



Security framework for smart cyber infrastructure

Shalaka Satam¹ · Pratik Satam¹ · Jesus Pacheco² · Salim Hariri¹

Received: 9 November 2021 / Revised: 9 November 2021 / Accepted: 13 November 2021 / Published online: 30 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The rapid deployment of the Internet of Things (IoT) devices have led to the development of innovative information services, unavailable a few years ago. To provide these services, IoT devices connect and communicate using networks like Bluetooth, Wi-Fi, and Ethernet. This full-stack connection of the IoT devices has introduced a grand security challenge. This paper presents an IoT security framework to protect smart infrastructures from cyber attacks. This IoT security framework is applied to Bluetooth protocol and IoT sensors networks. For the Bluetooth protocol, the intrusion detection system (IDS) uses n-grams to extract temporal and spatial features of Bluetooth communication. The Bluetooth IDS has a precision of 99.6% and a recall of 99.6% using classification technique like Ripper algorithm and Decision Tree (C4.5). We also used AdaBoost, support vector machine (SVM), Naive Bayes, and Bagging algorithm for intrusion detection. The Sensor IDS uses discrete wavelet transform (DWT) to extract spatial and temporal features characteristics of the observed signal. Using the detailed coefficients of Biorthogonal DWT, Daubechies DWT, Coiflets DWT, Discrete Meyer DWT, Reverse Biorthogonal DWT, Symlets DWT, we present the results for detecting attacks with One-Class SVM, Local Outlier Factor, and Elliptic Envelope. The attacks used in our evaluation include Denial of Service Attacks, Impersonation Attacks, Random Signal Attacks, and Replay Attacks on temperature sensors. The One-Class SVM performed the best when compared with the results of other machine learning techniques.

Keywords Internet of Things (IoT) · Threat model · Smart infrastructure · Bluetooth security · Anomaly behavior analysis · Intrusion detection

1 Introduction

Advances in mobile and pervasive computing, social network technologies, and the exponential growth in Internet applications and services have led to the development of the next generation of Internet services (Internet of Things

(IoT)) that are pervasive, ubiquitous, and touch all aspects of our life. These IoT services are the key enabling technologies that are revolutionizing business, healthcare, critical services, and Industrial Infrastructure. These advances have driven the integration of many disconnected networks. But this integration of traditionally disconnected systems is providing attackers with new attack surfaces to exploit and thus creating new security challenges.

Today, an attacker targets these IoT networks, through the Internet or through an insider. Cyber attacks like Stuxnet [1] have demonstrated the devastating impact such attacks can have on IoT systems. In such interconnected IoT systems, the attacker can exploit the communication networks used by IoT devices. Satam et.al in [2, 3] show numerous attacks on Wi-Fi networks. Similarly, the attackers can exploit vulnerability in Ethernet networks [4], DNS [5], HTML [6] to perform successful cyber attacks. The attackers can also use cloud-based services to perform cyber attacks on IoT devices [7]. The attackers can use a malicious insider to damage IoT sensors and perform

✉ Shalaka Satam
shalakasatam@email.arizona.edu

Pratik Satam
pratiksatham@email.arizona.edu

Jesus Pacheco
jesus.pacheco@unison.mx

Salim Hariri
hariri@ece.arizona.edu

¹ Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85719, USA

² Department of Industrial Engineering, Universidad de Sonora, Hermosillo 83000, Sonora, Mexico

attacks like Denial of Service (DoS) attacks, sensor impersonation attacks, random signal attacks, and replay attacks on IoT systems.

Thus, there is a need for a framework to secure IoT devices. Satam et al. in [7] and Pacheco et al. in [8] present a framework to secure smart cyberinfrastructures. This paper extends this framework and validates the framework by performing more rigorous experimentations. This paper makes the following contributions:

- Presents a novel framework for securing Internet of Thing and smart infrastructure.
- Applies the presented security framework for End-Nodes layer security (Sensor security) and Network layer security (Bluetooth Security).
- Presents a novel approach to detect attacks on sensors using different discrete wavelet transforms (DWTs) over a sample window.
- Presents an approach to detect attacks on Bluetooth protocol using $n - grams$ for obtaining temporal and spatial behavior of the protocol.

The rest of the paper is organized as follows: Sect. 2 presents the related work, Sect. 3 presents the security framework for IoT and smart cyber infrastructure. Section 4 applies the security framework presented in Sect. 3 to End-Nodes layer (sensors) and Network layer (Bluetooth Protocol), and Sect. 5 presents the Conclusion.

2 Related work

2.1 Smart infrastructures, cloud and Fog Computing, and IoT

Smart infrastructures (SI) integrate autonomy and adaptive control linking industrial controllers, sensors, and actuators with cloud services allowing sustainable development, intelligent services while reducing developmental and operational costs. SI are critical to address modern environmental concerns while improving human comfort and lifestyle [9, 10]. Cloud Computing provides computing, storage, and applications as services that are offered on demand in a cost-effective and a scalable way. Resources can be shared among a large number of users, who can access applications and data from anywhere at any time [11]. On the other hand, Fog Computing aims at providing computational power, storage, and network services to end devices. Since Fog Computing host services at the network edge, its advantages include low service latency, high quality services, support for mobility, location awareness, and easier implementation of security mechanisms. It has been shown that Fog Computing can be effective in supporting IoT applications that demand predictable latency

[12]. IoT is a ubiquitous network of smart heterogeneous devices allowing data collection and ability to provide better and personalized services to the users [7]. IoT devices consist of intelligent end-to-end systems that provide smart solutions, including sensing, communications, and networking. This widespread and diverse application of IoT systems brings with it challenges with respect to security [7].

2.2 Intrusion detection systems

Intrusion detection systems can be classified into four major types [2]. They are Signature based IDS, Anomaly based IDS, Specification based IDS and Hybrid IDS. Signature based IDS detects attacks by using a database containing known attack signatures. Signature IDS are easy to develop but they require regular updates to include new attack signatures [13]. Also, signature IDS are unable to detect new attacks that are known as zero-day attacks. The anomaly based IDS [2, 3, 5] is built by developing normal behavior model of the system so that any attack on the system will be detected because it produces an abnormal operation. Anomaly IDS can detect new and modified attacks. However, they generate high false alarms due to the difficulty in modeling the normal operations. Specification based IDS [14, 15] uses a set of rules and thresholds that define the expected behavior for different network components through specification features provided by a human expert. The fourth IDS type is a hybrid IDS [16], which is a combination of Signature-based, Anomaly-based, and Specification-based IDS.

2.3 Bluetooth protocol

Bluetooth (IEEE 802.15.1) is a short-range communication protocol with a communication range of up to a hundred meters. The Bluetooth protocol operates in the ISM band of 2.4 GHz. Bluetooth protocol communicates using frequency hopping spread spectrum (FHSS). The Bluetooth communication hops between seventy-nine different communication channels of one megahertz (1 MHz) each at the rate of 1600 times per second, minimizing transmission interference and increasing transmission security. Two Bluetooth devices while communicating with each other exchange packets (smallest data unit) consisting of access code, packet header, and payload. When two Bluetooth devices are in communication range, a device initiates the communication by sending an ID Packet. The receiver, if interested in setting up the communication, responds with a frequency hop synchronization (FHS) packet. The device initiating the communication is called the master device. The devices communicating with the master device are called slave devices. A master can connect up to seven

other slave devices to form an ad-hoc communication network called the Piconet. Each Piconet has its unique timing clock and frequency hopping sequence, ensuring no overlap with other Piconets operating in the communication range. The master in a Piconet can be a slave in another Piconet. This ad-hoc network formed by the master connecting two Piconets is called a Scatternet [17], as shown in Fig. 1. Bluetooth devices are of two types: (1) Basic Rate/Enhanced Data Rate (BR/EDR) device called classic Bluetooth device, and (2) Bluetooth Low Energy (LE) device also called Smart Bluetooth device. Classic Bluetooth devices cannot communicate with Bluetooth Low Energy (BLE) devices unless they have both chipsets. The main components of the Bluetooth Architecture are (1) Bluetooth Controller, (2) Host Controller Interface (HCI) Transport Layer, and (3) Bluetooth Host.

- *Bluetooth Controller* The Bluetooth Controller implements the Bluetooth protocol's physical layer and consists of the Link Manager layer, the Baseband layer, and the radio layer. The Bluetooth Controller establishes the communication with the destination through the Host Controller Interface (HCI). The Link manager layer initiates the communication link between two Bluetooth devices while the Baseband layer controls access to the radio link through the Radio Layer.
- *Host Controller Interface (HCI)* The HCI layer acts as a liaison between the Bluetooth host and the Bluetooth Controller. The Bluetooth Host communicates with the Bluetooth Controller through the HCI.
- *Bluetooth Host* The Bluetooth Host consists of the Bluetooth protocol stack's logical components, allowing applications to use the Bluetooth protocol.

3 Security framework for IoT and smart cyber infrastructure

In this section, we present a security framework that will be used to effectively model the behavior of any smart infrastructure and secure the services and applications that will run on the IoT and smart infrastructures. The security

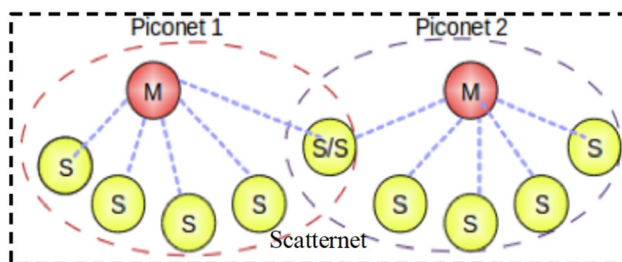


Fig. 1 Bluetooth scatternet

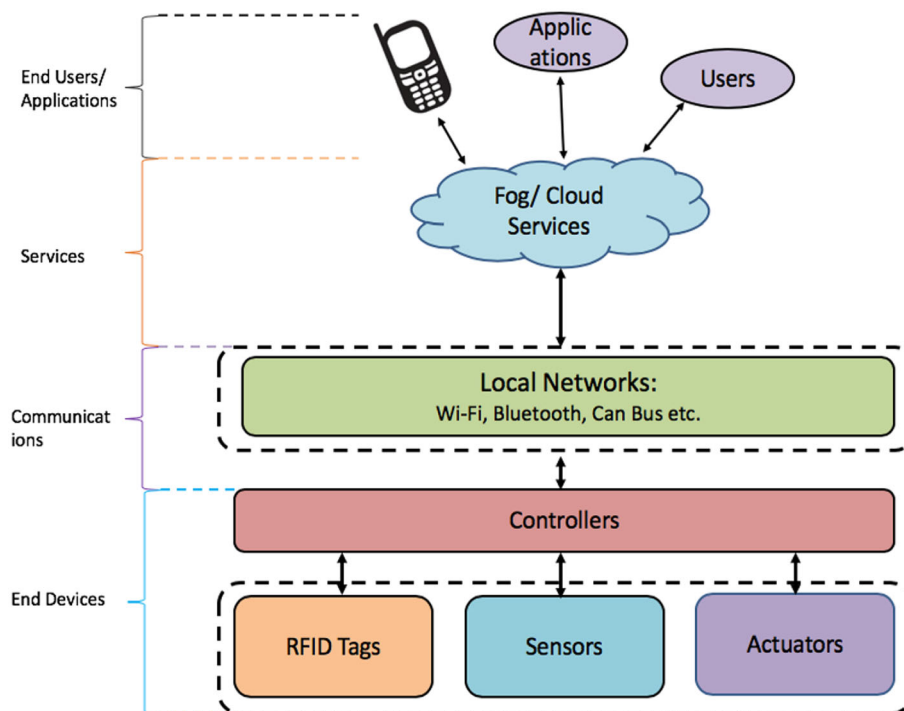
framework can be used to model the operations of the IoT devices and in understanding the behavior of the IoT devices.

3.1 IoT security framework

Several IoT frameworks can be used to create a threat model and apply mitigation strategies [7, 8, 18–20]. Figure 2 shows an architecture that can be used to guide the security development of IoT smart infrastructures. The framework consists of four layers: IoT end Nodes (end devices), Network, Services and Applications. Cyberattacks can be launched against the functions and services provided by each layer as shown in Fig. 2. For each layer in our framework we can define the threats in terms of target, impact, and mitigation methods.

- *End nodes layer* The first layer (end nodes), passes information through physical devices that represent the physical world. The information includes object properties, environmental conditions, data, etc. The key components in this layer are the sensors for capturing the current state of physical device, and the actuators to modify and drive the physical environment to a desired operational state. The attack targets at this level are local controllers, sensors, actuators, and information. The impact can be energy waste, monetary cost, human safety, provider's reputation, and waste of time. Mitigation mechanisms include lightweight encryption, sensor authentication, IDS, anti-jamming, and behavior analysis.
- *Network layer* Network layer is responsible for the reliable transmission of information from/to end nodes [21]. The technologies used in this layer include the Internet, mobile communication networks, wireless sensor networks, network infrastructures, and communication protocols. Network security and management play an important role to defend against cyber-attacks targeting firewalls, routers, protocols, and personal information. The impacts might be monetary, reputation, safety, energy, control, and time. Network mitigation mechanisms include authentication, anti DoS, encryption, packet filtering, congestion control, anti-jamming, intrusion detection, and behavior analysis.
- *Service layer* The services layer acts as an interface between the application layer in the top level and the network layer in the lower level [22]. At this layer, all the required computational power is mostly provided as a cloud service. In this layer, cyberattacks target personal and confidential information, IoT end devices, monitor and control functions. The impact includes people safety, money losses, and important information leakage. Protection mechanisms include encryption,

Fig. 2 Internet of Thing security framework



authentication, session identifiers, intrusion detection, selective disclosure, data distortion, and behavior analysis.

- *Applications layer* The application layer provides programmable services according to the needs of the user [21]. The access to the IoT services is through this layer and it can be via mobile technology such as cellphone, mobile applications, or a smart appliance. In this layer, data sharing is an important characteristic and consequently application security must address data privacy, access control and information leaks. The impacts are stolen intellectual properties, disclosure of critical business plans, money loss, and damaging business reputation. Some mitigation mechanisms include encryption, authentication, and anomaly behavior analysis of applications and their services.

Attackers may use any exploitable vulnerability to gain access to the system and launch an attack. Our framework can be used to identify the existing vulnerabilities and the appropriate mitigation mechanism. For instance, an IP temperature sensor located in a remote place can be easily replaced by a computer to obtain illegal information and to launch an attack (e.g. replay attack). Since sensors usually have low (or no) computational power, it is unrealistic to apply encryption techniques, a more suitable approach is to authenticate each sensor and its data.

4 Application of IoT security framework

In this section, we apply the IoT Security Framework presented in Sect. 3 to secure end nodes by applying anomaly behavior analysis on sensors, that are experiencing attacks on their Bluetooth network.

4.1 Sensor anomaly behavior analysis

In the smart infrastructure shown in Fig. 3, the observer nodes collect sensor data, actuator information, IP addresses, network packets sent and received. While implementing continuous monitoring, the observer node creates the *Device – DNA* data structure, to characterize the devices normal operations. The observer uses *Device – DNA*, to detect malicious device activity using machine learning models. The observer node performs

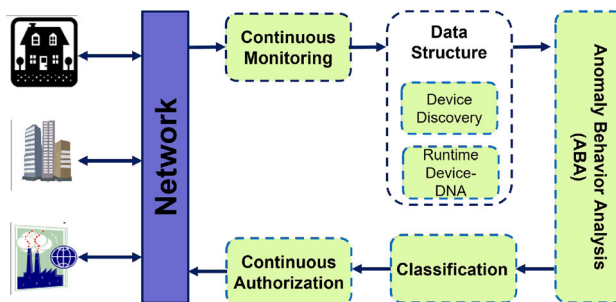


Fig. 3 Smart device observer continuous monitoring and continuous authorization architecture

continuous authorization by enforcing security policies and secure communications. We use DWT [23–28] to characterize temporal and the spatial behavior of the smart devices. For this paper, the IoT security framework from Sect. 3 is applied to sensor nodes.

4.1.1 Continuous monitoring for sensors

Figure 3 shows the IoT sensor nodes. The continuous monitoring for the sensor nodes consists of the following:

- *Sniffer* The sniffer *sniffs* the raw data from the sensor nodes as shown in Table 1.
- *Anomaly Behavior Analysis Module* The Anomaly Behavior Analysis Module performs data preprocessing and transforms the raw features from Table 1 to processed features in Table 2 for a selected window of t samples. Sensor attacks are detected using pretrained machine learning models. We perform DWT on the sensor data to obtain temporal and spatial features. DWT has two types of coefficients which are obtained from wavelet decomposition, they are Approximation coefficients denoted by cA and Detail coefficients denoted by cD. The cA obtain low frequency information and the detailed coefficients (cD) obtains high frequency information. We use cD coefficients in our analysis. For the collected sensor data, we use a window of size 5 to read the sensor data to find the coefficient mean and variance to generate a table on which we perform machine learning classification and analysis. We use different types of wavelets such as Biorthogonal DWT, Daubechies DWT, Coiflets DWT, Discrete Meyer DWT, Reverse Biorthogonal DWT and Symlets DWT. Biorthogonal DWT is a wavelet whose associated wavelet transform is invertible, but it is not necessarily orthogonal. Biorthogonal wavelets allow more degrees of freedom as compared to orthogonal wavelets. Daubechies wavelet is a family of orthogonal wavelets which defines DWT and is characterized by maximum number of vanishing moments. Coiflets wavelet function has $2N$ moments equal to 0 and

scaling function of $2N-1$ moments equal to 0. Both functions have support of $6N-1$ length. Discrete Meyer wavelet is an orthogonal wavelet which is infinitely differentiable with infinite support and is defined in frequency domain. Symlets DWT are symmetrical wavelets and they are similar to Daubechies wavelets.

4.1.2 Evaluation of sensor anomaly behavior approach

In this subsection, we evaluate the performance of detecting attacks against sensors. Table 3 shows the various machine learning classifiers we used in our evaluation. The machine learning classifiers are: *Adaboost* is a boosting algorithm that uses weak classifiers repeatedly for “ t ” rounds over a distribution D_t of the provided set D . The initial distribution $D_1(i) = 1/m$ is used for all i . Distribution $D_{(t+1)}$ from D_t is obtained by multiplying the weight of example i by some number $\beta_t \in [0, 1)$ if that round classified the event correctly or the weight is left unchanged. The iteration is rerun after normalizing the weights by dividing by the normalization factor Z_t . *Random Forest* is a classification and regression model which combines many decision trees in a randomized order and averages their predictions. *Decision Tree* uses a tree like model where a leaf node denotes the decision and the probability of an event happening. Decision trees algorithm uses conditional control statements for predictions. *SVM* or *Support Vector Machine* is used for regression and classification problems. The SVM algorithm checks for a hyperplane or line which separates the data in two classes.

Table 4 shows the various attacks used to perform the experiments. The studied attacks are as follows:

- *DoS attack* In the *Denial of Service* attack, the attacker makes the sensor inaccessible or shuts down the sensor by flooding its resources hence making it unavailable.
- *Impersonation attack* In the *Impersonation* attack, the attacker poses as a trusted person, sharing wrong sensor information.
- *Random signal attack* In *Random signal* attack, the attacker sends random sensor signals.
- *Replay attack* In the *Replay* attack, the attacker eavesdrops the sensor communication and intercepts it. Then the attacker replays the communication and consequently force the receiver to do what the attacker wants.

4.1.2.1 Experiment setup

- *Sensor attack detection* For the experimental evaluation of the continuous sensor monitoring and detection, a testbed consisting of 3 temperature sensors was setup,

Table 1 Raw features collected from sensors

| Features | Description |
|------------------|-------------------------|
| Frame epoch time | Epoch time |
| Sensor IP | Sensor’s IP address |
| Observer IP | Observer’s IP address |
| Sensor Port | Sensor’s port address |
| Observer port | Observer’s port address |
| Sensor data | Sensor’s measured data |

Table 2 Features extracted from the raw features

| Features | Description |
|-------------------------------|--|
| Biorthogonal DWT mean | Biorthogonal DWT [23] detail coefficient's mean over the selected window |
| Biorthogonal DWT var | Biorthogonal DWT detail coefficient's variance over the selected windows |
| Daubechies DWT mean | Daubechies DWT [24] detail coefficient's mean over the selected window |
| Daubechies DWT var | Daubechies DWT detail coefficient's variance over the selected windows |
| Coiflets DWT mean | Coiflets DWT [25] detail coefficient's mean over the selected window |
| Coiflets DWT var | Coiflets DWT detail coefficient's variance over the selected windows |
| Discrete Meyer DWT mean | Discrete Meyer DWT [26] detail coefficient's mean over the selected window |
| Discrete Meyer DWT var | Discrete Meyer DWT detail coefficient's variance over the selected windows |
| Reverse biorthogonal DWT mean | Reverse Biorthogonal DWT [27] detail coefficient's mean over the selected window |
| Reverse biorthogonal DWT var | Reverse Biorthogonal DWT detail coefficient's variance over the selected windows |
| Symlets DWT mean | Symlets DWT [28] detail coefficient's mean over the selected window |
| Symlets DWT var | Symlets DWT detail coefficient's variance over the selected windows |

Table 3 Sensor attack classification results

| Classifier | Precision | Recall | F-Measure |
|---------------|-----------|--------|-----------|
| AdaBoost | 0.815 | 0.685 | 0.745 |
| Random forest | 0.83 | 0.695 | 0.756 |
| Decision tree | 0.92 | 0.548 | 0.687 |
| SVM | 0.138 | 0.242 | 0.176 |

with a Linux observer monitoring the temperature sensor readings every second. The testbed also has an attacker device capable of injecting DoS attack, Impersonation attack, Random signal attack, and Replay attack. In the experimental analysis, a dataset consisting of 10800 normal samples (3 h), and 3600 attack samples for each attack was created. The raw data was processed and the DWT features were extracted with a window size of 5 samples.

4.1.2.2 Sensor detection performance evaluation In our evaluation of sensor attack detection, a AdaBoost Classifier, a Random Forest Classifier (with 100 trees), support vector machine (SVM) with a radial bias function (RBF) classifier, and a Decision Tree Classifier (C4.5) were trained. Table 3 shows the Precision, Recall, and F-Measures for these classifiers from 5-fold cross validation

and Table 4 shows the Recalls for each of the attacks. We conclude that the AdaBoost classifier gives the best performance, although its overall Precision, Recall, and F-measure [29] scores are lower than Random Forest (as seen in Table 3), as AdaBoost performs more consistently while Recalling individual attacks (as shown in Table 4). We also observe that this approach performs the worst while detecting the DoS attack for all the classifiers, hinting that more feature engineering is required to extract features that help detect DoS attacks. We use precision, recall and F-measure [29] to evaluate the results. Precision is defined as the fraction of the relevant instances among all the retrieved instances. Precision is defined by this equation: $Precision = True\ Positive / (True\ Positive + False\ Positive)$. Recall, also referred as 'sensitivity, it is the fraction of retrieved instances among all the relevant instances. Recall is defined by this equation: $Recall = True\ Positive / (True\ Positive + False\ Negative)$. F-Measure also known as F-Score is a way to combine precision and recall into a measure that captures both properties together and gives them equal weight. F-measure is defined by this equation: $F - measure = 2 * (precision * recall) / (precision + recall)$.

Table 4 Classifier recalls for each attack

| Attack | AdaBoost | Random forest | Decision tree | SVM |
|-------------------------|----------|---------------|---------------|------|
| Denial of Service (DoS) | 0.83 | 0.6 | 0.6 | 0.19 |
| Impersonation | 1 | 1 | 1 | 0.39 |
| Random signal | 0.91 | 0.97 | 0.83 | 0.57 |
| Replay | 1 | 1 | 1 | 0.21 |

4.2 Applying the IoT security framework to networking layer: bluetooth protocol

Continuous attack monitoring is performed by the Bluetooth Piconet Master that analyzes all the Bluetooth traffic flowing in the network. The abnormal behavior threshold is decided during the training phase of the system. When the detection and classification unit detects activities beyond the threshold, the Behavior Analysis Module generates an alert. As Bluetooth is a networking protocol, it is critical to analyze temporal and spatial protocol behavior while performing continuous monitoring and analysis. We use *traffic – flows* and *n – grams* to characterize the temporal and the spatial behavior of the Bluetooth protocol.

- *Traffic flow* A *traffic flow* is a continuous flow of frames or packets between a source-destination pair sampled at specific intervals of time t . The *traffic flow* characterizes the state transitions made by the networking protocol.
- *n-grams* An *n-gram* is an n -transition pattern in a specific time interval. It is sliding window of a predefined size n , over the sequence of frames or packets sampled from the *traffic flow*.

4.2.1 Anomaly analysis of the bluetooth protocol operations

Figure 4 shows the architecture for the anomaly behavior analysis methodology for the Bluetooth protocol.

- *Sniffer* The sniffer *sniffs* raw Bluetooth frames transmitted in the Bluetooth Piconet and extracts the features shown in Table 5 for each observed frame.
- *Behavior Analysis Module* The Behavior Analysis Module implements the Bluetooth Intrusion Detection System (BIDS) as in [30]. The data structures for *traffic flow* and *n-grams* are used for extracting spatial and temporal features from the raw feature set shown in

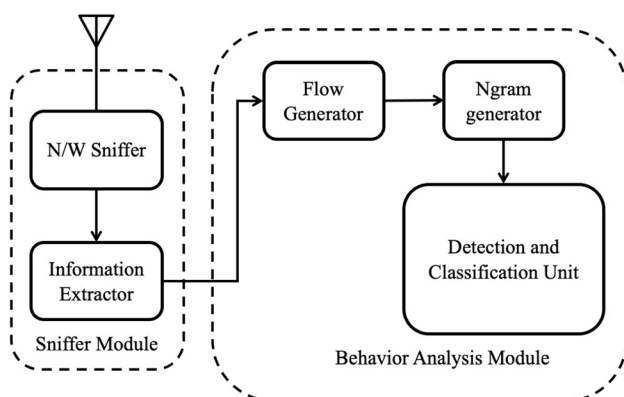


Fig. 4 Bluetooth protocol anomaly behavior analysis approach

Table 5 Features extracted from raw Bluetooth Frames

| Features | Description |
|---------------------|--|
| Frame epoch time | Epoch time |
| hci h4 type | Host Connect Interface (HCI) packet type |
| Blt hci evt code | Bluetooth HCI event code |
| Blt hci opcode | Bluetooth HCI OPCode |
| Blt L2CAP Scid | Bluetooth L2CAP protocol Source CID |
| Destination address | Destination address |
| Destination name | Destination device name |
| Destination role | Destination device role |
| Source address | Source address |
| Source name | Source device name |

Table 5. As shown in Fig. 5, the raw features from Table 5 are used to extract *traffic flows* which are used to form *n-grams*. From the *traffic flows* and *n-grams* the features shown in Table 6 are extracted. The final feature sets are probability of the flow extracted after Jelinek-Mercer smoothing [31], ratio of Host Control Interface (HCI) commands frames in the flow, ratio of ACL data frames in the flow, ratio of Synchronous Connection Oriented (SCO) data frames in the flow, and the ratio of HCI data frames in the flow. Malicious network flows are detected using a pretrained machine learning model for attack detection.

4.2.2 Device authorization in bluetooth network

In Bluetooth Protocol, we study the device authorization for a hierarchical Bluetooth network, as shown in Fig. 6. For the hierarchical Bluetooth network device authorization is implemented by white listing all the authorized Bluetooth devices. Security for the hierarchical Bluetooth network is achieved by continuous monitoring and analysis as the operations of the Bluetooth communications in order to allow only white listed devices to use the Bluetooth network.

4.2.3 Experiments for bluetooth anomaly behavior analysis

In this subsection, we evaluate the performance of continuous attack detection and device authorization for the Bluetooth Protocol.

4.2.3.1 Experiment setup

- *Bluetooth attack detection* For the experimental evaluation of attack detection, a Bluetooth Piconet with a master device, a slave device (legitimate device), and

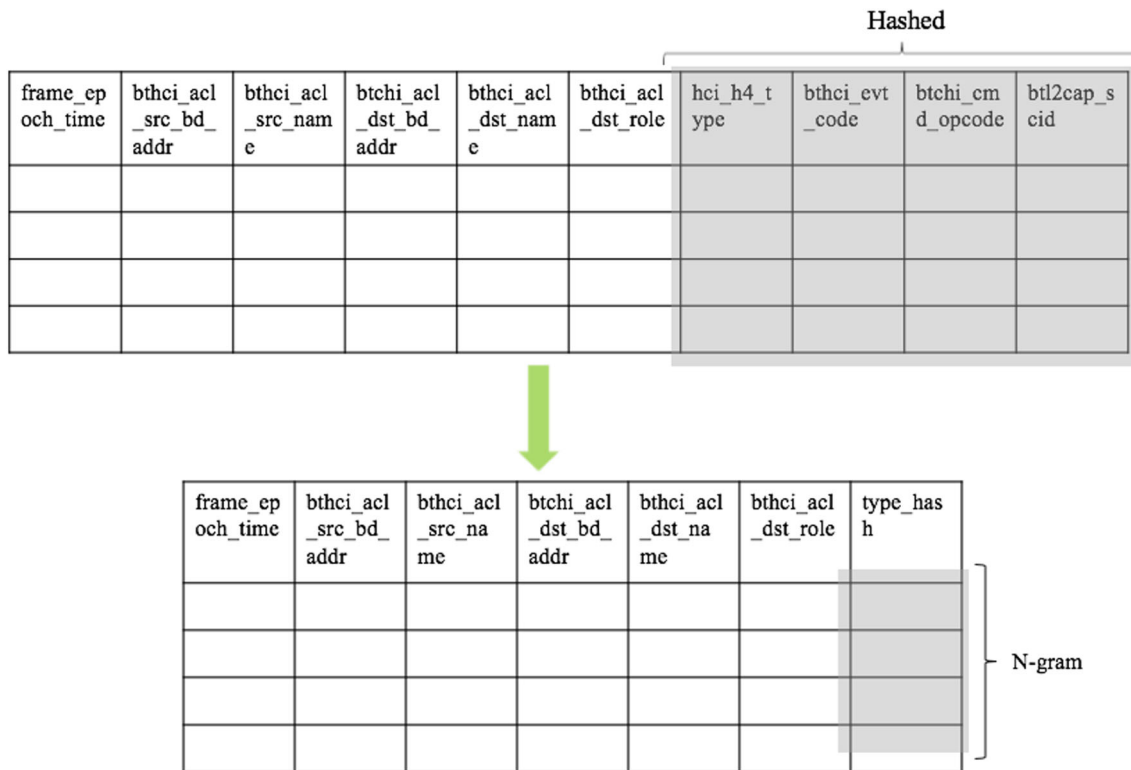
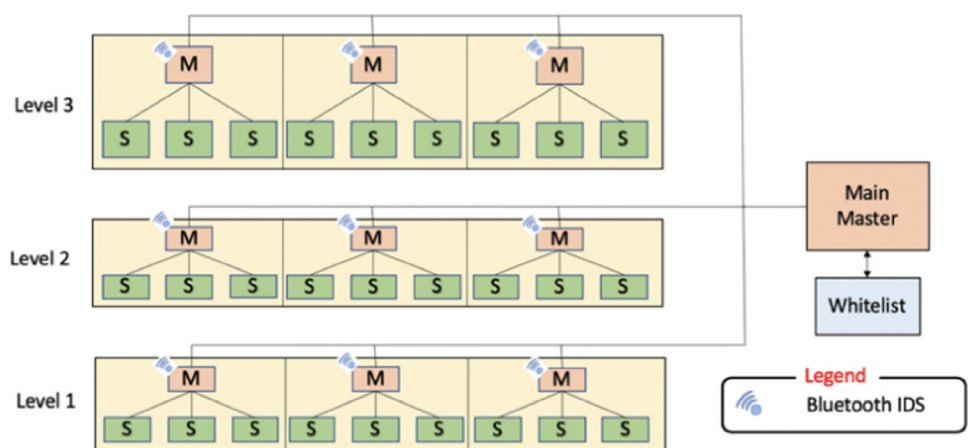


Fig. 5 Raw Bluetooth features to n-grams conversion

Table 6 Features extracted from raw Bluetooth Frames

| Features | Description |
|-----------------------------|--|
| Probability of Flow | Probability of flow extracted after Jelinek-Mercer smoothing |
| Ratio of HCI command frames | HCI packet type |
| Ratio of ACL data frame | Bluetooth HCI event code |
| Ratio of SCO data frame | Bluetooth HCI OPCode |
| Ratio of HCI data frame | Bluetooth L2CAP protocol Source CID |

Fig. 6 Bluetooth whitelisting architecture



an attacker device is used. The Bluetooth attacker targets the Bluetooth Piconet with power draining attack [32] and BlueSnarfing attacks [32]. For

experimental evaluation, normal Bluetooth traffic containing file transfers and music data were collected. Attack Bluetooth traffic includes Bluetooth attack such

as Bluesnarfing attack were also collected. The Bluetooth continuous monitoring evaluation dataset contains 3058 normal data points, and 422 attack data points after feature extraction and data preprocessing for n – gram size three and *traffic – flow* size of ten seconds.

- **Bluetooth device authorization** For the experimental evaluation of the bluetooth device authorization, a hierarchical Bluetooth network with two layers is setup, with master device receiving the results of continuous monitoring (through simulated input).

4.2.3.2 Experiment 1: N-grams size analysis This experiment aids in deciding the appropriate n – gram size for the Bluetooth continuous monitoring. As shown in Fig. 7, we observe the total number of unique n – grams learnt over the training period. The n – gram size was selected to be equal to 3. This value was shown experimentally to give the best performance [2, 3]. In Fig. 8, we observe the new n – grams learnt over time stops after first ten minutes, emphasizing for our test scenario we had captured all the normal traffic.

4.2.3.3 Experiment 2: performance evaluation In this experiment, we train a Decision Tree (C4.5), a AdaBoost classifier, a SVM with radial bias function (RBF) classifier, a Naive Bayes classifier, a Ripper Classifier, and a Bagging classifier. To address the class imbalance problem present in cybersecurity datasets, we use Synthetic Minority Oversampling Technique (SMOTE) with 0%, 100%, and 1000%. Figure 9 shows the precisions of the classifiers, while Fig. 10 shows the recalls of the classifiers. The Ripper algorithm with 100% SMOTE gives the best precision and recall, while Bagging algorithm gives the best overall precision, and recall for 0%, 100%, and 1000% SMOTE.

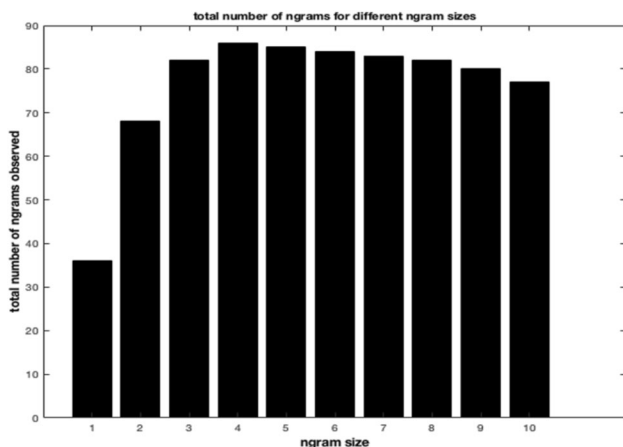


Fig. 7 Total n – grams learnt

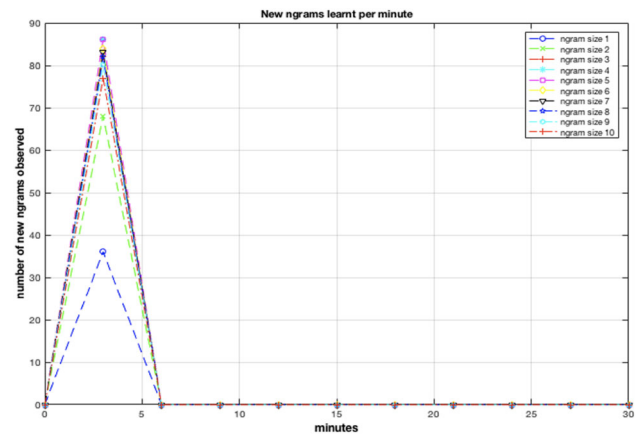


Fig. 8 New n – grams learnt with time

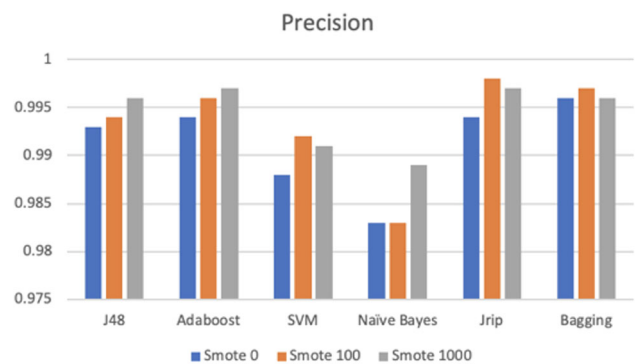


Fig. 9 Bluetooth continuous monitoring classifier precisions

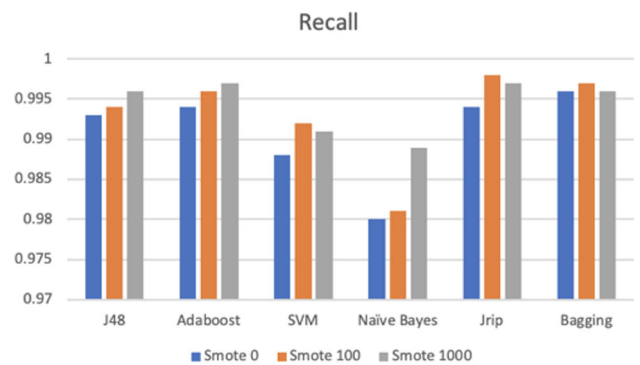


Fig. 10 Bluetooth continuous monitoring classifier recalls

4.2.3.4 Experiment 3: performance evaluation In this experiment, we evaluate the performance of the Bluetooth device authorization for the hierarchical Bluetooth network. The attacker device sends pairing request to the master device of the piconet. The master device sends the attacker device’ Bluetooth address and priority level to the Main Whitelisting Server (MWS) to check if the attacker’s Bluetooth address was authorized or not. Since the attacker device was not authorized by the MWS, the master device of the piconet did not authorize the attacker device from

joining the piconet. We also send a pairing request from the slave device to the master device, the master device was able to authorize and connect the slave device as the slave device's Bluetooth address was authorized by the MWS. Hence, the authorization approach was successfully able to detect and prevent unauthorized devices from connecting to the Bluetooth network.

5 Conclusion

In this paper, we present a Security Framework for the IoT and Smart Cyber Infrastructures. The presented security framework splits IoT and smart infrastructure into four layers consisting of End-Nodes layer, Network Layer, Service Layer, and Application Layer. In this paper, we showed how to apply the framework to End-Nodes layer and to Network layer. For the sensor-nodes, we developed a sensor model that uses the means and variances of cD for a five sample windows, of Biorthogonal, Daubechies, Coiflets, Discrete Meyer, Reverse Biorthogonal, Sysmllets DWT. Classification models (AdaBoost, Random Forest, Decision Tree, and SVMs) were trained on normal sensor data. The AdaBoost based model had the best performance, with overall precision of 0.815, overall recall of 0.685, and overall F-measure of 0.745. For the sensor-nodes, we developed a mitigation strategy to drop sensor data when attacks are detected. The continuous authorization method was used to only allow whitelisted Bluetooth device to access the network while rejecting any other devices. Classification models (Decision Tree, AdaBoost, SVM, Naive Bayes, Ripper, and Bagging) were trained using normal traffic dataset. The bagging based model had the best performance with overall Precision and Recall over 0.996 and recall of 0.996 for Synthetic Minority Over-sampling Technique (SMOTE) with 0%, 100%, and 100% additions.

Acknowledgements This work is partly supported by the Air Force Office of Scientific Research (AFOSR) Dynamic Data-Driven Application Systems (DDDAS) award number FA9550-18-1-0427, National Science Foundation (NSF) research projects NSF-1624668 and NSF-1849113, National Institute of Standards and Technology (NIST) 70NANB18H263 and Department of Energy/National Nuclear Security Administration under Award Number(s) DE-NA0003946.

Author contributions None.

Funding None.

Data availability None.

Declaration

Ethical approval None.

Informed consent None.

References

- Langner, R.: Stuxnet: dissecting a cyberwarfare weapon. *IEEE Secur. Privacy* **9**(3), 49–51 (2011). <https://doi.org/10.1109/MSP.2011.67>
- Satam, P., Hariri, S.: WIDS: an anomaly based intrusion detection system for Wi-Fi (IEEE 80.211) Protocol. *IEEE Trans. Netw. Serv. Manage.* **18**(1), 1077–1091 (2021). <https://doi.org/10.1109/TNSM.2020.3036138>
- Alipour, H., Al-Nashif, Y.B., Satam, P., Hariri, S.: Wireless anomaly detection based on IEEE 80.211 behavior analysis. *IEEE Trans. Inf. Forensics Secur.* **10**(10), 2158–2170 (2015). <https://doi.org/10.1109/TIFS.2015.2433898>
- Al-Nashif, Y., Kumar, A.A., Hariri, S., Qu, G., Luo, Y., Szidarovsky, F.: Multi-level intrusion detection system (ML-IDS). In: 5th International Conference on Autonomic Computing, ICAC 2008, pp. 131–140 (2008). <https://doi.org/10.1109/ICAC.2008.25>
- Satam, P., Alipour, H., Al-Nashif, Y., Hariri, S.: Anomaly behavior analysis of DNS protocol. *J. Internet Serv. Inf. Secur. JISIS* **5**(4), 85–97 (2015)
- Satam, P., Kelly, D., Hariri, S.: Anomaly behavior analysis of website vulnerability and security. In: Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, vol. 0 (2016). <https://doi.org/10.1109/AICCSA.2016.7945697>
- Satam, P., Satam, S., Hariri, S., Alshawi, A.: Anomaly behavior analysis of IoT protocols. *Model. Des. Secur. Internet Things* (2020). <https://doi.org/10.1002/9781119593386.ch13>
- Pacheco, J., Hariri, S.: IoT security framework for smart cyber infrastructures. In: Proceedings—IEEE 1st International Workshops on Foundations and Applications of Self-Systems, FAS-W 2016, pp. 242–247 (2016). <https://doi.org/10.1109/FAS-W.2016.58>
- Buckman, A.H., Mayfield, M., Beck, S.B.M.: What is a smart building? *Smart Sustain. Built Environ.* **3**(2), 92–109 (2014). <https://doi.org/10.1108/SASBE-01-2014-0003>
- Wang, Z., Wang, L., Dounis, A.I., Yang, R.: Multi-agent control system with information fusion based comfort model for smart buildings. *Appl. Energy* **99**, 247–254 (2012). <https://doi.org/10.1016/j.apenergy.2012.05.020>
- Sadiku, M.N.O., Musa, S.M.: Cloud computing: opportunities and challenges. *IEEE Potential* **33**, 34–36 (2014)
- Yi, S., Li, C., Li, Q.: A survey of fog computing: Concepts, applications and issues. In: Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2015 June, pp. 37–42 (2015). <https://doi.org/10.1145/2757384.2757397>
- Jabbar, M.A., Aluvalu, R.: Intrusion detection system for the internet of things: a review. *IET Conf. Publ.* (2018). <https://doi.org/10.1049/cp.2018.1419>
- Berthier, R., Sanders, W.H.: Specification-based intrusion detection for advanced metering infrastructures. In: Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC, pp. 184–193 (2011). <https://doi.org/10.1109/PRDC.2011.30>
- Olufowobi, H., Young, C., Zambreno, J., Bloom, G.: SAIDUCANT: specification-based automotive intrusion detection using controller area network (CAN) timing. *IEEE Trans. Veh. Technol.* **69**(2), 1484–1494 (2020). https://doi.org/10.1109/TVT.2019.2961344_rfseq1

16. Hajisalem, V., Babaie, S.: A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Comput. Netw.* **136**, 37–50 (2018). <https://doi.org/10.1016/j.comnet.2018.02.028>
17. Satam, P., Satam, S., Hariri, S.: Bluetooth Intrusion Detection System (BIDS). *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, 2018 November (2019). <https://doi.org/10.1109/AICCSA.2018.8612809>
18. Jin, J., Gubbi, J., Marusic, S., Palaniswami, M.: An information framework for creating a smart city through internet of things. *IEEE Internet Things J.* **1**(2), 112–121 (2014). <https://doi.org/10.1109/JIOT.2013.2296516>
19. Ferreira, H.G.C., Dias Canedo, E., De Sousa, R.T.: IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino. In: *International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 53–60 (2013). <https://doi.org/10.1109/WiMOB.2013.6673340>
20. Karagiannis, V., Chatzimisios, P., Vazquez-Gallego, F., Alonso-Zarate, J.: A survey on application layer protocols for the Internet of Things. *Trans. IoT Cloud Comput.* **3**(1), 11–17 (2015)
21. Manadhata, P.K., Wing, J.M.: An attack surface metric. *IEEE Trans. Softw. Eng.* **37**(3), 371–386 (2011). <https://doi.org/10.1109/TSE.2010.60>
22. Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., Lung, C.H.: Smart home: integrating internet of things with web services and cloud computing. *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom* **2**, 317–320 (2013). <https://doi.org/10.1109/CloudCom.2013.155>
23. Sweldens, W.: The lifting scheme: a custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* **3**(2), 186–200 (1996). <https://doi.org/10.1006/acha.1996.0015>
24. Wahid, K.A., Dimitrov, V.S., Jullien, G.A., Badawy, W.: An analysis of Daubechies discrete wavelet transform based on algebraic integer encoding scheme. In: *Proceedings—3rd International Workshop on Digital and Computational Video, DCV 2002*, pp. 27–34 (2002). <https://doi.org/10.1109/DCV.2002.1218740>
25. Graps, A.: An introduction to wavelets. *IEEE Comput. Sci. Eng.* **2**(2), 50–61 (1995). <https://doi.org/10.1109/99.388960>
26. Xu, L., Zhang, D., Wang, K.: Wavelet-based cascaded adaptive filter for removing baseline drift in pulse waveforms. *IEEE Trans. Biomed. Eng.* **52**(11), 1973–1975 (2005). <https://doi.org/10.1109/TBME.2005.856296>
27. Szewczyk, R., Grabowski, K., Napieralska, M., Sankowski, W., Zubert, M., Napieralski, A.: A reliable iris recognition algorithm based on reverse biorthogonal wavelet transform. *Pattern Recognit. Lett.* **33**(8), 1019–1026 (2012). <https://doi.org/10.1016/j.patrec.2011.08.018>
28. Phinyomark, A., Limsakul, C., Phukpattaranont, P.: An optimal wavelet function based on wavelet denoising for multifunction myoelectric control, pp. 1098–1101 (2009). <https://doi.org/10.1109/ecticon.2009.5137236>
29. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: *ICML '06: Proceedings of the 23rd international conference on Machine Learning*, pp. 233–240 (2006). <https://doi.org/10.1145/1143844.1143874>
30. Satam, S., Satam, P., Hariri, S.: Multi-level Bluetooth Intrusion Detection System. In: *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 1–8 November 2020* (2020). <https://doi.org/10.1109/AICCSA50499.2020.9316514>
31. Brown, P., Cocke, J., Pietra, S.D., Pietra, V.D., Jelinek, F., Mercer, R., Roossin, P.: A statistical approach to French/English translation. *Biol. Artif. Intell. Syst.* **16**(2), 547–561 (1988). https://doi.org/10.1007/978-94-009-3117-6_27
32. Satam, S.: Bluetooth anomaly based intrusion detection system by Shalaka Satam A Thesis submitted to the Faculty of the Department of Electrical and Computer Engineering in Partial Fulfillment of the Requirements For the Degree of Master of Science In the Graduate C (2017)

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



Shalaka Satam is currently pursuing her Ph.D. from University of Arizona. She received her MS from University of Arizona in 2017. She is a Graduate Research Assistant with the Department of Electrical and Computer Engineering, University of Arizona. Her research focuses on cyber security, Internet of Things and network security.



Pratik Satam received the M.S. and Ph.D. degrees from the University of Arizona in 2015 and 2019, respectively. He is a Research Assistant Professor with the University of Arizona. His current research focuses on autonomic computing, cybersecurity, cyber resilience, secure critical infrastructures, industrial control system security, and cloud security.



Jesus Pacheco received the bachelor's degree in electronics engineering with a focus on digital systems and the M.Sc. degree in computer science from the Tecnológico de Hermosillo and the Ph.D. degree from the Electrical and Computer Engineering Department, The University of Arizona. He is currently a Full Professor with the Industrial Engineering Department, Universidad de Sonora, and the Director of the NSF Center for Cloud and

Autonomic Computing. His research interests include cyber security for the Internet of Things and cyber-physical systems.



Salim Hariri (Senior Member, IEEE) received the M.Sc. degree from Ohio State University in 1982, and the Ph.D. degree in computer engineering from the University of Southern California in 1986. He is a Professor with the Department of Electrical and Computer Engineering, University of Arizona, and the Director of the NSF Center for Cloud and Autonomic Computing. His current research focuses on autonomic computing,

cybersecurity, cyber resilience, secure critical infrastructures, and cloud security.