A Light Weight Protocol for Secure Data Provenance in the Internet of Things using Wireless Fingerprints

Muhammad Naveed Aman, Mohamed Haroon Basheer, *Student Member, IEEE*
Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—Data provenance is an important security requirement to establish trust in the data produced by an IoT device. Existing works on data provenance for IoT are based on complex computations or costly hardware that may not be feasible for IoT systems. To solve this issue, this paper uses an analytical model to develop a threshold-based mechanism to establish data provenance in IoT systems. Moreover, using light weight security primitives, a light weight security protocol for data provenance is also proposed. The proposed protocol uses Physical Unclonable Functions (PUFs) and fingerprints extracted from the wireless channel to achieve data provenance, mutual authentication, and anonymity. The wireless fingerprints are generated using the link quality indicator (LQI) values. Experimental validation on MICA Z motes shows that the proposed technique can detect adversarial channels with high accuracy. Security analysis of the proposed protocol using formal proofs as well as simulations shows robustness against various types of attacks. Moreover, the energy requirements for the proposed protocol are shown to be significantly lower than existing protocols.**

***Index Terms*—Internet of Things, Data Provenance, Physical Unclonable Functions, Wireless Channel Characteristics, Link Quality Indicator, Data Provenance, Authentication.**

I. Introduction

The exponential growth of IoT devices in the near future producing large volumes of data may lead to many security and privacy issues. The most important security requirements for IoT include authentication, data provenance, and privacy. Data provenance establishes trust in the fidelity of data, i.e., that the data is actually collected at the location and time claimed by the specific IoT device. The reliable operation of IoT-based systems depends on the trustworthiness of the data produced by IoT devices [1]–[3]. For example, nuclear power plants may use IoT devices for monitoring and maintaining the pressure and temperature within a strict range. An attacker may invalidate this data by changing the location of an IoT device or cloning it.

This work was supported in part by the National Research Foundation, Prime Minister's Office, Singapore under its Corporate Laboratory@University Scheme, National University of Singapore, and in part by the Singapore Telecommunications Ltd.

M. N. Aman and M. H. Basheer are with the Department of Computer Science, National University of Singapore, 13 Computing Drive, Singapore 117417, e-mail: naveed@comp.nus.edu.sg, haroon.basheer@nus.edu.sg.

B. Sikdar is with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576, e-mail: bsikdar@nus.edu.sg.

The existing research on IoT data provenance is not exhaustive and most of these schemes are susceptible to impersonation, cloning, denial of service (DoS) and physical attacks. To solve these issues, this paper proposes a secure and lightweight protocol with privacy preservation for data provenance in IoT systems. This paper exploits the wireless channel characteristics between two entities to generate "wireless fingerprints" that are then used to provide data provenance. In particular, we use the Link Quality Indicator (LQI) values to identify the wireless link between two entities. The intuition behind creating wireless fingerprints is that the wireless channel between two communicating entities is intrinsically symmetric. However, according to Jake's fading model [4], if one of two communicating entities moves more than half of a wavelength, then the wireless channel de-correlates quickly and becomes independent for a distance exceeding one wavelength. This fact and the reciprocity attribute of electromagnetic wave propagation is used to derive security primitives from wireless channel characteristics. The theory behind our technique is as follows: An inherently symmetric wireless channel will always exist between two communicating entities, Alice and Bob, resulting in identical measurements such as delays, phase shifts, and gains. As a result, these measurements will be highly correlated, if taken separately by Alice and Bob, at their respective locations.

This paper proposes the use of Physical Unclonable Functions (PUFs) for hardware level authentication of IoT devices. PUFs provide the IoT devices with a unique hardware fingerprint by exploiting the inherent random variations at the physical (sub-)microscopic structure in an integrated circuit [5]. The data that many IoT devices produce is personal and sensitive in nature, requiring security protocols to achieve privacy preservation or anonymity. In this paper we use pseudonym identities constructed using PUF outputs and random numbers. This results in the proposed protocol being anonymous and secure against user identity profiling.

The overall process of the proposed data provenance technique is shown in Figure 1. An IoT device samples the LQI values of its wireless channel with a wireless gateway to generate a wireless fingerprint **F**. Similarly, the wireless gateway also generates a wireless fingerprint **F'** at its end. Both the entities send their respective wireless fingerprints to the verifier. The verifier can check the provenance of any data sent by the IoT device by comparing the two wireless fingerprints,

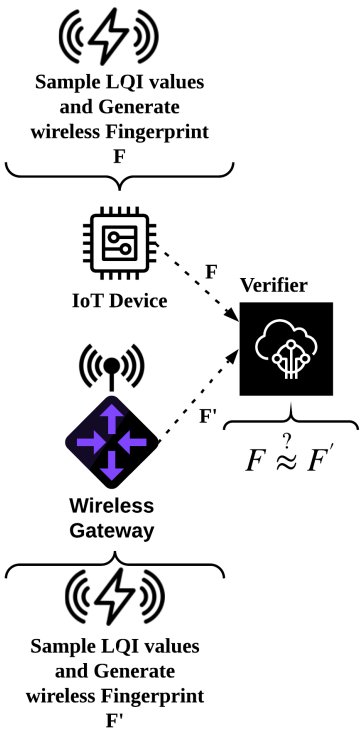


Fig. 1: Overview of proposed data provenance technique.

i.e., $F \overset{?}{\approx} F$. The comparison is based on the variance of the wireless fingerprints. This paper's major contributions are as follows:

- An analytical model based on wireless channel LQI measurements that can be used to differentiate between legitimate channels and adversarial channels. This results in data provenance with regard to the data location.
- Establishing data provenance with regard to the source of data using PUF based authentication protocols.
- Experimental validation of the proposed technique.

The rest of the paper is organized as follows. Related work and background are given in Sections II and III, respectively. Section IV discusses our network model, assumptions, and threat model. The proposed technique and protocol are presented in Sections V and VI, respectively. Section VII presents the formal security analysis and simulation results are presented in Section VIII. The implementation results and energy requirements are discussed in Sections IX and X, respectively. We conclude the paper in Section XI.

II. Related Work

The existing work on data provenance in IoT can be categorized into three categories: security-primitives based, hardware-based, and provenance using wireless channel characteristics. The security-primitives based data provenance techniques use filters, hash chains, blockchains, or zero-knowledge proofs (ZKP) to establish data provenance. The authors of [6] propose a data provenance technique for IoT devices using bloom filters and attribute based encryption. However, this technique requires IoT devices to store provenance information which may not be feasible as IoT devices has small memories. Moreover, an attacker can easily use physical attacks to tamper with the provenance information stored in an IoT device. In another work [7], provenance information is transmitted across multiple IoT devices using a hash chain based on identities. This technique is vulnerable to impersonation attacks as it relies on IoT device identities. The use of non-interactive zero-knowledge proofs (NI-ZKP) for data provenance is proposed in [8]. However, ZKP techniques may result in computationally complex solutions. The authors of [9] propose a data provenance compression algorithm. However, the proposed solution results in a computationally intensive. The recent techniques for data provenance using blockchains include [10]–[13]. However, these techniques result in higher computational overhead due to the use of blockchain. Hardware-based data provenance solutions use specialized hardware to establish data provenance such as trusted platform modules (TPM). One of the recent hardware-based data provenance technique is proposed in [14]. The authors in [14], propose a trust management system for IoT devices using data provenance. However, hardware-based techniques depends on specialized hardware which may not be available/feasible for IoT devices.

Wireless channel characteristics for security is a well studied and established area. The existing literature includes secret key generation [15], proximity based authentication [16], secure pairing [17], Sybil attack detection [18], and intrusion detection [19]. Using wireless channel characteristics for data provenance has been proposed in [20]. The authors in [20] generate unique wireless fingerprints in body area networks using the received signal strength indicator (RSSI) values. However, this technique suffers from high communication and computational overhead due to long wireless fingerprints and optimization. The authors in [21], [22] propose a multi-hop provenance protocol using the technique proposed in [20]. However, these protocols use RSSI values without an authentication mechanism. Thus, an attacker can easily spoof the RSSI values to hide its location.

We observe that the existing techniques for data provenance in IoT have one or more of the following problems:

1) Depend on **secure hardware** that is too expensive for IoT devices.
2) All devices must have the **same architecture**.
3) Rely on **complex computations** not feasible for simple IoT devices.
4) Vulnerable to **physical and cloning attacks** with **no privacy preservation**
5) Can be compromised using ephemeral secret leakage (ESL) attacks.

PUFs are commonly used for key generation and authentication [23]–[26]. However, PUFs have not been used to establish data provenance. This paper uses the following methods to solve the problems described above:

1) Developing an analytical model to establish data provenance without using any complex computations.
2) Eliminate the need for any specialized hardware ex-

TABLE I: Comparison of proposed technique with existing data provenance techniques.

Technique	[6]	[7]	[8]	[9]	[14]	[10]–[13]	[20]	[21], [22]	Proposed Technique
Computationally Complex	✗	✓	✓	✓	✗	✓	✗	✗	✗
Require Advanced Hardware	✗	✗	✗	✗	✓	✓	✗	✗	✗
Privacy Concerns	✓	✓	✓	✓	✓	✗	✓	✓	✗
Require Homogeneous Devices	✓	✓	✓	✗	✓	✓	✗	✗	✗
Physical Attacks	✓	✓	✓	✓	✗	✗	✓	✓	✗
ESL	✓	✓	✓	✓	✓	✓	✓	✓	✗

Computationally Complex: Does the technique use computationally intensive operations?

Require Advanced Hardware: Does the technique rely on costly hardware modules such as TPMs?

Privacy Concerns: Does the technique ensure IoT device anonymity?

Require Homogeneous Devices: Does the technique require all IoT devices to share the same architecture?

Physical Attacks: Can the technique be compromised using physical attacks?

ESL: Can the technique be compromised using ephemeral secret leakage?

cept for PUFs. Note that PUFs are extremely cheap to manufacture and can support ultra high throughput with extremely low energy and silicon area footprints [27].

3) Use of light weight symmetric key cryptography.
4) IoT devices do not store secrets in their memory.
5) PUFs establish trust in the origin of the data while wireless fingerprints institute trust in the location of the data, thus, providing data provenance.
6) Privacy preservation is achieved by hiding the actual identities of IoT devices using pseudonym identities.
7) Resilience against ESL attacks is achieved by combining the PUF output with the short term secrets to generate a session key. Thus, even if the attacker reveals the short term secrets, he/she can not calculate the session key.

To provide a comprehensive comparison of the proposed technique with existing literature, Table I provides a summary.

III. Background

A. Physical Unclonable Functions

Random variations within the fabricating process of integrated circuits give rise to an intractably complicated physical system enabling a novel challenge response mechanism. A PUF is characterized by a challenge-response-pair (CRP) i.e., $R = P(C)$, where R is the response to a challenge C by a PUF P. Every PUF produces a unique response when excited with the same challenge implying that each PUF is unique.

Environmental factors such as temperature and voltage may affect the output of a PUF to the same challenge. This problem can be avoided and we can get stable PUF responses good enough for security applications using fuzzy extractors [25]. Therefore, in this paper, we assume the use of ideal PUFs. IoT devices do not need to store secret keys in their memory when using PUFs which in turn safeguards them against physical attacks. Delay-based PUFs (exploiting circuit delay variation) and memory based PUFs (using the randomness in the power-up behavior of memory cells) are among the popular choices in security applications.

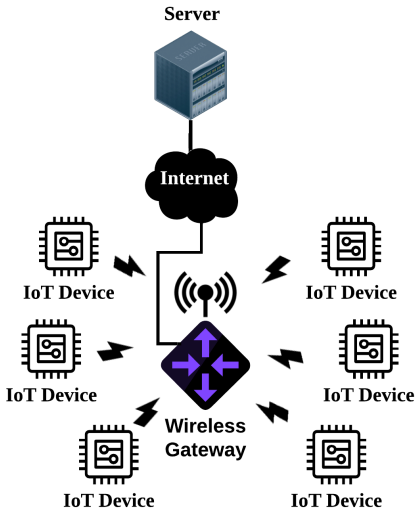


Fig. 2: Network model.

IV. Network Model, Assumptions And Threat Model

A. Network Model

We consider multiple IoT devices sending data to a server through a wireless gateway connected to the Internet as shown in Figure 2.

B. Assumptions

a. Every IoT device has a PUF and is considered a system-on-chip (SoC). The PUF is assumed to be useless and destroyed if separated from the IoT device [28].
b. The micro-controller and the PUF form a SoC and the communication between them is considered secure [28].
c. IoT devices are constrained in terms of memory, energy, and processing capabilities. However, the server is not resource constrained.
d. Table II gives the set of notations used in this paper.

TABLE II: Notations

Notation	Description
PUF	Physical Unclonable Function
x_n	n-th time domain reference signal
y_n	n-th time domain received OFDM signal
e_n	Error vector
N	Number of OFDM symbols
H_n	Rayleigh distributed channel coefficients
η_n	Additive white Gaussian noise (AWGN)
σ_η^2	Variance of AWGN
$\mathcal{N}$	Normal distribution
μ	Mean value
α	Path loss exponent
$\texttt{Var}(x)$	Variance of x
F_x	Wireless fingerprint for principal x
Δ	Threshold for detecting attacks
ID_i	ID of the IoT device
$H(X)$	Hash of X
$\|$	Concatenation operator
$\{M\}_k$	Message M is encrypted using key k
SID_A^i	Pseudonym identity of IoT device ID_A for the i-th iteration
C^i	Challenge for the i-th iteration
R^i	Response of the respective PUF for C^i
P_{FA}	Probability of false alarm
P_{MD}	Probability of missed detection

C. Threat Model

After authenticating with the server, an IoT device starts transmitting data packets to the server. The adversary may inject, replay, tamper and eavesdrop on packets sent by an IoT device. The proposed protocol is based on the CK-adversary model [29]. Under the CK-adversary model, the adversary is capable of revealing the session state, private, and session keys in addition to the capabilities under the DY model. We also assume that an adversary may gain physical access to an IoT device and subject it to physical attacks to extract stored secrets. The following set of queries can be used to model these attacks:

- `Send`S(S, `m0,r0,m1`) models the query where the adversary $\mathcal{A}$ attempts to impersonate a legitimate IoT device by sending a message `m0` to the server S. The server then replies with `r0` and the IoT device then sends `m1` to server S.
- `Send`ID(ID, `m0,r0`) models the query where the adversary $\mathcal{A}$ tries to impersonate a server by sending a message `m0` and receiving `r0` from an IoT device.
- `Monitor`(ID, S) models the adversary's capability to observe and eavesdrop the wireless channel between IoT device ID and server S.
- `Drop`($\mathcal{A}$) models the query where the adversary can drop packets between ID and S. An adversary may use this query to interrupt the synchronization between two parties by selectively dropping packets.
- `Reveal`(ID) models the adversary's ability to extract the secrets stored in an IoT device's memory using a physical attack.

The queries `Send`S, `Send`ID, `Monitor`, and `Drop` can be invoked by the adversary any polynomial number of times. Note that any attempt to physically alter an IoT device makes it useless. Therefore, `Reveal` can be called by $\mathcal{A}$ only once.

The proposed protocol is designed to achieve mutual authentication, data provenance (source and location), privacy preservation, and security against DoS and physical attacks.

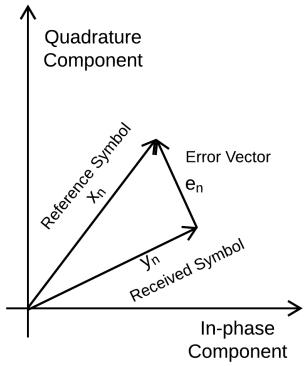


Fig. 3: Illustration of Error Vector

V. Proposed Data Provenance Technique

LQI is the average of the error between ideal constellations and the received signal over 64 symbols immediately after the sync word [30]. Let us represent LQI as:

$$L = \frac{1}{N \cdot P_0} \sum_{n=0}^{N-1} |y_n - x_n|^2 = \frac{1}{N \cdot P_0} \sum_{n=0}^{N-1} |e_n|^2, \quad (1)$$

where, y_n is the received time domain OFDM signal, x_n is $n-$th time domain reference signal, and e_n is the error vector, for $0 \leq n \leq N-1$ OFDM symbols, as shown in Figure 3. P_0 is the average symbol power for a given modulation and it makes LQI independent of the modulation order. The received time domain OFDM signal y_n can be represented by

$$y_n = H_n x_n + \eta_n \quad (2)$$

where H_n represents the Rayleigh distributed channel coefficients, and η_n is the additive white Gaussian noise (AWGN) with zero-mean and σ_η^2 variance.

For a large number of sub-carriers, x_n is approximately independent and identically distributed (i.i.d) Gaussian distributed with zero mean and σ_x^2 variance [31], [32]. For a large N, we can use the central limit theorem to approximate L as a Gaussian random variable, i.e., $L \sim \mathcal{N}(\mu_L, \sigma_L^2)$. Thus, to characterize L we need to find its mean μ_L and variance σ_L^2. Assuming the standard path loss law $l(r) = \frac{1}{r_i^\alpha}$ with path

a path loss exponent of α, and using $r_i^{-\alpha}$ as the mean power for H_n, we get μ_L as follows:

$$\begin{aligned}\mathtt{E}[L] = \mu_L &= \frac{1}{N \cdot P_0} \sum_{n=0}^{N-1} \mathtt{E}[|e_n|^2] \\ &= \frac{1}{P_0}\left[\sigma_x^2\left(\frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}}\right) + \sigma_\eta^2\right] \quad (3)\end{aligned}$$

where r_i denotes the distance between the IoT node and the wireless gateway. Moreover, $\mathtt{E}[|e_n|^2]$ is given as follows:

$$\begin{aligned}\mathtt{E}[e_n^2] &= \mathtt{E}[(y_n - x_n)^2] \\ &= \mathtt{E}[(H_n x_n + \eta_n - x_n)^2] \\ &= \sigma_x^2\left\{\frac{2}{r_i^\alpha} + 1 - 2\sqrt{\frac{\pi}{2r_i^\alpha}}\right\} + \sigma_\eta^2. \quad (4)\end{aligned}$$

To find the variance of L, we have $\sigma_L^2 = E[L^2] - (\mu_L)^2$. To find $E[L^2]$ we proceed as follows:

$$L^2 = \left(\frac{1}{NP_0}\sum_{n=0}^{N-1}|e_n|^2\right)^2 = \frac{1}{N^2P_0^2}\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}|e_{n_1}|^2|e_{n_2}|^2. \quad (5)$$

Assume block fading with m symbols per block. Then we get the expectation of L^2 as:

$$\mathtt{E}[L^2] = \frac{1}{NP_0^2}\left[m\mathtt{E}[e_n^4] + (N-m)\left(\mathtt{E}[e_n^2]\right)^2\right]. \quad (6)$$

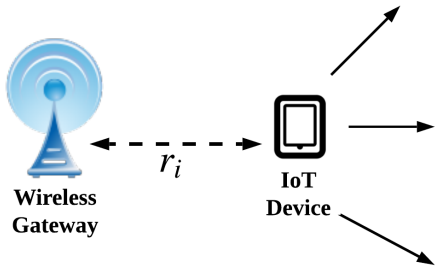
We can now obtain the variance σ_L^2 using (3) and (6). Note that $\mathtt{E}[e_n^4]$ can be obtained using a similar procedure as (4). If we plot the probability density function (pdf) of L at five different locations slowly moving away from a wireless gateway (as shown in Figure 4(a)), we get Figure 4(b). We observe that the mean μ_L remains approximately unchanged. However, the variance of L for different locations varies. We exploit this fact in this paper by comparing the variance of LQI at the legitimate IoT device with the variance of LQI at the wireless gateway. Note that the two measurements should be in high agreement.

Let us consider the scenario in Figure 5. Two entities Alice and Bob are talking to each other. Alice is an IoT device and Bob is the wireless gateway. Two adversaries located nearby but at least one wavelength away from Alice and Bob try to send tampered data to the gateway. An adversarial channel between Alice anb Bob can be detected using the following steps:

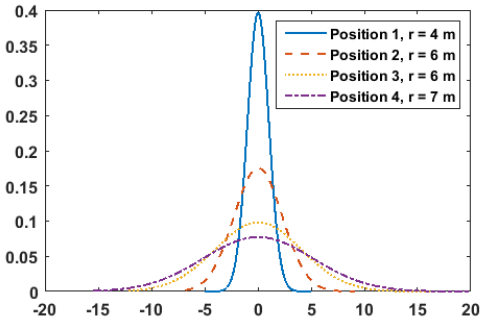
1) Alice and Bob sample the LQI values for the wireless channel between them to generate their respective wireless fingerprints.
2) Alice and Bob send their wireless fingerprints to a verifier.
3) The verifier calculates the variance of each wireless fingerprint and takes the difference between the two variances i.e.,

$$\Delta = \mathtt{Var}(F_{\text{Alice}}) - \mathtt{Var}(F_{\text{Bob}}) \quad (7)$$

where, F_{Alice} and F_{Bob} denote the wireless fingerprints for Alice and Bob, respectively. $\mathtt{Var}$ represents the variance operation. The server then compares Δ to a threshold value.



(a) Experimental Schematic



(b) pdf of L

Fig. 4: Detection of adversarial channels using L.

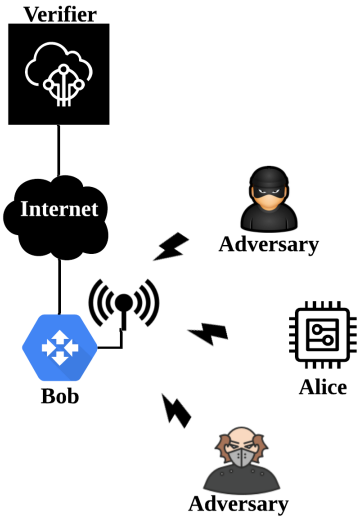


Fig. 5: Attack Scenario.

4) The wireless link between Alice and Bob is considered legitimate if Δ is less than the threshold. Otherwise, the channel between Alice and Bob is considered compromised and the data is discarded.

Using experiments, we determine the threshold value for Δ in Section IX.

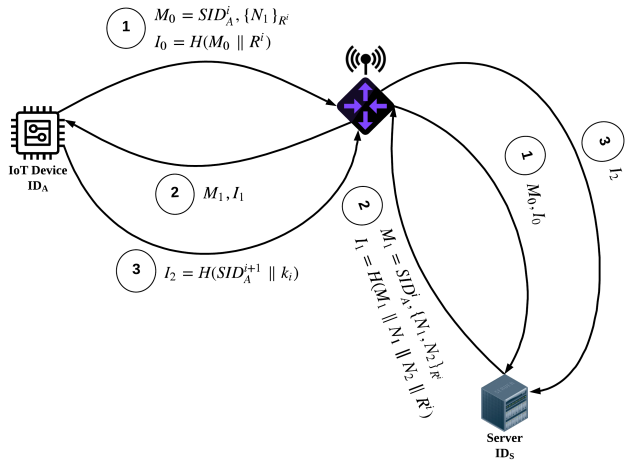


Fig. 6: Authentication Phase.

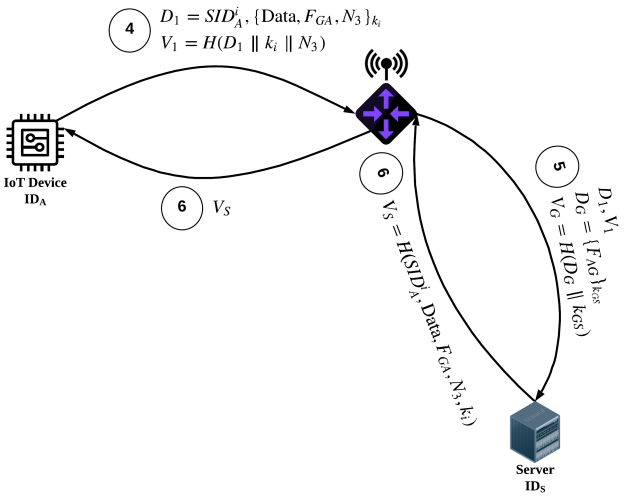


Fig. 7: Data Transfer Phase.

VI. Proposed Data Provenance Protocol

A. Device Registration

The server stores an initial CRP (C^i, R^i) and pseudonym identity (SID^i) for each IoT device. For each IoT device, the server also stores a an emergency CRP list (C_{em}) and an emergency identity list (EID) to mitigate denial of service (DoS) attacks. The initial parameters are obtained by the server using a time-based one-time password algorithm (TOTP) [33] and an operator using a password. Each IoT device stores C^i, SID^i, C_{em}, and EID. We assume that the server and wireless gateway have a pre-shared secret symmetric key k_{GS}.

B. Authentication Phase

The authentication phase of the proposed protocol is shown in Figure 6. In this figure IoT device ID_A intends to send data to the server via wireless gateway ID_G. Following are the steps for the authentication phase:

1) IoT device ID_A uses the stored challenge C^i and its PUF to generate the secret response R^i. The IoT device ID_A then generates a random nonce N_1 and sends a Message M_0 along with an authentication parameter I_0 to the server via a wireless gateway ID_G as shown in Message ① of Figure 6. We use authentication parameters to ensure data integrity of messages in this paper. An authentication parameter consists of a cryptographically secure hash of a message concatenated with freshness identifiers and a secret key. The receiver of an authentication parameter can verify the integrity of the message by calculating the hash using secrets stored in its memory. The two hashes (received and calculated) should be equal. Note that throughout this paper if an entity fails to verify an authentication parameter the protocol is terminated.
2) The wireless gateway ID_G forwards Message ① to the server after sampling the wireless fingerprint F_{AG}.
3) The server searches its memory for SID_A^i and reads the corresponding CRP (C^i, R^i). The server then uses I_0 to verify the integrity of Message ①. The server generates a random nonce N_2 and uses R^i to obtain N_1. It then sends Message $M_1 = SID_A^i, \{N_1, N_2\}_{R^i}$ along with the corresponding authentication parameter I_1 to the IoT device ID_A in Message ② in Figure 6.
4) On receiving Message ②, the IoT device ID_A samples the wireless channel to generate the wireless fingerprint F_{GA}. It then uses R^i to obtain N_2 and verifies I_1. The IoT device ID_A generates the session key $k_i = H(N_1 \oplus N_2) \oplus H(ID_A \oplus R^i)$ and updates its pseudonym identity $SID_A^{i+1} = H(ID_A \parallel N_1 \parallel R^i)$. It then sends Message ③ to the server as an acknowledgment as shown in Figure 6.
5) The server generates the new pseudonym identity $SID_A^{i+1}$and verifies I_2. The authentication is considered complete and the server stores SID_A^{i+1} for any future authentications.

C. Data Transfer Phase

After an IoT device successfully authenticates itself to the server, it can now send data using the following steps.

1) A random nonce N_3 is generated by IoT device ID_A. It then creates a Message $D_1 = SID_A^i, \{Data, F_{GA}, N_3\}_{k_i}$. It then sends D_1 along with the corresponding authentication parameter V_1 to the server through the wireless gateway in Message ④ in Figure 7.
2) After receiving Message ④, the wireless gateway creates Message $D_G = \{F_{AG}\}$ and the corresponding authentication parameter V_G. It then sends Message ④ along with D_G and V_G to the server.
3) The server uses k_i and k_{GS} to decrypt D_A and D_G, respectively, to get the data and wireless fingerprints. The authentication parameters V_1 and V_G are then verified and using the wireless fingerprints it validates the data provenance by employing the technique described in Section V. The server rejects the data if validation fails,

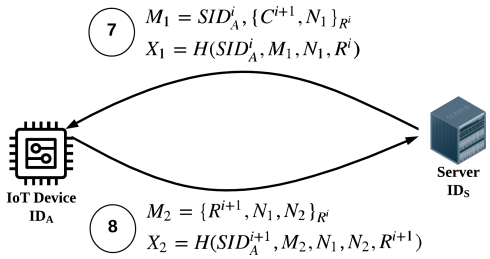


Fig. 8: Protocol for CRP update.

i.e., if $F_{AG} \neq F_{GA}$. Otherwise, the server accepts the data and sends an authentication parameter V_S to the IoT device ID_A as an acknowledgment in Message ⑥ in Figure 7.

4) After receiving Message ⑥, IoT device ID_A verifies V_S. The IoT device may resend the data if verification fails. Otherwise, the IoT device ID_A may send additional data using the same steps as above or the session may be concluded.

D. CRP Update

The server maintains a list of CRPs, i.e, one CRP per IoT device. The server may update the corresponding CRPs to ensure freshness by obtaining new CRPs. The protocol for CRP update is shown in Figure 8. The steps of this protocol are as follows:

1) To update the CRP for IoT device ID_A, the server sends Message ⑦ (with the new challenge C^{i+1}) to the device as shown in Figure 8.
2) The IoT device ID_A decrypts M_1 in Message ⑦ to obtain C^{i+1} and N_1, and verifies the authentication parameter X_1. The IoT device ID_A stores the new challenge C^{i+1}, and uses it to generate the new response R^{i+1}. It then generates a random nonce N_2 and generates the new pseudonym identity $SID_A^{i+1} = H(ID_A \parallel N_2 \parallel R^{i+1})$. The IoT device then sends Message $M_2 = \{R^{i+1}, N_1, N_2\}_{R^i}$ along with the corresponding authentication parameter to the server in Message ⑧ in Figure 8.
3) The server decrypts M_1 to obtain R^{i+1} and N_2. It then uses N_2 to generate the new pseudonym identity SID_A^{i+1} and verifies X_2. The server replaces the CRP for IoT device ID_A with (C^{i+1}, R^{i+1}).

VII. Security Analysis

Lemma 1. *The behavior of a PUF cannot be predicted.*

Proof. Every PUF produces a unique response and cannot be cloned [34]. If a PUF is excited by a challenge of length l_1, it produces a response of length l_2, i.e., $\{0,1\}^{l_1} \rightarrow \{0,1\}^{l_2}$. We model the security of a PUF with a security game $\text{Exp}_{PUF,\mathcal{A}}^{\text{Sec}}$ between an adversary $\mathcal{A}$ and challenger $\mathcal{C}$ as follows:

(i) $\mathcal{A}$ sends a randomly chosen challenge C^i to $\mathcal{C}$. $\mathcal{C}$ uses the PUF to reveal R^i to $\mathcal{A}$.
(ii) $\mathcal{C}$ uses another randomly chosen challenge C^x (not used before) to obtain the response R^x using the PUF, i.e., $R^x = PUF(C^x)$.
(iii) $\mathcal{A}$ is allowed to query the PUF using challenges other than C^x a polynomial number of times.
(iv) $\mathcal{A}$ reveals its guess $R^{x'}$ for the challenge C^x and wins the game if $R^{x'} = R^x$.

The adversary's advantage in this game is given by $\text{Adv}_{\mathcal{A}}^{PUF} = \Pr[R^{x'} = R^x]$. The adversary can only guess the output of a PUF to a given challenge. Therefore, $\text{Adv}_{\mathcal{A}}^{PUF} = \frac{1}{2^{l_2}}$. □

Lemma 2. *An adversary cannot predict a wireless fingerprint.*

The adversary is assumed to be located at least a single wavelength away from a legitimate IoT device. Therefore, the wireless channel seen by the adversary is independent of the one seen by the IoT device. Thus, it is not possible for the adversary to infer the wireless fingerprints between a legitimate IoT device and a wireless gateway. We model the security of the wireless fingerprints using the security game $\text{Exp}_{FP,\mathcal{A}}^{\text{Sec}}$ as follows:

(i) $\mathcal{C}$ randomly chooses an IoT device ID_1.
(ii) $\mathcal{C}$ obtains the wireless fingerprint F_{1G} between ID_1 and the wireless gateway by initiating a communication session between them.
(iii) $\mathcal{A}$ is allowed to acquire wireless fingerprints by initiating communication sessions with ID_1 and the wireless gateway a polynomial number of times. However, $\mathcal{A}$ should be located at least a single wavelength away from the two entities.
(iv) $\mathcal{A}$ reveals its guess F_{1G}^* for the wireless fingerprint between ID_1 and the wireless gateway and wins the game if $F_{1G}^* = F_{12}$.

The adversary's advantage in this game can be modeled as $\text{Adv}_{\mathcal{A}}^{FP} = \Pr[F_{1G}^* = F_{1G}]$. The adversary can only guess the wireless fingerprint between ID_1 and the wireless gateway. Therefore, for a fingerprint size of f, $\text{Adv}_{\mathcal{A}}^{FP} = \frac{1}{2^f}$.

Lemma 3. *The* `Reveal` *oracle cannot be used to extract the secrets used in the proposed protocol.*

Proof. The IoT device only stores the current challenge C^i, pseudonym identity SID_{ID}^i and the emergency identities EID and challenges C_{em} lists. Thus, the IoT device does not store any secret in its memory, and the adversary cannot obtain the secret response R^i even by invoking the `Reveal` oracle. It is worth noting that given the SoC assumption, $\mathcal{A}$ cannot obtain R^i using C^i or C_{em}. □

Lemma 4. *An IoT device's pseudonym identities cannot be correlated even by invoking the* `Reveal` *oracle.*

Proof. The pseudonym identity SID_{ID}^i of an IoT device is constructed as $H(ID_A, N_a, R^i)$, i.e., using a random nonce N_a which is refreshed after each round. Therefore, each pseudonym identity is valid for only a single round. Thus, it is not possible for the adversary to correlate the pseudonym identity for the current round with that of the next or previous round unless he/she can obtain the secret response R^i. However, according to Lemma 3, this is not possible. The

advantage of the adversary in this case can be modeled as $\text{Adv}_{\mathcal{A}}^{ID} = \texttt{Pr}[\texttt{Corr}(SID^i, SID^{i+1}) \neq 0] \approx 0$, where Corr represents the correlation coefficient. □

Theorem 5. *Mutual Authentication: A successful run of the protocol between an IoT device and a server is only possible if both entities are legitimate.*

Proof. An adversary may attempt to authenticate itself to the server by impersonating a legitimate IoT device. The following game between $\mathcal{C}$ and $\mathcal{A}$ is used to model this attack.

1) $\mathcal{C}$ chooses a legitimate IoT device ID_1 to run the proposed protocol with the server.
2) $\mathcal{A}$ queries the server and the IoT device ID_1 a polynomial number of times using SendID, Send$S$, Drop, and Monitor.
3) $\mathcal{A}$ tries to authenticate itself as a legitimate IoT device by calling the Send$S$ oracle.
4) $\mathcal{A}$ can win the game by successfully completing the proposed protocol's authentication phase.

$\mathcal{A}$ can only successfully authenticate itself if it can produce the correct authentication parameter $I_2 = H(SID_1^{i+1} \parallel k_i)$. To do so, $\mathcal{A}$ needs R^i. Assume $\mathcal{A}$ can reveal l_2' bits (out of l_2 bits) in R^i. Then the advantage of $\mathcal{A}$ in revealing R^i is given by $\texttt{Pr}[R^{i'} = R^i] = \frac{1}{2^{l_2 - l_2'}}$. We can model the adversary's advantage for successfully authenticating itself to the server as $Adv_{\mathcal{A}}^{Auth1} = \texttt{Pr}[R^{i'} = R^i] - \text{Adv}_{\mathcal{A}}^{PUF}$. However, by Lemmas 1 and 3, $\mathcal{A}$ can only randomly guess R^i, i.e., $l_2' = 0$ and $\texttt{Pr}[R^{i'} = R^i] = \frac{1}{2^{l_2}} = \text{Adv}_{\mathcal{A}}^{PUF}$. thus, $Adv_{\mathcal{A}}^{Auth1} = 0$.

$\mathcal{A}$ may also try to impersonate a server and authenticate itself to an IoT device. We denote the advantage of the adversary in this attack by $Adv_{\mathcal{A}}^{Auth2}$. Using a similar approach as above, we get $Adv_{\mathcal{A}}^{Auth2} = 0$.

□

Theorem 6. *Data Provenance: If the IoT device and server successfully complete a run of the proposed protocol then the source and location of the data is indeed true.*

Proof. The adversary $\mathcal{A}$ may try to invalidate the data sent to the server. The security game between $\mathcal{C}$ and $\mathcal{A}$ is given as follow:

1) $\mathcal{C}$ uses an IoT device ID_1 and the server to launch the proposed protocol.
2) $\mathcal{A}$ queries the server and the IoT device ID_1 a polynomial number of times using SendID, Send$S$, Drop, and Monitor.
3) $\mathcal{A}$ ties to impersonate an IoT device by invoking the Send$S$ oracle.
4) $\mathcal{A}$ wins the game if the server accepts the tampered data sent by $\mathcal{A}$.

$\mathcal{A}$ must generate a valid authentication parameter V_1 in order to pass the data integrity check during the proposed protocol's data transfer phase. To do so, $\mathcal{A}$ needs R^i and the valid wireless fingerprint F_{1G}. The advantage of the adversary is given by $Adv_{\mathcal{A}}^{Prov} = \Big[(\texttt{Pr}[R^{i'} = R^i] - \text{Adv}_{\mathcal{A}}^{PUF}) \times (\texttt{Pr}[F_{1G}* = F_{1G}] - \text{Adv}_{\mathcal{A}}^{FP})\Big]$. However, by Lemmas 1 and 3, $\mathcal{A}$ can only randomly guess R^i, i.e., $\texttt{Pr}[R^{i'} = R^i] = \text{Adv}_{\mathcal{A}}^{PUF}$. Similarly, by Lemma 2, $\texttt{Pr}[F_{1G}^* = F_{1G}] = \text{Adv}_{\mathcal{A}}^{FP}$. Thus, $Adv_{\mathcal{A}}^{Prov} = 0$. □

Theorem 7. *Privacy: The proposed protocol achieves anonymity of the IoT devices.*

Proof. If two successful runs of the proposed protocol by the same IoT device with the server cannot be correlated by $\mathcal{A}$, then the proposed protocol is termed untraceable. The following security game can be used to model this attack:

1) $\mathcal{C}$ chooses two IoT devices ID_1 and ID_2 and uses each one of them to launch the proposed protocol with the server.
2) $\mathcal{A}$ queries the server and the IoT devices a polynomial number of times using Send$S$, SendID, Monitor, and Drop.
3) $\mathcal{C}$ randomly chooses one of the IoT devices $ID*$.
4) $\mathcal{A}$ queries the server and the IoT device $ID*$ a polynomial number of times using Send$S$, SendID, Monitor, and Drop.
5) $\mathcal{A}$ announces her/his guess ID'.
6) if $ID' = ID*$, $\mathcal{A}$ wins the game.

The adversary's advantage of a successful guess for ID' can be modeled as $Adv_{\mathcal{A}}^{Pri1} = 2 \times (\texttt{Pr}[ID' = ID*] - \frac{1}{2})$. By lemma 4, the advantage of $\mathcal{A}$ in correlating the pseudonym identities of $ID*$ can be modeled as $\text{Adv}_{\mathcal{A}}^{Pri2} = \texttt{Pr}[\texttt{Corr}(SID^i, SID^{i+1}) \neq 0]$. The advantage of the adversary in winning this game can be modeled as $Adv_{\mathcal{A}}^{Pri} = Adv_{\mathcal{A}}^{Pri1} + Adv_{\mathcal{A}}^{Pri2} - Adv_{\mathcal{A}}^{Pri1} \times Adv_{\mathcal{A}}^{Pri2}$. If an adversary makes a random guess for $ID*$ then he/she has no advantage, i.e., $\texttt{Pr}[ID' = ID*] = \frac{1}{2}$. Using Lemmas 1, 3, and 4, we can conclude that $Adv_{\mathcal{A}}^{Pri} = 0$.

□

Lemma 8. *The proposed protocol is protected against DoS attacks.*

Proof. An adversary $\mathcal{A}$ may attempt to break the synchronization between an IoT device and the server by blocking/dropping specific packets, e.g., Message ③ in Figure 6 and Message ② in Figure 8. However, the IoT device stores an emergency identity list while the server stores a list of emergency identities as well as CRPs to deal with these kinds of situations. □

Lemma 9. *The proposed protocol is secure against physical and cloning attacks.*

Proof. IoT devices do not store any secrets in their memory. Furthermore, the PUF and the micro-controller are inseparable and it is not possible to eavesdrop on their communication [28]. Thus, according to Lemmas 1 and 3, we can conclude that the proposed protocol is safe against physical and cloning attacks. □

Lemma 10. *The proposed protocol is safe against ephemeral secret leakage attacks.*

Proof. The IoT device and server establish a common session key $k_i = H(N_1 \oplus N_2) \oplus H(ID_A \oplus R^i)$ during the authentication phase, where N_1 and N_2 are short term secrets.

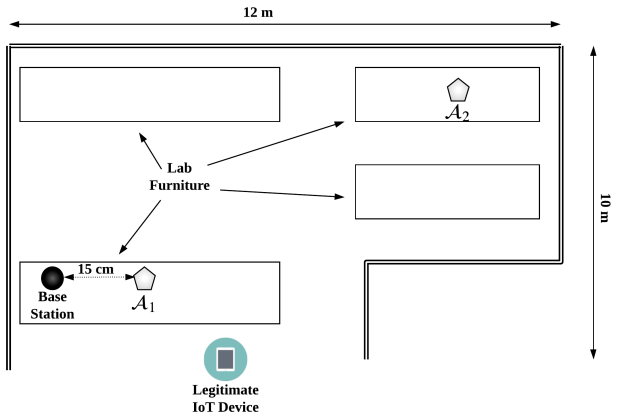


Fig. 9: Experiment Layout.

Assume that $\mathcal{A}$ has revealed the short term secrets. However, it is computationally infeasible for $\mathcal{A}$ to calculate k_i with out knowledge of ID_A and R^i. Thus, according to Lemmas 1 and 4, it is evident that the proposed protocol is resilient against ESL attacks. □

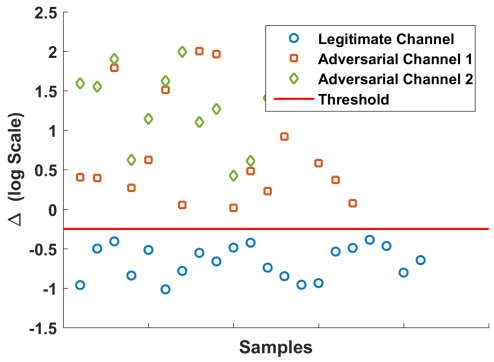
VIII. Security Verification and Simulations

ProVerif (PV) [35], an automated security verification tool, was used to perform rigorous simulations and experimentation to verify the security properties. ProVerif has been used to check the proposed protocols against the following security properties: mutual authentication, impersonation attack resistance, data tampering attack resistance, ephemeral secret leakage attack resistance, perfect forward secrecy, strong secrecy, and strong anonymity. The PV simulation scripts with the implementation source code can be found in [36].

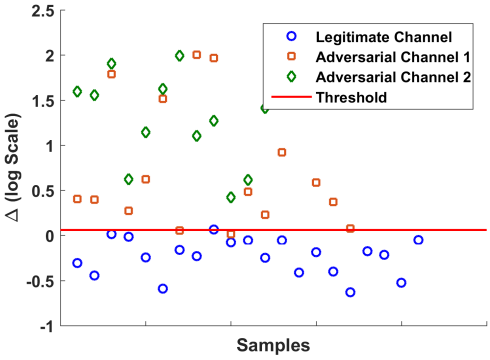
IX. Experimental Validation

We used MICA-Z motes with the CC2420 transceiver to conduct our experiments. These motes can communicate using the IEEE 802.15/Zigbee protocol and can output 8-bit unsigned LQI values. The experimental setup includes a typical indoor laboratory environment with furniture and WiFi devices including a base station, an actual IoT device, and two attackers $\mathcal{A}_1$ and $\mathcal{A}_2$ as shown in Figure 9. Note that the distance between the adversaries and the legitimate IoT device is greater than at least a single wavelength.

We conducted two sets of experiments: firstly, the IoT device can move around to different locations inside the laboratory area called *High Mobility*, and secondly, the IoT device moves sporadically inside a small space in the laboratory called *Low Mobility*. The wireless channel between the IoT device and the base station, and the adversaries and the base station was monitored for a duration of one hour and the corresponding Δ values are shown in Figures 10(a) and 10(b) for 32-byte wireless fingerprints. We observe a clear distinction between the Δ values of the adversarial channels and the legitimate channel. We observe threshold values of 0.9441 ($\log_{10}0.9441 = -0.25$) and 1.0233 ($\log_{10}1.0233 = 0.1$) for the low mobility and high mobility scenarios, respectively.



(a) Low Mobility



(b) High Mobility

Fig. 10: Comparison of Δ values for legitimate channel and adversarial channels using 32-byte fingerprints.

We compare the accuracy of our technique with the state-of-the-art work in [20]. The Pearson correlation coefficient r is used by the technique in [20] with a threshold value of 0.9 for r [20]. The comparison is made using two performance metrics, i.e., the probability of false alarm and the probability of missed detection. The probability that a legitimate channel is mistakenly flagged as an adversarial channel is termed as the probability of false alarm. Similarly, probability of failing to detect an adversarial channel in termed as probability of missed detection.

The results for the two scenarios considering three different sizes of the wireless fingerprints, i.e., 16, 32, and 64 bytes are given in Tables III and IV. Where P_{FA} denotes the probability of false alarm for the channel between the legitimate IoT device and base station, i.e., the ratio of the number of times the proposed technique flagged a legitimate channel as compromised to the total number of times the channel was checked. Similarly, P_{MD_1} represents the probability of missed detection for the channel between $\mathcal{A}_1$ and base station, i.e., the number of times the proposed technique failed to detect $\mathcal{A}_1$ to the total number of times the channel was checked. Similarly, P_{MD_2} represents the probability of missed detection for the channel between $\mathcal{A}_2$ and base station.

For the low mobility scenario, we observe that for a fingerprint size of 32 bytes or more, the proposed technique made no errors in classifying the wireless channels. However, the

technique proposed in [20], has a high P_{FA} of 40% even when the fingerprint size is 64 bytes. This shows that the performance of the proposed technique is significantly better than the technique in [20]. We observe similar results for the high mobility scenario where the proposed technique can accurately detect adversarial channels when the fingerprint size is 64 bytes or more. However, we observe a 70% P_{FA} for [20] even with 64 bytes fingerprints. Note that the fingerprints must be 2392 bytes long in [20] in order to achieve comparable accuracy. This further shows the superiority of proposed technique over the technique in [20].

TABLE III: comparison of the proposed protocol with reference [20]: low mobility.

Finger-print Size	P_{FA} (%)			P_{MD_1} (%)			P_{MD_2} (%)		
	Prop-osed	[20]	% Impro-vement	Prop-osed	[20]	% Impro-vement	Prop-osed	[20]	% Impro-vement
16	0	67.4	100	8.5	11.1	23.4	0	12.5	100
32	0	57.1	100	0	5.3	100	0	11.8	100
64	0	40	100	0	2.6	100	0	0	0

TABLE IV: Comparison of the proposed protocol with reference [20]: high mobility.

Finger-print Size	P_{FA} (%)			P_{MD_1} (%)			P_{MD_2} (%)		
	Prop-osed	[20]	% Impro-vement	Prop-osed	[20]	% Impro-vement	Prop-osed	[20]	% Impro-vement
16	19.05	66.67	71	12.5	12.5	0	7.6	33.3	77.1
32	18.1	63.6	71.5	5.88	6.7	12.2	0	25	100
64	0	70	100	0	0	0	0	0	0

X. Energy Requirements

The AVRORA energy analysis tool was used to evaluate the energy requirements of the proposed protocol on the MICA 2 mote platform. The proposed protocol was also compared with a new data provenance protocol for IoT by Sanchez et al. [8] with regard to energy consumption. Note that a full protocol description with authentication, data integrity, and privacy preservation is missing in [20]. Therefore, we do not consider [20] in this section.

The results for the average energy consumption for the CPU and radio subsystems for 100 runs of the protocol are shown in Table V for 128, 192, and 256 bits key sizes. The wireless fingerprints are considered to be 64 bytes long. The CPU and radio subsystem energies include the energy consumed by the security sub-system, as well as other tasks including boot, idle state, etc.

We observe that the proposed protocol consumes 83% and 73.5% less CPU and radio energy, respectively, than the protocol in [8] for a key size of 256 bits. This reduced energy consumption shows that the proposed protocol has significantly lower computation complexity. Moreover, the higher energy consumption in the other tasks column in Table V shows that the higher computational complexity of [8] results in the radio subsystem staying active for a longer period of time. Similarly, if the key size is increased by 64 bits, the CPU and radio energy consumption is increased by 45,347 μJ and 57,676 μJ, respectively, for [8]. However, for the proposed protocol we observe a mere increase of 110 μJ and 85 μJ. This shows that the proposed protocol has significantly lower energy requirements.

TABLE V: Energy Consumption

Key Size	Protocol Proposed by [8]			Proposed Protocol			Total Impro-vement %
	Protocol μJ	Other tasks μJ	Total μJ	Protocol μJ	Other tasks μJ	Total μJ	
CPU							
128-bits	47,653	30,131	77,785	508	26,490	26,999	12
192-bits	87,656	33,694	121,351	618	26,491	27,110	77
256-bits	131,223	37,257	168,480	728	26,492	27,220	83
Radio							
128-bits	1,924	82,036	83,961	2,993	49,634	52,627	59
192-bits	2,488	139,148	141,637	3,560	49,152	52,713	62
256-bits	305,176	196,261	199,312	4,127	48,670	52,798	73

XI. Conclusion

This paper presented a protocol to establish data provenance in the IoT. The protocol uses PUFs to verify the source of data. Wireless fingerprints derived from the wireless channel between an IoT device and wireless gateway are used to verify the location of data. In particular, LQI values are used to generate the wireless fingerprints. The proposed protocol uses light weight symmetric cryptography for security and also provides privacy preservation. Experiments conducted on MICA-Z motes in an indoor environment showed that the proposed technique for wireless fingerprints can detect attacks with high accuracy. Moreover, the energy requirement for the proposed protocol is significantly lower than existing techniques.

The future work for this paper may include a multiple-hop provenance protocol for IoT swarms, dynamically registering IoT devices without the need of an operator, and applying further optimizations to improve the accuracy of detecting attacks while using even shorter fingerprints.

References

[1] M. Naveed Aman, S. Taneja, B. Sikdar, K. C. Chua and M. Alioto, "Token-Based Security for the Internet of Things With Dynamic Energy-Quality Tradeoff," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2843-2859, April 2019.

[2] M. N. Aman, K. Javed, B. Sikdar and K. C. Chua, "Detecting data tampering attacks in synchrophasor networks using time hopping," in *Proc. 2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Ljubljana, 2016, pp. 1-6.

[3] M. N. Aman, M. H. Basheer and B. Sikdar, "Data Provenance for IoT With Light Weight Authentication and Privacy Preservation," in IEEE Internet of Things Journal, vol. 6, no. 6, pp. 10441-10457, Dec. 2019.

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

[5] M. N. Aman, B. Sikdar, K. C. Chua and A. Ali, "Low Power Data Integrity in IoT Systems," in *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3102-3113, Aug. 2018.

[6] M. S. Siddiqui, A. Rahman, and A. Nadeem, "Secure Data Provenance in IoT Network using Bloom Filters," in *Procedia Computer Science,* vol. 163, pp. 190-197, 2019.

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

[8] 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.

[9] Z. Liu and Y. Wu, "An Index-based Provenance Compression Scheme for Identifying Malicious Nodes in Multi-hop IoT Network," in *IEEE Internet of Things Journal*, 2019.

[10] U. Javaid, M. N. Aman, B. Sikdar, "BlockPro: Blockchain Based Data Provenance and Integrity for Secure IoT Environments," in *Proc. ACM BlockSys*, 2018, pp. 13-18.

[11] N. Baracaldo et al., "Securing Data Provenance in Internet of Things (IoT) Systems," in: *Drira K. et al. (eds) Service-Oriented Computing – ICSOC 2016 Workshops*, Lecture Notes in Computer Science, vol 10380. Springer, Cham.

[12] S. Ali et al., "Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts," in *Proc. 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI),* Guangzhou, 2018, pp. 991-998.

[13] M. Sigwart et al., "Blockchain-based Data Provenance for the Internet of Things," in *Proc. ACM International Conference on the Internet of Things*, New York, 2019, pp. 1-8.

[14] M. Elkhodr, B. Alsinglawi and M. Alshehri, "Data Provenance in the Internet of Things," in *Proc. WAINA,* Krakow, 2018, pp. 727-731.

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

[16] A. Kalamandeen et al., "Ensemble: Cooperative Proximity-based Authentication," in *Proc. ACM MobiSys*, San Francisco, CA, 2010, pp. 331–344.

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

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

[19] J. Tang, P. Fan and X. Tang, "A RSSI-Based Cooperative Anomaly Detection Scheme for Wireless Sensor Networks," in *Proc. WiCom*, Shanghai, 2007, pp. 2783-2786.

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

[21] M. Kamal and S. Tariq, "Light-Weight Security and Data Provenance for Multi-Hop Internet of Things," in *IEEE Access,* vol. 6, pp. 34439-34448, 2018.

[22] M. Kamal and M. Tariq, "Light-Weight Security and Blockchain Based Provenance for Advanced Metering Infrastructure," in IEEE Access, vol. 7, pp. 87345-87356, 2019.

[23] M. N. Aman et al., "HAtt: Hybrid Remote Attestation for the Internet of Things with High Availability," in *IEEE Internet of Things Journal*, early access.

[24] M. N. Aman, K. C. Chua and B. Sikdar, "A Light-Weight Mutual Authentication Protocol for IoT Systems," in *Proc. GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-6.

[25] P. Gope, A. K. Das, N. Kumar and Y. Cheng, "Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks," in *IEEE Trans. Indust. Inform.*, vol. 15, no. 9, pp. 4957-4968, Sept. 2019.

[26] P. Gope, J. Lee and T. Q. S. Quek, "Lightweight and Practical Anonymous Authentication Protocol for RFID Systems Using Physically Unclonable Functions," in *IEEE Trans. Inform. Forensics Sec.*, vol. 13, no. 11, pp. 2831-2843, Nov. 2018.

[27] M. N. Aman, M. H. Basheer and B. Sikdar, "Two-Factor Authentication for IoT With Location Information," in *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3335-3351, April 2019.

[28] 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.

[29] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *Proc. EUROCRYPT*, Amsterdam, The Netherlands, 2002, pp. 337-351.

[31] M. N. Aman, W. K. Chan and B. Sikdar, "Collision detection in IEEE 802.11 networks by error vector magnitude analysis," in *Proc. 2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, 2012, pp. 5218-5223.

[30] A. Bildea, "Link Quality in Wireless Sensor Networks", Ph.D. Thesis, School of Mathematics, Information Science and Technology, Computer Science, Universite de Grenoble, Grenoble, France, 2013.

[32] M. N. Aman and B. Sikdar, "Distinguishing between channel errors and collisions in IEEE 802.11," in *Proc. 2012 46th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, 2012, pp. 1-6,

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

[34] M. N. Aman and B. Sikdar, "ATT-Auth: A Hybrid Protocol for Industrial IoT Attestation With Authentication," in *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5119-5131, Dec. 2018.

[35] B. Blanchet and B. Smyth, *ProVerif: Automatic Cryptographic Protocol Verier, User Manual and Tutorial*, INRIA, Paris, France, 2016.

[36] M. N. Aman. *ProVerif Simulation Scripts for Proposed Protocols [Software]*. Accessed on Aug. 10, 2019. [Online]. Available: https://www.ece.nus.edu.sg/stfpage/bsikdar/scripts/TII.

Muhammad Naveed Aman received the B.Sc. degree in Computer Systems Engineering from KPK UET, Peshawar, Pakistan, M.Sc. degree in Computer Engineering from the Center for Advanced Studies in Engineering, Islamabad, Pakistan, M.Engg. degree in Industrial and Management Engineering and Ph.D. in Electrical Engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA in 2006, 2008, and 2012 respectively.

He is currently working as a Senior Research Fellow with the Department of Computer Science at the National University of Singapore, Singapore. Dr. Aman previously served on the faculty of National University of Computer and Emerging Sciences Pakistan as an Assistant Professor. His research interests include IoT and network security, wireless and mobile networks, and secure embedded systems.

Mohammad Haroon Basheer (S'18) is a Research Assistant with NUS-Singtel Cybersecurity Research & Development Laboratory since Nov 2017.He received his Bachelor of Technology in Electronic Engineering from National University of Singapore, where he is also pursuing his Master of Computing degree in Computer Science.

Biplab Sikdar (S'98-M'02-SM'09) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University,Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He was on the faculty of Rensselaer Polytechnic Institute from 2001 to 2013, first as an Assistant and then as an Associate Professor.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include wireless network, and security for IoT and cyber physical systems. Dr. Sikdar is a member of Eta Kappa Nu and Tau Beta Pi. He served as an Associate Editor for the *IEEE Transactions on Communications* from 2007 to 2012. He currently serves as an Associate Editor for the *IEEE Transactions on Mobile Computing*.