Secure Data Provenance in Home Energy Monitoring Networks

Ming Hong Chia
University of Glasgow
Lilybank Gardens
Glasgow G12 8RZ, UK
ChiaMingHong@gmail.com

Sye Loong Keoh
University of Glasgow
Lilybank Gardens
Glasgow G12 8RZ, UK
SyeLoong.Keoh@glasgow.ac.uk

Zhaohui Tang
Singapore Institute of Technology
10, Dover Drive
Singapore 138682
Zhaohui.Tang@singaporetech.edu.sg

ABSTRACT

Smart grid empowers home owners to efficiently manage their smart home appliances within a Home Area Network (HAN), by real time monitoring and fine-grained control. However, it offers the possibility for a malicious user to intrude into the HAN and deceive the smart metering system with fraudulent energy usage report. While most of the existing works have focused on how to prevent data tampering in HAN's communication channel, this paper looks into a relatively less studied security aspect namely *data provenance*. We propose a novel solution based on Shamir's secret sharing and threshold cryptography to guarantee that the reported energy usage is *collected from the specific appliance as claimed at a particular location*, and that it *reflects the real consumption* of the energy. A byproduct of the proposed security solution is a guarantee of data integrity. A prototype implementation is presented to demonstrate the feasibility and practicality of the proposed solution.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; **Security protocols**;

KEYWORDS

Data Provenance, Home Area Networks, Location Authenticity, IoT, Smart Grid

ACM Reference format:
Ming Hong Chia, Sye Loong Keoh, and Zhaohui Tang. 2017. Secure Data Provenance in Home Energy Monitoring Networks. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 8 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The growing demand of electricity power and the drive towards a low-carbon economy have urged the electricity services industry to develop an innovative energy management solution which can meet the user's demand for power as well as to increase the efficiency of energy distribution and usage. Smart grid therefore emerged to replace the traditional grid system to report real-time energy consumption and support fine-grained customer monitoring and control, hence providing the users with better control and more efficient management of their daily consumption.

Smart grid system relies heavily on information and communications technologies (ICT) for establishing communications among its components including smart home appliances, smart meters and utility providers. The security of smart grid [1, 13] has drawn a big attention to researchers from both academia and industry due to massive challenges posed by cyber-security threats [3, 12].

In October 2014, smart meters in Spain were hacked to cut power bills [20], where the attackers made use of leaked encryption keys and unique identifier associated with the smart meter to spoof messages sent from home appliances to utility to under-report energy usage. Additionally, the unique identifier can be spoofed by using another household's smart meter identity which resulted in fraudulent energy usage to be reported for financial gains. With such an inadequate protection against tampering and weak security, adversaries can easily make use of these exploits to attack the smart grid infrastructure.

This paper aims to investigate all aspects of data provenance and thus can be considered as the first paper to comprehensively study data provenance related attacks in a home energy monitoring networks. There are many attacks that can be launched to commit energy fraud: Firstly, the smart meter can be compromised to under-report the energy used in a household. Though the hardware of the smart meter can be hardened, it would be more effective to reliably monitor and measure the *real power consumption* at every power outlet in the house using smart plugs. This approach aims to reflect the true energy use which can then be cross-checked with the energy consumption recorded by the smart meter. With this, any discrepancy between the two readings can be detected and energy fraud is suspected.

However, it is also important to ensure the security of the smart plugs, Firstly (1) a compromised smart plug may tamper with the energy reading before sending it to the smart meter. This breaches the *data source authenticity* as the measurement itself has already been tampered with at the smart plug prior to data transmission. (2) Secondly, a smart plug's device identity may also be spoofed, by masquerading as other smart plugs to send out fraudulent energy reading. (3) Replay attacks can be potentially exploited by an impersonated smart plug to send a previously reported energy data.

We consider all the above challenges together as *data provenance* issues in a home energy monitoring network. This paper proposes a novel solution based on Shamir's secret sharing and threshold cryptography to guarantee that the reported energy usage is *collected from the specific appliance as claimed*, and that it *reflects the real consumption* of the energy. Specifically, the proposed solution contributes to achieving the following security goals:

(1) **Source Identity Authenticity** – ensures that the energy data is originated from the smart plug as claimed.
(2) **Source Data Authenticity** – means that the energy consumption measured by the smart plug on each power outlet reflects the real consumption.
(3) **Data Integrity** – guarantees that the data transmitted from the smart plug to the smart meter is not tampered with.

Conference'17, July 2017, Washington, DC, USA
2017. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(4) **Data Consistency** – ensures that there is no discrepancy between the energy consumption recorded by the smart meter and the aggregated data from the smart plugs.
(5) **Location Authenticity** – ensures that the energy data is collected from the location of the power outlet as stated.

This paper is organized as follows. In Section 2, we provide details on the background, followed by related work and some preliminary knowledge. Our detailed solution is presented in Section 3 and the implementation environment is presented in Section 4. We present the initial performance analysis in Section 5. We conclude the paper with future work and discussion in Section 6.

2 BACKGROUND AND RELATED WORK

2.1 Smart Grid and HAN

2.1.1 Smart Grid. A smart grid is an electric grid which integrates power generation, transmission, distribution and consumption using information communication network technologies [10]. One of the key characteristics of a smart grid is the support of bi-directional information flow between the consumer and utility provider to create an automated and distributed advanced energy delivery network [6]. This also allows electricity to be generated in real-time based on consumers' demands and power requests. The addition of innovative sensing and control systems enables smart grid to be capable of performing real time monitoring of power flows thus in turn providing better information about optimization and forecasting of power usage [16].

To realize this, an Advanced Metering Infrastructure (AMI) based on smart meters is the most critical technology required to allow transmission of information in the communication network. A smart meter is defined as an advanced meter that performs measurement of the energy consumption in deeper details compared to a conventional meter and it periodically sends the collected information back to the utility company for load monitoring and billing purposes. Prior to that, the data (energy usage) collected from the smart meter is also important for control centre (utility) to implement a demand and response mechanism.

With these powerful features available, it also brings numerous security concerns, e.g., this enables new ways to commit energy fraud, perform large scale attacks to cripple power supply to consumers which could be difficult to address in this emerging system.

2.1.2 Home Area Network (HAN). A home area network (HAN) is a subsystem within the smart grid which extends smart grid capabilities into the home using different networking protocols. HAN is typically found in consumer premises and it connects home (smart) appliances such as thermostats, refrigerators and other electrical devices to a smart meter. A HAN typically relies on a wireless link such as WiFi, ZigBee to communicate with the smart meter via a gateway. The gateway also acts as a bridge between the smart meter and the smart appliances in consumer's home. The smart meter collects energy usage data, network status from the utility for display to the consumer. Moreover, the smart meter also forwards demand/response and energy-pricing signals to the smart appliances for their information. This forms a home energy monitoring network at home.

However, this also makes HAN one of the most vulnerable systems in the smart grid because the wireless communication is being located in a physically insecure environment. It is possible for an attacker to intercept and monitor network traffic to gain sensitive information across the wireless communication.

In addition to HAN, Neighbourhood Area Network (NAN) is formed through meshing of smart meters and data concentrators in an area in order to conserve bandwidth. NAN ensures smooth communication links between a number of individual smart meters and a data concentrator using WiMAX, Wi-Fi or cellular technologies and Wi-SUN. A number of data concentrators are connected to a central system in the utility side through Wide Area Network.

2.2 Related Work

Aman *et al.* proposed to use Physical Unclonable Function (PUF) to secure data provenance in IoT Systems [2]. PUFs are used to protect the identity of an IoT device, thus providing physical security to enable an IoT device to be identified uniquely in the network. Device identity is thus guaranteed using PUF, additionally the location authenticity of the IoT device is preserved by using the wireless link's Received Signal Strength Indicator (RSSI) values as unique fingerprints. However, this research did not have any implementation and analysis to demonstrate the practicality of the approach, and it does not guarantee the source data authenticity.

Jiang *et al.* introduced an energy-theft detection scheme [11] for AMI using a machine learning approach. Classification-based technique is used by incorporating statistical learning theory to classify the load profiles of customer's abnormal behaviour in order to detect energy-theft suspects. This is complemented by a state-based detection method that monitors events derived from the network and AMI Intrusion Detection System (IDS) in order to perform mutual inspection, and event correlation, thus vastly improving the detection rate. Yet, all these detection schemes still do not address fundamental issue of hardware tampering where the energy consumption data is tampered with at its source.

In [19], data provenance research addresses the issue of tracking the information for data accountability. The existing literature relates data provenance to integrity or tamper-evident, authenticity and reliable data collection. Kairos [7] explored the authenticity of provenance records in a grid computing environment with a time-stamp authority to generate a tamper-evident proof through the use of a user digital signature. Bonsai [9] proposed an addition of digital signatures of users or operators to the provenance records whereby verification is executed only when requested by users for authenticity of provenance records. Boyen *et al.* [5] introduced a bi-linear pairing technique to uphold the properties of tamper-evident and confidentiality in provenance records so that provenance information can then be trusted. Lyle *et al.* suggested the use of Trusted Platform Module (TPM) to tackle the issue of reliability [14], whereby the hardware is made tamper-resistant.

2.3 Preliminary

This section presents some fundamental knowledge which will be used in future sections of this paper.

2.3.1 RSA Cryptosystem. RSA Cryptosystem, one of the most utilized public-key cryptosystems, can be used to ensure that the

message was indeed from the sender and not tampered with during the transmission. An RSA cryptosystem consists of three steps as below:

(1) **Key Generation**: An RSA key pair (PK, SK) is generated: $PK = (n, e)$ is the public key and $SK = d$ is the private key. For security consideration, (PK, SK) must satisfy the following requirements ($\phi = (p-1)*(q-1)$): $n = p*q$ with p, q randomly chosen big primes; $1 \leq e \leq \phi$, gcd $(e, \phi) = 1$; $d * e \equiv 1 \bmod \phi$.
(2) **Signature Generation**: For a message m, the sender generates the signature c as $c \equiv m^e \bmod n$.
(3) **Signature Verification**: Upon receiving a message m and its signature c, the receiver verifies that: $c^d \equiv m \bmod n$.

Only the sender who possesses the private key $SK = d$ is able to generate the valid signature which in turn proves that the message m was indeed sent by the sender as claimed.

2.3.2 (k,N) Threshold Secret Sharing and (k,N,SK) RSA Private Key Share Generation. Secret-sharing schemes (SSS) were first introduced in 1979 by Blakley [4] and Shamir [17] and used in many cryptographic protocols. The main idea of classical secret sharing is dividing a secret amongst a group of participants, where each is allocated a share of the secret. The secret can only be reconstructed when the shares are combined together from certain authorized groups of shares. Typically, in a (k, N) threshold secret sharing scheme, a minimum of k out of the total N shares are required to reconstruct the secret. An example of (k, N) threshold secret sharing is polynomial-based Shamir Secret Sharing.

In this work, we take the RSA private key SK as the secret and exploit Shamir Secret Sharing scheme to split the secret into shares among the participants. Details of generating the secret shares for RSA private key is presented in Figure 1.

- **Input:** SK, k, N where SK is the RSA private key, N is the total number of participants and k is the minimum number of participants who are allowed to recover SK.
- **Output:** $s = s_1, \cdots, s_N$ where s_i is computed by $s_i = f(i)$ with

$$f(i) = SK + a_1 * i + \ldots + a_{k-1} * (i)^{k-1}$$

Here, $a_i (1 \leq i \leq k-1)$ are randomly chosen unknown elements of Z_p.

Figure 1: (k, N, SK) RSA Private Key Share Generation.

2.3.3 (k,N,m,s) RSA Threshold Signature Scheme. This scheme is built upon Shoup's practical threshold scheme [18]. It allows a total of N participants to generate a partial RSA signatures on message m individually using their respective RSA private key share computed as described in Figure 1, where a minimum of k partial signatures are required to be combined to generate the full RSA signature on m. The verification of the full signature in this scheme is the same as the verification of standard RSA signature as shown in Section 2.3.1. Detail of this scheme is shown in Figure 2. Note that input s is the collection of RSA private key shares resulted from the *(k, N, SK) RSA Private Key Share Generation*.

- **Partial-Signature-Generation** (m, i, s)**:** Given x = Hash(m) from message m, the partial signature for signer (entity) i is generated as follows:

$$x_i = x^{2\Delta s_i} \in Q_n, \quad \Delta = N!$$

where Q_n is the subgroup of squares in Z_n^*

- **Full-Signature-Generation** (τ_x)**:** For a set τ_x of entities, where $\tau_x = \{x_{i_1}, \ldots, x_{i_k}\} \subset \{x_{i_1}, \ldots, x_{i_N}\}$. Let x = Hash(m) $\in Z_n^*$ and assume that $x_{ij}^2 = x^{4\Delta s_{ij}}$. The Full RSA Signature FS for message m, can be generated as:

$$FS = w^a x^b.$$

Here a and b are integers such that $e'a + eb = 1$ with $e' = 4\Delta^2$ and gcd(e', e) = 1; w is computed as:

$$w = x_{i_1}^{2\lambda_{0,i_1}^{\tau_x}} \quad \ldots \quad x_{i_k}^{2\lambda_{0,i_k}^{\tau_x}},$$

where the λ's are the integers defined as:

$$\lambda_{i,j}^{\tau_x} = \Delta \frac{\Pi_{j' \in \tau_x \setminus \{j\}} (i - j')}{\Pi_{j' \in \tau_x \setminus \{j\}} (j - j')} \in Z$$

- **Full-Signature-Verification** (m, FS, PK)**:** A verifier checks whether the following equation holds: $FS^e \equiv m \bmod n$. If it holds then verification passes, otherwise the verification fails.

Figure 2: (k,N,m,s) RSA Threshold Signature Scheme.

3 PROPOSED SOLUTION

This paper proposes to design an integrated device to be deployed in HAN to securely and accurately measure the energy consumption data, and then implement a two-phase communication protocol in order to fulfill all the security goals as defined in Section 1.

The most challenging goal to achieve is *source data authenticity* in a home energy monitoring network. We introduce cross-check validation in our scheme, ensuring that two or more independent and co-located energy monitoring devices measure the energy consumption at a power outlet simultaneously and subsequently send the energy data to the designated smart meter periodically. In this way, any sign of discrepancy between the measuring devices implies that there is a potential tampering with the energy data. As these devices are different in nature, tampering with one device does not imply that the other device can be tampered with in the same manner, thus providing reliability and robustness in our system.

This work exploits a novel magnetic sensor [8] that is co-located with the smart plug as a redundant measuring device to collect a home appliance's energy usage simultaneously but independently from the smart plug. The energy value drawn from the magnetic sensor (i.e., v_1), is used to cross-check with the value reported by the smart plug (i.e., v_2). The *source data authenticity* at a power outlet is thus achieved if $v_1 \approx v_2$; the source data authenticity is breached if $v_1 \not\approx v_2$ (which means either the smart plug or the magnetic sensor is sending fraudulent energy reading).

Since our scheme makes use of the magnetic sensor's inputs for cross-checking, it is imperative to ensure the integrity of the data sent from the magnetic sensor to the smart meter, and thus, a tag is necessarily generated by the magnetic sensor using its own private credential (this is similar to the case of smart plug). Therefore, a private credential is required for both the smart plug and the magnetic sensor. In case that tampering is detected, it is unclear which device is the dishonest one if a symmetric-key scheme is used. Conversely, with a public-key scheme, it is risky to let the smart plug and magnetic sensor to generate their own private keys, because any compromised device can create fake data with a valid tag if the full private-key is known.

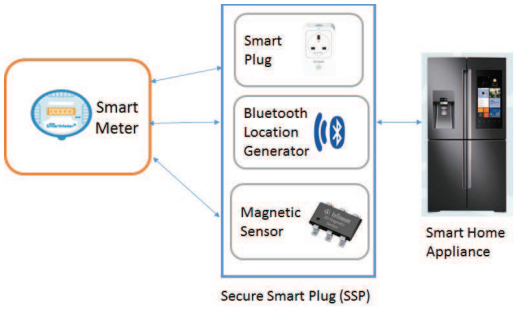

Figure 3: The Secure Smart Plug (SSP) periodically sends protected energy consumption data to the Smart Meter

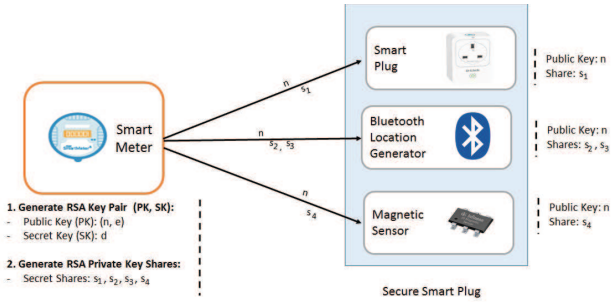

Figure 4: Commissioning of *SSP* to facilitate RSA Private Key Share Distribution

In this paper, we introduce a novel use of secret sharing scheme in home energy monitoring networks, where a secret, i.e., a single private key, is split among all the participants providing measurement of the energy consumption. In addition to the smart plug and the magnetic sensor, our scheme is concerned with the location authenticity, thus a *location generator* is included as an additional participant. The *location generator* uses the trilateration technique built on the Received Signal Strength Indicator (RSSI) values and RSSI filtering methodology to verify the location of the device, ensuring that the correct power source is being monitored and any relocation of the device can be detected. In our current implementation, we further assume that the *location generator* is fully trusted to report the location of the device. In total, there are three participants sharing a single private key for each energy measurement point: *smart plug*, *magnetic sensor*, and *location generator*.

The private key is split into multiple shares between the three entities such that it allows the combination of *location generator* and *smart plug* to recover the private key, and similarly permits the combination of *location generator* and *magnetic sensor* to reconstruct the key; however, a collusion between the *smart plug* and the *magnetic sensor* is prohibited from recovering the private key. Based on this requirement, our scheme adopts threshold secret sharing which means a minimum number (i.e., threshold, k) of shares are required to reconstruct the secret (the single key) successfully. In essence, our scheme ensures that the energy data received from the *smart plug* and the *magnetic sensor* are considered authentic if they can each produce k partial signatures from itself and the *location generator*, which can then be combined into a full valid signature. The *location generator* in this scheme will only produce a partial signature if and only if the location of the device is authentic, i.e., not relocated. As a result, if the full signature can be combined and then verified, it means that the energy consumption data is produced by an authentic device, measured at the designated location without any sign of tampering. Therefore, fulfilling the *source identity authenticity* and *location authenticity* requirements.

3.1 Secure Smart Plug (SSP)

We propose a device called *Secure Smart Plug (SSP)* which is essentially an integration of three individual components: Smart Plug, Bluetooth Location Generator, Magnetic Sensor to accurately measure the energy usage and to ensure location authenticity. The SSP is typically located inside indoor environment to connect home appliances and reports the energy consumption to smart meter, *SM*.

Figure 3 illustrates the various entities in the home energy monitoring network. The SSP is deployed in HAN environment to gather and report the electricity consumption of a smart home appliance e.g., a smart refrigerator, to the smart meter. The following describes the entities in the proposed system:

(1) *SM* – denotes the Smart Meter in a HAN.
(2) *SSP* – denotes the Secure Smart Plug, which consists of three components, namely (1) *Smart Plug* (SP) that is used to measure the energy consumption of a smart home appliance. (2) *Bluetooth Location Generator* (BT) to identify and determine the location information of the SSP. (3) *Magnetic Sensor* (MS), as an alternative medium to measure the energy usage of the smart home appliance.
(3) Smart Home Appliance – any specialized device that consumes electricity, and that its energy consumption can be measured by the SSP.

The proposed scheme operates in two-phases. First, in the *commissioning phase*, the *SSP*s are installed in the HAN, and each of them is commissioned so that credentials can be pre-loaded or configured. Second, the *operational phase* will allow the *SSP* to securely measure the energy consumption of the smart appliance and provide location authenticity guarantee to the smart meter through the use of RSA Threshold Signature Scheme (Section 2.3.3).

3.2 Commissioning of SSP and Smart Meter

In this phase, the *SSP* is commissioned prior to the operation. There are two part of commissioning: *RSA Private Key Share Distribution* , and *Location Registration* for *SSP*.

3.2.1 RSA Private Key Share Distribution. Each *SSP* is provisioned with a private key, as requiring each individual component in the *SSP* to have a private-key each is too heavyweight. Additionally, by splitting the private key into multiple shares and then distributing them to the three components of *SSP* can prevent the compromise of the private-key.

Figure 4 illustrates the key deployment and RSA private key share distribution. In this phase, the *SM* is responsible to firstly generate an RSA Key Pair (PK and SK), and thereafter, to execute $(3, 4, SK)$ *RSA Private Key Generation* (as described in Figure 1). We

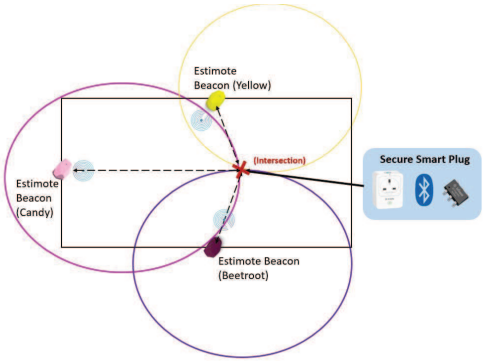

Figure 5: Commissioning of *SSP* for location registration

advocate that the SK is split into four shares for each *SSP* such that $s = \{s_1, s_2, s_3, s_4\}$ and then distribute the shares to the three entities in *SSP*: s_1 for SP, s_2 and s_3 for BT, and s_4 for MS. The allocated share(s) will be used in the *Operational Phase* by the three entities to generate their partial RSA signatures respectively.

3.2.2 Location Registration. Our scheme relies on Bluetooth Low Energy (BLE) beacons to be deployed in order to determine the location of the *SSP*. The *SSP* indoor location can be accurately determined on the basis of three known reference points from the BLE beacons using Trilateration Scheme as shown in Figure 5.

The Trilateration technique illustrated in Figure 5 is based on the RSSI measurement of the three deployed Bluetooth beacons, to calculate the estimated distance of the *SSP* from each beacon. The location registration is built upon the RSSI filtering methodology [15]. This is constructed on the basis of three reference nodes with the known position coordinates of the beacons identifier characteristics (MAJOR and RSSI) and the known dimension of length x and width y in an indoor environment. Once the deployed location of the *SSP* is registered, the *BT* component of the *SSP* is responsible to ensure that whenever an energy consumption is measured by the *SP* or *MS*, it must ensure that the location of the *SSP* is verified using the same trilateration technique.

3.3 Operational Phase

During the operational phase, energy reading is reported by each *SSP* to the *SM* periodically. Each *SP* and *MS* in the *SSP* is responsible for measuring the energy consumption to be reported independently, while the *BT* is responsible for verifying that the *SSP*'s location is authentic and has not been re-located. The energy consumption data is partially signed using the RSA Threshold Signature Scheme by the *SSP*, thus ensuring data integrity and authenticity, as well as location authenticity.

The following shows the steps during the Operational Phase to facilitate the reporting of energy consumption to the *SM*, and we assume that the communication between the three components in *SSP* are encrypted.

Step 1: Measure Energy Reading and Verify *SSP*'s Location. The three entities of *SP*, *MS* and *BT* execute Step 1 concurrently with specific details given below.

- **Step 1-SP:** *SP* measures the home appliance's energy consumption r_{sp} and sends a concatenated message
$$m_{sp} = (SP, r_{sp}, t_{sp})$$
to *SM* and *BT* where *SP* is the identity, and t_{sp} is a timestamp at which the energy reading r_{sp} is recorded.
- **Step 1-MS:** Similar to *SP*, *MS* in Step 1 extracts the home appliance's energy reading r_{ms} and sends
$$m_{ms} = (ms, r_{ms}, t_{ms})$$
to *SM* and *BT*. The message m_{ms} contains the electricity consumption measured and sent to *SM* at the time of t_{ms} and its value is r_{ms}.
- **Step 1-BT:** When SP reports the energy reading, *BT* is responsible to verify whether *SP*'s location remains the same as registered in the Commissioning Phase. If the verification shows that the location of *SP* remains, *BT* will continue with *Step 2-BT*; otherwise it will terminate the processing.

Step 2: Generate Partial RSA Signatures. In order to prevent the message containing the energy consumption from being tampered with during transmission, all three entities *SP*, *MS*, and *BT* are each required to generate and send *SM* their respective partial signature(s) based on their individual RSA private key share(s). The message containing the reported energy consumption is signed using the algorithm *Partial-Signature-Generation* as shown previously in Figure 2. All these partial RSA signatures will then be verified by the *SM* in Step 3.

- **Step 2-SP:** To prevent m_{sp} (refer to *Step 1-SP*) from being tampered with, *SP* in Step 2 generates a partial RSA signature, denoted as x_1, based on m_{sp} and its RSA private key share s_1, by running
$$x_1 = \text{Partial-Signature-Generation}(m_{sp}, 1, s)$$
- **Step 2-MS:** Similarly, *MS* signs its message m_{ms} (resulted from *Step 1-MS*) by computing its partial RSA signature x'_1, using its RSA private key share s_4, by running:
$$x'_1 = \text{Partial-Signature-Generation}(m_{ms}, 4, s)$$
- **Step 2-BT:** Note that *BT* receives messages from both *SP* (i.e., m_{sp}) and *MS* (i.e., m_{ms}). It is responsible to partially sign their messages. For m_{sp}, *BT* generates two partial signatures x_2, x_3 using the two RSA secret shares it received from *SM* as:
$$x_2 = \text{Partial-Signature-Generation}(m_{sp}, 2, s)$$
$$x_3 = \text{Partial-Signature-Generation}(m_{sp}, 3, s)$$
Similarly, for m_{ms}, *BT* generates the corresponding two partial signatures x'_2, x'_3 as:
$$x'_2 = \text{Partial-Signature-Generation}(m_{ms}, 2, s)$$
$$x'_3 = \text{Partial-Signature-Generation}(m_{ms}, 3, s)$$

Step 3: Verify Data Integrity & Source Data Authenticity. In this step, *SM* is required to verify the *data integrity* of messages received from *SP* and *MS* (in Step 1), and *source data authenticity* of the energy consumption report. The algorithms used in this Step are *Full-Signature-Generation* and *Full-Signature-Verification*, which have been described in Section 2.3.2.

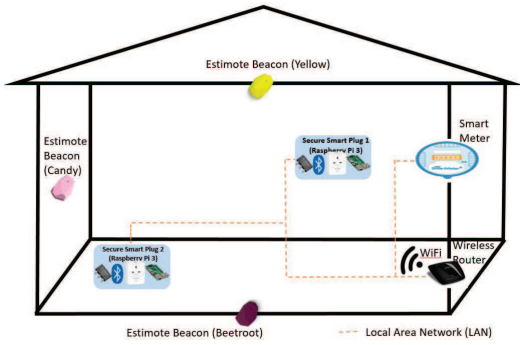

Figure 6: Environmental Setup

- **Step 3-SP:** Upon receiving the partial signature x_1 from *SP* and the other two partial signatures (x_2 and x_3) from *BT* (which implies that the location of *SP* is authentic), *SM* combines all the three partial signatures to generate a full RSA signature FS_{SP} on *SP*'s message m_{sp} by:

  $$FS_{sp} = \text{Full-Signature-Generation}(x_1, x_2, x_3)$$

  Afterwards, *SM* verifies FS_{sp} using the RSA public key *PK*, *Full-Signature-Verification* (m_{sp}, FS_{sp}, PK) to check the authenticity and integrity of the m_{sp}.
  If verification is successful, then the protocol proceeds to *Step 3-MS*; otherwise it terminates and stops further verification.
- **Step 3-MS:** Similarly, upon receiving the partial signature x'_1 from *MS* and two partial signatures (x'_2 and x'_3) from *BT*, *SM* combines them into a full RSA signature FS_{ms}, as:

  $$FS_{ms} = \text{Full-Signature-Generation}(x'_1, x'_2, x'_3)$$

  The full signature is then verified by *SM* to ensure data integrity and authenticity.
- **Step 3-SM:** Finally, in order to check whether there is any sign of data tampering at the source, it is important to ensure that both energy consumption reading reported by both *SP* and *MS* are approximately the same. *SM* checks whether $r_{sp} \approx r_{ms}$ (recall that r_{sp} is one part of m_{sp} and r_{ms} is one part of m_{ms}). If $r_{sp} \approx r_{ms}$, then *SM* is ensured that the energy reading is authentic; otherwise the energy reading should not be trusted and could well be tampered with at its source.

In this way, the smart meter has a good overview of the energy usage from all the power outlets in the household. Similarly, the utility will be able to determine whether there is any energy theft occurring outside the household if the recorded energy usage differs from the smart meter's measurement.

4 PROTOTYPE & IMPLEMENTATION

We have successfully implemented a Java prototype to demonstrate the feasiblity of the proposed solution. This section provides details on the measurement of energy consumption, implementation details of RSA Threshold Signature Scheme and the location verification strategy using *Estimote* Bluetooth beacons.

As shown in Figure 6, the deployed environment was a room consisting of two *Secure Smart Plug* (SSP) measuring the energy consumption at different power outlets, three *Estimote* Bluetooth beacons for determining the location of SSP, as well as a *Smart Meter* (SM) to collect the energy consumption report from the SSP via WiFi. We consider each household to have the same environmental setup in order to secure the data provenance and location authenticity.

4.1 Measurement of Energy Consumption

We used a Raspberry Pi 3 to act as the *SSP* in this prototype. The *Smart Plug* (SP) component in the SSP was a DLink DSP-W215 Wi-Fi Smart Plug that draws energy consumption readings in Kilowatt-hour (kWh). The reading was communicated periodically to the SSP, so that the energy data can be protected and then sent to *SM* for real-time monitoring.

As we do not currently have access to a magnetic sensor, as proof-of-concept prototype the Raspberry Pi acting as the *SSP* was also used to simulate the *Magnetic Sensor* (MS) component in the SSP, thus providing an alternate energy consumption reading to the SP. As the Raspberrry Pi 3 has a Bluetooth chip onboard, it also acted as the *Location Generator* (BT) component in the SSP (c.f. Section 4.3 for details on location authenticity verification).

4.2 Cryptographic Operations

As a proof-of-concept prototype, we implemented the Shamir Secret Sharing (SSS) Scheme and RSA Threshold Signature Scheme using Java Cryptographic Architecture (JCA). The SSS was used in the commissioning phase to split an RSA private key into four secret shares for each *SSP* in the deployment. The RSA Threshold Signature scheme was used in the operational phase to allow for components in the *SSP* to generate partial signatures, and subsequently enabling the *SM* to combine the relevant partial signatures into a full signature for verification.

The current implementation assumes that the communication between the components in the *SSP* are protected. We advocate that for real deployment, *SSP* would be an integrated device with secure bus communication between the *SP*, *MS*, and *BT*, so that when relaying energy consumption data (m_{sp}, m_{ms}) between *SP* and *BT*, as well as between *MS* and *BT*, they cannot be tampered with by the attacker through hardware attacks.

4.3 Location Verification Strategy

As described in Section 3.2.2, during the commissioning phase, the location of the deployed *SSP* is first registered. In the operational phase, for each energy consumption reading to be sent by the *SSP*, the location of the *SSP* at which the energy consumption reading was measured must be validated against the registered location.

Our implementation relied on the deployment of three *Estimote* Bluetooth beacons to accurately determine the location of *SSP* using trilateration method. We observed that the the fluctuation of Bluetooth RSSI signals affects precision of the location of *SSP*, and therefore we developed an innovative idea by defining a *border boundary* and the *nearest beacon found* for each *SSP*. As shown in Figure 7, the *BT* component in the *SSP*, i.e., the Raspberry Pi 3 device first uses trilateration to determine the coordinate of the *SSP*, ensuring that the detected coordinate falls within the defined

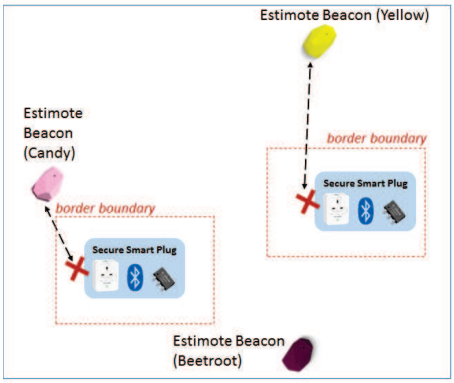

Figure 7: Location Verification Approach

border boundary of the *SSP*, and that the beacon closest to *SSP* is the same as the *nearest beacon found*.

With this, if the *SSP* had been re-located to another household in order to commit energy fraud, this can be easily detected by *BT* because the detected location would not fall within the *border boundary*, and it is highly likely that the *nearest beacon found* will not be the same as the one recorded. In most cases, the signals from the beacons may not even be found if re-located to another household next door.

Once the *BT* has verified that the *SSP*'s location information is authentic, the *BT* will then produce two valid partial signatures to confirm the location authenticity and send the partial signatures to *SM*. Thus, guaranteeing that the *SSP* has not been moved.

Finally, it is important that this *Location Generator*, *BT* is always executed correctly to guarantee location authenticity. We advocate that the *SSP* itself must be further integrated with a secure and tamper-resistant hardware module such as a TPM, and then use a secure bootloader to ensure the integrity of the application has not been compromised.

5 PERFORMANCE ANALYSIS

There are various factors that influence and contribute to the overall performance of any system such as functionality requirements, operation costs and reliability of the program. However, efficiency is no doubt one of the greatest concerns in measuring the performance of information security system as it is infeasible to deploy a security application which is computationally expensive. Hence, the time taken and memory consumption are used in this section to evaluate the efficiency of the proposed scheme.

5.1 Time Efficiency

The computational cost of the proposed scheme was evaluated. In particular, we investigated the cost incurred in both the Commissioning Phase and the Operational Phase.

5.1.1 Commissioning Phase: Computational Time. The *Commissioning Phase* consists of three main tasks to be undertaken by the *SM*, namely: RSA keypair generation, splitting of RSA private key and distribution of key shares to *SSP*. Figure 8 shows the computational time incurred for these three tasks for various key sizes

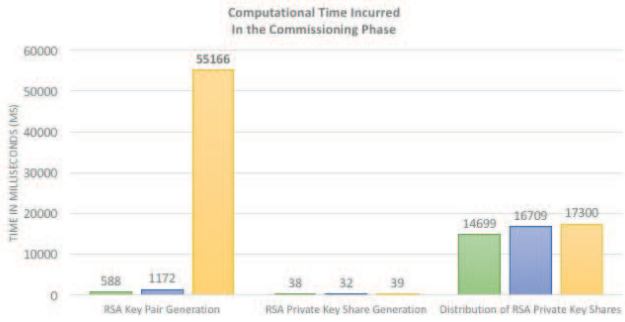

Figure 8: The computational time incurred in the Commissioning Phase for various key sizes

(512-bit, 1024-bit, and 2048-bit) on a laptop serving as *SM*. It is noteworthy that it took 1172 ms to generate a pair of 1024-bit RSA keypair, and the computational time increased significantly when the key size is increased to 2048-bit. In terms of key share generation, the cost of performing Shamir Secret Share generation is rather constant for all three key sizes. The distribution of four secret-shares to the *SSP* takes around 16709 ms, and an increase in the key size does not affect the results.

It is worthwhile to mention that commissioning is a one-time operation at the *SM* (instead of *SSP*) in the proposed solution, therefore, the total time required to complete all three tasks does not affect the practicality of the solution even if the *SSP* may be constrained in computational capability.

5.1.2 Operational Phase: Partial Signature Generation. All the components in *SSP* are required to generate partial signatures during the *Operational Phase* in order to guarantee data provenance and location authenticity of the *SSP*. For each energy reading to be reported, six partial signatures are generated in the proposed solution. We tested the performance of our partial RSA signature generation scheme on Raspberry Pi 3, which is assumed to have low processing power. Furthermore, the limited processing power is also an important factor to examine whether the security protocol can operate efficiently in such environment.

Table 1: Time taken to generate a partial signature

Entity	*SP*, *BT* or *MS*		
Key Size (bits)	**512**	**1024**	**2048**
Average (ms)	148.33	863.67	6419

Table 1 presents the time taken to compute a partial signature on a short message comprising [*timestamp*, *device id*, *energy data*] for different key length. The partial signature generation process was executed three times using different secret shares and the corresponding time incurred was recorded. This is to evaluate whether the partial signature generation algorithm is computationally feasible and to ascertain the effect on the required time when the RSA private key size increases. From the results shown in Table 1, the algorithm to generate partial signature indicates that the computational cost increases as the key size increases. The computational

Table 2: Time taken to combine partial signatures

Entity	*SM*		
Key Size (bits)	**512**	**1024**	**2048**
Average Time (ms)	5	8.33	18.33

time required increases significantly especially when the 2048-bit key was used. This is mainly attributed to the limited computational power of the Raspberry Pi 3 acting as the *SSP*. Having said that, it appears to be reasonable to use a 1024-bit key length, as it only incurred around 860 ms to generate a partial signature.

5.1.3 Operational Phase: Full Signature Generation. Similarly, for every reported energy consumption reading, the *SM* needs to verify its provenance by first combining the relevant partial signatures into a full RSA signature. We conducted performance tests to investigate the computational time required for this operation with three different RSA key sizes. The tests were executed on a laptop simulating the *SM*. The intention to identify whether the operation follows the same manner as it was on Partial RSA Signature Generation (as shown in Section 5.1.2) where time is affected by key size. This also allows more sufficient performance evaluation in order to determine the implemented security protocol is feasible to deploy on limited processing power environment.

Table 2 illustrates the time taken to combine three partial signatures into a full RSA signature. We observe that the computation required and time incurred were not much affected significantly by the key sizes. However, when running these tests on a Raspberry Pi 3 (which has lower computational power), the computational time required appears to be reasonable, e.g., combining partial signatures generated using a share of 1024-bit key, it took approximately 40 ms as compared to 8 ms on a laptop.

5.1.4 Operational Phase: Signature Verification. Table 3 shows the time taken in the *Operational Phase* to verify the combined full RSA signature on *SM*. Results show that the verification is significantly more computational intensive than combining partial signatures.

Table 3: Time Incurred for Signature Verification

Entity	*SM*		
Key Size (bits)	**512**	**1024**	**2048**
Signature Verification (ms)	132	157	875

6 CONCLUSIONS & FUTURE WORK

This paper investigated the data provenance and location authenticity problem in a home energy monitoring network. Particularly, the smart plug which collects and reports the energy value can potentially be compromised to falsify the power consumption before reporting it to smart meter.

We presented a solution to mitigate these attacks, and thus ensuring the provenance of the energy data reported by a secure smart plug to the smart meter. A prototype was implemented and we conducted security and efficiency analysis. We have shown that the proposed system is feasible to be deployed on low computational power devices hence demonstrated its practicality.

While the proposed solution is built on a public-key based solution for a scalability consideration of supporting newly added verifiers, a future work will be investigating the use of a message authentication code (MAC) based solution for single verifier and multiple verifier cases. In the future, we expect to work with a more efficient secret sharing scheme and use general access structure.

REFERENCES

[1] Fadi Aloul, AR Al-Ali, Rami Al-Dalky, Mamoun Al-Mardini, and Wassim El-Hajj. 2012. Smart grid security: Threats, vulnerabilities and solutions. *International Journal of Smart Grid and Clean Energy* 1, 1 (2012), 1–6.

[2] Muhammad Naveed Aman, Kee Chaing Chua, and Biplab Sikdar. 2017. Secure Data Provenance for the Internet of Things. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security*. ACM, 11–14.

[3] Yi Han Ang, Sye Loong Keoh, and Zhaohui Tang. 2016. Privacy-Preserving Spatial and Temporal Aggregation of Smart Energy Data. *Journal of Information Assurance & Security* 11, 4 (2016).

[4] George Robert Blakley. 1979. Safeguarding cryptographic keys. *Proc. of the National Computer Conference1979* 48 (1979), 313–317.

[5] Xavier Boyen and Brent Waters. 2007. Full-domain subgroup hiding and constant-size group signatures. In *International Workshop on Public Key Cryptography*. Springer, 1–15.

[6] Xi Fang, Satyajayant Misra, Guoliang Xue, and Dejun Yang. 2012. Smart gridâĂŤThe new and improved power grid: A survey. *IEEE communications surveys & tutorials* 14, 4 (2012), 944–980.

[7] Luiz MR Gadelha Jr and Marta Mattoso. 2008. Kairos: an architecture for securing authorship and temporal information of provenance data in grid-enabled workflow management systems. In *eScience, 2008. eScience'08. IEEE Fourth International Conference on*. IEEE, 597–602.

[8] Pengfei Gao, Shunfu Lin, and Wilsun Xu. 2014. A novel current sensor for home energy use monitoring. *IEEE Transactions on Smart Grid* 5, 4 (2014), 2021–2028.

[9] Ashish Gehani and Ulf Lindqvist. 2007. Bonsai: Balanced lineage authentication. In *Computer Security Applications Conference, 2007. ACSAC 2007. Twenty-Third Annual*. IEEE, 363–373.

[10] Ayesha Hafeez, Nourhan H Kandil, Ban Al-Omar, T Landolsi, and AR Al-Ali. 2014. Smart Home Area Networks Protocols within the Smart Grid Context. *journal of Communications* 9, 9 (2014).

[11] Rong Jiang, Rongxing Lu, Ye Wang, Jun Luo, Changxiang Shen, and Xuemin Sherman Shen. 2014. Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Science and Technology* 19, 2 (2014), 105–120.

[12] Sye Loong Keoh, Yi Han Ang, and Zhaohui Tang. 2015. A lightweight privacy-preserved spatial and temporal aggregation of energy data. In *Information Assurance and Security (IAS), 2015 11th International Conference on*. IEEE, 1–6.

[13] Nikos Komninos, Eleni Philippou, and Andreas Pitsillides. 2014. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials* 16, 4 (2014), 1933–1954.

[14] John Lyle, Andrew P Martin, and others. 2010. Trusted Computing and Provenance: Better Together.. In *TaPP*.

[15] Jenny Röbesaat, Peilin Zhang, Mohamed Abdelaal, and Oliver Theel. 2017. An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study. *Sensors* 17, 5 (2017), 951.

[16] Cristina Rottondi, Giacomo Verticale, and Christoph Krauss. 2013. Distributed privacy-preserving aggregation of metering data in smart grids. *IEEE Journal on Selected Areas in Communications* 31, 7 (2013), 1342–1354.

[17] Adi Shamir. 1979. How to share a secret. *Commun. ACM* 22, 11 (1979), 612–613.

[18] Victor Shoup. 2000. Practical threshold signatures. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 207–220.

[19] Yu Shyang Tan, Ryan KL Ko, and Geoff Holmes. 2013. Security and data accountability in distributed systems: A provenance survey. In *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*. IEEE, 1571–1578.

[20] Mark Ward. 2014. Smart meters can be hacked to cut power bills. http://www.bbc.com/news/technology-29643276. (2014). Retrieved November 29, 2016 from http://www.bbc.com/news/technology-29643276 Online; accessed 29 November 2016.