Research of Lightweight Encryption Algorithm Based on AES and Chaotic Sequences for Narrow-Band Internet of Things

Lianmin Shi[1,2](✉), Yihuai Wang[2], Rongyuan Jia[2], Tao Peng[2],
Jianwu Jiang[2], and Shilang Zhu[2]

[1] College of Information Technology, Suzhou Institute of Trade and Commerce,
Suzhou, China
18915418296@163.com

[2] School of Computer Science and Technology, Soochow University,
Suzhou, China

Abstract. Narrow-Band Internet of Things (NB-IoT), as a new LPWAN technology, has been applied in many fields, such as smart meter, smart parking, and so on. However, the security issues have become an important factor restricting rapid development of NB-IoT. A lightweight encryption algorithm based on AES and chaotic sequences (LCHAOSAES) is proposed to solve the data security in NB-IoT applications. Firstly, by reducing the number of AES (Advanced Encryption Standard) rounds and combining the three steps of 'SubBytes', 'ShiftRows' and 'MixColumns', time efficiency of LCHAOSAES is improved. Secondly, in order to make LCHAOSAES more secure, Logistic and Tent chaotic systems are adopted to generate dynamic keys for encryption. Finally, the remaining plaintext is encrypted by the keys, which are generated by Logistic and Tent chaotic systems, so that the length of the plaintext is equal to ciphertext. Theoretical analysis and field experiments performed in NB-IoT scenarios highlight significant performance when compared with, for instance, AES_128, LAES.

Keywords: NB-IoT · LCHAOSAES · AES · Chaotic sequences

1 Introduction

NB-IoT is a LPWAN technology based on cellular networks, which is proposed by the 3GPP in 2015, the series of standards was designated by the 3GPP in 2016 [1, 2]. Wide coverage, high capacity, flexible deployment, low power consumption and low cost are the most significant features of NB-IoT [3]. Nowadays, the ecological industry chain of NB-IoT has gradually matured. However, with the rapid development of NB-IoT applications, there will inevitably be many network security problems, such as, terminal nodes are counterfeited, the transmitted data is eavesdropped, and so on [4]. Therefore, according to the characteristics of NB-IoT, the research of security mechanisms can provide security for NB-IoT applications and promote the development of NB-IoT industry.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
X. B. Zhai et al. (Eds.): MLICOM 2019, LNICST 294, pp. 267–280, 2019.
https://doi.org/10.1007/978-3-030-32388-2_23

Encryption technology is an effective way to protect network security. The storage, computing power and energy of NB-IoT nodes are usually limited. Therefore, the asymmetric encryption algorithms with large resources consumption are not suitable for NB-IoT applications, as a result, symmetric encryption algorithms with less computation are more suitable for resource-constrained NB-IoT nodes [5]. Highly simplified AES and other block ciphers have become the major research subjects in the field of NB-IoT security [6–9]. In [10], authors have reduced the rounds of AES and used the method of looking up tables to simplify the round function, accordingly, the time efficiency is improved. In [11], the efficiency is promoted by improving round function. Although the methods above have improved efficiency, they neglect the reduced security. In addition, NB-IoT is sensitive to the length of data transmitted, because the longer data not only consumes more energy, but also increases communication cost. AES is the block cipher, the length of ciphertext may be longer than that of plaintext. If AES is to be applied to NB-IoT, it needs to be optimized.

In summary, in order to solve the problem of data security in NB-IoT, according to the technical characteristics of NB-IoT, the advantages and disadvantages of AES in efficiency and security are analyzed, and a lightweight encryption algorithm called LCHAOSAES is proposed. The main ideas of LCHAOSAES are summarized as follows:

(1) Reduce the number of AES rounds to improve efficiency.
(2) Exchange the order of 'SubBytes' and 'ShiftRows', combine 'SubBytes', 'ShiftRows' and 'MixColumns', and design look-up tables.
(3) Provide different initial key for each round to form 'one-time pad' encryption system. These initial keys are chaotic sequences generated by Logistic and Tent chaotic systems.
(4) Encrypt the remaining bytes of the plaintext byte-by-byte to ensure that the length of plaintext is consistent with the length of ciphertext.

The remainder of this article is organized as follows. In Section 2 we review some simplified encryption algorithms based on AES. In Sect. 3 we briefly introduce the basic theory of AES and chaotic encryption algorithm. In Sect. 4 the LCHAOSAES is described in details. In Sect. 5 the encryption model of NB-IoT application is given. Security of the LCHAOSAES and data statistics are analyzed in Sect. 6. Finally, our work is been concluded in Sect. 7.

2 Related Work

Encryption technology is a kind of data security protection way commonly used in wireless networks. Classic encryption algorithms include: DES, AES, RC5, SM1, ECC, SM2, RSA, and so on. In [12], authors take DES, AES, SM1, SM2 and RSA as the research objectives, and compare them in three aspects: energy consumption, time efficiency and spatial efficiency. Experimental results show that the symmetric encryption algorithms such as DES, AES and SM1 are more suitable for wireless communication nodes. Comparing with DES and SM1, although AES is slightly larger than DES and SM1 in storage and time, its energy consumption is least. Therefore, AES is a suitable encryption algorithm for wireless communication nodes. Formerly, AES was mostly implemented by hardware, but this way would increase the burden on the hardware of nodes. In [13–16], Osvik and other researchers have used high-level programming language to quickly achieve AES, and adopted time efficiency and spatial efficiency to measure the pros and cons of AES.

There are many studies on simplified AES. For instance, in [17], Zhao et al. have combined the three steps of 'SubBytes', 'ShiftRows' and 'MixColumns' to replace the XOR operation. In [10], time efficiency of the algorithm has been improved by reducing the rounds. Although these methods promote the efficiency, they don't consider the security which have been reduced.

There are also many studies on enhancing the security of AES. In [18], binary number of the chaotic sequence has been used to replace the initial key of AES, but efficiency is not sufficient. In [19], Chen et al. have proposed a key generation based on two-dimensional logistic mapping to enhance the independence of sub-keys, but it lacks of considerations on efficiency. In [20], Yan et al. have proposed a complex chaotic sequence based on the dynamic key, and they use a double chaotic system to dynamically initialize the key for AES, but this method has made the efficiency of AES decrease a lot.

The improved algorithms based on standard AES could not meet actual requirement for data encryption of NB-IoT in efficiency and safety. As a result, a lightweight encryption algorithm (LCHAOSAES) based on AES and chaotic sequences for NB-IoT is proposed.

3 AES and Chaotic Encryption

3.1 AES

AES is a packet encryption algorithm based on symmetric keys, which is resistant to all known types of attacks other than violent attacks. Furthermore, AES is a symmetric-key block cipher that operates under different combinations of block and key sizes, e.g., 128, 192 or 256 bits [21]. Due to resource-constraint of NB-IoT nodes, we improve the 128-bit AES algorithm. The encryption process consists of four basic transformations: 'SubBytes', 'ShiftRows', 'MixColumns' and 'AddRoundkey'. Figure 1 outlines the steps of AES_128 encryption.

(1) 'SubBytes' is a nonlinear function with the role of confusion bytes, S-box can be used to improve operation speed.
(2) 'ShiftRows' is a linear function that shifts the lines.
(3) 'MixColumns' is a function that converts each byte into a linear combination of input bytes.
(4) 'AddRoundkey' is a simple XOR operation, and its sub-keys are calculated from the master key.

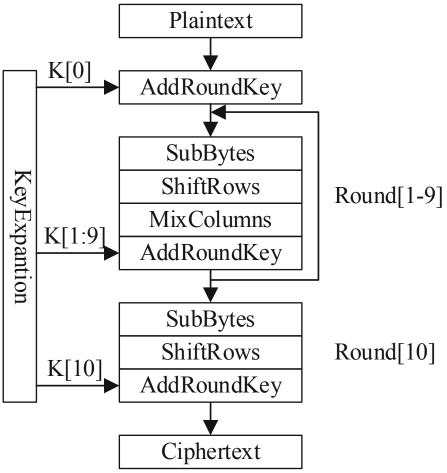

Fig. 1. Steps of AES_128 encryption

3.2 Chaotic Encryption

Chaotic encryption has been widely used in the field of information encryption with high computational speed, simple and real-time. In addition, chaos has many characteristics, such as nonperiodic, random, extremely sensitive to the initial state, easy to fork, and so on. Its model can be expressed as a nonlinear equation. The chaotic systems used in this paper are: Logistic and Tent mappings.

The one-dimensional Logistic equation is described as follows [22].

$$x_{n+1} = \mu x_n(1 - x_n) \tag{1}$$

where, x is a chaotic variable and $\mu \in (0,4]$ is a parameter that controls the chaotic equation. When $3.57 < \mu \leq 4$, the system exhibits a chaotic state and randomness of the sequence is better. When $\mu = 4$, the Logistic map has a great ductility and is very sensitive to initial value of the iteration, which is called the full-mapped chaotic state. Logistic chaotic sequences are similar to noise, and have high dependency on initial values and parameters. The same chaotic system will produce completely different chaotic sequence values even if it is disturbed by a small initial value or the number of iterations.

Tent mapping is a piecewise linear mapping system, the typical dynamic equation is described as follows.

$$y_{n+1} = \begin{cases} \lambda y_n, & 0 < y_n < 0.5 \\ \lambda(1 - y_n). & 0.5 < y_n < 1 \end{cases} \tag{2}$$

where, y is a chaotic variable and $\lambda \in (1.4, 2)$ is a parameter that controls the chaotic equation. Keep λ constant, any initial value $y0 \in (0, 1)$ can be iterated to get a definite numerical sequence. When $\lambda \in (1.4, 2)$, the system is in a chaotic state, and the value of λ is closer to 2, randomness of the chaotic sequence is better.

4 LCHAOSAES

In order to reduce the complexity of AES and improve the key security, we proposed LCHAOSAES, a lightweight encryption algorithm for NB-IoT applications.

4.1 Simplify AES

1. Reduce Number of Encrypted Rounds

In [10], Yao et al. have carried out an experiment of expanding key rounds with only 1-bit difference between each group of key seeds. Finally, the change rate in bits of each round key is obtained. The results show that the bit rate of change after 7 rounds is similar to that of 10. Consequently, setting the number of rounds to 7 is enough to resist known attacks. Based on the experimental result in [10], we take 7 as the number of encrypted rounds in LCHAOSAES.

2. Simplify Round Function

'MixColumns' involves multiplication on the finite field GF(28), computation of this magnitude are unbearable to NB-IoT nodes. Therefore, The LCHAOSAES exchanges 'SubBytes' with 'ShiftRows', and then replaces 'SubBytes', 'ShiftRows' and 'MixColumns' with method of looking up tables to improve efficiency.

During the first 6 rounds of LCHAOSAES, 'ShiftRows' does not actually operate on the input matrix, and only performs the row transform in the last round. R is the output of 'ShiftRows'. 'SubBytes' is represented by S(R), and the output is represented by B. 'MixColumns' takes B as input and C is its output.

'SubBytes' is described as follows.

$$B_{ij} = S(R_{ij}),\ (0 \le i,j \le 3) \tag{3}$$

'MixColumns' is described as follows.

$$\begin{aligned}\begin{bmatrix} C_{0j} \\ C_{1j} \\ C_{2j} \\ C_{3j} \end{bmatrix} &= \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} B_{0j} \\ B_{1j} \\ B_{2j} \\ B_{3j} \end{bmatrix} \\ &= \begin{bmatrix} 02 \times B_{0j} \\ 01 \times B_{0j} \\ 01 \times B_{0j} \\ 03 \times B_{0j} \end{bmatrix} \oplus \begin{bmatrix} 03 \times B_{1j} \\ 02 \times B_{1j} \\ 01 \times B_{1j} \\ 01 \times B_{1j} \end{bmatrix} \oplus \begin{bmatrix} 01 \times B_{2j} \\ 03 \times B_{2j} \\ 02 \times B_{2j} \\ 01 \times B_{2j} \end{bmatrix} \oplus \begin{bmatrix} 01 \times B_{3j} \\ 01 \times B_{3j} \\ 03 \times B_{3j} \\ 02 \times B_{3j} \end{bmatrix},\ (0 \le j \le 3).\end{aligned} \tag{4}$$

According to (3) and (4), we get the following Eq. (5).

$$\begin{bmatrix} C_{0j} \\ C_{1j} \\ C_{2j} \\ C_{3j} \end{bmatrix} = \begin{bmatrix} 02 \times S(R_{0j}) \\ 01 \times S(R_{0j}) \\ 01 \times S(R_{0j}) \\ 03 \times S(R_{0j}) \end{bmatrix} \oplus \begin{bmatrix} 03 \times S(R_{1j}) \\ 02 \times S(R_{1j}) \\ 01 \times S(R_{1j}) \\ 01 \times S(R_{1j}) \end{bmatrix} \oplus \begin{bmatrix} 01 \times S(R_{2j}) \\ 03 \times S(R_{2j}) \\ 02 \times S(R_{2j}) \\ 01 \times S(R_{2j}) \end{bmatrix} \oplus \begin{bmatrix} 01 \times S(R_{3j}) \\ 01 \times S(R_{3j}) \\ 03 \times S(R_{3j}) \\ 02 \times S(R_{3j}) \end{bmatrix}, \ (0 \le j \le 3). \tag{5}$$

In conclusion, it is concluded that the ‘ShiftRows’, ‘SubBytes’ and ‘MixColumns’ of the 1–6 round can be replaced by ‘SubBytes’, if the three tables are stored in advance, such as S, 2*S and 3*S (* is represented as multiplication on GF(28)). Figure 2 outlines the various steps of encryption of LCHAOSAES.

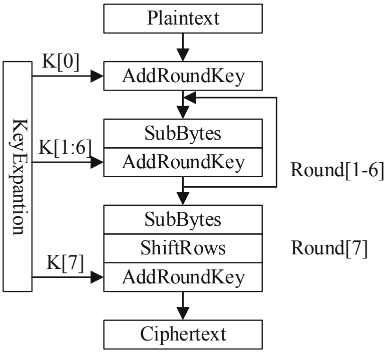

Fig. 2. Steps of LCHAOSAES encryption

4.2 Generate Chaotic Keys

One-dimensional chaotic system has some good characteristics, for instance, good initial sensitivity, pseudo-randomness and aperiodic. We use Logistic and Tent to construct a chaotic key generation system to generate chaotic sequences with high randomness and utilize these sequences to generate different initial keys for each AES plaintext packet. In this way, all plaintext blocks can be encrypted with different keys, which constitutes a ‘one-time pad’ encryption system.

LCHAOSAES requires that encryption and decryption use same parameters to create same chaotic system. And it generates the dynamic key through multiple iterations and encrypts plaintext block with the key.

The process of LCHAOSAES is described as follows.

1. Create chaotic systems with initialization parameters
   The key is passed through symmetric key encryption system. The key contains the control parameters μ and λ, the initial value x0 and y0, and the basic iteration

number N of Logistic and Tent maps to make encryption and decryption produce the same chaotic key generation system.

2. Block plaintext
   The plaintext is divided by 16 bytes, and the parameters m and n are calculated according to the length (length) of plaintext. m is described as the number of blocks, m = length/16. n represents the number of remaining bytes, n = length%16.
3. Generate chaotic sequences and compose the initial key
   Logistic N + m iterations and Tent N + n iterations to make the sequences sufficiently discrete, and correlate the number of iterations with the length of plaintext. A data block is encrypted at a time, Logistic and Tent are iterated 8 times to obtain 8 numbers. Then the first 4 digits are made up of the fractional part of each number and the integer is numbered on 256. The two sets of data are combined to form an initial key of 16 bytes.
4. Encrypt dynamically and handle the end of plaintext
   Encrypt the plaintext block by using the key generated in previous steps to obtain the ciphertext block. The remaining bytes of plaintext utilize XOR key to obtain ciphertext. This method of tail processing can greatly improve efficiency.

Finally, the m ciphertext blocks and n bytes of last ciphertext are combined to obtain the ciphertext, the length of ciphertext is equal to that of plaintext. Figures 3 and 4 depict the process of LCHAOSAES and a structure that uses a dynamic key for encryption, respectively.

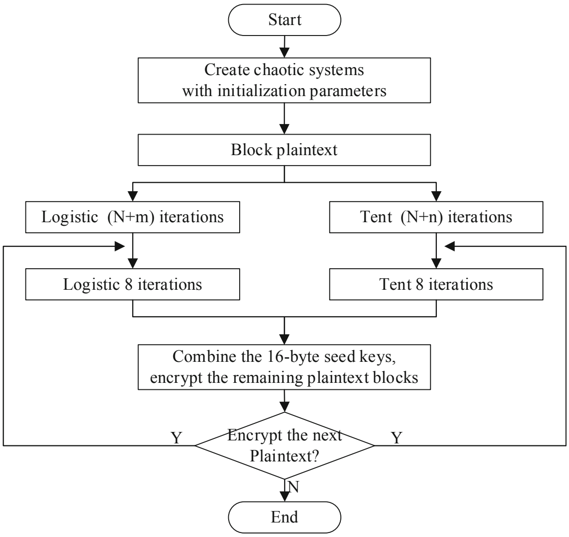
Start
Create chaotic systems with initialization parameters
Block plaintext
Logistic (N+m) iterations
Tent (N+n) iterations
Logistic 8 iterations
Tent 8 iterations
Combine the 16-byte seed keys, encrypt the remaining plaintext blocks
Y
Encrypt the next Plaintext?
Y
N
End

Fig. 3. The process of LCHAOSAES

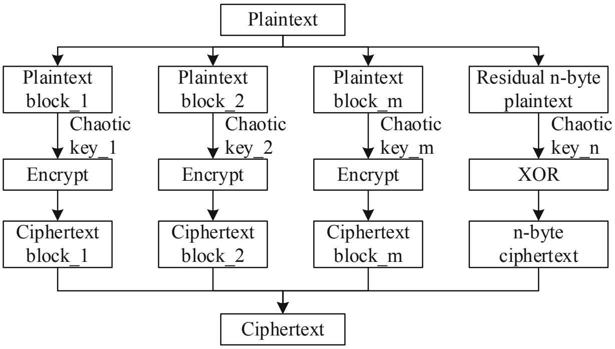

Fig. 4. Encryption structure using a dynamic key

5 The Encryption Model of NB-IoT Applications

5.1 NB-IoT Application Architecture

From perspective of technical science, NB-IoT application architecture can be abstracted as three components: UE (Ultimate-Equipment), MPO (Information Post Office) and HCI (Human-Computer Interaction System), as shown in Fig. 5. UE is an entity with hardware and software, which uses MCU as the core, with NB-IoT communication, data acquisition, control, computing and other functions, for instance, smart meters, etc. MPO is a data transmission system based on NB-IoT protocol, which is made up of operators' eNodeB, IoT controller and IoT Service Platform. HCI is able to achieve the specific application functions of the hardware and software systems, so that users can adopt laptops, mobile phones and other smart devices to achieve some intelligent applications, such as smart parking, smart logistics, and so on. In Fig. 5, the processes (1) and (2) represent the uplink data transmission, and processes (3) and (4) represent the downlink data transmission.

As can be seen from Fig. 5, Developers only need to design hardware and software of UE and HCI, and the problem about the data is transmitted within MPO will be resolved by the operators, such as China Mobile, Verizon and other operators. For the NB-IoT developers, they just need consider the development of hardware and software about UE and HCI.

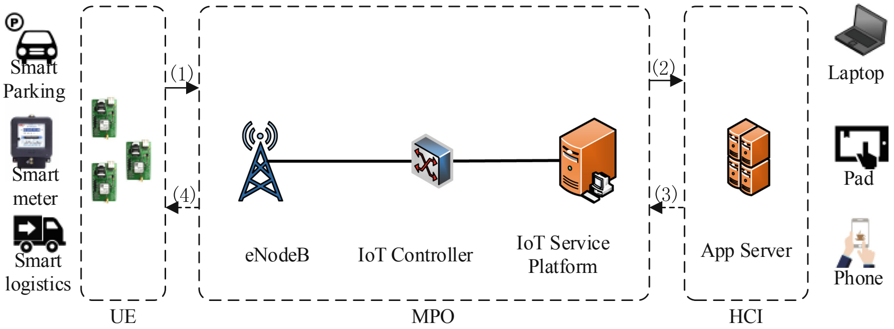

Fig. 5. NB-IoT application architecture

5.2 Encryption Model

In Fig. 5, data security may occur in any network communication process, such as processes (1), (2) (3), (4) and MPO. Each UE has a unique code IMEI (International Mobile Equipment Identity), it consists of 15-bit 'digital string'. We use IMEI to generate a fixed basic iteration number for each UE encryption key to reduce the difficulty of key management.

According to NB-IoT application architecture and IMEI, we propose the encryption model as shown in Fig. 6. When UE wants to send data to HCI, firstly, the last three digits of the IMEI are counted as number, and the number of basic LCHAOSAES iterations is set according to the number. If number < 100, then N = 100. Otherwise, N = number. After the key information is set, UE and HCI share the same key. UE encrypts the data using LCHAOSAES. The ciphertext is sent to HIC via MPO. Finally, HIC receives the ciphertext, decrypts it with the key, and gets the plaintext.

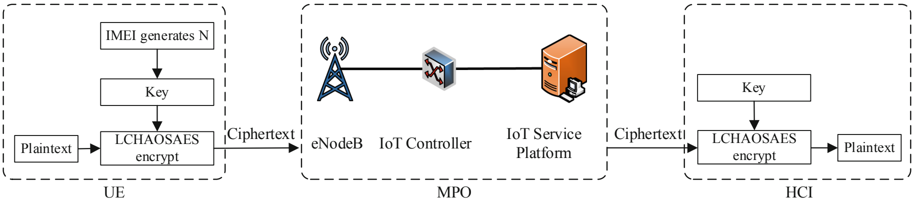

Fig. 6. Encryption model in NB-IoT applications

6 Experimental Evaluation

6.1 Experimental Environment

In this part, we will verify the security of LCHAOSAES by analysing key space and sensitivity, and compare the security with LAES and AES_128. Secondly, we will count the three algorithms used in encryption, and compare the results. The less time it takes, the higher efficiency it has.

In this paper, there are two experimental environments. One is VS2012 platform with VC ++ compiler, and 64-bit PC (clocked at 3.20 GHz) with operating system is Windows10. In this environment we will verify sensitivity of the key and efficiency of the algorithm. The other is the UE with NB-IoT encryption model. The MCU of UE is MKL36Z64VLH4, whose clock frequency is 48 MHz and Flash is 64 KB. The communication module used by the UE is Quectel BC95, which provides a series of AT commands. UE can communicate with MPO through these commands, or obtain IMEI of the module. We will test the efficiency of LCHAOSAES, AES_128, and LAES on the NB-IoT encryption model.

6.2 Security Analysis

1. Key space
The size of key space affects the ability of encryption algorithm about resisting key search attacks. The LCHAOSAES key is controlled by five parameters, including the rounds of Logistic chaotic system and the initial value, the parameter and the initial value of Tent chaotic system, and the basic iteration number. Double type of data can be obtained after the decimal point of 15 significant figures. The size of each parameter space is shown in Table 1.

Table 1. The Size of parameter space

Parameter	Type	Range	Size
μ	Double	[3.57,4.0]	$K_{\mu} \approx 0.43 \times 10^{15}$
x_0	Double	(0,1)	$K_{x0} \approx 1 \times 10^{15}$
λ	Double	(1.4,2.0)	$K_{\lambda} \approx 0.6 \times 10^{15}$
y_0	Double	(0,1)	$K_{y0} \approx 1 \times 10^{15}$
N	Int	[100,1000)	$K_N \approx 900$

The size of key space of LCHAOSAES is:

$$K = K_{\mu} \times K_{x0} \times K_{\lambda} \times K_{y0} \times K_N \approx 2.32 \times 10^{62} \tag{6}$$

In theory, key space sizes of AES_128 and LAES are 2128 ≈ 3.40 × 1038. Therefore, the size of key space of LCHAOSAES is much larger than that of AES_128 and LAES. The number of iterations of LCHAOSAES depends on the IMEI of UE and the length of plaintext, so its key space is uncertain, which further enhances the difficulty of using exhaustive method to crack the ciphertext information.
2. Sensitivity verification
Sensitivity is a common index to measure the security of encryption algorithm [23], that is, the key satisfies the avalanche criterion. When the key changes slightly, half

of ciphertext bit changes. In order to verify sensitivity of AES_128, LAES, and LCHAOSAES, we conduct an experiment. In this experiment, we calculate the change rate of ciphertext obtained by encrypting the same plaintext with different algorithms, and the length of plaintext is 1000 bytes. The average results obtained in 20 experiments are shown in Fig. 7.

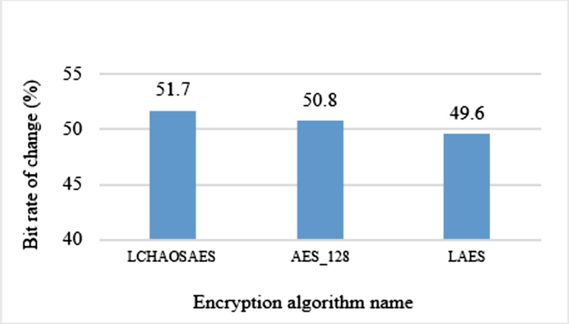

Fig. 7. The bit rate of change after encryption.

As can be seen from Fig. 7, the change bits rate of LCHAOSAES is the highest. This shows that LCHAOSAES is better in avalanche experiments. In the experiment, when the key changes slightly, the LCHAOSAES key changes to a small extent, resulting in a 51.7% change in the bit rate of ciphertext, and the avalanche of key is obvious. This proves that the sensitivity of LCHAOSAES key is superior to that of AES_128 and LAES, which means the higher security.

6.3 Algorithm Efficiency

The efficiency of algorithm is to evaluate the efficiency on each platform. The less time it takes for the algorithm, the higher efficiency it has. We use the C language to program AES_128, LAES proposed in [10] and LCHAOSAES. The operating system of 64-bit PC (clocked at 3.20 GHz) is Windows10 and the compiler is VS2012 platform VC ++. In the above experimental environment, 1000 bytes of text data were encrypted 50 times, and the average time consumed by each algorithm is shown in Table 2.

Table 2. Average time spent by each algorithm

Algorithm name	Average time (ms)
AES_128	2.271
LAES	1.245
LCHAOSAES	1.476

As can be seen from Table 2, AES_128 spent the most time in encryption, and LAES spend the least. Because LCHAOSAES improves efficiency while improving security, it is unavoidable that efficiency of LCHAOSAES is slightly lower than LAES, but it takes 35% less time than AES_128.

We transplanted the above three algorithms to our proposed encryption model. clock frequency of UE's MCU (MKL36Z64VLH4) is 48 MHz, flash size is 64 KB. The average time consumed by each algorithm to encrypt the data of different lengths is shown in Fig. 8.

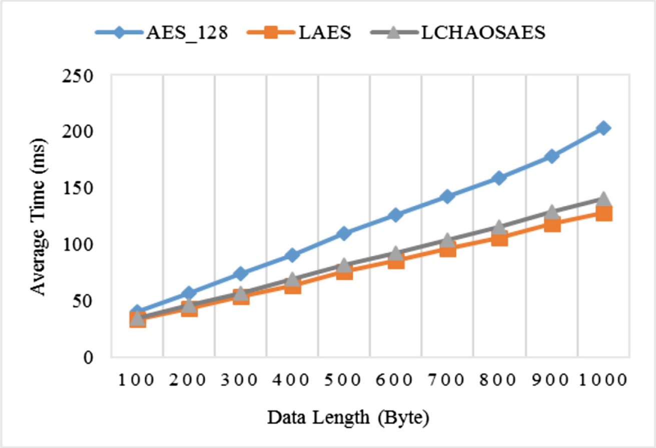

Fig. 8. Average time spent on data encryption for different lengths

Figure 8 shows that LCHAOSAES spent obviously less time than AES_128, slightly more than LAES. When the length of plaintext increases, AES_128 has a maximum growth rate, while time growth rate of LCHAOSAES and LAES are slightly smaller. Although LCHAOSAES is slightly less efficient than LAES, it is known from Table 2 that it is 35% more efficient than AES_128.

7 Conclusion

In order to solve the data security problem in NB-IoT applications, we proposed LCHAOSAES, a lightweight encryption algorithm based on AES and chaotic sequence. Furthermore, the encryption model is designed according to the NB-IoT application architecture. LCHAOSAES improves efficiency by reducing AES rounds and designing quick tables. In addition, it uses Logistic and Tent chaotic systems to generate complex chaotic sequences as encryption keys, and provides different keys for each round operation to improve the security and ensure that the length of ciphertext is the same as the plaintext. Through the theoretical analysis and experimental evaluation on the encryption model, we can see that the security of LCHAOSAES is obviously higher than that of AES_128 and LAES. Although the efficiency is slightly lower than

LAES, and 35% higher than AES_128. LCHAOSAES with high security and efficiency can be used as an encryption algorithm for NB-IoT applications.

In addition, the table designed to improve the efficiency of LCHAOSAES will take 1792 bytes of memory. Although the current MCU processor is enough to accommodate 1792 bytes, but we plan to solve this problem in the future work.

Acknowledgements. This work is supported in part by National Natural Science Foundation of China under Grant No. 61672370, 61672369 and 61872257, Natural Science Research Foundation of Jiangsu Higher Education Institutions under Grant No. 17KJB520037 and 17KJB520035, and CERNET Innovation Project under Grant No. NGII20170311.

References

1. 3GPP Revised Work Item: Narrowband IoT (NB-IoT), 3GPP RP-152284. 3GPP (2015)
2. Ericsson, Nokia, ZTE, NTT DOCOMO Inc: 3GPP RP-161248, Introduction of NB-IoT in 36.331. 3GPP TSG-RAN Meeting#72, Busan, South Korea, June 2016
3. 3GPP TR 23.720: Study on architecture enhancements for Cellular Internet of Things (Release 13). 3GPP (2016)
4. Ratasuk, R., Vejlgaard, B., Mangalvedhe, N., et al.: NB-IoT system for M2M communication. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 1–5. IEEE, Doha (2016)
5. Lee, J., Kapitanova, K., Son, S.H.: The price of security in wireless sensor networks. Comput. Netw. **54**(17), 2967–2978 (2010)
6. Ben Othman, S., Trad, A., Youssef, H.: Performance evaluation of encryption algorithm for wireless sensor networks. In: International Conference on Information Technology & E-services, pp. 1–8. IEEE, Sousse (2012)
7. Chen, Q., Chen, Q., Min, Y., et al.: Design of encryption algorithm of data security for Wireless Sensor Network. In: International Conference on Electrical & Control Engineering, pp. 2983–2986. IEEE, Yichang (2011)
8. Msolli, A., Helali, A., Maaref, H.: Image encryption with the AES algorithm in wireless sensor network. In: International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 41–45. IEEE, Monastir (2016)
9. Panda, M.: Data security in wireless sensor networks via AES algorithm. In: International Conference on Intelligent Systems & Control, pp. 1–5. IEEE, Coimbatore (2015)
10. Yao, Z., Ling, Y.E.: Encryption algorithms for wireless sensor networks based on AES. Comput. Eng. Des. **36**(3), 619–623 (2015)
11. Wei, G., Zhang, H.: The light-weight optimization of AES algorithm and its application in radio frequency identification tag. J. Wuhan Univ. **58**(6), 471–476 (2012)
12. Xi, Z., Li, L., Shi, G., Wang, S.: A comparative study of encryption algorithms in wireless sensor network. In: Zeng, Q.-A. (ed.) Wireless Communications, Networking and Applications. LNEE, vol. 348, pp. 1087–1097. Springer, New Delhi (2016). https://doi.org/10.1007/978-81-322-2580-5_99
13. Osvik, D.A., Bos, J.W., Stefan, D., et al.: Fast software AES encryption. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 75–93. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13858-4_5
14. Babu, T.R., Murthy, K.V.V.S., Sunil, G.: Implementation of AES algorithm on ARM. In: Proceedings of the ICWET 2011 International Conference & Workshop on Emerging Trends in Technology, pp. 1211–1213. ACM, Mumbai (2011)

15. Ahmed, W., Mahmood, H., Siddique, U.: The efficient implementation of S8 AES algorithm. Lect. Notes Eng. Comput. Sci. **2191**(1), 334–339 (2011)
16. Barnes, A., Fernando, R., Mettananda, K., et al.: Improving the throughput of the AES algorithm with multicore processors. In: International Conference on Industrial and Information Systems (ICIIS), pp. 1–6. IEEE, Chennai (2012)
17. Wong, M.M., Wong, M.L.D.: New lightweight AES S-box using LFSR. In: International Symposium on Intelligent Signal Processing & Communication Systems, pp. 115–120. IEEE, Kuching (2015)
18. Hong, C., Yi, C.: Research on AES algorithm based on CHAOS. J. Beijing Technol. Bus. Univ. **27**(5), 57–60 (2009)
19. Chen, D., Qing, D., Wang, D.: AES key expansion algorithm based on 2D logistic mapping. In: Fifth International Workshop on Chaos-fractals Theories & Applications, pp. 207–211. IEEE, Dalian (2012)
20. Wang, Y., Wang, J.: A new image encryption algorithm based on compound chaotic sequence. In: International Conference on Measurement, pp. 962–966. IEEE, Harbin (2012)
21. Muhaya, B.F.T.: Chaotic and AES cryptosystem for satellite imagery. Telecommun. Syst. **52** (2), 573–581 (2013)
22. He, C.G., Bao, S.D.: An encryption algorithm based on chaotic system for 3G security authentication. In: Youth Conference on Information, Computing and Telecommunications, pp. 351–354. IEEE, Beijing (2011)
23. Pradhan, C., Bisoi, A.K.: Chaotic variations of AES algorithm. Int. J. Chaos Control Model. Simul. **2**(2), 19–25 (2013)