Secure Data Provenance for the Internet of Things

Muhammad N. Aman
National University of Singapore
Singapore 117583
elemna@nus.edu.sg

Kee Chaing Chua
National University of Singapore
Singapore 117583
eleckc@nus.edu.sg

Biplab Sikdar
National University of Singapore
Singapore 117583
bsikdar@nus.edu.sg

ABSTRACT

The vision of smart environments, systems, and services is driven by the development of the Internet of Things (IoT). IoT devices produce large amounts of data and this data is used to make critical decisions in many systems. The data produced by these devices has to satisfy various security related requirements in order to be useful in practical scenarios. One of these requirements is data provenance which allows a user to trust the data regarding its origin and location. The low cost of many IoT devices and the fact that they may be deployed in unprotected spaces requires security protocols to be efficient and secure against physical attacks. This paper proposes a light-weight protocol for data provenance in the IoT. The proposed protocol uses physical unclonable functions (PUFs) to provide physical security and uniquely identify an IoT device. Moreover, wireless channel characteristics are used to uniquely identify a wireless link between an IoT device and a server/user. A brief security and performance analysis are presented to give a preliminary validation of the protocol.

1. INTRODUCTION

The rapid growth of IoT devices has opened the way for new and exciting applications such as smart cities, smart hospital care, and smart vehicles etc. However, the large amounts of data that these devices may produce and the sensitive nature of this data make the IoT a prime target for cyber attacks. The IoT devices are usually low cost and simple in nature with limited processing, memory, and energy resources. The main security challenges faced by IoT systems include authentication, data integrity, data provenance, privacy, and access control. Moreover, many IoT devices are deployed out in the open and cannot be considered physically secure. Thus, any protocol developed for IoT systems needs to by secure against physical attacks. For example, if an IoT device stores a secret key in its memory, an attacker may launch a physical attack (e.g. optical scrutiny) to read the contents of its memory.

IoTPTS'17, April 02 2017, Abu Dhabi, United Arab Emirates

DOI: http://dx.doi.org/10.1145/3055245.3055255

Data provenance establishes trust in the origin and creation process of data. This gives a guarantee that the user can trust the data received from an IoT device i.e., that the data is indeed collected by the specific IoT device at the stated location and time. Self trust or data provenance is critical to the correct operation of the IoT [1]. Recently, several techniques for providing data provenance using hardware security primitives have been proposed [2, 3, 4]. Other techniques using the wireless channel characteristics have also been proposed. The authors of [2] propose the use of sensor PUFs to establish data provenance in the IoT. However, if an adversary moves a sensor PUF from its original location the scheme breaks down i.e., the receiver of the data will continue accepting invalid sensor readings without knowing that the location of the data's origin has changed. Similarly, in [3] the authors propose a scheme for data provenance using wireless link fingerprints. They use the received signal strength indicator (RSSI) values to generate unique fingerprints. However, their scheme requires the sensor devices to store a secret key locally. This requirement exposes their protocol to physical attacks. Moreover, these techniques use public key cryptography and have high energy and processing requirements.

In this paper we present preliminary work on the development of a data provenance protocol for IoT systems. The proposed protocol uses PUFs and wireless link fingerprints to ensure self trust and data provenance in IoT systems. The use of a PUF ensures that the user can trust that the data is coming from the stated IoT device. Similarly, the use of wireless link fingerprints ensures that user can trust that the data has been collected from the stated location. Moreover, the proposed protocol does not require IoT devices to store any secret keys which makes it secure against physical attacks. The proposed protocol uses PUFs and symmetric key cryptography, making it a light weight and efficient solution for IoT systems.

This paper is organized as follows. Section 2 gives a brief introduction to PUFs and wireless link fingerprints. We present the details of the proposed data provenance protocol in Section 3. Section 3.3 and 3.4 present a security and performance analysis of the proposed protocol. We conclude the paper in Section 4.

2. BACKGROUND MATERIAL

Before presenting the proposed protocol, we first give a brief introduction to PUFs and wireless link fingerprints in this section.

A PUF can be described as [5] "an expression of an in-

herent and unclonable instance-specific feature of a physical object". Thus, A PUF is a physically disordered system that maps a set of challenges to a set of responses based on the underlying physical micro structure of the device. It can be shown that PUFs are extremely difficult or even impossible to clone [6]. A PUF can be considered a one way function which takes a challenge as the input and produces a corresponding response. A challenge C and its response R for a given PUF is called its challenge response pair (CRP). We can represent a PUF as follows:

$$R = P(C). \quad (1)$$

If a challenge is input to a PUF multiple times, the PUF always produces the same response with high probability. However, a different PUF produces a response far apart with high probability. Some of the desirable properties of PUFs for use with IoT devices include [7]: physical security, high throughput with low energy and silicon area footprints, low cost and simple, and unclonable.

The proposed protocol also uses wireless link fingerprints. Jakes uniform scattering model [8] states that signals are highly de-correlated when the transmitter or receiver moves a distance of over half a wavelength. Also, the channel characteristics are symmetric for the two parties. Moreover, the model implies that the wireless channels separated by a distance of one wavelength or more can be considered independent. Thus, any of the wireless channel parameters such as the received signal strength indicator (RSSI) may uniquely identify a wireless link between two parties and may be used as a wireless link fingerprint.

It has been shown that radio signals can be used to identify a transmitting party [9]. However, the strict assumptions regarding stationary deployment of a transmitter and low probability of success undermine these techniques. The authors of [3] show that the received signal strength indicator (RSSI) values of a wireless channel can be used as the fingerprint of a wireless link. The RSSI values are quantized using typical quantization mechanisms such as level crossing or ranking techniques. The current RSSI value of the wireless link between two parties is then used as the wireless fingerprint for a given session. However, their technique requires the device to store a secret key and depends on the public key infrastructure. These two requirements not only make the IoT devices vulnerable to physical attacks but the use of public keys makes the protocol less efficient. To solve these issues the proposed protocol uses PUFs to make it secure against physical attacks. Moreover, the use of secret key cryptography makes the proposed protocol more efficient and realizable for IoT systems.

3. A SECURE DATA PROVENANCE PROTOCOL

This section presents the proposed protocol for data provenance in IoT systems using PUFs and wireless link fingerprints.

3.1 Network Model, Assumptions, and Notations

The network model consists of IoT devices connected to a server/gateway through a wireless network as shown in Figure 1. We make the following assumptions for our network model and proposed protocol:

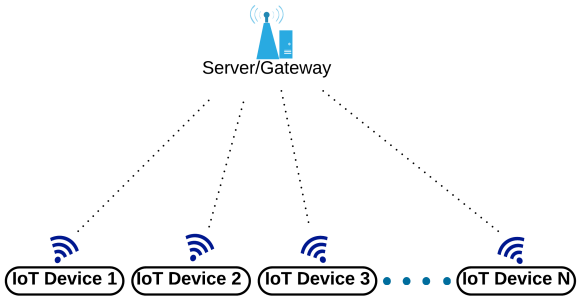

Figure 1: Network Model

a. A device's micro-controller and the PUF are assumed to be a system on chip. Therefore, any attempt to remove or tamper with the PUF makes it inoperable.

b. IoT devices have limited resources such as memory, processing power, and energy. However, the server has no such limitation.

We now present the threat model for the proposed protocol as follows:

1. IoT devices are assumed to be physically unprotected. An adversary may gain access to an IoT device and launch a physical attack. The objective of the adversary is to gain access to the device's memory and eventually steal secrets.

2. An adversary may try to imitate an IoT device. The objective of the adversary in this type of attack is to send invalid data to the server, in turn forcing the server to make wrong decisions.

3. An attacker can eavesdrop, modify, replay, and inject messages.

4. The attacker is located at least two wavelengths away from the legitimate parties.

5. The objective of the attacker is to tamper with the data sent from an IoT device to the server and invalidate its provenance.

3.2 Proposed Protocol

In this section we present a detailed description of the proposed protocol for secure data provenance in the IoT. The proposed protocol can be divided into two phases: the setup phase and the data transfer phase.

3.2.1 Setup Phase

The setup phase of the proposed protocol is shown in Figure 2. We assume that the initial CRP is obtained by the server when an IoT device is deployed in field for the first time. The initial CRP is sent to the server using a one-time-password authentication mechanism, whereby the operator inputs a one-time-password into the device. Once the initial CRP is exchanged between the server and the IoT device, the device can operate independently without the need of any human intervention.

The setup phase consists of the following steps.

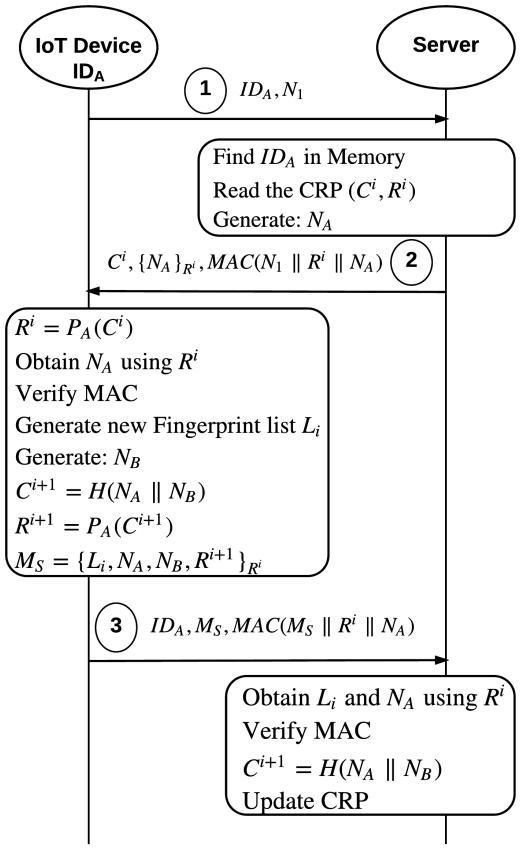

Figure 2: Setup Phase

1. The ID ID_A and a random nonce is sent to the server by the IoT device (message 1).

2. The server tries to locate the corresponding CRP (C^i, R^i) for this IoT device in its memory. If the CRP for ID_A is not found in the server's memory, the setup request is rejected. Otherwise, the server sends a random nonce N_A (encrypted using R^i i.e $\{N_A\}_{R^i}$) and the challenge C^i to the IoT device in message 2.

3. The IoT device obtains the corresponding response R^i for the challenge C^i using its PUF. The IoT device then carries out the following tasks:

   (a) Using R^i as the secret key, obtain N_A.

   (b) Verify the MAC using parameters in its memory.

   (c) Generate a new list of link fingerprints L_i as follows:

   $$F_i = P_A(H(W_i \parallel N_A)) \quad (2)$$

   where F_i represents the fingerprint of the wireless link between the IoT device and the server when the RSSI value is W_i. Note that we quantize the RSSI values, so that the practical range of the RSSI values can be used to determine the number of bits required to represent a wireless link fingerprint. Thus, we can consider only a finite set of RSSI values $W_1, W_2, ..., W_n$. The IoT device generates a fingerprint for each possible RSSI value and sends this list to the server in message 3.

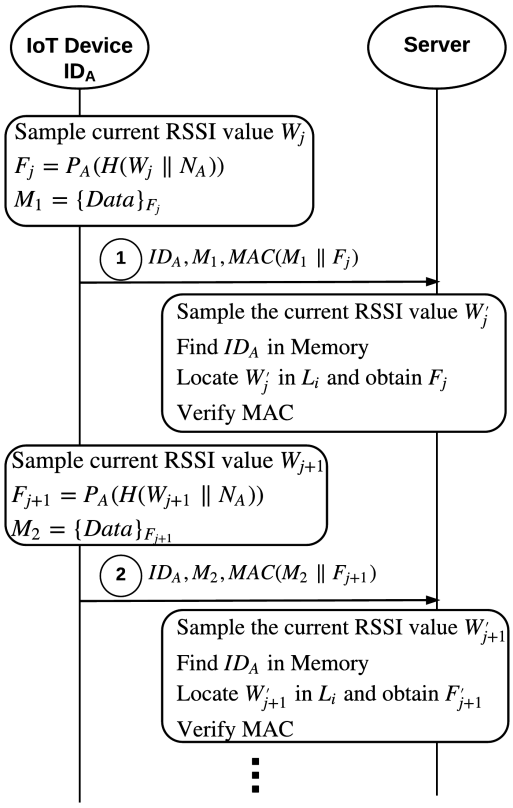

Figure 3: Data Phase

   (d) Generate a new random nonce N_B.

   (e) Generate a new CRP using N_A and N_B and send it to the server in message 3. This CRP will be used by the server for any future sessions with the current IoT device.

4. The server obtains the list of fingerprints L_i and N_A using the secret key R^i. The server then verifies the MAC, and if the verification fails the setup request is denied. Otherwise, the setup phase is considered complete and the server saves the list of fingerprints and the updated CRP for the corresponding IoT device in its memory. L_i is used by the server to check data provenance during the data transfer phase.

3.2.2 *Data Transfer Phase*

The data transfer phase of the proposed protocol is shown in Figure 3. The following steps are repeated for each packet during the data transfer phase:

1. The IoT device carries out the following steps for each data packet sent:

   (a) Samples the current RSSI value, W_j, for the wireless link between itself and the server.

   (b) Generates the current link fingerprint using the current RSSI value W_j and the server's secret nonce N_A using Equation (2).

   (c) Encrypt the data (packet) using the current fingerprint and send message 1 to the server. Message 1 also contains a MAC to ensure data integrity.

2. The server carries out the following steps for each data packet received:

   (a) Samples wireless link for the current RSSI value W_j' between itself and the IoT device.

   (b) Locate the ID of the corresponding IoT device in its memory, in this case ID_A.

   (c) Retrieve the fingerprint list L_i for the IoT device and find the fingerprint F_j' corresponding to the current RSSI value W_j'.

   (d) Decrypt the received data packet using F_j' and verify the MAC. If verification fails then reject the packet, otherwise, accept the packet.

3.3 Security Analysis

The objective of the proposed protocol is to establish the provenance of data received by a server. An attacker may try to impersonate an IoT device and try to give inaccurate data to the server with a malicious intent. However, we show that the proposed protocol is safe against these types of attacks as follows:

1. The attacker cannot obtain the current fingerprint from the messages in the data phase. Moreover, each IoT device has its own PUF. Therefore, even if an attacker manages to get the current RSSI value he cannot use it to produce the correct fingerprint, unless it can clone the IoT device's PUF. Note that the IoT devices do not store any secrets in their memory which makes them less vulnerable to physical attacks.

2. Even if an attacker somehow manages to authenticate its PUF with the server, its wireless link fingerprint will be different. For example, if the wireless network is operating in the 2.4 GHz band, the attacker needs to be closer than 13 cm to the IoT device to get the same channel characteristics and thus fingerprints. However, we assume that the attacker is at least two wavelengths away from an IoT device. Given such a small distance of 13 cm or even smaller for millimeter wave communications, this assumption can be considered realistic and maintains the generality of the proposed protocol.

3. An attacker may try to replay packets of an older session. However, due to the use of a new nonce N_A for every session, the protocol is safe against these type of attacks.

4. As a PUF produces the same output for the same challenge, if an attacker manages to sample the RSSI value at the IoT device, she may try to exploit this by replaying an older value from a previous session when the RSSI value at the IoT device was the same. However, this attack is thwarted by the use of a new random nonce in Equation (2) in each session which produces a different link fingerprint even if the RSSI value is the same.

5. The use of message authentication codes ensures the data integrity of the messages.

3.4 Performance Analysis

In this section we discuss the performance of the proposed protocol. The setup phase of the protocol needs to be invoked only once for each session. We can see from Figure 2 that the setup phase consists of an exchange of only three messages. The IoT device needs to perform only one encryption and one MAC operation in this phase. Similarly, in the data transfer phase an IoT device requires to perform one symmetric key encryption and one MAC operation for each packet it sends. As the encryption used is secret/symmetric key encryption, therefore, we can conclude that the proposed protocol introduces a low overhead for an IoT device. On the other hand similar protocols such as [3] use public key encryption, which increases the overhead for the IoT devices. Similarly, the use of sensor PUFs for data provenance in [2] requires a continuous stream of different challenge bits, which increases the complexity and reduces the effectiveness of this technique.

4. CONCLUSIONS

This paper presents preliminary work for a data provenance technique for the IoT using PUFs and wireless link fingerprints. The spatio-temporal characteristics of a wireless channel are exploited to generate the wireless link fingerprints. By combining wireless link fingerprints with PUFs, the proposed protocol achieves security against physical attacks. Moreover, the protocol achieves its desired security goals efficiently using symmetric key encryption. The proposed protocol uses RSSI values to generate the wireless link fingerprints. However, it may be interesting to also examine other channel parameters for generation of wireless link fingerprints. A preliminary security and performance analysis show that the proposed protocol can achieve the desired security goals efficiently. However, a more detailed and experimental evaluation of the proposed protocol is required to evaluate the performance and security of the proposed protocol.

5. REFERENCES

[1] E. Bertino, "Data Security and Privacy in the IoT," *Proc. EDBT,* March 2016.

[2] A. Kanuparthi et. al., "Hardware and Embedded Security in the Context of Internet of Things," *Proc. ACM CyCAR,* November 2013.

[3] S. T. Ali et. al., "Securing Data Provenance in Body Area Networks using Lightweight Wireless Link Fingerprints," *Proc. TrustED,* November 2013.

[4] D. Cheng et. al., "Wireless Device Authentication Using Acoustic Hardware Fingerprints," *Proc. Int. Conf. on Big Data Computing and Communications,* 2015.

[5] R. Maes, "Physically Unclonable Functions: Constructions, Properties and Applications," Katholieke Universiteit Leuven Belgium DEngg Thesis, 2013.

[6] C. Bohm, and M. Hofer, "Physical Unclonable Functions in Theory and Practice," Springer, 2012.

[7] M. N. Aman et. al., "Physical Unclonable Functions for IoT Security," *Proceedings of ACM AsiaCCS IoTPTS,* June 2016.

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

[9] K. B. Rasmussen and S. Capkun. "Implications of Radio Fingerprinting on the Security of Sensor Networks," *Proceedings of SecureComm,* September 2012.