

Article

# FeRHA: Fuzzy-Extractor-Based RF and Hardware Fingerprinting Two-Factor Authentication

Mona Alkanhal <sup>1,\*</sup> , Mohamed Younis <sup>1,\*</sup> , Abdulaziz Alali <sup>2</sup>  and Suhee Sanjana Mehjabin <sup>1</sup> 

<sup>1</sup> Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore, MD 21250, USA; suheesm1@umbc.edu

<sup>2</sup> Department of Electrical and Biomedical Engineering, University of Nevada, Reno, NV 89557, USA; abdulaziz.alali@howard.edu

\* Correspondence: monaa1@umbc.edu (M.A.); younis@umbc.edu (M.Y.)

**Abstract:** The Internet of Things (IoT) reflects the internetworking of numerous devices with limited computational capabilities. Given the ad-hoc network formation and the dynamic nature of node membership, secure device authentication mechanisms are critical. This paper proposes a novel two-factor authentication protocol for IoT devices. The protocol integrates physical unclonable functions (PUFs) and radio frequency fingerprints (RFFs), providing a unique identification method for each device. Compared with existing PUF-based schemes, the proposed protocol facilitates the mutual authentication of two devices without the need for a trusted third party. Our design is resilient to the intrinsic noise associated with PUFs and RFFs, ensuring reliable authentication, even under various operational conditions. Furthermore, we have implemented an obfuscation technique to secure shared authentication data against eavesdropping attempts aimed at modeling the security primitive, i.e., the PUF, through machine learning algorithms. We have validated the performance of our protocol and demonstrated its efficacy against various security threats, including impersonation, message replay, and PUF modeling attacks. Notably, the validation results indicate that predicting any given PUF response bit's accuracy does not exceed 56%, making it as unpredictable as a random guess.

**Keywords:** Internet of Things; physical unclonable function; RF fingerprint; authentication protocol



**Citation:** Alkanhal, M.; Younis, M.; Alali, A.; Mehjabin, S.S. FeRHA: Fuzzy-Extractor-Based RF and Hardware Fingerprinting Two-Factor Authentication. *Appl. Sci.* **2024**, *14*, 3363. <https://doi.org/10.3390/app14083363>

Academic Editors: David Sarabia-Jácome and Carlos Enrique Palau Salvador

Received: 11 March 2024

Revised: 9 April 2024

Accepted: 11 April 2024

Published: 16 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Internet of Things (IoT) conceptually refers to the internetworking of a large number of heterogeneous nodes in an ad-hoc manner [1]. An IoT is defined as a network of interconnected computers, sensors and electronic devices, and actuators that have unique identifiers and the ability to exchange data over a network without manual or human-in-loop management. IoT constitutes a global revolution that profoundly impacts the lives of millions of individuals. Its overarching ambition is to establish machine-to-machine (M2M) communication, thereby enabling the autonomous operation of devices. The fundamental objective of IoT has always been to connect objects anywhere, anytime, and for any purpose, using any intelligent network [2].

In light of the IoT characteristics, decentralized management is a preferred approach in order to ensure effective management of the system. The implementation of decentralized management can offer a strategic advantage by ensuring the continuity of operations, even when access to a central server is disrupted. This approach can effectively eliminate the potential risks and challenges that may arise due to central server unavailability. Hence, it is advisable to adopt decentralized management as a means of ensuring operational stability and continuity.

The widespread adoption of IoT networks has brought to the forefront significant security concerns. The openness and pervasiveness of these networks make them inherently vulnerable to infiltration, emphasizing the need for robust device authentication systems. Additionally, the connectivity of IoT devices renders them susceptible to unauthorized

access by adversaries from remote locations. Such an attack may use actuators to take harmful actions against the system or even endanger the user's safety. They can attack a device to obtain sensitive data such as location, health, and financial information [3–5]. Thus, device authentication is crucial to prevent infiltration of the IoT network. In essence, impersonating a legitimate device can be an effective way for an attacker to violate privacy and inject false data. Many lightweight protocols have been developed to establish mutual trust among nodes [6]. However, traditional methods of authentication that rely on key-based communication are susceptible to hacking, and can result in information leakage through side channel analysis and hardware Trojan insertion. These methods also result in significant power and area overheads [6].

To address the aforementioned challenges, quite a few security solutions have been proposed in the literature [7,8]. Given the limited computation and communication resources of IoT devices, employing lightweight hardware-based security primitives, such as PUFs, has become an attractive design option [9]. The PUF concept is based on the inherent variations in the physical characteristics of a device, which can be exploited to generate a unique and device-dependent identity that is practically hard to replicate [10]. This unique identity can be used for authentication purposes to verify the trustworthiness of an entity based on its own physical characteristics rather than the content it exchanges. From a cryptographic perspective, PUFs are also deemed an attractive option for key generation, as they do not require a secure initial key to be exchanged between the communicating parties. Instead, cryptographic keys are generated using the unique PUF identity of the communicating devices, thereby ensuring a high level of security [11]. While quite a few PUF-based protocols can be found in the literature, mutual authentication of a pair of nodes requires the involvement of a trusted third party, e.g., a secure server [12]. Such a server needs to retain a number of challenge–response pairs (CRPs) for each of the network nodes. However, such an approach does not suit IoT frameworks where distributed network management is favored, if not required [13].

Analogous to the hardware-based PUFs, communication medium impairments along with the physical characteristics' variation of transceivers have been exploited as a means for identifying transmitters [14,15]. Such a concept is often referred to as RF fingerprinting (RFF), where the transient features of a transmitter in both the time and frequency domains are used to identify the packet sender [16]. These transient features are random and almost constant, and hence can be captured and used to effectively differentiate among the various network nodes with high accuracy [17–19]. However, environmental conditions introduce noise that can significantly impact the RF fingerprint accuracy, which can lead to misclassification, especially when the computational complexity of the classifier is a concern, as is the case in IoT [20]. Such a limitation can degrade the authentication process and thus compromise the security of the system [21].

Given the inherent characteristics of IoT devices and their management structure, the authentication process should be decentralized and avoid the involvement of trusted third parties. To ensure efficiency and expediency, it should also not require complex computations or the exchange of numerous messages. This paper opts to fulfill such a need by developing an authentication protocol that leverages two effective techniques, namely PUFs and RF fingerprinting, that suit IoT applications while considering the noise effect. By exploiting the inherent manufacturing variations in IoT devices, the combination of RF fingerprinting and PUF enables the unique identification of each node in the network. Such a unique combination obviates the need for traditional identification methods that rely on key storage for authentication. Since both the PUF and the RF fingerprint are based on unintended variations caused by manufacturing, we aim to increase robustness and mitigate the potential effect of noise by applying the fuzzy extractor. We call the proposed protocol Fuzzy-extractor-based RF and Hardware fingerprinting two-Factor Authentication (FeRHA). FeRHA is validated using data collected from a 64-bit PUF implemented on an FPGA and is shown to achieve high robustness to cyberattacks.

The rest of the paper is organized as follows. The next section sets FeRHA apart from competing schemes in the literature. Section 3 provides some background information on PUFs, RFE, and fuzzy extractors. Section 4 states the made assumptions and provides an overview of the FeRHA. The detailed protocol design can be found in Section 5. Section 6 describes the validation setup and reports performance results. Finally, the paper is concluded in Section 7.

## 2. Related Work

Numerous authentication methodologies and security provisions have been employed for the protection of wireless networks [22]. Nevertheless, these techniques are not suitable for IoT networks, characterized by low-cost and resource-limited devices that operate in an unsupervised environment. The storage of a device's identity in the memory, as practiced in several authentication techniques, might not be deemed entirely secure. In this regard, PUF-based schemes have been developed to overcome this issue by exploiting the unclonability and uniqueness of PUFs to generate a device signature [23–25]. Some work combines a PUF and a cryptosystem to protect the exchanged CRP from eavesdroppers. For example, Chatterjee et al. [26] have developed an authentication protocol that utilizes PUFs in conjunction with identity-based encryption and a keyed hash function. Yilmaz et al. [27] have addressed the issue of counterfeit products by developing a tripartite authentication protocol and anti-counterfeit tag design based on Rabin public-key encryption and PUFs. The goal of this protocol is to ensure that no counterfeit products are produced and that only legitimate products are available to end users. PUF-IPA [28] is a cryptographic solution designed to safeguard the confidentiality of CRPs. This approach involves an advanced encryption standard (AES)-128-based encryption mechanism to avoid exposure of partial or complete CRPs. Yet, these approaches add complexity by encrypting the exchanged CRPs within IoT nodes. In addition, a third-party server is still required to manage the storage of CRPs and produce secret keys. In a similar vein, Patil et al. [29] leveraged blockchain smart contracts for authenticating IoT devices with miners in the blockchain network. Nonetheless, this protocol uses the computationally intensive Diffie–Hellman key exchange protocol, which could trigger performance difficulties. Fakroon et al. [30] pursued a multi-factor authentication protocol that combines PUFs with user passwords. Alladi and Chamola [31] used PUF for authentication and key generation for healthcare IoT devices. However, the CRPs are recorded in a database, which may pose a risk of vulnerability to PUF modeling attacks. Nimmy et al. [32] developed an authentication scheme by integrating geometric threshold secret sharing with PUFs. Such a solution is designed to prevent storing CRPs in the verifier's database. Yet, the verifier is still obligated to retain both the sub-challenge bit-string and the response's hash. In summary, the aforementioned solutions utilize hashing or encryption, which are resource demanding and could be a limiting factor, particularly in the context of small devices. FeRHA, on the other hand, offers a superior solution by leveraging a combination of PUFs and RF fingerprinting for two-factor authentication. Additionally, our protocol capitalizes on the utilized hardware primitive to obscure the exchanged CRPs without applying any cryptography technique. Table 1 presents a comparison of the various authentication techniques used in the context of IoT.

Despite the inherent unclonability, a PUF is still vulnerable to modeling attacks. An attacker could eavesdrop on a prover node with the intent of capturing transmitted authentication packets to other nodes, i.e., verifiers. Using the intercepted CRPs, the attacker may formulate a machine learning model, which mimics the prover's PUF. This enables the attacker to predict the mapping of responses to unobserved challenges. To mitigate such vulnerability, a number of techniques have been pursued. For example, Majzoobi et al. [33] transmitted a restricted number of response bits to the verifier rather than the complete bits. The selected response bits are defined by executing a synchronized random number generator between the prover and the verifier. An alternative strategy is to conceal the challenge bit-string using encryption [34]. Nozaki et al. [35] introduced a

PUF-based authentication method using secret sharing schemes. This method utilizes the distributed values generated during the process of authentication instead of communicating PUF responses directly. Using this method, the vulnerability of raw PUF responses being exposed is minimized. However, the proposed protocol is still susceptible to impersonation attacks and the disclosure of CRPs. P-MAP [13] enables mutual authentication while guarding against PUF modeling attacks by using two challenges and a unique bit-wise binary function specific to the devices involved. However, the challenge bits are vulnerable to unauthorized access by adversaries. Moreover, the security of the protocol heavily depends on keeping the binary function secured. Additionally, Jiang et al. [36] have proposed a three-factor authentication protocol for vehicular networks, which combines passwords, biometrics, and PUFs to authenticate vehicles and generate encryption. This protocol ensures secure communication by eliminating the need to store IDs in vehicle sensors. Mahalat et al. [37] have conducted a study that integrates Shamir's secret sharing and Pedersen's verifiable secret sharing with the PUF. However, these approaches bring a significant computational overhead and necessitate the involvement of a trusted third party (server) for mutual authentication. We posit that such involvement could have an adverse impact on performance, as the server may become a bottleneck and require continuous availability.

Meanwhile, RF fingerprinting utilizes the unique characteristics of wireless signals to identify and authenticate IoT devices. Unlike traditional authentication methods, which rely on passwords or digital certificates, RF fingerprinting does not require any pre-shared secrets or complex cryptographic algorithms. Instead, it leverages the inherent variability of wireless signals to generate a unique fingerprint for each device, which can be used to verify its identity. Various studies have investigated the application of RF fingerprinting in IoT authentication. Nouichi et al. [38] delved into the use of RF fingerprinting to authenticate IoT devices. They emphasized the efficacy of this method in achieving high identification accuracy without the need for additional hardware or complex cryptographic mechanisms. The authors demonstrated how RF fingerprinting can successfully differentiate between legitimate IoT devices and unauthorized entities. Recently, Zhang et al. [39] have presented a thorough analysis of RF fingerprinting techniques applied to IoT authentication. The study placed special emphasis on the integration of RF characteristics, such as signal strength, phase, and frequency response, as a means to identify devices. However, addressing the noise mitigation in the authentication process using RFF was not considered in their approach. Furthermore, Li et al. [40] investigated the potential of utilizing machine learning algorithms in combination with RF fingerprinting to enhance the authentication of IoT devices. By employing supervised learning techniques, the authors were able to demonstrate notable improvements in the accuracy and reliability of IoT device authentication. However, relying on RF fingerprinting as the primary authentication factor raises security concerns. Peng et al. [41] have investigated the susceptibility of RF fingerprinting systems to adversarial attacks, highlighting the potential for misclassification and unauthorized access. In particular, they demonstrate that adversarial perturbations can be deliberately crafted using machine learning algorithms to deceive RF fingerprinting classifiers. FeRHA mitigates such vulnerability by incorporating RF fingerprinting as a secondary factor. Our approach capitalizes on the unique characteristics of RF fingerprints to obscure the transmitted CRPs, while avoiding reliance solely on RF fingerprinting to authenticate network devices.

Few studies have considered PUFs and RF fingerprints as factors for authenticating IoT nodes. In instance, Ashtari et al. [21] introduced a framework that leverages the combination of RF-PUF and random forest classification for authenticating IoT nodes. Furthermore, Chatterjee et al. [42] presented a deep neural network-based framework for real-time authentication of wireless nodes using PUFs and RF fingerprinting. The framework detects process variation-induced effects on RF properties of wireless transmitters using in situ machine learning at the receiver end. Also, Bari et al. [43] introduced a secure authentication method that utilizes the RF-PUF for both static and quasi-static channels.

The proposed method aims to ensure reliable and efficient authentication by exploiting the inherent randomness of the RF-PUF. The aforementioned approaches have shown to be effective in accurately identifying IoT nodes. However, these approaches are susceptible to modeling attacks and do not account for the impact of noise on biometric data, which can cause authentication failure. Addressing the issue of noise in biometric data is critical to enhancing the security of authentication systems.

**Table 1.** Comparison of the existing PUF- or RFF-based mutual authentication techniques for IoT.

Ref.	Key Feature	Advantage	Disadvantage
[25]	PUF-based authentication	Enables mutual authentication between IoT devices	Relies on an intermediary server to store CRPs and generate secure tokens
[26]	PUF-based authentication	No stored explicit CRPs in the verifier database	Introduces additional complexity due to encrypted CRPs
[27]	PUF-based authentication	Develops a security solution to address counterfeit products	Utilizes Rabin public-key encryption
[28]	PUF-based authentication	No partial/full CRPs are stored	Incorporates AES-128-based encryption and require an intermediary server
[29]	Authentication using PUFs and blockchain	Utilizes blockchain smart contracts	Computationally heavy due to the use of Diffie–Hellman protocol
[30]	Multifactor authentication using PUFs	Suitable for telehealth system for mobile and IoT edge devices	Relies on user passwords during the authentication
[31]	PUF-based authentication and key generation	Use simple cryptographic primitives	Vulnerable to machine learning attacks.
[32]	PUF-based authentication	No explicit CRP storage in the verifier database	Overhead of cryptographic operations (geometric threshold secret sharing)
[33]	PUF-based authentication	Applies challenge obfuscation and uses a subset of response bits	Susceptible to impersonation attacks due to CRP disclosure
[34]	PUF-based authentication	Obfuscates challenge bit-strings to counter modeling attacks	Use of hash functions introduces a significant computational overhead
[35]	PUF-based authentication	Relies on secret sharing to minimize exposure of raw PUF responses	Susceptible to impersonation attacks due to increased CRP disclosure
[13]	PUF-based mutual authentication	Mitigates modeling attacks using unique binary operations	Attacker can access challenge bits; dependant on secrecy of binary operations
[36]	Three-factor authentication	Combines passwords, biometrics, and PUFs without storing node IDs	Introduces a significant computational overhead and requires the involvement of a server for mutual authentication
[37]	PUF-based authentication	Integrates secret sharing with PUF for authentication	Introduces computational overhead and requires the involvement of a server for mutual authentication
[38–40]	IoT authentication using RF fingerprinting	Utilizes unique characteristics of wireless signals and avoids complex cryptography	Vulnerable to adversarial attacks
[21,42,43]	Two-factor authentication	Leverages RFFs and PUF for authenticating wireless nodes	Susceptible to modeling attacks; no mitigation of the noise impact on biometric data

### 3. Preliminaries

This section provides some background information that is needed for explaining the design of FeRHA.

#### 3.1. Physical Unclonable Functions

A PUF refers to a digital circuit that maps an input, referred to as a challenge, to an output (a response) in an implementation-dependent manner [44,45]. The fundamental basis behind the PUF design lies in the existence of minor discrepancies in microelectronic circuits that arise due to manufacturing imperfections [10]. These imperfections are deemed insignificant and do not impact the operation and characteristics of integrated circuits. PUF circuits are constructed to leverage these variations to produce a unique hardware-driven fingerprint, thus creating a distinct mapping from an input bit-string challenge to an output bit PUF response [10].

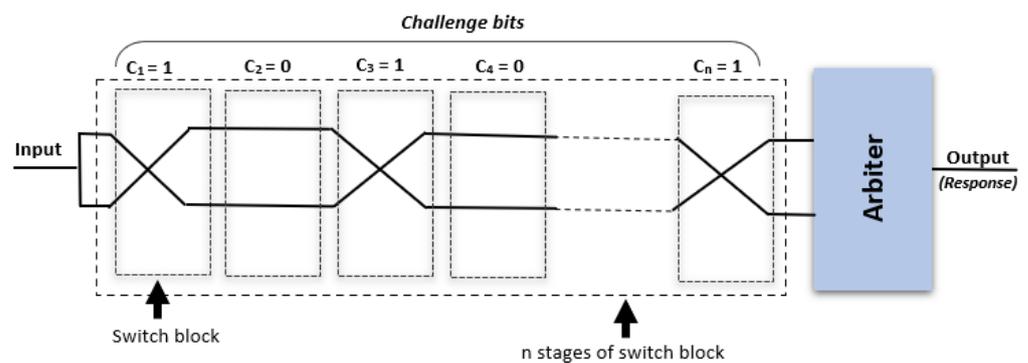
These features have made the use of PUFs a viable means for generating a unique signature that provides an additional layer of security and authenticity in various applications.

To illustrate, Figure 1 shows the design of an arbiter-PUF, a notable PUF design that leverages the variation in propagation delays. Due to the non-uniformity in delays among integrated circuits, the latched value for the same challenge bits will differ, even with the same circuit implementation. This variation reflects a random effect during device manufacturing, making physical cloning of a PUF infeasible, even with knowledge of all implementation parameters [46]. In fact, the CRPs of a PUF cannot be predicted or controlled by the manufacturer. An invasive attack on a device would result in changes in the signal delays, rendering PUFs tamper-resistant and one-way mapped. In summary, the responses received from different ICs for the same PUF are unique, serving as a fingerprint for individual ICs. A PUF can be defined mathematically as follows [47]:

$$f : C \rightarrow R. \tag{1}$$

$$f(C) : r(c \in C, r \in R). \tag{2}$$

where the output or response generated by a PUF is denoted by  $R$ , while the PUF challenge is represented by  $C$ .



**Figure 1.** An  $n$ -bit arbiter-PUF features a structure where signals propagate through different paths within each cell based on the setting of an active switch (multiplexer). The configuration of the cells is defined by the challenge bits, which in turn determine a unique path and propagation delay. Consequently, the response generated by the arbiter-PUF is based on the faster path of the two signals when the challenge bits are fed in.

#### 3.2. Radio Frequency Fingerprinting

RF fingerprinting is a promising paradigm for identifying wireless devices by extracting the unique features embedded within the electromagnetic waves emitted by transmitters. The presence of analog components (such as digital-to-analog converters, band-pass filters, frequency mixers, and power amplifiers) in the radio transmission chain and inherent randomness in the manufacturing process are the primary causes of these distinctive

features [48]. A number of wireless devices using various standards, such as CRN, UMTS, Wi-Fi, push-to-talk transmitters, Bluetooth, and radio-frequency identification (RFID), have been evaluated for RF fingerprinting. It has been shown that every transmitter has a distinct RF fingerprint, and it is also demonstrated how unlikely it is for two transmitters to share the same RF features. Accordingly, such a unique RF fingerprint can be used to authenticate the identity of a particular wireless transmitter and guarantee the confidentiality of the messages that are sent [48,49].

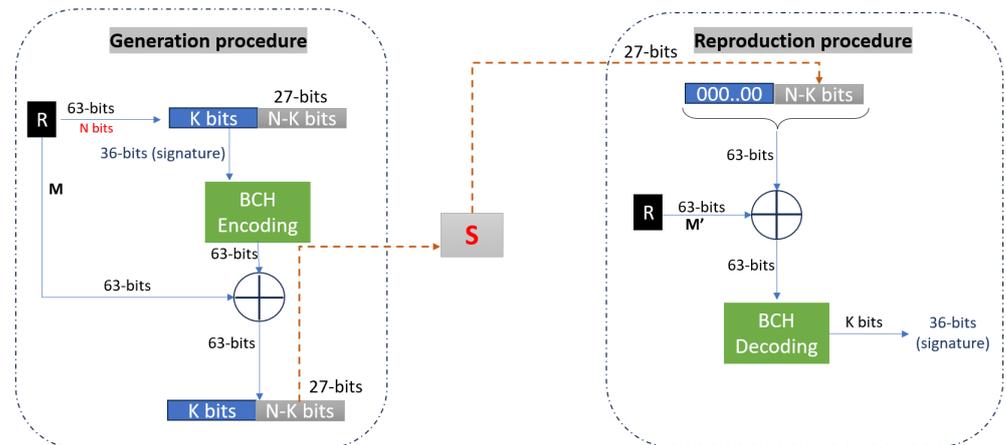
### 3.3. Fuzzy Extractor

Biometric data are inherently noisy and have limitations in terms of achieving high precision. To overcome these limitations, fuzzy extractors are pursued in security applications to enable stable matching. Fuzzy extractors transform biometric data into random strings, while secure sketches (SS) are used to build fuzzy extractors [50]. The secure sketches generate public helper data strings that correct input errors with minimal information leakage. If the *Hamming distance* between the *input* to the fuzzy extractor and the observed one,  $\widehat{input}$ , is less than  $T$  during the reproduction process,  $\widehat{input}$  can be corrected, even if there are  $T$  bit errors [51–53].

Fuzzy extractors have been widely utilized in hardware-based security primitives, such as PUFs and RF fingerprinting. Specifically, these extractors can rectify the erroneous bits present in the PUF response caused by environmental noise. Additionally, they can also be employed in RF fingerprinting to alleviate the impact of noise during radio signal generation, which can potentially affect the identification of the transmitter (node ID). For instance, in narrow-band systems, the oscillator imperfections of carrier frequency offset (CFO) and phase noise, mixer imperfections of in-phase (I) and quadrature (Q) imbalance, power amplifier (PA) non-linearity, and antenna patterns constitute the RF impairments [54]. These impairments can distort the signal emitted by a device, and a receiver captures the physical waveform to extract these impairments to determine the sender's identity. By utilizing a fuzzy extractor, the noise affecting the physical waveform can be mitigated, thereby enhancing the accuracy in identifying device IDs [55]. Thus, the integration of fuzzy extractors in hardware-based security primitives offers a robust solution to the issues of environmental noise and RF impairments.

Figure 2 shows the architecture of the adapted fuzzy extractor in this paper, where  $\mathbf{M}$  refers to the original data. As shown in the figure, the generation procedure takes the original template,  $\mathbf{M}$ , as input and returns a public helper string,  $S$ . The reproduction procedure takes noisy data,  $\mathbf{M}'$ , and public helper string,  $S$ , as input, and outputs  $K$  if  $\mathbf{M}$  and  $\mathbf{M}'$  are close, s.t. *Hamming distance*  $(\mathbf{M}, \mathbf{M}') \leq T$  [51,56]. In the context of PUFs, consider a response with 63 bits ( $N = 63$ ). Assuming a key size of 36 bits ( $K = 36$ ), the generation process involves feeding the  $K$  bit into the BCH encoder, resulting in a 63-bit value. This value is then subjected to an XOR operation with the original (i.e., non-noisy) 63-bit response. The  $(N - K)$  bit of the resulting value, which is 27 bits in this instance, constitutes a helper string,  $S$ . In the presence of noisy data,  $\mathbf{M}'$ , the helper string,  $S$ , is XORed with the noisy 63-bit response,  $\mathbf{M}'$ , as illustrated in Figure 2. The resulting value is then subjected to BCH decoding, yielding the noise-mitigated version of the PUF response, represented as  $\mathbf{M}'$ .

When implementing a fuzzy extractor scheme, it is imperative to take into account two critical factors: information reconciliation and privacy amplification. The former guarantees the removal of noise from the collected noisy data, while the latter ensures the uniform distribution of derived key bits. Recent works have utilized a cryptography function, such as SHA-256, and BCH code to satisfy these fundamental requirements [57,58]. Our proposed protocol employs a fuzzy extractor proposed by Hyunho et al. [56]. This approach incorporates a Bose–Chaudhuri–Hocquenghem (BCH) in the secure sketch to address the issue of compensating noise in biometrics while ensuring an information-theoretic security of  $K$ , where  $K$  refers to the key size [59]. The data perturbation refers to either PUF response bits or RF fingerprinting signals.



**Figure 2.** The fuzzy extractor diagram based on the syndrome ( $N = 63$  bits).

#### 4. System Model and Approach Overview

In this section, we state the assumptions about the underlying network operation, and highlight the security threat, and provide an overview on how FeRHA mitigates such a threat.

##### 4.1. System and Attack Models

FeRHA operates under the assumption that all IoT devices participating in the authentication process have embedded PUFs, as well as a fuzzy extractor. The authentication process necessitates the use of a strong PUF. The latter involves the generation of a large number of CRPs to render it infeasible for an adversary to try all possible challenge bit-strings, i.e., brute force. The paper's presentation is based on the use of an arbiter-PUF, as discussed in Section 3.1. However, the proposed authentication protocol can be applied to other strong PUF designs. Due to the high noise that affects PUFs and RFF, a fuzzy extractor is utilized to mitigate the effect of such noises on the authentication tokens. In other words, fuzzy extractors are proposed to enable stable matching. Our proposed protocol leverages the non-cryptographic fuzzy extractor, as elaborated in Section 3.3 [56].

Given the random and uncontrollable variations among the manufacturing processes, a PUF cannot be physically cloned. However, the PUF operation can be mimicked, something that is often referred to as a modeling attack. Basically, an adversary may eavesdrop on the network to intercept the prover's response to the various verifiers. The captured CRPs are then used to train a machine learning model of the PUF's behavior. Upon capturing a sufficiently large subset of CRPs, the accuracy of the model grows high enough to predict the PUF response to any challenge bit-string. Considering the major role that mutual authentication of IoT plays, an attacker would target one of the legitimate nodes to model its PUF and impersonate such a node to become part of the network. Quite a few machine learning techniques have been shown to be effective for such a purpose [60]. Nonetheless, in validating FeRHA, we assume that the adversary applies powerful techniques, namely, convolutional neural networks (CNNs) and extreme gradient boosting (XGBoost), in order to show FeRHA's resilience to modeling attacks. In addition to PUF modeling attempts, an eavesdropper may replay the intercepted authentication messages to impersonate legitimate IoT nodes. As explained in the balance of the paper, FeRHA thwarts the aforementioned impersonation threat by obfuscating the PUF response of node  $\delta_x$ , known as the prover, and each other IoT device, which acts as a verifier,  $\delta_y$ , while factoring in the RFF of the communicating nodes.

##### 4.2. Approach Overview

Our proposed protocol supports mutual authentication for IoT nodes. Two hardware primitives, namely, RFF and PUFs, are employed in the authentication process. Recall that

the biometric data are subjected to high noise levels, which subsequently could result in the corruption of the authentication token and rejecting connection requests from legitimate nodes. Therefore, we consider mitigating the noise in the utilized biometric to enhance the authentication efficiency by using the fuzzy extractor. Mainly, FeRHA aims to achieve the following design objectives:

1. Develop a novel lightweight PUF-based mechanism for authenticating IoT devices. Rather than applying encryption, our mechanism pursues two-factor authentication by combining the benefits of PUFs and RF fingerprinting.
2. Given the dynamic nature of IoT systems, intermediaries should be avoided, and authentication should be conducted in a decentralized manner.
3. The system should resist attempts by adversaries to eavesdrop and gain knowledge of many CRPs of the embedded PUF to model it using machine learning techniques.
4. To ensure accurate authentication, it is essential to mitigate the noise in biometric data.

The involvement of PUFs enables the concealment of device secrets and mitigates the threat of device hacking. To satisfy the above design goals, FeRHA adopts a distributed authentication strategy in which the network operation is not dependent on a trusted server. Only device enrollment could benefit from engaging a server. In contrast to existing PUF-based solutions, FeRHA does not retain CRPs of a node during the enrollment phase, nor does it incorporate a cryptosystem [26,61]. To elaborate, during the enrollment phase, an IoT node obtains a number of challenge bit-strings and their associated authentication tokens for every other node in the network. Specifically, a node,  $\delta_y$ , will obtain a set,  $\gamma_{x \rightarrow y}, \forall x \neq y$ , of challenges and tokens for each node,  $\delta_x$ , as shown in Figure 3. Thus, in the event that  $\gamma_{x \rightarrow y}$  is compromised by an eavesdropper, any attempts to replicate the PUF of  $\delta_x$  would fail. Additionally, the verifier,  $\delta_y$ , will factor in the physical layer features of the  $\delta_x$ 's transmitter. In this regard, IQ is considered as the fundamental attribute for RFF extraction, although the selection of any other feature can be made, based on the network environment. Then, the prover,  $\delta_x$ , will utilize the fuzzy extractor to generate the helper string,  $\mathcal{H}$ , for PUFs and RFF, s.t.  $\Pi(R_i^x) = \mathcal{H}_{R_i^x}, \forall i \in \gamma_{i \rightarrow j}$ , and  $\Pi(RFF_x) = \mathcal{H}_{RFF_x}$ ; where  $\Pi(\cdot)$  denotes the fuzzy extractor function, and  $\mathcal{H}_{R_i^x}$  and  $\mathcal{H}_{RFF_x}$  are the helper strings for  $R_i^x$  and  $RFF_x$ , respectively. The FeRHA protocol is discussed in detail in the next section.

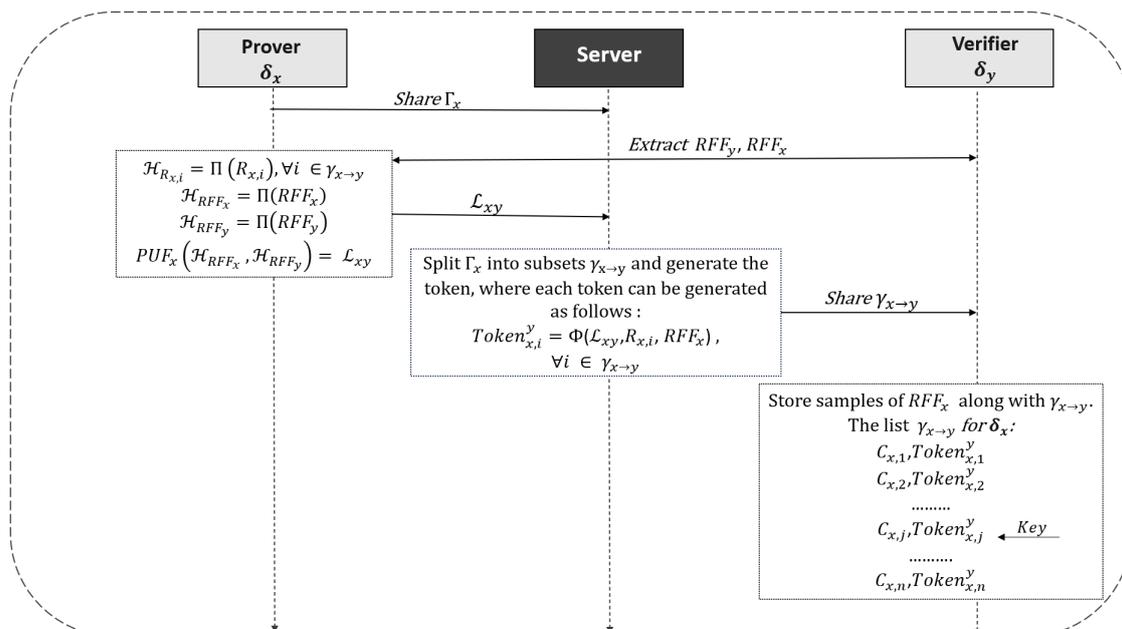


Figure 3. Illustrative diagram of FeRHA during the enrollment phase.

## 5. Combined PUF- and RFF-Based Authentication

Mainly, FeRHA has two phases: enrollment and operation. The enrollment phase deals with the information that nodes need to provide when joining the network. On the other hand, the operation phase focuses on how the nodes interact with various IoT devices while participating in different applications. The two phases are explained below.

### 5.1. Enrollment Phase

IoT has a vast conceptual coverage, encompassing both spatial and temporal dimensions. However, in practical terms, the interaction between devices is more localized. As a result, a node,  $\delta_y$ , does not need to authenticate numerous IoT devices. The practicality of FeRHA is evident in its ability to facilitate mutual authentication between nodes without the need for an intermediary trusted server. However, it is important to note that a server is indispensable during the device enrollment process to effectively manage the sharing of token list  $\gamma$ . During the enrollment of a device, it is necessary to establish connectivity with a local server that maintains an index of active IoT devices. For instance, in the domain of IoV (Internet of Vehicles), the department of motor vehicles is a viable candidate for serving as the server. In an IoV setting, if a vehicle intends to participate in peer-to-peer data sharing on the road, it should retrieve some CRPs for other affiliated vehicles.

Throughout the enrollment process, each IoT node is required to provide specific information in order to be integrated into the network. We assume that the enrollment phase is conducted through a secure channel. In our proposed protocol, the CRP list of a node will not be explicitly stored at any other IoT node. This feature is designed to enhance the security of the protocol by preventing an eavesdropper from capturing the CRPs of a node and model its PUF. Upon enrollment, an IoT device,  $\delta_x$ , will send to the server a CRP list,  $\Gamma_x$ , which reflects a small subset of all possible challenge–response combinations. Recall that FeRHA assumes that strong PUFs are employed. For each CRP in  $\Gamma_x$ , node  $\delta_x$  generates helper data and stores them in its memory. We note that this does not introduce any notable vulnerability since it is impossible to infer the response of a challenge using the corresponding helper data. In addition, the size of  $\Gamma_x$  is limited and would not constitute much of a burden for the IoT node. For example, storing helper data of 8 bits for 1024 CRPs would take only 8K bits.

The server assigns each potential verifier (i.e., IoT nodes other than  $\delta_x$ ) a distinct subset of  $\Gamma_x$ . To clarify, let us assume that the network consists of four IoT nodes, namely  $\delta_x, \delta_y, \delta_z$ , and  $\delta_w$ . As shown in the example in Figure 4, each node will obtain a subset of  $\Gamma_x$ , such that  $\gamma_{x \rightarrow z} \neq \gamma_{x \rightarrow w}$ , and  $\gamma_{x \rightarrow z} \cup \gamma_{x \rightarrow w} \neq \Gamma_x, \forall z \neq w$ . For each of these subsets, the response of the challenge is obfuscated in a manner that depends on the verifier. In essence, the obfuscated response becomes an authentication *Token*. Using the case of  $\delta_x$  and  $\delta_y$  as an example, the associated token,  $Token_{x,i}^y$ , in  $\gamma_{x \rightarrow y}$  is formed as follows:

$$Token_{x,i}^y = \Phi(\mathcal{L}_{xy}, R_{x,i}, RFF_x), \quad \forall (C_{x,i}, R_{x,i}) \in \gamma_{x \rightarrow y} \quad (3)$$

where  $R_{x,i}$  is the corresponding response of  $C_{x,i}$  using  $PUF_x$ , while  $RFF_x$  is the RF fingerprint for  $\delta_x$ .  $\Phi(\cdot)$  is a function to obfuscate the PUF response,  $R_{x,i}$ , using  $RFF_x$ ; yet such obfuscation is made to depend on the verifier,  $\delta_y$ , by factoring in  $\mathcal{L}_{xy}$ . For example,  $\Phi(\cdot)$  could be a Boolean function where  $\mathcal{L}_{xy}$  is used as an operator, or could be a bit-shuffling (rotate) function that uses  $\mathcal{L}_{xy}$  to determine the shuffling pattern or rotation count. The principal objective of performing this operation is to enable the receiver node to deduce the response bit-string since the actual response is not included in the authentication packet. Therefore, an interceptor of authentication packets will be prevented from knowing the response. FeRHA leverages the embedded PUF to introduce the verifier-dependent obfus-

cation by setting  $\mathcal{L}_{xy} = PUF_x(\mathcal{H}_{RFF_x}, \mathcal{H}_{RFF_y})$ , where  $\mathcal{H}_{RFF_x}$  and  $\mathcal{H}_{RFF_y}$  are the helper strings for  $RFF_x$  and  $RFF_y$ , and created by the fuzzy extractor as follows:

$$\mathcal{H}_{RFF_x} = \Pi(RFF_x) \tag{4}$$

$$\mathcal{H}_{RFF_y} = \Pi(RFF_y) \tag{5}$$

Finally, one of the tokens will be defaulted to serve as a key. This could simply implicit across the network, where each verifier,  $\delta_y$ , assumes the first token, or generally the  $j$ th in  $\gamma_{x \rightarrow y}$  to be used as a key, and the corresponding challenge should not be included in an authentication request. The use of such a key will be explained in the next subsection.

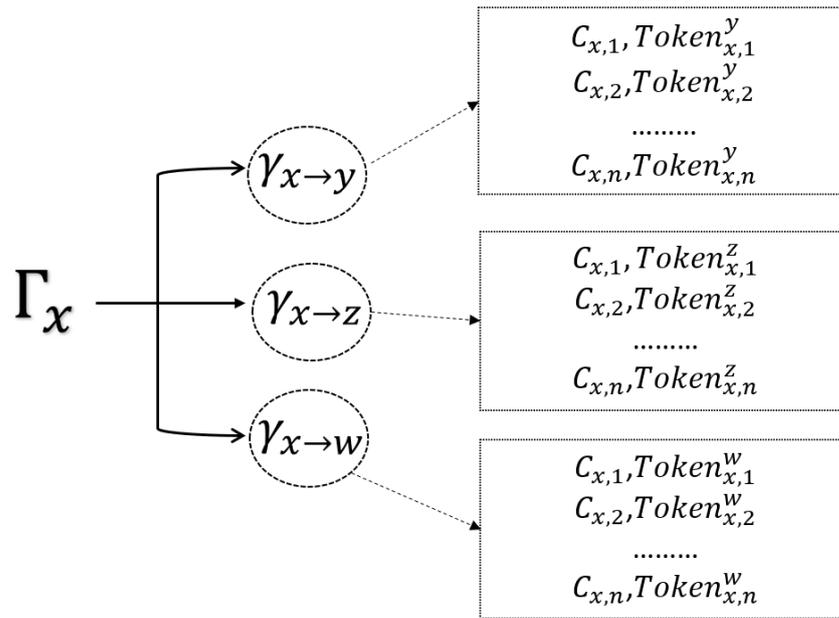


Figure 4. An illustrative example for constructing  $\gamma_x$ .

### 5.2. Operation Phase

After enrollment, a node can apply the FeRHA protocol to authenticate other IoT devices in the system. For such authentication, FeRHA utilizes a two-factor approach, which leverages the unique characteristics of the transmitted waveform to identify the device. This identification mechanism enables FeRHA to obfuscate the PUF response, circumventing the need for the incorporation of cryptographic primitives. Assume that  $\delta_x$  (prover) and  $\delta_y$  (verifier) need to authenticate each other. The main steps of FeRHA are covered in Algorithm 1. The verifier,  $\delta_y$ , will extract the RFF,  $\widehat{RFF}_{x'}$ , of the prover,  $\delta_x$ , and send it along with of the challenges,  $C_{x,i}$ , that is included in  $\gamma_{x \rightarrow y}$ . Therefore, the received signal of node  $\delta_x$  can be written as:

$$\Omega_{\delta_x}(t) = [W_{\delta_x \delta_y}(t) * D(A)]C_{x,i} + n(t) \tag{6}$$

where  $W(t)$  is the wireless channel and  $*$  denotes the conventional operation.  $D(\cdot)$  is the overall effect of hardware distortion on the transmitted analog signal,  $A$ , and  $n(t)$  is the noise. By extracting  $C_{x,i}$ ,  $\delta_x$  will input  $C_{x,i}$  to its PUF,  $PUF_x$ , to generate the corresponding response bit-string as follows:

$$\widehat{R}_{x,i} = PUF_x(C_{x,i}) \tag{7}$$

where  $\widehat{R}_{x,i}$  is the noise-affected response. FeRHA mitigates the noise in the generated biometric,  $RFF$ , and the PUF response,  $R$ , as discussed in Sections 3 and 4. Hence,  $\delta_x$  will

apply the fuzzy extractor to the generated response bit-string  $\widehat{R}_{x,i}$  and  $\widehat{RFF}_x$  while utilizing the helper string extracted during the enrollment phase, as follows:

$$R_{x,i} = \Pi(\widehat{R}_{x,i}, \mathcal{H}_{R_{x,i}}) \tag{8}$$

$$RFF_x = \Pi(\widehat{RFF}_x, \mathcal{H}_{RFF_x}) \tag{9}$$

where  $\mathcal{H}_{R_{x,i}}$  and  $\mathcal{H}_{RFF_x}$  are the helper bit-strings generated during the enrollment process for  $R_{x,i}$  and  $RFF_x$ , respectively.  $R_{x,i}$  and  $RFF_x$  denote the noise-mitigated versions of  $\widehat{R}_{x,i}$  and  $\widehat{RFF}_x$ , respectively.

In the context of FeRHA, the storage of the actual list of CRPs is not required. Instead, FeRHA has introduced a token list,  $\gamma$ , which comprises a series of challenge bits that are associated with an authentication token, *Token*, as illustrated in Figure 3. The use of a token list,  $\gamma$ , in FeRHA provides a secure mechanism for verifying the authenticity of a device while eliminating the exposure of the actual CRPs list. Thus, in the event of an eavesdropper gaining access to the token list, the ability to model  $\delta_x$ 's PUF will be hindered. This is because the eavesdropper will erroneously associate  $C$  with the authentication *Token*, rather than with the PUF response,  $R$ , ultimately leading to a flawed model of  $PUF_x$ . The authentication token corresponding to  $C_{x,i}$ , denoted as  $Token_{x,i}^y$ , is specific for verifier  $\delta_y$  and is in essence a bit-string that is mapped based on  $RFF_x$  and  $R_x$ . This mapping process is performed by utilizing an obfuscation function, represented by  $\Phi$ , as shown in Algorithm 1. In order for node  $\delta_x$  to construct  $Token_{x,i}^y$ , it will generate a factor,  $\mathcal{L}_{xy}$ , that is dependant on the communication link with  $\delta_y$  using its PUF, as follows:

$$\mathcal{L}_{xy} = PUF_x(\mathcal{H}_{RFF_x}, \mathcal{H}_{RFF_y}) \tag{10}$$

where  $\mathcal{H}_{RFF_x}$  and  $\mathcal{H}_{RFF_y}$  are the helper strings for  $RFF_x$  and  $RFF_y$ , respectively, which are generated using the fuzzy extractor during the enrollment phase. Thereafter,  $\delta_x$  will need to form the authentication token,  $\widehat{Token}_{x,i}^y$ , as illustrated in Figure 5, by applying the obfuscation function,  $\Phi$ , as follows:

$$\widehat{Token}_{x,i}^y = \Phi(\mathcal{L}_{xy}, R_{x,i}, RFF_x) \tag{11}$$

The FeRHA protocol incorporates an obfuscated counter, denoted as *count*, into the sending packet to provide protection against replay attacks. This mechanism restricts access to *count* and prevents unauthorized interception of the transmitted packets between  $\delta_x$  and  $\delta_y$ , as detailed in the following subsection. Additionally,  $\widehat{Token}_{x,i}^y$  is masked using a pre-selected key, designated as *Key*, which is chosen from  $\gamma_{x \rightarrow y}$ . It is important to note that this token and its accompanying challenge cannot be utilized for authentication and are assigned solely as a key. Therefore, the transmitted packet can be constructed as follows:

$$\{Key \oplus count, Key \oplus \widehat{Token}_{x,i}^y\} \tag{12}$$

Then,  $\delta_x$  will send the above packet to  $\delta_y$ . The mapping of the authentication token, *Token*, is not uniform across different nodes. This lack of uniformity means that an attacker attempting to eavesdrop on communication between node  $\delta_x$  and various verifiers,  $\delta_y$ ,  $\delta_z$ , and  $\delta_w$ , will not succeed in discovering *Token*. This is because the mapping is not exact, and  $\Phi$  does not yield an identical mapping across different verifiers. Therefore, such an attempt is futile. Lastly,  $\delta_y$  will compare  $Token_{x,i}^y$  that is received during the enrollment phase, with  $\widehat{Token}_{x,i}^y$ . If  $\widehat{Token}_{x,i}^y$  matches  $Token_{x,i}^y$ , then the authentication succeeds; otherwise, the received authentication token,  $Token_{x,i}^y$ , will be rejected.

**Algorithm 1** FeRHA( $\delta_x, \delta_y$ )

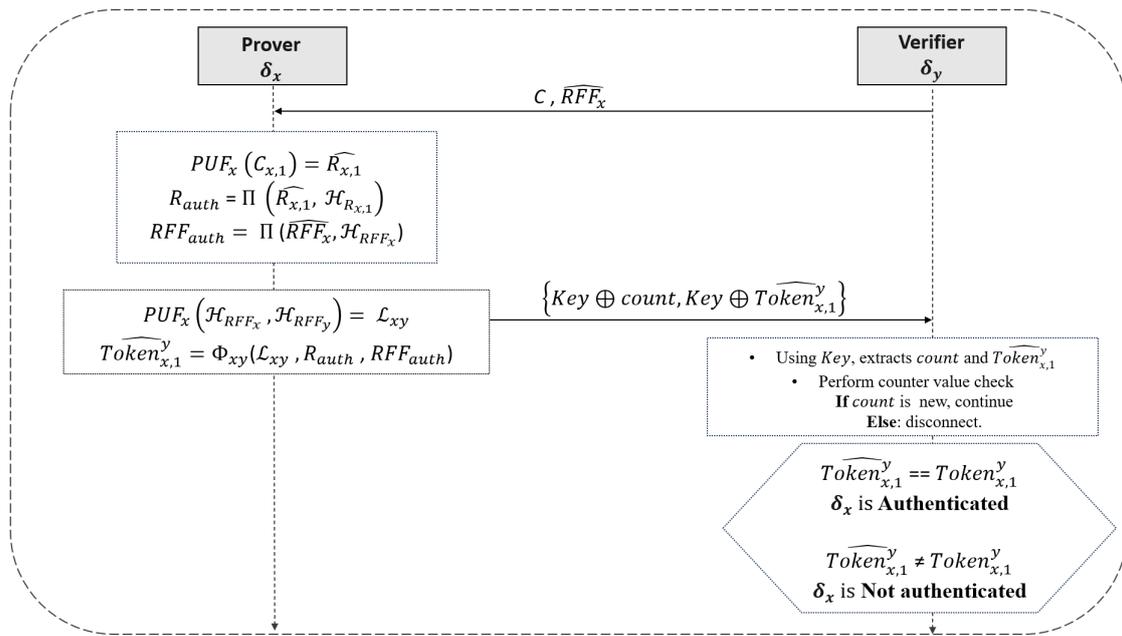
**Require:** Let  $\gamma_{x \rightarrow y}$  be the token list of  $\delta_x$  that shared with  $\delta_y$ .  $\Phi$  is the response obfuscation function, where  $\mathcal{H}_{RFF}$  and  $\mathcal{H}_{R_{x,1}}$  are the helper string for  $RFF$  and  $R_{x,1}$ , respectively.

- 1:  $\delta_y$  extracts  $\widehat{RFF}_x$
- 2:  $\delta_y$  sends  $(C_{x,i}, \widehat{RFF}_x)$  to  $\delta_x$ , where  $(C_{x,i}, \widehat{Token}_{x,i}^y) \in \gamma_{x \rightarrow y}$
- 3:  $\delta_x$  applies  $C_{x,i}$  to its PUF, i.e.,  $PUF_x(C_{x,i}) = \widehat{R}_{x,i}$
- 4:  $\delta_x$  applies FE to  $\widehat{R}_{x,i}$  and  $\widehat{RFF}_x$ , s.t.  $R_{x,i} = \Pi(\widehat{R}_{x,i}, \mathcal{H}_{R_{x,i}})$ , and  $RFF_x = \Pi(\widehat{RFF}_x, \mathcal{H}_{RFF_x})$
- 5:  $\delta_x$  determines  $\mathcal{L}_{xy}$  using its PUF, where  $\mathcal{L}_{xy} = PUF_x(\mathcal{H}_{RFF_x}, \mathcal{H}_{RFF_y})$
- 6:  $\delta_x$  calculate  $\widehat{Token}_{x,i}^y = \Phi(\mathcal{L}_{xy}, R_{x,i}, RFF_x)$
- 7:  $\delta_x$  masks  $\widehat{Token}_{x,i}^y$  as  $\{Key \oplus count, Key \oplus \widehat{Token}_{x,i}^y\}$  and send it to  $\delta_y$
- 8: Using  $Key$ ,  $\delta_y$  extracts  $count$  and  $\widehat{Token}_{x,i}^y$ , and performs  $count$  value check.
- 9: **if**  $count$  is new **then**
- 10:      $continue$ .
- 11: **else**
- 12:      $disconnect$ .
- 13: **end if**
- 14:  $\delta_y$  compare  $\widehat{Token}_{x,i}^y$  and  $Token_{x,i}^y$ ; If equals,  $\delta_x$  is *authenticated*, otherwise  $\delta_x$  is *rejected*

**5.3. Replay Attack Mitigation**

Replay attacks pose a significant threat to communication protocols, particularly in situations where message integrity and authenticity are paramount. Such an attack occurs when an adversary captures a legitimate packet and re-transmits it, potentially resulting in misleading a receiver into accepting a previously valid but now outdated or redundant message. To effectively mitigate replay attacks, FeRHA employs a novel counter-based mechanism that relies upon the unique collaboration between the prover and the verifier. To elaborate, we initialize a counter value  $count$  to be intimately linked to the verifier's state. The mechanism is represented in the following steps:

- **Initializing the counter:** the initialization value of  $count$  is established through the application of  $PUF_x$  to the helper string  $\mathcal{H}_{RFF_y}$ . The mathematical representation of this initialization process is  $count = PUF_x(\mathcal{H}_{RFF_y})$ . This method assures that the initial state of the counter is unpredictable and associated with the hardware characteristics of the involved devices, providing a robust foundation for the counter's integrity.
- **Counter obfuscation:** to mitigate the risk of interception or exploitation of  $count$ , we factor in the token that is shared between the prover and verifier. To obscure the counter value, an XOR operation is employed to mask  $count$  with the  $Key$ , resulting in  $(count \oplus Key)$ , as shown in Figure 5. The  $Key$  value is chosen from the token list  $\gamma_{x \rightarrow y}$  through a pre-agreement between the prover,  $\delta_x$ , and the verifier,  $\delta_y$ ; such a pre-agreement could be simply determined by the server by setting a default entry in the pairwise challenge-token lists for the entire IoT network, as noted in Section 5.1. It should be noted that  $Key$  will not be used as an authentication token, but rather assigned and selected to be the  $Key$ . This mechanism is dependent on the verifier side, thereby rendering an adversary who attempts to eavesdrop over multiple links to target  $\delta_x$  unsuccessful, since  $\gamma_{x \rightarrow y} \neq \gamma_{x \rightarrow z}$ , as illustrated in Figure 4. This operation is instrumental in ensuring that the actual counter value transmitted over the network remains indiscernible and cannot be directly leveraged by an adversary.
- **Verification operation:** upon receiving a packet, the  $Key$  is used to perform an XOR operation, enabling the recovery of the original counter value,  $count$ . Thereafter, the verifier performs a counter value check, ensuring that it is new and has not been previously utilized during the session.



**Figure 5.** A sequence diagram to illustrate the message exchange between  $\delta_x$  and  $\delta_y$  during the operation phase.

## 6. Results and Performance Analysis

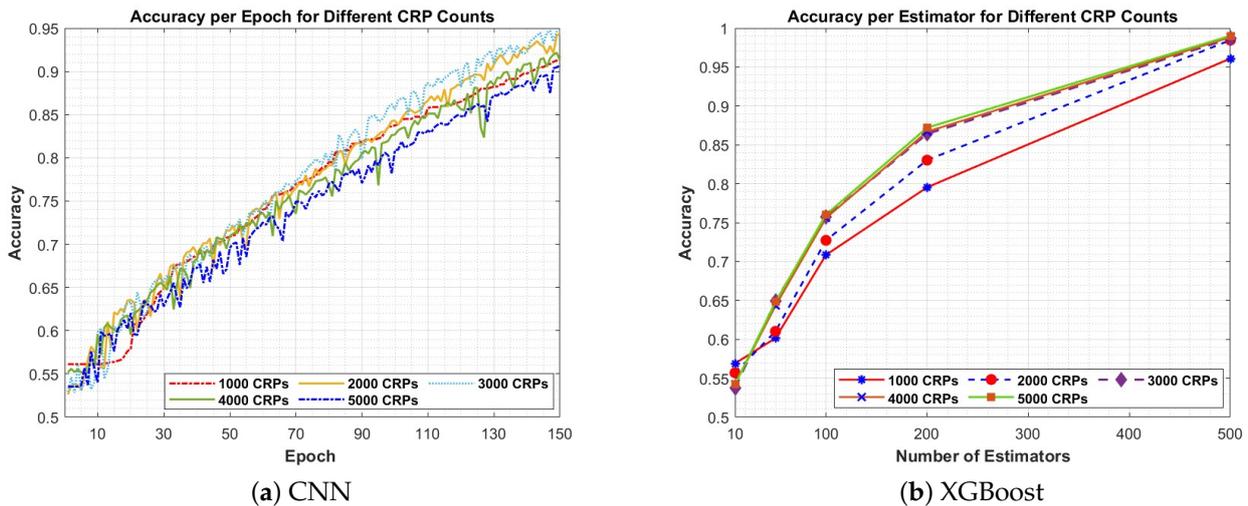
In this section, we provide a detailed account of the implementation settings that were used to validate FeRHA, and highlight important aspects of the authentication process.

### 6.1. FeRHA Implementation

To validate the effectiveness of FeRHA, we used a CRP dataset collected using the arbiter-PUF, explained in Section 3.1, implemented on a Xilinx ARTIX-7 FPGA. The PUF is used to map 64-bit challenge bit-strings to 64-bit responses. The RFF is picked based on a dataset of Wi-Fi transmissions based on the IEEE 802.11a/g standard. Such a dataset was obtained through the DARPA RFMLS program and published by ORACLE [62]. The data were collected using an experimental setup of a USRP software-defined radio (SDR), wherein the fixed USRP B210 designated as the receiver. The data frames were emitted from bit-similar USRP X310 radios that conformed to the IEEE 802.11a standard. The data frames were generated utilizing the MATLAB WLAN system toolbox and contained random payloads with shared address fields. Subsequently, the data frames were streamed to the designated SDR for over-the-air wireless transmission. The receiver SDR samples the incoming signals at a sampling rate of 5 MS/s and a center frequency of 2.45 GHz for Wi-Fi. This paper investigates the potential use of machine learning techniques as part of a PUF modeling attack against FeRHA. Specifically, we employ CNN and XGBoost as representative models that an adversary might utilize to perform the attack.

The successful execution of a cyberattack, including impersonation, data forgery, and man-in-the-middle attacks, is dependent upon the exposure of the underlying device secrets. In the specific context of PUFs, this requires the attacker to accurately model the challenge–response mapping, achieved through machine learning (ML) techniques. It is imperative to note that a key advantage of PUF design is its tamper-resistant nature, and the device secrets (i.e., CRPs) are not stored in the memory. Therefore, our investigative analysis has been focused on thwarting modeling attacks. In our initial attempt to model the arbiter-PUF, we employed CNN and XGBoost without the utilization of our approach. We conducted experiments using varying dataset sizes. The results are depicted in Figure 6. XGBoost achieved a prediction accuracy rate of 99% when using 5000 CRPs, as reported in Figure 6b. To further validate our approach, we carried out another set of experiments using CNN. The CNN architecture comprised an input layer with 32 nodes, along with

two hidden layers that contained 64 and 128 neurons, respectively. An output layer was integrated with a sigmoid activation function [63]. The first three layers utilized a rectified linear activation function (Relu) to achieve optimal performance. The *Adam optimizer* [64] was utilized to update the weights, while *binary cross entropy* served as the loss function with a learning rate of 0.0001. The model was trained for 150 epochs, with a batch size of 64. Our results indicated that the NN successfully modeled the PUF, with a prediction accuracy rate ranging from 92% to 99%, as depicted in Figure 6a. FeRHA thwarts the threat of ML-based modeling vulnerability, as demonstrated next.



**Figure 6.** Accuracy of modeling 64-bit arbiter-PUF using NN and XGBoost for various training set sizes (CRP count).

## 6.2. Performance Results

During the operation phase, an eavesdropper may attempt to intercept communication between entities, e.g.,  $\delta_x$  and  $\delta_y$ , to obtain a significant number of CRPs. These data can then be used to mimic the PUF and impersonate the legitimate node  $\delta_x$ . In order to prevent such an attack, FeRHA pursues a two-factor authentication protocol that effectively obfuscates exchanged response bits “R”. To evaluate the efficacy of FeRHA, we conducted two distinct attack scenarios based on the attack model discussed earlier in Section 4. These scenarios were designed to model the behavior of  $PUF_x$  and impersonate  $\delta_x$ . In the first scenario, an attacker intercepts the transmission of a single communication link between a verifier,  $\delta_y$ , and a prover,  $\delta_x$ , with the goal of modeling the PUF of  $\delta_x$ . In the second scenario, an attacker eavesdrops on links between a prover,  $\delta_x$ , and multiple verifiers, and uses the intercepted packets, i.e., the exchanged CRPs to model the PUF of  $\delta_x$ . The results for these attack scenarios are discussed in detail below.

- **Single-link attack:** in this scenario, we focus on the authentication of a prover,  $\delta_x$ , by a verifier,  $\delta_y$ , where an adversary may intercept transmissions between them. It is important to note that the adversary’s primary objective is to capture as many CRPs as possible for  $PUF_x$  during the operation phase of FeRHA. Specifically, the adversary aims to obtain the maximum number of CRPs for the prover’s  $PUF_x$ . Recall, FeRHA utilizes two primitives, PUFs and RFF, to generate an authentication token. Notably, the PUF response,  $R$ , is not included in the verifier’s request. Instead, the obfuscation function,  $\Phi$ , is applied to produce an authentication token,  $Token_{x,i}^y$ , for every challenge bit-string,  $C_{x,i}$ , where  $\Phi$  factors in the pairwise link between  $\delta_x$  and  $\delta_y$ . Thus, an adversary will record the exchanged  $Token_{x,i}^y$  as the PUF response and, consequently, will use these tokens as input to the machine learning model, with the aim of mapping the challenge bits to the corresponding token. The results of the modeling attack under FeRHA are depicted in Figure 7. We performed the modeling attack using the ML models applied earlier in Figure 6 to model the PUF with the

application of FeRHA. As demonstrated by the results, an adversary cannot achieve an accuracy of more than 53% and 56% using CNN and XGBoost, respectively. Such an accuracy is equivalent to a random guess, given the binary nature of the PUF response. The effectiveness of FeRHA in preventing modeling attacks is clearly evident from the comparison of the results in Figure 7 with those in Figure 6.

- Multi-link attack: in this scenario, we assume that an adversary is actively eavesdropping on all links that the prover,  $\delta_x$ , has with other nodes. The adversary’s primary objective is to collect as many CRPs of  $PUF_x$  as possible, with the ultimate goal of modeling  $PUF_x$  and impersonating  $\delta_x$ . It is worth noting that, in contrast to the single-link scenario, the adversary will intercept the exchanged packets between  $\delta_x$  and multiple verifiers, such as  $\delta_y, \delta_z, \delta_w$ , and so on. Hence, the applied function,  $\Phi$ , will be varied among the intercepted packet (since  $\mathcal{L}_{xy} \neq \mathcal{L}_{xz} \neq \mathcal{L}_{xw}$ ). Despite the adversary’s ability to capture more data, the semantic of such data is not coherent since the tokens are derived from the PUF response differently for the various verifiers. The results depicted in Figure 8 illustrate the outcomes of utilizing CNN and XGBoost models in conjunction with a 64-bit PUF integrated into  $\delta_x$ . It is important to note that the CNN and XGBoost models have demonstrated the ability to model the 64-bit arbiter-PUF with an accuracy of over 90% (see Figure 6). However, Figure 8 indicates that neither CNN nor XGBoost was successful in defeating FeRHA. The results reflect the accuracy of the modeling attack when applying intercepted packets over connections containing varying numbers of verifiers. The results of such a modeling attack confidently indicate that the adversary cannot gain any advantages from eavesdropping on additional links, as the accuracy remains unchanged compared with that of a single link.

### 6.3. Formal Security Analysis

This section presents a formal security assessment of our protocol’s efficacy by utilizing the automated validation of internet security protocols and applications (AVISPA) tool [65]. The modular and expressive formal language of the AVISPA tool is widely recognized for providing the specification of protocols and their security attributes. AVISPA also integrates the on-the-fly model checker (OFMC) that leverages state-of-the-art automatic analysis techniques to validate the robustness of security protocols; such a property is referred to as “Safety” in AVISPA. We have created sessions for interactions among provers and verifiers using FeRHA. The security goals for the AVISPA simulation are the authentication of a prover and the secrecy of its PUF response. The tool classifies whether the protocol is SAFE or UNSAFE, or INCONCLUSIVE. Our protocol’s formal security verification results are shown in Figure 9. As indicated by the results, FeRHA is deemed SAFE, implying robustness against man-in-the-middle, replay, and impersonation attacks.

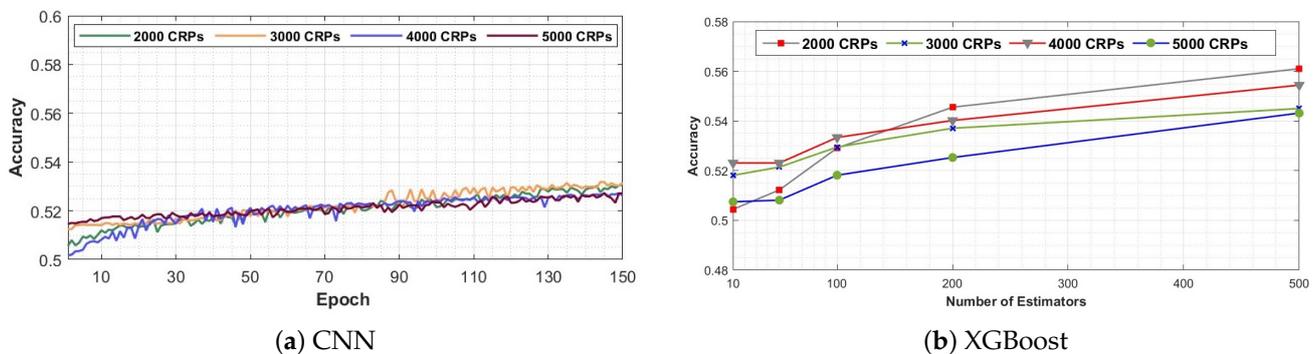


Figure 7. The PUF modeling accuracy when  $PUF_x$  is under a single-link attack while applying FeRHA.

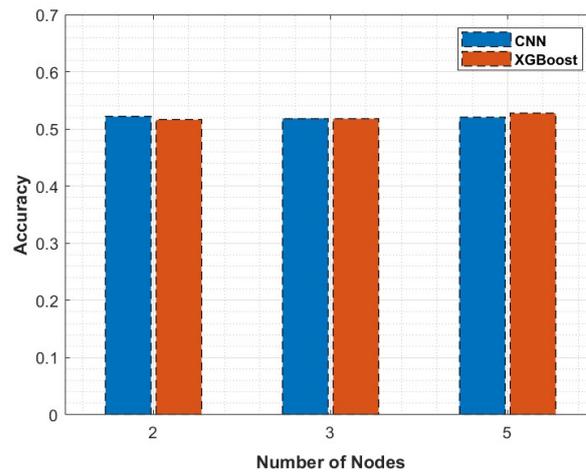


Figure 8. The PUF modeling accuracy when FeRHA is subjected to a multi-link attack.

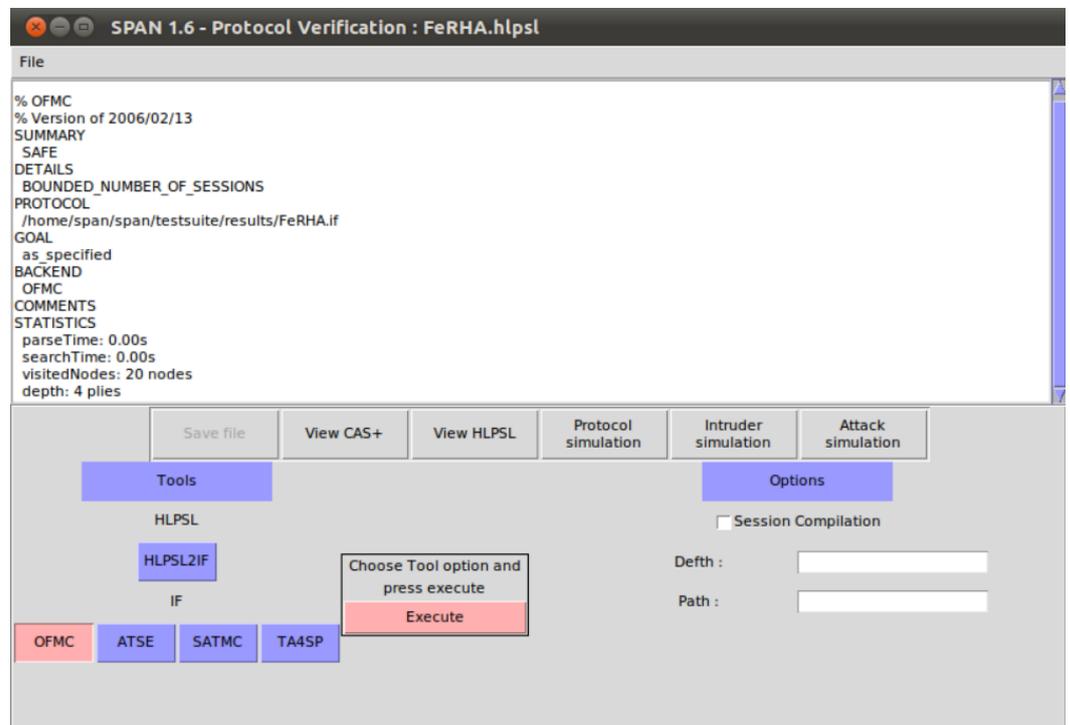


Figure 9. The result of the analysis of FeRHA using OFMC.

#### 6.4. Fuzzy Extractor Performance

To evaluate the reliability of our implementation, we conducted tests using the fuzzy extractor in combination with the generated PUF responses and RFF. The implementation diagram of the fuzzy extractor utilizing the BCH code and syndrome concept ( $N = 63$ ) is illustrated in Figure 2. Our assessment of the proposed fuzzy extractor’s performance is based on Table 2. This table showcases all conceivable combinations of message length  $L$  for BCH code, where the code-word length is fixed to 63 bits. Notably, the variable  $t$  signifies the maximum number of bits that the fuzzy extractor can correct for the corresponding key size. Given that FeRHA aims to mitigate the impact of noise, we have evaluated the performance when the fuzzy extractor is employed in conjunction with the PUF by considering three temperature settings: 70 °C, 30 °C, and 15 °C. To clarify, the PUF is used to map 64-bit challenge bit-strings to 64-bit responses at a baseline temperature of 30 °C. To factor in the effect of noise, the same challenge bit-strings were applied to generate 64-bit responses at a lower temperature of 15 °C and at a higher temperature of 70 °C.

**Table 2.** Number of correctable errors in the BCH code for  $N = 63$ .

Index	$N$	$L$	$t$
1	63	57	1
2	63	51	2
3	63	45	3
4	63	39	4
5	63	36	5
6	63	30	6
7	63	24	7
8	63	18	8
9	63	16	9
10	63	10	10
11	63	7	11

The notion of PUF reliability refers to how consistent its response is to a given challenge under varying operating conditions, e.g., fluctuations in temperature or voltage supply. The reliability of a PUF can be calculated using (13) and (14) below. By applying these equations, the fuzzy extractor effectively could mitigate the temperature noise in the PUF, leading to a reliability of 99.88% when the temperature is 70 °C and 99.8% for 15 °C. The key size considered in this experiment is 45.

$$HD_{INTRA} = \frac{1}{m} \sum_{j=1}^m \frac{HD(R_j(n), R_j(n))}{n} \times 100\% \quad (13)$$

From (13), the PUF reliability can be written as:

$$Reliability = 100\% - HD_{INTRA} \quad (14)$$

where  $HD$  is the hamming distance,  $m$  denotes the number of calculated samples/chips, and  $n$  is the number of bits.

We have repeated the same experiment using *RFF* data. According to ORACLE's published dataset [62], the noise is modeled as a Gaussian variable. We note that noise can cause alterations in the demodulated IQ sample pattern, resulting in a slightly different pattern from the original noise-free IQ sample pattern. To address this issue, we leveraged a fuzzy extractor to minimize the noise to below  $-13$  dB. This action ensures that the Earth Mover's Distance (EMD) between the original and altered patterns remains below the defined ORACLE threshold of 0.1. EMD was computed using the formula expressed in Equation (15). EMD is a widely used metric for measuring similarities between two multi-dimensional distributions. To clarify, consider two sets of points in a two-dimensional metric space, denoted by  $\mathbb{M}^2$ . Let  $I$  and  $J$  be two subsets of  $\mathbb{M}^2$ , each of equal size, denoted by  $|I| = |J|$ . Let  $F$  be the set of all possible bijections, which are one-to-one and onto mappings from  $I$  to  $J$ . The EMD between  $I$  and  $J$  can be calculated using this set  $F$  as follows:

$$EMD(I, J) = \min_{f \in F} \sum_{i \in I} \|i - f(i)\|. \quad (15)$$

In other words, EMD is calculated by identifying the smallest possible sum of Euclidean distances between points in two given sets ( $I$  and  $J$ ), while considering all valid bijections between them ( $f : A \rightarrow J$ ). EMD provides a reliable metric for measuring similarity accurately. A smaller EMD value indicates a greater level of similarity between the two patterns, and vice versa. In our experiments, EMD has consistently stayed below 0.1, reflecting a highly reliable RFF, which demonstrates the effectiveness of the employed fuzzy extractor in mitigating the noise. Overall, the reliability values discussed above

demonstrate that the fuzzy extractor has successfully mitigated the noise, reducing the impact of the noise.

## 7. Conclusions and Future Work

This paper has presented FeRHA, a novel two-factor authentication protocol that is specifically designed for IoT devices. The protocol addresses the critical need for secure device authentication and achieves this by integrating PUFs and RFF. One of the key contributions of our protocol is its ability to facilitate mutual authentication between two devices without relying on a trusted third party. This enhances the overall security of IoT networks by minimizing potential vulnerabilities and performance bottlenecks that are associated with centralized authentication mechanisms. Additionally, our design achieves resilience to the inherent noise present in PUFs and RFFs, ensuring reliable authentication performance under various operational conditions. This resilience is further enhanced by the implementation of obfuscation techniques that safeguard shared authentication data against potential eavesdropping and modeling attacks aimed at compromising the security primitives. Our validation and testing demonstrate the efficacy of our protocol in mitigating PUF modeling attacks, launched by applying prominent machine learning techniques. When XGBoost and CNN were applied, FeRHA could diminish the PUF response prediction accuracy to below 57%, which is a major drop from the 95% plus accuracy when FeRHA is not employed. In the future, we plan to test our protocol's performance by using a prototype IoT network. We also aim to explore the feasibility of implementing FeRHA in emerging fields such as the Internet of Vehicles (IoV). Secure communication and authentication in vehicular networks can be challenging due to their high mobility and dynamic nature. Therefore, applying FeRHA in the IoV context could provide reliable and resilient authentication mechanisms for connected vehicles and roadside infrastructure.

**Author Contributions:** Conceptualization, M.A. and M.Y.; Methodology, M.A. and M.Y.; Software, M.A., A.A. and S.S.M.; Validation, M.A., M.Y. and A.A.; Investigation, M.A., M.Y. and A.A.; Data curation, M.A., M.Y. and A.A.; Writing—original draft, M.A. and M.Y.; Writing—review & editing, M.Y.; Supervision, M.Y.; Project administration, M.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy and ethical restrictions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Elijah, O.; Rahman, T.A.; Orikumhi, I.; Leow, C.Y.; Hindia, M.N. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [[CrossRef](#)]
2. Younis, M. Internet of everything and everybody: Architecture and service virtualization. *Comput. Commun.* **2018**, *131*, 66–72. [[CrossRef](#)]
3. Halperin, D.; Heydt-Benjamin, T.S.; Ransford, B.; Clark, S.S.; Defend, B.; Morgan, W.; Fu, K.; Kohno, T.; Maisel, W.H. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In Proceedings of the 2008 IEEE Symposium on Security and Privacy (sp 2008), Oakland, CA, USA, 18–22 May 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 129–142.
4. Arias, O.; Wurm, J.; Hoang, K.; Jin, Y. Privacy and security in internet of things and wearable devices. *IEEE Trans.-Multi-Scale Comput. Syst.* **2015**, *1*, 99–109. [[CrossRef](#)]
5. Papp, D.; Ma, Z.; Buttyan, L. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 21–23 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 145–152.
6. Al-Naji, F.H.; Zagrouba, R. A survey on continuous authentication methods in Internet of Things environment. *Comput. Commun.* **2020**, *163*, 109–133. [[CrossRef](#)]
7. Mohanta, B.K.; Jena, D.; Satapathy, U.; Patnaik, S. Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet Things* **2020**, *11*, 100227. [[CrossRef](#)]

8. Williams, P.; Dutta, I.K.; Daoud, H.; Bayoumi, M. A survey on security in internet of things with a focus on the impact of emerging technologies. *Internet Things* **2022**, *19*, 100564. [[CrossRef](#)]
9. Halak, B.; Zwolinski, M.; Mispan, M.S. Overview of PUF-based hardware security solutions for the internet of things. In Proceedings of the 2016 IEEE 59th International Midwest Symposium on Circuits and Systems (MWSCAS), Abu Dhabi, United Arab Emirates, 16 October–19 October 2016; pp. 1–4.
10. McGrath, T.; Bagci, I.E.; Wang, Z.M.; Roedig, U.; Young, R.J. A puf taxonomy. *Appl. Phys. Rev.* **2019**, *6*, 011303. [[CrossRef](#)]
11. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 44th Annual Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
12. Lounis, K.; Zulkernine, M. Lessons learned: Analysis of PUF-based authentication protocols for IoT. *Digit. Threat. Res. Pract.* **2023**, *4*, 1–33. [[CrossRef](#)]
13. Alkanhal, M.; Younis, M. P-MAP: PUF-based Mutual Authentication Protocol. In Proceedings of the ICC 2022-IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3424–3429.
14. Rojas, P.; Alahmadi, S.; Bayoumi, M. Physical Layer Security for IoT Communications—A Survey. In Proceedings of the 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 14 June–31 July 2021; pp. 95–100.
15. Zhang, Z.; Guo, X.; Lin, Y. Trust Management Method of D2D Communication Based on RF Fingerprint Identification. *IEEE Access* **2018**, *6*, 66082–66087. [[CrossRef](#)]
16. Xu, Q.; Zheng, R.; Saad, W.; Han, Z. Device Fingerprinting in Wireless Networks: Challenges and Opportunities. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 94–104. [[CrossRef](#)]
17. Ellis, K.; Serinken, N. Characteristics of radio transmitter fingerprints. *Radio Sci.* **2001**, *36*, 585–597. [[CrossRef](#)]
18. Xie, F.; Wen, H.; Li, Y.; Chen, S.; Hu, L.; Chen, Y.; Song, H. Optimized coherent integration-based radio frequency fingerprinting in Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 3967–3977. [[CrossRef](#)]
19. Shao, Y.; Liu, J.; Zeng, Y.; Gong, Y. A Radio Frequency Fingerprinting Scheme Using Learnable Signal Representation. *IEEE Commun. Lett.* **2024**, *28*, 73–77. [[CrossRef](#)]
20. Gu, H.; Su, L.; Zhang, W.; Ran, C. Attention is Needed for RF Fingerprinting. *IEEE Access* **2023**, *11*, 87316–87329. [[CrossRef](#)]
21. Ashtari, A.; Shabani, A.; Alizadeh, B. A new RF-PUF based authentication of internet of things using random forest classification. In Proceedings of the 2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), Mashhad, Iran, 28–29 August 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 21–26.
22. Román-Castro, R.; López, J.; Gritzalis, S. Evolution and trends in IoT security. *Computer* **2018**, *51*, 16–25. [[CrossRef](#)]
23. Mall, P.; Amin, R.; Das, A.K.; Leung, M.T.; Choo, K.K.R. PUF-based authentication and key agreement protocols for IoT, WSNs, and Smart Grids: A comprehensive survey. *IEEE Internet Things J.* **2022**, *9*, 8205–8228. [[CrossRef](#)]
24. Alkanhal, M.; Alali, A.; Younis, M. A Distributed Lightweight PUF-Based Mutual Authentication Protocol for IoV. *IoT* **2024**, *5*, 1–19. [[CrossRef](#)]
25. Yoon, S.; Kim, B.; Kang, Y.; Choi, D. Puf-based authentication scheme for iot devices. In Proceedings of the 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 21–23 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1792–1794.
26. Chatterjee, U.; Govindan, V.; Sadhukhan, R.; Mukhopadhyay, D.; Chakraborty, R.S.; Mahata, D.; Prabhu, M.M. Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database. *IEEE Trans. Dependable Secur. Comput.* **2018**, *16*, 424–437. [[CrossRef](#)]
27. Yilmaz, Y.; Do, V.H.; Halak, B. ARMOR: An anti-counterfeit security Mechanism for IOW cost Radio frequency identification systems. *IEEE Trans. Emerg. Top. Comput.* **2020**, *9*, 2125–2138. [[CrossRef](#)]
28. Qureshi, M.A.; Munir, A. PUF-IPA: A PUF-based identity preserving protocol for Internet of Things authentication. In Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 10–13 January 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–7.
29. Patil, A.S.; Hamza, R.; Hassan, A.; Jiang, N.; Yan, H.; Li, J. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts. *Comput. Secur.* **2020**, *97*, 101958. [[CrossRef](#)]
30. Fakroon, M.; Gebali, F.; Mamun, M. Multifactor authentication scheme using physically unclonable functions. *Internet Things* **2021**, *13*, 100343. [[CrossRef](#)]
31. Alladi, T.; Chamola, V. HARCHI: A two-way authentication protocol for three entity healthcare IoT networks. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 361–369. [[CrossRef](#)]
32. Nimmy, K.; Sankaran, S.; Achuthan, K. A novel lightweight PUF based authentication protocol for IoT without explicit CRPs in verifier database. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 6227–6242. [[CrossRef](#)]
33. Majzoobi, M.; Rostami, M.; Koushanfar, F.; Wallach, D.S.; Devadas, S. Slender PUF protocol: A lightweight, robust, and secure authentication by substring matching. In Proceedings of the 2012 IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 24–25 May 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 33–44.
34. Farha, F.; Ning, H.; Ali, K.; Chen, L.; Nugent, C. SRAM-PUF-based entities authentication scheme for resource-constrained IoT devices. *IEEE Internet Things J.* **2020**, *8*, 5904–5913. [[CrossRef](#)]
35. Nozaki, Y.; Yoshikawa, M. Secret sharing schemes based secure authentication for physical unclonable function. In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Piscataway, NJ, USA, 23–25 February 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 445–449.

36. Jiang, Q.; Zhang, X.; Zhang, N.; Tian, Y.; Ma, X.; Ma, J. Three-factor authentication protocol using physical unclonable function for IoT. *Comput. Commun.* **2021**, *173*, 45–55. [[CrossRef](#)]
37. Mahalat, M.H.; Karmakar, D.; Mondal, A.; Sen, B. PUF based secure and lightweight authentication and key-sharing scheme for wireless sensor network. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2021**, *18*, 1–23. [[CrossRef](#)]
38. Nouichi, D.; Abdelsalam, M.; Nasir, Q.; Abbas, S. IoT Devices Security Using RF Fingerprinting. In Proceedings of the 2019 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 26 March–10 April 2019; pp. 1–7. [[CrossRef](#)]
39. Zhang, J.; Shen, G.; Saad, W.; Chowdhury, K. Radio Frequency Fingerprint Identification for Device Authentication in the Internet of Things. *IEEE Commun. Mag.* **2023**, *61*, 110–115. [[CrossRef](#)]
40. Tian, Q.; Lin, Y.; Guo, X.; Wen, J.; Fang, Y.; Rodriguez, J.; Mumtaz, S. New Security Mechanisms of High-Reliability IoT Communication Based on Radio Frequency Fingerprint. *IEEE Internet Things J.* **2019**, *6*, 7980–7987. [[CrossRef](#)]
41. Peng, L.; Zhang, J.; Liu, M.; Hu, A. Deep Learning Based RF Fingerprint Identification Using Differential Constellation Trace Figure. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1091–1095. [[CrossRef](#)]
42. Chatterjee, B.; Das, D.; Maity, S.; Sen, S. RF-PUF: Enhancing IoT security through authentication of wireless nodes using in-situ machine learning. *IEEE Internet Things J.* **2018**, *6*, 388–398. [[CrossRef](#)]
43. Bari, M.F.; Chatterjee, B.; Sivanesan, K.; Yang, L.L.; Sen, S. High accuracy RF-PUF for EM security through physical feature assistance using public Wi-Fi dataset. In Proceedings of the 2021 IEEE MTT-S International Microwave Symposium (IMS), Los Angeles, CA, USA, 18–27 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 108–111.
44. Rührmair, U.; Holcomb, D.E. PUFs at a glance. In Proceedings of the 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 24–28 March 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
45. Maes, R.; Verbauwhede, I. Physically unclonable functions: A study on the state of the art and future research directions. In *Towards Hardware-Intrinsic Security: Foundations and Practice*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 3–37. [[CrossRef](#)]
46. Guo, Y.; Dee, T.; Tyagi, A. Multi-block APUF with 2-Level Voltage Supply. In Proceedings of the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Hong Kong, China, 8–11 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 327–332.
47. Hemavathy, S.; Bhaaskaran, V.K. Arbiter PUF-A Review of Design, Composition, and Security Aspects. *IEEE Access* **2023**, *11*, 33979–34004. [[CrossRef](#)]
48. Rehman, S.U.; Sowerby, K.W.; Alam, S.; Ardekani, I. Radio frequency fingerprinting and its challenges. In Proceedings of the 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 496–497.
49. Parmaksız, H.; Karakuzu, C. A review of recent developments on secure authentication using RF fingerprints techniques. *Sak. Univ. J. Comput. Inf. Sci.* **2022**, *5*, 278–303. [[CrossRef](#)]
50. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the Advances in Cryptology-EUROCRYPT 2004: International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; Proceedings 23; Springer: Berlin/Heidelberg, Germany, 2004; pp. 523–540.
51. Wen, Y.; Lao, Y. Efficient fuzzy extractor implementations for PUF based authentication. In Proceedings of the 2017 12th International Conference on Malicious and Unwanted Software (MALWARE), Nancy, France, 19–20 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 119–125.
52. Liu, W.; Zhang, Z.; Li, M.; Liu, Z. A trustworthy key generation prototype based on DDR3 PUF for wireless sensor networks. *Sensors* **2014**, *14*, 11542–11556. [[CrossRef](#)]
53. Huffman, W.C.; Pless, V. *Fundamentals of Error-Correcting Codes*; Cambridge University Press: Cambridge, UK, 2010.
54. Zhu, Z.; Leung, H.; Huang, X. Challenges in reconfigurable radio transceivers and application of nonlinear signal processing for RF impairment mitigation. *IEEE Circuits Syst. Mag.* **2013**, *13*, 44–65.
55. Zhang, J.; Woods, R.; Sandell, M.; Valkama, M.; Marshall, A.; Cavallaro, J. Radio frequency fingerprint identification for narrowband systems, modelling and classification. *IEEE Trans. Inf. Foren. Secur.* **2021**, *16*, 3974–3987. [[CrossRef](#)]
56. Kang, H.; Hori, Y.; Katashita, T.; Hagiwara, M.; Iwamura, K. Cryptographic key generation from PUF data using efficient fuzzy extractors. In Proceedings of the 16th International Conference on Advanced Communication Technology, PyeongChang, Republic of Korea, 16–19 February 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 23–26.
57. Böhm, C.; Hofer, M. *Physical Unclonable Functions in Theory and Practice*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
58. Herder, C.; Ren, L.; Van Dijk, M.; Yu, M.D.; Devadas, S. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Trans. Dependable Secur. Comput.* **2016**, *14*, 65–82. [[CrossRef](#)]
59. Teoh, A.B.J.; Kim, J. Error correction codes for biometric cryptosystem: An overview. *Inf. Commun. Mag.* **2015**, *32*, 39–49.
60. Rührmair, U.; Sölter, J. PUF modeling attacks: An introduction and overview. In Proceedings of the 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 24–28 March 2014; pp. 1–6.
61. Wallrabenstein, J.R. Practical and secure IoT device authentication using physical unclonable functions. In Proceedings of the 2016 IEEE 4th International Conference on Future Internet of Things and cloud (FiCloud), Vienna, Austria, 22–24 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 99–106.

62. Sankhe, K.; Belgiovine, M.; Zhou, F.; Riyaz, S.; Ioannidis, S.; Chowdhury, K. ORACLE: Optimized Radio Classification through Convolutional neural Networks. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 370–378.
63. Santikellur, P.; Bhattacharyay, A.; Chakraborty, R.S. Deep learning based model building attacks on arbiter PUF compositions. *Cryptol. Eprint Arch.* **2019**.
64. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
65. Armando, A.; Basin, D.; Cuellar, J.; Rusinowitch, M.; Viganò, L. Avispa: Automated Validation of Internet Security Protocols and Applications. *ERCIM News*, January 2006, p. 64.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.