Latent Space Modeling for Cloning Encrypted PUF-Based Authentication

Vishalini Laguduva Ramnath, Sathyanarayanan N. Aakur, Srinivas Katkoori

Latent Space Modeling for Cloning Encrypted PUF-based Authentication

Vishalini Laguduva Ramnath[1], Sathyanarayanan N. Aakur[1 2], and Srinivas Katkoori[1]

[1] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA
[2] Department of Computer Science, Oklahoma State University, Stillwater, OK 74078, USA

Abstract. Physically Unclonable Functions (PUFs) have emerged as a lightweight, viable security protocol in the Internet of Things (IoT) framework. While there have been recent works on crypt-analysis of PUF-based models, they require physical access to the device and knowledge of the underlying architecture along with unlimited access to the challenge-response pairs in plain text without encryption. In this work, we are the first to tackle the problem of encrypted PUF-based authentication in an IoT framework. We propose a novel, generative framework based on variational autoencoders that is PUF architecture-independent and can handle encryption protocols on the transmitted CRPs. We show that the proposed framework can successfully clone three (3) different PUF architectures encrypted using two (2) different encryption protocols in DES and AES. We also show that the proposed approach outperforms a brute-force machine learning-based attack model by over 20%.

Keywords: Physically Unclonable Function · Cloning · Encryption · Latent Space Modeling.

1 Introduction

Rapid progress in computing technologies, especially space and power-efficient devices, have enabled the advent of the "*age of Internet of Things (IoT)*". The IoT ecosystem refers to the massive collection of ubiquitous and pervasive devices that have 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. Due to the somewhat limited scope of computing resources, the IoT nodes themselves do not process such information. Instead, they are used as data collection agents that transmit the collected data to more powerful edge servers for information processing. This information transmission is often done through wireless networks, which are prone to attacks and hence require robust security protocols for ensuring the integrity of the transmitted

data. Security protocols, such as node authentication, have to be sufficiently lightweight, yet highly secure to ensure that these protocols can be performed on power-constrained IoT nodes.

Authentication protocols can vary from being very simple, such as physical storage of a secret key on silicon devices, to complex cryptography-based algorithms that can require significant power and area requirements on the device. It has, however, been shown that the most straightforward authentication that of physically storing the secret key on the node device can be bypassed through physical and side-channel attacks [19]. Recovering the secret key through such physical attacks can compromise the entire IoT network and hence compromise the integrity and anonymity of the transmitted data. With the need for lightweight, yet secure authentication protocols increasing with the rapidly growing use of IoT nodes, physically unclonable functions (PUFs) [15] have emerged as a viable option for IoT node security [3].

Physically unclonable functions, or PUFs for short, are physical random functions that exploit the unique physical variations that can occur during the manufacturing process to create a "digital signature" for the device. This digital signature is dependent on the uniqueness of the device's physical microstructure. Since the physical structure is dependent on random physical aspects introduced in the manufacturing process, it is not feasible to clone or duplicate the exact physical structure of the device. In addition to their unclonable nature, the PUF-based authentication protocol extends the single key-based authentication to using the challenge-response pair (CRP) based authentication. CRPs are characterized by the application of an external stimulus (the challenge) to the PUF and receiving an unpredictable, but a repeatable response. Each challenge-response pair is unique to a PUF and hence can be used to verify the identity of a given device. These characteristics of PUFs have made them highly conducive for their widespread use in cryptography applications such as for identification and authentication [21], digital rights management [14], bit-commitment protocol [21], and secure multi-party communication [23], to name a few.

The use of PUFs as the basis for IoT node authentication has gained momentum in recent times [1, 2, 6–8]. PUF-based IoT node authentication has two fundamental processes - (1) an enrollment phase and (2) an authentication phase. The enrollment phase involves the building of a database of CRPs between the authenticating edge server and a data node. This is typically done before the data node is "*deployed*" into the wild and involves the collection of a large number of CRPs to ensure that the "*replay*" attack is prevented. The authentication phase is the application of an authentication protocol, typically the use of the challenge to the PUF and verification of the corresponding response. Figure 1 illustrates these processes in a typical IoT framework. While proven to be effective, the enrollment phase allows for a malicious attacker to eavesdrop and construct a complementary database of CRPs that they can then use to emulate, or rather *clone* the PUF and thus compromise the integrity of the data node. There have been advances that have now been proposed that the extraction of

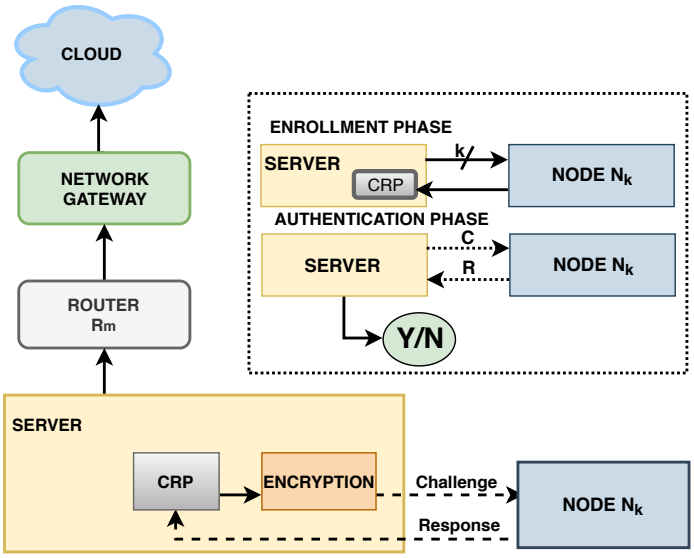

Fig. 1. A typical IoT architecture is illustrated. The inner figure shows the enrollment phase and the authentication phase of a PUF-based IoT node authentication scheme.

CRPs is then destroyed, i.e., fuse the extraction wires, thereby eradicating the possibility of cloning via this method.

The use of PUFs for IoT node security holds some **security assumptions** as defined in [20]. Many of the proposed IoT networks using PUF authentication in existing literature [1, 6, 7] make the following underlying assumptions: (1) a malicious agent can have access to the collection of CRPs obtained in the enrollment phase through malicious software attacks, although secret keys are not explicitly known, (2) the challenge-response characteristics of the PUF within the data IoT node is an implicit property and is not accessible to an adversary, (3) the malicious agent has unrestricted to the communication channel and (4) the modeling of PUF characteristics, either physical, mathematical or otherwise is a complex task. Given that current designs of IoT nodes ensure that they are tamper-proof [18, 33], physical access to the PUF such as micro-probing is somewhat tricky. Hence, PUF-based authentication has proven to be an effective strategy for securing data nodes in an IoT framework.

While highly sophisticated and secure, PUF models are susceptible to cloning using complex mathematical models and cryptanalysis. Common modes of cryptanalyses include side-channel attacks [19,25], machine-learning (ML) attacks [24] and software attacks, for example, worms and viruses [28]. Machine learning models are particularly adept at cloning PUF models. The pioneering work of Rührmair et al [24], have shown great success in cloning PUFs, gaining cloning accuracy of up to 99.99%. Most approaches to PUF cloning make two criti-

cal assumptions: (1)the underlying architecture must be known *a priori*, either through invasive physical intrusions or explicit architecture knowledge and (2) the challenge and response are sent through the communication channel in plain text i.e., no encryption masks the direct relationship between challenge and response characteristics of the PUF within the data node. Given that most, if not all, communication in the wireless channel is encrypted through some hashing or encryption technique, and most IoT data nodes are tamper-proof, these are very strong assumptions to make, especially in the context of node security in an IoT framework.

In this work, we aim to address these challenges and propose an *architecture independent* modeling approach based on machine learning that does not require any prior knowledge of the underlying PUF architecture. Additionally, we do not assume that the challenge-response authentication is done via clear text transmission, as is the case with existing approaches in the literature. This does, however, come with an additional set of challenges that need to be addressed for successful cloning of the PUF-based authentication. Namely, the challenges are as follows: (1) the encryption protocols mask the relationship between external challenge and the corresponding response, (2) most encryption protocols are not easily broken and hence require us to uncover the secret key, which might not be even possible if the challenges are encrypted using a one-way hash function and (3) lack of physical access to the data node does not give us any auxiliary data such as the PUF architecture type and/or other PUF characteristics.

We aim to overcome these challenges by learning an auto-generative model which helps us to learn a discriminative latent space. This latent space modeling allows us to bypass the need to correlate the input challenge and the corresponding response. This is achieved through the use of a variational autoencoder (VAE). A variational autoencoder (VAE) consists of two parts, an encoder and a decoder. We decrease the dimensionality of the input challenge into a smaller dimensional subspace called the latent space. We then reconstruct the original input using a decoder model from this latent representation. Hence, the latent space forms a bottleneck, forcing the model to effectively compress the input data to a more discriminative representation for easier PUF response modeling. However, in addition to the traditional decoder, which attempts to regenerate the input challenge, we also introduce a decryption decoder head. The decryption head attempts to decrypt the original challenge from the encrypted version without the need for knowing the secret key. This allows us to ensure that the bottleneck layer, or the latent space, to be influenced by both the discriminative nature of the compressed representation as well as the original, plain text challenge.

In short, our paper makes the following novel contributions:

- we propose a machine learning-based cloning model on PUF architectures that do not require any prior knowledge and physical access to the IoT node,
- we show that the proposed approach can successfully clone the PUF model even if the challenge-response pair is encrypted,

- we show that the use of a generative model such as a variational autoencoder can help learn a discriminative latent space that is robust to noise, encryption, and masking which are common traits of many cryptography models used for data encryption, and
- we show that generative modeling can potentially lead to more effective probing of the PUF models to create or *recreate* the PUF's CRP database without explicit access to the server.

To the best of authors' knowledge, this is the first such framework to evaluate the case of PUF-based IoT node authentication with encryption techniques while not requiring any prior knowledge of the PUF architecture. We show that the proposed approach can successfully clone three (3) common PUF architectures encrypted using two (2) common encryption protocols. Combined, they form some of the more common IoT node authentication protocols proposed in the existing literature.

The rest of the paper is laid out as follows. We give a brief introduction to physically unclonable functions (PUFs), their use in IoT node security and the associated encryption protocols in Section 2. We introduce the proposed latent space modeling using a variational autoencoder and the training strategy for cloning an encrypted PUF protocol in Section 3. We present a baseline approach for cloning an encrypted PUF protocol by brute-force machine learning models in Section 4.1 following the experimental evaluation of the proposed approach in Section 4.2. Finally, in Section 5, we conclude with a discussion on the feasibility of the proposed approach.

2 Background and Related Work

In this section, we introduce the necessary terms and background knowledge that are relevant to the proposed approach. We begin with an introduction to physically unclonable functions and their application in IoT node security. We then review existing work on cloning or attack models on PUF models. We conclude with a short review of commonly used encryption protocols.

2.1 Physically Unclonable Functions

Physically Unclonable Functions [14, 15], or physical random functions, are an embodied version of physical functions that maps an external stimulus (the challenge) to a random, but a repeatable response. The physical function is characterized by the inherent randomness introduced during the manufacturing process and is nearly impossible to replicate given a polynomial amount of resources. A PUF model's characteristics are best expressed through the collection of challenge-response pairs (CRPs) and hence form the basis of most, if not all, PUF-based security protocols. PUFs can be categorized into two types based on the number of valid CRPs, namely weak PUFs and strong PUFs [26]. A PUF is said to be a *weak* PUF if it has a fixed, small set of CRPs that are valid and are

assumed to be access restricted. *Strong* PUFs, on the other hand, leverage large amounts of the inherent unpredictability and hence possess a large number of CRPs. They are also considered to have an unprotected physical interface and are more commonly used in security applications. We refer the reader to [26] for an extensive review of weak and strong PUF models.

There have been numerous PUF models introduced and evaluated over the years. Broadly, they can be divided into two major groups - the time-delay based models and the memory-based models. *Time delay-based models* include ring oscillator PUFs and Arbiter PUFs or APUF and its variations such as feed-forward arbiter PUFs. Such PUF models can generate real-time, chip-specific signatures without the need for expensive memory for key storage and thus, have been particularly conducive to device authentication, intellectual property, and data privacy preservation to name a few. *Memory-based PUF* models, on the other hand, exploit the variations between matched silicon devices of memory elements to characterize the inherent random function. Some common bistable memory elements that are exploited for the PUF functions are SRAM, latches, and flip-flops. Again, we refer the reader to [16] for a more detailed review of PUF architectures, which is beyond the scope of this paper.

2.2 PUFs for IoT Node Security

The use of PUFs for IoT node security [1, 2, 6–8, 17] has gained momentum in recent days. Such approaches can be classified into two major categories - PUF-based authentication and PUF-based key generation for cryptography-based approaches [30]. In PUF-based device authentication, the nature of strong PUF models to possess a large number of CRPs is exploited to build a robust authentication protocol. A trusted party, the authentication server, randomly applies a set of external stimuli or challenges to create a database of valid CRPs for authentication. This process is called the enrollment phase. Every time there is a need for authenticating the node of data transmission within the IoT framework, the server authenticates the node with a random challenge from the database of CRPs. This process is called the authentication phase. Figure 1 illustrates both these processes in a typical IoT framework. The other approach consists of using the PUF response to generate cryptographic keys. The keys are typically generated by hashing the PUF's response to a given challenge, which is processed through an error-correcting circuit.

2.3 Cloning Attacks on PUF Models

The widespread introduction of PUF models into IoT node authentication has seen an increase in approaches that attempt to test their effectiveness through attacking or *cloning* the PUF model. Cloning a PUF model typically involves the fitting of a complex mathematical function to capture the correlation between the input challenge and the corresponding PUF response. There have been several approaches, including leveraging machine learning models and physical modeling. Perhaps the most influential approach was introduced by Rührmair *et*

al. [24], who proposed a machine learning-based modeling of strong PUF models using a predictive approach. The authors were able to clone the functionality of the underlying PUF given the PUF model by evaluating model parameters using LR with RProp and ES. While highly successful, they make the assumptions outlined in Section 1 and as such cannot be widely applied to practical IoT node cloning using PUF models. The other type of approach [4,11] involves physical access to the PUF model beyond just knowledge about PUF architecture and model. They typically involve the use of machine learning approaches to model the PUF response by exploiting the physical characteristics obtained through side-channel approaches. Recently efforts have shifted to a combined ML and side-channel (timing and power) to present an improved hybrid attack surface [19,25]. A mathematical model-free ML attack using PAC (Probably Approximately Correct) learning framework has been proposed in [12]. The authors presented that an influential bit, if present in stable PUF response, can predict the future response corresponding to a challenge with low probability.

2.4 Encryption Protocols for IoT Node Authentication

With the use of CRPs for IoT node authentication, the need for encryption protocols has risen due to the need for added security from eavesdropping protocols. The use of encryption protocols in IoT node communication and authentication has seen staggering rise [5, 27, 29, 31, 32]. In summary, the encryption protocols used are the Data Encryption Standard (DES) [9] and the Advanced Encryption Standard (AES) [10]. While there has been successful cryptanalysis of DES, it still takes an extraordinary amount of compute and access to data to achieve it, whereas there has not been a successful attack on the 128-bit AES encryption protocol. While encryption protocols have been used extensively in IoT node communication, it requires some semblance of computation to get working. Hence, there have been other protocols proposed to overcome such computation power such as obfuscated CRPs [13] and substring matching [22], 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.

3 Learning a Latent Subspace for Encrypted CRPs

In this section, we introduce the proposed approach for learning a discriminative latent subspace that can be used for machine learning-based cryptanalysis of the security protocols in a typical IoT ecosystem. We begin with a brief introduction to *variational autoencoders*, which form the backbone of the proposed approach. We then introduce the proposed approach with a multi-headed decoder, which helps learn a more robust subspace for better modeling of the encryption protocols. Finally, we expand on the strategy employed in the optimization process for end-to-end training of the proposed network.

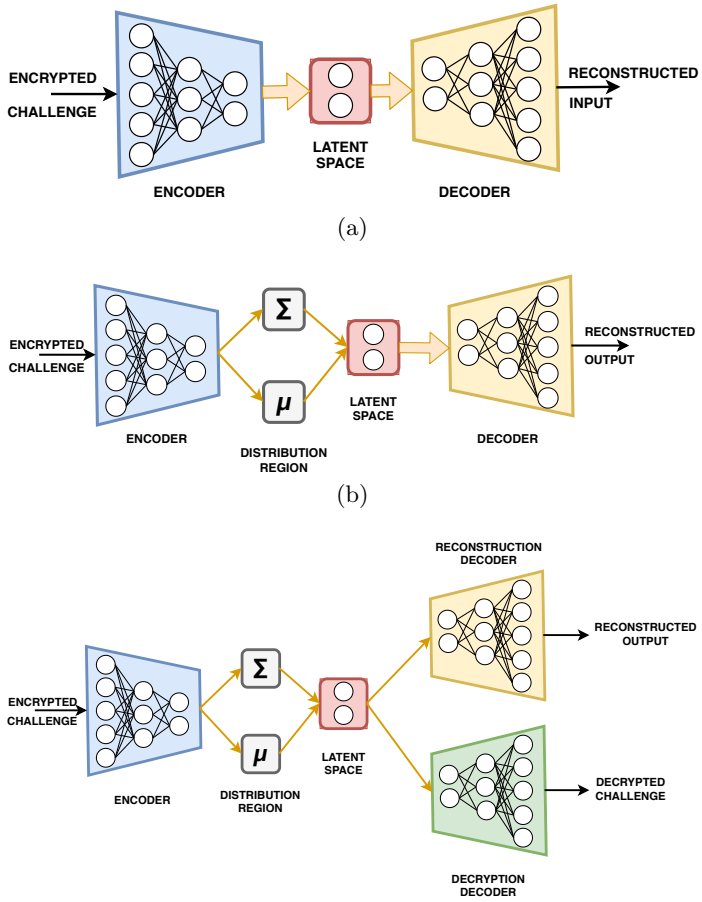

Fig. 2. We illustrate the architecture of (a) typical autoencoder, (b) a typical variational autoencoder and (c) the proposed approach with multi-headed decoders.

3.1 Variational Autoencoders

Encryption techniques such as AES and DES, to name a few, secure the transmitted data by injecting noise into the data through various techniques including, but not limited to hashing and block cipher. By doing so, the actual data within the transmitted information is hidden from prying influences. Hence, any attempt to break the security of the encryption must either (1) know the encryption techniques and the hidden cipher to recover the original data, or (2) model the underlying data distribution effectively to learn a model for manipulating the information stream. While there have been existing work in crypt-analysis for the former approach, the latter has not been explored extensively. Modeling the internal structure of the data distribution offers three significant advantages:

(1) knowing the underlying distribution allows us to reduce the dimensionality of the data by ignoring the noise in the transmission, (2) allows for the possibility of learning a generative model that clones the source of the data distribution, which in our case is the PUF within the IoT data node, and (3) learning a generative model allows the attacker to probe the PUF with genuine, or rather, valid challenges to further extract the PUF characteristics. To achieve the above, we employ the use of an unsupervised neural network called autoencoders, or more specifically, *variational autoencoders*.

An *autoencoder* is an artificial neural network trained in an unsupervised manner. The major objective of the autoencoder network is to compress the input data into an encoded representation and, more importantly, *reconstruct* the original input from the compressed encoding. The autoencoder typically consists of two networks working in tandem - an *encoder* and a *decoder*. The encoder network compresses the input into a lower-dimensional representation, called the *latent space*, by learning to ignore the noise and modeling the underlying data distribution. This latent space is represented by the *bottleneck layer* of the network. The decoder network, on the other hand, aims to reconstruct a representation that is as close as possible to the original input from the bottleneck layer. This process is represented in Figure 2(a), where it can be seen that the input to the encoder and reconstructed output from the decoder have the same dimensions whereas the latent space or bottleneck layer has a lower dimensionality. The training objective for an autoencoder network is to minimize the reconstruction loss, which is typically an $L2$ loss or binary cross-entropy.

While incredibly useful in learning a compressed representation of a (potentially) noisy input data, there is no way to restrict, or rather, *predict* the latent space representation of a given input in a deterministic manner. This poses two critical concerns. First, while very useful for compression, the latent space learned in a traditional autoencoder is scattered. This leads to better reconstructions of the input image but is not conducive to *generate* new samples that match the valid distribution. Second, a deterministic latent space allows for better probing of the PUF model through generating legitimate challenges. It also allows us to model the PUF characteristics in a model agnostic manner. To overcome these limitations, we employ the use of a *variational autoencoder*. A modification on the traditional autoencoder network paradigm, a variational autoencoder aims to restrict the latent space into a more deterministic manner by introducing an additional optimization constraint. Figure 2(b) illustrates the typical architecture of a variational autoencoder. As can be seen, the bottleneck layer is not passed through to the decoder network directly. Rather, it is used to generate a normal distribution $N(\mu, \sigma)$ (i.e. mean μ and standard deviation σ). The latent space is then sampled from this distribution to ensure that the bottleneck layer follows a given set of distribution and hence is deterministic. The training objective then becomes the reconstruction loss and the KL divergence loss to ensure that the distribution follows the standard normal distribution $N(0, 1)$. This additional loss ensures that the parameters μ and σ do not regress such that the latent space of the encoder network is preserved. The objective function

is given by

$$\mathcal{L}(\theta, \phi, X) = E_{z \sim q_\phi(Z|X)}(logP_\theta(X|Z)) - \mathcal{D}_{KL}(q_\phi(Z|X)||p_\theta(Z)) \quad (1)$$

where X is the input to be modelled (the encrypted challenge in our case), Z is the hidden variables (the latent space) from which to generate new challenges, $p_\theta(X|Z)$ is the generative process done by the decoder and $q_\phi(Z|X)$ represents the encoding process. θ and ϕ represent the parameters of the decoding and encoding processes, respectively.

3.2 Multi-headed Decoding for Robust Latent Subspace Modeling

The use of a variational autoencoder helps in providing a deterministic latent space by forcing the encoder representations to follow a normal distribution. Given that the only task of the encoder is to learn representations that can be reconstructed, there can be a tendency to overfit to the sample distribution due to the single-task learning paradigm. To overcome this inhibition, we propose the use of a multi-headed decoder network to introduce a form of multi-task learning. This provides a form of inductive transfer and allows us to form better representations for modeling the PUF characteristics. In addition to the traditional reconstruction head, we introduce a second decoder which acts as a brute-force decrypting mechanism. We assume that a minimal amount of CRPs is available to the attacker in both plain-text and encrypted forms. Given the multitude of possible eavesdropping mechanisms, this is not an unreasonable assumption. The proposed architecture is shown in Figure 2(c), where it can be seen that a joint representation, learning by the encoder, is used as the latent space for both reconstructing the original challenge as well as the decrypted challenge. This allows the model to learn a latent space representation that captures the inherent structure of a valid CRP while learning to ignore the noise induced by the encryption protocols. In Section 4.2, we can see that the use of the second decoder network as a brute-force decryption method offers better modeling of the underlying PUF architecture.

Formally, the objective of the proposed network differs from the traditional variational autoencoder (Equation 1). First, there is another generative process to uncover the plain-text challenge represented by $d_\psi(\widetilde{X}|Z)$, where $\widetilde{X}$ represents the plain-text challenge. Second, the generation of the decrypted challenge must also be dependent on the encoded representation Z. This results in the updated objective function given by

$$\begin{aligned}\mathcal{L}(\theta, \phi, \psi, X, \widetilde{X}) = E_{z \sim q_\phi(Z|X)}(logP_\theta(X|Z) + logP_\theta(\widetilde{X}|Z)) \\ - \mathcal{D}_{KL}(q_\phi(Z|X)||p_\theta(Z))\end{aligned} \quad (2)$$

where $\widetilde{X}$ is the clear text challenge, X is the input to be modelled (the encrypted challenge in our case), Z is the hidden variables (the latent space) from which to generate new challenges, $p_\theta(X|Z)$ is the auto-generative process done by the first decoder , $d_\theta(\widetilde{X}|Z)$ is the decrypted generative process done by the second

decoder and $q_\phi(Z|X)$ represents the encoding process. θ, ψ and ϕ represent the parameters of the two decoding processes and the lone encoding process, respectively.

The addition of the second decoder network introduces the notion of multi-task learning (MTT). The use of multi-task learning is crucial in many aspects, especially considering that the number of CRPs available are often very low, ranging from the low hundreds to a thousand. Since the encoder network is shared among the two decoders, this reduces the possibility of the network to overfit to the training set of the CRPs and helps generalize to unknown CRPs. In addition to preventing overfitting, the hard parameter sharing paradigm offers other benefits such as attention focusing, implicit data augmentation, reducing representation bias, and regularization, to name a few.

3.3 Implementation Details and Training Strategy

Since the proposed architecture has a complex structure, we detail the implementation details and the training strategy for the approach here. The encoder consists of four (4) densely connected layers, with each layer interspersed with a dropout layer. Each dropout layer has a dropout probability of 50%. We reduce the dimensionality of the input by $0.5\times$ at each fully connected (dense) layer. This follows the standard protocol in autoencoders to induce the bottleneck at the end of the encoding network. Each of the two decoders (reconstruction and decryption) consist of two fully connected layers that increase the dimensionality back to the original dimension and decrypted challenge dimensions, respectively. We also have a series of two (2) fully connected layers that take the latent space as input and produces the PUF response as output. This is the only part of the network that is trained in a supervised manner, i.e., using labels and target dimensions. The encoder and two decoders are trained in an unsupervised manner.

Since the training data is limited, most neural networks tend to overfit to the smaller amounts of data and do not generalize well to the other, unobserved challenge-response pairs. To overcome this, we propose the following training regimen. For ten epochs, we first train the network end-to-end only with the reconstruction decoder as active i.e., it is trained first as a traditional variational autoencoder. For the next ten epochs, we then train the decryption decoder for ten epochs while freezing the weights of the reconstruction decoder. This represents the unsupervised training portion of the proposed training regimen. We then begin the supervised training process. In this part of the training, we freeze the layers of the decoding structures and take the latent space produced by the encoder network and feed it to a series of fully connected layers and model the PUF response to the input challenge. The neural network's target is the PUF response. We train for a total of 100 epochs, with the unsupervised and supervised portions interspersed together.

4 Experimental Evaluation

In this section, we present the experimental evaluation of the proposed approach. We begin with a description of a baseline approach against which we compare the proposed approach. We then continue with the presentation of the quantitative metrics from the experimental evaluation. We then conclude with a discussion on the qualitative aspects of the proposed approach.

4.1 Baseline Approach: A Brute Force Attack on Encrypted PUFs

PUF Model	Encryption	Approach	Accuracy (%)	Cloning Time
64-Stage Aribter	DES	LR *(Brute)*	46.9	1.2 sec
		RF *(Brute)*	51.6	0.001 sec
		MLP *(Brute)*	56.1	35.8 sec
		Ours *(no decrypt)*	69.4	84.7 sec
		Ours *(no reconstr.)*	67.8	45.3 sec
		Ours *(full)*	**75.6**	98.6 sec
	AES	LR *(Brute)*	48.7	1.9 sec
		RF *(Brute)*	54.7	0.005 sec
		MLP *(Brute)*	53.6	33.1 sec
		Ours *(no decrypt)*	68.2	83.3 sec
		Ours *(no reconstr.)*	65.2	48.6 sec
		Ours *(full)*	**73.9**	93.2 sec

Table 1. ML Model cloning accuracy and the time required for cloning a 64-Stage Arbiter PUF encrypted with 128-bit DES and AES algorithms.

Given the one-to-one nature of the challenge-response mappings, it could be argued that a simple mathematical model, such as any of those used in various machine learning approaches, could be a viable alternative for cloning an encrypted PUF architecture. To this end, we train and evaluate two (2) machine learning-based models and one (1) neural network-based model. The two machine learning-based models that we trained were logistic regression (LR) and random forest (RF). We chose logistic regression as a baseline approach due to the fact the pioneering work of Rührmair *et al* [24] successfully used the method to clone various PUF architectures. While successful for cloning plain-text challenge-response characteristics of PUF architectures, we evaluate the ability of logistic regression-based approaches on the encrypted CRP setting. We chose the random forest algorithm as another baseline approach due to its tendency to reduce the overfitting nature of decision trees. Given the limited training data and the inherent non-linear nature of the data distribution, the ensemble of decision trees generated by the random forest algorithm provides a strong baseline. As a final baseline, we use a neural network that is similar to the proposed approach. Instead of pretraining the feature extraction using

the proposed approach of variational autoencoders with multiple decoders, we use a standard multilayer perceptron (MLP) network. It consists of an input layer, followed by two (2) hidden layers (analogous to the encoder) that reduce the dimensionality of the input and two hidden layers that increase the dimensionality (comparable to the decoder) followed by the output layer that models the PUF's characteristic response. We choose this MLP architecture to emphasize the importance of the proposed approach, which enhances the ability of the neural network to learn discriminative features.

PUF Model	Encryption	Approach	Accuracy (%)	Cloning Time (s)
3-XOR PUF	DES	LR *(Brute)*	60.9	26.2 sec
		RF *(Brute)*	59.4	0.31 sec
		MLP *(Brute)*	51.1	70.8 sec
		Ours *(no decrypt)*	61.5	87.0 sec
		Ours *(no reconstr.)*	62.4	51.9 sec
		Ours *(full)*	**64.8**	83.6 sec
	AES	LR *(Brute)*	53.8	30.2 sec
		RF *(Brute)*	54.7	0.29 sec
		MLP *(Brute)*	52.3	46.7 sec
		Ours *(no decrypt)*	65.4	76.9 sec
		Ours *(no reconstr.)*	62.1	42.6 sec
		Ours *(full)*	**68.9**	73.6 sec

Table 2. ML Model cloning accuracy and the time required for cloning a 3 XOR PUF encrypted with 128-bit DES and AES algorithms.

4.2 Quantitative Evaluation

Following experimental setup by [24], we report the upper bound of the attacker's ability to successfully clone a given PUF architecture as its accuracy in a supervised setting. To evaluate the ability of the proposed approach to cloning a given PUF successfully, we consider two strong PUF architectures in a 64-stage Arbiter PUF and XOR PUFs. We consider two (2) 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 also consider two (2) conventional encryption techniques - the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). We use the 128-bit versions of both encryption methods. This gives us a total of six (6) different strong PUF architectures for validating the efficacy of the proposed method. We present the average results of the experiments conducted over ten (10) trials and on a limited CRP regime of less than 250 CRP pairs for both training and testing. Although DES is susceptible to crypt-analysis, 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.

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. We present these results in Table 1 It can be seen that the brute force 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. Additionally, 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, our proposed 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, even considering the similarly structured MLP approach, which differs from the proposed approach only in that the unsupervised training regime is not conducted on it during the training phase.

XOR PUFs offer a significantly higher challenge to the cloning problem compare to the arbiter PUFs. As the number of stages grows, the predictability of the PUF architecture reduces. This makes the XOR PUF more suitable for nodes requiring additional security. The addition of encryption protocols such as DES and AES makes it even more challenging to clone a given PUF architecture. We summarize the results of our experiments with 3 XOR and 4 XOR PUFs in Table 1 and Table 2 respectively. We can see that as the number of stages increases, the ability of the machine learning models to clone the PUF device reduces drastically. It is important to note that in the literature [24] [23], the maximum number of XORs used is 6. We experiment up to 4 XOR PUFs in this paper. We also find that in XOR PUFs, the role of the decryption head is significantly higher than in arbiter PUFs. This could arguably be attributed to the fact that each of the XOR nodes in the PUF architecture adds to the non-linearity of the PUF characteristics, thereby reducing its predictability and hence providing added security against machine learning attacks.

We also perform **ablation studies** to evaluate the impact of each of the components that are part of the proposed framework: (1) decryption decoder head, (2) the reconstruction decoder head and (3) the use of variational autoencoders for unsupervised pretraining of the encoder network. It can be seen from each of Table 1, Table 2 and Table 3 that each decoder head adds significant improvements over the base model. The performance improvement due to the addition of the decryption decoder can be as high as 5.7% (Table 1). Additionally, the mere use of neural networks is not sufficient to guarantee successful cloning of a PUF architecture, especially with the employment of encryption schemes. We can see that the use of the objective functions described in Equation 1 and Equation 2 and the unsupervised pre-training regimen described in Section 3.3 add significant performance gains over the vanilla neural networks (MLP). We observe as much as 20.6% improvement in cloning accuracy for arbiter PUFs.

PUF Model	Encryption	Approach	Accuracy (%)	Cloning Time (s)
4-XOR PUF	DES	LR *(Brute)*	43.75	53.9 sec
		RF *(Brute)*	42.2	1.8 sec
		MLP *(Brute)*	50.1	98.7 sec
		Ours *(no decrypt)*	55.5	86.7 sec
		Ours *(no reconstr.)*	57.9	65.7 sec
		Ours *(full)*	**60.3**	82.6 sec
	AES	LR *(Brute)*	40.62	49.9 sec
		RF *(Brute)*	48.43	1.3 sec
		MLP *(Brute)*	50.23	112.9 sec
		Ours *(no decrypt)*	57.6	93.1 sec
		Ours *(no reconstr.)*	59.7	81.4 sec
		Ours *(full)*	**63.9**	97.6 sec

Table 3. ML Model cloning accuracy and the time required for cloning a 4 XOR Arbiter PUF encrypted with 128-bit DES and AES algorithms.

5 Conclusion and Future Work

In this work, we introduce and evaluate a novel, generative framework using based on a variational autoencoder to clone PUF models over an encrypted communication channel, which is a realistic scenario. We are, to the best of our knowledge, the first to address the problem of encrypted CRPs. We show that the use of the unsupervised pretraining using the proposed framework and training regimen allows us to successfully clone a given PUF model without the need for knowing the secret key used in the encryption protocol. Extensive experiments show that the proposed approach can generalize even with a limited number of CRPs and can show significantly higher cloning accuracy compared to brute force machine learning models. In the future, we aim to show that the proposed approach can generate or recover CRPs that are transmitted with obfuscation and noisy channels.

References

1. Aman, M.N., Chua, K.C., Sikdar, B.: Hardware Primitives-Based Security Protocols for the Internet of Things. In: Cryptographic Security Solutions for the Internet of Things, pp. 117–141. IGI Global (2019)
2. Aman, M.N., Taneja, S., Sikdar, B., Chua, K.C., Alioto, M.: Token-based security for the Internet of Things with dynamic energy-quality tradeoff. IEEE Internet of Things Journal (2018)
3. Aman, M.N., Chua, K.C., Sikdar, B.: Position paper: Physical unclonable functions for iot security. In: Proceedings of the 2nd ACM international workshop on IoT privacy, trust, and security. pp. 10–13. ACM (2016)
4. Becker, G.T., Kumar, R., et al.: Active and passive side-channel attacks on delay based puf designs. IACR Cryptology ePrint Archive **2014**, 287 (2014)

5. Bokefode, J.D., Bhise, A.S., Satarkar, P.A., Modani, D.G.: Developing a secure cloud storage system for storing iot data by applying role based encryption. Procedia Computer Science **89**, 43–50 (2016)
6. Braeken, A.: PUF Based Authentication Protocol for IoT. Symmetry **10**(8), 352 (2018)
7. Chatterjee, U., Chakraborty, R.S., Mukhopadhyay, D.: A PUF-based secure communication protocol for IoT. ACM Transactions on Embedded Computing Systems (TECS) **16**(3), 67 (2017)
8. 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 Transactions on Dependable and Secure Computing (2018)
9. Coppersmith, D.: The data encryption standard (des) and its strength against attacks. IBM journal of research and development **38**(3), 243–250 (1994)
10. Daemen, J., Rijmen, V.: The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media (2013)
11. Delvaux, J., Verbauwhede, I.: Side channel modeling attacks on 65nm arbiter pufs exploiting cmos device noise. In: 2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). pp. 137–142. IEEE (2013)
12. Ganji, F., Tajik, S., Fler, F., Seifert, J.P.: Strong machine learning attack against pufs with no mathematical model. Cryptology ePrint Archive, Report 2016/606 (2016), `https://eprint.iacr.org/2016/606`
13. Gao, Y., Li, G., Ma, H., Al-Sarawi, S.F., Kavehei, O., Abbott, D., Ranasinghe, D.C.: 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). pp. 1–6. IEEE (2016)
14. Gassend, B., Clarke, D., van Dijk, M., Devadas, S.: Controlled Physical Random Functions. In: Proceedings of the 18th Annual Computer Security Applications Conference. pp. 149–. ACSAC '02, IEEE Computer Society, Washington, DC, USA (2002), `http://dl.acm.org/citation.cfm?id=784592.784802`
15. Gassend, B., Clarke, D., Van Dijk, M., Devadas, S.: Silicon physical random functions. In: Proceedings of the 9th ACM conference on Computer and communications security. pp. 148–160. ACM (2002)
16. Herder, C., Yu, M.D., Koushanfar, F., Devadas, S.: Physical Unclonable Functions and Applications: A Tutorial. Proceedings of the IEEE **102**(8), 1126–1141 (Aug 2014). https://doi.org/10.1109/JPROC.2014.2320516
17. 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). pp. 700–705. IEEE (2016)
18. 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. pp. 308–327. Springer (2006)
19. 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`
20. 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. pp. 702–718. Springer (2013)

21. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical One-Way Functions. Science **297**(5589), 2026–2030 (2002). https://doi.org/10.1126/science.1074376, http://science.sciencemag.org/content/297/5589/2026
22. Rostami, M., Majzoobi, M., Koushanfar, F., Wallach, D.S., Devadas, S.: Robust and reverse-engineering resilient puf authentication and key-exchange by substring matching. IEEE Transactions on Emerging Topics in Computing **2**(1), 37–49 (2014)
23. Rührmair, U.: Oblivious transfer based on physical unclonable functions. In: Acquisti, A., Smith, S.W., Sadeghi, A.R. (eds.) Trust and Trustworthy Computing. pp. 430–440. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
24. 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. pp. 237–249. CCS '10, ACM, New York, NY, USA (2010). https://doi.org/10.1145/1866307.1866335, http://doi.acm.org/10.1145/1866307.1866335
25. 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
26. Rührmair, U., Holcomb, D.E.: Pufs at a glance. In: Proceedings of the conference on Design, Automation & Test in Europe. p. 347. European Design and Automation Association (2014)
27. Sehgal, A., Perelman, V., Kuryla, S., Schonwalder, J.: Management of resource constrained devices in the internet of things. IEEE Communications Magazine **50**(12), 144–149 (2012)
28. Stallings, W., Brown, L., Bauer, M.D., Bhattacharjee, A.K.: Computer security: principles and practice. Pearson Education (2012)
29. Stergiou, C., Psannis, K.E., Kim, B.G., Gupta, B.: Secure integration of iot and cloud computing. Future Generation Computer Systems **78**, 964–975 (2018)
30. Suh, G.E., Devadas, S.: Physical Unclonable Functions for Device Authentication and Secret Key Generation. In: 2007 44th ACM/IEEE Design Automation Conference. pp. 9–14 (June 2007)
31. 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, pp. 648–651. IEEE (2012)
32. Wang, X., Zhang, J., Schooler, E.M., Ion, M.: Performance evaluation of attribute-based encryption: Toward data privacy in the iot. In: 2014 IEEE International Conference on Communications (ICC). pp. 725–730. IEEE (2014)
33. 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. pp. 351–356. IEEE Press (2015)