PUF-based Authentication Protocol with Physical Layer-based Obfuscated Challenge-Response Pair

Mona Alkanhal
University of Maryland, Baltimore County
Baltimore, MD
monaa1@umbc.edu

Abdulaziz Alali
Howard University
Washington,DC
Abdulaziz.Alali@bison.howard.edu

Mohamed Younis,
University of Maryland, Baltimore County
Baltimore, MD
younis@umbc.edu

***Abstract*—Node authentication is essential in IoT systems where interconnected resource-constrained devices operate autonomously. Developing a lightweight authentication technique is requisite to prevent malicious nodes from joining the network and impersonating legitimate nodes. This paper develops a novel protocol to enable mutual authentication for node pairs. The protocol combines the advantage of hardware-based security primitives, namely physically unclonable functions (PUFs), and agility and configurability of physical-layer communication mechanisms, specifically the Multi-Input Multi-Output (MIMO) method. Distinct from existing PUF-based techniques, our protocol allows two devices to mutually authenticate each other without the involvement of an intermediary server. Moreover, the proposed protocol encodes the transmitted information to protect it from any possible eavesdropping attempts that aim to model the PUF of a legitimate node. The experimental results demonstrate the efficacy of the proposed protocol against modeling attacks and impersonation attempts.**

***Index Terms*—Internet of things (IoT) , Physical Unclonable Functions (PUFs), authentication, Physical layer security.**

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10278899

I. Introduction

The Internet of Things (IoT) expands the scope of computer networks to include devices that are used in everyday life such as personal electronic gadgets, sensors, vehicles, digital assistants, etc. These devices are used in different fields such as healthcare and transportation [1]. As a general characterization, an IoT system comprises low-cost and resource-constrained devices that operate autonomously in unattended settings. However, the major role of IoT in critical applications and its proliferation can be a fertile environment for cyberattacks. An adversary may launch Denial of Service (DoS), masquerading, eavesdropping, and man-in-the-middle attacks. Therefore, a lightweight robust solution is requisite for protecting the authenticity, confidentiality, and integrity of IoT systems [2]. Particularly, device authentication is crucial to deny unauthorized access to the IoT framework as well as prevent impersonating legitimate nodes. Given the decentralized nature of IoT, the authentication process should be distributed with no need for the involvement of a trusted third party. Also, reducing the number of exchanged messages is necessary to limit the communication overhead.

With the resource constraints and heterogeneity of IoT devices, it is often impractical to apply conventional asymmetric cryptographic schemes to achieve the aforementioned security goals. Although relying on storage of device secrets in non-volatile memories is an attractive option, such a technique is vulnerable to security attacks where the device could be hacked resulting in reading the memory and revealing the key. An effective hardware-based solution that has been proposed in the literature is to generate secret keys [3]. A prominent example of hardware primitives that support such a solution is Physical Unclonable Functions (PUFs). A PUF benefits from the random and uncontrollable variations experienced during the manufacturing of integrated circuits in constructing a device signature that uniquely maps input bits, referred to as challenge, into an output bit(s) that reflects the PUF response. PUF-based security solutions have been investigated by exploiting the unique signature generated by hardware primitives without storing secrets in device memories [4]. The main advantage of such a solution is that generating the secret key is on demand with no need for a key storage. Although PUF-based authentication solutions have been proposed in the literature many of them require a trusted third party (i.e. secure server) to store a subset of challenge-response pairs (CRPs) for each of the legitimate nodes in the network. Such a solution is not suitable for IoT environments which favor autonomous management strategies [2].

A fundamental issue with distributed authentication using PUFs is that the challenge-response exchange is among IoT nodes rather than the secure server and hence becomes subject to increased vulnerability to attacks. Particularly, eavesdroppers could intercept the inter-node interactions to collect sufficient CRPs for modeling the underlying PUF using machine learning (ML) techniques [5]. Obfuscating the challenge and response through encryption is not practical since there are no secret keys stored on the individual nodes. This paper aims at filling the technical gap by exploiting the physical-layer properties of communication links as a means for obfuscating the transmitted challenge and response bits among nodes. In particular, we leverage the growing popularity of MIMO in wireless communication. Every IoT device will have an embedded PUF as well as a MIMO antenna array. We propose a novel PUF and physical-layer based lightweight protocol to enable mutual authentication for node pairs (PUPH). PUPH gets a prover δ_x to share a limited number of CRPs $\gamma_{x\to y}$ with a verifier δ_y. PUPH utilizes an innovative method to prevent a modeling attack that might intercept the communication between δ_y and δ_x, consequently capture a set of CRPs to precisely model δ_x's PUF. PUPH encodes the challenge bit using a MIMO antenna array in a manner that is controlled by the verifier and that varies over time. The validation results based on a PUF implementation show that PUPH robustly defeats cyberattacks.

The remainder of the paper is organized as follows. Section

II discusses the recent work on IoT authentication and PUF-based solutions. In Section III, we provide some background on PUFs. Section IV represents the system model and describes the proposed protocol in detail. Section VI reports the performance results. We conclude the paper in Section VII.

II. Related Work

Several authentication techniques and security provisions have been utilized to safeguard wireless networks [6]. However, these techniques are not suitable for an IoT network that is composed of low-cost and resource-constrained devices, and operates in an unattended setup. Storing the device identity in its memory, which is adopted in several authentication techniques, might not be sufficiently secure. PUF-based schemes avoid such a shortcoming by exploiting the unclonability and uniqueness of the device signature generated by PUF [5]. Nevertheless, PUF-based authentication techniques are still vulnerable to message replay attacks. To overcome such vulnerability, Chatterjee et al. [7] have proposed a PUF-based authentication protocol that employs identity-based encryption and a keyed hash function. However, their solution imposes heavy computational load and is not suited for resource constrained IoT devices. J. R. Wallrabenstein [8] apply elliptic curve cryptography to reduce the computational complexity of the authentication process; yet their approach requires some alterations in the IoT device hardware. The protocol of Yoon et al. [9] is geared to enable mutual authentication among IoT devices. However, in addition to the complexity introduced for encrypting the exchanged CRPs between devices, an intermediary server is still needed for storing CRPs as well as generating secure keys.

While a PUF is physically unclonable, it is vulnerable to modeling attacks if the adversary successfully captures enough CRPs. For example, the adversary could eavesdrop on a prover node to intercept the exchanged authentication messages with other nodes, i.e. verifiers. Based on the intercepted messages, the eavesdropper can build a machine learning model that mimics the prover's PUF and predicts the responses for unused challenges. To mitigate such vulnerability, Majzoobi et al. [10] send only a subset of the response to the verifier instead of the entire response. A synchronized random number generator between the prover and verifier is used to determine the subset of the response. In [11], the challenge bit string is shuffled and split over multiple messages. Additionally, challenge obfuscation has been investigated, where a hash function or encryption is applied on the challenge bit strings, and the encrypted version is then fed to the PUF [12]. However, the aforementioned protection techniques require a secure server for authenticating the nodes.

P-MAP [2] supports mutual authentication while countering modeling attacks by using two challenges and a bitwise binary operation that is known only to the two communicating nodes. Although deemed effective, P-MAP still allows an adversary to access the challenge bits and relies on the secrecy of the binary operation. PUPH adopts the features of MIMO to deprive the adversary from knowing the challenge and response bits and hence makes the ML-based modeling attempt useless. J.Tang et al. [13] benefit from MIMO to secure the transmission between two nodes where a "key bit" is used to encrypt the confidential information. Such a key is encoded in the indexes of the activated/deactivated antenna combination of the receiver. The approach is further extended in [14] to support sharing a broadcast key with a group of devices. However, their approach is prone to impersonation attacks. PUPH avoids such a weakness by incorporating PUFs.

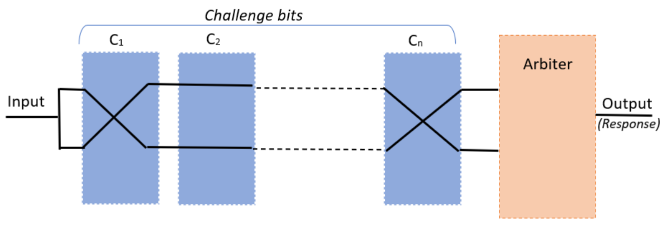


Fig. 1. The structure of n-bits challenge arbiter-PUF. Depending on the setting of an active switch (multiplexer) in each cell, every signal propagates through different paths within the cell. Using the challenge bits which configures the cells, a distinct path and propagation delay will be defined. Therefore, the response of the arbiter-PUF is generated based on the faster path of the two signals when the challenge bits are fed in.

III. System Model and Approach Overview

A. Physical Unclonable Functions

The underlying design principle of a PUF is that there are always slight variations among microelectronic circuits caused by imperfection during fabrication. Such imperfection is tolerable and fundamentally does not impact the functionality of integrated circuits. PUFs are designed to exploit these variations to devise a unique hardware-based fingerprint where a PUF produces a unique mapping from an input bit-string (challenge) into an output bit (response). For example, the variation of propagation delays is leveraged in the design of the Arbiter-PUF, shown in Fig 1, where the delay will not be equal for every integrated circuit and consequently the latched value for the same challenge bits is not fixed and becomes dependent on the device rather than the circuit design. Thus, a response R is uniquely generated for each challenge C, the relationship between C and R is represented as $PUF(C) = R$. Therefore, a PUF is deemed to be unclonable as the delay of every integrated circuit is uncontrollable.

PUFs are characterized based on the length of the challenge bit string into strong and weak. In essence such a length determines the number of combinations, i.e. how many CRPs a PUF has. Strong PUFs are more suited for authentication purposes given the large set of CRPs. Meanwhile weak PUFs are often used for generating cryptographic keys [16]. The PUF response can be influenced by environment conditions which might produce noise and cause bit flipping. Hence, we assume an error correction technique, e.g., fuzzy extractor or ECC, is utilized to mitigate these noises [15].

B. System and Threat Models

PUPH assumes that each device is equipped with a MIMO antenna array that helps improve the link quality and the utilization of the available spectrum. Each antenna array element is separated by greater than a half wavelength of radio frequency. Each verifier δ_y has N_{δ_y} number of antennas where the prover δ_x has N_{δ_x} antennas ($N_{\delta_x} > 1$). A standard time division multiplexing is assumed for medium access. PUPH also assumes that all of the IoT devices have embedded PUFs. A strong PUF is incorporated so that a large number of CRPs can be used to make it impractical for an attacker to brute force all possible CRPs. Without loss of generality, the presentation is based on the assumption that an Arbiter PUF used. The Arbiter-PUF, which is discussed earlier, is a prominent strong PUF design.

A PUF could be vulnerable to modeling attacks if the adversary captures CRPs and uses them to develop a ML model that mimics the PUF behavior. The adversary opts to have sufficient number of CRPs so that the formed ML model yields high accuracy in predicting the response of the PUF to any challenge bit pattern. In the context of PUPH, the adversary is an eavesdropper that intercepts the exchanged CRPs between a node δ_x, i.e., a prover, and other IoT devices, i.e., multiple verifiers. If the eavesdropping attempts succeed in capturing a sufficiently large subset of CRPs, the PUF model accuracy grows and consequently the authentication process becomes compromised. Therefore, safeguarding the CRPs is very critical for thwarting impersonation attacks.

C. Approach Overview

PUPH is a lightweight protocol that enables mutual authentication among IoT devices. The involvement of PUFs enables the concealment of device secrets and mitigates the threat of device hacking. PUPH pursues a distribution authentication strategy where a trusted server is not needed during network operation. Only device enrollment could leverage the participation of a server, although it is not a must. In the enrollment phase, each node gets a number of CRPs of the PUF for other nodes in the network. To elaborate, a node δ_x will get a set $\gamma_{y\to x} \quad \forall x \neq y$ of CRPs, where $|\gamma_{y\to x}|$, is not sufficient for developing an effective ML model of the PUF of δ_x.

To prevent CRPs accessibility to an eavesdropper the exchanged challenge and response are obfuscated. Unlike existing PUF-based solutions where a cryptosystem is incorporated [7], [8], the proposed protocol utilizes a MIMO-based mapping technique to transmit the challenge bit pattern. Depending on the number of antennas that the receiving node has, the challenge bits can be divided into segments the number of $Seg = N_{\delta_x} - 1$. The segmentation order of C is inferred based on the node ID. One segmentation order for δ_x could be $C = \{c_1, c_2, c_3\}$. Then the verifier δ_y will force one of δ_x's antenna segments to be non-active and activate the remaining ones. Thus, the encoded index will represent which antenna holds a part of C. The prover can use the same mechanism to transmit the PUF response to the verifier. The PUPH protocol is discussed in detail in the next section.

IV. PUPH Protocol Design

The proposed protocol operates in two main phases namely: Enrollment phase and Operation phase. The latter includes covering physical-layer channel estimation, challenge bits obfuscation using MIMO precoding at the sender, and decoding at the receiver.

A. Enrollment Phase

Device enrollment is mainly concerned with what required information that should be delivered to a node at the time of joining the system. Upon enrollment each node will get a subset of CRPs of other devices in the IoT network. To elaborate, let node δ_z shares Γ_z in the network, where Γ_z is a subset of all CRPs of the PUF of δ_z, denoted by PUF_z. Node δ_x will get $\gamma_{z\to x}$ where $\gamma_{z\to x} \subset \Gamma_z$ and the shared subsets are distinct for each node, i.e., $\gamma_{z\to x} \neq \gamma_{z\to y}$, $\forall x \neq y$. The size of Γ_z is based on the number of nodes in the system to allow sufficient diversity of the shared CRPs of PUF_z among the network nodes. Additionally, each IoT device needs to get a segmentation order Seg_z for other nodes in the network. PUPH leverages the embedded PUF to define such segmentation by setting $Seg_z = PUF_x(ID_z)$.

We note that the enrollment phase may be facilitated through a trusted server, which will be quite helpful if nodes join the network in a staggered manner; nonetheless, in such a case PUPH is still deemed as a distributed scheme since the enrollment phase constitutes an initialization stage and the operation phase does not involve any centralized entity. Upon enrollment, the node switches to the operation phase reflects the application-based interaction between the network nodes and is described in the following subsections.

B. Channel Estimation

PUPH employs a physical layer mechanism for obfuscating the challenge and response bits during the exchange of authentication messages. Recall that time division duplexing (TDD) is pursued, where the uplink and downlink transmissions are assigned distinct time slots with the same carrier frequency. To apply PUPH, first the prover δ_x transmits a pilot signal to the verifier δ_y in a time that is lower than the channel coherence time [15]. The verifier δ_y estimates the uplink channel matrix $\mathbf{H}_{xy}$ from the prover δ_x, where $\mathbf{H}_{xy} \in M^{N_{\delta_x} \times N_{\delta_y}}$ which reflects the engagement of all antennas of δ_x and δ_y,where M defines a matrix with size of $i \times j$. Due to the reciprocity of TDD, δ_y transposes $\mathbf{H}_{xy}$ in order to obtain the corresponding downlink channel as $\mathbf{H}_{yx} = \mathbf{H}_{xy}^T$.

Before the precoding weight, $\mathcal{V}$, is calculated, the verifier will need to normalize the estimated channel matrix to remove any possible impact of path loss. Therefor, the verifier δ_y computes $\mathcal{V} \in M^{N_{\delta_y} \times N_{\delta_x}}$ as the right inverse of $\mathbf{H}_{xy}$, where $\mathcal{V}$ is known as the zero-forcing precoding [17] such that

$$\mathcal{V} = \frac{\mathbf{H}_{xy}^H (\mathbf{H}_{xy} \mathbf{H}_{xy}^H)^{-1}}{\|\mathbf{H}_{xy}^H (\mathbf{H}_{xy} \mathbf{H}_{xy}^H)^{-1}\|} = (v_1, v_2, ..., v_{N_{\delta_x}}) \tag{1}$$

where each column vector $v_j \in M^{N_{\delta_y} \times 1}, j = 1, 2, .., N_{\delta_x}$ is normalized as $\|v_j\|^2 = 1$, therefore

$$\mathbf{H}_{yx}\mathcal{V} = \operatorname{diag}\left(\frac{1}{\|v_1\|}, \frac{1}{\|v_2\|}, ..., \frac{1}{\|v_{N_{\delta_x}}\|}\right) \tag{2}$$

C. Challenge Bits Mapping

Based on the number of δ_x's antennas (N_{δ_x}), the challenge bit string C is mapped using antenna indexes $\mathbf{e} = (e_1, e_2, e_3, ..., e_{N_{\delta_x}})$, where each $e_j \in (0, 1)$ $\forall j \in (1, 2, .., N_{\delta_x})$. To illustrate, the verifier δ_y will manipulate the precoding to consciously activate N number of δ_x's antenna segments to transmit C. Hence, the activated antenna index of δ_x that carries C is represented as "1" whereas the index of an inactive antenna is assigned a "0" to indicate that it does not participate in transmitting C. Therefore, from (1) and (2) the δ_x's received signal can be represented as

$$\mathbf{Y}_{\delta_x} = \mathbf{H}_{yx} \sum_{j=1}^{(N_{\delta_x}-1)} v_j C + n \tag{3}$$

where $n \in M^{N_{\delta_x} \times 1}$ is the additive Gaussian noise of the signal received by δ_x, and $\mathbf{Y}_{\delta_\mathbf{x}} = (y_1, y_2, y_3, ..., y_{N_{\delta_x}}) \in M^{N_{\delta_x} \times 1}$ indicates the received signals vector. Based on the activated/deactivated antenna segments, the received signal of δ_x s.t. $y \in \mathbf{Y}_{\delta_x}$, can be written as

$$y_j = \frac{\sqrt{P_w}}{\|v_j\|} C + n, \qquad \text{(an active antenna)}$$

$$y_j = n, \qquad \text{(non-active antenna)} \tag{4}$$

Where $\mathbf{e}_j = 0, \forall j \in (1, 2, .., N_{\delta_x})$ and P_w denotes the transmit power.From (2) and (4), the received signal of δ_x will be:

$$\mathbf{Y}_{\delta_x} = \sqrt{P_s}\left(\frac{1}{\|v_1\|}, ..., 0, ..., \frac{1}{\|v_{N_{\delta_x}}\|}\right)^T C + n \tag{5}$$

Where "0" indicates the order of the non-active antenna and P_s is the transmit power on each stream s.t. $P_s = \frac{P_w}{(N_{\delta_x}-1)}$. Based on the above analysis, assume $N_{\delta_x} = 4$ the verifier δ_y forces the first antenna to be non-active and use the remaining antennas to transmit the challenge bits $c_1, c_2,$ and c_3 where $(c_1, c_2, c_3) = C$. Therefore, the antenna indexes will be $\mathbf{e} = (0111)$ Let the segmentation order of δ_x be $Seg_x = (c_2, c_3, c_1)$. Since the first antenna is not activated which will not hold any part of C, the antenna index mapping will be $\mathbf{e}(1) = \{c_2\}, \mathbf{e}(2) = \{c_3\}, \mathbf{e}(3) = \{c_1\}$. Therefore, even if the antenna index $\mathbf{e}$ is uncovered, the eavesdropper still needs to identify the partitioning of C. As mentioned earlier, the partition order is inferred based on the node ID.

D. Decoding Transmitted Bits

In the last step, after the prover δ_x receives the downlink signal, δ_x will need to find the index of deactivated antenna segments in order to construct the challenge bit C correctly. To clarify, δ_x identifies whether the ith antenna index is activated or not by finding the least signal-plus-noise using the following function,

$$f_{ind}(\mathbf{Y}_{\delta_x}) = \text{index}\left[\text{arg min}(|y_1|^2, |y_2|^2, ..., |y_{N_{\delta_x}}|^2)\right] \tag{6}$$
$$= \mathbf{e}(\hat{i})$$

Hence, the observed antenna indices by δ_x can be written as

$$\hat{\mathbf{e}} = (1..1, 0, 1..1) \tag{7}$$

Based on $\hat{\mathbf{e}}$, δ_x can recognize the activated antenna segments that cover C. Thereafter, the prover δ_x combines all the observed bits from each antenna segment to construct the complete challenge bits C. To illustrate, δ_x uses his Seg_x to construct C, such that $Seg_x(c_2, c_3, c_1) = C$. Then, δ_x applies

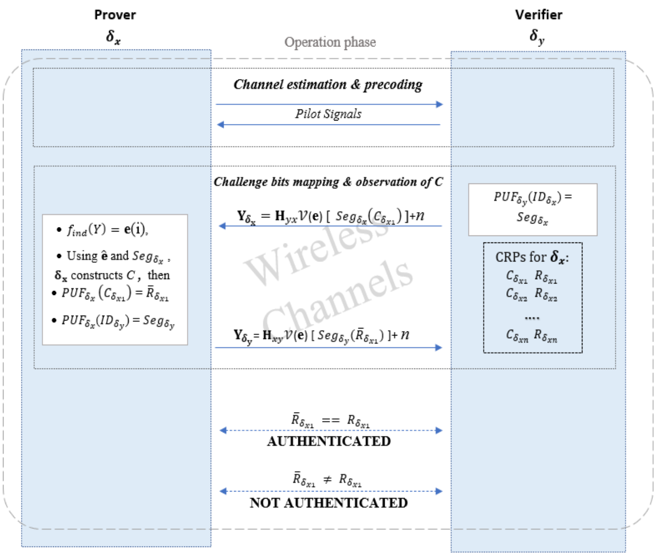

Prover
δ_x
Operation phase
Verifier
δ_y
Channel estimation & precoding
Pilot Signals
Challenge bits mapping & observation of C
$\mathbf{Y}_{\delta_x} = \mathbf{H}_{yx}\mathcal{V}(\mathbf{e})\,[\,Seg_{\delta_x}(C_{\delta_{x1}})\,]+n$
• $f_{ind}(Y) = \mathbf{e}(\mathbf{i})$,
• Using $\hat{\mathbf{e}}$ and Seg_{δ_x}, δ_x constructs C, then
• $PUF_{\delta_x}(C_{\delta_{x1}}) = \bar{R}_{\delta_{x1}}$
• $PUF_{\delta_x}(ID_{\delta_y}) = Seg_{\delta_y}$
$PUF_{\delta_y}(ID_{\delta_x}) = Seg_{\delta_x}$
CRPs for δ_x:
$C_{\delta_{x1}}$ $R_{\delta_{x1}}$
$C_{\delta_{x2}}$ $R_{\delta_{x2}}$
....
$C_{\delta_{xn}}$ $R_{\delta_{xn}}$
Wireless Channels
$\mathbf{Y}_{\delta_y} = \mathbf{H}_{xy}\mathcal{V}(\mathbf{e})\,[\,Seg_{\delta_y}(\bar{R}_{\delta_{x1}})\,]+n$
$\bar{R}_{\delta_{x1}} == R_{\delta_{x1}}$
AUTHENTICATED
$\bar{R}_{\delta_{x1}} \neq R_{\delta_{x1}}$
NOT AUTHENTICATED

Fig. 2. A sequence diagram to illustrate the message exchange between a prover δ_x and a verifier δ_y.

C to his PUF_x to generate R, $PUF_x(C) = R$. Lastly, the prover follows the above steps to transmit R. Then the verifier δ_y will check R with the value it got during the enrollment phase; if it matches then δ_x is authenticated. Fig. 2 shows the message sequence of the operation phase of PUPH.

V. Simulation Results and Analysis

PUPH has been validated through simulation. MATLAB is used where MIMO is implemented. The arbiter-PUF explained earlier in Fig 1 has been implemented in each simulated node. We have considered two configurations where the PUF maps a 16-bit or 64-bit challenge to a one-bit response. Unless otherwise stated, the parameters used in the simulation are as follows: all nodes, i.e., provers, verifiers, and the eavesdropper, have the same number of antennas, i.e. $N_{\delta_x} = N_{\delta_y} = N_{\delta_{Eve}} = 5$. The SNR varies from 0 dB to 30 dB. White Gaussian noise is assumed with zero mean. SVM is employed as a representative ML technique that the adversary δ_{Eve} might pursue to perform the cloning attack against the authentication protocol. The implementation considered the following three attack scenarios aiming to impersonate δ_x by modeling its PUF (PUF_x): (1) the attacker is a malicious node in the network, i.e., an insider attacker, that has a full knowledge of the authentication process; (2) Unlike the previous scenario, the adversary does not know the detailed PUPH protocol. Yet the attacker realizes that there is an index mapping function (f_{ind}) being used but does not know how f_{ind} is applied; (3) an outsider attacker who does not have any knowledge about the authentication protocol. More details are provided next.

A. Performance Results

We consider a scenario where a mutual authentication is established between a prover δ_x and a verifier δ_y. During the operation phase, an eavesdropper δ_{Eve} intercepts the connection between δ_x and δ_y aiming to collect a large number of CRPs in order to model δ_x's PUF (PUF_x) and therefore

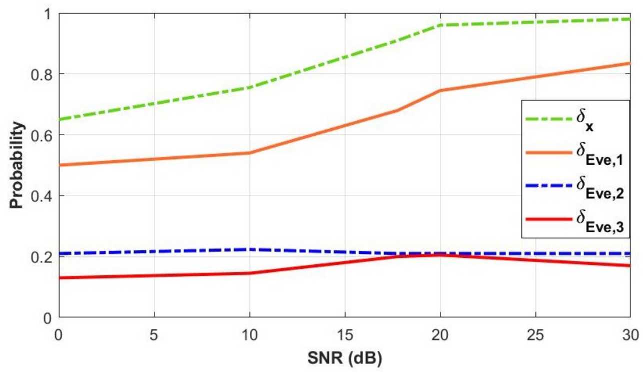


Fig. 3. The probability of observing **e**.

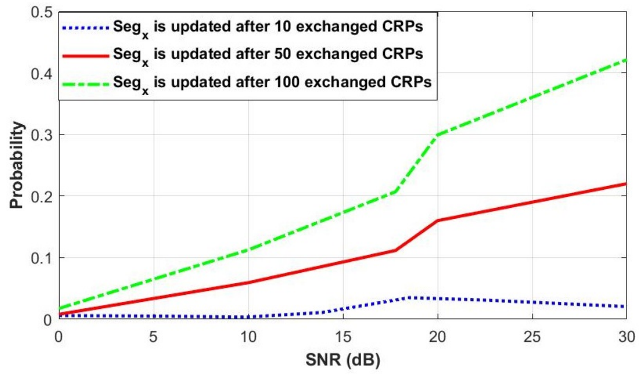


Fig. 4. The probability of observing C correctly by $\delta_{Eve,1}$.

impersonating δ_x. Due to the MIMO mapping technique of PUPH, the eavesdropper δ_{Eve} will fail to capture the challenge bits C directly without knowing f_{ind} to get **e**.

Full knowledge ($\delta_{Eve,1}$): In the worst-case scenario $\delta_{Eve,1}$ is aware of f_{ind}. Consequently, $\delta_{Eve,1}$ obtains **e** that determines which antenna should be active or inactive. The results are shown in Fig. 3 corresponding to $\delta_{Eve,1}$ where $\delta_{Eve,1}$ obtains probability 83% of finding **e** at SNR value of 30dB, compared with the prover δ_x which has 99%. Since the verifier usefully includes $Seg_x(C)$ in the request message, the eavesdropper will record $\hat{C}$ rather than C. This is clearly shown in Fig. 5, where the x-axis reflects the number of captured CRPs$_x$ by δ_{Eve}, and the y-axis shows the accuracy of the PUF model (i.e PUF_x) for an SNR of 10 dB. As demonstrated by the results in Fig. 5, full knowledge of the PUPH's operation and parameter settings does not enable effective modeling attacks, where for both the considered PUF sizes (Fig. 5(a) and fig. 5(b)) the accuracy of the ML model does not exceed 54%. Such a low accuracy is obviously a random guess and is consistent with the probabilities shown in Fig. 4, which reflect the scenario where $\delta_{Eve,1}$ accesses $Seg(C)$. Being aware of the application of PUPH, $\delta_{Eve,1}$ will seek to know the segmentation order (Seg_x). However, the segmentation order Seg is periodically updated. Therefore, if $\delta_{Eve,1}$ successfully uncovers Seg_x, little gain will be made by the attacker since Seg_x will soon change. This is reflected in Fig. 4 where the Seg_x is updated at various periods with varying SNR values. At the highest update rate, i.e., updating after 10 exchanged CRPs, the probability that $\delta_{Eve,1}$ observes C correctly is less than 0.1. Such a probability grows to only 0.4 when the update takes place after the exchange of 100 CRPs (i.e., at a slower rate). Thus, under that worst case attack scenario (i.e. full knowledge) PUPH stays resilient against modeling attack.

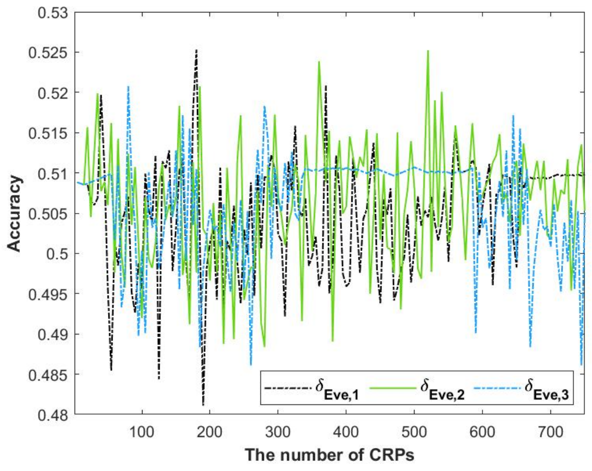
Partial knowledge ($\delta_{Eve,2}$): This scenario reflects awareness that there is f_{ind} without knowing how it operates. To elaborate, $\delta_{Eve,2}$ understands that there is an antenna index being ignored but does not know which one. Therefore, $\delta_{Eve,2}$ discards one antenna segment randomly and uses the other segments to extract the challenge bits. For example, if $\mathbf{e} = (0111)$ is used, one possible estimation by the adversary could be $\mathbf{e}_{Eve} = (1110)$. Therefore, the probability that $\delta_{Eve,2}$ correctly estimates **e** of δ_x is $\frac{1}{N_{\delta_x}}$, i.e., $Pr[\mathbf{e} = \mathbf{e}_{Eve}] = \frac{1}{N_{\delta_x}}$, which is consistent with the results in Fig. 3 where the curve for $\delta_{Eve,2}$, is flat with probability of 0.2. Accordingly, modeling the observed CRPs by based on such an attack scenario is meaningless as also confrimed by Fig. 5 (a) and (b), where $\delta_{Eve,2}$ does not exceed 53% in performing cloning attack in 16-bit and 64-bit PUFs,respectively.

No knowledge ($\delta_{Eve,3}$): In this scenario, $\delta_{Eve,3}$ does not know f_{ind} and **e**. Therefore, $\delta_{Eve,3}$ will consider all antennas to be active. Each antenna receives $c_i \in C$, where $|c_i| = 4$ bits and $|C| = 16$ bits. Since $N_{\delta_{Eve},3} = 5$ the assumed total will be 20 bits. Thus, $\delta_{Eve,3}$ will randomly discard 4 bits. As shown in Fig. 3, the probability of $\delta_{Eve,3}$ in observing **e** is almost 0.2. Consequently, it is reasonable to find that the modeling accuracy of $\delta_{Eve,3}$ is ranging between 49% and 53% in 16-bit and 64-bit PUF as shown in Fig. 5, which reflects a purely random guess.

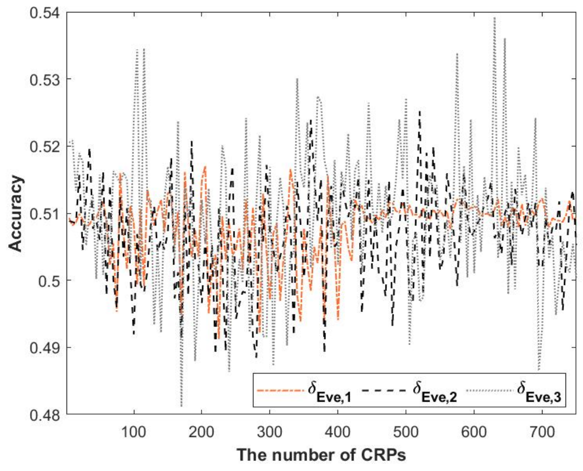
B. Security Analysis

We analyze PUPH's ability in thwarting any attempts by an eavesdropper to impersonate a legitimate node by modeling its PUF. Essentially, PUF modeling techniques attempt to simulate a mapping of an n-bit challenge to an m-bit response. An adversary will eavesdrop transmissions to record samples of PUF CRPs. Accordingly, the adversary will use the collected CRPs to build a ML model for predicting the response for unused challenges. In PUPH, the adversary has to observe the inactive antenna index i.e., guess f_{ind}. Thus, in the worst case scenario the adversary $\delta_{Eve,1}$ would know the value of N_{δ_x}, and consequently $\delta_{Eve,1}$ sets $N_{\delta_{Eve,1}} = N_{\delta_x}$. In order to model δ_x's PUF, $\delta_{Eve,1}$ should capture a sufficient number of correct (valid) CRPs. Assume that $\delta_{Eve,1}$ discovers **e** successfully. Hence, $\delta_{Eve,1}$ will access $Seg_x(C)$ rather than C. Recall, Seg in PUPH is inferred based on the node ID and is periodically updated; hence, knowing Seg_x has no benefit since $\delta_{Eve,1}$ should also observe the corresponding response where different **e** and Seg is applied.

The analysis is verified by the results of Fig. 4, which shows the probability that $\delta_{Eve,1}$ correctly infer C under three



(a) 16-bit PUF



(b) 64-bit PUF

Fig. 5. Accuracy of modeling 16-bit and 64-bit Arbiter-PUF using SVM.

different periods of updating CRPs (i.e., updated every 50 exchanged CRPs) with varying SNR values. As indicated by the results in the figure, the highest success probability for $\delta_{Eve,1}$ in observing C only is not exceeding 42%, where the CRPs is set to be updated after 100 exchanged CRPs and SNR is 30dB. Hence,$\delta_{Eve,1}$ could observe R with the highest probability of 42%. Let the observed CRPs by $\delta_{Eve,1}$ be $C\hat{R}P$ and the observed C and R be $\hat{C}$ and $\hat{R}$,respectively. The probability that $\delta_{Eve,1}$ correctly infer both C and R both is Pr($C\hat{R}P$ = CRP) = Pr($\hat{C} = C$) . Pr($\hat{R} = R \mid \hat{C} = C$); this is performed during a single period of exchanged CRPs. Therefore, PUPH is effectively defeating such modeling attempts.

VI. Conclusion

This paper has presented, PUPH, a novel lightweight authentication protocol that leverages a hardware security primitive, namely PUF, and the MIMO-based physical layer design. PUPH allows IoT nodes to mutually authenticate each other in a distributed manner. Additionally, PUPH obfuscates the transmitted CRPs to deny eavesdropping from accessing these CRPs and impersonating a legitimate node. We have demonstrated the resilience of PUPH against attacks through simulation using data collected from an FPGA-based implementation of PUF. In the future we would like to test PUPH using a prototype IoT network.

Acknowledgement

A. Alali is supported by NSF under grant agreement DMS-2022448, CCF-0939370, and by Nvidia Academic Hardware Grant Program.

References

[1] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, “A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis,” *Sensors*, vol. 20, no. 13, 2020, 3625.

[2] M. Alkanhal and M. Younis, ”P-MAP: PUF-based Mutual Authentication Protocol,” *ICC 2022*, pp. 3424-3429.

[3] B. Kim, S. Yoon and Y. Kang, ”PUF-based IoT Device Authentication Scheme on IoT Open Platform,” *ICTC* , 2021, pp. 1873-1875.

[4] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, “A Survey on Physical Unclonable Function (PUF)-based Security Solutions for Internet of Things,” *Computer Networks*, vol. 183, 2020, 107593.

[5] P. Mall, R. Amin, A. K. Das, M. T. Leung and K. -K. R. Choo, ”PUF-Based Authentication and Key Agreement Protocols for IoT, WSNs, and Smart Grids: A Comprehensive Survey,” *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8205-8228, June 2022.

[6] R. Román-Castro, J. López and S. Gritzalis, “Evolution and Trends in IoT Security,” *IEEE Computer*, vol. 51, no. 7, pp. 16-25, July 2018.

[7] U. Chatterjee, et al. “Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database,” *IEEE trans. on dependa. & sec. comp.*, 16(3), pp. 424-437, 2018.

[8] J. R. Wallrabenstein, “Practical and secure IoT device authentication using physical unclonable functions” *Proc. IEEE* 4^{th} *international conference on future internet of things and cloud (FiCloud)*,2016.

[9] S. Yoon, B. Kim, Y. Kang and D. Choi, ”PUF-based Authentication Scheme for IoT Devices,” *Proc. International Conf. on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1792-1794.

[10] Majzoobi, Mehrdad, et al. “Slender PUF protocol: A lightweight,robust, and secure authentication by substring matching,” *Proc. IEEE Symposium on Security and Privacy Workshops*, 2012.

[11] M. Ebrahimabadi, M. Younis and N. Karimi, “A PUF-Based Modeling-Attack Resilient Authentication Protocol for IoT Devices,” *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3684-3703, March 2022.

[12] F. Farha, et al., “SRAM-PUF based entities authentication scheme for resource-constrained iot devices,” *IEEE Internet of Things* J., Vol. 8,No. 7, pp. 5904 - 5913, April 2021.

[13] J. Tang, L. Jiao, K. Zeng, H. Wen and K. -Y. Qin, “Physical Layer Secure MIMO Communications Against Eavesdroppers With Arbitrary Number of Antennas,” in *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 466-481, 2021

[14] J. Tang, H. Wen, H. -H. Song, L. Jiao and K. Zeng, “Sharing Secrets via Wireless Broadcasting: A New Efficient Physical Layer Group Secret Key Generation for Multiple IoT Devices,” in *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 15228-15239, 15 Aug.15, 2022.

[15] Y. Lao et al., “Reliable PUF-Based Local Authentication with Self Correction,” *IEEE Trans. on CAD*, vol. 36, no. 2, pp. 201–213, 2016.

[16] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J Young, “A PUF Taxonomy,” *Applied Physics Reviews*, Vol. 6, No. 1, 2019.

[17] K.-K. Wong and Z. Pan, “Array gain and diversity order of multiuser MISO antenna systems,” *International J. Wire. Inf. Networks*, vol. 15, no. 2, pp. 82–89, 2008.

[18] Y. Shen and E. Martinez, “Channel estimation in OFDM systems,”*Freescale semiconductor application note*, pp. 1–15, 2006.

[19] K.-K. Wong and Z. Pan, “Array gain and diversity order of multiuser MISO antenna systems,” *International J. Wire. Inf. Networks,* vol. 15, no. 2, pp. 82–89, 2008.