Data Provenance for IoT using Wireless Channel Characteristics and Physically Unclonable Functions

Muhammad Naveed Aman
Deptt. of Computer Science
National University of Singapore
Singapore

Mohammed Haroon Basheer
Deptt. of Computer Science
National University of Singapore
Singapore

Biplab Sikdar
Deptt. of Electrical & Computer Engg.
National University of Singapore
Singapore

***Abstract*—IoT can provide many new exciting services in energy management, home and commercial automation, and environmental monitoring etc. Data provenance establishes the trust in the origin and location of data. This paper takes an information theoretic approach to solve the problem of data provenance in IoT systems. The proposed protocol uses Physically Unclonable Functions to prove the origin of data and wireless fingerprints derived from the received signal strength indicator (RSSI) measurements to verify the location of the IoT device producing the data. The security analysis of the proposed protocol shows that it is robust against different types of attacks. Experimental results show that the proposed technique can improve the accuracy of detecting attacks by 100% as compared to existing techniques.**

***Index Terms*—Internet of Things, Physically Unclonable Functions, RSSI, Data Provenance, Authentication.**

I. Introduction

IoT devices are growing at an exponential rate will cross 50 million devices connected to the Internet by 2020 [1]. The simple nature and huge volumes of data that these devices produce makes them vulnerable to cyber threats. The security requirements of IoT systems include authentication, data integrity, data provenance, and privacy, among others.

IoT devices may be installed in open and remote locations, which exposes them to physical attacks. Therefore, it is important that the IoT devices do not store any secret keys in their memory. Secure authentication is crucial to the correct operation of IoT systems. This paper uses Physically Unclonable Functions (PUFs) to verify the identity of an IoT device and establish mutual authentication of the entities. PUFs exploit the randomness in the (sub-) microscopic structure of integrated circuits introduced due to manufacturing process and can be used effectively for hardware obfuscation [2].

Trust in the fidelity of data is institutes data provenance i.e., if a piece of data has been received from an IoT device then it is indeed collected by the stated device at the stated location. Trust in the data generated by IoT devices is crucial to the success of IoT systems [3]. Consider the case when a patient is developed some lung ailment. The insurance provider may offer a discount on the patients premium given he/she quits smoking. To keep track of the patient, the patient is asked to put on a wearable IoT device with sensors to monitor smoking. However, the patient may attempt to cover up smoking episodes by hacking into the IoT device and maliciously tampering the sensor readings.

The existing work on data provenance is mostly focused on databases. However, the literature on data provenance for IoT systems is limited and most of the existing techniques are at unsafe against physical, cloning, impersonation, and DoS attacks. Furthermore, this techniques employ computationally expensive cryptographic operations not suitable for IoT devices. To solve these issues, this paper takes an information theoretic approach to data provenance and capitalizes on the wireless channel characteristics between two communicating parties. In particular, we utilize the received signal strength indicator (RSSI) measurements to generate "wireless fingerprints" between two entities.

The major contributions of this paper are as follows: (i) Establishing data provenance (for location) using RSSI measurements to distinguish between legitimate channels and adversarial channels, (ii) a PUF based authentication protocol which establishes the data provenance in terms of the source of data, (iii) Experimental results validating the proposed technique for wireless fingerprint generation, and (iv) a security analysis of the proposed protocol using formal security proofs.

The rest of the paper is organized as follows. Section II discusses the related work and Section III provides an introduction to PUFs. Section IV discusses our network model, assumptions and security requirements. Section V describes the proposed technique to generate wireless fingerprints along with the experimental results and Section VI presents the proposed data provenance protocol. The security analysis is presented in Section VII. The experimental validation is presented in Section VIII and we finally conclude the paper in Section IX.

II. Related Work

Some of the recently proposed protocols for authentication using PUFs include [4], [5]. However, these techniques do not provide any means of data provenance. The challenges of implementing and integrating data provenance in IoT systems is presented in [6], [7]. A provenance based trust management systems is proposed in [8]. The use of hash chains to propagate provenance data for IoT devices is proposed in [9]. Further-

more, the authors of [10] propose the use of non-interactive zero-knowledge proofs (NI-ZKP) for data provenance in IoT.

We observe that the above techniques suffer from one or more of the following issues:

1) Rely on **specialized hardware** not feasible for low cost IoT devices.
2) Use **complex computations** not feasible for constrained IoT devices.
3) Depend on stored secret keys in the device's memory exposing them to **physical and cloning attacks**.

This paper solves these issues as follows:

1) The proposed protocol uses light weight symmetric cryptography feasible for resource constrained IoT devices.
2) The IoT devices are protected against physical and cloning attacks using PUFs. Note that PUFs can generate secret keys when ever needed eliminating the need for storing secret keys.
3) The source of the data is verified using PUFs while the location of data is verified using wireless fingerprints. To the best of our knowledge PUFs and wireless fingerprints have not been used together in the existing literature.

III. Introduction to Physically Unclonable Functions

PUFs provide a challenge response mechanism using the randomness embedded in the complex physical system of an integrated circuity. A PUF is characterized by a challenge response pair (CRP), i.e., $R = P(C)$, where R is the response a PUF P produces when excited with a challenge C. PUFs can support ultra high throughput with ultra low energy and silicon area footprints making them attractive to use as hardware security primitives in IoT devices [11].

If the same input is given a PUF multiple times, it will produce the same output with high probability. However, if the same challenge is given to a different PUF, it will produce a different output with high probability. The PUF output is sensitive to environmental factors. However, a stable PUF can be obtained using fuzzy extractors [12]. Hence, PUFs can be used to produce unique outputs. This paper assumes an ideal PUF i.e., the output of the PUF is stable and does not change with environmental factors.

IV. Network Model, Assumptions, and Threat Model

A. Network Model

The network model consists of a set of IoT devices, 6LoWPAN boarder routers (wireless gateways), and the server as shown in Figure 1. The IoT devices are connected to the boarder routers through wireless link.

B. Notations

ID_A, $\{M\}_k$, C^i, and R^i represent the ID of IoT device A, message M encrypted using key k, challenge to a PUF, and the response of a PUF for C^i, respectively. Similarly, the concatenation operator is denoted by $\|$.

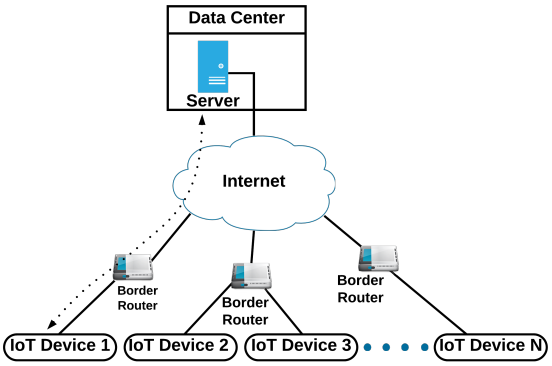


Fig. 1: Network model.

C. Assumptions

The following assumptions are made:

a. Every IoT device is equipped with a PUF. The PUF and the device's microcontroller forms a system-on-chip (SoC). Any effort to meddle/separate the PUF from the device will render the PUF useless [13], [14].
b. An adversary cannot eavesdrop on the communication between the PUF and microcontroller given the SoC assumption [13], [14].
c. The server is assumed to be trusted and secure.
d. IoT devices have limited resources such as energy, memory, and processing capabilities. However, the server does not have such limitations.

D. Threat Model

IoT devices authenticate with the server and send data to the server on an insecure network. The objective of the adversary is to launch an impersonation attack and authenticate itself with the server and/or tamper with the data sent by IoT devices. The adversary is able to eavesdrop, inject, replay, modify, and drop packets sent on the in-secure network.

E. Security Requirements

we intend to design a protocol with the following security requirements:

1) Mutual Authentication of the IoT device and server.
2) Establish data provenance in terms of the identity and location of an IoT device.
3) Protection against physical and cloning attacks by ensuring no secrets are stored in an IoT device's memory.

V. Extracting Wireless Fingerprints

The theory behind using wireless channel characteristics for security is as follows: The wireless channel between two communicating parties is intrinsically symmetric. For example, if two entities Alice and Bob transmit identical signals using identical transceivers and antennas, they will receive identical signals. Therefore, if Alice and Bob sample the wireless channel between them for parameters such as radio signal strength, the measurements will be similar to a high degree. However, if

an adversary located at least one wavelength away from Alice measures the radio signal strength of its wireless channel with Bob, the measurements will be significantly different from the ones between Alice and Bob [15]. This shows that the wireless channel characteristics between two communicating parties can be used to generate unique fingerprints.

Wireless channel characteristics for security is a well studied and established area. Secret key generation for different wireless technologies including Bluetooth [16], UWB [17], and WiFi [18]. Other applications in security include proximity based authentication [19], intrusion detection [20], secure pairing [21], and detecting Sybil and spoofing attacks [22], [23].

The existing work on using wireless channel characteristics for data provenance is limited. The authors of [24] proposed the use of RSSI measurements to extract wireless link fingerprints. Their technique compares the Pearson correlation coefficient for the wireless link fingerprints derived at the transmitter and receiver individually with that of adversary. However, this technique relies on long wireless fingerprints (around 2000 bytes), complex public key encryption and the authors do not provide a complete protocol for data provenance.

In this paper we propose the use of mean squared error (MSE) as the metric to detect attacks on data provenance i.e., wireless fingerprints derived from the RSSI measurements at the transmitter and receiver are compared on the basis of MSE. Thus, we calculate the MSE as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \tag{1}$$

where X_i and Y_i denote the RSSI values for the *i*th packet of each entity and n is the wireless fingerprint size.

The wireless fingerprints between two entities Alice and Bob are validated using the following procedure:

1) Alice and Bob concatenate the RSSI values for a specific period of time (according to the wireless fingerprint size used) to construct their respective wireless fingerprints individually
2) Alice and Bob send their wireless fingerprints to a server for verification.
3) The server calculates the *MSE* of the two wireless fingerprints and compares the resulting $\boldsymbol{MSE}$ to a threshold value.
4) If the *MSE* of the wireless fingerprints is below the threshold the wireless link between Alice and Bob is considered valid. Otherwise, the wireless fingerprints are considered invalid indicating a possible attack by an adversary.

The *MSE* threshold for detecting attacks is determined experimentally in Section VIII.

VI. Proposed Data Provenance Protocol

This section discusses the proposed protocol for data provenance in IoT systems.

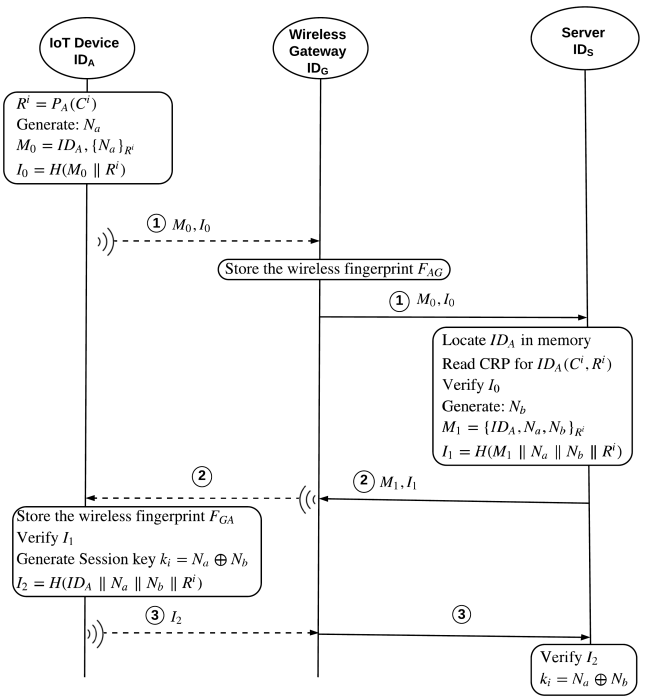


Fig. 2: Authentication Phase.

A. Device Registration

The server and the IoT device exchange the initial CRP during this phase. This can be done using a time-based one-time password algorithm (TOTP) [25]. When an IoT device is first deployed the initial parameters are exchanged with the help of an operator using a password and TOTP. As a result, the server stores an initial CRP (C^i, R^i) for each IoT device. Whereas, each IoT device stores the current challenge in its memory. Note that the IoT device does not store any secrets in its memory i.e., even if an adversary extracts C^i from an IoT device, he/she can not obtain the secret response R^i. We assume that the wireless gateway and the server have a pre-established secure symmetric key k_{GS}.

B. Authentication Phase

Let us consider a scenario where an IoT device ID_A wants to send some data to the server. The IoT device needs to authenticate with the server before the data transfer as shown in Figure 2. The steps of the authentication phase are as follows:

1) The IoT device ID_A uses its PUF and the stored C^i to generate a response R^i. It then generates a random nonce N_A and sends message $M_0 = ID_A, \{N_a\}_{R^i}$ and the corresponding authentication parameter I_0 to the server through the wireless gateway i.e., message 1 in Figure 2. Note that the authentication parameter is used to ensure data integrity of the message M_0. We use a similar approach throughout this paper.

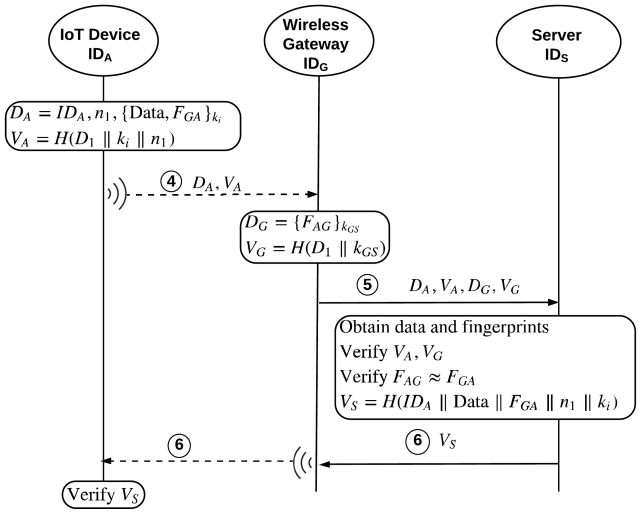


Fig. 3: Data Transfer Phase.

2) The wireless gateway forwards message 1 to the server. However, it also samples the wireless link between itself and ID_A to generate the wireless fingerprint F_{AG}.
3) The sever searches its memory for ID_A and reads the corresponding CRP (C^i, R^i). The server then verifies I_0. If the verification fails, the authentication request is rejected. Otherwise, the server generates a random nonce N_b and sends message $M_1 = \{ID_A, N_a, N_b\}_{R^i}$ along with the corresponding authentication parameter I_1 to the IoT device in message 2 in Figure 2.
4) The IoT device ID_A samples the wireless link between itself and the wireless gateway to generate the wireless fingerprint at its end F_{GA}. It then verifies the authentication parameter. If verification fails, the authentication request is terminated. Otherwise, the IoT device generates a session key using the secret nonces i.e., $k_i = N_a \oplus N_b$ and sends and acknowledgement in the form of the authentication parameter I_2 to the server.
5) The server verifies I_2. If verification fails, the authentication request is rejected. Otherwise, the server generates the session key k_i and authentication is considered complete.

C. Data Transfer Phase

The IoT device can start to transfer data to the server after successfully completing the authentication phase as shown in Figure 3. The steps of the data transfer phase are as follows:

1) The IoT device sends the data by encrypting it and its wireless fingerprint using the session key in the message $D_A = ID_A, n_1, \{Data, F_{GA}\}_{k_i}$ along with the corresponding authentication parameter in message 4 in Figure 3.
2) The wireless gateway forwards message 4 to the server along with its wireless fingerprint and the corresponding authentication parameter as shown in message 5 in Figure 3.
3) The server obtains the data and the wireless fingerprints using the corresponding secret keys for IoT device ID_A and the wireless gateway. It then verifies the authentication parameters. If verification fails, the data is rejected. Otherwise, the server verifies the provenance of the data using the wireless fingerprints and technique described in Section V. If the wireless fingerprints fail the validation, the data is rejected. Otherwise, the server accepts the data and sends an acknowledgment to the IoT device ID_A in the form of the authentication parameter V_S in message 6 of Figure 3.
4) The IoT device verifies V_S and if verification fails it may retry to send the same data. Otherwise, the IoT device may send the next piece of data to the server using the same steps above or conclude the session.

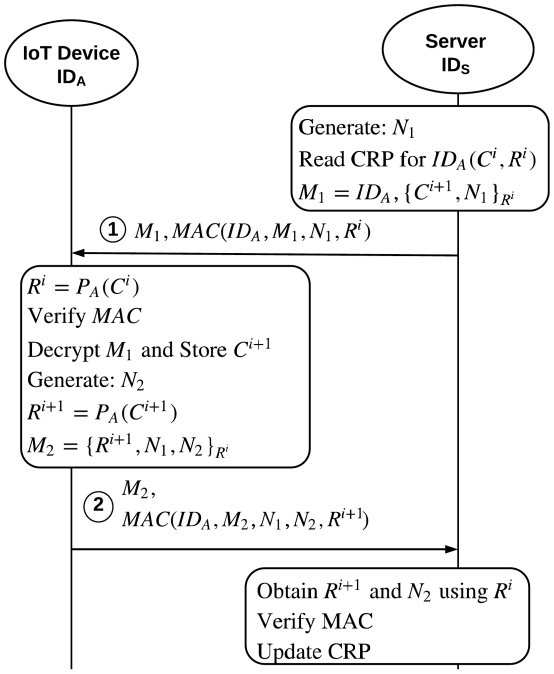


Fig. 4: Protocol for CRP update.

D. CRP Update

In the proposed data provenance protocol the server stores one CRP for each IoT device. To maintain the freshness, the server may need to update the CRP from time to time and obtain a new CRP. This can be done using the CRP update protocol shown in Figure 4. Let us assume the server wants to update the CRP for an IoT device ID_A. The steps for CRP update are as follows:

1) The server commences the CRP update protocol as follows. It generates a random nonce N_1 and reads the CRP for the IoT device ID_A. The server then sends message $M_1 = ID_A, \{C^{i+1}, N_1\}_{R^i}$, to the IoT device

ID_A, where C^{i+1} is the new challenge. The server also sends a message authentication code (MAC) along with M_1 to establish data integrity.

2) The IoT device uses the stored challenge C^i to generate the secret PUF response R^i. It then verifies the MAC, if verification fails, the CRP update is rejected. Otherwise, the IoT device decrypts M_1 to obtain the new challenge C^{i+1}. The IoT device then uses this new challenge to obtain the new response R^{i+1}. The IoT device ID_A then sends the new response in the encrypted message $M_2 = \{R^{i+1}, N_1, N_2\}_{R^i}$ along with the corresponding MAC to the server in message 2 of Figure 4.
3) The server decrypts M_2 to obtain R^{i+1} and N_2 using R^i. It then verifies the MAC, if verification fails, the CRP update is terminated. Otherwise, the server updates the CRP for IoT device ID_A in its memory with the new CRP C^{i+1}, R^{i+1}).

VII. Formal Proof of Security Using BAN Logic

A formal security analysis of the proposed protocol is presented using an extension of BAN logic i.e., the Mao and Boyd logic [26]. Formal logical approaches for security analysis of new protocols helps in ensuring that an adversary cannot reveal or tamper vital information leading to a successfull attack. For ease of notation we represent the server using S and the IoT device ID_A as A. We show that an adversary cannot launch impersonation, man-in-the-middle, data tampering, and replay attacks successfully against the proposed protocol by proving the authentication properties and establishing the syntactic secrecy of N_a, N_b, and R^{i+1}.

The idealized messages for the authentication phase are as follows:

1) $A \rightarrow S : A, \{N_a\}_{R^i}$.
2) $S \rightarrow A : A, \{N_a, N_b\}_{R^i}$.
3) $A \rightarrow S : \{A, N_a, N_b\}_{R^i}$.

The initial beliefs/assumptions for the authentication phase are as follows:

1) $A \models A \overset{R^i}{\leftrightarrow} S$ and $S \models A \overset{R^i}{\leftrightarrow} S$.
2) $A \models S^c\triangleleft \parallel N_a$ and $S \models A \models \{S\}^c\triangleleft \| N_a$.
3) $A \models S \models \{A\}^c\triangleleft \| N_b$ and $S \models A^c\triangleleft \| N_b$.
4) $A \models \#(N_a)$ and $S \models \#(N_b)$.
5) $A \models sup(S)$ and $S \models sup(A)$.
6) $A \overset{R^i}{\triangleleft} N_a$ **R** N_b.
7) $S \overset{R^i}{\triangleleft} N_a$ **R** N_b.
8) $S \overset{R^i}{\mid\sim} N_b$ and $A \overset{R^i}{\mid\sim} N_a$.

Figure 5 shows the tableau to establish the authentication properties of the proposed protocol. The set of inference rules to establish our security claims can be found at [26]. For example, to prove the authentication of A to S, we need to show that the following claim is true: $S \models A \overset{R^i}{\mid\sim} N_a$ i.e., S believes that A sent N_a using R^i as the encryption key. To prove this the authentication rule from [26] is applied i.e., we need to prove $A \models A \overset{R^i}{\leftrightarrow} S$ i.e., R^i is a good shared secret

$$\frac{A \models A \overset{R^i}{\leftrightarrow} S \bigwedge S \overset{R^i}{\triangleleft} N_a}{S \models A \overset{R^i}{\mid\sim} N_a}$$

(a) "**S** believes **A** sent N_a using R^i as the encryption key". This proves authentication of A to S.

$$\frac{A \models A \overset{R^i}{\leftrightarrow} S \bigwedge A \overset{R^i}{\triangleleft} N_b}{A \models S \overset{R^i}{\mid\sim} N_b}$$

(b) "**A** believes **S** sent N_b using R^i as the encryption key". This proves authentication of S to A.

Fig. 5: Security proofs for authentication.

between A and S and that $S \overset{R^i}{\triangleleft} N_a$ i.e., S decrypted N_a using R^i as the key. Both of these statements can be found in the initial assumptions/beliefs. Therefore, we can infer the authentication of the IoT device ID_A to the server from Figure 5(a). Similarly, to prove that the session key k_i between A and S is secure we need to establish the secrecy of N_a and N_b. This is done in the tableaux in Figure 6. Thus, these proofs show that the proposed protocol is safe against various types of attacks.

In a similar fashion, the security of the CRP update protocol can be established using the tableaux in Figure 7.

VIII. Experimental Validation

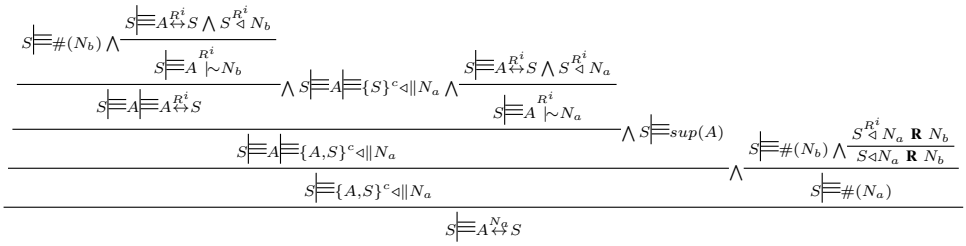
To validate the proposed technique for data provenance using wireless fingerprints we conducted experiments using MICA-Z motes running TinyOS. These motes have the CC2420 tranceiver on board which operates in the 2.4 GHz band with the IEEE 802.15/zigbee wireless communication protocol.

We also compare the proposed technique's accuracy in terms of detecting attacks with the state of the art technique proposed by [24]. The technique proposed by [24] uses the pearson correlation coefficient of the wireless fingerprints between two entities to establish data provenance as follows:

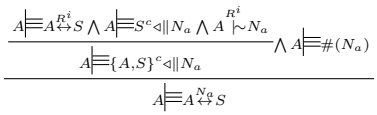
$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y}}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2} \cdot \sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \qquad (2)$$

where X_i and Y_i denote the RSSI values for the ith packet of each entity, respectively. $\bar{X}$ and $\bar{Y}$ are respective mean RSSI values of a sequence of n packets. [24] use a threshold of 0.9 to differentiate between adversarial channels and legitimate channels i.e., if the value of r is greater than 0.9 we conclude a legitimate channel and vice versa.

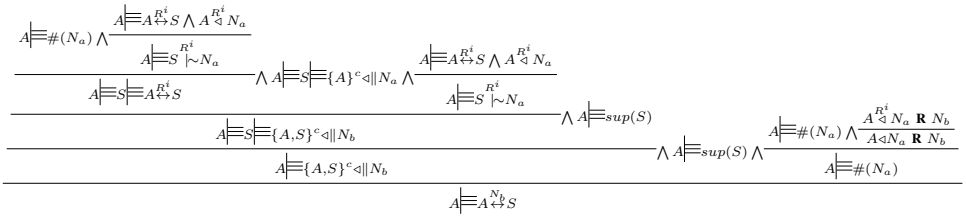
We carried out the experiments in an indoor laboratory environment with typical furniture and WiFi equipment as shown in the layout in Figure 8. The setup includes a legitimate IoT device, a base stations, and two adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. The adversaries are located at least one wavelength away from the legitimate IoT device. We considered the scenario when the IoT device moves around the lab to different locations



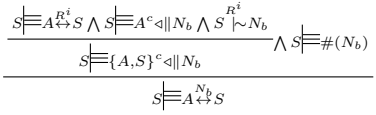
(a) Proof of "**S** believes N_a is a good shared key of **A** and **S**".



(b) Proof of "**A** believes N_a is a good shared key of **A** and **S**".

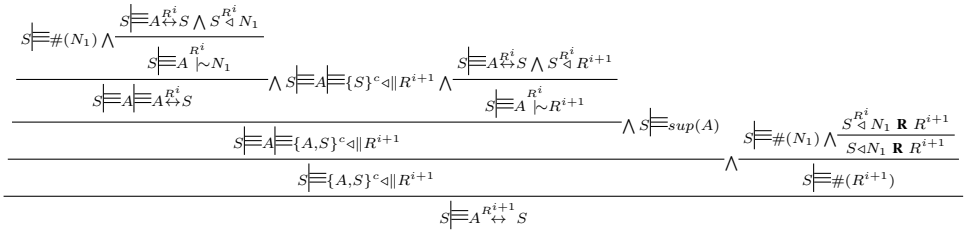


(c) Proof of "**A** believes N_b is a good shared key of **A** and **S**".

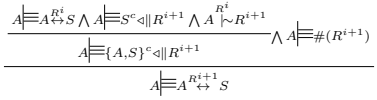


(d) Proof of "**S** believes N_b is a good shared key of **A** and **S**".

Fig. 6: Security proofs for secrecy.



(a) Proof of "**S** believes R^{i+1} is a good shared key of **A** and **S**".



(b) Proof of "**A** believes R^{i+1} is a good shared key of **A** and **S**".

Fig. 7: Security proofs for CRP update protocol

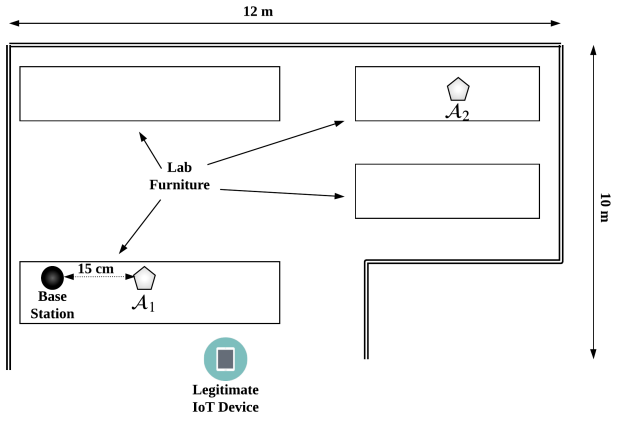


Fig. 8: Experiment Layout.

within our lab. Traces were gathered for RSSI values between the IoT device and base station; and the adversaries and base station for a period of one hour at a packet rate of 1 packet per 2 seconds. Matlab was used to analyze the resulting traces.

Figure 9 shows the *MSE* values for the legitimate channel and adversarial channels. We observe that the *MSE* values for adversarial channels are orders of magnitude larger than those for the legitimate channel. Note that to fit in the large *MSE* values into the plot in Figure 9, we plot *MSE* on the `log` scale. From Figure 9 we can determine the *MSE* threshold for detecting attacks. We observe that the *MSE* values for adversarial channel are always greater than 15 ($\texttt{log}_{10}15 = 1.1761$). Therefore, we use 15 as the threshold for *MSE*.

To evaluate the performance of the proposed technique and compare with [24] we use the following metrics:

1) **Probability of False Alarm**: the probability that a legitimate channel is classified as an adversarial channel.
2) **Probability of Missed Detection**: The probability that an adversarial channel is classified as a legitimate channel.

We evaluate the performance for three different sizes for the wireless fingerprints i.e., 16, 32, and 64 bytes. The results are shown in Table I, where P_{FA} denotes the probability of false alarm between the IoT device and base station, while, P_{MD_1}

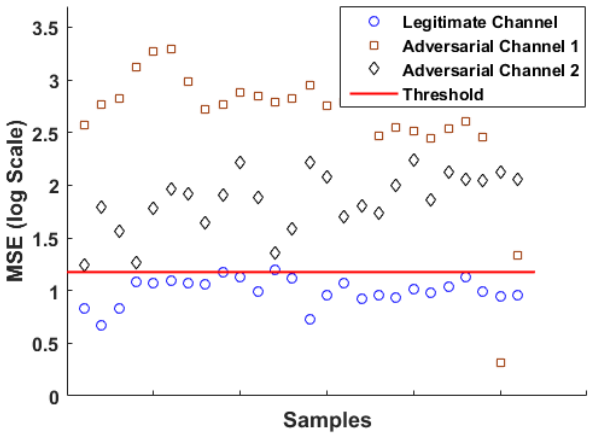


Fig. 9: Comparison of *MSE* for adversarial channels and legitimate channel

and P_{MD_2} represent the probability of missed detection for the channel between $\mathcal{A}_1$ and base station, and $\mathcal{A}_2$ and base station, respectively. Table I shows that the proposed technique can effectively detect attacks on data provenance and clearly outperforms the technique proposed by [24].

TABLE I: Miss classification rates for proposed protocol and reference [24] - high mobility

Finger Print Size	P_{FA} (%)			P_{MD_1} (%)			P_{MD_2} (%)		
	Proposed	[24]	% Impro-vement	Proposed	[24]	% Impro-vement	Proposed	[24]	% Impro-vement
16	3.8	34.6	89	3.7	7.7	52	0	33.3	100
32	0	20	100	0	0	0	0	25	100
64	0	8.3	100	0	0	0	0	0	0

IX. Conclusions

This paper present a protocol for establish data provenance in IoT systems using PUFs and wireless channel characteristics. Wireless fingerprints are derived from the wireless channel between two entities using RSSI measurements. The wireless fingerprints are used with a thresholding mechanism using mean squared error to detect attacks. A formal security analysis of the proposed protocol shows that it is robust against different types of attacks. An experimental validation on MICA-Z motes showed that the proposed wireless fingerprinting technique can improved the accuracy of detecting attacks by 100% as compared to existing techniques.

References

[1] D. Evans, "The internet of things how the next evolution of the internet is changing everything," https://www.cisco.com/c/dam/en us/about/ac79/docs/innov/IoT IBSG 0411FINAL.pdf, Apr 2011, accessed: 2018-10-01.

[2] M. Alioto (Ed.), *Enabling the Internet of Things from Integrated Circuits to Integrated Systems*, Springer, 2017.

[3] V. Varadharajan, S. Bansal, "Data Security and Privacy in the Internet of Things (IoT) Environment," in: Z. Mahmood (eds) *Connectivity Frameworks for Smart Devices, Computer Communications and Networks*. Springer, Cham, 2016.

[4] U. Chatterjee et al., "Building PUF based Authentication and Key Exchange Protocol for IoT without Explicit CRPs in Verifier Database," in *IEEE Trans. Dependable and Secure Computing*, preprint, May 2018.

[5] M. N. Aman, K. C. Chua and B. Sikdar, "Mutual Authentication in IoT Systems Using Physical Unclonable Functions," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1327-1340, Oct. 2017.

[6] S. Suhail et al., "Introducing Secure Provenance in IoT: Requirements and Challenges," in *Proc. SIoT*, Heraklion, 2016, pp. 39-46.

[7] E. Nwafor et al., "Towards a provenance collection framework for Internet of Things devices," in *Proc. IEEE SmartWorld,* San Francisco, CA, 2017, pp. 1-6.

[8] M. Elkhodr et al., "Data Provenance in the Internet of Things," in *Proc. WAINA,* Krakow, 2018, pp. 727-731.

[9] S. Suhail et al., "Data trustworthiness in IoT," in *Proc. ICOIN,* Chiang Mai, 2018, pp. 414-419.

[10] J. L. C. Sanchez, J. B. Bernabe and A. F. Skarmeta, "Towards privacy preserving data provenance for the Internet of Things," in *IEEE WF-IoT,* Singapore, 2018, pp. 41-46.

[11] M. N. Aman, et. al. "Physical Unclonable Functions for IoT Security." *In Proc. ACM IoTPTS*, New York, NY, USA, 10-13, 2016.

[12] P. Tuyls and L. Batina, "RFID-Tags for Anti-Counterfeiting," *In Proc. CT-RSA),* vol. 3860 of LNCS, pp. 115-131, Springer Verlag, February 2005.

[13] S. Guilley, and R. Pacalet, "SoCs security: a war against side-channels", *Annals of Telecommunications,* 59(7), pp. 998-1009, 2004.

[14] M. Kirkpatrick et. al., "System on Chip and Method for Cryptography using a Physically Unclonable Function," U.S. Patent 8750502 B2, issued March 22, 2012.

[15] W. C. Jakes. "Microwave Mobile Communications". Wiley, 1974.

[16] S. N. Premnath et al., "Secret key extraction using Bluetooth wireless signal strength measurements," in Proc. IEEE SECON, Singapore, 2014, pp. 293-301.

[17] R. Wilson, D. Tse and R. A. Scholtz, "Channel Identification: Secret Sharing Using Reciprocity in Ultrawideband Channels," in *IEEE Trans. on Inform. Forensics Sec.*, vol. 2, no. 3, pp. 364-375, Sept. 2007.

[18] N. Patwari et al., "High-Rate Uncorrelated Bit Extraction for Shared Secret Key Generation from Channel Measurements," in *IEEE Trans. Mobile Comput.,* vol. 9, no. 1, pp. 17-30, Jan. 2010.

[19] A. Kalamandeen et al., "Ensemble: Cooperative Proximity-based Authentication," in *Proc. ACM MobiSys*, San Francisco, CA, 2010, pp. 331344.

[20] J. Tang, P. Fan and X. Tang, "A RSSI-Based Cooperative Anomaly Detection Scheme for Wireless Sensor Networks," in *Proc. International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, 2007, pp. 2783-2786.

[21] S. Mathur et al., "ProxiMate: Proximity-based Secure Pairing using Ambient Wireless Signals," in *Proc. ACM MobiSys*, Bethesda, MS, 2011.

[22] Y. Chen et al., "Detecting and Localizing Identity-Based Attacks in Wireless and Sensor Networks," in *IEEE Trans. Veh. Tech.*, vol. 59, no. 5, pp. 2418-2434, Jun 2010.

[23] J. Yang et al., "Detection and Localization of Multiple Spoofing Attackers in Wireless Networks," in *IEEE Trans. Parallel and Distrib. Sys.*, vol. 24, no. 1, pp. 44-58, Jan. 2013.

[24] S. T. Ali et al., "Securing First-Hop Data Provenance for Bodyworn Devices Using Wireless Link Fingerprints," in *IEEE Trans. Inform. Forensics and Sec.*, vol. 9, no. 12, pp. 2193-2204, Dec. 2014.

[25] "TOTP: Time-Based One-Time Password Algorithm", IETF RFC 6238, 2011.

[26] W. Mao and C. Boyd, "Towards formal analysis of security protocols", *Proc. Comp. Sec. Foundations Workshop VI,* pp. 147-158, 1993.