A SECURE SYMMETRIC KEY SYNCHRONIZATION USING SESSION KEY IDENTIFIER BLOCK ALGORITHM FOR LPWAN NETWORKS

By

G. V. HINDUMATHI *

D. LALITHA BHASKARI **

* Department of Computer Science & Engineering, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India.
** Department of Computer Science and Systems Engineering, Andhra University, College of Engineering, Visakhapatnam, Andhra Pradesh, India.

Date Received: 12/08/2022 Date Revised: 01/12/2022 Date Accepted: 12/12/2022

ABSTRACT

Low-Power Wide Area Networks (LPWAN) have gained considerable importance with the usage and sustainable development of the Internet of Things (IoT). LPWANs have advantages; however, they have many limitations, including limited bandwidth, restricted payload size, and low power devices. Due to these limitations, LPWAN networks confront many challenges, like encrypting the data with the cypher chaining mode or implementing complex algorithms. Here, the cypher chaining mode cannot identify the next message if any packet has been lost because of the device's limited payload size and low memory. So, currently available algorithms cannot be used for LPWAN networks. Key is very important in cryptography; if an attacker compromises a secret key, it leads to a breach of the whole data. The best possible solution to prevent this breach is to frequently change the secret key. Here, we propose a novel algorithm to change the key for every message by synchronizing the sender's IoT device key. This system also proposes a Session Key Identifier Block (SKIB), implementing a session key for every message without any sequence number. The SKIB module identifies the session keys with the hash function and a shared secret key. The proposed work also compared the previous methods with experimental results.

Keywords: Session Key, Low Power Networks, Random Key Generation, Sigfox, Internet of Things.

INTRODUCTION

IoT is a promising technology that connects billions of devices around the world to the internet. These devices may also be called smart devices or smart things since they can communicate independently without depending on human instructions. IoT technology in the present day has a wide range of advantages. Among them, the major advantage is productivity management, which allows for monitoring various distributed applications. Another advantage is simplifying data analysis and reducing human errors by embedding new technologies like artificial intelligence into smart devices. The important technologies that have been used in the IoT are LPWAN, cellular, Zigbee, Wi-Fi, Bluetooth, and RFID. The rest of the paper is categorized as follows:

1. IoT Applications and Security

The main leading applications of IoT technology are smart homes, smart health care systems, smart parking, smart water supply, and smart agriculture systems (AlHammadi et al., 2019; Baranwa & Pateriya, 2016; Marques et al., 2017; Shah & Yaqoob, 2016;). For example, in smart homes, multiple electronic devices have been used in human life, like refrigerators, ACs, ovens, lights, etc. The IoT transmits the data to different networks connected to it, combining two different networks without errors. In this regard, security is critical in the IoT network, which is a result of the rapid expansion of

IoT objects. However, it has some drawbacks concerning the security and privacy of information. It creates security issues due to the lack of complex algorithms in the devices used to collect the data (Kraijak & Tuwanut, 2015). Hence, the IoT is easily attacked by an attacker who obtains the data from the transmission channel and creates some trouble with attacks like masquerade, denial of service, etc. The main security challenges are confidentiality, authentication, access control, privacy, trust, and policy enforcement (Pawar & Ghumbre, 2016). The counter measures for these challenges are applying security algorithms like RSA, AES, and ECC to various technologies used in the IoT.

2. LPWAN

The LPWAN is the most advanced technology in research and industry since it consumes low power, communicates over long distances, and has a low cost to implement. With a rapidly expanding Internet of Things (IoT), LPWAN has become a demanding technology (Mekki et al., 2019). It makes available for communication up to 40 km in rural areas and 5 km in urban regions (Centenaro et al., 2016). Furthermore, it consumes very little energy for communication, enhancing the battery's power for more than ten years of the battery's lifetime (Patel & Won, 2017), and LPWAN is a cost-effective technology (Raza et al., 2017). The updated primary technologies in LPWAN are Sigfox, Long Range Wide Area Network (LoRaWAN) and Narrow Bandwidth – Internet of Things (NB-IoT) based on frequency and distance range.

3. Sigfox

The Sigfox network operator, invented in 2010, establishes wireless networks to connect low-powered devices. These devices are used to produce continuously small amounts of data using Ultra Narrow Band (UNB), a very low-bandwidth transmission method. The battery power of the device is nearly 2400 mAh. So, it makes the life of the device increase to 1.5 or 2.5 years while transmitting the data every 10 minutes at 100 bits/sec or 600 bits/sec, and if the data rate decreases to below 600 bits/sec, then the lifetime of the device increases to 14.6 years (Gomez et al., 2019). There are three main attributes of Sigfox for implementing the network by connecting devices (Fourtet & Ponsard, 2020).

3.1 Low cost

The cost of Sigfox is very low, at two dollars, and bulk production makes it even cheaper.

3.2 Low Volume of Data

To communicate and send data to other devices in the network, the packet volume must be very small. Here, Sigfox used 12 bytes as a data packet volume.

3.3 Massive Number of Connected Objects

The network can connect to more objects because of the low cost and ease of code implementation using IoT devices with low bandwidth.

Here, Figure 1 depicts the overall architecture of Sigfox and gives clear information about the end-to-end data communication procedure. The IoT objects will communicate to the IoT cloud through Sigfox, and then the application server gets the data packet from the cloud and forwards the same to the application process.

4. Related Works

LPWAN networks face various issues with encryption methods, despite their advantages. These networks

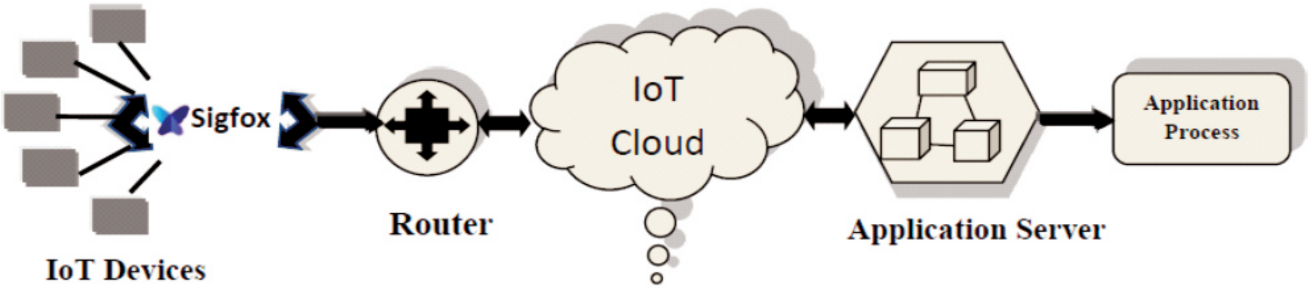


Figure 1. Sigfox Architecture

cannot support complex algorithms since they use devices with low memory capacities and low power consumption. Hence, there is a need to implement lightweight encryption algorithms that can handle packet loss as well. Here, we have mentioned some of the lightweight encryption algorithms for LPWAN, which had been published earlier.

The key distribution methods for asymmetric algorithms, like RSA, Diffie Hellman, and elliptic curve cryptography, among low-power IoT devices have been compared by Goyal and Sahula (2016). Li and Cao (2016) present lightweight algorithms for symmetric encryption methods to simplify the generation of a secret key.

Another study of existing lightweight algorithms is built upon the size of blocks, key size, and functional rounds (Luhach & Luhach, 2016). Another novel proposed design for lightweight algorithms is the key pair attribute-based encryption scheme, proposed by Yao et al. (2015), which can fit IoT networks with resource constraints.

Beaulieu et al. (2015a) implemented different block cypher algorithms for lightweight algorithms; however, they have various resource-constrained environments that could not be suitable for the AES algorithm. This research was proposed by the U.S. National Security Agency (NSA). Singh et al. (2017) discussed the primitives of lightweight algorithms: block cypher methods, hash functions, high-performance systems, stream ciphers, and resource-constrained environments.

Implementing a block cypher is quite easy and applies deterministic algorithms with a secret key. There is no preference for using in-stream cyphers that are used to encrypt one bit. Electronic Code Book (ECB) mode, a basic mode for the first generation of block ciphers, is easy to implement compared to Cipher Block Chaining (CBC) mode. However, the main disadvantage of ECB is the lack of distribution of Ciphertext, as it generates the same Ciphertext for the same plaintexts; it does not hide the data samples. The best solution for this problem is to implement CBC mode, where all blocks of plain text have to be encrypted interdependently to generate the Ciphertext. Hence, some extra complexity for plain text is added when compared to ECB—using a unique secret key to encrypt each block, which is part of the plain text, and it depends upon the position in the chain. Here, the encryption algorithm of each block strongly depends on the previous block Ciphertext and a distinct key assigned to that particular block. Hence, on the receiver's side, initially, it generates the unique key from the previous ciphertext content to decrypt the next cypher block (Sehrawat & Gill, 2018). Some novel solutions for implementing block chaining are the cypher block chaining mode for small-size blocks, feedback mode, output feedback mode, propagating cypher block chaining, and the message-based key distribution method (Beaulieu et al., 2015b; Dinu et al., 2019; Hanounik, 2006; Hindumathi & Bhaskari, 2019; Hwang & Gope, 2015; Perlman et al., 2016).

The negative aspect of the above solutions is that the receiver cannot decrypt the received Ciphertext if the attacker's prior Ciphertext might be compromised or lost in the IoT network. These methods can be suitable for connection-oriented networks like TCP.

For this problem, two solutions could be used to implement block cypher chaining: The first solution is to get the multiple keys to store as many blocks as possible in plain text, which causes high memory consumption. Second, independent Ciphertext generation with unique keys, which results in not relying on the previous Ciphertext to decrypt the received cypher block and in generating a new key for the received Ciphertext without storing multiple keys in low memory capability devices. Datagram Transport Layer Security (DTLS) (Rescorla & Modadugu, 2012) uses the sequence number of a datagram to generate the key.

The Counter Mode Encryption (CTR) (Lipmaa et al., 2000) proposes to make the key for decrypting the Ciphertext by using a pseudo-random number generator function with the input of the master key or a previously existing key. The next method for this problem is the One Time Pad (OTP) (Bellovin, 2011) technique, which implements a Linear Feedback Shift Register module to create a new key based on a previously implemented key. However, the drawback of an OTP method is that the receiver must be conscious of the sequence number of every block received to make an appropriate key; otherwise, the receiver decrypts the cypher block with the incorrectly generated key.

Another solution is that a message digest would be generated for a block that appends to the block, thereby permitting the receiver to identify the appropriate key. There was a novel approach to de facto methods and patents (Epstein, 1996; Tehranchi, 2007; Wilson, 1993). Epstein and Tehranch had obtained patent rights for an encryption module that generates a key for each block using a random number. Hence, an attacker could not decrypt the block simply by knowing the random number attached to the block, as an attacker is unaware of the keys. These blocks use a fixed output structure that cannot work in resource-constrained networks. All the above-discussed proposed methods use one of the following techniques to implement their models for finding the appropriate keys. They use random numbers and a message digest to obtain message acknowledgement and use sequence numbers attached to blocks.

Even though these are not suitable for LPWAN networks because LPWAN uses a very narrow band and the sender needs to acquire acknowledgement from the receiver. Hence, both parties in the network could not identify the loss of message in the network. The user could not retrieve any sequence number of a message using the Sigfox protocol because it does not have any command to identify the sequence number of a message. Due to the fixed payload size of the LPWAN, it cannot use the embedded sequence number in the message method because the sequence number itself occupies some extra memory space in the message. Due to the fixed payload size of the LPWAN, it cannot use the embedded sequence number in the message method or message digest method because the sequence number or message digest itself occupies some extra memory space in the message.

Bidgoly and Bidgoly (2019) proposed a novel key synchronization module using a hash function. The main advantage of this method is that there is no problem with a payload since it is not embedding any extra information about the key in the block that is sent to the receiver. In this approach, the entire message is divided into two parts. It uses two different keys to decrypt the entire message: one is the secret key that was shared earlier, and the other is generated using a hash function. The first part of the message was decrypted directly by the secret key. However, the second part applies the hash function to the already decrypted part of the message. It compares this hash value with the key generated by the random key generator function, and the secret key was the input for the random key generator function. The drawback of this method is that it generates multiple keys to compare with the hash equivalent value of the decrypted part of the message to identify one key to decrypt the second part of the message. LPWAN is used to implement lightweight algorithms because of its constraints. So, in this paper, a novel key synchronization module with fewer computation cycles is proposed and implemented.

5. Proposed Work

The main objective of the present paper is to provide a lightweight algorithm for low-power devices mainly used for key synchronization while transmitting data. In the reference paper, we have implemented some enhancements in the computation of the Appropriate Key Finder (AKF) module (Bidgoly & Bidgoly, 2019). Moreover, we are glad to announce that we are getting better results with the AKF algorithm. The proposed system architecture for encryption is given in Figure 2. The encryption contains two permutation boxes, two split boxes, one for plain text and another for the secret key, one merge box, two-block encryptions, and mainly a key generator function. Here, any lightweight symmetric algorithm could be used, which can be applied to block encryption. On the sender side, the original input data is split into multiple parts to send the encrypted message to a receiver.

The salient block of architecture is the Session Key Identifier Block (SKIB) module, which generates a key for every part of the input data. The key process behind this architecture is to split the entire data into two segments: one segment is directly encrypted, while the other is

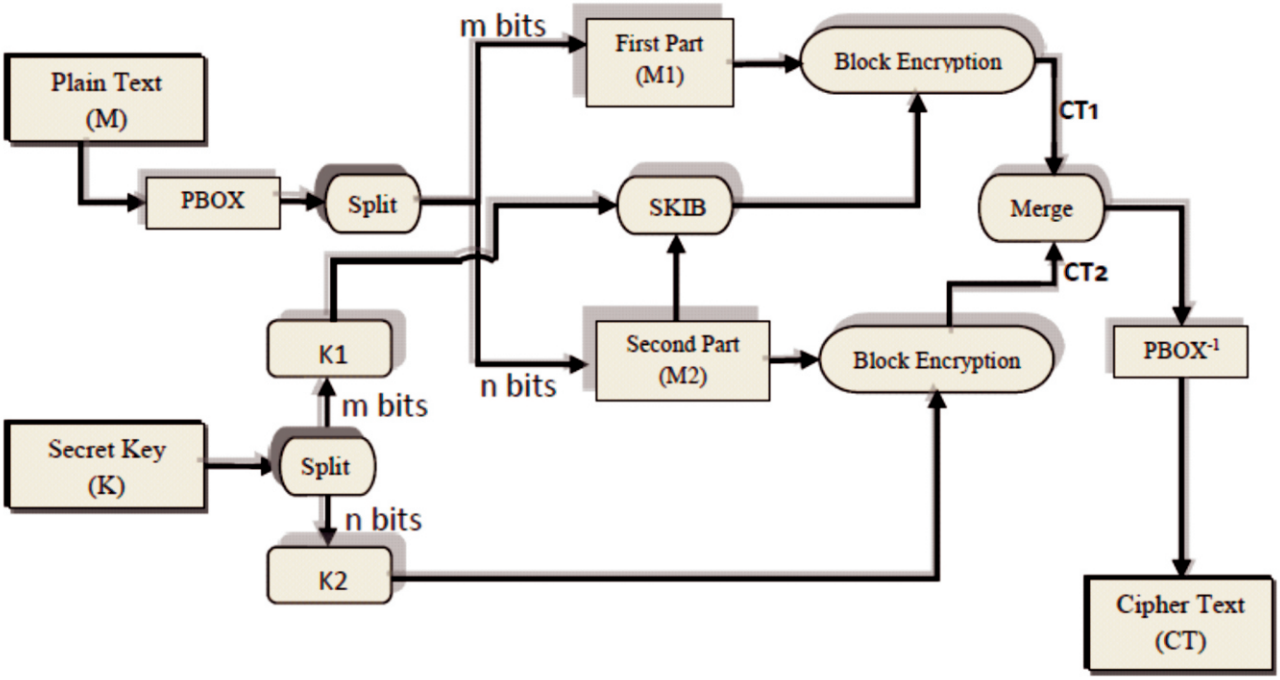


Figure 2. Encryption Process at Source IoT Device

dedicated to synchronizing between the two parties. Both parties should agree on a secret key (K) before the process starts. Prior to encryption, the input data is permuted using PBOX and then split into two chunks represented by M1 and M2 of m and n bits length, respectively. The secret key is divided into keys K1 and K2 with m and n bits, respectively. M2 is directly encrypted using a block encryption module by key K1, but M1 is encrypted using block encryption with a key generated through SKIB.

SKIB is the main building block for this entire system, as mentioned earlier in the paper. The main responsibility of SKIB is to determine the Session Key (SK) to encrypt the M1 block in less time. Figure 3 is a flowchart that elucidates the complete information about SKIB.

The SKIB can identify the session key by computing H(M2), where H represents a hash function, and, finally, XORing it with the K1 key that has been computed to get the appropriate session key to encrypt the message M1 and merge both texts after encryption. The Ciphertext will be generated for this text after applying PBOX^{-1}. The complete Encryption algorithm with SKIB has given as algorithm in Figure 4.

At the receiver's side, initially, the inverse of PBOX (PBOX^{-1}) should apply to the Ciphertext and split the entire text into two parts, CT1 and CT2. Figure 5 elucidates the decryption module in the proposed system. Message M2 has been calculated for Ciphertext CT2 by using a decryption block to split the secret key K2. Both parties could already synchronize that secret key. Both parties could already synchronize that secret key.

SKIB is used to decrypt the message M1 by using M2 and

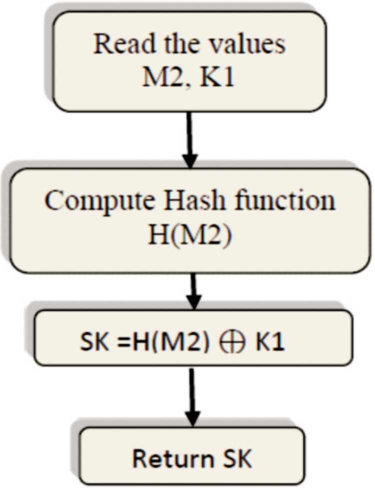


Figure 3. Session Key Identifier Block (SKIB)

```
         Input:  Plain Text (PT), Secret Key (K)
         Output: Cipher Text (CT)
         Begin
Step 1       Apply Permutation function (PBOX) to PT
Step 2       Divide PT and K into two different parts m,
             n with respective to n<<m
Step 3       Select second part of the PT (M2), First part of the key (K1)
Step 4       Compute SKIB function by using M2 and K1,
             So Session Key (SK)=H(M2) ⊕ K1
Step 5       Encrypt first part of the PT (M1) with using computed the
             SK so produced value is CT1
Step 6       Encrypt M2 with using second part of the key (K2) that is CT2
Step 7       Compute CT = (CT1 || CT2)
Step 8       Apply inverse permutation function (PBOX-1) to CT
         End
```

Figure 4. Encryption Procedure with Using SKIB Function

K1 for Ciphertext CT1. Finally, merge the M1 and M2 texts and apply the inverse permutation using PBOX (PBOX). Here, the decryption algorithm is enclosed in Figure 6.

Because the receiver can decrypt the Ciphertext of a block even if any part of the block is lost in the network, our proposed system has achieved complete key synchronization between two parties connected in the LPWAN network. The receiver synchronizes the sender with M2, which is part of the key and can be identified as the session key. This session will be used for the SKIB module for the next message, even if a message is lost.

6. Experimental Results

Based on the experiments, the timing, encryption process of the IOT, decryption process on the receiver's side, and comparison of the two algorithms were obtained.

6.1 Timing

Mathematical equations are given as follows,

Compute the time for the equivalent frequency and it is given by the Equation (1),

$$T = 1/F \qquad (1)$$

$$\text{So, } T = \frac{1}{1000\text{KHz}} = > 1\mu\text{s};$$

The given frequency is 1000 kHz (1 MHz), and one machine cycle is executed in 1 microsecond.

To compute the SKIB, we first need to calculate the time for the hash function, and hence the time of the hash function H(M) is given in Equation (2),

$$T_{(M)} = \text{Denotes no.of machine} \qquad (2)$$

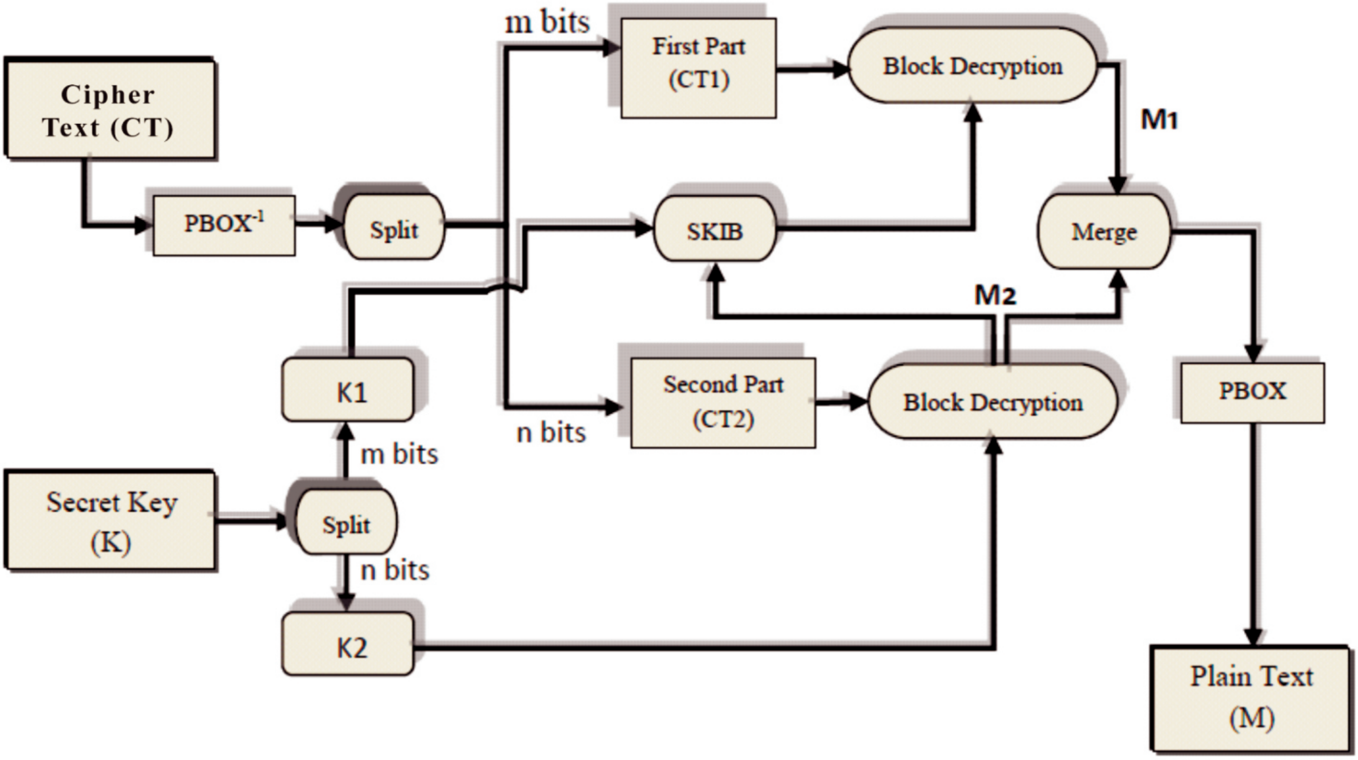


Figure 5. Decryption Process at Receiver's Side

```
        Input:   Cipher Text (CT), Secret Key (K)
        Output: Plain Text (PT)
        Begin
Step 1      Apply Inverse Permutation function (PBOX-1) to CT
Step 2      Divide CT and K into two different parts m,
            n with respective to n<<m
Step 3      Decrypt second part of the CT (CT2) with
            second part of the key (K2) is M2
Step 4      Select first part of the key (K1)
Step 5      Compute SKIB function by using Session Key (SK) =H (M2)⊕K1
Step 6      Decrypt first part of the CT (CT1) with using computed
            SK value is M1
Step 7      Compute PT = (M1 || M2)
Step 8      Apply Permutation function (PBOX) to PT
        End
```

Figure 6. Decryption Procedure with Using SKIB Function

cycles required for H(M)*T

Then calculate the Session Key (SK) for encrypting or decrypting the message and the total session key is given as Equations (3) and (4),

$$SK = H(M) \oplus K \tag{3}$$

$$T_{SK} = \text{Total no.of machine cycles required for SKIB function*T} \tag{4}$$

In our system, complete plain text (M) is divided into two parts, M1 and M2. And the secret key (K) also has two parts, K1 and K2.

6.1.1 Encryption Process at IoT Device

In the encryption process at the IOT device, the first and second parts of the cypher text have been given in Equations (5) and (6).

Compute the first part of the cipher text

$$Ct1 = E(M1, H(M2) \oplus K1) \tag{5}$$

Compute the second part of the cipher text

$$CT2 = E(M2, K2) \tag{6}$$

Combining Equations 5 and 6 results in Equation (7) as,

$$CT = E(M1, H(M) \oplus K1) || E(M2, K2) \text{ OR } CT = CT1 || CT2 \tag{7}$$

The total encryption is given as a formula in Equation (8) as,

$$T_{Enc} = \text{Total no.of machine cycles to encrypt the plaintext*T} \tag{8}$$

6.1.2 Decryption Process on Receivers Side

Divide the received Ciphertext into two parts, CT1 and CT2, and the secret key is also split into two parts, K1 and K2.

Initially decrypt the second part of the cypher text, given as Equation (9),

$$M2 = D(CT2, K2) \tag{9}$$

Decrypt the first part of the Ciphertext using M2, which will be provided by Equation (10),

$$M1 = D(CT1, H(M2, K1)) \tag{10}$$

Then combination of Equations 9 and 10 results in Equation (11) is given by,

$$PL = M1 || M2 \tag{11}$$

The total decryption is given as formula in Equation (12) as,

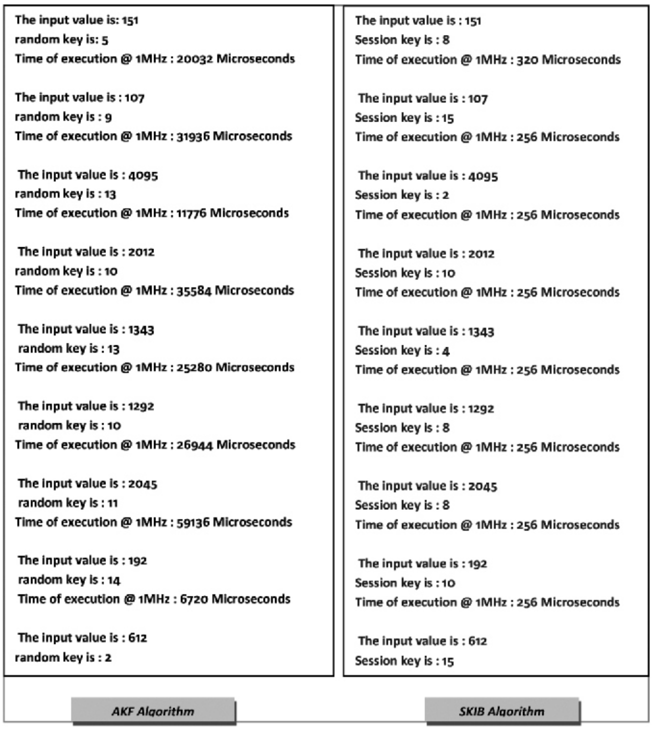
$$T_{Dec} = \text{Total no.of machine cycles to decrypt the ciphertext*T} \tag{12}$$

6.2 Comparative Studies

This proposed method was executed and produced encouraging results to evaluate the object of our proposed architecture. This proposed work uses the block-chaining model to implement the algorithms. Moreover, we implemented this system through the ATmega328P microcontroller to calculate the performance of the proposed encryption algorithm. The proposed algorithm is uploaded to the controller and designed into a new prototype. The sample data was collected from sensors and divided into 12 bytes so that Sigfox could handle transmitting the small packets.

The proposed system, 12-byte plain text, is divided into two parts, m and n. Applying AKF and SKIB functions to the same input data has produced the computed execution times by applying various keys as given in Figure 7. It directly shows the execution times of both algorithms in microseconds.

Based on the data produced by the IoT device and plotted, a graph is shown in Figure 8. The graph itself compares two algorithms and elucidates the best one to implement further. The AKF takes more time to complete its function because only the pseudo-random generator function has been used in the Appropriate Key Finder (AKF). The random function randomly generates a number, computes its hash value, and compares it with the message. However, if that value is not the same as the



```
The input value is: 151
random key is: 5
Time of execution @ 1MHz : 20032 Microseconds

The input value is : 107
random key is : 9
Time of execution @ 1MHz : 31936 Microseconds

 The input value is : 4095
random key is : 13
Time of execution @ 1MHz : 11776 Microseconds

 The input value is : 2012
random key is : 10
Time of execution @ 1MHz : 35584 Microseconds

 The input value is : 1343
 random key is : 13
Time of execution @ 1MHz : 25280 Microseconds

 The input value is : 1292
 random key is : 10
Time of execution @ 1MHz : 26944 Microseconds

 The input value is : 2045
 random key is : 11
Time of execution @ 1MHz : 59136 Microseconds

 The input value is : 192
 random key is : 14
 Time of execution @ 1MHz : 6720 Microseconds

 The input value is : 612
random key is : 2
```

AKF Algorithm

```
The input value is : 151
Session key is : 8
Time of execution @ 1MHz : 320 Microseconds

 The input value is : 107
Session key is : 15
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 4095
Session key is : 2
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 2012
Session key is : 10
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 1343
Session key is : 4
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 1292
Session key is : 8
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 2045
Session key is : 8
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 192
Session key is : 10
Time of execution @ 1MHz : 256 Microseconds

 The input value is : 612
Session key is : 15
```

SKIB Algorithm

Figure 7. Execution Time in Microseconds for Both AKF and SKIB

message, it will generate a new one. So it takes more time to compute the AKF. However, in the case of SKIB, we were not using the random function and implemented a hash function to give the message but not a key. So the execution time was drastically reduced compared to the AKF algorithm. Finally, this research paper used approximately 4000 samples to run these algorithms, and in Figure 8, we plotted the graph for various groups of packets, where each packet is 12 bytes wide for both algorithms.

Furthermore, it specified the throughput of both algorithms, which are AKF and SKIB. The AKF algorithm gradually increases its execution time compared to our proposed system. The SKIB is being executed with a minimum amount of time for the message with any key that was used here. Figure 9 shows comparing the execution periods with number of executed packets for AKF and SKIB. Table 1 can be used to list all annotations for the entire paper.

7. Discussions

With LPWAN networks advancing and thriving, significant data will be produced by the different sensors connected to the IoT devices from a structural perspective. LPWAN networks have many advantages, including a broad range of operations, are supported economically, and use very little energy for computation. Despite these advantages, they have many limitations, including a restricted send rate and a small payload size. Due to these limitations, LPWAN requires strong security

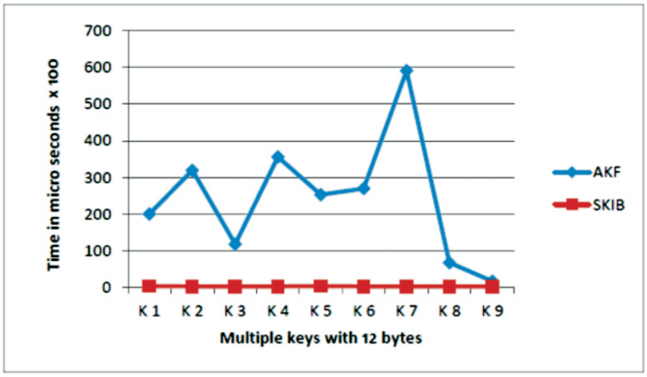


Figure 8. Comparing the Execution Periods with Various Keys For AKF and SKIB

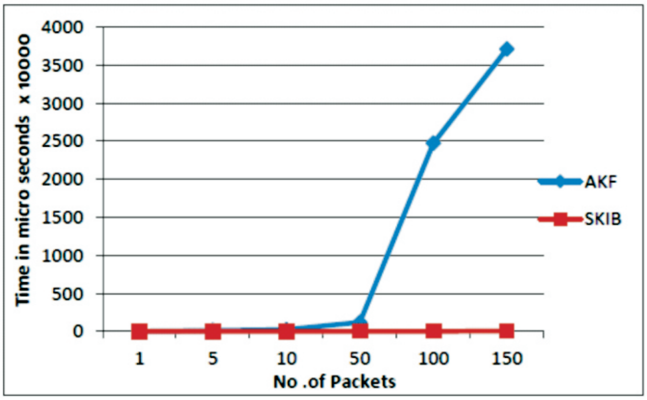


Figure 9. Comparing the Execution Periods with No. Of Executed Packets for AKF and SKIB

PL	Plain Text;
K	Initial key;
M1	First part of M;
M2	Second part of M;
K1	The confidential seed for SKIB;
K2	The encryption key for M2;
SK	The encryption key for M1 obtained from SKIB;
m	Bit length of M1,K1;
n	Bit length of M2,K2;
CT	Cipher Text;
CT1	First Part of a CT;
CT2	Second part of a CT;
PBOX	Permutation box;
$PBOX^{-1}$	Inverse Permutation box

Table 1. Nomenclatures

algorithms to provide a better transmission rate. In our proposed work, a novel method is produced using a cypher chaining mode to resolve the security challenges of LPWAN. This paper proposes the SKIB module, which is introduced to generate the new session key for every message using a hash function. The suggested module makes it possible to obtain the secret as a session key using simple algorithms instead of complicated ones for devices with limited resources.

Conclusion

The suggested model was implemented using a basic hash function due to the acknowledged limitations of the already-known algorithms for resource constraint devices. The proposed work can handle the message lost in transmission, so it is independent of the previous message to identify the session key. The SKIB model ensures security and privacy and can reduce the interference of attackers. The main advantage of the SKIB model is that it uses fewer iterations and fewer statements to compute the session key. So, it reduces the time it takes to generate a new key for every session. By generating a new key, both parties are synchronized with that key directly, which means the same key will encrypt and decrypt. The outcome of the experimental analysis proved the efficiency of the proposed system.

Future Work

There is a limitation in our proposed work. We have not implemented any authentication modules for this. If any attackers might have gotten the initial key shared by both parties, then an attacker could directly access the remaining data. Therefore, we intended to implement a lightweight authentication procedure in less time.

References

[1]. **AlHammadi, A., AlZaabi, A., AlMarzooqi, B., AlNeyadi, S., AlHashmi, Z., & Shatnawi, M. (2019, March).** Survey of IoT-based smart home approaches. In 2019 *Advances in Science and Engineering Technology International Conferences (ASET)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICASET.2019.8714572

[2]. **Baranwal, T., & Pateriya, P. K. (2016, January).** Development of IoT based smart security and monitoring devices for agriculture. In 2016 6th *International Conference-Cloud System and Big Data Engineering (Confluence)* (pp. 597-602). IEEE. https://doi.org/10.1109/CONFLUENCE.2016.7508189

[3]. **Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015a).** SIMON and SPECK: Block ciphers for the internet of things. *Cryptology ePrint Archive,* 2015, 585.

[4]. **Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2015b, June).** The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference* (pp. 1-6). https://doi.org/10.1145/2744769.2747946

[5]. **Bellovin, S. M. (2011).** Frank Miller: Inventor of the one-time pad. *Cryptologia,* 35(3), 203-222. https://doi.org/10.1080/01611194.2011.583711

[6]. **Bidgoly, A. J., & Bidgoly, H. J. (2019).** A novel chaining encryption algorithm for LPWAN IoT network. *IEEE Sensors Journal,* 19(16), 7027-7034. https://doi.org/10.1109/JSEN.2019.2910850

[7]. **Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016).** Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios. *IEEE Wireless Communications,* 23(5), 60-67. https://doi.org/10.1109/MWC.2016.7721743

[8]. **Dinu, D., Corre, Y. L., Khovratovich, D., Perrin, L., Großschädl, J., & Biryukov, A. (2019).** Triathlon of lightweight block ciphers for the internet of things. *Journal of Cryptographic Engineering,* 9(3), 283-302. https://doi.org/10.1007/s13389-018-0193-x

[9]. **Epstein, P. (1996).** *Key Distribution System,* United States.

[10]. **Fourtet, C., & Ponsard, B. (2020).** An introduction to Sigfox radio system. In *LPWAN Technologies for IoT and M2M Applications* (pp. 103-118). Academic Press. https://doi.org/10.1016/B978-0-12-818880-4.00005-3

[11]. **Gomez, C., Veras, J. C., Vidal, R., Casals, L., & Paradells, J. (2019).** A sigfox energy consumption model. *Sensors,* 19(3), 681. https://doi.org/10.3390/s19030681

[12]. **Goyal, T. K., & Sahula, V. (2016, September).** Lightweight security algorithm for low power IoT devices.

In 2016 *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1725-1729). IEEE. https://doi.org/10.1109/ICACCI.2016.7732296

[13]. Hanounik, B. (2006). *Cipher Block Chaining Mode in Encryption/Decryption Processing.* United States.

[14]. Hindumathi, G. V., & Bhaskari, D. L. (2019). Message based key distribution technique for establishing a secure communication channel in IoT networks. *International Journal of Computer Network & Information Security,* 11(11), 28-35. https://doi.org/10.5815/ijcnis.2019.11.04

[15]. Hwang, T., & Gope, P. (2015). IAR-CTR and IAR-CFB: integrity aware real-time based counter and cipher feedback modes. *Security and Communication Networks,* 8(18), 3939-3952. https://doi.org/10.1002/sec.1312

[16]. Kraijak, S., & Tuwanut, P. (2015, September). A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends. In *11th International Conference on Wireless Communications, Networking and Mobile Computing (Wicom 2015)* (pp. 1-6). IET. https://doi.org/10.1049/cp.2015.0714

[17]. Li, Y., & Cao, Y. (2016). Performance evaluation and analysis of lightweight symmetric encryption algorithms for internet of things. *International Journal of Reasoning-based Intelligent Systems,* 8(1-2), 84-90. https://doi.org/10.1504/IJRIS.2016.080072

[18]. Lipmaa, H., Rogaway, P., & Wagner, D. (2000, October). CTR-mode encryption. In *First NIST Workshop on Modes of Operation* (Vol. 39).

[19]. Luhach, A. K., & Luhach, A. K. (2016). Analysis of lightweight cryptographic solutions for internet of things. *Indian Journal of Science and Technology,* 9(28), 1-7. https://doi.org/10.17485/ijst/2016/v9i28/98382

[20]. Marques, G., Garcia, N., & Pombo, N. (2017). A survey on IoT: architectures, elements, applications, QoS, platforms and security concepts. In *Advances in Mobile Cloud Computing and Big Data in the 5G Era* (pp. 115-130). Springer, Cham. https://doi.org/10.1007/978-3-319-45145-9_5

[21]. Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express,* 5(1), 1-7. https://doi.org/10.1016/j.icte.2017.12.005

[22]. Patel, D., & Won, M. (2017, June). Experimental study on low power wide area networks (LPWAN) for mobile internet of things. In 2017 *IEEE 85th Vehicular Technology Conference (VTC Spring)* (pp. 1-5). IEEE. https://doi.org/10.1109/VTCSpring.2017.8108501

[23]. Pawar, A. B., & Ghumbre, S. (2016, December). A survey on IoT applications, security challenges and counter measures. In 2016 *International Conference on Computing, Analytics and Security Trends (CAST)* (pp. 294-299). IEEE. https://doi.org/10.1109/CAST.2016.7914983

[24]. Perlman, R., Kaufman, C., & Speciner, M. (2016). *Network Security: Private Communication in a Public World.* Pearson Education India.

[25]. Raza, U., Kulkarni, P., & Sooriyabandara, M. (2017). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials,* 19(2), 855-873. https://doi.org/10.1109/COMST.2017.2652320

[26]. Rescorla, E., & Modadugu, N. (2012). *Datagram Transport Layer Security Version 1.2.* Retrieved from https://datatracker.ietf.org/doc/rfc6347/

[27]. Sehrawat, D., & Gill, N. S. (2018). Lightweight block ciphers for IoT based applications: a review. *International Journal of Applied Engineering Research,* 13(5), 2258-2270.

[28]. Shah, S. H., & Yaqoob, I. (2016). A survey: Internet of things (IoT) technologies, applications and challenges. 2016 *IEEE Smart Energy Grid Engineering (SEGE),* 381-385. https://doi.org/10.1109/SEGE.2016.7589556

[29]. Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing,* 1-18. https://doi.org/10.1007/s12652-017-0494-4

[30]. Tehranchi, B. (2007). *Encryption Apparatus and Method for Synchronizing Multiple Encryption Keys with a Data Stream.* Patent and Trademark Office, Washington, United States.

[31]. Wilson, A. L. (1993). *Encryption Synchronization Combined with Encryption Key Identification.* Patent and Trademark Office, Washington, United States.

[32]. Yao, X., Chen, Z., & Tian, Y. (2015). A lightweight attribute-based encryption scheme for the internet of things. *Future Generation Computer Systems,* 49, 104-112. https://doi.org/10.1016/j.future.2014.10.010

ABOUT THE AUTHORS

G.V. Hindumathi is currently pursuing Ph.D. in Jawaharlal Nehru Technological University, Kakinada, India. She is working as an Assistant Professor at Gayatri Vidya Parishad College of Engineering (Autonomous), Visakhapatnam, Andhra Pradesh, India. She is specialized in Internet of Things and Network Security. Her research topic is on Security issues on Internet of Things.

Dr. D. Lalitha Bhaskari is currently working as a Professor in Andhra University, Visakhapatnam, Andhra Pradesh, India. Her areas of expertise includes Deep Learning, Network Security, and Image Processing. She has received Young scientist award from by IEI.