2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

A Hardware-Software Codesign Approach to Identity, Trust, and Resilience for IoT/CPS at Scale

Farah Kandah[◇], Joseph Cancelleri[†], Donald Reising[†], Amani Altarawneh[◇], Anthony Skjellum[#◇]
Computer Science and Engineering[◇], *Electrical Engineering*[†], *SimCenter*[#]
University of Tennessee at Chattanooga
Chattanooga, TN, USA

***Abstract*—Advancement in communication technologies and the Internet of Things (IoT) is driving adoption in smart cities that aims to increase operational efficiency and improve the quality of services and citizen welfare, among other potential benefits. The privacy, reliability, and integrity of communications must be ensured so that actions can be appropriate, safe, accurate, and implemented promptly after receiving actionable information. In this work, we present a multi-tier methodology consisting of an authentication and trust-building/distribution framework designed to ensure the safety and validity of the information exchanged in the system. Blockchain protocols and Radio Frequency-Distinct Native Attributes (RF-DNA) combine to provide a hardware-software codesigned system for enhanced device identity and overall system trustworthiness. Our threat model accounts for counterfeiting, breakout fraud, and bad mouthing of one entity by others.**

Entity trust (e.g., IoT devices) depends on quality and level of participation, quality of messages, lifetime of a given entity in the system, and the number of known "bad" (non-consensus) messages sent by that entity. Based on this approach to trust, we are able to adjust trust upward and downward as a function of real-time and past behavior, providing other participants with a trust value upon which to judge information from and interactions with the given entity. This approach thereby reduces the potential for manipulation of an IoT system by a bad or byzantine actor.

Keywords*—*IoT, RF-DNA fingerprinting, Trust management, Blockchain

I. Introduction

The integration of wireless communications into IoT devices has facilitated ubiquity of IoT devices and increased the ease with which such devices have been integrated into daily life. Although IoT provides users with easy control-at-a-distance over such things as lights, thermostats, and doors, IoT infrastructure is not without vulnerabilities. Three such vulnerabilities are that (1) approximately 70% of IoT devices employ weak or no encryption [1], (2) IoT devices often use commercially available and open source wireless communications standards, and (3) wireless access points remain a key point through which attacks occur [2]. Thus, there remains a need for effective approaches capable of bolstering IoT security before there can be trustworthy systems at scale that incorporate IoT.

Over the past two decades, significant research has focused on enhancing a wireless network's digital security mechanisms (e.g., encryption) through the use of Physical (PHY) layer techniques [3]–[8]. Radio Frequency-Distinct Native Attributes (RF-DNA) fingerprinting is one PHY layer approach that has been introduced to augment digital security mechanisms by facilitating the identification of wireless transmitters by exploiting PHY layer characteristics. In the context of this communication, RF-DNA will be the mechanism used to provide a *physically unclonable* property for IoT devices, yielding a unique identity for each device within our architecture.

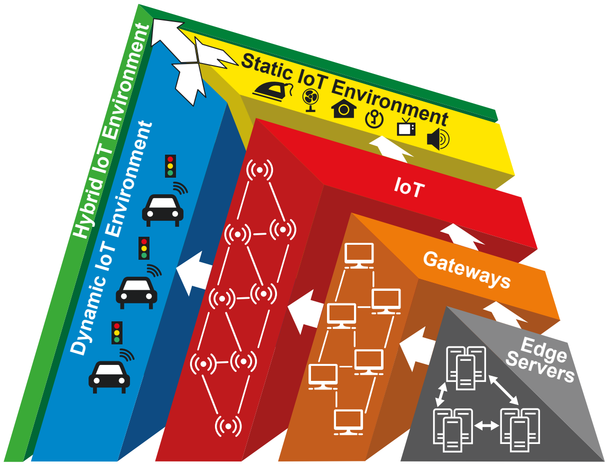

Figure 1: IoT integration toward a resilient architecture

Over the past decade plus, significant work on Blockchain technology, commencing with Bitcoin [9], has led both to a cryptocurrency revolution and to innovative uses of distributed, immutable ledgers to provide consensus information, smart contracts, and highly available, permanent, unforgeable, distributed data sets (i.e., providing for completeness and integrity) [10]–[13]. Here, trust information is considered at both local and non-local ("global") levels (generalizable to N-level hierarchies) in order to enable and manage trust dissemination.

IoT systems are based on a collective organization in which devices collaborate to provide better and more accurate decisions. It is important to ensure that the information being shared is legitimate to avoid any significant degradation in system performance because of false or inaccurate information. Building trust—the "assurance" between two devices that the information being shared can be used with confidence that it is accurate—will create a trustworthy, secure system in which all devices are identified and no information is accepted from any unauthorized device.

Therefore, the aim of this work is to combine (1) strong identity based on a lightweight blockchain technology and (2) Radio Frequency-Distinct Native Attributes (RF-DNA) fingerprinting to produce a secure, scalable, trustworthy environment for IoT devices. This environment will be suitable for

978-1-7281-2980-8/19/$31.00 ©2019 IEEE
DOI 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00191

use in many applications, such as scale-out to support smart cities, in which security is paramount. This work is timely, transformative, and of broad impact because of the rapid roll-out and scale-out of IoT systems, the rapid development of smart environments, and emerging threats involving large-scale IoT systems.

A key assertion is that the system must identify devices accurately both initially and over time in order to establish and maintain useful measures of trust. A second key assertion of this work is that hardware-software codesign is essential to achieve strong device identity. Third, we assert that Blockchains provide a transformative architectural component for IoT trust management because of their unique properties of immutability, decentralization, integrity, and DDoS resistance.

A key outcome is the trust mechanism for each entity (IoT device). Here, entity trust depends on quality and level of participation, quality of messages, lifetime of a given entity in the system, and the number of known "bad" (non-consensus) messages sent by that entity. Based on this approach to trust, we are able to adjust trust upward and downward as a function of real-time and past behavior, providing other participants with a trust value upon which to judge information and interactions of the given entity. This approach consequently reduces the potential for system manipulation by bad or byzantine actors.

The remainder of the paper is organized as follows: We discuss related work in Section II, followed by our motivations and contributions in Section III. We present the threat model in Section IV. Our Hardware-Software Codesign Approach is given in Section V, followed by our experimentation and evaluation in Section VI. We discuss how our proposed approach mitigates the threat model in Section VII. Finally, we conclude and discuss future directions in Section VIII.

II. Related Work

Related work presented in this section covers three underlying areas for this paper: Trust Management, Blockchain, and Wireless Transmitter Identity Verification.

A. Trust Management

Trust is based on the history of interactions and the validity of the information exchanged between network entities [14]–[16]. Recently, the idea of managing trust in the network has received significant attention since it adds an additional security layer designed to ensure that the data being exchanged in the network is valid and originates from a trustworthy source [17], [18]. Several trust management schemes have been proposed, including entity-based, data-based, and hybrid trust models [19]. One area of interest in cyber-physical systems is connected vehicles. Compared to static networks, the dynamic nature of connected vehicles requires a distributed system that allows vehicles to gather and share information toward building trust in the network as they move from one place to another (this trust building can be achieved through collaboration between the connected vehicles and fixed roadside units).

Previous work has proposed solutions for trust management implementation in Vehicle Ad Hoc Network—Intelligent Transportation Systems (VANET-ITS) (e.g., [19]–[23]). The authors in [19] proposed a decentralized system, claiming that a centralized system is impractical for the growth that a VANET-ITS would require; this concern is clearly valid. By having several roadside units (RSUs) located throughout a city, each area within it can be divided roughly equally, and therefore the load will be reasonably balanced as well. Furthermore, the authors in [19] continue by proposing trust-factor calculations. Each vehicle begins with a neutral value, and, as messages are passed between vehicles, the trust value will be updated depending upon the accuracy of messages that were previously passed. The method of evaluating the accuracy of a message is based on the experiences that other vehicles in the network have had with a given message. The critical drawback of this approach is the scenario in which there are several malicious vehicles in the network and these vehicles collude to evaluate their messages as accurate. This scenario increases the malicious vehicles' trust factor, thus decreasing the overall integrity of the system.

B. Blockchain

Blockchain is a recent, breakthrough technology used in the financial industry. Blockchains create a consistent, tamper-proof ledger that records information without the need for a centralized bank [12]. Notable Blockchains include Bitcoin [9] and Ethereum [10]. For instance, Ethereum's key features include decentralized control, availability, tamper-proof properties, and the consensus (mining) algorithm through proof-of-work (PoW) and/or proof-of-stake (PoS) [11]. Tamper-proof is a key feature of Blockchains. Any malicious nodes in the system would be unable to tamper with previous blocks because of the data structure (subject to Byzantine limits). Later blocks depend on data collected from earlier blocks, and, if any changes are made to such earlier blocks, this manipulation will create a disparity in the chain [12]. Some Blockchain protocols support smart contracts, which are self-executing scripts; smart contracts are immutable and possible because of this tamper-proof Blockchain feature [13].

In [19], the authors proposed to use Blockchain as a means of storing the trust factor of each vehicle. As vehicles exchange messages, these messages will be compiled into one data block that will then be uploaded to a local RSU. A given RSU will then compete against other RSUs in the network via a joint Proof-of-Work (PoW) and/or Proof-of-Stake (PoS) to determine which will be elected as the miner. Using a joint PoW/PoS will evidently prevent RSUs that are in a high-demand area (with lots of ratings (stakes)) from always uploading to the chain by also allowing smaller and less commonly accessed RSUs to upload their data to the Blockchain too. Although this work proposed the use of Blockchain to support trust management in the network, it lacks the ability to verify the data being added to the block by RSUs.

The authors in [24] proposed the solution of retaining only the last few blocks of a Blockchain on each vehicle to save storage. While this solution will certainly require less storage capacity on vehicles, it will greatly increase the communications needed between vehicles and RSUs. Should a vehicle be unable to access all of the Blockchain data and come into contact with another vehicle whose trust is unknown, then an RSU query must be made to determine the trust factor of said vehicle. Querying the RSU each time a device comes into contact with a new vehicle can be a costly action. A method to eliminate such delays is necessary for a connected vehicle system to function in real time.

C. Wireless Transmitter Identity Verification

The work in [25] presented the first case in which RF-DNA fingerprinting was used not only to verify the identity of known/trusted transmitters but also in the rejection of rogue transmitters. A rogue transmitter is one that falsifies its digital credentials (e.g., MAC address) to pose digitally as a trusted transmitter in order to circumvent digital-based network security mechanisms. For the eighteen Worldwide Interoperability for Microwave Access (WiMAX) transmitters used in [25], a 93% rogue transmitter rejection was achieved for 72 unique digital identity spoofing attacks at a signal-to-noise ratio (SNR) of 21 dB. At the same SNR, the rogue rejection performance was improved to 100% when an accurate sample of the overall WiMAX transmitter population was used. These results were achieved using dimensionally reduced RF-DNA fingerprints and a neural-network-based classifier known as Generalized Relevance Learning Vector Quantization-Improved (GRLVQI). In [25], the models used for verification of a trusted transmitter's identity and rogue transmitter rejection were developed by training the classifier using a distinct class for each of the trusted transmitters (six and eight classes for the 93% and 100% rejection performance at SNR=21 dB). For a given SNR, the results in this work are based upon training the classifier using only two distinct classes. One class represents the trusted transmitter whose identity is to be verified based upon the presented digital credentials, and the second class consists of the remaining trusted transmitters; thus, there is a classifier model, consisting of two classes, for each of the trusted WiMAX transmitters.

The work in [26] presents rogue device feature classification using k-Nearest Neighbor (KNN) machine learning. Their results demonstrated an accuracy from 30% up to 94% at SNR=15 dB. In comparison to KNN, the choice of Support Vector Machine (SVM) was driven by (1) less sensitivity to outliers, (2) the ability to use kernel functions to facilitate the mapping of the RF-DNA fingerprints to an n-dimensional feature space to reduce problems associated with nonlinear data, (3) the ability to fine tune its hyperparameters without requiring the same level of precision to produce the same results, and (4) greater robustness in cases of unpredictability.

Identity verification of five wireless transmitters using SVM and RF fingerprints is first presented in [27]. The wireless transmitters used are 3GPP UMTS mobile handsets, and verification is performed by comparing a given handset's RF fingerprints with the SVM model corresponding to that handset's digital credentials. In this case the digital credential is the International Mobile Subscriber Identifier (IMSI) assigned to each mobile handset. The RF fingerprints are extracted from the preamble associated with the Random Access Channel (RACH). The paper [27] investigates identity verification using SVM in both a single class and a "customized ensemble" case. The customized ensemble was investigated using three approaches: tiered, weighted tiered, and double weighted tiered. For all three tiered approaches, an SVM model is developed using a pair of handsets (that is, three given handsets, A, B, and C, require the development of three SVM models: (A and B), (A and C), and (B and C)). Thus, if a handset presents the IMSI of handset A, then two SVM models are required in the verification of the handset identity. In the approach presented here, a handset presenting the IMSI of A would have its RF fingerprints compared to a single SVM trained using RF fingerprints from A, B, and C in which one class represents A and the second class represents B and C. Lastly, the paper [27] used signals that were collected within an anechoic chamber and neglected analysis as SNR degraded.

III. Motivations and Contributions

This project ensures the security of IoT/Cyber-physical Systems (CPS) through a dynamic information infrastructure that incorporates lightweight blockchain technology for integrity, availability, and identity while ensuring trust among IoT/CPS devices. Since the vast majority of these components are manufactured in environments of limited trust and are later deployed in critical infrastructures worldwide, we need to design and develop solutions for protecting both hardware and software that take into account the variety of attacks and threats inherent in such devices. Attacks can originate from untrusted hardware of an IoT/CPS system and/or from the Internet by exploiting existing communication protocols and/or network traffic. Hardware attacks against such systems can occur via physical tampering of a device and/or by the introduction of a cloned or counterfeit device [28]–[32]. Software attacks against the system can be performed through network attacks such as phishing, Denial of Service (DoS), and/or data spoofing [33], [34]. Our approach aims to disallow classes of cloning and counterfeiting threats—as well as man-in-the-middle attacks against secure connections—through unique identity.

The contributions of this work are as follows:

- Design of an RF-DNA fingerprinting approach to provide unique identity to devices (hardware-based security)
- Introduction of a behavioral trust management approach between the devices that include hysteresis
- Architecture of a multi-layer decentralized database based on Blockchains that manage trust information (presently two levels)
- Construction of dynamic trust available "locally" and "globally" by integrating RF-DNA fingerprinting, trust algorithms, and a two-level blockchain

IV. Threat model

We present a number of threats that target IoT devices:

Threat 1. Breakout Fraud: Devices in the network can participate and exchange messages collaboratively, and decisions will be made based on these interactions. Devices can attack the system by maintaining a period of (or initial) good behavior that yields a high level of trust, then start injecting the network with invalid information. We present the behavioral monitoring trust approach (Section V-A) that helps the network identify anomalous behaviors, thus quarantining malicious devices.

Threat 2. Bad Mouthing: Devices in the network will work collaboratively and exchange messages to enhance system performance and create a shared state between network devices. Malicious devices may incorrectly report low trust values of peers in the network, thus enabling them to gain higher trust values compared to their peers; this situation will mislead the system to quarantine legitimate devices improperly. We present the multi-level Blockchain approach (Section V-B) to overcome this issue; we do so by creating a multi-level verification system to validate the trust updates in the system. **Threat 3. Counterfeiting/Impersonation:** The availability of cheap knockoffs of big brands creates a challenge of true identify and genuine devices in the network. Adversaries can inject the network with malicious devices that will participate in illegal ways in order to interfere with legitimate network devices, thus degrading their performance. We present the RF-DNA fingerprinting approach (Section V-C) to aid the network in identifying and better authenticating network devices to discover both initial counterfeits and device impersonation during system operation.

V. Hardware-Software Codesign Approach

We start by presenting our trust management approach with focus on how trust between devices will be realized and dynamically managed by taking device behavior in the system into consideration; trust is evaluated based on interactions with peers and the quantity and quality of those interactions as compared with its peers.

A. Behavioral Trust Management

Previous trust-based schemes of which we are aware have been based solely on the history of communications [20]–[23]. While message validation is an essential component in such systems, it is critical for system entities to know who to trust; therefore, in our design, we consider three types of trust:

a) Direct trust ($D_t(u, v)$) is established between device u and device v that are within each other's direct transmission range.
b) Indirect trust ($I_t(u, w)$) is established between device u and device w based on neighbor-of-neighbor connection.
c) Reputational trust ($R_t(u, v, w)$) can be formed between device u and its directly connected device v based on the information gathered from device w.

Along with the aforementioned levels of trust, we will use an integral-action approach to manage and build the trust between devices. Within its peer group of devices, a device's trust is based on the quality and quantity of interactions, as well as the lifetime of the device in the system.

Four major factors drive trust: the first measures participation of the device and its current trust as compared to its peers in the system; second, the device's behavior in the system, which is monitored by the number of messages it generates and has shared in the system during its lifetime, and the critically (rank) of the messages being shared by the device during its lifetime; and third, the time it took a device to build these messages as compared to the total uptime of the system since the last update. The fourth factor is based on the fraction of bad messages sent by the device in the system. We weighted these four factors as θ (20%) for the current behavior and participation, where the device will be rewarded for participation based on its behavior among other devices. Sometimes this proves to be a good incentive for the devices to participate in order to maintain their trust value. But, it is not the case for all devices. For example, if the device is among the best behaving and its trust value is 92% or higher, then the maximum weight it achieves is 20% (θ) , 20% (χ) for the rank and number of messages, 20% (ϕ) for the total uptime of the device being active to share their messages, and 40% (τ) for the fraction of bad messages shared by the device.

B. Multi-level decentralized database

A Blockchain enables a decentralized approach with its key features such as tamper-proofing, consistency, and timeliness, which makes it a strong candidate for storing data regarding the system. The use of Blockchain can assist in critical aspects of the system. Tamper-proofing is possible because, by design, Blockchains prevent any previous data from being modified without the change being discovered. If a malicious node should attempt to modify any contents of the Blockchain, it will be indisputable to all other nodes in the network and the attempted change will subsequently be disregarded. Blockchain also satisfies consistency by frequently creating and distributing new blocks, allowing all devices to maintain a consistent data set. Finally, timeliness assures the low-cost ability to verify that newly created blocks are valid and can be added to the chain, which in turn allows the system to be updated in real time.

Our design incorporates two levels of Blockchains: At the IoT devices level, a Cluster Blockchain (C_B) will be formed for each cluster of devices. And, at the gateway level, the Global Blockchain (G_B) will be formed based on the cluster's Blockchain, where gateways begin mining to add the data (cluster's Blockchain) onto the global Blockchain. The concept behind the global Blockchain is to form a global trust management system that provides devices with a global view of the trust in the system, which will benefit IoT devices as they move to different regions of the system. The cluster Blockchain is formed to store a localized trust consensus based on each cluster, as it will only be accessible by members of said cluster. This limited accessibility will ensure that the data being sent to the gateway is valid and will be loaded onto the global Blockchain with confidence.

1) Cluster Blockchain (C_B): Because of the high overhead associated with Blockchains, we propose the use of a smaller, less computationally demanding Blockchain that stores trust values of devices in a cluster. A cluster will be able to add data onto its Blockchain quicker than it could onto the global Blockchain. This approach better supports the real-time requirement of the connected IoT system. These groups will work similarly to the overall design of a connected IoT system, in which the devices are passing messages, and, when enough data is collected, after a distributed consensus on the data, a miner will be elected to add the block to the chain.

The use of a less computationally demanding algorithm allows for blocks to be generated and mined quickly, which in turn allows the dynamic nature of connected devices to be unaffected by the high overhead associated with adding blocks to a chain. Furthermore, there are dynamic scenarios where IoT devices move or are being moved under a new gateway zone. A threshold will be set to determine when to upload the cluster's chain onto the global Blockchain, where the elected miner will be required to push the cluster chain to a gateway upon meeting the threshold of blocks. The gateway will be required to verify that the cluster chain is valid and will then proceed to perform the calculations necessary to create a new block out of the data and upload this and potentially other cluster' chains to the global Blockchain. Upon uploading, the cluster will "forget" the previous chain and begin constructing a new one. Because an index block is also necessary and must be provided by a trusted source, we propose that the gateway will respond to the uploaded cluster chain with the index block, which will contain the trust values for all of the devices in the group. This arrangement will allow all devices to have a quick reference point for the trust factors of other cluster members, therefore decreasing the overhead of having to query the gateway to obtain this information.

By using a cluster Blockchain between groups of devices, along with the tamper-proofing nature of the Blockchain in general, we seek to prevent any malicious device from modifying contents of the Blockchain prior to data being distributed to all devices in the network. Furthermore, with each group of devices working on their own data sets and cluster chains, cluster Blockchains will enable rapid upload of data to the global Blockchain, which ensures the real-time nature that is required by the system.

2) Global Blockchain (G_B): The global Blockchain stores trust factors of all devices in the system. It will obtain these values from the numerous cluster Blockchains that will be uploaded to the gateways. As a gateway receives data from the cluster Blockchains, it will begin to mine a new block. When a gateway is attempting to mine a block to add to the global Blockchain, the data will likely be several cluster Blockchains from the many clusters that the gateway has come into contact with recently. The global Blockchain is considered the source of trust management in the system because of the tamper-proof concept discussed earlier. Upon mining a block, the gateway will broadcast the newly created block to maintain consistency throughout the network.

C. RF-DNA Fingerprinting

RF-DNA fingerprinting is presented here as a PHY layer mechanism for verifying the identity of a specific transmitter. RF-DNA fingerprints exploit the distinct and unique coloration that is imparted on the waveform during its formation and transmission.

1) Signal of Interest: This work uses signals collected from eighteen Alvarion BreezeMAX Extreme 5000 802.16e WiMAX Mobile Subscriber (MS) transmitters. The network implemented a 60/40 Time Division Duplexing (TDD), meaning the first 60% of the 5.0 ms time frame includes a Base Transceiver Station (BTS) Down-Link (DL) transmission, and the remaining 40% is for MS Up-Link transmission. The frames themselves have been shown to have three sub-frame UL modes: Data-Only, Range-Plus-Data, and Range-Only. RF-DNA fingerprints used in this work were generated from from Range-Only responses. The WiMAX TDD UL signals lack consistency between frames with respect to both time and spectrum. It has also been observed that the sub-frames comprising BreezeMAX UL responses possess an inherent DC offset over their duration (Fig. 2). The cause of this characteristic bias is uncertain, but it is this region from which the RF-DNA fingerprints are generated [35].

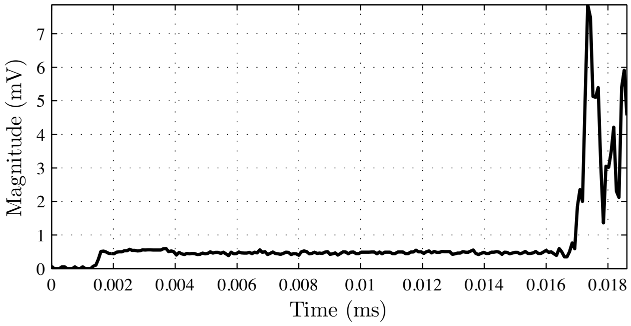

Figure 2: "Near-Transient" TDD UL sub-frame Range-Only response. [35].

2) Signal Collection and Detection: The signal collection and processing technique adopted by this work is based on the process in [36] and illustrated in Figure 3. Generated signals were collected with an adjustable 36 MHz RF filter equipped Agilent E3238S-based RF Intercept and Collection System (RFSICS). Post-collection, the band of frequencies were down-converted via the implementation of a 70 MHz Intermediate Frequency (IF), then digitized using a 12-bit analog-to-digital converter (ADC) with a 95 mega-samples-per-second operating frequency. Next, the signal is once again converted to baseband. It is digitally filtered at a value of 9.28 MHz bandwidth, subsequently Nyquist sub-sampled, then finally formatted into complex In-phase (I) and Quadrature (Q) samples. Prior to RF-DNA fingerprint generation, the initial device transmission point was determined using the Variance Trajectory (VT) burst detection processes detailed in [37].

3) RF-DNA Fingerprint Generation: The RF-DNA fingerprints used here are the same as those generated and used in [25], which are extracted from the Discrete Gabor Transform

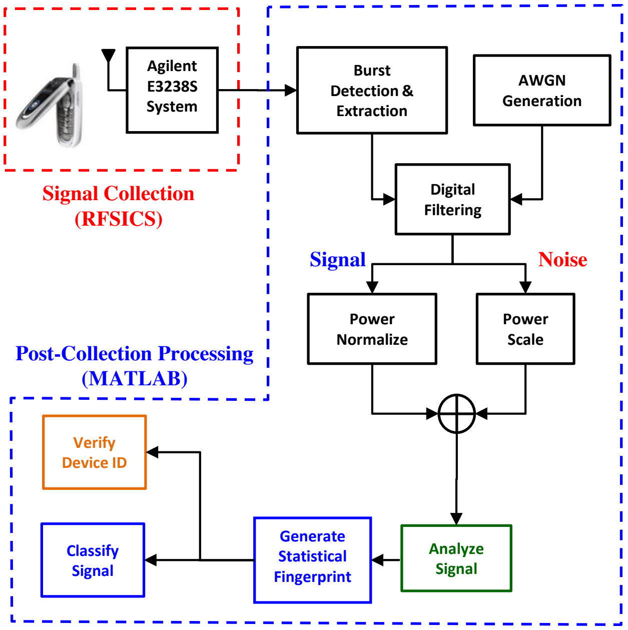

Figure 3: Signal collection and post-collection processing [36].

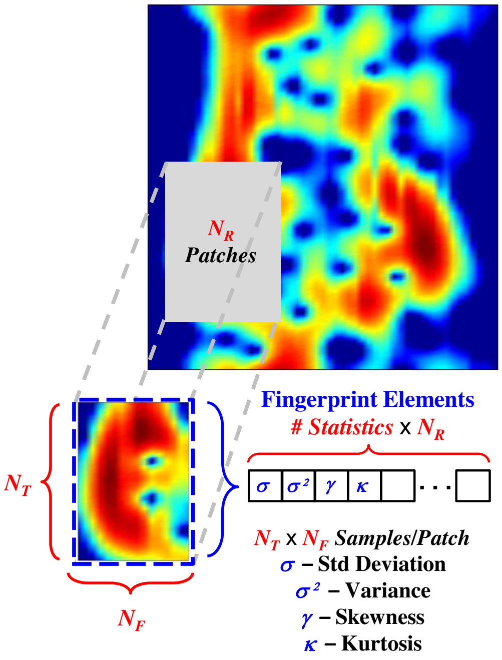

Figure 4: Gabor-based RF-DNA fingerprint generation process showing $N_T{\times}N_F$ 2D patches taken from centered, normalized, magnitude-squared coefficients ($|G_{mk}|^2){\in}[0,1]$ of a WiMAX signal [36].

(DGT). The DGT is given by

$$G_{\eta k} = \sum_{m=1}^{MN_\Delta} s(m) W^*(m - \eta N_\Delta) \exp^{-j2\pi km/K_G}, \tag{1}$$

where $s(m){=}s(m+lMN_\Delta)$ is the periodic input signal, $W(m){=}W(m+lMN_\Delta)$ is the periodic analysis window, N_Δ is the number of samples shifted, $\eta{=}1,2,\ldots,M$ for M total shifts, and $k{=}0,1,\ldots,K_G-1$ for $K_G{\geq}N_\Delta$ and $mod(MN_\Delta, K_G){=}0$ are satisfied. As in [25], the RF-DNA fingerprints were generated from the normalized magnitude-squared Gabor coefficients in which the DGT is oversampled (i.e., $K_G{\geq}N_\Delta$). Figure 4 shows a representative normalized Gabor magnitude-squared response. This two-dimensional time-frequency (T-F) response was subdivided into $N_T{\times}N_F$ patches, formed into N_{TF} vectors, and features extracted. The features used in forming the RF-DNA fingerprints are standard deviation (σ), variance (σ^2), skewness (γ), and kurtosis (κ). The DGT was calculated using values of $M{=}150$, $K_G{=}150$, and $N_\Delta{=}1$; thus, the DGT was oversampled with a factor of 150. Prior to dimensional reduction, each RF-DNA fingerprint is comprised of $N_f{=}204$ total features.

4) Dimensional Reduction: Dimensional reduction is used to minimize the number of RF-DNA fingerprint features in an effort to reduce computational complexity and data storage space while maintaining transmitter discrimination performance. The work in [25] used the feature relevance vector (λ) to perform dimensional reduction. The relevance vector is produced when training the Generalized Relevance Learning Vector Quantization-Improved (GRLVQI) classifier. For a given signal-to-noise ratio (SNR), the "best" relevance vector (λ_B) is given as

$$\lambda_B(\text{SNR}) = \left[\lambda_1, \lambda_2, \ldots, \lambda_{N_f}\right], \tag{2}$$

where λ_j is the relevance value associated with the j^{th} RF-DNA fingerprint feature and $\lambda_j{>}\lambda_k$ indicates that the j^{th} feature is more influential on the classification decision than that of the k^{th} feature. For a given SNR, the "best" relevance vector corresponds to the GRLVQI classifier model that resulted in the lowest error across all noise realization and cross-validation folds. For the results in [25] and the dimensionally reduced RF-DNA fingerprints used here, the "best" relevance vectors from each SNR were averaged together and the resulting average relevance vector ($\bar{\lambda}_B$) used in selecting the most (i.e., top 10%) relevant features.

5) Support Vector Machines: For non-separable data, Support Vector Machines (SVM) is defined as

$$\min||\beta|| \text{ subject to } \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i \forall i, \\ \xi_i \geq 0, \Sigma\xi_i \leq \text{ constant.} \end{cases} \tag{3}$$

where β is the inverse of the half length of the margin, $y_i{\in}[-1,1]$, $x_i^T\beta + \beta_0$ is the definition of a hyper-plane, and ξ_i are the slack variables to account for point on the wrong side of the margin [38]. A quadratic programming solution, through the use of Lagrange multipliers, is used to solve the convex optimization problem in (3) and is rewritten as

$$\min_{\beta,\beta_0} \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}\xi_i, \tag{4}$$

where,

$$\xi_i \geq 0, \tag{5}$$

$$y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \ \forall i, \tag{6}$$

and C is the "cost" parameter used to tune the function, which was set to 1 for all results presented in Section VI. The primal Lagrange function is given as

$$\begin{aligned} L_P = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i \\ - C\sum_{i=1}^{N}\alpha_i\left[y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)\right], \end{aligned} \tag{7}$$

and is minimized with respect to β, β_0, and α_i, which is a scaling factor determined by the Lagrange multipliers. The result of (7) is the dual form of the Lagrange function, which is given by

$$L_D = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_i^T x_j, \tag{8}$$

and serves as the lower bound of (3). The dual form is constrained by and satisfies the Karush-Kuhn-Tucker (KKT) conditions. This form is maximized based upon $0{\leq}\alpha_i{\leq}C$ and

$$\sum_{i=1}^{N}\alpha_i y_i = 0.$$

The characteristic solutions to the primal (7) and dual (8) Lagrangian functions are achieved through the minimization of β, β_0, and α_i and are given by

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i, \tag{9}$$

where observations of $\hat{\beta}$ are the support vectors. Support vectors at the edge of the margin have value $\hat{\xi}_i{=}0$ and $0{\leq}\hat{\alpha}_i{\leq}C$. The remaining support vectors have values of $\hat{\xi}_i{>}0$ and $\hat{\alpha}_i{=}C$. Based upon these and the KKT conditions, $\hat{\beta}_0$ can be calculated and the decision function expressed as

$$\hat{S}(x) = \text{sign}\left[x^T\beta + \hat{\beta}_0\right], \tag{10}$$

where the sign[...] notation indicates a binary decision function that determines the location of a point with respect to the margin and assigns a -1 or 1 to that point. The SVM classifier searches for a linear boundary in the feature space of the given input. The process can be broadened in application by mapping the points to a larger feature space using kernels. The input features are represented as

$$h(x_i) = \{h_1(x_i), h_2(x_i), \ldots, h_M(x_i)\}\,,\ i = 1, \ldots, N,$$

and due to non-separable data provides the nonlinear function

$$\hat{f}(x) = h(x)^T\beta + \beta_0, \tag{11}$$

where class assignment is performed by (10). The input $h(x)$ is transformed into a higher feature space using the general kernel function given by

$$K(x, x') = \langle h(x), h(x')\rangle\,, \tag{12}$$

where K is the Radial Basis Function (RBF) kernel. The RBF kernel is given by

$$K(x, x') = \exp\left(-\psi||x - x'||^2\right), \tag{13}$$

where ψ is a positive constant [38].

VI. Experimentation and Evaluation

First we consider the trust mechanism, then ID verification.

A. Trust Mechanism

The trust mechanism includes four factors and covers the current and past behavior for devices in a way that reduces the potential for threats. Following is the approach to updating the trust for each device in detail. The first function below represents the devices' participation and their current behaviors among other devices. $F(C_t)$ is defined by the following algorithm for each device:

a) Find the mean (μ) value for all the trust values and the standard deviation (σ).

b) Using the empirical rule [39] to calculate the interval that covers at a maximum 99.7% of the devices and builds the categories; for instance: $\mu \pm \sigma$ is approximately 68% of the measurements, $\mu \pm 2\sigma$ is approximately 95%, and $\mu \pm 3\sigma$ is approximately 99.7% of the measurements, which will lead us to create five categories around the mean to measure the current behavior among other peers. Thus, the update to the current trust value is as follows:

- The "Average" category has impact factor of 0.05; values within the interval $[\mu \pm \sigma]$ are in the majority, which represents 68% of the devices in the system.
- The "Above average" category has impact factor of 0.07; values are within the interval $((\mu+\sigma)$,$(\mu+2\sigma)]$.
- The "Best" category has impact factor equal to 0.09; values are greater than $(\mu + 2\sigma)$.
- The "Below average" category has impact factor of 0.03; values are within the interval $[(\mu-2\sigma)$, $(\mu-\sigma)]$.
- The "Worst" category has impact factor of 0.01; values are less than $(\mu - 2\sigma)$.

c) Determine the device category based on the five categories above and calculate the corresponding equation of that category with the device's current trust value using the following function:

$$F(C_t) = \begin{cases} C_t \geq (\mu + 2\sigma) & , C_t \times 1.09 \\ (\mu + \sigma) \leq C_t < (\mu + 2\sigma) & , C_t \times 1.07 \\ (\mu - \sigma) < C_t < (\mu + \sigma) & , C_t \times 1.05 \\ (\mu - 2\sigma) < C_t \leq (\mu - \sigma) & , C_t \times 1.03 \\ C_t \leq (\mu - 2\sigma) & , C_t \times 1.01 \end{cases}$$

The second factor is based on the history of the device. In our system, the messages are ranked from 1 to 10, which reflects how critical the messages being shared by this specific device during its lifetime are. So, the number of messages of each rank are stored as a list for each device, which results in calculating the number of points M_{points} as follows:

$$M_{points(id)} = \sum_{i=1}^{m}(\text{Message}_{freq(rank)} \times \text{rank}), \qquad rank \in \{1 - 10\}. \tag{14}$$

This value will be calculated and later will be compared to the maximum value achieved among all devices using g(M_{points}):

$$g(M_{points(id)}) = \frac{M_{points(id)}}{Max(M_{points})}. \tag{15}$$

The third factor, which is the time it took the device to build the messages, is evaluated using $L(lt_{id})$, which is the ratio of the time that a device spent to build its message history to the amount of time on the system since the last update (uptime):

$$L(lt_{id}) = \frac{lt_{id}}{TotalTime}. \tag{16}$$

The fourth factor is 40% of the updated trust. It is computed based on the following function, where B is the fraction of bad messages, *count* is the number of times the device behaves maliciously, and *Reset* is the flag value to reset the total trust value to zero if *count* exceeds 3 or B is greater than 30%:

$$P(B) = \begin{cases} 0.4 \times [1 - (count \times B)] & , count \leq 3, B \leq 0.3 \\ Reset = 1 & , count > 3, B > 0.3 \end{cases}$$

The final trust value will be the sum of all four factors if the flag *Reset* is zero, where each is multiplied with its corresponding weight; thus, the updated value for each device in the system participating in that round will be calculated as

$$Trust(Reset) = \begin{cases} Reset = 0 & , T_{new} \\ Reset = 1 & , T_{new} = 0 \end{cases}$$

where T_{new} is

$$\begin{aligned} T_{new} = F(C_t) \times \theta + \\ g(M_{points(id)}) \times \chi + \\ L(lt_{id}) \times \phi + \\ P(B) \times \tau. \end{aligned} \tag{17}$$

This final formula reflects the punishment and reward in the new trust based on the current and the past behavior of the device, considering the quality and the quantity of messages during its lifetime, the fraction of bad messages, and how many times the device repeated that behavior. The trust value does not increase or decrease independently for each device; rather, the formula rebuilds the new trust for each device based on its behavior in comparison with other devices' behaviors (current and past). All devices start their trust at 50%. Thus, each device's trust in the next update will be nearly the same if they were active to the same degree and communicated honestly with the same number of messages and no bad messages (see Figure 5). The weight for participation and current behavior in comparison with one another in the formula is the same for all given their common initial trust. Differences arise from other factors such as the active uptime for each, the number of generated messages, and the fraction of bad messages produced by each. In Figure 5, we assume that all devices are honest, where the fraction of bad messages is zero. It can be seen that on the third update the trust values for all three devices decreased, while device B's trust value decreased less than that of the others because of B's higher percentage of active uptime. This is because our algorithm values total uptime of a given device when assigning updated trust over time.

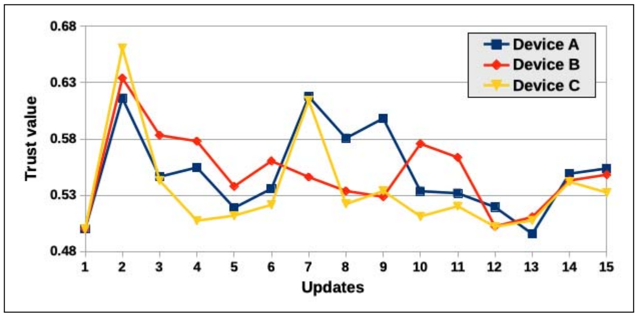

Figure 5: Devices Trust Updates

The system allows up to 30% of messages to be bad with negligible increase or decrease of the total updated trust value. However, this allowance has a short duration: this percentage seeks to address possible temporary malfunctions rather than malicious behavior. The device could have a percentage of 30% or less for three times the maximum, then the system resets its trust value to zero as punishment. In addition, the fraction of bad messages is multiplied by the number of times the device repeats this behavior; thus, the trust value decreases quickly. In Figure 6—which shows the updated trust curves for one device with different percentages of bad messages—one can see that it takes the system only three consecutive or nonconsecutive updates to drop the trust value of an device by a maximum a 30% of bad messages, while it takes one trust update to drop its trust when the percentage of bad messages is higher than 30%.

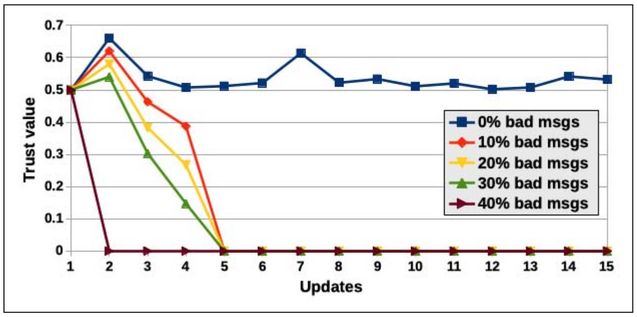

Figure 6: Percentage of bad messages allowed in the system

If a device starts with honest behavior to build up its trust value and then it starts manipulating the system by sharing bad messages, the trust mechanism resets its trust to zero after three trust updates in the worst case, or after one trust update in the best case; both reactions reduce the impact of the breakout fraud threat (see Figure 7).

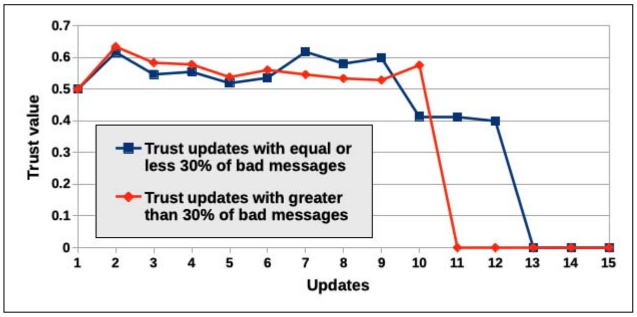

Figure 7: Malicious behavior to enable breakout fraud

B. SVM-based ID Verification Results

Similarly to the work in [26], [27], [40]–[43], transmitter identity (ID) verification is leveraged here as a means of authenticating a given IoT transmitter's *claimed* digital ID (e.g., MAC, IMSI) using the reference model generated from the RF-DNA fingerprints of the actual, trusted transmitters. As in [25], ID verification serves two purposes: (1) *verification* of an authorized IoT transmitter's ID for the purpose of granting the user network access and (2) *rejection* of a rogue IoT transmitter to deny the corresponding user access to the network. There are four possible outcomes associated with IoT transmitter ID verification:

1) *True Verification*: A trusted transmitter's ID is correctly deemed to be authentic and granted network access.
2) *True Rejection*: An impersonating transmitter is correctly deemed to be a rogue and denied network access.

Table I: Verification Outcomes [25].

	System Declaration	
Actual ID	Authentic	Rogue
Authentic	True Verification (TVR)	False Reject (FRR)
Rogue	False Verification (FVR)	True Reject (TRR)

Table II: SVM ID verification results for N_t=6 trusted and N_r=12 rogue transmitters at SNR=9 dB.

		SVM Declaration				SVM Declaration	
Claimed Digital ID	Actual ID	Verified	Rejected	Claimed Digital ID	Actual ID	Verified	Rejected
MS63A7	MS63A7	99%	1%	MS6373	MS6373	84%	16%
	Rogue	0%	100%		Rogue	2%	98%
MS63A9	MS63A9	91%	9%	MS6387	MS6387	93%	7%
	Rogue	0%	100%		Rogue	9%	91%
MS66E7	MS66E7	84%	16%	MSD905	MSD905	100%	0%
	Rogue	8%	92%		Rogue	88%	12%

Table III: SVM ID verification results for MSD905.

		SVM Declaration	
SNR (dB)	Actual ID	Verified	Rejected
9	MSD905	100%	0%
	Rogue	88%	12%
18	MSD905	100%	0%
	Rogue	85%	15%
27	MSD905	100%	0%
	Rogue	84%	16%

3) *False Verification*: An impersonating transmitter is incorrectly deemed to be a trusted transmitter and granted network access.
4) *False Rejection*: A trusted transmitter's ID is incorrectly deemed to be rogue and denied network access.

The last two outcomes constitute an error being made by the ID verification algorithm, which in this work is the SVM. A summary of these outcomes is presented in Table I. All SVM-based ID verification results presented hereafter conform to the presentation of the outcomes shown in Table I.

The ID verification results presented here are generated using SVM reference models that are developed using Monte Carlo simulation with N_z=10 independent and like-filtered Additive White Gaussian Noise (AWGN) realizations, k=5-fold cross-validation, and N_b=900 dimensionally reduced RF-DNA fingerprints extracted from independent WiMAX near-transient responses for each of the trusted transmitters as described in Sect. V-C and [25]. A total of N_D=18 WiMAX transmitters are used in this study with digital IDs MS63A7, MS63A9, MS66E7, MS6373, MS6387, MSD905, MS637D, MS9993, MSDAB9, MSDAC9, MSDADB, MSC2FF, MSDAC5, MSDDC7, MSDF5B, MSDF7D, and MSDF65. As in [25], N_t=6 transmitters are designated as the trusted transmitters: MS63A7, MS63A9, MS66E7, MS6373, MS6387, and MSD905. The remaining N_r=12 transmitters serve as the rogues that present false digital credentials in an attempt to gain unauthorized network access, thus representing 72 possible, unique, digital ID spoofing attacks. Different than the approaches presented in [25]–[27], the SVM is trained using the RF-DNA fingerprints for all trusted WiMAX transmitters, where class one represents the transmitter whose digital ID is to be verified and class two represents the remaining trusted transmitters. In this approach, the remaining trusted transmitters serve as a representative sample of the overall population.

Table II presents the SVM-based ID verification results for the N_t=6 trusted and N_r=12 rogue WiMAX transmitters at SNR=9 dB. The rogue transmitter results are the average ID verification performance across $N_B \times N_z \times N_r$=12,000 independent, rogue transmissions. At SNR=9 dB, the ID of every trusted transmitter is verified at a rate of 84% or higher. Specifically, the identities of MS66E7 and MS6373 are verified at rates of 84%, while all others have ID verification rates of 91% or better. Rogue transmitters are rejected at a rate of 91% or higher when spoofing the digital credentials of the trusted transmitters, MS63A7, MS63A9, MS66E7, MS6373, and MS6387. However, for the case of rogue transmitters spoofing the digital ID of MSD905, the rejection rate is 12% at SNR=9 dB. Thus, the rogue transmitters would be granted network access at a false verification rate of 88%. For MSD905, ID verification is also performed at SNR=[18,27] dB and the results presented in Table III. These results show that the ability to effectively reject rogue transmitters that spoof the digital ID of MSD905 is independent of the SNR. This independence indicates that the issue is either with the SVM settings, the RF-DNA fingerprints of the trusted transmitters not being sufficiently distinct to facilitate 90% or better rogue transmitter rejection, or a combination thereof.

VII. How our approach mitigates the threat model

Here are our impacts on the threats (see Section IV):

Counterfeiting/Impersonation: Strong identity achieved through RF-DNA Fingerprinting helps prevent these threats.

Breakout Fraud: Adjusting trust dynamically means that trust will decrease rapidly in the system if behavior degrades. This effect indirectly accounts for device takeover, which could produce such a change after a long period of compliance.

Bad Mouthing: The blockchain manages the trust values to prohibit reporting of incorrect trust values. Verification is first done by peers and the data is shared on the global blockchain so that the device can change domains while maintaining its trust from its previous domain.

VIII. Conclusion and Future work

With regard to our results for RF-DNA fingerprinting, the selected SVM settings may be contributing to the poor rogue rejection performance when the digital ID of MSD905 is spoofed. Future work will investigate the impact of some of the SVM settings. The cost parameter C was set to 1, but this value may not be the best for rejection of rogue transmitters. The results presented in Table II and Table III are based on a binary decision. Thus, the decision is absolute and does not consider how much a given RF-DNA fingerprint resembles the assigned class, trusted versus rogue. One approach would be to consider a measure of similarity such as probability to aid in distinguishing rogue from trusted transmitters. Lastly, this work used the dimensionally reduced RF-DNA fingerprint data set from [25], which demonstrated improved rogue transmitter rejection performance when the number of fingerprint features was reduced from 204 to 20. For the case of the SVM classifier, this reduction may be inhibiting its ability to distinguish rogue from trusted transmitters.

Our architecture reduces the potential for counterfeiting by providing strong identity; it also reduces the possibility of bad-mouthing and breakout fraud by providing trust that varies over time as a function of quality/ quality of the data provided. While we may choose to update specific parameters for the trust algorithm in the future, this approach already produces useful feedback as a function of entity behavior. Crucial to the dissemination of trust is the highly available, high integrity of the blockchain ledgers utilized. Our system successfully uses a hardware-software codesign approach to delivering a secure and trustworthy IoT infrastructure.

Acknowledgement

The authors acknowledge support from the University of Tennessee at Chattanooga. Research reported in this publication was supported by the 2019 Center of Excellence for Applied Computational Science and Engineering grant competition (CEACSE). This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. 1821926. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Rawlinson, K. HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack, Jul. 2014.

[2] Top 10 Network Security Threats, *Government Technology*, Sep 2010.

[3] Toonstra J. and W. Kinsnew. Transient Analysis and Genetic Algorithms for Classification. In *Proc of the IEEE Conf on Communications, Power and Computing (WESCANEX95)*, May 1995.

[4] Ureten O. and N. Serinken. Detection of Radio Transmitter Turn-On Transients. *IEE Electronics Letters*, 35(23), Nov 1999.

[5] Hall, J., M. Barbeau, and E. Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *Communications, Internet, and Information Technology*, pages 201–206, 2004.

[6] Jana S. and S. Kasera. Wireless Device Identification with Radiometric Signatures. In *Proc of the ACM 14th Int'l Conf on Mobile Computing and Networking (MOBICOM08)*, Sep 2008.

[7] Zhao, C., L. Huang, L. Hu, and Y. Yao. Transient fingerprint feature extraction for wlan cards based on polynomial fitting. In *2011 6th International Conference on Computer Science Education (ICCSE)*, pages 1099–1102, Aug 2011.

[8] Fadul, M., D. Reising, T. Loveless and A. Ofoli. RF-DNA Fingerprint Classification of OFDM Signals Using a Rayleigh Fading Channel Model. In *Wireless Communications and Networking Conference (WCNC)*, Accepted 2019.

[9] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[10] Dr Gavin Wood. Ethereum: a secure decentralised generalised transaction ledger. 2016.

[11] Alex Mizrahi Meni Rosenfeld Iddo Bentov, Charles Lee. Proof of activity: Extending bitcoins proof of work via proof of stake. 2014.

[12] G. Zyskind, O. Nathan, and A. '. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Security and Privacy Workshops*, pages 180–184, May 2015.

[13] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the Internet of Things.

[14] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 164–173, May 1996.

[15] Li Xiong and Ling Liu. Building trust in decentralized peer-to-peer electronic communities. 2002.

[16] Li Xiong and Ling Liu. Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, July 2004.

[17] C. A. Kerrache, C. T. Calafate, J. Cano, N. Lagraa, and P. Manzoni. Trust management for vehicular networks: An adversary-oriented overview. *IEEE Access*, 4:9293–9307, 2016.

[18] W. Li, H. Song, and F. Zeng. Policy-based secure and trustworthy sensing for Internet of Things in smart cities. *IEEE Internet of Things Journal*, 5(2):716–723, April 2018.

[19] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung. Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet of Things Journal*, pages 1–1, 2018.

[20] J. Jiang, G. Han, F. Wang, L. Shu, and M. Guizani. An efficient distributed trust model for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(5):1228–1237, May 2015.

[21] Jiangwen Wan, Xiang Zhou, Xiaofeng Xu, and Renjian Feng. A trust evaluation algorithm for wireless sensor networks based on node behaviors and d-s evidence theory. 2011.

[22] G. Zhan, W. Shi, and J. Deng. Design and implementation of TARF: A trust-aware routing framework for wsns. *IEEE Transactions on Dependable and Secure Computing*, 9(2):184–197, March 2012.

[23] I. Butun, S. D. Morgera, and R. Sankar. A survey of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys Tutorials*, 16(1):266–282, First 2014.

[24] Shiho Kim. Chapter two - blockchain for a trust network among intelligent vehicles. 111:43 – 68, 2018.

[25] Reising, D., M. Temple and J. Jackson. Authorized and Rogue Device Discrimination Using Dimensionally Reduced RF-DNA Fingerprints. *IEEE Trans Inf. Forens. Security*, Vol. 10, No. 6, Jun 2015.

[26] Rehman, S., K. Sowerby, and C. Coghill. Analysis of impersonation attacks on systems using RF fingerprinting and low-end receivers. *Journal of Computer and System Sciences*, 80:591601, 05 2014.

[27] Kroon, B., S. Bergin, I. Kennedy, and G. O'Mahony Zamora, Georgina. Steady State RF Fingerprinting for Identity Verification: One Class Classifier versus Customized Ensemble. In *Artificial Intelligence and Cognitive Science*, pages 198–206, 2010.

[28] Mark M Tehranipoor, Ujjwal Guin, and Swarup Bhunia. Invasion of the hardware snatchers. *IEEE Spectrum*, 54(5):36–41, 2017.

[29] Mark (Mohammad) Tehranipoor, Ujjwal Guin, and Domenic Forte. *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer, 2015.

[30] U. Guin, Ke Huang, D. DiMase, J.M. Carulli, M. Tehranipoor, and Y. Makris. Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain. *Proceedings of the IEEE*, 102(8):1207–1228, Aug 2014.

[31] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead. *Journal of Electronic Testing*, 30(1):9–23, 2014.

[32] Ujjwal Guin, Pinchen Cui, and Anthony Skjellum. Ensuring proof-of-authenticity of iot edge devices using blockchain technology. *IEEE International Conference on Blockchain*, 2018.

[33] Tuhin Borgohain, Uday Kumar, and Sugata Sanyal. Survey of security and privacy issues of Internet of things. *arXiv preprint arXiv:1501.02211*, 2015.

[34] Hongmei He, Carsten Maple, Tim Watson, Ashutosh Tiwari, Jörn Mehnen, Yaochu Jin, and Bogdan Gabrys. The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 1015–1021. IEEE, 2016.

[35] Reising, D., M. Temple and M. Oxley. Gabor-based RF-DNA fingerprinting for classifying 802.16e WiMAX Mobile Subscribers. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 7–13, Jan 2012.

[36] Reising D., M. Temple and M. Mendenhall. Improved Wireless Security for GMSK-Based Devices Using RF Fingerprinting. *Int'l J. Electronic Security & Digital Forensics*, vol. 3, no. 1, 2010.

[37] Klein, R., M. Temple, M. Mendenhall, and D. Reising. Sensitivity analysis of burst detection and RF fingerprinting classification performance. In *2009 IEEE International Conference on Communications*, pages 1–5, June 2009.

[38] Hastie, T., R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition, 2017.

[39] R Lyman Ott and Micheal T Longnecker. *An Introduction to Statistical Methods and Data Analysis*. Cengage Learning, 2015.

[40] Jain A., A. Ross and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Trans on Circuits and Systems for Video Technology*, 14(1), 2004.

[41] Danev B., H. Luecken, S. Capkun and K. El Defrawy. Attacks on Physical-layer Identification. In *Proc of the 3rd ACM Int'l Conf on Wireless Network Security (WiSec10)*, Mar 2010.

[42] Cobb W., E. Laspe, R. Baldwin, M. Temple and Y. Kim. Intrinsic Physical Layer Authentication of ICs. *IEEE Trans on Information Forensics and Security*, 2(4):7, Dec 2011.

[43] Dubendorfer C., B. Ramsey and M. Temple. An RF-DNA Verification Process for ZigBee Networks. In *Proc of 2012 IEEE Military Comm Conf (MILCOM12)*, Oct 2012.