SAND2019-14591C

Machine Learning Bluetooth Profile Operation Verification via Monitoring the Transmission Pattern

Abdelrahman Elkanishy*, Abdel-Hameed A. Badawy*‡, Paul M. Furth*, and Laura E. Boucheron*
Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA
† Sandia National Laboratories, Albuquerque, NM 87185, USA
‡ Los Alamos National Laboratory, Los Alamos, NM 87545, USA
{anasser, badawy, pfurth, lbouchr}@nmsu.edu

***Abstract*—Manufacturers normally buy and/or fabricate communication chips using third-party suppliers, which are then integrated into a complex hardware-software stack with a variety of potential vulnerabilities. This work proposes a compact supervisory circuit to classify the operation of a Bluetooth SoC at low frequencies by monitoring the input power and radio frequency (RF) output of the Bluetooth chip passed through an envelope detector. The idea is to inexpensively fabricate an envelope detector, power supply current monitor, and classification algorithm on a custom low-frequency integrated circuit in a trusted legacy technology. When the supervisory circuit detects unexpected behavior, it can shut off power to the Bluetooth SoC. In this preliminary work, we proto-type the supervisory circuit using off-the-shelf components. We extract simple yet descriptive features from the envelope of the RF output signal. Then, we train machine learning models to classify different Bluetooth operation profiles, including sensor, hands-free, and headset. Our results show ∼100% classification accuracy.**

***Index Terms*—Hardware Security, Supervisory Circuit, Bluetooth, Machine Learning, Security, RF Power, Classifier.**

I. Introduction

Due to the complexity and multi-functionality of smart systems, most manufacturers outsource communication chips from third-party suppliers. The integration between many outsourced ICs has resulted in the need to add a hardware security layer to ensure appropriate operation. For example, in Apple's smartphones, there is a dedicated co-processor, Secure Enclave, to handle all cryptographic operations and maintain the integrity of data protection for the entire system [1].

Bluetooth, like any communication protocol, has vulnerabilities. For instance, in 2017, Armis [2] identified a new Bluetooth attack vector called BlueBorne that can take control of the target device. BlueBorne attacks regular computers, smartphones, and IoT devices. This security breach occurs without pairing to the targeted device nor even while the Bluetooth IC is in discovery mode. As the Bluetooth chip is responsible for establishing connections and controlling data flow, BlueBorne and other security breaches could attack the Bluetooth IC without the consent of the controller chip. Therefore, monitoring a Bluetooth chip at the hardware level is necessary to verify expected operation.

As shown in Fig. 1, one way to monitor the chip is to consider it as a black box which consumes and transmits power. Thus, abnormal behavior can be detected by learning the normal input/output (I/O) power signatures. A second way is to parameterize aspects of the Bluetooth connection (e.g., profile type, the distance between paired devices, number of connected devices...) and compare the detected behavior to the expected behavior based on the controller instructions. Supervisory circuits are commonly used in detecting power failures but are not common for security purposes [3]. For example, PFP Cybersecurity [4] has partnered with XILINX to detect security breaches in XILINX's devices using artificial intelligence. Their work is focused on self-monitoring, not

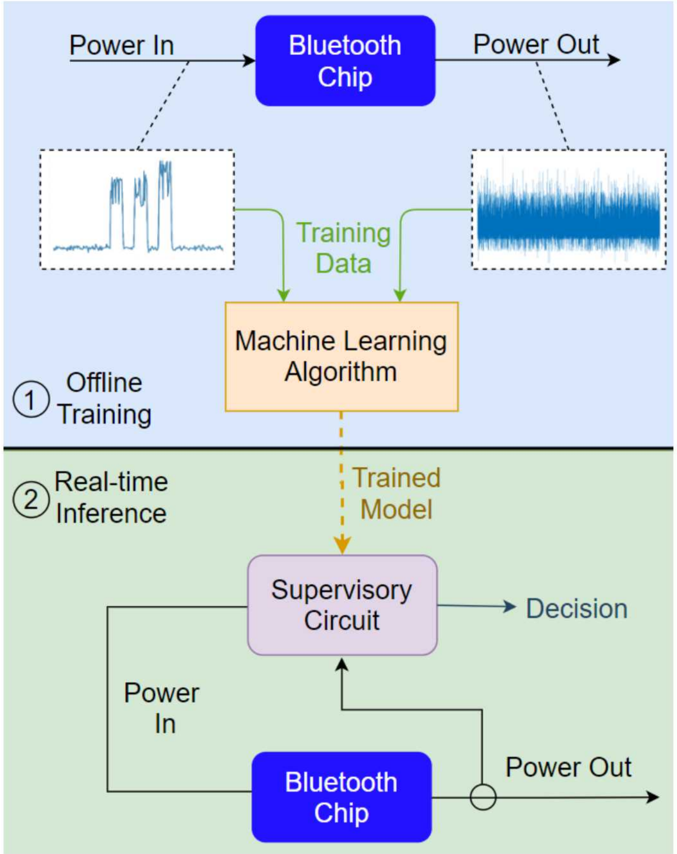

Fig. 1. Concept diagram of supervisory circuit. Input and output power signals are collected to train a machine learning model. Then, the model is used in real-time to classify modes of operation.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

monitoring another IC, and intended for XILINX devices only.

A. Background

A profile is a layer in the Bluetooth protocol stack that defines the behavior of the device [5]. The Bluetooth protocol includes profiles in order to specify the type of data transmitted by the Bluetooth module. Bluetooth profiles include information on other profiles' dependencies and also recommended user interface formats [5]. For connecting two devices, they must both support the same profile. Bluetooth profiles that must be included are determined according to the application of the device [5]. There are a wide range of traditional profiles which are a comprehensive group of services such as hands-free, headset, and health device capabilities. For example, the Heart Rate Profile combines the Heart Rate Service and the Device Information Service [6]. The Bluetooth protocol allows developers to build new profiles using Generic Attribute Profile (GATT) [5]. GATT consists of different services which are a compilation of properties and relations to other services [7]. The combination of GATT services shape the device behavior and defines the slave (GATT client) and master (GATT server) roles [8].

B. Proposed Supervisory Circuit

The goal is to create a supervisory circuit to detect unexpected operation of a complex, mixed-signal communication System-on-Chip (SoC). We design the supervisory circuit to operate at low frequency and low power, and to be inexpensive computationally. This will facilitate our ability to fabricate the supervisory circuit in an inexpensive process technology, or integrate soft intellectual property into a more advanced SoC. When the supervisory circuit detects a security abnormality, the circuit can intervene and shut down the SoC. Bluetooth is the communications standard chosen for this preliminary work; however, we anticipate that the methods described will be useful for monitoring other communication protocols.

The supervisory circuit design is split into several major blocks, as shown in Fig. 2. First, the circuit that provides and controls power is comprised of a controlled low-dropout (LDO) voltage regulator. LDOs are widely used in portable communications systems since they occupy small area, have low noise, and provide good transient performance. Embedded in the LDO is a current sensor that monitors the output current of the LDO. External to the supervisory circuit is an RF coupler, which splits the transmitted RF signal into a main path and a monitored path. The monitored path passes through the envelope detector, which lowers the frequency of the RF signal in order to be able to sample it at frequencies much lower than the 2.4 GHz Bluetooth signal. As such, the supervisory circuit can be entirely implemented using low-speed technology. The outputs of the current sensor and the envelope detector are digitized. Finally, a digital signal processing (DSP) circuit, or soft IP, will be used to extract the features from all relevant signals. At run-time, the system extracts the necessary features to feed into the Machine Learning (ML) models to determine what operation is running on the Bluetooth IC. In future work, we will compare monitored behavior to expected behavior via the Control Bus shown in Fig. 2. Details about the implementation of the controlled LDO with current sensor and envelope detector are found in [9].

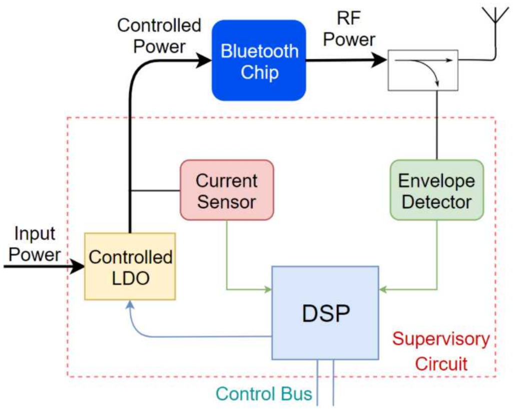

Fig. 2. Block diagram of supervisory circuit comprised of a DSP block to implement the classification algorithm and two monitoring circuits: a controlled LDO with current sensor and an envelope detector.

C. Related Work

Outsourcing IC fabrication to third party manufacturers increases the possibility of an untrusted modification to the circuitry, i.e., a hardware Trojan, during production. According to [10], non-destructive, non-invasive hardware Trojan detection techniques can be classified as either test-time or run-time. Both of these approaches are based on comparing test IC parameters with a golden IC model, i.e., parameters obtained from a known Trojan-free IC. Test-time approaches use logic testing and/or side-channel analysis to inspect the IC before integrating it into a system. Even when combining logic testing and side-channel analysis [11], test-time approaches are limited, since attacks may only be triggered after deployment. Run-time hardware Trojan detection methods monitor the chip continuously through the addition of monitoring circuitry. Bao et al. [12] use variations in temperature sensor readings to detect hardware Trojans. Hasan et al. [13] use formal verification as a framework to develop run-time hardware Trojan detection units for digital circuits. Unlike the aforementioned run-time hardware Trojan detection techniques, we are not focused only on the security of digital circuits; rather, we propose a monitoring scheme to detect hardware attacks on a high-speed mixed-signal communication SoC.

On-chip classification of IC behavior requires relatively simple computations. Iwase et al. [14] use a discrete Fourier transform feature of the voltage signal. Other classifiers select features from multiple domains [15], [16] after transforming the signal using both wavelet and/or Fourier analysis. Some researchers use statistical features from both time and frequency domains [17]. Still others extract large numbers of features from signals, then apply computationally-intensive dimensionality reduction techniques [18], [19].

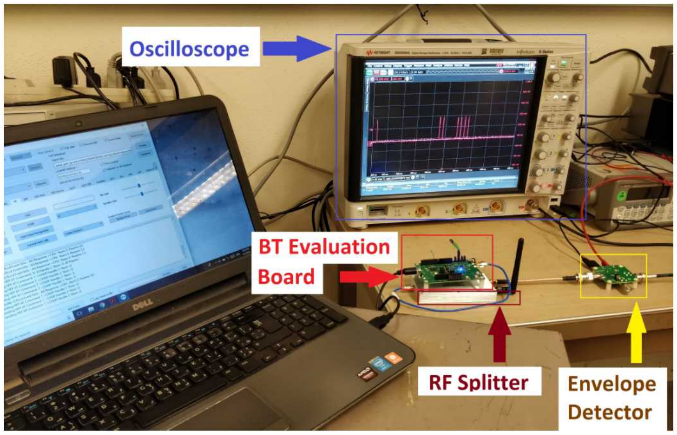

Fig. 3. Preliminary laboratory setup showing the laptop, Bluetooth evaluation board, RF splitter, envelope detection evaluation board, and oscilloscope.

In this work, we are concerned with the computational complexity of the selected features and classification algorithms. Since frequency transformations require high computational overhead, the selected features are exclusively extracted from the time domain. Indeed, we experimented with frequency domain features to verify that they provide more computational complexity without any performance advantages. In addition, we selected novel features that are computationally smart and cheap, while achieving high classification accuracy $\sim$100%.

II. Methodology

Prior to fabricating a custom supervisory circuit IC, it was prototyped using off-the-shelf components and an oscilloscope in order to collect a data set adequate for training and testing the classification algorithm. We placed small-valued series resistors in series with supply pins to the CYW20706 [20] Bluetooth SoC in order to monitor the supply current to the transceiver block. In addition, the RF output of the Bluetooth IC was passed through an RF splitter. One side of the splitter went to an antenna for pairing with other Bluetooth devices, while the other side was attached to an AN-2264 LMH2121 envelope detector [21]. This particular envelope detector has an input bandwidth from 0.1 to 3 GHz which covers the Bluetooth band. The envelope detection stage effectively lowers the bandwidth of the RF signal. This experimental setup is depicted in Fig. 3. A laptop controls the Bluetooth board via USB. The oscilloscope samples and saves the envelope-detected RF stream and input power signal.

The Bluetooth board is programmed to act as two popular profiles: hands-free and headset, in addition to customized profiles using GATT services. While each profile is running, different events occur, such as dialing, hangup, and streaming music. The events are controlled using a graphical user interface, as shown in Fig. 4, which utilizes a serial port through USB to send commands to the Bluetooth evaluation board. The network topology of two devices is defined. Moreover, we collect the RF streams of each profile in both the advertising and transmitting/receiving (transceiving) states. First, the hands-free profile RF output signal is recorded while executing multiple events, including dialing, answering, and hang up. Second, the headset profile RF output signal is captured during various events, such as streaming music, rewind, scrub, and volume control. Lastly, a customized profile is used to simulate a simple embedded system connected through Bluetooth. Basically, it notifies the Bluetooth evaluation board of a sensor reading that controls the number of blinks of an LED.

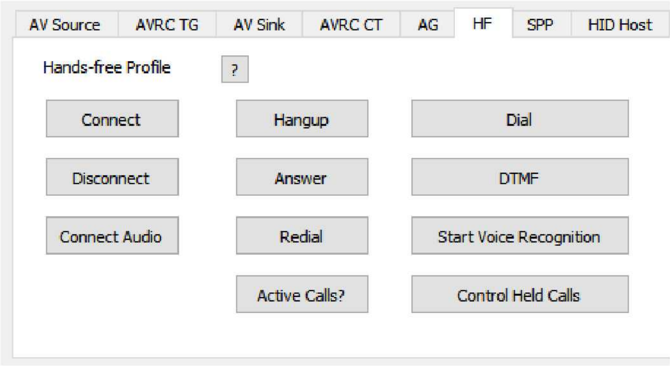

Fig. 4. Graphical user interface of Bluetooth board in the hand-free profile.

The oscilloscope captures the RF envelope-detected signal at a sampling frequency of 50 kHz. A processing window of 640 ms (corresponding to 32,000 samples) with one sample advance is selected to collect as many transmitting events as possible with a minimal computational load. Three features are extracted from each window to train the ML models. The first feature is the maximum signal value in the given window, since the maximum signal value is expected to vary from one transmitting state to the next. Changes in the maximum value are related to the different profiles. As we are interested in the pattern of the Bluetooth transmission, the other two features are extracted after thresholding the envelope-detected stream into two binary levels. In other words, the signal is 1-bit quantized, where value 1 means the Bluetooth is transmitting, whereas value 0 indicates no transmission. The remaining two features extracted in each window are the total number of 1's (or area) and the number of 0-to-1 transitions (or number of pulses). The area is correlated to the total transmission duration in a certain window, whereas the number of pulses represents the density of the transmission events.

TABLE I
Observation distribution of the data set.

Profile	Advertising State Observations	Transceiving State Observations	Total
Sensor	$2,823$	$2,736$	$5,559$
Hands-free	$2,033$	$159,211$	$161,244$
Headset	409	$107,469$	$107,878$
Total	$5,265$	$269,416$	$274,681$

The data set is constructed of 189,522 unique windows, or, in the language of ML, observations. MATLAB is used to train and test the models [22]. Table I shows the distribution of the data set across the three profile types: sensor, hands-free, and headset. At early stages of the experiment, data was only

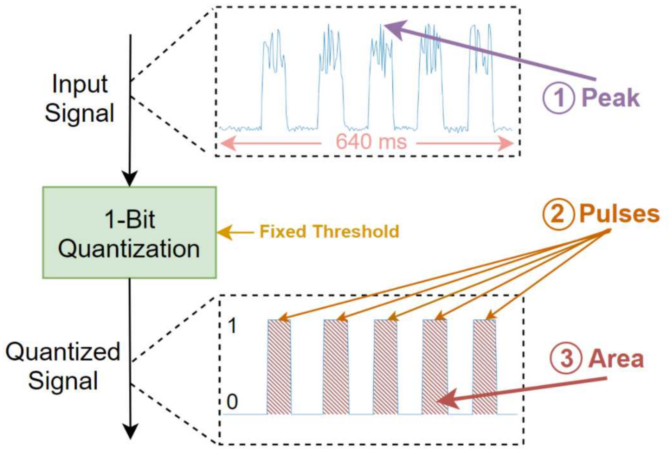

Fig. 5. Time-domain feature extraction process in each 640 ms window. The Peak, or maximum, feature is extracted first. Then, after 1-bit quantization, the Pulses feature, which is the number of 0-to-1 transitions, and the Area feature, which is the total number of 1's, are extracted.

collected for the sensor profile. Based on the scatter plot of sensor profile data, the two classes (advertising and transceiving) were linearly separable. As such, linear techniques were sufficient. After the other two profiles were added (hands-free and headset), linear methods were not enough to reach high accuracy. Thus, quadratic, cosine and cubic based algorithms were explored. However, in order to achieve high accuracy and prediction speed, we eventually looked to multi-region separation methods, such as KNN and decision tree.

The entire data set of 274,681 observations is fed to several different ML algorithms to classify the state as either advertising or transceiving. As discussed earlier, the chosen models included decision tree, K-Nearest Neighbor (KNN), support vector machine (SVM), and quadratic discriminant analysis. The purpose is to compare their accuracy and prediction speed. For all classifiers, 25% holdout validation is used for testing the models. Prediction speeds are measured on the same computer using MATLAB.

III. Results and Discussion

Table II summarizes the performance of the different ML algorithms which are applied to classify the profile. After training, we calculate the prediction speed and classification accuracy. As can be seen, logistic regression is the fastest algorithm in prediction but is less accurate than eight other tested algorithms. Among the remaining eight algorithms, decision tree is the most accurate model with 99.99% accuracy and the second fastest predictor. KNN with k=1 (1-NN), Cubic KNN and weighted KNN have similarly high prediction accuracy, but their prediction speeds are significantly lower than that of decision tree.

Looking simultaneously for both high accuracy and prediction speed, we note that decision tree is $3.5\times$ faster than KNN (K=1) in prediction. Also, regarding the hardware implementation of the models, the KNN (K=1) model needs more storage than that of the decision tree, because KNN (K=1) keeps a copy of the training data in order to calculate the Euclidean distance between the prediction point and the nearest training set observation. The point is then classified according to the class of the closest observation. In contrast, the implementation of the decision tree algorithm is based on branching, with a maximum number of branches per feature of 100 in this case. Therefore, the computational load of the decision tree is much less than that of KNN (K=1).

TABLE II
Machine Learning model comparison in terms of accuracy and prediction speed, as measured in observations per second.

	Accuracy	Prediction Speed (obs/sec)
Decision Tree	**99.99%**	**890,000**
KNN (K=1)	99.98%	250,000
Quadratic Discriminant	67.30%	640,000
Logistic Regression	90.10%	1,500,000
Cosine KNN	98.90%	380
Cubic KNN	99.98%	20,000
Weighted KNN	99.98%	78,000
Linear SVM	71.80%	2,700
Quadratic SVM	35.20%	46,000
Qubic SVM	30.60%	1,300,000
Gaussian SVM	99.80%	13,000

This initial proof-of-concept demonstrates high classification accuracy for Bluetooth profiles. We note, however, that the current classifier is limited to only three profiles in this preliminary work: sensor, hands-free, and headset.

As the classifier is applied at the last point of the Bluetooth physical layer, the proposed design of the supervisory circuit can detect many security breaches in data transfer behavior. For example, the BlueBorne attack takes control of Bluetooth enabled devices without authorization from the user. The transmission state classifier can detect the connection to the attacking device, and report it to the target device. Thus, the target device can discover that there is a connection at the physical layer without authentication at the software layer. Then, the target device can shut down the Bluetooth chip through the controlled LDO.

IV. Conclusion

In this paper, we demonstrated the ability to monitor and verify the operation of a Bluetooth SoC, thus preventing unauthorized connections and/or data transmission. We used low-frequency measurements of the envelope of the RF output signal sampled at 50 kHz for training and testing the classifier. Moreover, three computationally-simple features were extracted in each window, enough to achieve very high classification accuracy ($\sim$$100\%$). These three features, as well as the decision tree algorithm itself, require low computational resources, which allow the DSP block of the supervisory circuit to be implemented in small area and operate at low speeds with low power.

Acknowledgment

The authors thank Christopher Michael of Sandia National Laboratories for his feedback and suggestions on this research. This work was funded through Sandia National Laboratories under contract # GR0005652. Sandia National Laboratories is managed/operated by a subsidiary of Honeywell Inc., for the NNSA under contract DE-NA0003525. Any views or opinions expressed in this paper are the authors only and do not necessarily represent the views of the US DOE or the US Government.

REFERENCES

[1] Apple Inc., *iOS Security, iOS 11*, Jan 2018. [Online]. Available: https://www.apple.com/business/docs/iOS_Security_Guide.pdf

[2] Armis Inc., *The Attack Vector BlueBorne Exposes Almost Every Connected Device*, 2019. [Online]. Available: https://www.armis.com/blueborne/

[3] Tyco Electronics, *Coordinated Circuit Protection Schemes Help Prevent Overvoltage and Overcurrent Damage*, 2005. [Online]. Available: http://www.te.com/documentation/whitepapers/pdf/eDigest-Circuit_Protection_Devices.pdf

[4] Power Fingerprinting Inc., *Power Fingerprinting (PFP) cybersecurity*, 2019. [Online]. Available: https://www.pfpcyber.com/

[5] *Traditional Profile Specifications*. [Online]. Available: https://www.bluetooth.com/specifications/profiles-overview

[6] *GATT — Introduction to Bluetooth Low Energy*. [Online]. Available: https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt

[7] *GATT Specifications*. [Online]. Available: https://www.bluetooth.com/specifications/gatt

[8] *GATT Overview*. [Online]. Available: https://www.bluetooth.com/specifications/gatt/generic-attributes-overview

[9] P. M. Furth *et al.*, "Supervisory circuits for low-frequency monitoring of a communication SoC," *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2019.

[10] K. Hasegawa, M. Yanagisawa, and N. Togawa, "Hardware trojan: Threats and emerging solutions," in *HLDVT Workshop*, Nov 2009.

[11] S. Mal-Sarkar *et al.*, "Design and validation for fpga trust under hardware trojan attacks," *IEEE Trans MSCS*, Jul 2016.

[12] C. Bao, D. Forte, and A. Srivastava, "Temperature tracking: Toward robust run-time detection of hardware trojans," *IEEE Trans CADICS*, Oct 2015.

[13] S. R. Hasan *et al.*, "Translating circuit behavior manifestations of hardware trojans using model checkers into run-time trojan detection monitors," in *IEEE AsianHOST*, Dec 2016.

[14] T. Iwase *et al.*, "Detection technique for hardware trojans using machine learning in frequency domain," in *IEEE GCCE*, Oct 2015.

[15] G.-S. Hu, J. Xie, and F.-F. Zhu, "Classification of power quality disturbances using wavelet and fuzzy support vector machines," in *ICMLC*, Aug 2005.

[16] P. Geng *et al.*, "Fault pattern recognition method for the high voltage circuit breaker based on the incremental learning algorithms for svm," in *IEEE DEIS CMD*, Sep 2016.

[17] A. Emrani and M. Pourhomayoun, "Applying machine learning techniques to recognize arc in vehicle 48 electrical systems," in *IEEE COMPEL*, Jul 2017.

[18] R. Shende and D. D. Ambawade, "A side channel based power analysis technique for hardware trojan detection using statistical learning approach," in *IEEE WOCN*, Jul 2016.

[19] J. G. Ferreira and A. Warzecha, "An application of machine learning approach to fault detection of a synchronous machine," in *IEEE SME*, Jun 2017.

[20] Cypress Semiconductor Corp., *CYW920706WCDEVAL Hardware User Guide*, 2017. [Online]. Available: http://www.cypress.com/file/378381/download

[21] Texas Instruments Inc., *AN-2264 LMH2121 Evaluation Board*, May 2013. [Online]. Available: http://www.ti.com/lit/an/snoa873b/snoa873b.pdf

[22] The MathWorks Inc., *Classification Learner*, 2019. [Online]. Available: https://www.mathworks.com/help/stats/classificationlearner-app.html