitted from PMU to DAU. As illustrated in the figure, HAIG uses the PUF response as a data encryption key to avoid the communication overhead for establishing symmetric encryption keys and the computational overhead for applying asymmetric cryptography systems. Our approach is very powerful as the encryption key will vary per transmission and thwarts any cryptoanalysis attempt by the adversary. Moreover, the incorporation of a timestamp in the challenge bit enables the DAU to judge the freshness of the data, which is highly critical for power control in the grid. Since the DAU (or the grid controller) uses the collected measurements to determine the state of the grid and apply adjustments as needed, using old data can lead to incorrect state and in turn wrong decisions. HAIG enables the DAU to assess data freshness through: (i) appending a timestamp to the PMU data prior to transmission, and (ii) using such a timestamp to generate the encryption key, as noted above and shown in Fig. 2. The timestamp generator (timer) will serve as a monotonic counter to reflect the order within a sequence of transmitted data packets.

The encrypted data ($EData$) is found based on Eq. (1) where a Cyclic Redundancy Check (CRC) is appended to the data as an integrity check to inform the DAU about any data corruption related to transmission errors. As discussed in Section IV-B, a challenge is sent from the DAU to PMU every Δ minutes. This challenge is first combined with the timestamp T_j generated by the PMU counter based on its GPS (Fig. 2), and then this combination ($C_k \parallel T_j$) is fed the PUF. The PUF response is used as a key, $EKey$, for encrypting the data before sending it to the DAU. Note that C_k is received every Δ minutes. However, as Fig. 2 shows, T_j is changed with

a rate of $1/\tau$, i.e., a new timestamp and in turn a new $EKey$ is generated for each measured data. Using a different key for each data transfer makes the system secure against several type of attacks including replay attack, data forgery, etc. We discuss the resiliency of HAIG against these attacks in Section V-A. Note that HAIG doesn't need to send a challenge to PMU for every measurement, as the timestamp which is generated locally is used as part of the PUF challenge. This reduces the communication traffic between the DAU and PMUs.

$$\begin{aligned} &f(DATA) = DATA \parallel T_j \parallel CRC(DATA, T_j) \\ &EKey = PUFResponse(C_k \parallel T_j) \\ &EData = [f(DATA)]_{EKey} \end{aligned} \tag{1}$$

Since C_k is initially generated by the DAU and also T_j is predicted using the DAU's clock, the DAU can determine the response of the PUF ($Ekey$), similar to the authentication process. Therefore, upon receiving $EData$, the DAU will be able to decrypt and extract $DATA$. Incorporating the CRC enables the detection of corruption that could have happened to the data due to transmission errors. Basically, when a DAU determines $EKey$ and decrypts $EData$, it checks the integrity of $f(Data)$ using the CRC. On the other hand, having T_j as part of $f(Data)$ makes HAIG resilient against data forgery, replay attacks, and any PUF-related errors, as discussed in Section V. Note that errors in the PUF response may be related to circuit-level noise; such noise is sporadic in nature [20]. HAIG assumes that the responses stored in the DAU table have been confirmed through multiple measurements at the time of PMU installation in the grid, and thus do not suffer from noise effects. During HAIG operation, if noise affects the PUF response ($EKey$), the DAU will fail to validate the data integrity and request a re-transmission from the PMU. The PMU then will regenerate a new $Ekey$ and form a new packet payload for the data. Note that based on our lab experiments as well as published studies, e.g., [20], the noise probability is insignificant$\approx 0.2\%$ (as will be discussed in Section V-C); thereby the probability of re-transmission request is so low.

Finally, we discuss the interplay between the authentication and data transfer mechanisms in HAIG. Given how the encryption key is generated, one can easily conclude that the PMU is implicitly being authenticated through the transmission of every data packet. Thus, with a measurement rate of $1/\tau$ that exceeds the desired authentication rate of $1/\Delta$, there is no need for explicit packet authentication. This not only simplifies HAIG's implementation but also facilitates integration with standard protocols, e.g., IEEE C37 and IEC TR 61850. Indeed, HAIG will be adopted at the network layer of the protocol stack. However, if $\Delta<\tau$, explicit authentication message is needed. HAIG supports this case via pre-determined data pattern agreed upon by the DAU and PMU so that the data will be discarded by the DAU while the PMU gets authenticated.

V. Security and Performance Analysis

A. Protecting Against Attacks

HAIG guards the power grid against the following serious attacks that opt to disrupt the operation and cause instability:

1) Impersonation and Sybil Attacks: In HAIG, the challenge bits are augmented with a local timestamp to generate $EKey$ which serves as a crypto identity for the PMU and varies every τ time units. Moreover, the challenge bits are changed by the DAU every Δ time units. Such a rapid variation rate makes it impractical for an adversary to apply cryptanalysis to uncover $EKey$ and reuse it before a new $EKey$ is generated. Obviously, the values of τ and Δ, and the size of the timestamp T (L_T) are all influential; Section V-B provides guidelines on the appropriate settings of these parameters. In sum, even if the adversary can infer $EKey$ of a PMU P_i, i.e., PUF response at time T_j, P_i cannot be impersonated at any other time. HAIG also prevents Sybil attacks, in which an adversary claims multiple valid identities, since an adversary cannot even impersonate a single PMU.

2) Data Forgery Attack: As shown in Fig. 3, HAIG prevents data manipulation by: 1) appending the packet payload $f(DATA)$ with a CRC for the measurement data and the current timestamp, and 2) encrypting the packet payload using time- and hardware-based signatures. If the data payload is manipulated, the CRC will not be consistent with the data and timestamp, and consequently the attack will be detected.

3) Message Replay Attack: This attack is tackled via using the timing-based counter shown in Fig. 2. As the packet payload and encryption key are functions of time, a replayed message will be rejected by the DAU if it arrives after the tolerable communication latency. Recall that when determining which entries in $Table_{CR}(P_i)$ are considered in decrypting the packet payload, HAIG factors in the delay between the time a PUF response is generated at the PMU P_i and the time a packet is received at the DAU. Thus, replaying a message within such time frame can be successful, yet has no negative impact. Meanwhile, replaying the message after such delay will be detected since the DAU will not use the right entry in $Table_{CR}(P_i)$ and the payload will fail the integrity check.

B. Size of CRP Database Stored in DAU

HAIG makes the grid resilient to attacks since only a part of each challenge is sent by the DAU, while the other portion is implicitly inferred. In this section, we analyze the implications of the parameter settings on the communication security and storage overhead at the DAU. The analysis is based on the assumption that the PMU sends measurements every τ millisecond, i.e., at the rate of $1/\tau$ per millisecond), and the DAU sends a new challenge to the PMU (for re-authentication) every Δ minutes, the timestamp is generated via a modulo-Ω counter, clocked with frequency of $1/\tau$, where $\Omega = H/\tau$, and H is the range of timestamps (in minutes) and at least should be equal to 2Δ to ensure that consecutive challenges are combined with different timestamps. The greater the value of H is, the more distinct timestamps are, and in turn the lower possibility of impersonation is. Moreover, each PUF challenge is L_{PUF_C} bits and each PUF response is L_{PUF_R} bits. Thereby, in order to prevent impersonation attacks realized via replaying the intercepted CRPs for at least M months, Eq. 2 shows the memory size, Mem_Size, in bits, needed at DAU to store the PUF CRPs for each PMU, i.e., size of $Table_{RC}$.

$$\begin{aligned} &N(\#\ of\ Rows) = M \times 30 \times 24 \times 60/\Delta \\ &\eta(\#\ of\ Columns) = \Omega = H/\tau \\ &L_C = L_{PUF_C} - log_2\eta \\ &Mem\_size = Row \times Column \times L_{PUF_R} + Row \times L_C \end{aligned} \tag{2}$$

HAIG's performance is characterized by the PUF size, measurement rate, CRP subset size, etc. To show how to determine the setting of these parameters, we discuss an example considering a PUF with 64 challenge bits, and 64 response bits (i.e., L_{PUF_C}=64, and L_{PUF_R}=64). A typical measurement rate made by PMUs is once every $17ms$ to $100ms$ [21]. Let's assume that the measurement rate is 20Hz, i.e., $\tau = 50ms$. Let's also assume that the DAU sends a new challenge to PMUs every 10 minutes, i.e. Δ=10 minutes, and the timestamp period is 1 hour, i.e., H=60 minutes. Thus, we need a 72,000 modulo counter clocked every 50ms. To avoid repeating the use of a CRP for at least for 1 month, the size of needed memory at the DAU would be 2.3 GByte. Note that in this case, we only need a 17 bit counter in each of DAU and PMU to generate timestamp based on GPS timing. Obviously, using a smaller PUF diminishes the storage overhead, yet at the price of decreased attack resilience. Such a trade-off is typically faced in practice as the increased computational capacity boosts the threat to cryptanalysis, motivating growth in key sizes and application of complicated key management processes. In that regard, HAIG is very powerful in countering the threat of cryptanalysis since it enables the use of distinct key per data transmission and further avoids reuse of keys (with increased storage overhead).

C. System Overhead and noise effects

Communication Overhead: Assume that data transfer from PMU to DAU takes at most α millisecond. In this case, DAU should check $\lceil \frac{\alpha}{\tau} \rceil + 1$ locations in its table for data extraction or device authentication. Thereby, the DAU decryption time is affected by the communication delay. Moreover, in this research, We follow the IEEE C37.118.2 protocol [22], which is a popular standard in PMU applications. Based on this standard, the typical packet size is 816 bits (including 352 bit of payload and 464 bits of header). Based on Fig. 3, HAIG includes time stamp T_j as well as the CRC bits to the payload. This results in a negligible packet size overhead. In our example, the overhead would be $\frac{8(\text{for CRC})+17(\text{for timestamp})}{816} \approx 3\%$.

Power and Area Overhead: To evaluate the overhead of applying HAIG, we have implemented the proposed circuit in Fig. 2 using Xilinx Artix 7 FPGA. The circuit includes a 64-bit PUF (with 64-bit response bits), a 17-bit counter, and a small controller. We then extracted the area and power consumption of this circuit via Xilinx Vivado. The power consumption is highly negligible, ≈6mW. The required hardware includes 128 2-bit multiplexers to generate each response bit, totally 128×64 = 8192 multiplexers. Such hardware overhead is reasonable compared to the PMU size itself. Note that the PMU circuit delay is not changed in this method as PUFs are isolated from the PMU critical path.

Noise Effects: HAIG is also robust against communication noise. Communication between PMUs and DAU is often over wireless links. Radio signal interference could hinder the correct delivery of messages. Mitigation of such interference is usually handled at the level of link and network layer protocols. Yet, HAIG enables increased protection by safeguarding the integrity of the data payload. As mentioned earlier, HAIG incorporates a CRC code within the data payload of each packet. This CRC is not only a function of the PMU measurement but also the time of packet formation. Thus, any transmission error can be detected when it skips protocol level protection. In other words, HAIG protects data integrity and freshness against both adversary attacks and radio interference.

VI. Conclusion and Future Directions

We presented a novel authentication scheme for smart grids. The PUF circuitries are deployed at grid nodes such as PMUs to generate hardware based signatures. In addition, our scheme benefits from GPS receivers residing in the grid nodes to extract the time-based fingerprints. The combination of hardware-based signatures and time-varying fingerprints are used to generate symmetric cryptographic keys to encrypt the exchanged data messages among nodes. We have shown that the proposed authentication scheme protects the smart grid against impersonation, data forgery, message replay and Sybil attacks; any of which threatens the security of the power grid and results in taking inappropriate decisions that could lead to catastrophic consequences. The hardware and communication overhead of the proposed scheme are shown to be negligible.

References

[1] H. Gharavi, H.-H. Chen, and C. Wietfeld, "Guest editorial special section on cyber-physical systems and security for smart grid," *IEEE Trans. on Smart Grid*, vol. 6, no. 5, pp. 2405–2408, 2015.

[2] N. Komninos et al., "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014.

[3] Y. Mo et al., "Cyber–physical security of a smart grid infrastructure," *Proc. of the IEEE*, vol. 100, no. 1, pp. 195–209, 2011.

[4] M. A. Ferrag et al., "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, 2017.

[5] Y. Challal et all., "A taxonomy of multicast data origin authentication: Issues and solutions," *IEEE Comm. Surveys & Tutorials*, vol. 6, no. 3, pp. 34–57, 2004.

[6] G. E. Suh and S. Devadas, "Physical Unclonable Functions for device Authentication and secret key generation," in *DAC*, 2007, pp. 9–14.

[7] M. Ebrahimabadi et al., "A novel modeling-attack resilient arbiter-puf design," in *VLSID*, 2021.

[8] N. Hong, "A security framework for the internet of things based on public key infrastructure," in *Advanced Materials Research*, vol. 671. Trans Tech Publ, 2013, pp. 3223–3226.

[9] Q. Wang et al., "Time valid one-time signature for time-critical multicast data authentication," in *IEEE INFOCOM*, 2009, pp. 1233–1241.

[10] Y.-S. Chen et al., "Broadcast authentication in sensor networks using compressed bloom filters," in *Distributed Computing in Sensor Systems*, 2008, pp. 99–111.

[11] M. El-hajj et al., "A survey of internet of things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, pp. 1–43, 2019.

[12] D. Chakraborty and S. Bugiel, "SimFIDO: FIDO2 user authentication with simtpm," in *CCS*, 2019, p. 2569–2571.

[13] B. Kauer, "Oslo: Improving the security of trusted computing." in *USENIX Security Symp.*, vol. 24, 2007, p. 173.

[14] M. Aman et al., "Position Paper: Physical Unclonable Functions for IoT Security," in *int'l W. on IoT Privacy, Trust, and Sec.*, 2016, pp. 10– 13.

[15] U. Chatterjee et al., "A PUF-based secure communication protocol for IoT," *ACM Trans. on Embedded Computing Systems (TECS)*, 2017.

[16] S. U. Hussain et al., "Shaip: Secure hamming distance for authentication of intrinsic PUFs," *Design Automation of Electronic Systems*, 2018.

[17] J. Delvaux et al., "Secure lightweight entity authentication with strong PUFs: Mission impossible?" in *CHES*, 2014, pp. 451–475.

[18] T. Idriss et al., "A PUF-Based Paradigm for IoT Security," in *World Forum on Internet of Things (WF-IoT)*, 2016, pp. 700–705.

[19] U. Rührmair et al., "PUF modeling attacks on simulated and silicon data," *TIFS*, vol. 8, no. 11, pp. 1876–1891, 2013.

[20] U. Chatterjee et al., "Building PUF Based Authentication and Key Exchange Protocol for IoT Without Explicit CRPs in Verifier Database," *IEEE TDSC*, vol. 16, no. 3, pp. 424–437, 2019.

[21] S. Kumar et al., "Performance monitoring of a PMU in a microgrid environment based on iec 61850-90-5," in *AUPEC*, 2016, pp. 1–5.

[22] J. Carroll et al., "A comparison of phasor communications protocols," 2019, Tech. Rep. , US. Dep. of Energy, "https://www.osti.gov/biblio/1504742-comparison-phasor-communications-protocols".