Received January 23, 2017, accepted May 18, 2018, date of publication July 5, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2852563

AES-128 Based Secure Low Power Communication for LoRaWAN IoT Environments

KUN-LIN TSAI[1], (Member, IEEE), YI-LI HUANG[2], (Member, IEEE), FANG-YIE LEU[2], (Member, IEEE), ILSUN YOU[3], (Senior Member, IEEE), YU-LING HUANG[1], AND CHENG-HAN TSAI[1]

[1]Department of Electrical Engineering, Tunghai University, Taichung 407, Taiwan
[2]Department of Computer Science, Tunghai University, Taichung 407, Taiwan
[3]Department of Information Security Engineering, Soonchunhyang University, Asan 31538, South Korea

Corresponding author: Ilsun You (ilsunu@gmail.com)

This work was supported in part by the Cross-Ministry Giga KOREA Project through the Korea government (MSIT) (Development of 20Gbps P2MP wireless backhaul for 5G convergence service) under Grant GK18N0600 and in part by the Soonchunhyang University Research Fund.

ABSTRACT Currently, Internet of Things (IoT) as an essential infrastructure proposed for industries and different applications has been popularly applied to different domains, such as healthcare and smart farming, for helping people to do something, aiming to improve our living environments. LoRaWAN, as a Long-Range Wide Area Network specification recommended by the LoRa Alliance, is a low power and long distance communication protocol suitable for IoT environments. This protocol adopts a widely used data encryption method, i.e., Advanced Encryption Standard (AES), developed based on powerful algebra operations and multiple encryption cycles to ensure its communication security. LoRaWAN reduces communication power by setting different transmission latencies for different end-devices; however, AES does not take into account its end device's encryption power. In this paper, a high secure but low-power consumption communication scheme for the LoRaWAN, named the Secure Low Power Communication (SeLPC) method, is proposed to further reduce end-devices' data encryption power by reducing encryption cycles of AES. In the SeLPC, encryption key and D-Box update procedure is presented to enhance security level and simplify the AES encryption process so that the power consumption can be further lowered. Comparing with the traditional AES, the analysis results show that the SeLPC can minimize the encryption power up to 26.2%. The SeLPC can also resist three attacks, including known-key, replay, and eavesdropping attacks and is practically helpful for use in LoRaWAN IoT environments.

INDEX TERMS LoRaWAN, low power, data encryption, AES, Internet of Things.

I. INTRODUCTION

The Internet of things (IoT), as one of the inter-networking infrastructures, often contains physical devices, vehicles, buildings, and other items embedded in IoT objects, like sensors, integrated circuits, software, and actuators. Basically, it enables these objects to collect and exchange data directly or indirectly used by system managers or servers to operate or control some concerned devices. The purpose is achieving specific goals, e.g., giving patients a better medical treatment on time. With the rapid evolution in communication field, various IoT based applications have been proposed, such as environmental monitoring [1], [2], smart factories and smart houses [3], [4], medical treatment and public health [5], smart farming [6], intelligent transportation systems [7] and so on. According to Ericsson Mobility Report 2017 [8], more than 18 billion IoT devices will be connected by 2022. Currently, conventional wireless communication technologies, e.g., 4G, Wi-Fi, Bluetooth, and Zigbee, and traditional Ethernet are the major data exchanging protocols in an IoT system. However, with the development of IoT applications, e.g., in a smart city, the communication technology with features of long distance, high reliability, low power consumption, but very few transmitted data, is more important than before.

Some Low Power Wide Area Network (LPWAN) protocols, including Narrow Band IoT (NB-IoT) [9], LoRaWAN [10], Sigfox [11], Weightless [12], HaLow [13], and RPMA [14], have been proposed for IoT data communication. Among them, NB-IoT and LoRaWAN attract the highest attention. NB-IoT standardized by the 3rd Generation Partnership Project (3GPP) is a narrowband radio technology for IoT. It enables devices and equipment to be linked together by utilizing telecommunications bands. The focuses of NB-IoT are on low communication cost, long battery life, and connecting a huge number of IoT devices. The NB-IoT technology is deployed "in-band" in spectrum which is allocated to Long Term Evolution (LTE), using resource blocks within a normal LTE carrier or "standalone" for deployments in dedicated spectrum.

LoRaWAN, developed by LoRa Alliance, is another attractive LPWAN protocol. It supports long range communication, specific bandwidth, long battery lifetime, and high network capacity, quality of service, and security. Fig. 1 shows the network topology of LoRaWAN, of which components, including end-devices, gateways, network server, and application servers, have been defined. The end-device, which communicates with gateways using LoRa technologies, could be a sensor, meter, monitor, controller, machine, etc. Gateways deliver messages sent by end-devices to the network server or reverse (to operate controller or machine) by using Ethernet, 3G/4G network, or Wi-Fi. Then the network server checks messages' integrity and sends these messages to one of the application servers. On receiving these messages, the application server decrypts messages, and responses with the corresponding action based on the information carried in the messages.

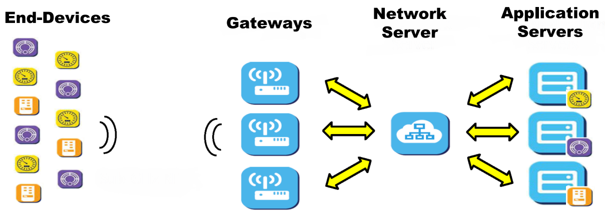

FIGURE 1. **LoRaWAN topology [10].**

In the past decades, many researchers focused on IoT security problems. Chahid *et al.* [15] presented IoT security issues and discussed many solutions. Data integrity as well as communication security may be protected by using modern encryption method [17]; however, the complex encryption steps of those methods also waste a huge amount of energy [18]. Generally, the end-devices in an IoT are often expected to operate with lightweight batteries and have limited energy, memory capacity and processing capability. Hence, in the past few years, lots of studies have introduced various schemes to minimize end-device's power consumption for IoT [19], [20]. However, these studies considering both power consumption and security on a WSN can be further enhanced [21], [22].

To balance power consumption and security, in this study, we propose a secure but low power consumption communication scheme for LoRaWAN, named the Secure Low Power Communication method (SeLPC for short), which lowers data encryption power consumed by end-devices by simplifying their encryption process. The dynamic encryption key as well as lookup table are utilized to enhance the communication security. Comparing with the traditional AES, the analysis results show that the SeLPC can reduce 26.2% of encryption power. Besides, the SeLPC is able to resist known-key attack [23], replay attack [24], and eavesdropping attack [25]. Partial results of this study were published in [26].

The remaining part of the paper is organized as follows. Section II explains the LoRaWAN security specifications. Section III reviews the related studies and background of this paper. The SeLPC is presented in Section IV. In Section V, the encryption power and security features of the SeLPC are analyzed and discussed. Section VI concludes this paper and addresses our future studies.

II. LoRaWAN SECURITY SPECIFICATIONS

LoRaWAN has the properties of long range communication, low power and low cost. According to the LoRaWAN specifications defined by LoRa Alliance, the longest communication distance can be 15 km to 20 km, so that only few gateways are needed in a LoRaWAN environment. The low power feature extends battery lifetime and its non-licenced bandwidth reduces devices' usage cost. For secure communication, LoRaWAN uses modern encryption scheme, i.e., Advanced Encryption Standard (AES) [27], to guarantee its end-to-end security. The basic features include bi-directional authentication, integrity checking, and data encryption. Bi-directional authentication between end-devices and network servers ensures that only authenticated devices can be connected to LoRaWAN, meaning the eavesdropper and invalid devices cannot be successfully authenticated. Some IoT communication protocols encrypt the data transmitted between the gateway and server, but not for end-device, so as to save end-devices' energy. However, as shown in Fig. 2, LoRaWAN provides data encryption for the end-to-end transmission from end-devices to the network server by using Network Session Key (NwkSKey) and from end-devices to application servers by utilizing Application Session Key (AppSKey).

Fig. 3 illustrates that the MAC header, Frame header and encrypted payload are protected by a Message Integrity Code (MIC) derived from them and AES, where MIC is appended at the end of the message for message integrity checking. Once the message content is falsified, or delivers by a fake device, the MIC calculated by the network server itself will not be equal to the receiving MIC. In addition, the data security between an end-device and the application server is insured by the fact that the end-device encrypts plaintext by utilizing 128-bit AES algorithm with AppSKey,

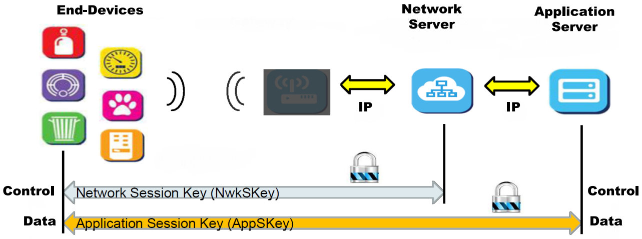

FIGURE 2. Two session keys are used to protect end-to-end security [10].

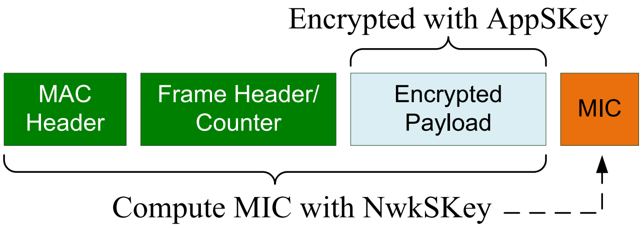

FIGURE 3. LoRaWAN message structure. Two session keys, i.e., NwkSKey and AppSKey, are utilized to encrypt payload and generate MIC [10].

and generates a message with the ciphertext as the payload. The application server decrypts the ciphertext with the same AppSKey. It is worth mentioning that each pair of end-device and network server (application server) has an unique NwkSKey (AppSKey). Besides, the transmissions among gateways, network server and application servers also use TLS (Transport Layer Security) protocol to protect delivered messages.

As abovementioned, the data is encrypted by using AES algorithm which will be described in Section III-D. When an end-device newly joins an existing LoRaWAN, it will be given a unique key (i.e., AppKey (rather than AppSKey), pre-shared with the network server) of 128 bits long and a unique identification number (DevEUI), from which two session keys, i.e. AppSKey and NwkSKey, are derived.

III. RELATED STUDIES AND BACKGROUND

In this section, we introduce background and related studies of this study.

A. IoT SECURITY

IoT extends the Internet technology to many people's lives, but it also accompanies new security challenges and privacy threats [28]. To secure the IoT, many studies [15], [29]–[32] have been proposed. Ning and Liu [29] introduced a cyber-physical-social-based security architecture to deal with three IoT security perspectives, including information security, physical security, and management security. They used the architecture to support unit IoT and ubiquitous IoT, and established an information security model for them. Li *et al.* [30] claimed that besides these three mentioned security issues, the data confidentiality, integrity, availability, and privacy should also be considered. Authors further pointed out that the IoT was a hybrid and heterogeneous network, requiring multi-faceted security solutions, including trust, algorithms, authentications, access control, and governance of frameworks. Based on these requirements, Horton *et al.* [31] examined, developed, and enhanced the secure aspects between a private cloud-based server infrastructure and its associated IoT enabled robots. Riahi *et al.* [32] utilized a triangular pyramid to represent the IoT security, where the four vertexes of this pyramid are persons, technological ecosystems, processes and intelligent objects, and the four planes are used to distinguish the interactions between every triad of vertexes, so that possible research issues could be additionally discussed for future studies.

B. LoRaWAN PERFORMANCE AND SECURITY

Recently, some LoRaWAN related researches [33]–[35] have been proposed. De Silva *et al.* [33] introduced a LoRaWAN architecture and protocol, and compared the battery lifetime, data rate, communication range, security, etc. among LoRaWAN, Sigfox, NB-IoT, and LET-M. Authors continue pointing out that LoRaWAN is better than other LPWAN technologies in power consumption for long range communication. However, more gateways are required to increase its network performance. Bankov *et al.* [34] analyzed the limitations of LoRaWAN performance by using mathematical analysis and experimental simulation. According to LoRaWAN specifications, the total amount of data transmitted by a single gateway increases rapidly when the number of end-device is higher, often increasing message error rates. One of the solutions is to place more gateways in a LoRaWAN; however, it costs higher. Mikhaylov *et al.* [35] also indicated the same problem. The LoRaWAN has high coverage and satisfactory scalability under low uplink traffic, but low reliability, substantial delays and potentially poor performance in terms of downlink traffic. Thus, authors assume that LoRa can be effectively utilized for those moderately dense networks of very low traffic devices which do not impose strict latency or reliability requirements.

Some other studies focused on LoRaWAN security issues [36]–[38]. Miller [36] analyzed possible attacks on LoRaWAN, and asserted that the encryption key generation process and key management policy can be enhanced. In a LoRaWAN, all end-devices, gateways and servers should have their own user verification and key protection policies, so as to guarantee the communication security. Tomasin *et al.* [37] and Aras *et al.* [38] also investigated the security weakness of LoRaWAN. Tomasin *et al.* claimed that replay attack and DoS attack may occur when an end-device is newly added. Besides replay attacks, Aras *et al.* further indicated that long-distance communication may suffer radio jamming and wormhole attacks. Accordingly, many security design issues for current LoRaWAN need to be improved.

Naoui *et al.* [39] proposed a new LoRaWAN security architecture which uses proxy nodes to perform partial functions of gateways. These proxy nodes evaluate the reliability of neighboring proxy nodes and then create a reliability table which is then delivered to all end-devices. According to this table, each end-device selects an available proxy node with the highest reliability to transmit its data. Girard [40] utilized the trusted third party to protect two session keys' generation process. Kim and Song [41] also thought that there are some security problems in session key generation and update process. They used a new network key, named NwkKey, to protect the update process of two session keys without utilizing the trusted third party. Their experimental result demonstrated the security level can be enhanced; however, the key generation and key update processing time also increases, thus consuming more energy than the original system does.

In [39]–[41], it could be seen that in order to enhance the security level of LoRaWAN, some complex operations or procedures are often added. However, it also increases the power consumed by the data encryption, data decryption, and message verification processes. McGrew [42] observed that many encryption algorithms, including AES, are too complex to reduce power consumption of IoT devices. He proposed an authenticated encryption method, named Authenticated Encryption with Replay protection (AERO), for securing communication. The AERO verifies both plaintext and a sequence number. All or partial digits of the variable sequence number are hidden in delivered messages. Thus attackers cannot obtain the encryption key by collecting a large amount of messages. In the AERO, the power consumption can be reduced due to its simple operation. However, the security level of the AERO needs to be confirmed.

C. ENERGY ISSUE ON IoT

Energy consumption is an important concern when an IoT network is under construction. In order to extend the lifetime of an IoT device, many studies [20], [43]–[45] have investigated into energy management of an IoT. The energy consumption of an IoT is mainly from data communication and data processing, including the amount of transmission data, data encoding, analog-to-digital signal conversion, etc. However, the energy consumption is worsened when IoT security needs to be dealt with. Heer *et al.* [46] indicated that complex encryption methods should not be used for IoT so as to balance the network performance and energy consumption. Trappe *et al.* [20] pointed out that IoT end-devices have limited energy and memory space, and conventional cryptography is inappropriate for IoT systems. They suggested reusing existing functions, e.g., using physical layer information to check the location of transmitter and receiver. Salami *et al.* [43] utilized an identity key to encrypt the data in a smart home. They emphasized that the encryption process is simple and does not need complex certification. Bui *et al.* [44] presented a low power AES architecture by utilizing simple shift registers and permutation for key/data storage to reduce circuit size and power consumption. A low-power technique, named clock gating, was also proposed for power saving on S-box.

Currently, security is popularly concerned in many applications, such as cloud based services, smart health care, and so forth [47]–[49]. Nevertheless, adopting complex security (or encryption) scheme on IoT consumes much power/energy for IoT devices; what is worse, it may reduce network performance [50]. It is a trade-off among security, performance, circuit area, network throughputs, and power/energy consumption [51]. Thus, in this paper, the SeLPC is proposed to provide a secure but low-power consumption method for IoT data encryption.

D. AES-128 ENCRYPTION METHOD

AES [27], a symmetric block cipher scheme, is used to protect sensitive data and has a fixed block size. AES supports key sizes of 128, 192, and 256 bits, and consists of 10, 12, and 14 encryption repetition (also known as rounds), respectively. Each round mixes the data with a round-key derived from encryption key. Except last round, each round comprises four processing steps, including SubBytes, ShiftRows, MixColumns, and AddRoundKey.

(1) SubBytes is an invertible and nonlinear transformation, which adopts 16 identical 256-byte substitution tables (i.e., S-box) for individually mapping bytes of the data block into other bytes. S-box entries are produced by calculating multiplicative inverses in Galois Field GF(2^8) and applying an affine transformation.

(2) ShiftRows performs a byte transposition by cyclically shifting rows of the data block according to predefined offsets, i.e., left shift of the second, third, and fourth row by one, two, and three bytes, respectively.

(3) MixColumns multiplies each column of the data block with a modular polynomial in GF(2^8). Instead of computing separately, SubBytes and MixColumns can also be combined into large Look-Up-Tables (LUT).

(4) AddRoundKey transformation adds the data block with round-key derived from initial secret key in the key schedule unit. This function XORed each byte of the block with the corresponding bye in the round-key.

IV. SECURE LOW POWER COMMUNICATION (SeLPC) METHOD

In order to create a secure but low-power communication environment, operations of the SeLPC can be divided into two phases: key generation and data encryption. In AES encryption process, the SubBytes looks up S-Box to encrypt and decrypt data stream. The processing speed is high. Also, given different plaintexts, the ciphertexts generated by the encryption process are sensitively different. However, the contents of the S-Box in AES are fixed, thus greatly reducing its security level since the only nonlinear component of this block ciphering technique is the manipulation on S-Box. To enhance AES's cryptographic strength, an encryption key that generates the corresponding dynamic box (D-Box) to substitute for the primary substitution box (S-Box) is derived. The security of AES is then significantly improved. The D-Box generation process is described below.

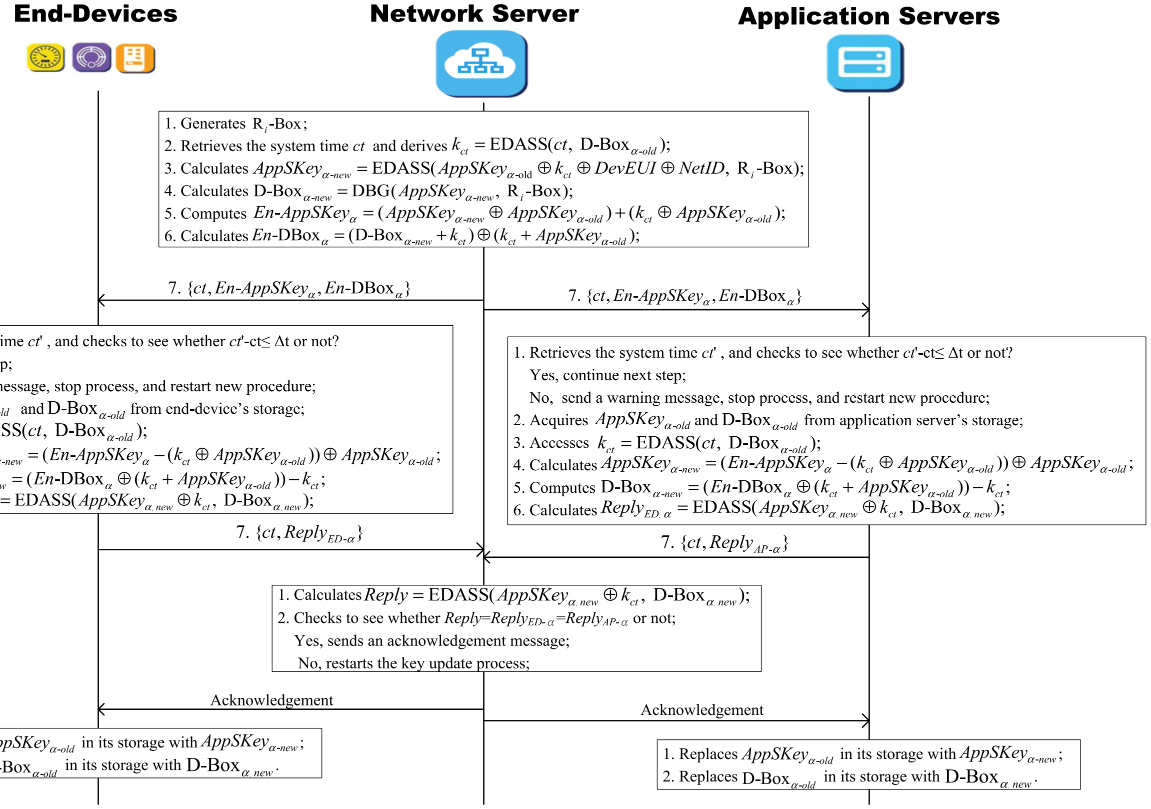
End-Devices
Network Server
Application Servers
1. Generates R_i-Box;
2. Retrieves the system time ct and derives k_{ct} = EDASS(ct, D-Box$_{\alpha\text{-}old}$);
3. Calculates $AppSKey_{\alpha\text{-}new}$ = EDASS($AppSKey_{\alpha\text{-old}} \oplus k_{ct} \oplus DevEUI \oplus NetID$, R_i-Box);
4. Calculates D-Box$_{\alpha\text{-}new}$ = DBG($AppSKey_{\alpha\text{-}new}$, R_i-Box);
5. Computes $En\text{-}AppSKey_{\alpha} = (AppSKey_{\alpha\text{-}new} \oplus AppSKey_{\alpha\text{-}old}) + (k_{ct} \oplus AppSKey_{\alpha\text{-}old})$;
6. Calculates En-DBox$_{\alpha}$ = (D-Box$_{\alpha\text{-}new} + k_{ct}$) $\oplus$ ($k_{ct} + AppSKey_{\alpha\text{-}old}$);
7. {ct, $En\text{-}AppSKey_{\alpha}$, En-DBox$_{\alpha}$}
7. {ct, $En\text{-}AppSKey_{\alpha}$, En-DBox$_{\alpha}$}
1. Retrieves the system time ct' , and checks to see whether ct'-ct≤ Δt or not?
Yes, continue next step;
No, send a warning message, stop process, and restart new procedure;
2. Acquires $AppSKey_{\alpha\text{-}old}$ and D-Box$_{\alpha\text{-}old}$ from end-device's storage;
3. Accesses k_{ct} = EDASS(ct, D-Box$_{\alpha\text{-}old}$);
4. Calculates $AppSKey_{\alpha\text{-}new} = (En\text{-}AppSKey_{\alpha} - (k_{ct} \oplus AppSKey_{\alpha\text{-}old})) \oplus AppSKey_{\alpha\text{-}old}$;
5. Computes D-Box$_{\alpha\text{-}new}$ = (En-DBox$_{\alpha} \oplus (k_{ct} + AppSKey_{\alpha\text{-}old})) - k_{ct}$;
6. Calculates $Reply_{ED\ \alpha}$ = EDASS($AppSKey_{\alpha\ new} \oplus k_{ct}$, D-Box$_{\alpha\ new}$);
1. Retrieves the system time ct' , and checks to see whether ct'-ct≤ Δt or not?
Yes, continue next step;
No, send a warning message, stop process, and restart new procedure;
2. Acquires $AppSKey_{\alpha\text{-}old}$ and D-Box$_{\alpha\text{-}old}$ from application server's storage;
3. Accesses k_{ct} = EDASS(ct, D-Box$_{\alpha\text{-}old}$);
4. Calculates $AppSKey_{\alpha\text{-}new} = (En\text{-}AppSKey_{\alpha} - (k_{ct} \oplus AppSKey_{\alpha\text{-}old})) \oplus AppSKey_{\alpha\text{-}old}$;
5. Computes D-Box$_{\alpha\text{-}new}$ = (En-DBox$_{\alpha} \oplus (k_{ct} + AppSKey_{\alpha\text{-}old})) - k_{ct}$;
6. Calculates $Reply_{ED\ \alpha}$ = EDASS($AppSKey_{\alpha\ new} \oplus k_{ct}$, D-Box$_{\alpha\ new}$);
7. {ct, $Reply_{ED\text{-}\alpha}$}
7. {ct, $Reply_{AP\text{-}\alpha}$}
1. Calculates $Reply$ = EDASS($AppSKey_{\alpha\ new} \oplus k_{ct}$, D-Box$_{\alpha\ new}$);
2. Checks to see whether $Reply$=$Reply_{ED\text{-}\alpha}$=$Reply_{AP\text{-}\alpha}$ or not;
Yes, sends an acknowledgement message;
No, restarts the key update process;
Acknowledgement
Acknowledgement
1. Replaces $AppSKey_{\alpha\text{-}old}$ in its storage with $AppSKey_{\alpha\text{-}new}$;
2. Replaces D-Box$_{\alpha\text{-}old}$ in its storage with D-Box$_{\alpha\ new}$.
1. Replaces $AppSKey_{\alpha\text{-}old}$ in its storage with $AppSKey_{\alpha\text{-}new}$;
2. Replaces D-Box$_{\alpha\text{-}old}$ in its storage with D-Box$_{\alpha\ new}$.

FIGURE 4. The sequence chart of the AppSKey and D-Box update procedure.

A. PHASE 1: KEY GENERATION

To reduce the computational complexity and enhance security level of LoRaWAN, the application layer's dynamic encryption key (AppSKey) and S-Box of AES will be updated every k days, where k can be defined by network manager. The newest AppSKey and S-Box of end-device α are denoted by $AppSKey_{\alpha\text{-}new}$ and D-Box$_{\alpha\text{-}new}$, respectively. Next, two algorithms for one-way key generation and D-Box generation are firstly introduced. Then, the $AppSKey_{\alpha\text{-}new}$ and D-Box$_{\alpha\text{-}new}$ update procedure is presented.

1) ENHANCED DASS ALGORITHM

The dynamic accumulated shifting substitution (DASS) algorithm proposed by Huang *et al.* [52] is a one-way function which encrypts a plaintext into an irreversible ciphertext. However, the shifting counter (ct) in the DASS algorithm is linearly changed and only ranged between 0 and 8 on each time of looking up S-Box. It may result in ineffective defensing Brute-force Attacks [53]. To solve this problem, the Enhanced DASS (EDASS), an enhanced version of the DASS algorithm also encrypting a plaintext into an irreversible ciphertext, is proposed [26] as a part of this study. The inputs of the EDASS algorithm are 128-bit plaintext and a 16*16 random-box. The output is a 128-bit ciphertext. By non-linearly increasing the dynamic shifting count (dsc) and looking up random box, the output ciphertext is highly sensitive to the input plaintext. That is to say that the outputs vary dramatically once the inputs change a little. The complexity of the EDASS algorithm is O(n), where n is 1/8 of input plaintext's size.

2) D-BOX GENERATION ALGORITHM

Liu *et al.* [26] also introduced the D-Box generation (DBG for short) algorithm which, as a part of this study, is used to generate a look-up-table for AES encryption process. The D-Box as a dynamic box is also updated every k days. In the DBG algorithm, 3 inner keys and 3 insertion arrays derived from the 3 inner keys are used to establish the D-Box. A flag array, used to identify the elements of the D-Box to guarantee that each element of the D-Box is unique without colliding with others, is also given. The security of D-Box has also been discussed in [26].

3) APPSKEY AND D-BOX UPDATE PROCEDURE

Those end-devices connected to the same gateway are clustered as a group, and their own AppSKeys and D-Boxes employed to encrypt application layer's payloads will be updated, as mentioned above, every k days. For security reason, different groups may have different ks. When invoking the update procedure, the network server generates new AppSKey and D-Box, and then sends them to all end-devices of this group and their application servers. The update procedure presented as a sequence chart is shown in Fig. 4.

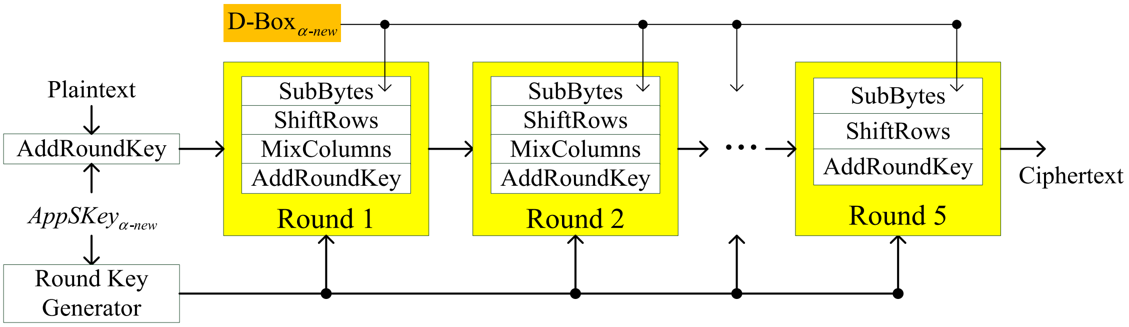

FIGURE 5. The simplified-AES encryption process.

At the beginning of the update procedure, the network server

1. generates a random box R$_i$-Box for the end-device$_\alpha$ ($\alpha = 1, 2, \ldots, m$) connected to gateway G_i;
2. retrieves system time ct from itself and derives a 128-bit time key $k_{ct} = \text{EDASS}(ct, \text{D-Box}_{\alpha\text{-}old})$;
3. calculates $AppSKey_{\alpha\text{-}new} = \text{EDASS}(AppSKey_{\alpha\text{-old}} \otimes k_{ct} \otimes DevEUI \otimes NetID, \text{R}_i\text{-Box})$, where $DevEUI$ is global end-device identifier following the IEEE EUI-64 address space format, and $NetID$ is a 24-bit identifier, of which the 5 LSBs are $NwkID$ used to differentiate addresses of geographically duplicated LoRa networks, and the other bits are determined by network server;
4. calculates $\text{D-Box}_{\alpha\text{-}new} = \text{DBG}(AppSKey_{\alpha\text{-}new}, \text{R}_i\text{-Box})$;
5. computes $En - AppSKey_\alpha = (AppSKey_{\alpha\text{-}new} \oplus AppSKey_{\alpha\text{-}old}) + (k_{ct} \oplus AppSKey_{\alpha\text{-}old})$;
6. calculates $En\text{-DBox}_\alpha = (\text{D-Box}_{\alpha\text{-}new} + k_{ct}) \oplus (k_{ct} + AppSKey_{\alpha\text{-}old})$;
7. sends $\{ct, En - AppSKey_\alpha, En\text{-DBox}_\alpha\}$ to end-device$_\alpha$ and its application server, $\alpha = 1, 2, \ldots, m$.

On receiving the update message, end-device$_\alpha$

1. retrieves the system time ct' from itself, and checks to see whether $ct' - ct \leq \Delta t$ or not, where Δt is a predefined delay including maximal transmission delay and key update processing time. If not, it sends a warning message to network server, stops the key update process, and then the network server will perform a new update procedure; otherwise, it
2. acquires $AppSKey_{\alpha\text{-old}}$ and $\text{D-Box}_{\alpha\text{-}old}$ from its own storage;
3. accesses $\text{D-Box}_{\alpha\text{-}old}$ and time key k_{ct} by using ct carried in the receiving message;
4. calculates $AppSKey_{\alpha\text{-}new} = (En - AppSKey_\alpha - (k_{ct} \oplus AppSKey_{\alpha\text{-}old})) \oplus AppSKey_{\alpha\text{-}old}$;
5. computes $\text{D-Box}_{\alpha\text{-}new} = En\text{-DBox}_\alpha \oplus (k_{ct} + AppSKey_{\alpha\text{-}old}) - k_{ct}$;
6. calculates $Reply_{ED-\alpha} = \text{EDASS}(AppSKey_{\alpha\text{-}new} \oplus k_{ct}, \text{D-Box}_{\alpha\text{-}new})$;
7. sends $\{ct, Reply_{ED-\alpha}\}$ to the network server.

The application server executes the same process as that performed by end-device$_\alpha$, and sends $\{ct, Reply_{AP-\alpha}\}$ to the network server at Step 7. When receiving the reply messages individually sent by end-device$_\alpha$ and application server, the network server

1. calculates $Reply = \text{EDASS}(AppSKey_{\alpha\text{-}new} \oplus k_{ct}, \text{D-Box}_{\alpha\text{-}new})$;
2. checks to see whether $Reply = Reply_{ED-\alpha} = Reply_{AP-\alpha}$ or not;

   If yes, it sends an acknowledgement message to each of end-device$_\alpha$ and the application server; Otherwise, it restarts the key update process;

When receiving an acknowledgement message, end-device$_\alpha$ (the application server)

1. replaces $AppSKey_{\alpha\text{-}old}$ in its storage with $AppSKey_{\alpha\text{-}new}$;
2. replaces $\text{D-Box}_{\alpha\text{-}old}$ in its storage with $\text{D-Box}_{\alpha\text{-}new}$.

B. PHASE 2: DATA ENCRYPTION PROCESS

As mentioned in Section II, LoRaWAN uses AppSKey to encrypt application layer's payloads and NwkSKey to generate MIC code for Mac layer's integrity. For both the two security activities, end-device uses AES-128 encryption method, discussed in Section III-D. The AES-128 repeats 10 encryption cycles. Since AppSKey and D-Box are updated every k days, it is hard for hackers to attack the encryption method. Basically, only 5 encryption cycles is needed. The purpose is to reduce the computational complexity and power consumed by end-devices. Fig. 5 shows the simplified-AES encryption process. Similar to that of traditional AES, each round of the simplified-AES, except Round 5, has SubBytes, ShiftRows, MixColumns, and AddRoundKey steps. The $AppSKey_{\alpha\text{-}new}$ is inputted to Round Key Generator to generate round-keys, and the $\text{D-Box}_{\alpha\text{-}new}$ is referenced by SubBytes step in each round. After the 5th rounds, the ciphertext is generated to be the MAC layer's payload.

V. SECURITY AND POWER ANALYSES

In this section, we first evaluate the security of the SeLPC, including why the SeLPC is secure, and how to protect the security system from replay attack, eavesdropping attack and known-key attack. Then, the power consumption of the SeLPC is analyzed.

TABLE 1. Data encryption power consumption of AES-128 and Simplified-AES.

Encryption step	AES-128		Simplified-AES	
	Rounds	Power (μW)	Rounds	Power (μW)
AddRoundKey	11	17.3	6	9.5
SubBytes	10	1883.0	5	941.5
ShiftRows	10	7.9	5	3.9
MixColumns	9	1694.7	4	753.2
Total Power		3602.9		1708.1

A. SECURITY EVALUATION

The security of the SeLPC is evaluated as follows.

1) THE SECURITY OF THE SELPC

In the EDASS algorithm, an input of 128-bit long is first logically divided into 16 characters and the corresponding ciphertext of an input character is acquired by looking up a random table, according to the sum of the character's integer value and the value of *dsc*. In other words, the EDASS security in looking-up R-Box and substitution steps is conducted by changing *dsc*'s value. It is especially noteworthy that two *dsc*'s values are totally different when two similar plaintexts (with only few differences) are inputted to the EDASS algorithm, resulting in two different ciphertexts. The ciphertext is the corresponding content in the R-Box after looking up the R-Box with the given *ch*. The nonlinear and dynamic accumulative increment of *dsc* truly improves the randomness of *ch*.

In this study, the DBG algorithm generates the dynamic keys by using the EDASS algorithm, as a result, generating a D-Box. Basically, due to invoking the EDASS algorithm which is one with high input sensitivity and randomness, it is hard for hackers to recover the plaintext by manipulating the D-Box. That is, the DBG algorithm is convenient to use and able to improve the security of the D-Box. For the AES-128, the D-Box has 256 elements, indicating that there are 256! possible lookup tables in the SubBytes step of AES encryption. If a hacker would like to decrypt an application-layer message, he/she needs to know the 128-bit AppSKey and D-Box. The possibility of the AppSKey and D-Box combination is up to $2^{128} \times 256!$. We assume the hacker needs n days (normally in years) to successfully attack traditional AES by using state-of-the-art techniques and computers. In the SeLPC, the AppSKey and D-Box are updated frequently, like that mentioned above, every k days, where $n \gg k$. It is very hard for hackers to decrypt the encrypted message without having both AppSKey and D-Box.

2) KNOWN-KEY ATTACK

When a hacker knows AppSKey, the known-key attack may occur, and then the ciphering mechanism can be discovered. In the SeLPC, the application layer's data is encrypted by using AES-128 with AppSKey and D-Box which are updated every k days. If the hacker has obtained previous AppSKey, i.e. $AppSKey_{\alpha\text{-}old}$, he/she still does not know D-Box. Therefore, the encrypted data is safe. Besides, the hacker cannot calculate correct k_{ct} since D-Box$_{\alpha\text{-}old}$ is unknown. Hence, $AppSKey_{\alpha\text{-}new}$ is unable to be derived from $AppSKey_{\alpha\text{-}old}$. Now we dare to say that the SeLPC can prevent the known-key attack effectively.

3) REPLAY ATTACKS

In the SeLPC, the time key k_{ct} is derived from the network server's system time ct in the AppSKey and D-Box update procedure. A replay attack is that a hacker duplicates a valid message transmitted by the network server, and pretends the network server to send the message to an end-device (or the application server), attempting to obtain related information. Two situations may occur. The first on is the hacker transmits the original message to the end-device (or the application server) without modifying it. Then, $ct' - ct \leq \Delta t$ cannot be held since the retransmission delay will make $ct' - ct > \Delta t$. The second situation is the hacker modifies the time ct to make the condition of $ct' - ct \leq \Delta t$ hold. However, the *Reply* message calculated by using $AppSKey_{\alpha\text{-}new}$ and k_{ct} will be different from the one carried in the delivered message (see Steps 1 and 2 when the network server receives both $\{ct, Reply_{ED-\alpha}\}$ and $\{ct, Reply_{AP-\alpha}\}$ from end-device and the application server respectively.) The network server will not send acknowledgement message to end-devices as well as application server in the final step of update procedure, indicating that the SeLPC is able to resist replay attacks.

4) EAVESDROPPING ATTACK

A hacker may extract important information when he/she captures a large amount of messages from the underlying network. In our SeLPC, the AppSKey and D-Box sent by the network server are encrypted by time key k_{ct} and previous AppSKey, i.e., $AppSKey_{\alpha\text{-}old}$. Since k_{ct} varies with time, he/she is still unable to extract AppSKey and D-Box from these messages. Thus, the SeLPC is invulnerable to the eavesdropping attack.

B. POWER ANALYSES

In order to analyze the power consumption of the SeLPC, ARM Cortex-M4 processor [54] and low power content addressable memory (CAM) architecture [55] power data are utilized to simulate encryption process and lookup table, respectively. CAM performs very high speed search but consumes a lot of power during its processing stage. Both Cortex-M4 processor and low power CAM are designed with 90nm technology. Table 1 shows data encryption power

TABLE 2. One-day power consumption by the AES-128 and Simplified-AES.

	Traditional LoRaWAN		SeLPC	
	Times	Power (mW)	Times	Power (mW)
Key update	0/day	0.00	1/day	0.40
Data encryption	48/day	172.94	48/day	81.98
Message integrity	48/day	172.94	48/day	172.94
Total Power		345.88		255.32

consumed by AES-128 and our simplified-AES. The data is collected from the dynamic power consumption of the 4 encryption steps, i.e., SubBytes, ShiftRows, MixColumns, and AddRoundKey, without taking into account the context switch of a processor, I/O operation, memory access, etc. It is clear that the SubBytes and MixColumns consume more power than the other two steps do, since they need to look up S-Box (D-Box) implemented in CAM. Also, the simplified-AES saves 52.6% ($= (3602.9 - 1708.1)/3602.9$) of encryption power compared with that of the AES-128.

Table 2 lists end-device's power consumption in one day. We assume that AppSKey and D-Box are updated every day, i.e., $k = 1$, and end-device sends data to its network server every 30 minutes, i.e., 48 times per day. To achieve data integrity for each message, both traditional LoRaWAN and the SeLPC use AES-128 encryption method with NwkSKey, i.e., 10 repeated encryption cycles. The analytical results show that the SeLPC saves 26.2% ($= (345.88-255.32)/345.88$) of power compared with that of traditional LoRaWAN. In LoRaWAN, each transmitted message from an end-device to the application server is encrypted twice, one for application layer's payload and the other for message integrity. Since NwkSKey is not updated every k days, the power consumption of data encryption (encrypting the application layer's payload) in the SeLPC is reduced, but the power consumption of message integrity (the second row from last) is still the same with that of traditional LoRaWAN. In Table 2, the power consumption of key update is much smaller than others. When k is larger than 1, the power consumption of key update can further be ignored.

Theorem 1: Compared with traditional AES-128 used in LoRaWAN, the SeLPC saves power $(1 - \delta)\%$, where δ is the power consumption raito.

Proof: Assume that the power consumption of AppSKey and D-Box update procedure is P_{KU}, and traditional end-device's data encryption power is P_{EN}. Since the SeLPC uses 5 repeated encryption cycles, the end-device's data encryption power in SeLPC is δP_{EN}, where $0 < \delta < 1$. Assume the data transmission frequency from end-device to application server is m times per day. The power saving percentage R is

$$R = (1 - \frac{m\delta P_{EN} + \frac{1}{k}P_{KU}}{mP_{EN}}) \times 100\%. \tag{1}$$

Since $\frac{1}{k} \ll m$ and $P_{KU} \ll P_{EN}$, the power consumption of key update can further be ignored, and thus

$$R = (1 - \delta) \times 100\%. \tag{2}$$

Q.E.D.

VI. CONCLUSION AND FUTURE STUDIES

In this paper, the AES-128 based SeLPC is proposed to achieve the secure and low-power-consumption goal for LoRaWAN. By periodically updating encryption key (AppSKey) and lookup table (D-Box) on both end-devices and application-server sides, the security level of LoRaWAN communication can be enhanced greatly. Besides, the 10-round AES-128 encryption process is reduced to 5 rounds, so as to save encryption power and extend an end-device' battery life. Our analyses show that the SeLPC is able to save 26.2% of power consumption and resist known-key attack, replay attack and eavesdropping attack. We now conclude that the SeLPC is a secure scheme with the feature of low power. It is practically helpful in protecting LoRaWAN communication and saving its power consumption.

However, in this study, only application layer's data encryption is discussed. The MAC layer's encryption key (NwkSKey) used to generate MIC code is not updated periodically. Besides, the MIC-code generation process has not been simplified. In the future, a secure and low power method for NwkSKey update and MIC code generation will be provided so that the power consumption of end-devices in LoRaWAN can be further minimized, and the security can also be higher than current version. Besides, new end-device joining process will be discussed. These constitute our future studies.

REFERENCES

[1] T. Robles *et al.*, "An IoT based reference architecture for smart water management processes," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, no. 1, pp. 4–23, Mar. 2015.

[2] B. Pokrić *et al.*, "Augmented reality enabled IoT services for environmental monitoring utilising serious gaming concept," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl. (JoWUA)*, vol. 6, no. 1, pp. 37–55, Mar. 2015.

[3] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proc. ACM/EDAC/IEEE Design Automat. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[4] S. P. Tseng, B. R. Li, J. L. Pan, and C. J. Lin, "An application of Internet of Things with motion sensing on smart house," in *Proc. IEEE Int. Conf. Orange Technol. (ICOT)*, Sep. 2014, pp. 65–68.

[5] G. Yang *et al.*, "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Trans Ind. Informat.*, vol. 10, no. 4, pp. 2180–2191, Nov. 2014.

[6] M. Ryu, J. Yun, T. Miao, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Design and implementation of a connected farm for smart farming system," in *Proc. IEEE Sensors*, Nov. 2015, pp. 1–4.

[7] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of Things technologies," *IEEE Wireless Commun.*, vol. 22, no. 6, pp. 122–128, Dec. 2015.

[8] P. Cerwall *et al.*, "Ericsson mobility report," Stockholm, Sweden, Ericsson, Tech. Rep. EAB-17:005964, Jun. 2017.

[9] D. Flore, "3GPP standards for the Internet-of-Things," GSMA MIoT, Huawei, Shenzhen, China, Tech. Rep., Feb. 2016.

[10] Accessed: Apr. 30, 2018. [Online]. Available: https://www.lora-alliance.org/

[11] Accessed: Apr. 30, 2018. [Online]. Available: https://www.sigfox.com/

[12] Accessed: Apr. 30, 2018. [Online]. Available: http://www.weightless.org/

[13] Accessed: Apr. 30, 2018. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-halow/

[14] Accessed: Apr. 30, 2018. [Online]. Available: https://www.ingenu.com/

[15] Y. Chahid, M. Benabdellah, and A. Azizi, "Internet of Things security," in *Proc. Int. Conf. Wireless Technol., Embedded Intell. Syst. (WITS)*, Apr. 2017, pp. 1–6.

[16] K. Islam, W. Shen, and X. Wang, "Wireless sensor network reliability and security in factory automation: A survey," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 6, pp. 1243–1256, Nov. 2012.

[17] K. L. Tsai, F. Y. Leu, and S. H. Tsai, "Data encryption method using environmental secret key with server assistance," *Intell. Autom. Soft Comput.*, vol. 22, no. 3, pp. 423–430, Apr. 2016.

[18] P. Prasithsangaree and P. Krishnamurthy, "Analysis of energy consumption of RC4 and AES algorithms in wireless LANs," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2003, pp. 1445–1449.

[19] J. M. Kim, H. S. Lee, J. Yi, and M. Park, "Power adaptive data encryption for energy-efficient and secure communication in solar-powered wireless sensor networks," *J. Sensors*, vol. 2016, Feb. 2016, Art. no. 2678269.

[20] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the Internet of Things," *IEEE Security Privacy*, vol. 13, no. 1, pp. 14–21, Jan. 2015.

[21] K. L. Tsai, M. Y. Ye, and F. Y. Leu, "Secure power management scheme for WSN," in *Proc. Int. Workshop Manag. Insider Secur. Threats (MIST)*, Oct. 2015, pp. 63–66.

[22] K.-L. Tsai, M. Ye, S.-H. Tsai, Y.-Y. Wang, and Y.-H. Zhuang, "Attack-resistant power management scheme for wireless sensor network," in *Proc. Int. Adv. Robot. Intell. Syst. (ARIS)*, May 2015, pp. 1–4.

[23] B. Cogliati and Y. Seurin, "Strengthening the known-key security notion for block ciphers," in *Proc. Int. Conf. Fast Softw. Encryption (FSE)*, Mar. 2016, pp. 494–513.

[24] P. Syverson, "A taxonomy of replay attacks," in *Proc. Comput. Secur. Found. Workshop (CSFW)*, Jun. 1994, pp. 187–191.

[25] J. K. Tugnait, "Detection of active eavesdropping attack by spoofing relay in multiple antenna systems," *IEEE Wireless Commun. Lett.*, vol. 5, no. 5, pp. 460–463, Oct. 2016.

[26] J.-J. Liu, Y.-L. Huang, F.-Y. Leu, X.-Y. Pan, and L.-R. Chen, "Generating dynamic box by using an input string," in *Proc. Int. Symp. Mobile Internet Secur.*, Oct. 2017, pp. 1–13.

[27] *Announcing the Advanced Encryption Standard (AES)*, Federal Inf. Process. Standards Publication, United States Nat. Inst. Standards Technol., Nov. 2001.

[28] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges, countermeasures, and future directions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 26–33, Jan. 2017.

[29] H. Ning and H. Liu, "Cyber-physical-social based security architecture for future Internet of Things," *Adv. Internet Things*, vol. 2, no. 1, pp. 1–7, Jan. 2012.

[30] S. Li, T. Tryfonas, and H. Li, "The Internet of Things: A security point of view," *Internet Res.*, vol. 26, no. 2, pp. 337–359, Apr. 2016.

[31] M. Horton, L. Chen, and B. Samanta, "Enhancing the security of IoT enabled robotics: Protecting TurtleBot file system and communication," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 1–5.

[32] A. Riahi, E. Natalizio, Y. Challal, N. Mitton, and A. Iera, "A systemic and cognitive approach for IoT security," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 183–188.

[33] J. C. de Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino, "LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities," in *Proc. Int. Multidisciplinary Conf. Comput. Energy Sci.*, Jul. 2017, pp. 1–6.

[34] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Proc. Int. Conf. Eng. Telecommun.*, Nov. 2016, pp. 10–14.

[35] K. Mikhaylov, J. Petäjäjärvi, and T. Hänninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *Proc. Eur. Wireless Conf.*, May 2016, pp. 119–124.

[36] R. Miller, "LoRa security—Building a secure LoRa solution," MWR Labs, London, U.K., White Paper, Mar. 2016.

[37] S. Tomasin, S. Zulian, and L. Vangelista, "Security analysis of LoRaWAN join procedure for Internet of Things networks," in *Proc. Wireless Commun. Netw. Conf. Workshops*, Mar. 2017, pp. 1–6.

[38] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the security vulnerabilities of LoRa," in *Proc. IEEE Int. Conf. Cybern.*, Jun. 2017, pp. 1–6.

[39] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Enhancing the security of the IoT LoraWAN architecture," in *Proc. Int. Conf. Perform. Eval. Modeling Wired Wireless Netw.*, Nov. 2016, pp. 1–7.

[40] P. Girard. *Low Power Wide Area Networks Security*. Accessed: Apr. 30, 2018. [Online]. Available: https://docbox.etsi.org/workshop/2015/201512_M2MWORKSHOP/S04_WirelessTechnoforIoTandSecurityChallenges/GEMALTO_GIRARD.pdf

[41] J. Kim and J. Song, "A dual key-based activation scheme for secure LoRaWAN," *Wireless Commun. Mobile Comput.*, vol. 2017, Nov. 2017, Art. no. 6590713.

[42] D. McGrew, "Low power wireless scenarios and techniques for saving bandwidth without sacrificing security," in *Proc. NIST Lightweight Cryptogr. Workshop*, Jul. 2015, pp. 1–15.

[43] S. A. Salami, J. Baek, K. Salah, and E. Damiani, "Lightweight encryption for smart home," in *Proc. Int. Conf. Availability, Rel. Secur. (ARES)*, Aug. 2016, pp. 382–388.

[44] D.-H. Bui, D. Puschini, S. Bacles-Min, E. Beigné, and X.-T. Tran, "Ultra low-power and low-energy 32-bit datapath AES architecture for IoT applications," in *Proc. Int. Conf. IC Des. Technol. (ICICDT)*, Jun. 2016, pp. 1–4.

[45] C. E. Weng, V. Sharma, H. C. Chen, and C. H. Mao, "PEER: Proximity-based energy-efficient routing algorithm for wireless sensor networks," *J. Internet Services Inf. Secur.*, vol. 6, no. 1, pp. 47–56, Feb. 2016.

[46] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle, "Security challenges in the IP-based Internet of Things," *Wireless Pers. Commun.*, vol. 61, no. 3, pp. 527–542, 2011.

[47] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.

[48] J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, and D. S. Wong, "L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing," *Knowl.-Based Syst.*, vol. 79, pp. 18–26, May 2015.

[49] Z. Cai, H. Yan, P. Li, Z. Huang, and C. Gao, "Towards secure and flexible EHR sharing in mobile health cloud under static assumptions," *Cluster Comput.*, vol. 20, no. 3, pp. 2415–2422, Sep. 2017.

[50] F. Zhang, R. Dojen, and T. Coffey, "Comparative performance and energy consumption analysis of different aes implementations on a wireless sensor network node," *Int. J. Sensor Netw.*, vol. 10, no. 4, pp. 192–201, 2011.

[51] L. Batina *et al.*, "Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures," in *Proc. Int. Workshop Radio Freq. Identification, Secur. Privacy Issues (RFIDSec)*, Nov. 2013, pp. 103–112.

[52] Y. L. Huang, F. Y. Leu, P. H. Su, T. H. Sung, and S. C. Liu, "A secure and high performance wireless sensor network based on symmetric key matrix," in *Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2016, pp. 470–475.

[53] H. Alanazi, B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir, and Y. Al-Nabhani, "New comparative study between DES, 3DES and AES within nine factors," *J. Comput.*, vol. 2, no. 3, pp. 152–157, Mar. 2010.

[54] *Cortex-M4 Technical Reference Manual*, Cambridge, U.K., ARM Ltd., 2009.

[55] K.-L. Tsai, Y.-J. Chang, and Y.-C. Cheng, "Automatic charge balancing content addressable memory with self-control mechanism," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 10, pp. 2834–2841, Oct. 2014.

KUN-LIN TSAI (M'00) received the B.S. degree in computer and information science from Tunghai University, Taichung, Taiwan, in 1999, and the M.S. degree in computer science and information engineering and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, in 2001 and 2006, respectively. He held a post-doctoral position at the National Taiwan University of Science and Technology in 2007. He is currently an Associate Professor and the Chairman of the Department of Electrical Engineering, Tunghai University. His research interests are low-power system design, information security system, and VLSI design.

YI-LI HUANG (M'13) received the master's degree from the Department of Physics, National Central University, Taiwan, in 1983. He is currently an Associate Professor with Tunghai University, Taiwan, and also the Director of the Information Security Laboratory. His research interests include security of network and wireless communication, solar active-tracking system, pseudorandom number generator design, and file protection theory.

FANG-YIE LEU (M'87) received the bachelor's, master's, and Ph.D. degrees from the National Taiwan University of Science and Technology, Taiwan, in 1983, 1986, and 1991, respectively. He was a Visiting Scholar with Pittsburg University. He is currently a Professor with the Computer Science Department and the Chairperson of the Bigdata Master Program, Tunghai University, Taiwan. His research interests include wireless communication, network security, grid applications, and sensor network. He also acts as one of the editorial board members of at least seven journals and serves as the TPC member of at least 10 international conferences. He currently organizes MCNCS and CWECS international workshops.

ILSUN YOU (SM'13) received the M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, South Korea, in 1997 and 2002, respectively, and the Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was with THINmultimedia, Inc., Internet Security Co., Ltd., and Hanjo Engineering Co., Ltd., as a Research Engineer. He is currently an Associate Professor with the Department of Information Security Engineering, Soonchunhyang University. His main research interests include internet security, authentication, access control, and formal security analysis. He is a fellow of the IET. He has served or is currently serving as a main organizer of international conferences and workshops, such as MobiWorld, MIST, SeCIHD, AsiaARES, IMIS, and so forth. He is the Editor-in-Chief of the *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*. He is on the Editorial Board for *Intelligent Automation & Soft Computing*, the *Journal of Network and Computer Applications*, the *International Journal of Ad Hoc and Ubiquitous Computing*, *Computing and Informatics*, the *Journal of High Speed Networks*, and *Security and Communication Networks*.

YU-LING HUANG received the B.S. degree in electrical engineering from Tunghai University, Taichung, Taiwan, in 2016, where she is currently pursuing the master's degree with the Department of Electrical Engineering. Her research interests are information security system, machine learning, smart manufacturing, and VLSI design.

CHENG-HAN TSAI received the B.S. degree in electrical engineering from Tunghai University, Taichung, Taiwan, in 2016, where he is currently pursuing the master's degree with the Department of Electrical Engineering. His research interests are information security system, machine learning, Internet of Things system, and big data analysis.