SN Computer Science (2020) 1:282
https://doi.org/10.1007/s42979-020-00303-y

ORIGINAL RESEARCH

Machine Learning Attacks and Countermeasures for PUF-Based IoT Edge Node Security

Vishalini R. Laguduva[1] · Srinivas Katkoori[1] · Robert Karam[1]

Received: 1 June 2020 / Accepted: 11 August 2020 / Published online: 27 August 2020

Abstract
The Internet of things (IoT) ecosystem has grown exponentially with the convergence of various technologies such as deep learning, sensor systems, and advances in computing platforms. With such a highly pervasive nature of "smart" devices, the nature of data being collected and processed can be increasingly private and require safeguards to ensure the data's integrity and security. Physically unclonable functions (PUFs) have emerged as a lightweight, viable security protocol in the Internet of Things (IoT) framework. Malicious modeling of PUF architectures has proven to be difficult due to the inherently stochastic nature of PUF architectures. In this work, we show that knowledge of the underlying PUF structure is unnecessary to clone a PUF. We tackle the problem of cloning PUF-based edge nodes in different settings such as unencrypted, encrypted, and obfuscated challenges in an IoT framework. We present a novel non-invasive, architecture-independent, machine learning attack for robust PUF designs and can handle encryption and obfuscation-based security measures on the transmitted challenge response pairs (CRPs). We show that the proposed framework can successfully clone different PUF architectures, including those encrypted using two (2) different encryption protocols in DES and AES and with varying degrees of obfuscation. We also show that the proposed approach outperforms a two-stage brute force attack model. Finally, we offer a machine learning-based countermeasure, a discriminator, which can distinguish cloned PUF devices and authentic PUFs with an average accuracy of 96%. The proposed discriminator can be used for rapidly authenticating millions of IoT nodes remotely from the cloud server.

Keywords Machine learning · Internet of things · Physically unclonable functions · Edge node security

Introduction

The Internet of things (IoT) ecosystem has grown exponentially with the convergence of various technologies such as deep learning, sensor systems, and advances in computing platforms. The advent of 5G technology and the promise of higher bandwidth is expected to increase the highly connected nature of today's IoT ecosystem. The massive collection of ubiquitous and pervasive devices in the IoT ecosystem has been deployed across a variety of environments to collect and process massive amounts of data. Applications of IoT devices range from wearable computing devices, bio-implantable devices to monitor vital bodily functions for direct human interaction, as well as for "smart" devices that we interact with on a day-to-day basis. With such a highly pervasive nature of "smart" devices, the nature of data being collected and processed can be increasingly private and require safeguards to ensure the integrity and security of the data [5, 27].

With such highly private data, IoT nodes need to be adequately authenticated before collecting and processing such data. The authentication protocol can be as simple as storing the secret key on physical, silicon-based devices or as complicated as cryptography-based protocols. Choosing the authentication protocol has the following set of challenges

This article is part of the topical collection "Hardware-Assisted Security Solutions for Electronic Systems" guest edited by Himanshu Thapliyal, Saraju P. Mohanty, Wujie Wen and Yiran Chen.

✉ Vishalini R. Laguduva
vishalini@usf.edu

Srinivas Katkoori
katkoori@usf.edu

Robert Karam
rkaram@usf.edu

[1] CSE Department, University of South Florida, Tampa, FL 33620, USA

that must be addressed: (1) IoT devices are typically resource constrained, thus requiring high energy efficient security protocols; (2) their distributed nature can provide easy physical access to the node; and (3) the highly connected nature of IoT framework requires fast and secure security protocols. Traditional approaches to cryptography, while useful, have not proven to be sufficiently lightweight and fast for IoT device authentication. For example, authentication protocols that require storing the secret key on each node device, while an effective strategy can be bypassed through physical and side-channel attacks on the node device [21] and compromise the integrity of the IoT network and associated data. Recent efforts have shifted to leveraging the inherent randomness induced in silicon devices during the manufacturing process as the secret key, opposed to the traditional binary key stored in silicon devices, which can be susceptible to physical attacks. Such approaches, called physically unclonable functions (PUFs), have helped provide a higher security level against direct physical attacks. This alleviates the need for costly physical protection measures. PUFs have become increasingly popular and have been used for IoT device authentication [1, 2, 4, 6, 7] and other security tasks [25, 30].

Today's IoT nodes are designed such that they are tamper-proof [16, 41], which makes it difficult or impossible for micro-probing. Even if the attacker is successful in micro-probing, given the myriad of PUF architectures in literature, extracting information on the underlying PUF architecture is extremely difficult. Hence, earlier ML-based PUF attacks with the assumption of knowing underlying architecture are either not practical or extremely difficult to stage. Additionally, these methods assume that the challenge is available to the attacker in *plain-text*, i.e., there is no encryption applied to the problem. Given that most communication through a wireless channel is encrypted, these are very strong assumptions to make, especially in the context of node security in an IoT framework. In this work, we present, for the first time, an ML-based attack that does not require PUF architecture information. We also offer a countermeasure for this attack that can be effectively used to evaluate an IoT node's trust level remotely.

We focus on an architecture-independent attack that assumes no prior knowledge of the PUF architecture in the system. We show that observed challenge respose pairs (CRPs) are sufficient to improve the cloning accuracy of a strong PUF irrespective of the underlying architecture. The attack can simulate PUF-based data node without knowing underlying PUF architecture. To evaluate the effectiveness of our approach, we compare against a brute force attack model (Sect. Brute Force Attack on Strong PUFs) that leverages the current advances in PUF-architecture cloning. We leverage architecture-specific cloning [32] through a cascaded framework of (1) PUF architecture identification; (2) employing architecture-specific cloning models; and (3) evaluate the prediction accuracy of the model by combining the architecture classification accuracy and the cloning accuracy in a harmonic mean.

Inspired from the pioneering work of Goodfellow et al. [13] on Generative Adversarial Networks (GANs), we propose a machine learning-based defense, a *discriminator*, to identify the possibility of cloning using any ML-based attack non-invasive attack. Extant countermeasures [23, 28] to ML-based cloning have focused on creating complex cloning-resistant PUF architecture. As we enter into a more realizable IoT ecosystem, complex PUF architectures may not be suitable for lightweight IoT systems. Hence, we propose a lightweight, probabilistic identification of cloning through machine learning. To the best of the authors' knowledge, this is the first such framework for the non-invasive attack of PUF-based IoT network authentication schemes and a proposed mechanism to differentiate original PUFs from cloned ones. In short, our paper makes the following novel contributions:

- propose a non-invasive, architecture-independent cloning attack on strong PUFs,
- show that a brute force attack on strong PUFs to identify the PUF architecture for cloning is increasingly complex and not trivial for feasible cloning,
- show that the proposed approach can successfully clone the PUF model even if the challenge–response pair is encrypted or obfuscated, and
- propose a probabilistic discriminator model to bolster the CRP protocol's security by identifying possible instances of cloning attacks.

In summary, we present one of the first frameworks to clone PUF-based authentication in an IoT setting, without any physical access to the device and any prior knowledge of the underlying PUF architecture. We also show that the approach can be extended, through unsupervised noisy pre-training to handle two (2) standard encryption protocols and three (3) common PUF architectures, which form some of the more common node authentication setups in practice. The preliminary version of this work has been published in [19, 26].

The rest of this paper is organized as follows. Section Background presents the background on physically unclonable functions (PUFs), their usage in IoT nodes, and their security assumptions. Section Related Work briefly reviews existing machine learning attacks on PUFs and corresponding countermeasures. Section Brute Force Attack on Strong PUFs describes and evaluates a baseline brute force approach. Section Architecture-Independent PUF Modeling describes the proposed ML-based attack. Section Machine Learning-Based Countermeasure proposes a countermeasure based on the discriminator model. Section Evaluation and Analysis presents our empirical evaluation of the proposed approach. Finally, Sect. Conclusions draws conclusions.

Background

In this section, we will present the background on silicon-based physically unclonable functions (PUFs), their usage in IoT node authentication, and the common security assumptions.

Silicon-based PUF devices [25] are easily fabricated physical structures that leverage the stochastic nature of the manufacturing process to create physically unclonable, unique identifiers for each manufactured unit. This typically results in a one-way function. Given an electronic stimulus, the response of a PUF device is an unpredictable, repeatable function. This response identifies each device with a unique signature. This is primarily attributed to the interaction of the external stimulus and the physical structure of the PUF. This interaction is termed as the challenge–response pair (CRP), where the *challenge* is the external stimulus, and the PUF's reaction is termed as the *response*. The unpredictable nature of the PUF can be highly sensitive to noise and error correction circuits [20]. This nature of PUF is used to reduce the uncertainty in the PUF's response to make it more reliable. PUFs with a sufficiently large set of challenge–response pairs are called *strong PUFs* and are typically chosen for most practical security applications.

The use of PUFs as the basis for IoT node authentication has gained momentum in recent times [1, 2, 4, 6, 7]. Using PUFs for IoT security protocols typically involves an initial enrollment phase and an authentication protocol during the actual data exchange. Figure 1 illustrates the typical architecture of an IoT network and the generic enrollment protocol. A typical IoT network consists of remote, resource-constrained data nodes $(N_1, N_2, N_3 \ldots N_k)$ connected to static server nodes $(S_1, S_2, S_3 \ldots S_n)$ that transfer the acquired data to the cloud using routers $(R_1, R_2, R_3 \ldots R_m)$. The data is transmitted from the routers to the cloud using a network gateway. IoT edge nodes can range from simple sensors to complex systems with a processor, memory, communication, etc. Strong PUFs implemented in complex IoT nodes are subject to attacks, which is the focus of this work. When a data node is added to the IoT network, the enrollment phase is executed to create a CRP database for the PUF within the data node. This database of CRPs is used in the authentication phase when two nodes corresponding to the same server node want to communicate. The shared server node authenticates both data nodes, generates security key pairs, and helps secure key sharing. While practical, a malicious attacker can use the enrollment phase to eavesdrop and clone the set of

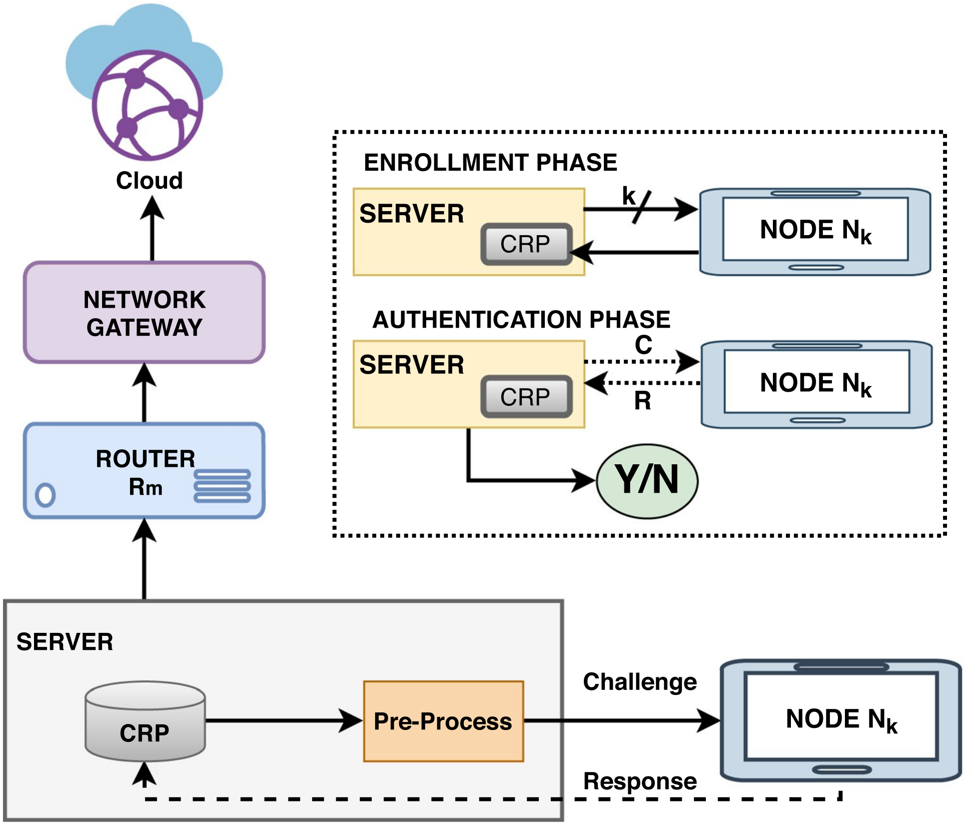


Fig. 1 A typical IoT architecture is illustrated. The inner figure shows the enrollment phase and the authentication phase of a PUF-based IoT node authentication scheme. The pre-process block represents an optional encryption and/or obfuscation process

CRPs, which can be used to bypass the PUF-based authentication and compromise the security of the data nodes. There have been advances that the extraction of CRPs is then destroyed, i.e., fuse the extraction wires, thereby eradicating the possibility of cloning via this method.

Following the protocols established in [24], extant IoT networks using PUF authentication [1, 2, 4, 6, 7] make the following underlying *security assumptions*:

1. cloning a PUF architecture, either physically or mathematically, is a difficult problem, especially if the underlying architecture is unknown;
2. an adversary has unrestricted physical access to the communication channel;
3. the challenge–response characteristics of the PUF within the data IoT node is an implicit property and is not accessible to an adversary; and
4. the attacker can obtain access to the database of CRPs through malicious software attacks.

Given these security assumptions, the goal of the adversary then becomes straightforward. In essence, it must be able to spoof the server nodes into accepting a malicious node on behalf of the original data nodes without actual possession of the node in question. Any physical intrusions can compromise the integrity of the PUF and hence render the attack harmless. The underlying stochastic nature of PUFs and the above constraints lend itself to a robust security protocol that can be hard to breach. However, advances in machine learning have led to a vast majority of non-invasive attacks on PUF-based security. Machine learning-based approaches can be characterized by applying a learned mathematical model on a collected subset of valid CRPs. The curation of such data is typically assumed to be an eavesdropping protocol, which is not an unreasonable assumption. Prior works, especially the pioneering work of Rühmair et al. [32], have shown great success in cloning PUFs, gaining cloning accuracy of up to 99.99%. Such success does come with a caveat—the underlying architecture must be known *a priori*, either through invasive physical intrusions or explicit architecture knowledge.

Related Work

In this section, we briefly summarize related work on machine learning-based attack and prevention techniques in the strong PUF design.

Strong PUF Architectures

A strong PUF can support a large number of complex CRPs with physical access to the PUF for a query such that an attacker cannot generate correct response given finite resources and time [14, 17, 31]. While a weak PUF has only a few CRPs, which makes it difficult for the attack and prediction techniques, hence in this paper, we consider strong PUF. The number of CRPs of strong PUFs can grow exponentially depending on the number of module blocks available for generating responses for a large number of corresponding challenges. Error due to noise in the response of PUF can be minimized using helper data [10, 18]. For completeness, we assume such an error-correction mechanism incorporating temperature, voltage, and aging variations are already present in the PUF to be cloned. A strong PUF does not contain a read-out protection scheme assuming an attacker has to enumerate a large number of CRPs. Hence, it makes an invasive attack infeasible while impelling attackers to apply ML-based techniques to succeed beyond the underlying complexity of strong PUFs. For a detailed analysis of constructions and description of strong PUFs, we refer the reader to [14].

The linear additive behavior of Arbiter PUF (APUF) has made it an ideal target for ML attack. Hence, higher non-linearity in a given PUF architecture can improve the uniqueness and randomness with increased defense against modeling attack. Other approaches to ML-resistant PUFs have been randomized challenges [42], obfuscation [12, 23], and sub-string-based challenges [28]. Rostami et al. presented a prover–verifier framework for successful authentication based on a subset of response substring [28]. Vijayakumar et al. proposed to utilize bagging and boosting ML algorithms to improve the accuracy of classifiers given sufficient entropy of cascading PUFs [39].

The majority of works describing ML-resistant PUFs employ clearly defined architecture and adequately large CRPs for the training process. The randomness and uniqueness, instead, deteriorate substantially when CRPs that do not belong to original CRPs for a particular PUF is used as the case we are tackling in this work. We present a discriminator model that permits the investigation of CRPs received at a PUF challenge–response interface to lower the attacker attempt in reverse-engineering the PUF model.

PUF-Based IoT Security

Physical unclonable functions (PUFs) have, increasingly, been proposed as the basis for node security in the IoT framework [1, 2, 4, 6, 7, 15]. PUF-based IoT node security has primarily been implemented in two ways—CRP-based authentication and PUF-based key generation [37]. In the latter, a PUF's response is typically used to create secret keys for use in traditional cryptography. The PUF's response to a given challenge (processed through an error-correcting circuit) generally is hashed to generate the secret keys. The former approach, i.e., CRP-based authentication, is more widely used, especially with strong PUF models, to create robust authentication protocols. The resulting authentication protocol involves evaluating the identity of a PUF model by a central authentication server by applying a set of pre-defined external challenges and validating the resulting response, as illustrated in Fig. 1. The CRPs are collected in an enrollment phase before deployment, and the resulting database forms the basis of authentication during deployment.

Encryption Protocols for IoT Node Authentication

Given that most of the communication in the IoT framework occurs over an unsecured wireless network, the use of encryption protocols in communication and authentication has become essential [3, 34, 36, 38, 40]. There have been many encryption protocols proposed with the two commonly used protocols being Data Encryption Standard (DES) [8] and the Advanced Encryption Standard (AES) [9]. Given their widespread use and success, there have been numerous cryptanalysis of both protocols and has led to successful attempts on the DES protocol. However, it takes tremendous computational power and large amounts of data to successfully break the DES protocol, whereas the 128-bit AES protocol has not been successfully broken. There have also been some alternatives to encryption protocols such as obfuscated CRPs [12] and substring matching [29], to name a few. In this work, we consider the encryption protocols AES and DES as the encryption mechanisms used for encrypting the CRPs in the IoT framework.

Machine Learning-Based Attacks on PUF Models

Given the growing popularity of PUF-based authentication, there have been numerous attempts to test the approach's effectiveness, primarily through mathematical modeling of the PUF's characteristic function. Rührmair et al. [32] proposed an ML-based attack on strong PUFs based on a predictive model. The authors were able to clone the functionality of the underlying PUF given the PUF model by evaluating model parameters using logistic regression (LR) with resilient backpropagation(RProp) and evolution strategies (ES). Though the method was quite successful in cloning, the attacker needs to know the underlying PUF architecture and the corresponding signature function, which are part of the security assumptions outlined in Sect. 1. While it is reasonable to assume that CRPs can be obtained by eavesdropping or other interfaces [31], it is not always possible to ascertain the underlying PUF model without physical access to the PUF. Although the presented attacks work better under a given PUF size and architectural complexity, an attacker should have the idea of underlying PUF architecture to make the generated clone samples match the statistics of the real CRPs. There have also been other approaches such as PAC [11] and hybrid methods [33] that have successfully cloned PUFs using a combination of ML and invasive techniques.

Brute Force Attack on Strong PUFs

The proposed models by Rühmair et al. [32] allows us to successfully clone strong PUF models with a prediction accuracy of 99.9%. The Brute Force method is a two-step process where we would first need to identify the underlying PUF architecture, as the approaches in [32] require intimate knowledge of the PUF architecture such as PUF type, number of stages and number of XOR gates, to name a few. Once the architecture is identified, we use the prior work to clone the PUF. We use the term brute force, because we search through all possible combination of PUF architectures to clone the PUF.

To address this, we propose the use of a machine learning model to identify the PUF architecture through observation of the challenge–response pairs, as illustrated in Fig. 2. One primary assumption in this approach is that there exists a subset of challenges $\tilde{C} \in C$ that is valid for all PUF architectures in a given network, where C is the collection of all valid CRPs. Given the number of PUF architectures and their use for authentication, this is not an unreasonable assumption.

Identifying PUF Architectures

Given the set of challenges $\tilde{C}$, we can observe the set of valid responses R_{c_i} for each PUF architecture $c_i \in C_{\text{puf}}$, where C_{puf} is the set of all known PUF architectures described in Sect. Related Work. Hence, the objective of the classification is to learn a function f_c which maximizes the probability

$$\arg \max_{\tilde{C}_i \in \tilde{C}} P(c_i|\tilde{C}_i, R_{c_i}), \tag{1}$$

where the objective is to find the PUF architecture c_i given the challenge $\tilde{C}_i$, and the subsequent response R_{c_i}. We use the following machine learning models as the basis for the function $f_c(\cdot)$: logistic regression, artificial neural network, and random forests.

Empirical Evaluation

We evaluate the performance of the proposed brute force attack to identify the architecture of eight common strong PUF architectures. We use a fixed number of randomly sampled 100 CRPs for evaluation for each PUF architecture for a total of 800 CRPs. We report average results from five different runs, with the test set sampled each time randomly. We curate a collection of 100,000 CRPs for training the classification model.

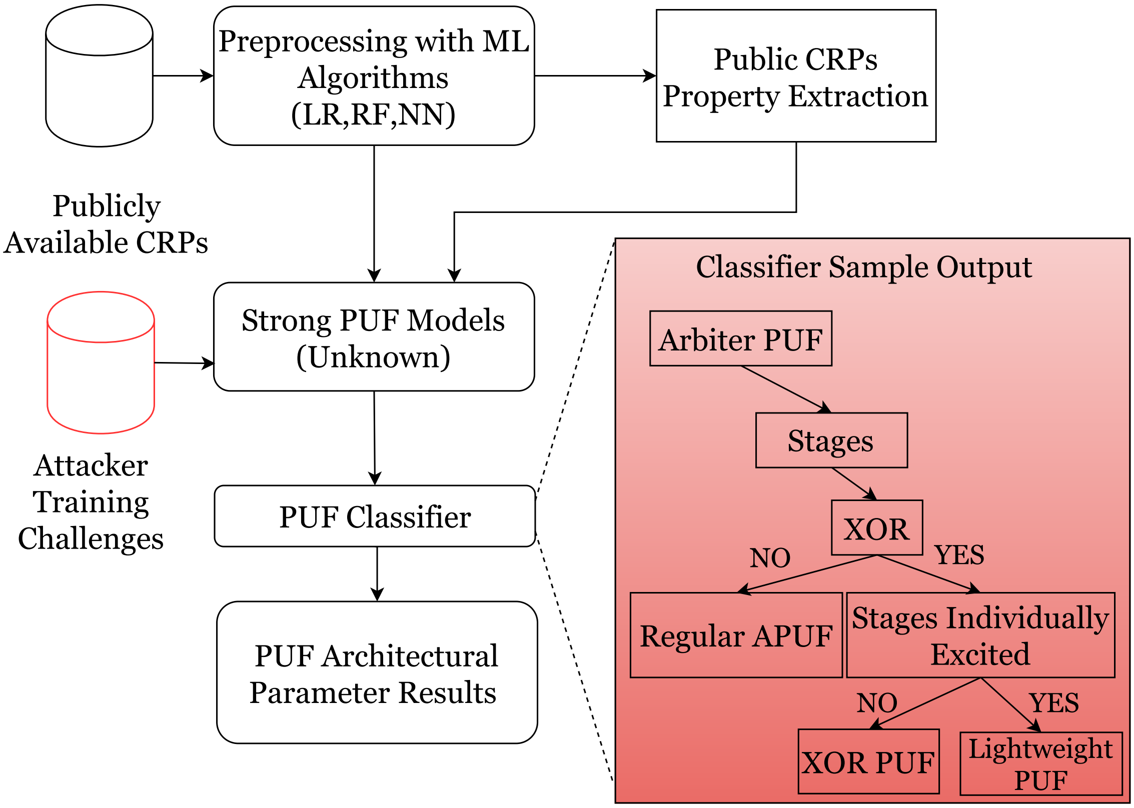

Preprocessing with ML Algorithms (LR,RF,NN)
Public CRPs Property Extraction
Publicly Available CRPs
Strong PUF Models (Unknown)
Attacker Training Challenges
PUF Classifier
PUF Architectural Parameter Results
Classifier Sample Output
Arbiter PUF
Stages
XOR
NO
YES
Regular APUF
Stages Individually Excited
NO
YES
XOR PUF
Lightweight PUF

Fig. 2 The proposed attack model on oblivious PUF architecture. The brute force attack has an additional PUF architecture detection process as indicated by the block in red

Table 1 Brute force attack: PUF architecture classification performance and subsequent cloning accuracy

PUF model	PUF Classification rate (%)	Cloning rate (%)
APUF	81.49	77.42
3 XOR APUF	76.53	72.71
4 XOR APUF	65.01	61.76
5 XOR APUF	63.57	60.39
6 XOR APUF	61.31	58.25
LW 3 XOR APUF	76.91	73.05
LW 4 XOR APUF	65.37	62.10
LW 5 XOR APUF	59.32	56.33

As can be seen from Table 1, identifying the PUF architecture from an observed set of CRPs is not a trivial task. Even with 100% cloning accuracy for a given PUF architecture, identifying the said architecture requires a large set of CRPs for training a model. The maximum performance that we were able to obtain was using the logistic regression model, which took 100 iterations to converge, resulting in the maximum classification rate for Arbiter PUF architecture. There was a large confusion among different design variations of each PUF type. The prediction rate for XOR PUFs decreased as the complexity of the architecture increased. It can be seen that identifying the PUF architecture requires significant training resources of 100,000 CRPs while recognizing the arbiter PUF with an average accuracy of 81.49%. The classifier performed worst on the lightweight PUFs, yielding a maximum identification accuracy for the 3 bit XOR lightweight PUF. The identification rate also affected the cloning prediction rate of the brute force approach, with each misclassified PUF architecture affecting the cloning quality. While the average cloning accuracy can be as high as 77.42% (for the Arbiter PUF), the numbers can be misleading in practice. The performance of the two-stage attack model is rather low; considering the possible gap between the intra-Hamming and inter-Hamming distances of PUF CRPs, this prediction rate cannot be considered to be successful cloning.

Architecture-Independent PUF Modeling

In this section, we describe our proposed approach for a PUF-independent attack model on various PUF architectures by exploiting the CRP authentication protocol. We begin with a discussion on using machine learning models to capture the underlying correlation between challenge–response pairs to model the randomness unique to a given PUF architecture. We then introduce a noisy autoencoder-based pretraining of the neural network model for handling noise and obfuscation-based techniques for more robust feature learning. We then follow with a discussion on defending against such attacks using complementary machine learning models.

Attack Model

Each PUF is made unique through a digital signature characterized by its response to a given challenge. This signature is representative of the randomness encoded in its state due to manufacturing variations and other physical disorders. To compromise the integrity of the CRP protocol, one has to model this randomness to generate a response representative of the PUF's signature. There are two approaches to this problem: a model-based solution and a model-agnostic solution. The model-based solution, explored in [32], attempted to capture this randomness through modeling the characteristics of a PUF using domain knowledge (PUF architecture) and characteristics (delay model, thermal response characteristics, etc.). Thus, the attack consists of a regression of the model's parameters.

However, we consider an architecture-independent approach to the solution by disregarding the need for a characteristic equation for the PUF. We postulate that the challenge and subsequent response of any given PUF is representative of its characteristic function. Thus, modeling the dependency between the various features of a given challenge and the target response allows us to capture the randomness of a given PUF architecture. To this end, we use several approaches to capture the dependency between the challenge and response pairs of various PUF architectures. Since the underlying dependency is not linear or non-linear, we explore several different machine learning models that characterize the dependence with a linear decision boundary (logistic regression) or with a non-linear decision boundary (random forest and artificial neural networks).

The attack model consists of learning the optimal function that maps the given n-bit challenge $C = c_1, c_2, \dots, c_n$ to an appropriate output response $R \in \{-1, 1\}$ with a probability $p(R|C)$. The objective of the attack model is to learn the function $f : C \rightarrow R$ such that the difference between the generated and actual response of the PUF is minimized. Hence, the best attack model is characterized by the search for the optimal function f given by

$$\arg \min_{(C_s, R_s)} E[(\hat{f}(C) - f(C))^2], \tag{2}$$

where $\hat{f}(C)$ is the characteristic function of the given PUF architecture and (C_s, R_s) represents the space of all known challenge–response pairs obtained through the eavesdropping protocol. We search for the optimal function $f(C)$ through the characteristic equation of the different machine learning models defined above. For example, in a logistic regression model, f is defined as

$$f = \arg\max\,(\sigma(R \times d(\mathbf{w}, C))), \tag{3}$$

where $\mathbf{w}$ is a learned vector that represents the decision boundary (d) for the logistic regression model and σ is the logistic function.

Denoising Autoencoders for Robust Feature Learning

While the attack model presented in "Attack Model" can handle clear-text challenges, the encryption protocols such as AES and DES can inject noise into the relationship between the challenge and the response, hence obscuring the characteristic function of the PUF architecture. To account for this, one must either: (1) break the encryption through traditional cryptanalysis; or (2) learn robust representations that can decouple the noise from essential information within the input challenge. Since the computational resource for pursuing the former can be expensive, we take the latter approach and attempt to learn robust representations through unsupervised pretraining using a denoising autoencoder. In this approach, we train a neural network (multilayer perceptron, MLP) as our attack model.

A traditional *autoencoder* is an unsupervised neural network, whose objective is to learn a compressed representation of the input data through a cascaded encoding–decoding operation. The network architecture comprises two neural networks, an encoder network and a decoder network, working together to learn an encoded representation or *latent space*. The encoder's role is to compress the input data into a lower-dimensional representation that captures the underlying pattern of the data by learning to ignore as much of the spurious patterns or *noise* as possible. This compressed representation represents the *bottleneck layer* of the network. The role of the decoder is to learn to reconstruct the original input from this compressed representation. This process is represented in Fig. 3, where it can be seen that the latent space has a lower dimensionality compared to the larger-dimensional input and output. The input and output of the autoencoder framework have the same dimensions. An autoencoder network's training objective is to minimize the reconstruction loss, which is typically an $L2$ loss or binary cross-entropy.

While autoencoders learn useful features (the latent space) that can be used for downstream classification tasks, noise or perturbations in the input can drastically change the representations unless added during training. To account for noise injected through encryption, we

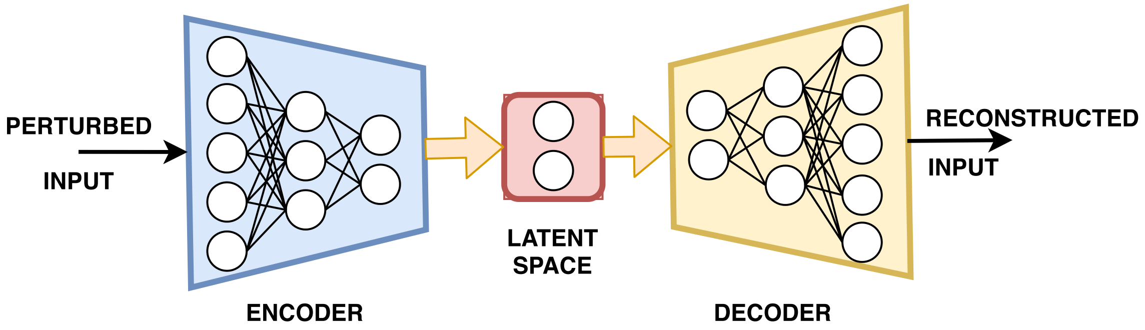


Fig. 3 A typical autoencoder structure is illustrated. In our approach, we use a denoising autoencoder. Hence, the input is a randomly perturbed input and the output is the original, clean challenge

train the autoencoder as a *denoising autoencoder*. The idea is to train the autoencoder to reconstruct the input from a corrupted or randomly perturbed version of the input. This training strategy is applied to force the hidden layer to discover more robust features and prevent it from merely learning the identity function. We construct the denoising autoencoder by adding a stochastic corruption step to the input. While the input can be perturbed in many ways, we want our representations to handle the inherent noise applied to the wireless channel, obfuscation, and encryption. Hence, in our implementation, we apply the following perturbations: (1) randomly mask part of the input by making them zero; (2) add random white noise to the input; and (3) add a hashing function to the CRP to simulate the encryption techniques. At every training iteration, one of the above perturbations is applied to the input, and the output of the decoder network is compared to the original input.

Implementation and Training Details

Due to the complex nature of the proposed network, we present the implementation details for understanding. The encoder network is a four-layer network of fully connected layers. Between each subsequent layer is a dropout layer [35], which helps prevent overfitting. Each dropout layer has a dropout probability of 50%. The number of neurons in each layer is reduced by 0.5× to reduce the dimensionality of the processed data. This follows the standard protocol in autoencoders to induce the bottleneck at the end of the encoding network. The decoding network is a mirror of the encoding network, with the number of neurons increasing to match the output dimensions. We train the network for ten epochs at a learning rate or $1e^{-4}$ using the standard gradient descent optimizer.

Machine Learning-Based Countermeasure

The modeling of the internal randomness of a given PUF architecture puts the integrity of the CRP-based authentication into question. Hence, it becomes critical that we are able to differentiate between the original PUF and an adversarial attack, such as the ones described in Sects. Brute Force Attack on Strong PUFs and Architecture-Independent PUF Modeling. To this end, we introduce a mathematical model that is able to discriminate between an original and a cloned PUF called the *discriminator model*, as illustrated in Fig. 4. The discriminator decides whether each instance of the response belongs to the actual PUF or a malicious attacker. As seen in Fig. 4, the discriminator model takes in the response of the original PUF along with the response of the PUF cloned with several ML attacks as the input to predict whether the PUF is an original or a cloned and returns the probabilities. The cloned part of the response is shown in red. The output of this discriminator is a single scalar value $D(C)$, indicative of an adversarial attack. The value $D(C)$ is a probability function that maps a given response (R) to the distribution belonging to either the original PUF ($\hat{f}(C)$) or an attacker ($f(C)$) for a given n-bit challenge C. Hence, the optimal discriminator model is given by

$$D^{\star}(C,R) = \frac{p(\hat{f}(C))}{p(f(C)) + p(\hat{f}(C))}, \tag{4}$$

where $D^{\star}(C,R)$ is a mathematical model that maps the response R for a given challenge (C) into the probability space of either the original PUF ($\hat{f}(.)$) or the attack model ($f(.)$). Again, we explore the use of well-known machine learning models as the basis for our discriminator mathematical model.

The search space for the optimal discriminator is similarly characterized by the optimization function defined in

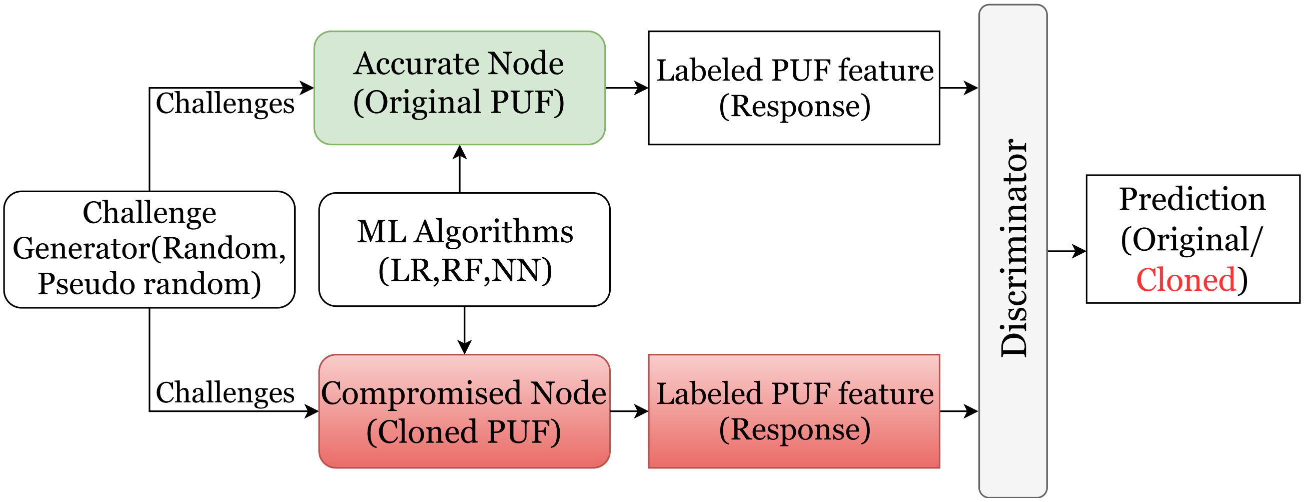

Challenges
Accurate Node (Original PUF)
Labeled PUF feature (Response)
Challenge Generator(Random, Pseudo random)
ML Algorithms (LR,RF,NN)
Challenges
Compromised Node (Cloned PUF)
Labeled PUF feature (Response)
Discriminator
Prediction (Original/ Cloned)

Fig. 4 ML-based discriminator model to ascertain a PUF integrity

Eq. (2). However, the search is represented by the discriminator to distinguish between the original PUF's response and a cloning attack.

The search space for the optimal attack model and discriminator model is defined by the optimizer functions defined in Eq. (2) and its subsequent adaptation for the discriminator, respectively. We employ a simple grid search algorithm to find the optimal attack model ($f(.)$) from a given set of possible models (F). The attack model's space, F, comprises all transformation functions that satisfy the condition $f : C \rightarrow R$. We restrict the search space to the given three machine learning models: logistic regression (LR), random forest (RF), and neural network (NN). We also ensure that the optimal discriminator is chosen from a set of discriminative functions $G(.) \in G_s$, where G_s is the collection of all discriminative functions that optimize the probability function defined in Eq. (4). Again, we restrict the search space to the three aforementioned models. While the grid search suffers from the curse of dimensionality and does not scale to large search spaces of F and G_s, limiting the number of plausible functions allows us to exhaustively search for the optimal discriminator for a given attack model and a target PUF. Additionally, the grid search is a reasonable approach, given that it can be embarrassingly parallel.

Evaluation and Analysis

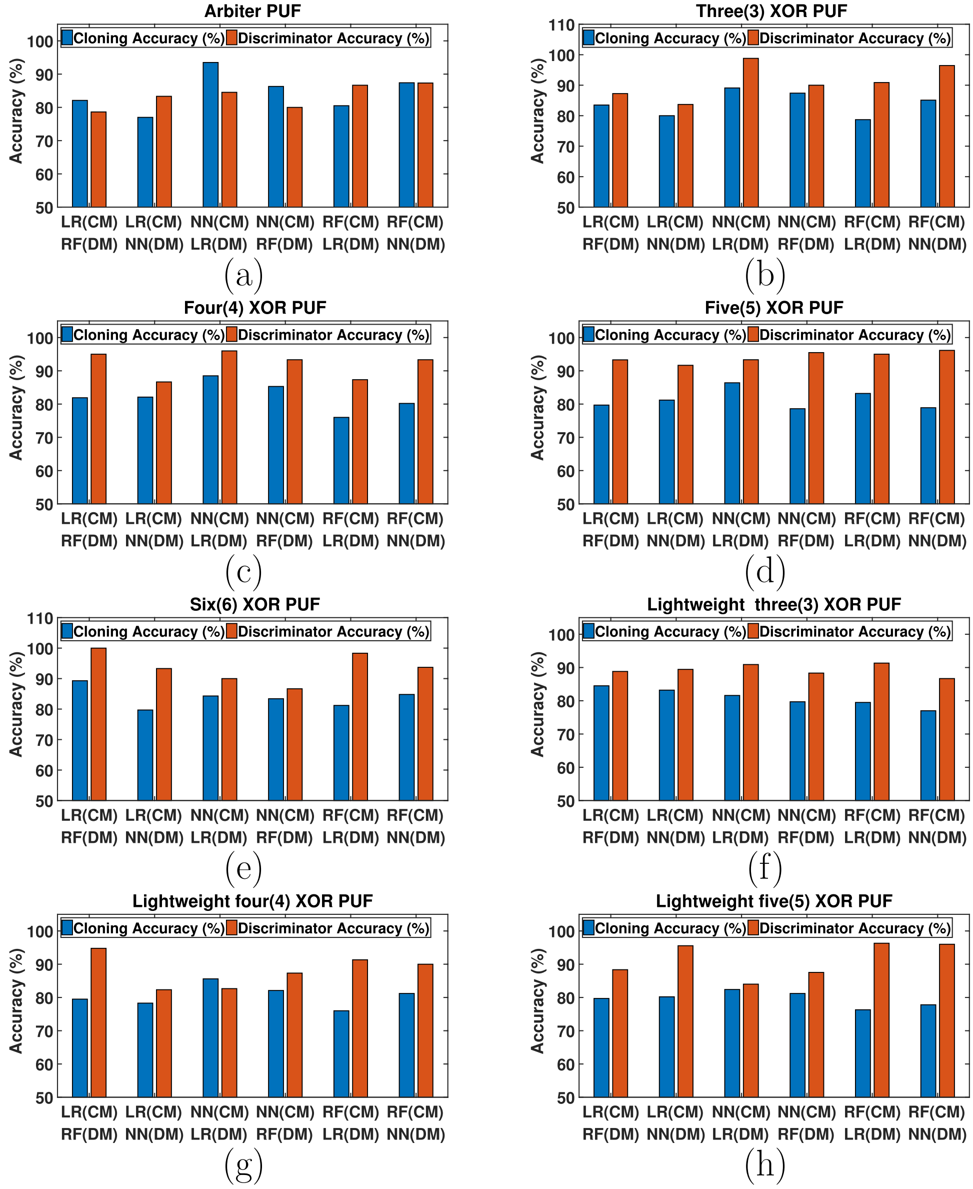
In this section, we quantitatively evaluate and analyze the performance of the three machine learning-based models proposed in Sects. Brute Force Attack on Strong PUFs and Architecture-Independent PUF Modeling. We begin with a discussion on the experimental setup and metrics. We then evaluate the proposed approaches in three different settings: (1) unencrypted authentication protocol; (2) encrypted authentication protocol; and (3) authentication using obfuscated challenges. We conclude with an evaluation of the machine learning-based countermeasure, proposed in Sect. Machine Learning-Based Countermeasure, for each of the proposed approaches.

Experimental Setup

We follow the same experimental setup by [32] and report the upper bound of the attacker's ability to successfully clone a given PUF architecture as its accuracy in a supervised setting. We report all results as the average of ten experimental runs. For evaluating under the unencrypted setting, we consider three strong PUF architectures (Arbiter, XOR, and Lightweight), while each of them contains three stages (64, 128, and 256) and the number of XOR is limited to (3, 4, and 5) for both XOR and lightweight PUFs. This gives us a total of 24 different strong PUF architectures for validating the efficacy of the proposed cloning models. For evaluating under the encrypted setting, we consider two conventional encryption techniques—the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). We use the 128-bit versions of both encryption methods. We consider two strong PUF architectures in a 64-stage Arbiter PUF and XOR PUFs, as well as two variations of the XOR PUF—3-XOR and 4-XOR PUFs to evaluate the ability of the proposed approach to generalize to more complex architectures. We present the average results of the experiments conducted on a limited CRP regime of less than 250 CRP pairs for both training and testing. Although DES is susceptible to cryptanalysis, it is a non-trivial task. 128-bit AES is resistant to brute force attacks, given that there can exist as much as 3.4×10^{38} key combinations. Such characteristics make the task of cloning an encrypted PUF a challenging problem. For evaluating under the obfuscated challenge setting, we use a simplified version of the OB-PUF proposed in [12]. We use the arbiter 64-stage PUF as the base PUF architecture. We randomly perturb the $n\%$ of the challenge and evaluate the ability of the cloning model to reconstruct and generate the cloned response.

Unencrypted PUF-Based Authentication

We evaluate the ability of the proposed approaches in the unencrypted PUF-based authentication setting. This is the commonly used setting in machine learning-based cloning attacks, such as [32] on PUF architectures. We summarize the cloning results in Table 2, from the optimization process described in Sect. Brute Force Attack on Strong PUFs. Results for each machine learning model can be seen in Fig. 5. Our approach does not require physical access or prior knowledge on the PUF architecture. The average cloning accuracy of our approach can be as high as 93.50%. It is to be noted that while [32] achieve cloning accuracy of 99.9%, they do require that the underlying architecture is known, and physical access is available. The cloning accuracy of the proposed model drops as the complexity of the PUF architecture grows, with the lightweight 5-XOR Arbiter PUF being the hardest to clone. This could arguably be attributed to the randomness introduced by the complex PUF architecture. From Table 2, we can see that, on average, a strong PUF can be cloned with a cloning error of 10.83% irrespective of its underlying architecture of the PUF. The



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5 Comparison of cloning and discriminator accuracies for cloning models under different PUF architectures. Along *X*-axis, *X*(*Y*) refers to machine learning model *X* is used for tasks *Y*-cloning model (CM) or discriminator model (DM)

Table 2 Cloning error and time for different PUF models

PUF Model	Cloning error (%)	Cloning time (min)
APUF	6.50	0.00001
3 XOR APUF	8.20	1.18083
4 XOR APUF	10.70	1.63333
5 XOR APUF	9.00	62.8010
6 XOR APUF*	10.70	240.040
LW 3 XOR APUF	12.00	0.02650
LW 4 XOR APUF	12.50	30.9667
LW 5 XOR APUF*	17.00	180.025
Average	10.83	64.5759

*Note that in the literature [30, 32], the maximum number of XORs used is 6. It is known that six XORs is sufficient to give a strong PUF

Table 3 Cloning rates and times for different PUF architectures under different encryption protocols

Approach	PUF Model	DES		AES	
		Acc.	Time(s)	Acc.	Time(s)
LR	64-Stage Aribter	46.9	1.2	48.7	1.9
	3-XOR	60.9	26.2	53.8	30.2
	4-XOR	43.8	53.9	40.6	49.9
RF	64-Stage Arbiter	51.6	0.001	54.7	0.007
	3-XOR	59.4	0.31	54.7	29.4
	4-XOR	42.2	1.8	48.4	1.3
MLP	64-Stage Arbiter	56.1	35.8	53.6	33.1
	3-XOR	51.1	70.8	52.3	46.7
	4-XOR	50.1	98.7	50.2	112.9
Standard Autoencoder	64-Stage Arbiter	58.7	43.1	57.4	48.3
	3-XOR	54.5	44.0	55.4	56.9
	4-XOR	51.9	46.7	53.6	53.1
Denoising Autoencoder	64-Stage Arbiter	65.6	45.6	63.9	47.2
	3-XOR	58.1	43.6	58.9	43.6
	4-XOR	57.3	42.6	59.2	47.6

Acc stand for Accuracy

aging of the PUF [22] affects the delay characteristic, which produces a different pattern of the responses compared to the compromised node. It can be seen that the cloning time is reasonable, particularly given the complexity and stochastic nature of the considered PUFs.

Encrypted PUF-Based Authentication

In this setting, we evaluate the cloning ability of machine learning models when the challenge is encrypted, which is the standard practice in most practical IoT systems. We summarize the cloning rates and times for different PUF architectures under different encryption protocols in Table 3. We observe that adding the encryption protocols cause significant problems to standard machine learning cloning models. For example, Arbiter PUFs are often considered by many to be strongly predictable and hence more susceptible to machine learning-based attacks. However, with the added security of an encryption protocol, the predictability of an arbiter PUF model can be considered to lower significantly. We can corroborate this in our experiments with a 64-stage arbiter PUF. It can be seen that the standard attacks do not perform well on this task, although some, such as logistic regression, have shown up to 99.9% accuracy in cases when the challenge is not encrypted. Further, the addition of even a relatively weak encryption scheme such as 128-bit DES significantly degrades the performance of machine learning models. On the other hand, the autoencoder-based approach can clone the Arbiter PUF model with significantly higher accuracy. There is a significant difference in performance between the proposed approach and the brute force models.

Obfuscated PUF-Based Authentication

We also evaluate the performance of the cloning models when the challenge is obfuscated, as postulated in obfuscated PUF architectures such as [12, 23]. We consider a simpler version of these approaches for our experiments. We use the 64-stage Arbiter PUF as the base PUF model. We randomly perturb or obfuscate the plain text challenge to an arbitrary constant. This results in an obfuscated challenge, which is then presented to the cloning model to generate a response. We present results in Fig. 6. It can be seen that traditional machine learning-based cloning models such as logistic regression (LR), random forests (RF), and neural networks (MLP) are drastically affected by increasing amounts of obfuscation. The autoencoder models, on the other hand, can maintain their performance to reasonable levels, with the denoising autoencoder performing a little better at higher obfuscation levels.

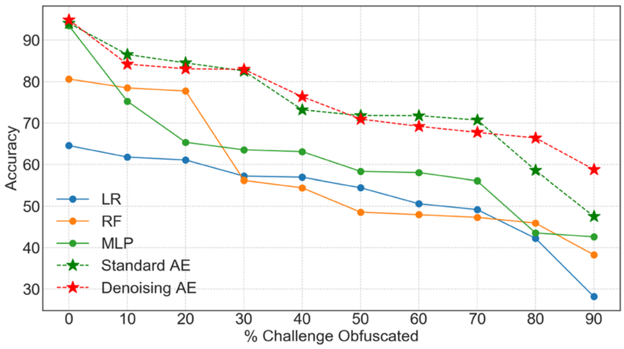


Fig. 6 Effect of challenge obfuscation on cloning performance. Accuracy is shown in comparison with varying amounts of challenge obfuscation

It should be noted that the performance is tested only on a limited evaluation set of 200 CRPs. More complicated obfuscation techniques such as those proposed in [12, 23] and less training would further degrade the performance of machine learning-based cloning attacks.

Discriminator Performance

Given the competitive performance of the machine learning models for cloning PUF architectures under different conditions, it becomes imperative that we are able to distinguish between a cloned PUF and the original PUF. We evaluate the ability of the proposed *discriminator* model (Sect. Machine Learning-Based Countermeasure) to identify a cloned PUF. We present the results in Fig. 5. It can be seen that it is possible to identify a cloned PUF with a high degree of confidence from its response to the presented challenge. We are able to identify cloned PUFs with up to 95% accuracy (Fig. 5a, e,h) for some PUF architectures such as lightweight XOR PUFs and Arbiter PUFs. Other architectures such as 4-XOR and 5-XOR PUFs are harder to clone and harder to discriminate between cloned and original PUFs.

Conclusions

In this work, we presented and evaluated three different machine learning approaches to attack PUF-based edge node authentication through cloning the underlying PUF model. To the best of our knowledge, we are the first to address the problem of encrypted and obfuscated CRPs. We showed that a priori knowledge and physical access to the PUF architecture is not necessary to clone the PUF model. Additionally, autoencoder-based pre-training allowed us to handle additional challenges such as encryption and simple obfuscation. We showed that machine learning models could be powerful enough to clone PUF models in different settings successfully. We also introduce a novel discriminator model to identify cloned and original PUFs with a high degree of confidence. Extensive experiments show that the proposed approach can generalize even with a limited number of CRPs and show significantly higher cloning accuracy than brute force machine learning models. We aim to show that the proposed approach can recover CRPs that are transmitted with complex obfuscation techniques and handle noise induced through channels and aging of PUF devices.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Aman MN, Chua KC, Sikdar B. Hardware Primitives-Based Security Protocols for the Internet of Things. In: Cryptographic Security Solutions for the Internet of Things, 2019:117–141. IGI Global
2. Aman, MN, Taneja S, Sikdar B, Chua KC, Alioto M. Token-based security for the Internet of Things with dynamic energy-quality tradeoff. IEEE Internet Things J. 2018;6(2):2843–2859.
3. Bokefode JD, Bhise AS, Satarkar PA, Modani DG. Developing a secure cloud storage system for storing IoT data by applying role based encryption. Proc Comput Sci. 2016;89:43–50.
4. Braeken A. PUF based authentication protocol for IoT. Symmetry. 2018;10(8):352.
5. Cam-Winget N, Sadeghi A, Jin Y. Can IoT be secured: Emerging challenges in connecting the unconnected. In: Proceedings of the 53rd Annual Design Automation Conference, 2016:122. ACM
6. Chatterjee U, Chakraborty RS, Mukhopadhyay D. A PUF-based secure communication protocol for IoT. ACM Trans Embed Comput Syst (TECS). 2017;16(3):67.
7. Chatterjee U, Govindan V, Sadhukhan R, Mukhopadhyay D, Chakraborty RS, Mahata D, Prabhu MM. Building PUF based authentication and key exchange protocol for IoT without explicit crps in verifier database. IEEE Transactions on Dependable and Secure Computing. 2018.
8. Coppersmith D. The data encryption standard (DES) and its strength against attacks. IBM J Res Dev. 1994;38(3):243–50.
9. Daemen J, Rijmen V. The design of Rijndael: AES-the advanced encryption standard. Berlin: Springer; 2013.
10. Dodis Y, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin C, Camenisch JL, editors. Advances in cryptology. Berlin, Heidelberg: EUROCRYPT 2004; 2004. p. 523–40.
11. Ganji F, Tajik S, Fäßler F, Seifert JP. Strong machine learning attack against PUFs with no mathematical model. Cryptology ePrint Archive, Report 2016/606 (2016). https://eprint.iacr.org/2016/606.
12. Gao Y, Li G, Ma H, Al-Sarawi SF, Kavehei O, Abbott D, Ranasinghe DC. Obfuscated challenge-response: A secure lightweight authentication mechanism for puf-based pervasive devices. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), 2016:1–6. IEEE
13. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. Ad Neural Inform Process Syst. 2014;2014:2672–80.
14. Herder C, Yu MD, Koushanfar F, Devadas S. Physical unclonable functions and applications: a tutorial. Proc IEEE. 2014;102(8):1126–41. https://doi.org/10.1109/JPROC.2014.2320516.
15. Idriss T, Idriss H, Bayoumi M. A PUF-based paradigm for IoT security. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016:700–705. IEEE
16. Ishai Y, Prabhakaran M, Sahai A, Wagner D. Private circuits II: keeping secrets in tamperable circuits. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2006:308–327. Springer
17. Islam SA, Katkoori S. High-level synthesis of key based obfuscated RTL datapaths. In: 2018 19th International Symposium on Quality Electronic Design (ISQED), 2018:407–412. https://doi.org/10.1109/ISQED.2018.8357321
18. Islam SA, Sah LK, Katkoori S. Empirical word-level analysis of arithmetic module architectures for hardware trojan susceptibility. In: 2018 Asian Hardware Oriented Security and

Trust Symposium (AsianHOST), 2018:109–114. https://doi.org/10.1109/AsianHOST.2018.8607170

19. Laguduva V, Islam SA, Aakur S, Katkoori S, Karam R. Machine learning based iot edge node security attack and countermeasures. In: 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019:670–675. IEEE
20. Maes R, Tuyls P, Verbauwhede I. Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs. In: Cryptographic hardware and embedded systems-CHES 2009, 2009:332–347. Springer
21. Mahmoud A, Rührmair U, Majzoobi M, Koushanfar F. Combined modeling and side channel attacks on strong PUFs. Cryptology ePrint Archive, Report 2013/632 (2013). https://eprint.iacr.org/2013/632.
22. Meguerdichian S, Potkonjak M. Device aging-based physically unclonable functions. In: 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC), 2011:288–289. IEEE
23. Mispan MS, Halak B, Zwolinski M. Lightweight obfuscation techniques for modeling attacks resistant PUFs. In: 2017 IEEE 2nd International Verification and Security Workshop (IVSW), 2017:19–24. https://doi.org/10.1109/IVSW.2017.8031539
24. Ostrovsky R, Scafuro A, Visconti I, Wadia A. Universally composable secure computation with (malicious) physically uncloneable functions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013:702–718. Springer
25. Pappu R, Recht B, Taylor J, Gershenfeld N. physical one-way functions. Science. 2002;297(5589):2026–30. https://doi.org/10.1126/science.1074376. http://science.sciencemag.org/content/297/5589/2026.
26. Ramnath VL, Aakur SN, Katkoori S. Latent space modeling for cloning encrypted PUF-based authentication. In: IFIP International Internet of Things Conference, 2019:142–158. Springer
27. Ray S, Bhunia S, Jin Y, Tehranipoor M. security validation in IoT space. In: 2016 IEEE 34th VLSI Test Symposium (VTS), 2016:1–1. IEEE
28. Rostami M, Majzoobi M, Koushanfar F, Wallach DS, Devadas S. Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching. IEEE Trans Emerg Top Comput. 2014;2(1):37–49. https://doi.org/10.1109/TETC.2014.2300635.
29. Rostami M, Majzoobi M, Koushanfar F, Wallach DS, Devadas S. Robust and reverse-engineering resilient PUF authentication and key-exchange by substring matching. IEEE Trans Emerg Top Comput. 2014;2(1):37–49.
30. Rührmair U. Oblivious transfer based on physical unclonable functions. In: Acquisti A, Smith SW, Sadeghi AR, editors. Trust and trustworthy computing. Berlin Heidelberg: Springer; 2010. p. 430–40.
31. Rührmair U, Holcomb DE. PUFs at a glance. In: 2014 Design, Automation Test in Europe Conference Exhibition (DATE), 2014:1–6 . https://doi.org/10.7873/DATE.2014.360
32. Rührmair U, Sehnke F, Sölter J, Dror G, Devadas S, Schmidhuber J. modeling attacks on physical unclonable functions. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS '10, 2010:237–249. ACM, New York, NY, USA. https://doi.org/10.1145/1866307.1866335.
33. Rührmair U, Xu X., Sölter J, Mahmoud A. Koushanfar F, Burleson W. Power and timing side channels for pufs and their efficient exploitation. Cryptology ePrint Archive, Report 2013/851 (2013). https://eprint.iacr.org/2013/851.
34. Sehgal A, Perelman V, Kuryla S, Schonwalder J. Management of resource constrained devices in the internet of things. IEEE Commun Mag. 2012;50(12):144–9.
35. Srivastava N. Improving neural networks with dropout. Univ Toronto. 2013;182(566):7.
36. Stergiou C, Psannis KE, Kim BG, Gupta B. Secure integration of IoT and cloud computing. Fut Gen Comput Syst. 2018;78:964–75.
37. Suh GE, Devadas S. Physical unclonable functions for device authentication and secret key generation. In: 2007 44th ACM/IEEE Design Automation Conference, 2007:9–14
38. Suo H, Wan J, Zou C, Liu J. Security in the internet of things: a review. In: 2012 international conference on computer science and electronics engineering, vol. 3, 2012:648–651. IEEE
39. Vijayakumar A, Patil VC, Prado CB, Kundu S. Machine learning resistant strong PUF: Possible or a pipe dream? In: 2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST), 2016:19–24. https://doi.org/10.1109/HST.2016.7495550
40. Wang X, Zhang J, Schooler EM, Ion M. Performance evaluation of attribute-based encryption: Toward data privacy in the IoT. In: 2014 IEEE International Conference on Communications (ICC), 2014:725–730. IEEE
41. Yang K, Forte D, Tehranipoor M. Protecting endpoint devices in IoT supply chain. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, 2015:351–356. IEEE Press
42. Ye J, Hu Y, Li X. RPUF: Physical unclonable function with randomized challenge to resist modeling attack. In: 2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST), 2016:1–6. https://doi.org/10.1109/AsianHOST.2016.7835567

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.