Secure Smart Metering Based on LoRa Technology

Yao Cheng[1], Hendra Saputra[2], Leng Meng Goh[1], Yongdong Wu[1]
[1] Institute for Infocomm Research, A*STAR, Singapore.
#21-01 Connexis South Tower, 1 Fusionopolis Way, Singapore 138632
{cheng_yao, gohlm, wydong}@i2r.a-star.edu.sg
[2] Secure Mobile Centre, Singapore Management University, Singapore.
71 Stamford Road, Singapore 178895
hsaputra@smu.edu.sg

Abstract

Smart metering allows Substation Automation System (SAS) to remotely and timely read smart meters. Despite its advantages, smart metering brings some challenges. a) It introduces cyber attack risks to the metering system, which may lead to user privacy leakage or even the compromise of smart metering systems. b) Although the majority of meters are located within a regional power supply area, some hard-to-reach nodes are geographically far from the clustered area, which account for a big portion of the entire smart metering operation cost. Facing the above challenges, we propose a secure smart metering infrastructure based on LoRa technology which facilitates long-range wireless communications. We adopt symmetric cryptography to protect the end-to-end communication between the SAS and the smart meter. Moreover, in order to maintain a long-term security of the proposed metering system, we design a key management protocol to update the keys periodically. Implementation and experiment are presented to evaluate the usability of our system. Finally, the potentiality of the proposed system being applicable to the generalized utility metering is discussed.

1. Introduction

A smart grid is an enhanced power grid that utilizes digital communication technologies to improve the efficiency, sustainability, and reliability in power grids. One of the most salient advantages is that entities in smart grid can mutually communicate with each other in a real-time manner. For example, smart metering, which is a basic function in smart grids, allows a power provider to remotely and timely obtain the user consumption, and accordingly re-allocate the resource to generate the right amount of power to satisfy the user demand without unnecessary waste.

Although smart metering has its advantages, it also brings challenges. Smart metering using modern communication technologies is in the risk of cyber attacks [12]. The meters are distributed and the transmission medium is physically exposed to attackers. The readings collected from meters may face the threat of being *sniffed*, *intercepted*, or *altered*. In order to protect the user privacy and the integrity of meter readings, it is of great importance to take security into consideration. Moreover, although the majority of meters are distributed in clustered power supply areas, there are some hard-to-reach meters far from the clustered area. Data shows that the 1% hard-to-reach meters can account for as much as 50% of the entire smart metering network operational cost [2]. Therefore, it can save up to 50% of unnecessary cost by solving such 1% problem.

In this paper, our research aims to solve the above challenges by proposing a secure smart metering system based on LoRa technology which is a new type of wireless telecommunication network designed to allow long-range communications. We use symmetric cryptography to secure the data during the transmission. To maintain a long-term security feature, we further design a key management protocol to remotely and securely update the encryption keys that are used to protect the transmitted data. We implement the system and conduct experiments to evaluate its usability and efficiency. In summary, we make the following contributions:

- We make the pioneer exploration in designing a smart metering infrastructure based on LoRa providing an end-to-end data protection.
- We propose a lightweight key management protocol

2018 IEEE 4^{th} International Conference on Identity, Security, and Behavior Analysis (ISBA) 978-1-5386-2248-3/18/$31.00

that can update keys securely and remotely, which provides the system with a persistent security over time.

- We implement our design and conduct evaluation experiments which demonstrate the practicality of our design in real-world smart grid scenario.
- We discuss a variant of metering topology considering the battery limit of other meters to facilitate the application of our design to other energy/utility domains, e.g., water metering and gas metering.

2. Background

2.1. Last-Mile Communication in Smart Grids

There are at least three entities involved in a typical smart grid, i.e., power plants, substations, and users. The power is generated at various types of power plants, and then transmitted at high voltage over a long distance to substations which are responsible for reducing the power voltage and distributing the power to users.

Last-mile communication in smart grids normally represents the communication between the smart meter and the data collector. Data collectors are the subnodes of Substation Automation System (SAS, an information system located at the substation) in the smart grid topology. The data collector serves as an agency that collects meter readings directly from smart meters and transmits the readings back to the SAS according to SAS's requests or pre-defined schedules. There are several optional communication channels in the last-mile communication, e.g., PLC (Power Line Communication), low-power RF (Radio Frequency) technology and cellular network. Each channel has its own advantages and disadvantages. PLC supports sending data over the existing power cables. However, it suffers from unpredictable and widely varying channel characteristics which lead to data noise, signal attenuation, and distortion [11][16]. Low-power RF uses certain frequencies with transmit power equal to or less than 50 mW. RF signals are vulnerable to obstructions such as walls and floors. The signal instability and short communication distances caused by abstractions reduce its application to the scenario that many smart meters within single consecutive space such as within single floor. Communication over cellular networks relies on network operators, which can achieve long distance data transmission but with an extra data fee.

The data collector is not the final destination of the meter reading. After the data collector obtains the meter reading, it then connects to its up-layer SAS via another set of communication channels, such as cellular network, Internet and private network. Before the meter reading reaches the SAS, it may still in risk in terms of integrity and confidentiality due to the potential cyber attacks along the transmission channels. Taking this risk into consideration, we redefine the last-mile communication by extending it to the communication between the SAS and the smart meter, so that we can consider it as a whole and propose an end-to-end protection.

2.2. LoRa

LoRa is a wireless communication technology developed to provide the low-power, low-rate, but long-range communication [1]. It uses the free ISM band (the Industrial, Scientific, and Medical radio band) which varies in accordance with government regulations. The communication range of LoRa can hit up to 22 kilometers [4]. It would be helpful to eliminate the "1% hard-to-reach problem" in the smart metering network, which can save as much as 50% of the entire smart metering network operational cost [2]. In addition, statistic data shows that each smart meter only sends 48 messages per day and 12 bytes per message [3]. The requirements of smart metering, i.e., long communication range and light communication load, make LoRa a prevailing choice for this scenario. There are some off-the-shelf LoRa modules, e.g., Libelium SX1272 [4], which provides communication using LoRa technology along with a range of optional frequency bands, coding rates and transmission rates.

3. System Design

The proposed system composes of two parts, i.e., the secure metering via LoRa technology and the key management protocol.

3.1. Adversary Model

Before diving to the technical design, we explain the adversary model first. The attacker may have the capability of sniffing, intercepting, and altering data transmitted over the air or cable. However, (s)he cannot compromise the key management server to obtain the keys of the whole management domain. This is a rational assumption because cryptographic solutions rely on the secrecy of keys.

We assume that the smart meter is not compromised and the reading can reflect the actual power consumption of users. Electricity theft by manipulating power meters happens. This kind of attacks is within the scope of providing effective management by the power utility companies to ensure the integrity of smart meters. Some modern smart meters also provide anti-tamper function. Moreover, unlike traditional power meters, smart meters involve digital systems which may face the threat of being hacked. We leave this problem to another research area, i.e., remote code attestation [9], which focuses on providing digital technique supports for the integrity of smart meter systems.

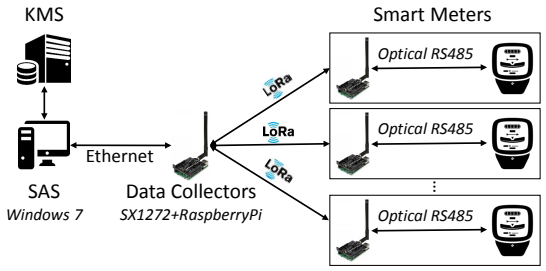

Figure 1. System overview

3.2. System Overview

Our system aims to enable remote metering in a secure and cost-efficient way by adopting LoRa technology and key management protocol. There are three entities involved in this system, i.e., the SAS, the data collector and the smart meter. There can be more than one SAS in smart girds. Each SAS can support a number of data collectors which can further support a number of smart meters. For illustration purpose, we use the case of one SAS, one data collector, and one smart meter to explain our design.

The system overview is shown in Figure 1. Each smart meter is equipped with a LoRa component which is responsible for reading the corresponding smart meter and sending out the reading upon requests. Physically, the LoRa component should be integrated to the existing meter so that it is well protected instead of being exposed outside the smart meter box. Through the LoRa component, the data collector can communicate with the smart meter using wireless LoRa technology. The data collector and the smart meter can be located at different places with a distance. The distance between them, i.e., the distance supported by LoRa technology, can be up to 22 kilometers depending on the environment. Meanwhile, the data collector is connected to SAS via Ethernet.

To secure the transmission of meter readings, we use symmetric cryptography to protect the communication between the SAS and the meter. Each pair of communication counterparts shares a unique secret key. The meter reading is encrypted with the secret key before being sent out. In order to maintain a long-term security, we introduce the KMS (Key Management Server) to manage keys used in the system. An automated periodical key update is feasible using our KMS. The management domain of KMS is limited within a single layer, i.e., managing the keys between one SAS and the meters under its supply. Therefore, there can be multiple KMS systems. In addition, the KMS is a conceptual server which can be deployed on a separate server or integrated to SAS. We detail each component in the following subsections.

3.3. Metering via LoRa

The SAS is responsible for managing metering requests according to the configuration, e.g., requesting power meter readings once a month. The messages shown in the right part of Figure 2 are the metering steps for an SAS to read power consumption from a smart meter. First of all, the SAS needs to establish a connection (Message 1 and 2) and authenticate to the smart meter (Message 3 and 4). Smart meters are provisioned with a secret key when they are deployed. Anyone without the secret key cannot access the smart meter. After successful connection and authentication, the SAS then can issue the reading commands (Message 5 and 6). Finally, the connection is closed (Message 7 and 8).

To explain the data transmission among SAS, data collector, LoRa component, and smart meter, we take establishing connection as an example, which is shown in the left part of Figure 2. The SAS requests to connect to the meter through the data collector which is indexed by IP addresses. Upon receiving the request, the data collector reaches out to the LoRa component attached to the target meter to issue the connection command. The LoRa component is assigned a node ID for index. The corresponding LoRa component forwards the connection command to the smart meter and waits for the response. As soon as the LoRa component receives the response from the smart meter, it sends back the response to the data collector which then feeds the response to the SAS. It works in a similar manner for other messages pairs. Messages with odd IDs are sent from the SAS to the smart meter, while messages with even IDs are responded reversely.

A counter field is included in each message to prevent replay attacks. Communication counterparts store a counter and maintain its increment. Each time an entity receives a message, it compares the received counter with the stored counter. Any situation that a received counter is less than the stored value means a possible replay attack.

We use symmetric keys to protect data transmitted from the SAS to the LoRa component. Each LoRa component attached to a smart meter shares a secret symmetric key with the SAS. By encrypting data using symmetric keys, we avoid any plain text from being transmitted over the air or cable to prevent the data from being sniffed or altered. However, it is not recommended to use fixed keys for a very long time that smart meters usually serve. To solve such problem, we propose a key management protocol to update the keys periodically.

3.4. Key Management Protocol

As shown in Figure 3, we take two layers of the smart grids topology for illustration. There is a secret key sharing between each pair of communication counterparts so that any intruder without knowing the key cannot tamper with

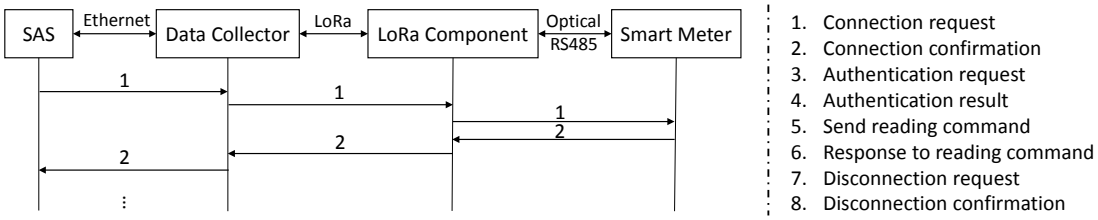

Figure 2. Communication details in smart metering using LoRa technology.

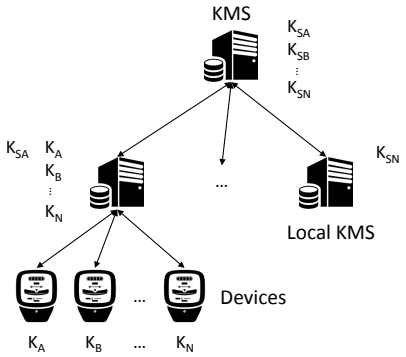

Figure 3. Key management topology. Devices here refer to the LoRa-enabled smart meters.

the transmitted message. The key is managed by the KMS. Instead of centralized, the KMS is distributed following the typical topology of smart grids.

In comparison with the existing key schemes, our key management protocol provides long-term security by maintaining the pairwise secret keys and updating the keys periodically. The existing single-key scheme assigns the same key for all devices in the system for simple management, whose security robustness is low as compromising one device means compromising the whole system [8]. Some key schemes uses pairwise symmetric [10][17] or asymmetric [5] keys. The key is pre-installed on the device. It requires a safe environment to initialize the device, which becomes a problem when the smart meter needs a reset or re-initialization after being installed [7, 6]. Moreover, the fixed key faces the threat of being compromised in the long period of time that a smart meter is supposed to be used for.

In our design, the secret key shared by communication counterparts is not fixed anymore. One secret key K^i only validates for the i-th period. Figure 4 demonstrates the core design of our protocol. We illustrate the protocol using two communication counterparts, i.e., they can be the KMS and the local KMS, or the KMS and the device.

Key Initialization. As shown in Figure 4, at the beginning, the device is initialized with a pre-installed key K_p which is known to its KMS. First of all, the KMS calculates its K^n by hashing its secret K_k, the device identity $DeviceID$, and the offline initialization time $InitTime$, where $H(\cdot)$ is a one-way hash function and n represents the total times of update over the device's management lifetime. Then, the KMS calculates a series of K^i using Equation 1 to finally obtain K^0. After that, K^0 is sent along with the initialization command and a random nonce n_0, encrypted with K_p (m_1 in Figure 4).

$$K^i = H(K^{i+1}), \quad i = 0, 1, 2, ..., n-1 \tag{1}$$

The device decrypts the received message using K_p and obtains K^0. It then responses a confirmation code, a random nonce n_1, and the received n_0, encrypted with K^0 (m_2 in Figure 4).

The KMS decrypts the received message with K^0 and verifies whether the received n_0' equals n_0. If it does, the KMS stores K^0, sets i to 0, sets the expiry time for K^0, and cleans all the intermediate calculation results. A confirmation message with a new random nonce n_2 and the received n_1 is sent back to the device (m_3 in Figure 4).

Upon receiving the confirmation, the device verifies whether the received n_1' equals n_1. If it does, the device stores K^0 and cleans the intermediate calculation results.

Key Updating. The KMS routinely scans for expiring keys and updates them. Once a key K^i is expiring, it needs to go through the key updating procedure as shown in Figure 4. The KMS iteratively calculates K^{i+1} according to the value of i, starting from K^n and following Equation 1. The device calculates whether $H(K^{i+1\prime})$ equals its current key K^i to verify the originality of the new key. The rest of updating procedure is similar with initializing K^0.

4. Security Analysis

End-to-End Security. The communication between the SAS and the smart meter is secured by a pairwise shared secret key in our system. The transmitted meter reading is not available to others, for example, the data collector, because they do not possess the corresponding key. Meanwhile, there is a secret key for each meter, which is only valid for a time period before next update. Therefore, compromising one secret key would not *1)* impact other devices under the same domain and *2)* have an impact longer than the key validation period.

The Self-Verification and Secrecy of Future Keys. The device can efficiently verify the integrity of the new key $K^{i+1\prime}$ by calculating whether $H(K^{i+1\prime})$ is equal to its

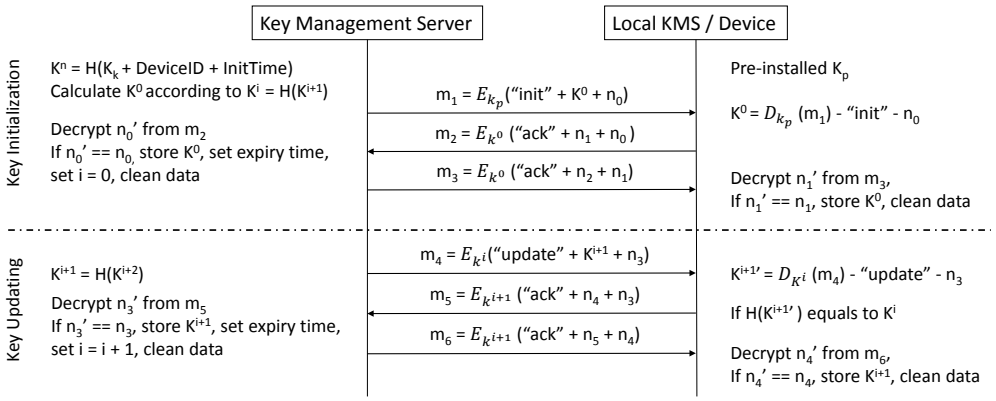

Figure 4. Key management protocol.

current key K^i. Therefore, it requires no trusted third party, e.g., a certificate authority in public key infrastructure, to verify the originality of the new key. Meanwhile, even if the attacker knows K^i, it is computationally difficult for him/her to calculate K^{i+1} due to the one-way feature of hash function $H(\cdot)$ [13]. This one-way virtue ensures the secrecy of future keys and hence the security of future meter readings under the protection of future keys.

Resilience to Replay Attacks. In the key management phase, we use random nonce challenge in each step. If an attacker does not own the encryption key and cannot truly decrypt the message, (s)he cannot obtain the nonce challenge which is supposed to be included in the response. Therefore, any replay would be detected due to the failure in replying the correct challenge. In the metering phase, we introduce counter as mentioned in Section 3.3. The counter value in the received message is expected to be incremental. Any discrepancy, either indicating a network delay or a possible replay attack, leads to a communication failure. In this way, the proposed key management system can be resilient against replay attacks.

5. Experiments

5.1. Experiment Setup

We set up a simplified metering infrastructure to conduct our experiment, including one SAS, one data collector, and one smart meter (Figure 5). The SAS runs on Windows 7, with Intel Core 2 Duo E8400 processor and 8 GB memory. The data collector is a LoRa-enabled device which is connected to SAS via network cable. We use Raspberry Pi 3 Model B which is an affordable and tiny single-board computer to support the program in the data collector. SX1272 LoRa module for Arduino, Raspberry Pi and Intel Galileo (900 MHz) manufactured by Libelium based on Semtech's chipset is connected to Raspberry Pi via Raspberry Pi to Arduino Shields Connection Bridge to provide LoRa capability. SX1272 (900 MHz) supports 13 communication channels with a bandwidth of 2.16 MHz per channel. The model of our smart meter is MPA34D which is manufactured by Mega Power Automation International Limited. It supports remote communications such as 3G, GPRS, and RF, as well as local communication by optical port RS485. In addition, this smart meter provides anti-tamper function. It can detect open meter cover and terminal cover and magnetic disturb. According to our design, we add LoRa component to the MPA34D smart meter through Optical RS485 which is usually used in on-site debugging. In future commercial design, the LoRa component can be integrated into the smart meter chipset.

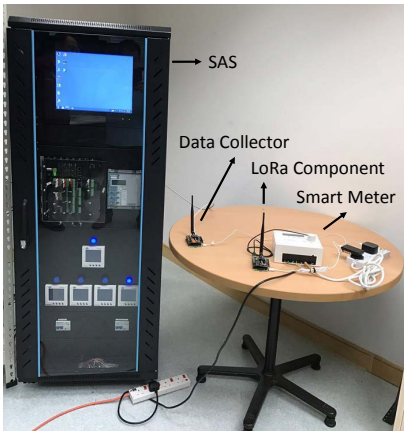

Figure 5. Experiment configuration.

The frequency of LoRa SX1272 should be set to certain radio band range under government regulator guideline, which is from 920 MHz to 925 MHz in our setting. Regarding the SX1272 channels, the allowed channels are channel 08, 09 and 10. We use channel 10 in our experiment. Except for frequency, SX1272 has three configurable parameters [4], i.e., the bandwidth (BW), the coding rate (CR), and the spreading factor (SF). SX1272 offers an option, i.e., transmission mode, that predefines the above three parame-

Table 1. SX1272 transmission mode 4.

Mode	BW	CR	SF	Sensitivity	$Time^1$	$Time^2$
4	500	4/5	12	-128 dB	1167	2040

$Time^1$: Transmission time (ms) for a 100-byte packet.
$Time^2$: Transmissiontime (ms) for a 100-byte packet sent and ACK received

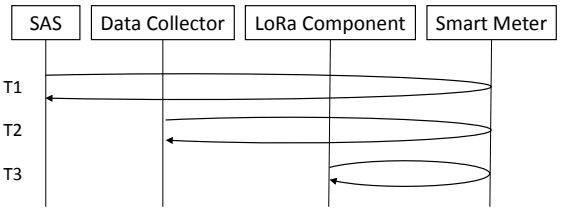

Figure 6. Illustration for test time costs.

ters. We set it to mode 4 (whose predefined parameters are shown in Table 1) reflecting a moderate transmission rate.

We implement our system in C++. 128-bit AES-CBC is used as the encryption algorithm. There are 303, 177, and 268 lines of code added to the SAS, the data collector, and the LoRa component for the smart meter, respectively. The LoRa components communicate in a way that every packet is expected an acknowledge before timeout.

5.2. Performance

We evaluate the performance of the system by measuring the inquiry time of import active energy which is used for billing. In order to accurately measure the time cost based on a synchronized clock, we measure the time cost at each entity separately. As shown in Figure 6, we measure T1, T2, and T3 from the SAS, the data collector and the LoRa component attached to the smart meter, respectively. The time costs are measured starting from the time when a message reaches the entity till the time after the message leaves. T1 represents the time of a single round of query and response. Similarly, T2 and T3 represent the time that a single round of query and response takes between data collector/LoRa component and the smart meter, respectively. T1, T2, and T3 are measured based on the transmission of the same message. The experiment results are shown in Table 2. Multiple experiments demonstrate a steady time cost. It takes about 5.3 seconds for the SAS to issue a reading command and obtain the response. The time cost over the LoRa (T2-T3) is about 2.0 seconds including both request and response. The communication load in smart metering is light according to the statistic data that each smart meter only sends 48 messages per day and 12 bytes per message [3]. In the real-world scenario, the metering frequency is normally once every month for electricity billing purpose and 30 or 60 minutes for the power demand forecasting task. Based on our experiment results in Table 2, one SAS can support more than 300,000 such queries in 30 minutes. Therefore, our system can satisfy the requirement of the real-world usage well.

Table 2. Time costs for querying import active energy.

Measure Period	$T1$	$T2$	$T3$
Time Cost (ms)	5337.9±5.1	3329.1±4.1	1351.0±2.0

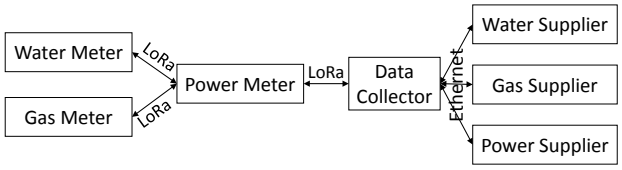

Figure 7. Integrated utility meter reading infrastructure.

6. Discussion

6.1. Generalized Utility Metering

Our solution can be applied to other utility metering scenarios sharing similar infrastructure networks, such as water metering and gas metering. However, the challenge is that different from power meters, water meters and gas meters are not powered. Taking the limited battery into consideration, we propose an overall metering infrastructure based on our metering solution.

The power meter which is usually connected to electricity supply works as a relay for other meters powered by battery, as shown in Figure 7. The data collector here is shared by various utility suppliers. As the data collector is not involved in any data processing and does not possess any secret key, a shared data collector would not affect the overall security. The reading request for all types of meters is sent from the data collector to the smart power meter via LoRa. If the request is to read the power meter, the power meter replies as requested. If the request is for other meter reading, the power meter forwards the request to corresponding meters. It follows the same routine when the response is back. The key management protocol can also work as stated in Section 3.4 with the existence of the power meter relay.

On one hand, it is common for a household to have gas, water, and power supplies. It is environment-friendly to share metering infrastructures which are with similar network topology. On the other hand, the meters powered by battery can save battery by communicating with the power meter which is geographically located nearby and avoiding direct distant communication with the data collector.

6.2. Key Updating Failure

Although the attacker cannot obtain the plain message because (s)he does not own the shared secret key, interception may cause loss of messages, while alteration may cause decryption errors. Therefore, consistent key updating failures which can be caused by various reasons may require

human on-site inspections. To avoid intentionally thwarting key update which may lead to human resource waste, we suggest that the update time should be random even the periodical update interval is averagely fixed.

6.3. Comparison with Existing Key Management Protocols

We compare our work with some recent key management protocols[20, 14, 19, 15, 18].

The *management hierarchy* is compatible across the schemes, although different communication requirements are considered. Long et al. [15], Uludag et al. [18], and our paper account for the role of the central server, while the rest consider only localized communication following typical grid operations. Uludag et al. [18] provide security between control center to end device at the cost of managing extra secret keys, and Long et al. [15] simply route communication between the control center and end device through the SAS. Our key management works in a single-layer way. It involves entities in single hierarchy layer instead of multiple layers, which divides the responsibility for central and local KMS clearly.

In terms of *key scheme*, SKM [19] and Uludag's scheme [18] both involve PKI which is used in combination with the symmetric key scheme. Uludag et al. [18] use the node public/private key to authenticate the exchange of pairwise symmetric keys, which is equivalent to the pairwise secret key in our scheme. In our scheme, the authentication is not provided by the costly PKI, but through the very first offline initialization.

Our scheme uses the pairwise, short-lived secret key directly to secure communications for minimum overhead, while other schemes derive single-use keys from the node key or the channel key in the case of Uludag's [18]. Notably, many schemes do not consider the periodical *key update*. The schemes that do (hash-update [20, 14]) use hash chain, deriving the new key by applying hash function on the current key in order to provide key independence. In contrast, our scheme uses the reverse hash chain to prevent inference of future keys.

7. Conclusion

This paper proposes and implements a secure and cost-efficient smart metering solution, i.e., secure smart metering infrastructure based on LoRa technology. Moreover, a key management protocol with self-verification and future key secrecy is designed to work with the proposed infrastructure to maintain a long-term security by updating keys periodically. The evaluation also demonstrates the practicability of the proposed solution in real-world scenarios.

Acknowledgement

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under the Energy Programme and administrated by the Energy Market Authority (EP Award No. NRF2014EWT-EIRP002-040). We appreciate William Tan, Shaoshen Zhao, and Haoyun You from Mirai for their domain knowledge guidelines.

References

[1] Lora. http://www.semtech.com/wireless-rf/internet-of-things/what-is-lora/.

[2] Semtech announces the industry's first single chip hybrid plc and lora wireless platform for smart grid, smart metering and iot applications. http://investors.semtech.com/releasedetail.cfm?ReleaseID=968700.

[3] Smart metering: Lorawan vs. sigfox vs. weightless-p. https://iot-daily.com/2016/12/09/which-lpwan-technology-is-most-suitable-for-smart-metering/.

[4] Waspmote sx1272 networking guide. http://www.libelium.com/downloads/documentation/waspmote_lora_868mhz_915mhz_sx1272_networking_guide.pdf.

[5] M. Badra and S. Zeadally. Key management solutions in the smart grid environment. In *Wireless and Mobile Networking Conference (WMNC), 2013 6th Joint IFIP*, pages 1–7. IEEE, 2013.

[6] C. Beaver, D. Gallup, W. Neumann, and M. Torgerson. Key management for scada. *Cryptog. Information Sys. Security Dept., Sandia Nat. Labs, Tech. Rep. SAND2001-3252*, 2002.

[7] R. Dawson, C. Boyd, E. Dawson, and J. M. G. Nieto. Skma: a key management architecture for scada systems. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54*, pages 183–192. Australian Computer Society, Inc., 2006.

[8] F. F. Demertzis, G. Karopoulos, C. Xenakis, and A. Colarieti. Self-organised key management for the smart grid. In *International Conference on Ad-Hoc Networks and Wireless*, pages 303–316. Springer, 2015.

[9] X. Dong, S. Jauhar, and B. Chen. Swapguard: A software-only solution for attesting hot-swappable devices in power grids. In *Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on*, pages 357–363. IEEE, 2016.

[10] S. Fuloria, R. Anderson, F. Alvarez, and K. McGrath. Key management for substations: Symmetric keys, public keys or no keys? In *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*, pages 1–6. IEEE, 2011.

[11] S. Galli, A. Scaglione, and Z. Wang. For the grid and through the grid: The role of power line communications in the smart grid. *Proceedings of the IEEE*, 99(6):998–1027, 2011.

[12] E. Hayden. There is no smart in smart grid without secure and reliable communications. *Energy & Utilities, white paper*, 2010.

[13] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.

[14] N. Liu, J. Chen, L. Zhu, J. Zhang, and Y. He. A key management scheme for secure communications of advanced metering infrastructure in smart grid. *IEEE Transactions on Industrial Electronics*, 60(10):4746–4756, 2013.

[15] X. Long, D. Tipper, and Y. Qian. An advanced key management scheme for secure smart grid communications. In *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*, pages 504–509. IEEE, 2013.

[16] P. Mlynek, J. Misurec, Z. Kolka, J. Slacik, and R. Fujdiak. Narrowband power line communication for smart metering and street lighting control. *IFAC-PapersOnLine*, 48(4):215–219, 2015.

[17] X. Nie. A method for secure data transmission in wireless sensor network. Technical report, US Patent 2010293379, 2010.

[18] S. Uludag, K.-S. Lui, W. Ren, and K. Nahrstedt. Secure and scalable data collection with time minimization in the smart grid. *IEEE Transactions on Smart Grid*, 7(1):43–54, 2016.

[19] Z. Wan, G. Wang, Y. Yang, and S. Shi. Skm: Scalable key management for advanced metering infrastructure in smart grids. *IEEE Transactions on Industrial Electronics*, 61(12):7055–7066, 2014.

[20] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato. A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid. *IEEE transactions on instrumentation and measurement*, 64(8):2072–2085, 2015.