RESEARCH ON PRIVACY DATA SECURITY PROTECTION IN SENSOR NETWORKS USING ENCRYPTION ALGORITHMS

Yunxia YAO[1], Renjie PENG[2], Wenli MI[3]

This paper briefly introduces wireless sensor networks (WSNs) and proposes an enhancement to a block encryption algorithm using chaotic mapping principles. The proposed encryption algorithm was tested on an established WSN platform in a simulation experiment, and a comparative analysis was conducted with two encryption algorithms, advanced encryption standard (AES) and data encryption standard (DES). The results indicated that, as the volume of encrypted data grew, the encryption time for all three algorithms exhibited an upward trend, the 0-1 balance degree of the ciphertext decreased, and the information entropy increased. Specifically, for a given size of encrypted data, the optimized encryption algorithm demonstrated the shortest encryption time, the smallest 0-1 balance degree in the ciphertext, and the highest information entropy, closely approaching the ideal value.

Keywords: encryption algorithm, sensor network, data security, chaotic mapping

1. Introduction

Networks comprised of sensors generate substantial data [1]. For instance, environmental sensor networks yield data on temperature, humidity, and light, while networks for smart homes and industrial automation produce dynamic information such as device status and personnel behavior [2]. However, this data often contains sensitive information, raising concerns about security, data leakage, and misuse. Privacy data in sensor networks typically includes information that can identify an individual or infer private details. To address this, encryption algorithms are applied to data transmission and storage [3]. Wen et al. [4] proposed a coupled chaotic system based on k-sine transform to protect medical data, which combines any two one-dimensional chaotic maps to generate a new chaotic map. Experimental results and security analysis demonstrated that the proposed encryption scheme for electroencephalogram signals performed well and had passed rigorous cryptographic security tests. Min et al. [5] proposed an algorithm that utilizes density for encrypting databases to address the issue that the

[1] School of Mathematics and Information Engineering, Longdong University, Qingyang, Gansu 745000, China, e-mail: yyunxiayx@hotmail.com

[2] School of Intelligent Manufacturing, Longdong University, Qingyang, Gansu 745000, China

[3] School of Mathematics and Information Engineering, Longdong University, Qingyang, Gansu 745000, China

performance of query processing algorithm may vary with the tree depth. To enhance query efficiency, they employed a hash index based on algebraic encoding. Through performance analysis, this approach offered superior query processing performance compared to existing solutions while also guaranteeing user privacy. Gai et al. [6] suggested a dynamic data encryption strategy to address privacy concerns. This strategy aimed to meet execution time requirements while ensuring data security through selective encryption. Afterwards, the protective effect of this strategy on privacy data was verified through experiments. This paper utilized chaotic mapping principles to enhance block encryption algorithm security, followed by testing the encryption algorithm in simulation experiments on a constructed wireless sensor network (WSN) platform.

2. Wireless sensor network

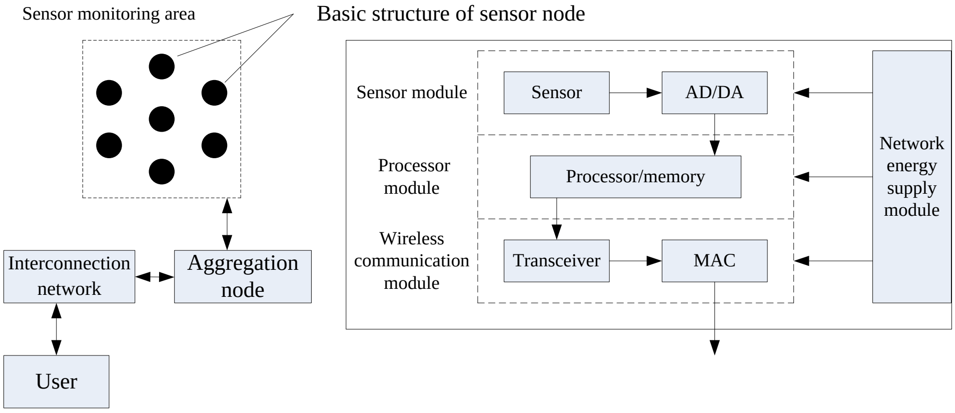


Fig. 1. Basic structure of wireless sensor network as well as sensor nodes

The fundamental structure of a WSN and its sensor nodes is illustrated in Fig. 1 [7]. The WSN comprises a sensing and monitoring area, aggregation node, interconnection network, and users. The sensing and monitoring area accommodates multiple sensor nodes responsible for collecting information. Initially, the collected data is centralized at the aggregation node before being transmitted to users through the interconnection network [8].

3. Encryption algorithms

As previously mentioned, WSNs leverage low-cost and quickly deployable sensors to cover extensive surveillance areas [9]. However, due to the limited energy and signal transmission ranges of individual sensors, they need to self-organize into a network within the designated area. Additional sensors serve as relay nodes for information transmission. Given that the transmitted information within WSNs often contains sensitive and private data and the wireless communication links between sensor nodes are exposed in the open space, there is an increased risk of eavesdropping or tampering between sensor nodes. Therefore, encryption algorithms are essential to safeguard the transmitted information [10]. However, considering that sensors have constrained energy resources, it is crucial for encryption algorithms not to be overly complex. The available algorithms for information encryption include symmetric encryption, where a common key is used for both encryption and decryption, and asymmetric encryption, where distinct keys are employed for encryption and decryption. In the context of asymmetric encryption, the public key is used for encryption purposes, whereas the private key is solely employed for decryption. Although asymmetric encryption algorithms are theoretically more secure due to the complexity of number-theoretic problems, they also involve increased arithmetic complexity [11].

The block encryption algorithm with a Feistel structure [12], a symmetric encryption algorithm, employs a general principle where the plaintext is divided into left and right parts. In each round of the encryption iteration process, the right part undergoes encryption using the key, followed by addition with the left part. The outcome of this process is exchanged with the right part. The procedure is repeated until the predetermined number of rounds is reached.

The round function in the block encryption algorithm is the key component, which is responsible for obfuscating and diffusing the plaintext. The higher its complexity, the better the security of the algorithm. It uses the same round function for encryption and decryption, so it can be designed to be more complex, but considering the limited computing resources of the sensors, it is better not to use multiplication and division operations. Chaotic mapping comes from chaos theory, which has the characteristics of ergodicity, overall stability, local randomness, and initial condition sensitivity, which is very similar to the needs of encryption algorithms. As shown in Fig. 2, $\oplus$ denotes the exclusive OR operation [13], and the specific steps are as follows.

① A 32-bit plaintext is entered and set $i=0$.

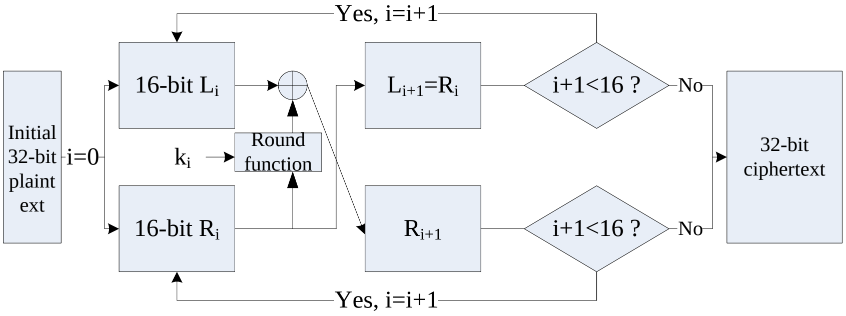
② The plaintext is split into left data L_i and right data R_i.

③ R_i is directly used as the left data (L_{i+1}) after one round of encryption, i.e., $L_{i+1}=R_i$.

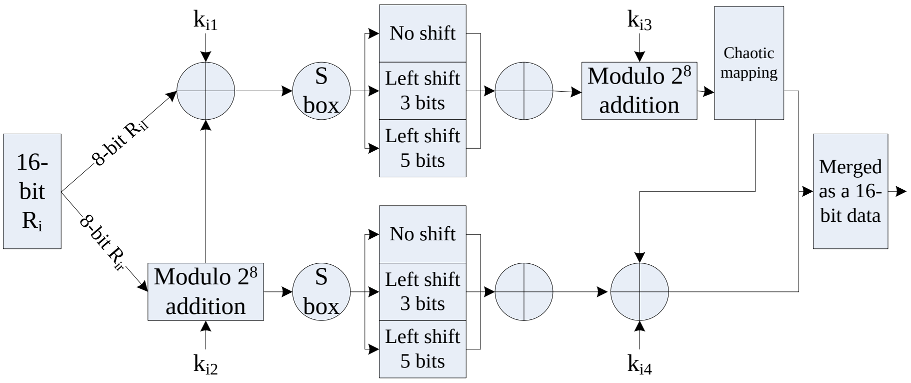
④ R_i is simultaneously encrypted using the round function and key k_i, followed by $\oplus$ computation. A new right data (R_{i+1}) is obtained.

⑤ Whether $i+1<16$ is established or not is determined. If it is, it means that the encryption round is not finished yet, and return to step ③ after denoting

$i = i + 1$; if not, it means that the encryption round is finished, and a 32-bit ciphertext is obtained by combining L_{i+1} and R_{i+1}.



(a) General flow of a block encryption algorithm combined with chaotic mapping



(b) The calculation flow of the round function combined with chaotic mapping

Fig. 2. Flow of a block encryption algorithm incorporating chaotic mapping

The exact process of encrypting using the wheel function is as follows.

① The 16-bit R_i is divided into 8-bit left data R_{il} and 8-bit right data R_{ir}.

② Modulo 2^8 addition is performed for R_{ir} and k_{i2}, and the result of the addition undergoes $\oplus$ along with R_{il} and k_{i1} in addition to the table lookup computation using the S-box, after which the S-box is also used [14].

③ After the S-box table lookup calculation, the left and right parts of the data are not shifted, left shifted by 3 bits, and left shifted by 5 bits, respectively.

④ The data of the left part without shifting, left shifting by 3 bits, and left shifting by 5 bits are subjected to the exclusive OR operation, and the right part is treated in the same way.

⑤ Then, the left part of the data is combined with k_{i3} for modulo 2^8 addition, and the operation result is processed by chaotic mapping [15]. Integerized Logistic chaotic mapping is used here, and the corresponding equation is:

$$x_{n+1} = 4x_n - \frac{2x_n^2}{7} - 1, \tag{1}$$

where x_n is the current input and x_{n+1} is the mapped output.

⑥ The right part of the data and the chaotically mapped left part are combined with k_{i4} for $\oplus$ computation, and the result is merged with the left part.

4. Simulation experiments

4.1 Experimental setup

The fundamental structure of the laboratory simulation platform is depicted in Fig. 3. The sensor cluster encompassed ten terminal devices, and the essential parameters of the terminal devices are detailed in Table 1. The coordinator was composed of a wireless sensing device and a router. The parameters of wireless sensing devices are consistent with those of terminal devices, or in other words, they are the same type of device that performs different functions. Wired connections link the wireless sensing device and the router.

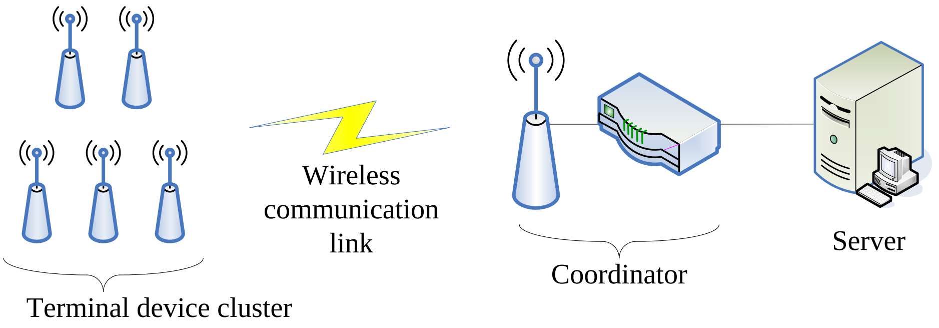


Fig. 3. Basic architecture of the WSN simulation platform

Table 1

Basic parameters of terminal devices

Element	Configuration/model	Element	Configuration/model
Microprogrammed control unit	CC2651R3	Flash memory	352 KB
Cache (RAM)	8 KB	Programmable pins	23

External sensors	TMP4718 temperature sensor	Wireless transmission protocol	ZigBee 3.0

The encryption algorithm proposed in this paper was implemented in both the terminal devices and the coordinator. Specifically, the encryption algorithm was applied in the terminal devices, and the corresponding decryption algorithm was deployed in the coordinator. The decrypted plaintext was then uploaded to the server, where the received temperature data was displayed. In the simulation experiment, in addition to the encryption algorithm used in this paper, advanced encryption standard (AES) and data encryption standard (DES) algorithms were also tested as a comparison.

4.2 Test programs

(1) Encryption speed test

The terminal device cluster temporarily stored the gathered ambient temperature data in flash memory. The data was encrypted and uploaded to the server once the storage size reached a specific threshold. To assess the encryption time under various data sizes, the storage sizes were configured as 20, 40, 60, 80, and 100 KB in the experiments.

(2) Key sensitivity test

In the plaintext sensitivity test, two plaintexts were provided, differing by only one character. The key for the encryption algorithm was set to "32568894," and the encryption result was recorded.

In the key sensitivity test of the algorithm, a given plaintext was used, and two keys, "32568894" and "32568899," were set. There was only one character difference between the keys. The encryption results of the algorithm were recorded after using the two keys.

(3) Ciphertext statistics test

The data stored in the flash memory of the terminal devices with the size of 20, 40, 60, 80, and 100 KB were encrypted, and then the "0-1" balance degree and information entropy of the ciphertext was tested. The calculation formulas are:

$$\begin{cases} \xi = \dfrac{|K_1 - K_2|}{n} \\ H(S) = -\sum_{S} P(S_i) \log_2 P(S_i) \end{cases}, \tag{2}$$

where K_1 and K_2 are the number of 0 and 1 after the ciphertext is converted to binary, respectively, n denotes the total number of 0 and 1 after the ciphertext is converted to binary, ξ denotes the degree of balance, $H(S)$ is the information entropy of the ciphertext (its ideal value is 8 when ASCII characters randomly distributed, i.e., the characters are distributed uniformly), and $P(S_i)$ is the probability that the i-th ASCII character is in the ciphertext.

4.3 Test results

As depicted in Fig. 4, the encryption time for all three algorithms increased with the growth of data. For a given file size, the AES algorithm consumed the most time for encryption, followed by the DES algorithm. In contrast, the block encryption algorithm combined with chaotic mapping required the least time for encryption.

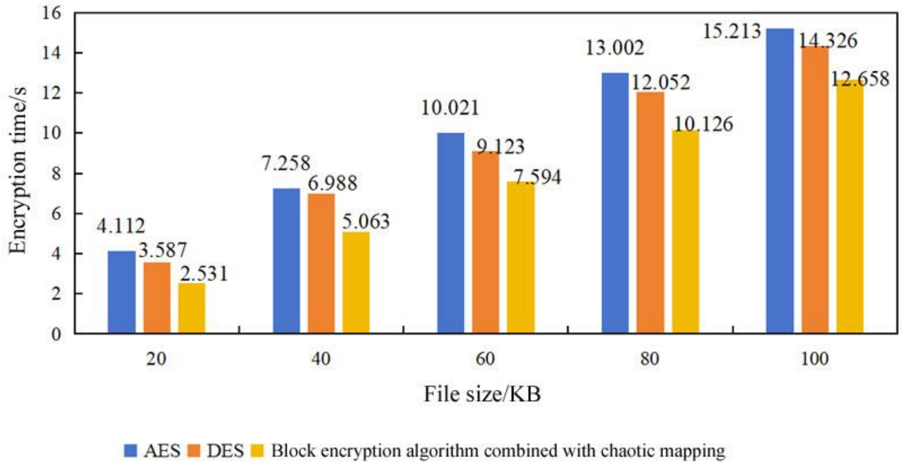


Fig. 4. Encryption time

Table 2

Sensitivity test results

Plaintext sensitivity testing	Plaintext	温度13℃	温度14℃
	Key	32568894	
	Encryption result	娞预嫫?K%拓)邈	2唇?趧?;+[劚
Key sensitivity test	Plaintext	温度13℃	
	Key	32568894	32568899
	Encryption result	娞预嫫?K%拓)邈	??篃?o?[?鉡.

The test results for plaintext sensitivity and key sensitivity of the block encryption algorithm combined with chaotic mapping are presented in Table 2. Table 2 indicates that under the same key, even if only one character changes in the plaintext, its final encryption result will be significantly altered. Similarly, with the same plaintext, even if just one character in the key is altered, the resulting ciphertext will exhibit a noticeable difference.

Table 3

Ciphertext statistics of the three encryption algorithms

Data size		20 KB	40 KB	60 KB	80 KB	100 KB
AES	Balance degree	0.02163	0.01257	0.00897	0.00523	0.00125
	Information entropy	7.789	7.798	7.823	7.859	7.897
DES	Balance degree	0.01211	0.00635	0.00415	0.00236	0.00067
	Information entropy	7.858	7.872	7.899	7.932	7.958
The block encryption algorithm combined with chaotic mapping	Balance degree	0.00289	0.00173	0.00078	0.00063	0.00037
	Information entropy	7.956	7.975	7.987	7.996	7.998

Three encryption algorithms were employed to encrypt data of varying sizes, and the resulting ciphertexts were analyzed to calculate the 0-1 balance degree and information entropy. The statistical findings are presented in Table 3. It is observed that as the volume of encrypted data increased, the 0-1 balance degree of the ciphertexts for all three encryption algorithms decreased, while the information entropy increased. Under equivalent data sizes, the ciphertext generated by the encryption algorithm proposed in this paper exhibited the smallest 0-1 balance degree and the largest information entropy, closely approximating the ideal value of 8. The ciphertext produced by the DES algorithm fell in the middle range for both 0-1 balance degree and information entropy, whereas the ciphertext from the AES algorithm displayed the highest 0-1 balance degree and the lowest information entropy. These results demonstrated that the ciphertext generated by the designed encryption algorithm exhibited favorable randomness.

5. Conclusion

This paper provides a brief overview of WSN and explores block encryption algorithms designed for such networks. To enhance the security of encryption algorithms, chaotic mapping was introduced. The proposed encryption algorithm was then tested in simulation experiments conducted on a constructed WSN platform. The key findings are summarized as follows. The encryption time for all three algorithms exhibited an upward trend with the growth of data. When the file size was fixed, the AES algorithm consumed the most time, followed by the DES algorithm, and the proposed encryption algorithm demonstrated the shortest encryption time. Both plaintext and key sensitivity tests revealed that even a single character change significantly altered the final ciphertext. As the volume of encrypted data increased, the 0-1 balance degree of ciphertexts for all three encryption algorithms decreased, while information entropy increased. For the same data size, the ciphertext generated by the proposed encryption algorithm exhibited the smallest 0-1 balance degree and the highest information entropy, approaching the ideal value of 8.

REFERENCES

[1]. *J. B. Lim, B. Jang and M. L. Sichitiu*, “MCAS-MAC: A multichannel asynchronous scheduled MAC protocol for wireless sensor networks”, Comput. Commun., **vol. 56**, no. feb.1, 2015, pp. 98-107.

[2]. *T. Zheng, M. Gidlund and J. Akerberg*, “WirArb: A New MAC Protocol for Time Critical Industrial Wireless Sensor Network Applications”, IEEE Sens. J., **vol. 16**, no. 7, 2015, pp. 2127-2139.

[3]. *P. K. Sahoo, S. R. Pattanaik and S. L. Wu*, “Design and Analysis of a Low Latency Deterministic Network MAC for Wireless Sensor Networks”, Sensors, **vol. 17**, no. 10, 2017, pp. 1-24.

[4]. *D. Wen, W. Jiao, X. Li, X. Wan, Y. Zhou, X. Dong, X. Lan and W. Han*, “The EEG signals encryption algorithm with K-sine-transform-based coupling chaotic system”, Inf. Sci., **vol. 622**, 2022, pp. 962-984.

[5]. *M. Yoon, M. Jang, Y. S. Shin and J. W. Chang*, “A Bitmap based Data Encryption Scheme in Cloud Computing”, Int. J. Softw. Eng. Its Appl., **vol. 9**, no. 5, 2015, pp. 345-360.

[6]. *K. Gai, M. Qiu, H. Zhao and J. Xiong*, “Privacy-Aware Adaptive Data Encryption Strategy of Big Data in Cloud Computing”, IEEE International Conference on Cyber Security & Cloud Computing, 2016, pp. 273-278.

[7]. *Y. M. Koukou, S. H. Othman and M. Nkiama*, “Comparative Study Of AES, Blowfish, CAST-128 And DES Encryption Algorithm”, IOSR J. Eng., **vol. 06**, no. 6, 2016, pp. 01-07.

[8]. *Z. Mihret and M. W. Ahmad*, “The Reverse Engineering of Reverse Encryption Algorithm and a Systematic Comparison to DES”, Proc. Comput. Sci., **vol. 85**, 2016, pp. 558-570.

[9]. *Z. L. Lan, L. Zhu, Y. C. Li and J. Liu*, “A Color Image Encryption Algorithm Based on Improved DES”, Appl. Mech. Mater., **vol. 743**, 2015, pp. 379-384.

[10]. *A. B. Nasution, S. Efendi and S. Suwilo*, “Image Steganography In Securing Sound File Using Arithmetic Coding Algorithm, Triple Data Encryption Standard (3DES) and Modified Least Significant Bit (MLSB)”, J. Phys. Conf., **vol. 1007**, 2018, pp. 1-6.

[11]. *Z. Hua, Y. Zhou, C. M. Pun and C. L. P. Chen*, “2D Sine Logistic modulation map for image encryption”, Inform. Sciences, **vol. 297**, no. C, 2015, pp. 80-94.

[12]. *Y. Liu, S. Tang, R. Liu, L. Zhang and Z. Ma*, “Secure and robust digital image watermarking scheme using logistic and RSA encryption”, Expert Syst. Appl., **vol. 97**, 2018, pp. 95-105.

[13]. *Y. Aono, T. Hayashi, L. T. Phong and L. Wang*, “Privacy-Preserving Logistic Regression with Distributed Data Sources via Homomorphic Encryption”, IEICE T. Inf. Syst., **vol. E99.D**, no. 8, 2016, pp. 2079-2089.

[14]. *A. Chadha, S. Mallik, A. R. Chadha, R. Johar and M. M. Roja*, “Dual-Layer Video Encryption using RSA Algorithm”, Int. J. Comput. Appl., **vol. 116**, no. 1, 2015, pp. 33-40.

[15]. *G. Iovane, A. Amorosia, E. Benedetto and G. Lamponi*, “An Information Fusion approach based on prime numbers coming from RSA algorithm and Fractals for secure coding”, J. Discrete Math. Sci. C., **vol. 18**, no. 5, 2015, pp. 455-479.