RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using *In-Situ* Machine Learning

Baibhab Chatterjee, *Student Member, IEEE*, Debayan Das, *Student Member, IEEE*, Shovan Maity, *Student Member, IEEE*, and Shreyas Sen, *Senior Member, IEEE*

***Abstract*—Traditional authentication in radio-frequency (RF) systems enable secure data communication within a network through techniques such as digital signatures and hash-based message authentication codes (HMAC), which suffer from key-recovery attacks. State-of-the-art Internet of Things networks such as Nest also use open authentication (OAuth 2.0) protocols that are vulnerable to cross-site-recovery forgery (CSRF), which shows that these techniques may not prevent an adversary from copying or modeling the secret IDs or encryption keys using invasive, side channel, learning or software attacks. Physical unclonable functions (PUFs), on the other hand, can exploit manufacturing process variations to uniquely identify silicon chips which makes a PUF-based system extremely robust and secure at low cost, as it is practically impossible to replicate the same silicon characteristics across dies. Taking inspiration from human communication, which utilizes inherent variations in the voice signatures to identify a certain speaker, we present RF-PUF: a deep neural network-based framework that allows real-time authentication of wireless nodes, using the effects of inherent process variation on RF properties of the wireless transmitters (Tx), detected through *in-situ* machine learning at the receiver (Rx) end. The proposed method utilizes the already-existing asymmetric RF communication framework and does not require any additional circuitry for PUF generation or feature extraction. The burden of device identification is completely shifted to the gateway Rx, similar to the operation of a human listener's brain. Simulation results involving the process variations in a standard 65-nm technology node, and features such as local oscillator offset and *I*–*Q* imbalance detected with a neural network having 50 neurons in the hidden layer indicate that the framework can distinguish up to 4800 Tx(s) with an accuracy of 99.9% [≈99% for 10 000 Tx(s)] under varying channel conditions, and without the need for traditional preambles. The proposed scheme can be used as a stand-alone security feature, or as a part of traditional multifactor authentication.**

***Index Terms*—Artificial neural networks (ANNs), authentication, deep neural network, device signatures, Internet-of-Things (IoT), machine learning (ML), physical unclonable function (PUF), radio frequency (RF), security.**

I. Introduction

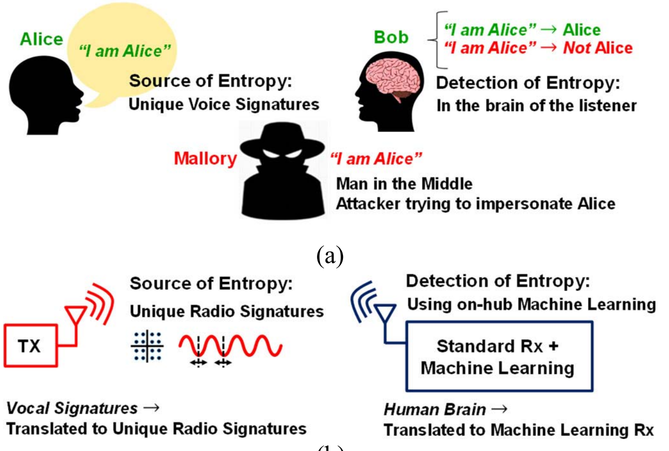
The advancements in sensor electronics, wearable technology, and mobile computing/communication platforms have resulted in an unprecedented data deluge in the domain of Internet of Things (IoT). According to CISCO's visual networking index-based global mobile data traffic forecast, machine-to-machine communication systems are expected to have about 27.1 billion connected devices by 2021 [1]. Due to their inherent mobile nature, these devices perennially operate under untrusted environmental conditions and are exposed to a number of potentially malicious attacks. The development of mobile hardware security has been comparatively slower than the improvements in computation power [2]. When these devices are required to be securely authenticated using a symmetric-key implementation, a secret key is usually placed in a nonvolatile memory (NVM) or a battery-backed SRAM and is subsequently used in a digital signature or hash-based encryption. However, these techniques are vulnerable to key-hacking (through invasive/semi-invasive/software/side channel attacks) and come with significant area and power overhead for the NVM/SRAM implementation. The widely used open authentication (OAuth 2.0) protocol [3] for current IoT devices suffer from cross-site-recovery-forgery attacks, and may eventually become cumbersome as the number of devices per user grows (OAuth requires the user to manually authenticate every device in the network). Because of these reasons, physical unclonable functions (PUF) have emerged as a promising alternative/augmentation which exploit manufacturing process variations to generate a unique and device-specific identity for a physical system [4]–[7]. PUF implementations are simpler than memory-based solutions as they consume significantly less energy and chip area than expensive cryptographic hardware such as secure hash algorithm or public/private key encryption with NVM/SRAM that may also require anti-tamper mechanisms to detect invasive attacks.

PUFs are usually classified into strong and weak PUFs depending on the number of challenge-response pairs (CRPs) that they can handle. Weak PUFs support a small number of CRPs which are linearly related to the number of components used to build the PUF. Strong PUFs, on the other hand, support a large number of CRPs such that polynomial time attacks become infeasible. These type of PUFs are usually employed in device authentication applications [2].

Manuscript received February 15, 2018; revised April 30, 2018 and June 3, 2018; accepted June 5, 2018. Date of publication June 20, 2018; date of current version February 25, 2019. This work was supported in part by the National Science Foundation SaTC, CNS, under Grant 1719235 and in part by the Semiconductor Research Corporation under Grant 2720.001. *(Corresponding author: Baibhab Chatterjee.)*

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: bchatte@purdue.edu; das60@purdue.edu; maity@purdue.edu; shreyas@purdue.edu).

Digital Object Identifier 10.1109/JIOT.2018.2849324



(a)

(b)

Fig. 1. (a) Authentication in human voice communication: Bob (the receiver) can identify Alice (the Tx) based on the unique voice signatures, and not based on the contents of what Alice says. Mallory (the impersonator) can also be identified (as not Alice), since his unique voice signatures would be different from Alice. (b) Analogous system that utilizes the proposed RF-PUF framework for secure radio communication.

IoT systems can significantly benefit from a PUF-based authentication protocol, wherein the physical characteristics of each Tx in the wireless sensor network can be analyzed and stored in a secure server as a general technique, thereby augmenting or replacing traditional key-based authentication schemes. In modern digital communication, ideal digitally modulated data pass through device-dependent unique analog/radio frequency (RF) impairments (for example, frequency error/offset and *I*–*Q* imbalance) in the transmitter chain, which are compensated for at the Rx. These process-dependent nonidealities are already present in the wireless communication signal path, and are traditionally discarded/minimized as unwanted nonidealities. In RF-PUF, we embrace those existing nonidealities through an *in-situ* light-weight machine learning (ML) engine at the Rx side, that extracts the "entropy" and creates a "strong PUF" to securely identify the Tx(s). This is similar to the inherent authentication in human voice communication as shown in Fig. 1(a). Bob (the Rx) can identify Alice (the transmitter) based on the unique voice signatures, and not based on the contents of what Alice says. Mallory (the impersonator) can also be identified as his unique voice signatures would be different from Alice. The source of entropy is in the vocal signatures of speaker (no extra hardware for entropy extraction), whereas the decoding of the entropy is in the listener's brain (heavy lifting at the Rx). Fig. 1(b) shows the analogous system (RF-PUF), wherein the unique signatures in each Tx are used for device identification in the brain of the Rx, represented by the ML hardware. The entire system is envisioned in Fig. 2 in presence of channel impairments, with each of the Tx(s) inherently working as a PUF instance [8]. Such an implementation not only helps in authentication of physical nodes in resource-constrained IoT environment but also enables applications such as intrusion detection, forensic data collection, defect detection/monitoring, and body-connected biosensors as shown in [9]–[12].

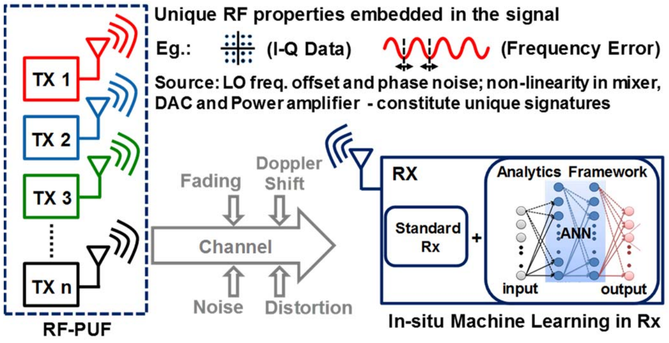

Unique RF properties embedded in the signal
Eg.:
(I-Q Data)
(Frequency Error)
Source: LO freq. offset and phase noise; non-linearity in mixer, DAC and Power amplifier - constitute unique signatures
TX 1
TX 2
TX 3
TX n
RF-PUF
Fading
Doppler Shift
Channel
Noise
Distortion
RX
Standard Rx
+
Analytics Framework
ANN
input
output
In-situ Machine Learning in Rx

Fig. 2. Visualization of RF-PUF at a system level [8].

A. Our Contribution

We have previously demonstrated the methods of process-detection in wireless radios [13]–[15] and the effect of variations in analog/mixed-signal/RF properties of such radios to adapt it for zero-margin operations [16]–[18]. In this paper, we build on that expertise to identify radio instances based on their inherent signatures automatically imparted on communicated signal, leading to a detailed analysis of the PUF properties of radios for enhanced physical layer security. The major *contributions* of this paper are as follows.

1) A conceptual development of RF-PUF is presented for an *asymmetric* IoT network, which consists multiple low-cost, low-power, distributed Tx(s), and a single central hub as a Rx. To the best of the authors' knowledge, this is the first work in this area for low-cost, preamble-less, intrinsic PUF-based authentication of IoT nodes. This is in contrast to traditional RF fingerprinting methods which are usually preamble-based and/or software defined as discussed in the next section.
2) Enabling RF-PUF operation without any extra-hardware at the resource-constrained IoT node. As described in Section III, RF-PUF does not require any additional on-chip/off-chip circuitry for PUF implementation at the Tx. The proposed scheme makes use of inherent variations resulting from factors such as process variability (on-chip) and component tolerance (on-board) for each Tx. A method to compensate for nonideal Rx signatures is also proposed in Section V.
3) Conforming to our earlier work on on-hub analytics where we proposed a method of staged inference using conditional deep learning [12], a light-weight ML framework is developed in the current work, which compensates for Rx nonidealities, and accounts for both data variability and channel variability at the same time. Since this is a nonlinear multidimensional classification problem, an artificial neural network (ANN) is employed as a learning engine. Simulation results with ≈10000 Tx(s) demonstrate ≈99% accuracy using supervised learning which proves the practical feasibility of RF-PUF for IoT-based applications targeted toward small to medium-scale smart systems with about a thousand devices connected to a single gateway Rx.

In essence, the proposed method lends the biggest benefits in asymmetric smart networks as: 1) no extra hardware is required at the resource-constrained IoT nodes, while the heavy-lifting is performed by the gateway Rx, which is similar to a listener's brain and 2) the method can be employed as a stand-alone physical-layer security feature, or for multifactor authentication, in conjunction with network-layer, transport-layer and application-layer security features.

The remainder of this paper is organized as follows. Section II lists the most recent developments in the area of PUF and RF authentication, along with their applications in the relevant domains. Section III presents the architecture of the proposed PUF in detail, while Section IV illustrates the performance metrics of the proposed method. The simulation results, along with the security aspects are analyzed in detail in Section V. Finally, Section VI summarizes this paper and points to the future directions.

II. Related Work

The notion of silicon PUFs was introduced by Gassend *et al.* [19], where the authors illustrated the use of PUF in anti-counterfeiting applications by measuring the intrinsic delays in a self-oscillating circuit. However, the additional requirement of robustness in a large sample pool led to the inclusion of various error-correction mechanisms at the system-level [20]–[23]. The error correction improves the system reliability at the cost of additional software/hardware burden on the PUF implementation. As shown in [2], many of these techniques tend to leak the secret keys that are used to generate the syndrome bits. In such a scenario, a higher number of PUF bits are generated first, and then are down-mixed to increase entropy.

RF fingerprinting has been a popular method to automatically identify the wireless nodes in a network by using the time and frequency-domain properties extracted during transmitter power-on [24]–[27]. While the transient properties are consistent, they offer acceptable classification accuracy only when the beginning and the end of the transient can be reliably identified [10]. Moreover, the analysis of transient properties require very high oversampling rates (500 MS/s in [25] and 50 GS/s in [26]) which pose significant power and precision requirements that lead to expensive Rx architectures. An alternative and less adopted approach involves the use of steady-state properties of the Tx(s), which are extracted after the communication loop transients are settled. However, the steady-state signal is data-dependent in different transmissions (with different bit-streams) which makes it unsuitable for identification purpose. For this reason, previously reported literature [9], [10] use fixed random channel access (RACH) preambles along with techniques such as spectral averaging or matched filtering to correctly identify the Tx(s). The steady-state analysis method is relatively unexplored, but is promising as it does not require sophisticated and power-hungry Rx architectures. More importantly, the steady-state portion of the signal can easily be identified as opposed to the transient states during power-on.

In this paper, we combine the concept of PUF with RF fingerprinting to develop a system architecture that utilizes RF properties of the Tx(s) to identify nodes using an *in-situ* ML framework at the Rx. A preamble-less steady state approach is adopted which is implemented by training the learning subsystem with multiple data streams and with different channel conditions. Although the proposed approach utilizes manufacturing variabilities in the transmitters for device identification (similar to state-of-the-art RF fingerprinting [28]–[31]), *RF-PUF is unique from RF fingerprinting in four different aspects:* 1) the operation for RF-PUF is not preamble-based; 2) unlike transient mode RF fingerprinting, RF-PUF does not require high oversampling ratio at the Rx; 3) RF-PUF utilizes significantly higher dimensionalities in the feature space than steady-state RF fingerprinting that gives rise to its strong PUF properties (Section III-C6) while providing justification for the nomenclature; and 4) RF-PUF compensates for the nonideal Rx signatures (Section V-A), thus allowing a large number of devices/CRPs. ML had been used in prior work [29], [30] for device identification, but the proposed work also identifies ML as a solution to the practical challenge of Rx signature compensation that often limits larger system implementations. As compared to previous implementations such as the RF-DNA [32], RF-PUF does not require any additional analog/RF hardware for PUF implementation at the Tx as the features are selected such that feature generation and extraction is ingrained in the transceiver operation (RF-DNA involved measuring the reflected/refracted EM waves based on the 3-D-positioning of scattering antennas which are different for each RF unit). Moreover, error-correction and noise cancellation measures are also intrinsic to the transceiver architecture, which increases the reliability of the proposed RF-PUF without the need for any specific error correction mechanism dedicated for the PUF operation.

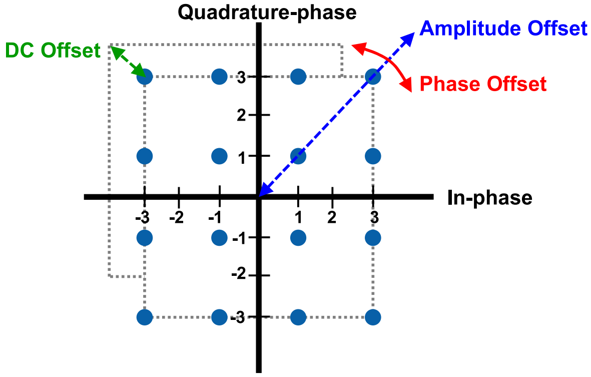


Fig. 3. DC, amplitude, and phase imbalance in 16-QAM.

III. Proposed PUF

This paper primarily focuses on a technique of authenticating devices within a low-cost swarm of IoT nodes, which can have significant variation from node to node. If all the components are tightly controlled during fabrication and manufacturing, the standard deviation of variation will be less and the number of unique devices which are correctly identified will reduce. Interestingly, that will increase the cost of fabrication significantly and hence it is cheaper and easier to embrace the nonidealities (up to a point where it does not affect the overall performance) which justifies the use of RF-PUF.

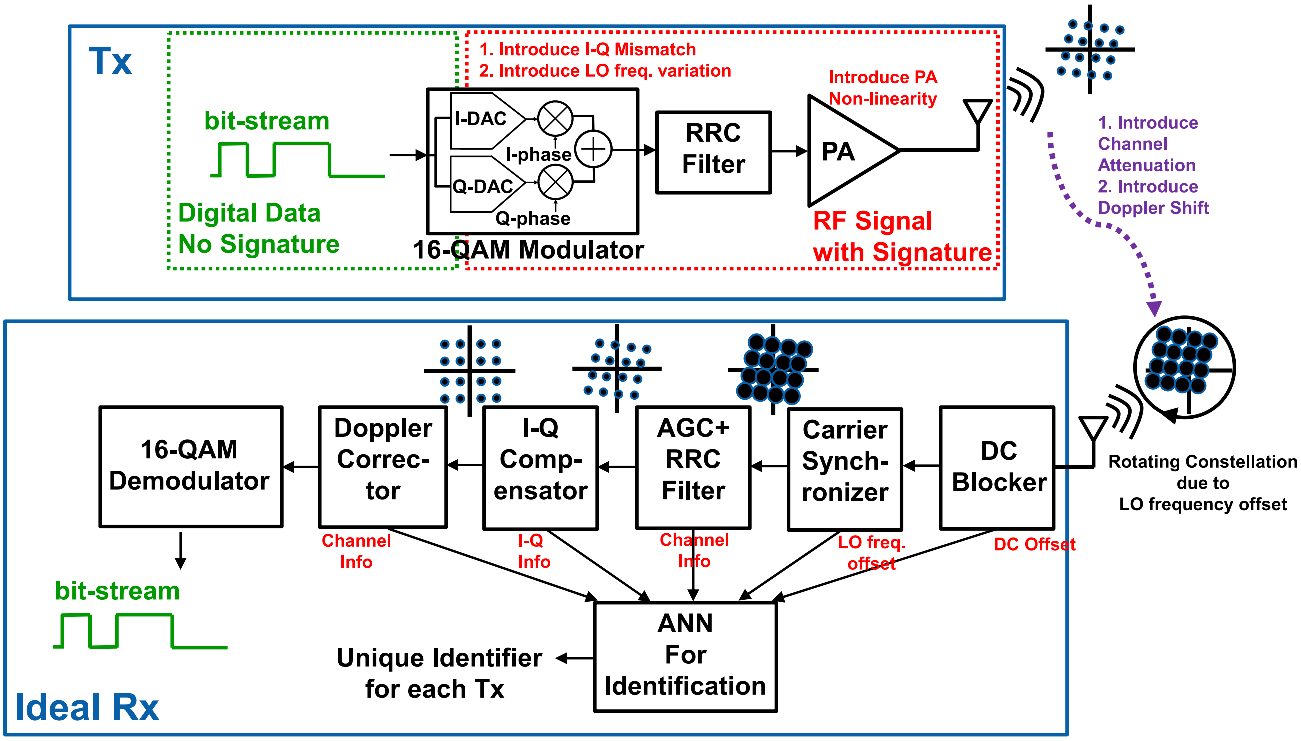


Fig. 4. System-level simulation setup involving transmitter and receiver for RF-PUF implementation.

A. Features Utilized in RF-PUF Implementation

The PUF properties of the system originate from the manufacturing variability of the Tx(s). The identification of each node is performed in the Rx subsystem which extracts multiple features from the received signals.

1) Frequency Features: Every Tx has its unique frequency offset with respect to the ideal carrier frequency because of the inherent variations in the local oscillator (LO). This offset has been used as a prime feature for device identification in [10] and [33]. The allowable limits for the frequency offsets can be different for different established standards. For example, the frequency error must be within ±25 ppm of the center frequency for the IEEE 802.11b standard for WiFi (and within ±40 ppm for IEEE 802.15.4 which is one of the preferred standard for IoT). This corresponds to a normal distribution with standard deviation (σ) of 20.1 kHz on either side of the 2.412 GHz center frequency for IEEE 802.11b. With a high-quality reference clock (low jitter with zero mean) in the Rx, frequency offsets for multiple Tx(s) in the system can be calculated. In our implementation, a carrier synchronizer module (which already exists in a standard Rx for LO offset compensation) is employed in the Rx that finds the frequency offset and compensates for it. The ppm value of the offset is provided to a three-layer machine-learning framework for identifying the device.

2) I–Q Features: The amplitude and phase mismatch between the in-phase (I) and quadrature (Q) components of the transmitted signal is unique for different transmitters. Fig. 3 shows the nature of these imbalances and how they can impact the constellation diagram at the Rx for a 16-quadrature amplitude modulated (QAM) signal. Other Tx variabilities such as power-amplifier (PA) back-off and gain variations also affect the constellation. Compressive nonlinearity affects the outer symbols in the constellation more than the inner symbols. Hence it is necessary to extract amplitude and phase information for all symbols in the constellation. These features, along with the frequency errors from each Tx has the potential to uniquely identify each Tx in the network.

3) Channel Features for Compensation: The communication channel introduces time and frequency dependent variations in various forms such as attenuation, distortion, and Doppler shift. To establish reliable operation of the RF-PUF, these channel properties need to be estimated and compensated. For this purpose, an automatic gain control (AGC) block, a root-raised cosine (RRC) filter and a Doppler corrector is employed as indicated in Fig. 4. The RRC filter, along with the AGC module, helps reducing intersymbol interference (ISI) and provides a measure of the channel attenuation to an ANN. Similarly, the Doppler corrector module estimates and corrects the amount of Doppler shift due to any physical movement of the Tx(s) and Rx(s) in the network, and provides the information to the ANN for channel compensation.

B. Communication System Example in RF-PUF: 16-QAM

Fig. 4 shows the entire transceiver system for RF-PUF authentication. The 16-QAM Tx does not have any additional circuitry for PUF implementation. The Rx, on the other hand, has multiple stages for RF signal processing and simultaneously performs feature extraction at various stages. A simple 3-layer neural network takes the extracted features as inputs and identifies the Tx(s) based on training data.

C. Properties of the RF-PUF

The proposed system has the necessary and sufficient properties of a PUF [5] that makes it suitable for security/authentication applications.

1) Constructability: A PUF class $\mathbb{P}$ is constructible if a random PUF instance ($p_r \in \mathbb{P}$) can be created by invoking a particular creation procedure, $\mathbb{P}$.Create: $p_r \leftarrow \mathbb{P}$.Create.

RF-PUF aims to exploit the technical limitations that exist in the physical process of fabricating the RF Tx(s). Hence, the manufacturing process itself serves as the creation procedure $\mathbb{P}$.Create for each of the PUF instances [RF Tx(s)].

2) Evaluability: A constructible PUF class $\mathbb{P}$ is evaluable if for a random PUF instance ($p_r \in \mathbb{P}$) and a random challenge (x), it is possible to evaluate a response $y : y \leftarrow \mathbb{P}.\text{Eval}[p_r(x)]$.

For RF-PUF, x is the challenge input bit-stream to the transmitter, and y is the unique analog response for each PUF instance. The uniqueness in the response can be attributed to multiple features, due to the die-to-die and within-die variations.

3) Reproducibility: A PUF class $\mathbb{P}$ is reproducible/reliable if it is evaluable and if the probability of intra-PUF variation being lower than a system-defined small number is very high.

For the RF-PUF, the measure of reproducibility can be defined as the difference (D_{intra}) between two distinct evaluations $(y(x), y'(x))$ of a particular PUF on the challenge x.

$$\text{D}_{\text{intra}}(x) \cong \text{dist}[y_1(x), y_1'(x)]$$

D_{intra} is the intra-chip (intra-PUF) distance that serves as a metric to measure the resilience of the RF-PUF to varying environmental conditions. Reproducibility/reliability is also a measure of stability as $\text{D}_{\text{intra}} = 0\%$ in an ideal scenario.

4) Uniqueness: A PUF class $\mathbb{P}$ is unique if it is evaluable and if the probability of inter-PUF variation being higher than a system-defined large number is very high.

For the RF-PUF, the measure of uniqueness can be defined as the difference of the responses between two PUF instances $(y_1(x), y_2(x))$ evaluated with the same challenge x.

$$\text{D}_{\text{inter}}(x) \cong \text{dist}[y_1(x), y_2(x)]$$

5) Identifiability: A PUF class $\mathbb{P}$ is easily identifiable if it is reproducible as well as unique, and if the probability of intra-PUF variation being lower than inter-PUF variation is very high.

$$\text{Prob}(\text{D}_{\text{intra}} < \text{D}_{\text{inter}}) \approx 1$$

As will be seen in Section IV, RF-PUF simultaneously exhibits reproducibility, uniqueness, and identifiability to an acceptable degree. In the simulation results shown in Section IV, the worst case D_{inter} (3.9 ppm: geometric mean of ppm variations over all features) was found to be larger than the corresponding D_{intra} (2.9 ppm: geometric mean of ppm variations over all features) for 1000 Tx(s). These properties coupled with the physical unclonability and unpredictability of the silicon manufacturing process makes the RF-PUF implementation practically feasible.

6) PUF Strength: The challenge for RF-PUF is a digital data sequence, while the response contains the analog features embedded in actually transmitted RF signal (and hence in the received signal) which is unique for each Tx. Since the CRPs consist features which are real-valued analog numbers, the total number of CRPs for m distinct analog/RF features will be $\mathbb{R}^m$ where $\mathbb{R}$ represents all values in the real number space within a range of $\pm 3\sigma$ around the mean. When each of these features is quantized using a 16 bit ADC, for example, $\mathbb{R}^m$ translates to 2^{16m} which is a large number even for small values of m (3–10). Hence, if a probabilistic polynomial time (PPT) adversary knows the responses from k Tx(s), it is possible to predict the response for the $(k+1)$th Tx only with a negligible probability of $(1/2^{16m})$. This property makes RF-PUF a strong PUF, which is suitable for authentication applications [2].

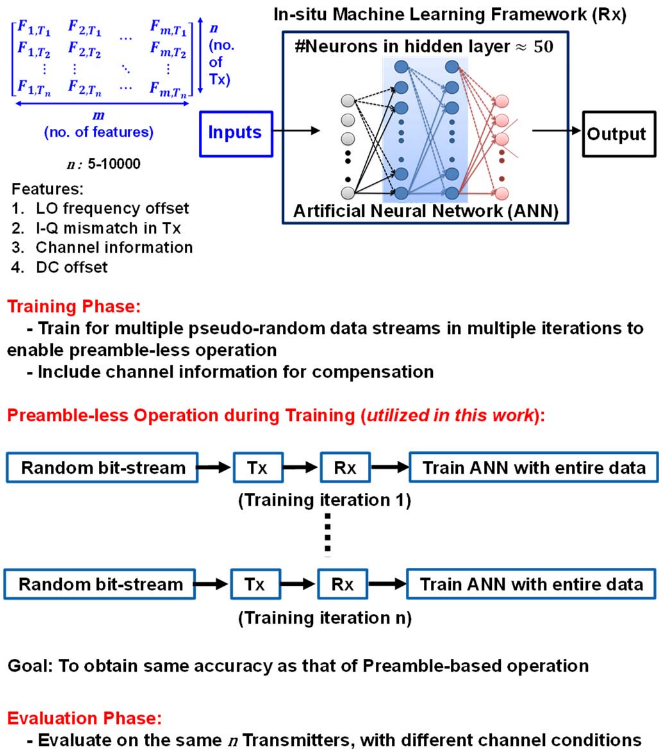
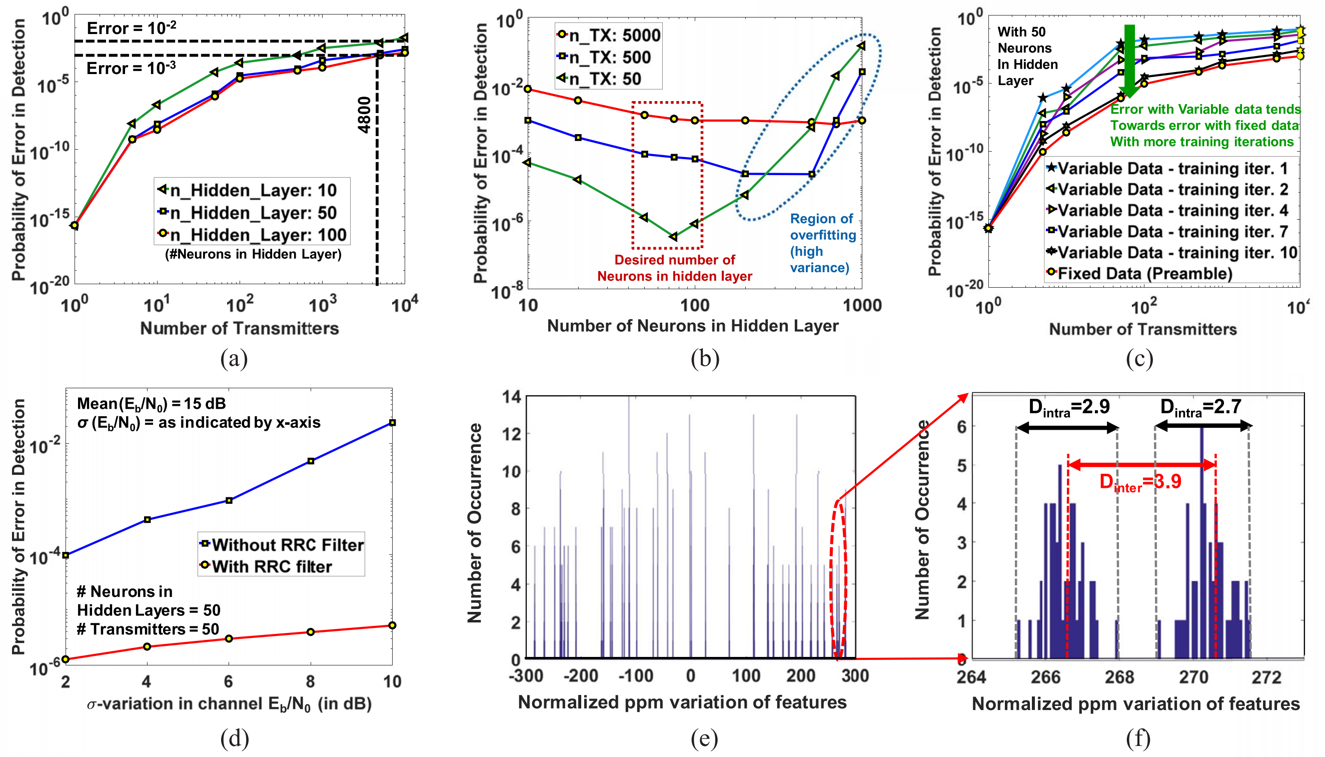

$\begin{bmatrix} F_{1,T_1} & F_{2,T_1} & \cdots & F_{m,T_1} \\ F_{1,T_2} & F_{2,T_2} & & F_{m,T_2} \\ \vdots & \vdots & \ddots & \vdots \\ F_{1,T_n} & F_{2,T_n} & \cdots & F_{m,T_n} \end{bmatrix}$
n (no. of Tx)
m (no. of features)
n: 5-10000
Features:
1. LO frequency offset
2. I-Q mismatch in Tx
3. Channel information
4. DC offset
In-situ Machine Learning Framework (Rx)
#Neurons in hidden layer ≈ 50
Inputs
Output
Artificial Neural Network (ANN)
Training Phase:
- Train for multiple pseudo-random data streams in multiple iterations to enable preamble-less operation
- Include channel information for compensation
Preamble-less Operation during Training (utilized in this work):
Random bit-stream
Tx
Rx
Train ANN with entire data
(Training iteration 1)
Random bit-stream
Tx
Rx
Train ANN with entire data
(Training iteration n)
Goal: To obtain same accuracy as that of Preamble-based operation
Evaluation Phase:
- Evaluate on the same n Transmitters, with different channel conditions

Fig. 5. Preamble-less training for the ANN. This method enables on-the-fly authentication without knowing the expected bit-stream from the Tx [8].

D. Training the ANN for Device Identification

The three-layer ANN is trained in multiple iterations with different pseudo-random bit-streams to account for data variability in evaluation stage and to enable preamble-less operation. The hyper-parameters, including the number of training iterations [shown in Fig. 6(c), that represent sampling and collection of data streams under dynamic channel conditions which vary slowly during a single evaluation period but can change significantly from one evaluation to another], were optimized aggressively through scaled conjugate gradient backpropagation algorithm, using variable number of epochs and a target training-set-error. For the data presented in subsequent sections, the number of training iterations was set to 10, with a stream length of 30 000 bits. The training dataset was presented to the ANN as a $n \times m$ feature matrix as shown in



(a) (b) (c)

(d) (e) (f)

Fig. 6. (a) Probability of false detection as a function of the total number of Tx in the system. (b) Probability of False detection as a function of the number of neurons in the hidden layer of 3-layer ANN. (c) Reduction in prediction error with increased number of training iterations. Each iteration trains the network with a different bit-stream and different channel conditions. (d) Probability of false detection as a function of the standard deviation (σ) of the E_b/N_o in the channel in presence of additive white Gaussian noise. (e) PUF instances with 1000 Tx (50 instances for each Tx, with different channel conditions) and 50 neurons in hidden layer with respect to geometric mean of normalized ppm variation of features. (f) Worst case reliability (D_{intra}) and uniqueness (D_{inter}) for the 1000 transmitters.

Fig. 5. The extracted channel conditions for the ten iterations were also presented as features so that the network learns to compensate for the channel variations. The number of neurons in the hidden layer was varied and was optimized for performance (detection accuracy). During evaluation phase, a different pseudo-random data-stream from any of the n Tx(s) is provided to the ANN in presence of a different channel condition. This mode of training helps the neural network to learn both data variabilities and channel variabilities, thereby leading to a robust system.

IV. Performance Metrics

The proposed PUF is simulated using the Neural Network toolbox in MATLAB, with a 16-QAM modulation scheme. The manufacturing process variations of a standard 65-nm technology is included during the simulations to model the statistical variability in the range of ($\mu \pm 3\sigma$). A total of 10 000 PUF devices were simulated under varying channel conditions. Table I shows the mean and standard deviation of the Tx features and channel features that are used during simulations. The LO frequency and frequency error follows the IEEE 802.11b standard. I–Q imbalance has a mean value of 0, while the linearity of the PA is defined by the back-off value (with respect to 1 dB compression point). E_b/N_0 is the ratio of energy per bit and noise which defines the signal to noise ratio at the Rx.

A. Probability of False Detection

Fig. 6(a) shows the probability of false detection of a Tx as a function of the total number of Tx(s) (n_Tx) in the network, which indicates that the error is $<10^{-3}$ up to 4800 Tx(s), and $<10^{-2}$ for 10 000 Tx(s). Fig. 6(b) illustrates the probability of false detection of a transmitter as a function of the number of neurons in the hidden layer of the ANN (n_Hidden_Layer). It is to be noted from both Fig. 6(a) and (b) that the probability of an error in detection does not reduce much when the n_Hidden_Layer is more than 50. Increasing the size of the neural network beyond this limit will cause overfitting and will increase the power and area cost without significant performance benefit.

Fig. 6(c) shows the effect of training in multiple iterations with variable data as compared to training with a fixed data-stream (preamble). In case of preamble-based training, the ANN is trained using only the fixed headers in the data-stream. In our implementation, conversely, the ANN is trained using variable data-stream in multiple iterations. As the number of training iterations increase, the performance of the system tends toward the performance achieved in fixed-preamble case. This overhead of additional training iterations are also useful in terms of learning the channel conditions and variabilities, as each iteration has a different channel condition which the network learns to compensate.

TABLE I
TRANSMITTER AND CHANNEL FEATURES USED FOR SIMULATION [34], [35]

Feature	Average (μ)	Standard Deviation (σ)
LO Frequency	2.412 GHz	20.1 kHz (8.3 ppm)
I-Q Amplitude Imbalance	0 dB	1 dB
I-Q Phase Imbalance	0º	5º
PA back-off (linearity)	30 dB	1 dB
E_b/N_0	15 dB	2 dB
Doppler Shift	0 Hz	1 Hz

B. Effect of Noise, Channel Variation and ISI

Even for short-range (<30 m) communication, the channel is affected by noise/attenuation in the communication medium, interference, Doppler shift, and fading, out of which the contribution of attenuation is the most dominant [36]. Fig. 6(d) illustrates the effect of channel attenuation on the probability of false detection. Without the RRC filter, ISI increases which leads to an error probability of $\approx$ 0.02 in device identification when the standard deviation of channel E_b/N_0 is 10 dB (with mean $E_b/N_0 = 15$ dB). In presence of the RRC filter at the Rx, it is easier to extract features from the *I–Q* data for different Tx(s), because of a reduction in ISI. As a result, probability of error reduces to as low as 10^{-5} for the range of variations shown in Fig. 6(d).

C. Intra-PUF Hamming Distance (Reliability) and Inter-PUF Hamming Distance (Uniqueness)

Fig. 6(e) and (f) illustrates the reliability and uniqueness of RF-PUF with respect to the input features. It is to be noted that unlike a traditional PUF circuit that produces digital output, RF-PUF embeds the unique signature of the Tx in the RF properties of the message signal, and hence the intra-PUF and inter-PUF distances are plotted using the normalized parts-per-million (ppm) variation of the features. To represent the ppm variations of multiple features on a single axis, a transformation of the feature spaces is required. Since the total number of possible PUF instances is proportional to $\prod_{i=1}^{m} N_i$ (m = number of features, N_i = number of possible values for feature i), it is intuitive to utilize the geometric mean of the ppm values of all the features for representing a PUF instance.

The worst case inter-PUF variation with 1000 Tx(s) is found from the simulations and is shown Fig. 6(f). The worst-case inter-PUF difference thus defined is very close to the intra-PUF difference which explains the high (in the range of 10^{-3} or higher) probability of false detection when number of Tx(s) are more than a few thousand [Fig. 6(a)].

D. Randomness and Bias

For the guessing entropy to be low, the randomness of the PUF needs to be high, whereas the bias needs to be low. For the simulations shown in this paper, the *NIST recommended* random bits [37], [38] are used to generate the features within a range of $(\mu \pm 3\sigma)$, as shown in Table I. The resulting pass rates for the RF-PUF output (normalized geometric mean of feature values) is shown in Fig. 7, alongside the pass rates for NIST recommended random bits (NIST RND). Evidently, the pass rates for the PUF output are higher than 0.9 for each of the 15 tests, which matches closely with the NIST RND. Any bias in the PUF output is refuted by the frequency test, which exhibits a pass rate >0.95.

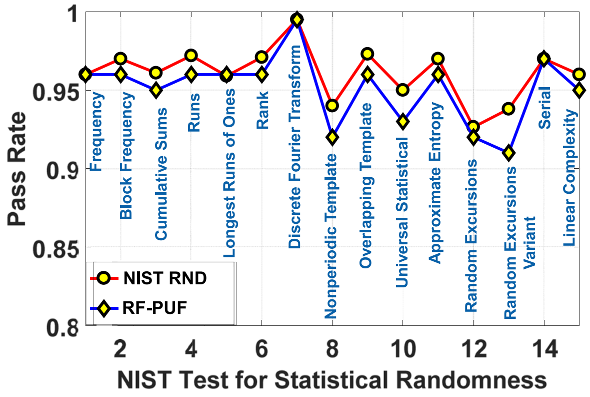


Fig. 7. Results of randomness test for RF-PUF using NIST test suite [38].

E. Experimental Validation of RF-PUF: Physical Implementation With SDRs

To prove the feasibility of RF-PUF in hardware, an experimental setup (as shown in Fig. 8) is developed using software-defined-radios (SDR). The SDRs are ideally programmed to operate at 2.4 GHz, with zero *I–Q* imbalance. However, there will be both frequency error and *I–Q* nonidealities in the practical scenario. The frequency error is captured using a spectrum analyzer, while the *I–Q* imbalance can be captured by configuring one of the SDRs as a Rx, using the GNU radio platform. Fig. 8 shows 2 SDRs, with a 23 kHz difference in their carrier frequencies. This difference in frequency can be detected at the Rx side from the down-converted *I–Q* data, by sensing the baseband signal over a duration which is inversely proportional to the difference in frequency. To generate enough number of unique classes that help validating the learning engine, multiple unique Tx(s) are artificially emulated from these SDRs by modifying the ambient temperature in a closed environment. Changing the temperature in discrete steps of 5 °C in the range of 0 °C–25 °C modifies the Tx properties, and the Rx considers every 5 °C change in the temperature as a different Tx when it does not have any information about the temperature. The nonideality information from the 2 SDRs, along with data from 8 other emulated Tx(s) are provided as input to the neural network framework (Section III-D), which detects all ten Tx(s) correctly. This confirms that the neural network can identify multiple Tx(s) with a small difference in their RF features. In a more realistic scenario, the features from the Tx(s) need to be automatically calibrated/compensated at the Rx with respect to different temperature and supply voltages before the classification using the ANN. For that, the system would require a temperature and supply sensing mechanism, and additional preprocessing at the on-board application processor. Alternatively, the temperature and voltage information can be directly provided to the neural network so that it learns to compensate for the variations, which would be included as a future work.

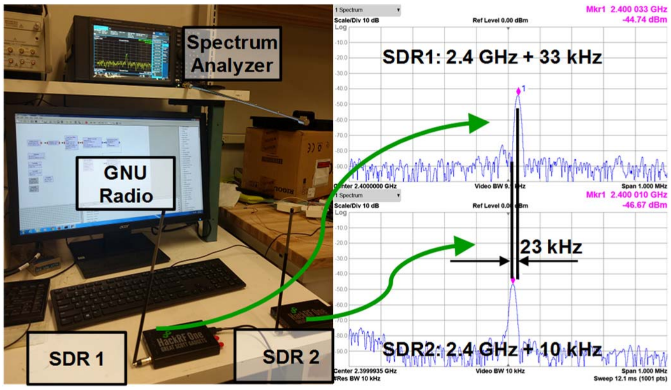


Fig. 8. Experimental setup to extract physical transmitter properties. Two SDR @ 2.4 GHz are shown in this setup, exhibiting a frequency difference of 23 kHz as nonideality. The amplitude and phase imbalance is captured in the GNU radio software after reception.

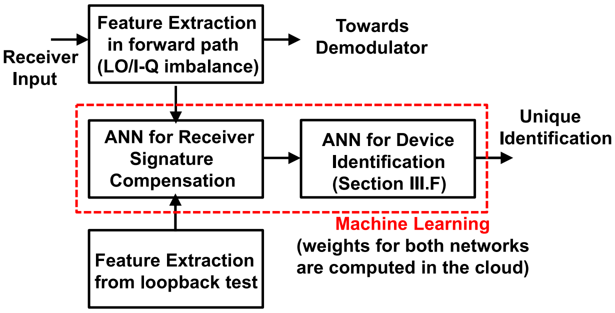

Receiver Input
Feature Extraction in forward path (LO/I-Q imbalance)
Towards Demodulator
ANN for Receiver Signature Compensation
ANN for Device Identification (Section III.F)
Unique Identification
Machine Learning
(weights for both networks are computed in the cloud)
Feature Extraction from loopback test

Fig. 9. One possible method that accounts for receiver nonidealities while performing device identification. A neural network performs the receiver signature compensation before device identification.

V. Discussion on Rx Signatures, Attack Models, Security, and Robustness

A. *Compensating Rx Signatures*

It is evident from Fig. 6(a) that the ML framework can support up to 10 000 Tx(s) with a probability of false detection $< 10^{-2}$ and up to 4800 Tx(s) with a probability of false detection $< 10^{-3}$. Fig. 6(b) indicates that the required number of neurons in the hidden layer should be within 50–100 to successfully detect the Tx(s), which can be implemented on a processor in case of an SDR environment. While this is promising in terms of the conceptual feasibility, the Rx nonidealities will pose significant implementation constraints on the system.

For the initial system simulation, an ideal Rx has been assumed which does not insert any signature of its own into the feature set extracted from the Tx(s). However, in a practical scenario, the Rx will alter the signatures in the received signal. This can be taken care of using a loopback analysis to find the Rx properties [39], post which the system can be incrementally trained for such Rx nonidealities. Alternatively, the Rx subsystem can be corrected pre-emptively for nonidealities (values of which are obtained through the loopback test) using a second set of LO offset compensator (carrier synchronizer), AGC, and I–Q compensator modules as shown in Fig. 4.

Fig. 9 shows the training-based method for compensating the Rx signatures. A second neural network learns the functions to be employed on individual features to compensate the nonidealities, and then performs the linear transformations on each of those features. The compensated feature vector is then provided to the original ANN that performs the device identification. Fig. 10 illustrates the effect of this compensation method. When the number of Tx(s) is more than 25, the probability of false detection increases from 10^{-7} in the ideal Rx case to 10^{-2} in the practical Rx case without compensation. With compensation, this value again reduces to 10^{-6}. When the number of Tx(s) is >8000, the error is about 10^{-2} in the case with Rx compensation, which is sufficient for most smart-home applications.

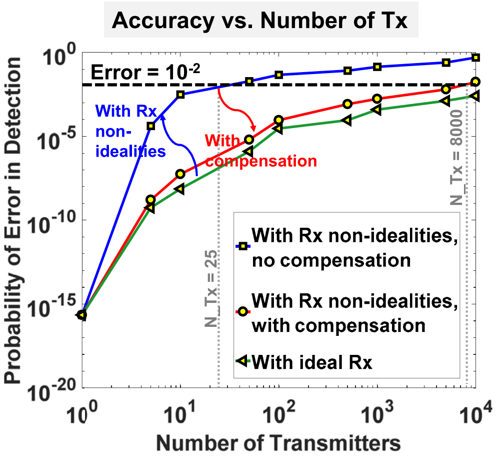


Fig. 10. Probability of false detection as a function of the total number of transmitters in the system, with and without receiver signature compensation.

B. *Possible Attack Models*

In [40], the possible attack mechanisms on a strong PUF are classified into two primary categories: 1) the PUF reuse model and 2) the malicious PUF model. In most practical applications, the attack usually comprises of a combination of the two models. The PUF reuse model is based on the scenario that an adversary can have repeated temporary physical access to the PUF when the PUF is communicating with an authenticating medium. This presents the adversary an opportunity to model and replay the responses. On the other hand, the malicious PUF model assumes that either the PUF responses can be simulated using a software algorithm, or the adversary can have direct access to all the CRPs through a built-in logger program/implanted Trojan. Since RF-PUF does not store any digitally encoded signature, it does not suffer from the malicious PUF model. However, it can potentially suffer from PUF reuse models, as described in the next part of the discussion.

1) *Replay Attack Model:* Replay attacks are theoretically possible in the proposed RF-PUF based authentication system, as a wireless Tx can send identical digital data-streams repeatedly (in other words, PUF reuse cannot be avoided). This scenario is presented in Fig. 11. Alice, the Tx wants to send a message, "Hi, I'm Alice" to the gateway Rx. This message inherently contains the Tx signature (TxS). We assume that the adversary (Ad) intercepts the transmitted data and in the

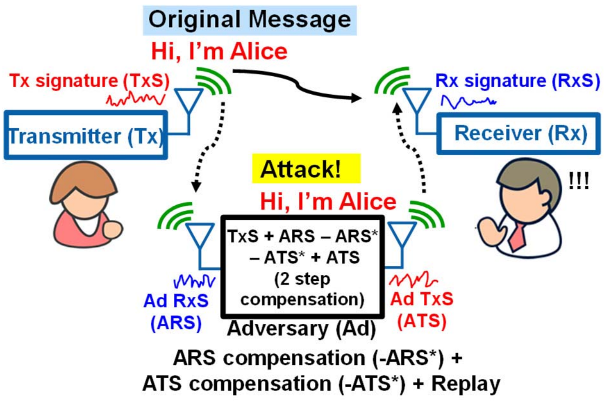


Fig. 11. One possible replay attack model: adversary needs to perform its own receiver signature compensation (−ARS*) and transmitter signature compensation (−ATS*). The necessary conditions to successfully replay the message from Alice with the original TxS are ARS=ARS* and ATS=ATS*.

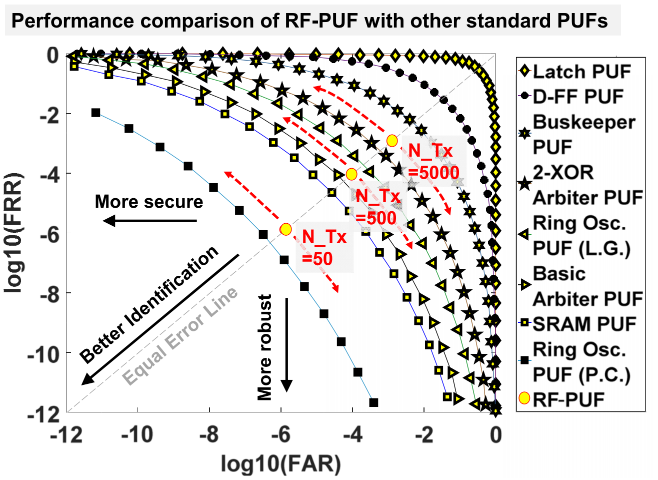


Fig. 12. Comparison of RF-PUF with various other PUFs available in literature. The values for FAR and FRR for other PUFS are taken from [44].

process, adds the attacker's Rx signature (ARS) to the signal. Subsequently, Ad compensates for the ARS using −ARS*, which require high-speed and high resolution circuits that can negate minuscule changes in voltage and time. To successfully mimic Alice's signatures, Ad needs to compensate for its own Tx signatures (ATS) as well. This also utilizes high speed and high resolution circuits and compensates ATS using −ATS*. If ARS=ARS* and ATS=ATS*, the adversary can mimic Alice by sending the same message and same signatures. However, implementing both ATS and ARS compensation would require expensive ADCs and DACs for achieving negligible residual errors.

One may argue that the machine-learning-based gateway Rx compensation shown in Section V-A could also need similar high-resolution ADCs. However, imperfect compensations in the gateway Rx results in a small residual error, which only introduces a constant shift in the detection threshold for all the Tx(s). On the other hand, for the attacker model, both ARS and ATS compensations have to be accurate. Otherwise, the replay attack is imperfect as the replayed signal does not mimic Alice's signature, and hence the adversary may not be successful in impersonating Alice. This limitation makes the replay attack extremely costly in the RF-PUF scenario.

2) *ML-Based Modeling Attacks:* The proposed system can also suffer from a machine-learning attack as an external attacker can model the responses (using a separate learning engine) from RF-PUF through repeated access to the transmitted data.

However, it must be noted that the proposed system utilizes a supervised learning algorithm to uniquely identify the PUF devices. For an external attacker, mimicking a particular Tx will translate to an unsupervised learning (clustering) problem, and hence the modeling accuracy will directly depend on the ratio of the number of CRPs accessed by the attacker to the total number of CRPs [41]. The modeling time, on the other hand, will depend on the number of CRPs that the adversary has access to. Hence, there exists a tradeoff between the modeling time and accuracy, which is described with an example in the next section.

C. Countermeasures Against PUF Reuse Attacks

Ostrovsky *et al.* [42] implemented a strict one-time-usability protocol to avoid PUF reuse and modeling attacks. However, for our application, one-time-use is not feasible, as the transmitters (i.e., the PUF instances) need to send data whenever they are required to. As shown in [40], an alternative way to thwart such attacks is to simultaneously incorporate two additional properties during PUF implementation: 1) erasability and 2) certifiability. Erasability requires that the single responses from the PUF would be made impossible to read back without affecting other responses. This facilitates tamper-detection, but requires logical reconfigurability and additional control circuitry to be implemented on-chip. Certifiability, on the other hand, signifies that a PUF response must be checked offline (without any external "Trusted Authority") for possibilities of tampering, or any change in the PUF properties. This also requires additional circuitry in each PUF instance. A detailed analysis of the additional circuitry required for erasability and certifiability is out of scope of this paper.

In the absence of additional circuits for erasability and certifiability, RF-PUF can suffer from external ML attacks. In this section, we perform an estimate of the training time for an ML attack in order to demonstrate its practical limits for a problem with high dimensionality. As shown in Section III-C6, the number of CRPs for the RF-PUF is 2^{16m} where m is the number of features considered. For a nominal value of $m = 5$, the number of CRPs become $2^{80} \approx 10^{24}$. Reference [43] shows that a machine-learning based attack model on 10^6 CRPs takes 267 days to get completed on an INTEL Quadcore Q9300 processor. 10^{24} CRPs will result in significantly higher training time, leading to a wait time of several years before completing the attack.

D. Security and Robustness

In Fig. 12, RF-PUF is compared with various other PUFs in state-of-the-art literature [44]. Two well-known metrics: 1) false acceptance rate (FAR) and 2) false rejection rate (FRR) are used to define the security and robustness, respectively, for the PUFs. A low value for both FAR and FRR ensures that the authentication is both secure and robust. However, there is a tradeoff between these two quantities which can be controlled by modifying the detection threshold of PUF inferencing mechanism (the ML framework for RF-PUF). If the inferencing is done by a threshold which is much higher than the inter-PUF distance, FAR increases while FRR reduces, which means better robustness. On the other hand, if the decision is made using a threshold which is significantly lower than the inter-PUF distance, FAR reduces while FRR increases, leading to higher security. By varying the detection threshold, different levels of FAR and FRR can be achieved.

With a small number of Tx(s) (≈ 50), the FAR and FRR for RF-PUF along the equal error line is very close to the pairwise-compared (P.C.) ring oscillator PUF [45], and is much better than the other PUFs in Fig. 12. As the number of Tx(s) increase, the overall error rate increases as the inter-PUF distances reduce. However, depending on the application scenario, the detection threshold within the ML framework can be altered, leading to either highly secured or highly robust systems.

VI. Conclusion

In this paper, a conceptual development of RF-PUF is presented along with a feasibility study showing that the inherent RF properties arising from the manufacturing process in a wireless node can be exploited as a strong PUF for device authentication in asymmetric IoT networks without any additional hardware at the Tx. Using an *in-situ* ML-based framework, up to 10 000 Tx(s) can be detected with about 99% accuracy. The proposed method also eliminates the need for preamble-based or key-based identification of modern IoT nodes and enables low-cost secure authentication using the intrinsic properties embedded in the RF signal that does not have any extra hardware cost at the transmitter. Consequently, the proposed scheme does not consume any additional power at the Tx side. The Rx, however, requires two neural networks which can be implemented using the on-board microprocessor at a nominal power cost (additional 3%–5% overhead when the neural networks are powered on, as estimated from [46] and [47]) which is not significant if the network is asymmetric. As a future direction, more advanced methods of Rx signature compensation will be analyzed along with circuit techniques to implement erasability and certifiability, leading to practical and efficient implementation of the RF-PUF hardware. A formal or experimental validation of the achievable protection degree against different attack models will also be analyzed as a part of the future work. One other research direction involves the stability analysis of RF-PUF in presence of temperature and supply voltage variation.

References

[1] *CISCO VNI Forecast Highlights Tool*. Accessed: Sep. 29, 2017. [Online]. Available: https://www.cisco.com/c/m/en_us/solutions/service-provider/vni-forecast-highlights.html

[2] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," *Proc. IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014.

[3] *OAuth2.0*. Accessed: Jun. 5, 2018. [Online]. Available: https://oauth.net/2/

[4] M.-D. Yu, R. Sowell, A. Singh, D. M'Raihi, and S. Devadas, "Performance metrics and empirical results of a PUF cryptographic key generation ASIC," in *Proc. IEEE Int. Symp. Hardw. Oriented Security Trust (HOST)*, San Francisco, CA, USA, 2012, pp. 108–115.

[5] R. Maes, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, KU Leuven, Leuven, Belgium, 2012.

[6] G. Kömürcü and G. Dündar, "Determining the quality metrics for PUFs and performance evaluation of two RO-PUFs," in *Proc. IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, Montreal, QC, Canada, 2012, pp. 73–76.

[7] K. Xiao *et al.*, "Bit selection algorithm suitable for high-volume production of SRAM-PUF," in *Proc. IEEE Int. Symp. Hardw. Oriented Security Trust (HOST)*, Arlington, VA, USA, 2014, pp. 101–106.

[8] B. Chatterjee, D. Das, and S. Sen, "RF-PUF: IoT security enhancement through authentication of wireless nodes using *in-situ* machine learning," in *Proc. IEEE Int. Symp. Hardw. Oriented Security Trust (HOST)*, Washington, DC, USA, 2018, pp. 205–208.

[9] R. M. Gerdes, T. E. Daniels, M. Mina, and S. F. Russell, "Device identification via analog signal fingerprinting: A matched filter approach," in *Proc. 13th Annu. Netw. Distrib. Syst. Security Symp. (NDSS)*, 2006, p. 78.

[10] P. Scanlon, I. O. Kennedy, and Y. Liu, "Feature extraction approaches to RF fingerprinting for device identification in femtocells," *Bell Labs Tech. J.*, vol. 15, no. 3, pp. 141–151, 2010.

[11] S. Maity, D. Das, and S. Sen, "Wearable health monitoring using capacitive voltage-mode human body communication," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2017, pp. 1–4.

[12] M. Parsa, P. Panda, S. Sen, and K. Roy, "Staged inference using conditional deep learning for energy efficient real-time smart diagnosis," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, 2017, pp. 78–81.

[13] S. Sen, V. Natarajan, R. Senguttuvan, and A. Chatterjee, "Pro-VIZOR: Process tunable virtually zero margin low power adaptive RF for wireless systems," in *Proc. Design Autom. Conf. (DAC)*, Anaheim, CA, USA, 2008, pp. 492–497.

[14] S. Sen and A. Chatterjee, "Design of process variation tolerant radio frequency low noise amplifier," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seattle, WA, USA, 2008, pp. 392–395.

[15] S. Sen, D. Banerjee, M. Verhelst, and A. Chatterjee, "A power-scalable channel-adaptive wireless receiver based on built-in orthogonally tunable LNA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 946–957, May 2012.

[16] D. Banerjee, B. Muldrey, S. Sen, X. Wang, and A. Chatterjee, "Self-learning MIMO-RF receiver systems: Process resilient real-time adaptation to channel conditions for low power operation," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2014, pp. 710–717.

[17] S. Sen, V. Natarajan, S. Devarakond, and A. Chatterjee, "Process-variation tolerant channel-adaptive virtually zero-margin low-power wireless receiver systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 12, pp. 1764–1777, Dec. 2014.

[18] S. Sen, "Invited: Context-aware energy-efficient communication for IoT sensor nodes," in *Proc. Design Autom. Conf. (DAC)*, Austin, TX, USA, 2016, pp. 1–6.

[19] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proc. ACM Conf. Comput. Commun. Security*, 2002, pp. 148–160.

[20] G. Suh, "AEGIS: A single-chip secure processor," Ph.D. dissertation, Dept. EECS, Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.

[21] R. Maes, P. Tuyls, and I. Verbauwhede, "A soft decision helper data algorithm for SRAM PUFs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, South Korea, 2009, pp. 2101–2105.

[22] M.-D. Yu and S. Devadas, "Secure and robust error correction for physical unclonable functions," *IEEE Des. Test*, vol. 27, no. 1, pp. 48–65, Jan./Feb. 2010.

[23] Z. Paral and S. Devadas, "Reliable and efficient PUF-based key generation using pattern matching," in *Proc. IEEE Symp. Hardw. Oriented Security Trust (HOST)*, San Diego CA, USA, 2011, pp. 128–133.

[24] K. J. Ellis and N. Serinken, "Characteristics of radio transmitter fingerprints," *Radio Sci.*, vol. 26, no. 4, pp. 585–597, Jul./Aug. 2001.

[25] J. Hall, M. Barbeau, and E. Kranakis,“Detecting rogue devices in Bluetooth networks using radio frequency fingerprinting,” in *Proc. Int. Conf. Commun. Comput. Netw. (CNN)*, 2006, pp. 108–113.

[26] O. H. Tekbas, N. Serinken, and O. Ureten, “An experimental performance evaluation of a novel radio-transmitter identification system under diverse environmental conditions,” *Can. J. Elect. Comput. Eng.*, vol. 29, no. 3, pp. 203–209, Jul. 2004.

[27] R. Matheson. *Fingerprinting Chips to Fight Counterfeiting*. Accessed: Sep. 29, 2017. [Online]. Available: http://news.mit.edu/2015/fingerprinting-chips-fight-counterfeiting-0501

[28] I. O. Kennedy, P. Scanlon, and M. M. Buddhikot, “Passive steady state RF fingerprinting: A cognitive technique for scalable deployment of co-channel femto cell underlays,” in *Proc. IEEE Symp. New Front. Dyn. Spectr. Access Netw.*, Chicago, IL, USA, 2008, pp. 1–12.

[29] B. Kroon, S. Bergin, I. O. Kennedy, and G. O. Zamora, “Steady state RF fingerprinting for identity verification: One class classifier versus customized ensemble,” in *Proc. Irish Conf. Artif. Intell. Cogn. Si. (AICS)*, 2009, pp. 198–206.

[30] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, “Deep learning for RF device fingerprinting in cognitive communication networks,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[31] T. J. Bihl, K. W. Bauer, and M. A. Temple, “Feature selection for RF fingerprinting with multiple discriminant analysis and using ZigBee device emissions,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 8, pp. 1862–1874, Aug. 2016.

[32] G. DeJean and D. Kirovski, “RF-DNA: Radio-frequency certificates of authenticity,” in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, 2007, pp. 346–363.

[33] P. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoust.*, vol. AE-15, no. 2, pp. 70–73, Jun. 1967.

[34] *ANSI/IEEE 802.11b-1999*. Accessed: Apr. 4, 2018. [Online]. Available: https://standards.ieee.org/findstds/standard/802.11b-1999.html

[35] *End-to-End QAM Simulation With RF Impairments and Corrections*. Accessed: Apr. 7, 2018. [Online]. Available: https://www.mathworks.com/help/comm/examples/end-to-end-qam-simulation-with-rf-impairments-and-corrections.html

[36] A. Iyer, C. Rosenberg, and A. Karnik, “What is the right model for wireless channel interference?” *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2662–2671, May 2009.

[37] A. Rukhin *et al.*, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, Rev 1a*, document 800-22, NIST, Gaithersburg, MD, USA, 2000.

[38] *NIST SP 800-22*. Accessed: Feb. 13, 2018. [Online]. Available: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software

[39] D. Banerjee, B. Muldrey, X. Wang, S. Sen, and A. Chatterjee, “Self-learning RF receiver systems: Process aware real-time adaptation to channel conditions for low power operation,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 195–207, Jan. 2017.

[40] U. Ruhrmair and M. van Dijk, “PUFs in security protocols: Attack models and security evaluations,” in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, 2013, pp. 286–300.

[41] J. Delvaux and I. Verbauwhede, “Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise,” in *Proc. IEEE Int. Symp. Hardw. Oriented Security Trust (HOST)*, 2013, pp. 137–142.

[42] R. Ostrovsky, A. Scafuro, I. Visconti, and A. Wadia, “Universally composable secure computation with (malicious) physically uncloneable functions,” in *Eurocrypt* (LNCS). Heidelberg, Germany: Springer, 2013.

[43] U. Rührmair *et al.*, “Modeling attacks on physical unclonable functions,” in *Proc. ACM Conf. Comput. Commun. Security*, 2010, pp. 237–249.

[44] R. Maes, “Chapter 5: PUF-based entity identification and authentication,” in *Physically Unclonable Functions: Constructions, Properties and Applications Book*. Heidelberg, Germany: Springer, 2013.

[45] G. E. Suh and S. Devadas, “Physical unclonable functions for device authentication and secret key generation,” in *Proc. Design Autom. Conf. (DAC)*, San Diego, CA, USA, 2007, pp. 9–14.

[46] J. Jiang and C. Claudel, “A high performance, low power computational platform for complex sensing operations in smart cities,” *HardwareX*, vol. 1, pp. 22–37, Apr. 2017.

[47] E. Painkras *et al.*, “SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation,” *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013.

Baibhab Chatterjee (S’17) received the B.Tech. degree in electronics and communication engineering from the National Institute of Technology Durgapur, Durgapur, India, in 2011, and the M.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2015. He is currently pursuing the Ph.D. degree at the School of Electrical Engineering, Purdue University, West Lafayette, IN, USA.

His industry experience includes two years as a Digital Design Engineer/Senior Digital Design Engineer with Intel, Bengaluru, India, and one year as a Research and Development Engineer with Tejas Networks, Bengaluru. His current research interests include low-power analog and RF and mixed-signal circuit design for secure biomedical applications.

Mr. Chatterjee was a recipient of the University Gold Medal, both while he was earning the B.Tech. and M.Tech. degrees and the Andrews Fellowship at Purdue University in 2017–2018.

Debayan Das (S’17) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2015. He is currently pursuing the Ph.D. degree at the SPARC Laboratory, Purdue University, West Lafayette, IN, USA.

He was an Analog Design Engineer with xSi Semiconductors, Bengaluru, India, a start-up company, for one year. His current research interests include hardware security and mixed-signal IC design.

Mr. Das was a recipient of the IEEE HOST Best Student Paper Award in 2017.

Shovan Maity (S’18) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2012, and the M.Tech. degree in electrical engineering from Indian Institute of Technology Bombay, Mumbai, India, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering at Purdue University, West Lafayette, IN, USA.

He was an Analog Design Engineer with Intel, Bengaluru, India, from 2014 to 2016. His current research interests include design of circuits and systems for human body communication, hardware security, and mixed signal circuits design.

Shreyas Sen (S’06–M’11–SM’17) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2011.

He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. He has over five years of industry research experience with Intel Labs, Santa Clara, CA, USA, Qualcomm, San Diego, CA, USA, and Rambus, Sunnyvale, CA, USA. He has authored or co-authored 2 book chapters, over 100 conference and journal papers, and has 13 patents granted/pending. His current research interests include mixed-signal circuits/systems for Internet of Things and biomedical and security.

Dr. Sen was a recipient of the AFOSR Young Investigator Award in 2017, the NSF CISE CRII Award in 2017, the Google Faculty Research Award in 2017, the Intel Labs Divisional Recognition Award in 2014 for industry-wide impact on USB-C type, the Intel Ph.D. Fellowship in 2010, the IEEE Microwave Fellowship in 2008, the GSRC Margarida Jacome Best Research Award in 2007, the Best Paper Awards of HOST in 2017 and 2018, the ICCAD Best-in-Track Award in 2014, the VTS Honorable Mention Award in 2014, the RWS Best Paper Award in 2008, the Intel Labs Quality Award in 2012, the SRC Inventor Recognition Award in 2008, and the Young Engineering Fellowship in 2005. He was chosen as one of the top 10 Indian Inventors Worldwide under 35 (MIT TR35 India Award) in 2018, by the MIT Technology Review for the invention of using the Human Body as a Wire, which has the potential to transform healthcare, neuroscience, and human–computer interaction. He serves/has served as an Associate Editor for *IEEE Design & Test*, an Executive Committee member of the IEEE Central Indiana Section, and an ETS and Technical Program Committee member of DAC, CICC, DATE, ISLPED, ICCAD, ITC, VLSI Design, IMSTW, and VDAT.