Proceedings of 2019 IEEE
International Conference on Mechatronics and Automation
August 4 - 7, Tianjin, China

Hybrid Encryption Algorithm Based on Wireless Sensor Networks

Tongxu Yue
School of Computer Science & Technology
Xi'an University of Posts & Telecommunications
Xi'an 710121,China
641389537@qq.com

Chuang Wang and Zhi-xiang Zhu
Institute of Internet of Things and IT-based Industrialization
Xi'an University of Posts & Telecommunications
Xi'an 710061,China

***Abstract* - Based on the analysis of existing wireless sensor networks(WSNs) security vulnerability, combining the characteristics of high encryption efficiency of the symmetric encryption algorithm and high encryption intensity of asymmetric encryption algorithm, a hybrid encryption algorithm based on wireless sensor networks is proposed. Firstly, by grouping plaintext messages, this algorithm uses advanced encryption standard (AES) of symmetric encryption algorithm and elliptic curve encryption (ECC) of asymmetric encryption algorithm to encrypt plaintext blocks, then uses data compression technology to get cipher blocks, and finally connects MAC address and AES key encrypted by ECC to form a complete ciphertext message. Through the description and implementation of the algorithm, the results show that the algorithm can reduce the encryption time, decryption time and total running time complexity without losing security.**

Index Terms* - *wireless sensor networks(WSNs), AES, ECC, data compression, hybrid encryption algorithm

I. INTRODUCTION

Wireless Sensor Networks (WSNs) is a distributed or centralized sensor network, widely used in many fields such as health care, intelligent transportation, environmental monitoring, military and so on. Wireless communication is the main way to exchange data between sensor nodes. Compared with wired communication, wireless communication networks are more vulnerable to attacks from the physical layer, link layer and network layers, such as congestion attack, forgery attack, and collision attack. Therefore, it is very important for WSNs to establish a security prevention mechanism. A safe and efficient encryption algorithm is the most important of many security mechanisms.

Symmetric encryption and asymmetric encryption are two important parts of traditional encryption algorithms. Commonly used symmetric encryption algorithms are data encryption standard [1] (DES), advanced data encryption standard [2] (AES). AES is more secure than DES. Commonly used asymmetric encryption algorithms are RSA encryption algorithm [3] and elliptic curve algorithm [4] (ECC). Compared with ECC, ECC is more efficient than RSA.

At present, many scholars have proposed encryption algorithms for WSNs. In [5], Chennai et al. provide a solution to the flooding attacks in WSNs by message digest algorithm combined with public key encryption, which improves the detection capability of the network for data packet transmission. NS2 simulation results show that this method is effective for flood attack defense in WSNs. However, this method only signatures data transmitted by nodes to prevent flooding attacks, and has a limited preventive effect on data eavesdropping attacks, so it needs to be combined with other encryption methods in practical application. In [6], Rennis et al. proposed a method of data storage combined with homomorphic encryption for some WSNs which are insensitive to real-time data. In some WSNs (such as underwater sensor networks), people are more concerned about the data as a whole, rather than the data at a certain time. In view of this situation, and considering that large, long-term and long-distance transmission of data may cause loss to sensor network nodes, the paper proposes to pre-store data to a relay node and preprocess it, so as to avoid it. Data is transmitted in a fragmented and uninterrupted manner. At the same time, homomorphic encryption is used to ensure security. This method saves energy and data can be stored safely. But it is not suitable for most networking environments and has some limitations. In [7], Hyoungsuk et al. proposed another encryption scheme. Considering the limited energy and processing capacity of WSNs, they propose a method of data encryption without data encryption and data transmission channel encryption. One of the advantages of this method is that it saves the cost of data encryption, and at the same time, the security is guaranteed. Of course, its limitations are also great. Firstly, WSNs have limited channels, which makes it impossible for all channels to be secure. Malicious attackers can eavesdrop on all channels to crack. In addition, for nodes, the energy consumption of data transmission is sometimes greater than that of data processing, which also leads to the ultimate energy consumption of this method may even be greater than that of the data encryption method, so there are some defects. In [8], Soufiene et al. compared the current symmetric encryption algorithms. By comparing AES, RC5 and RC6 algorithms, it is concluded that AES encryption is the most symmetric and secure symmetric encryption algorithm. The author further improves the AES encryption algorithm. He pointed out that the authentication ability of symmetric encryption algorithm has limitations. AES algorithm should be better applied to WSNs, and it is better to use the corresponding asymmetric encryption algorithm to achieve node authentication. But how to further improve, the author did not give a specific plan. In [9], to solve the problem of limited energy in WSNs, Hani proposed an ECC encryption

978-1-7281-1699-0/19/$31.00 ©2019 IEEE

scheme based on the ElGamal system. This method draws lessons from Elgamal's encryption and decryption process but does not optimize according to the characteristics of WSNs. Firstly, data encryption and decryption based on asymmetric encryption will consume a lot of energy. In addition, WSNs need a lot of authentication operations. This algorithm needs efficient authentication, but the scheme does not give the corresponding method, so its practicability is not high.

In this paper, a hybrid AES encryption algorithm and ECC encryption algorithm is proposed to optimize the encryption algorithm of sensor networks, which can effectively reduce the encryption time, decryption time and total running time complexity on the premise of ensuring security, combining the different characteristics of the symmetric encryption algorithm and asymmetric encryption algorithm.

II. HYBRID ENCRYPTION ALGORITHM

This hybrid encryption algorithm mainly includes three modules: AES encryption module, ECC encryption module, and LZW compression module. AES encryption module and ECC encryption module are used to encrypt the data of WSNs after 1M grouping. Because of the large number of ciphertext sections and the problem of effective grouping, the LZW compression method is used to compress ciphertext blocks.

A. AES Encryption Module

AES encryption algorithm is an iterative block cipher with variable key length and block length. This scheme uses a 128bit key to encrypt plaintext for 10 rounds. The algorithm includes four operations: byte substitution, row shift, column mixing, and round key. The algorithm can effectively resist differential attack and related key attack.

B. ECC Encryption Module

ECC encryption algorithm is a kind of public key cryptography, which has the characteristics of small total computation, fast processing speed, low bandwidth communication requirement, high security, and low total time complexity.

The ECC algorithm is described as follows:

Let $E_p(a,b)$ be the Abel group generated by an elliptic curve in the finite field Z_p ($p>3$ prime number), take $f \in E_p(a,b)$, and satisfy the elliptic curve discrete logarithm problem on the subgroup H generated by f is difficult to solve. Then take a positive integer $k < p$ and calculate $y = xf$. The public key is taken as f, y, p, and the private key is x.

1) Encryption process: For plaintext m, randomly select positive integer $k < p$ and calculate $C_1 = kf$, $C_2 = m \oplus ky$. The ciphertext is $c = (C_1, C_2)$.

2) Decryption process: $m = C_2 \oplus (-x C_2)$.

C. LZW Compression Module

The LZW compression algorithm is a dictionary-based algorithm. The compression module consists of three main steps:

1) Put all individual characters in the dictionary library.

2) Read and evaluate characters to the prefix string.

3) Read the following character and evaluate if it is the end of the file.

The algorithm encoding process is shown in Fig.1.

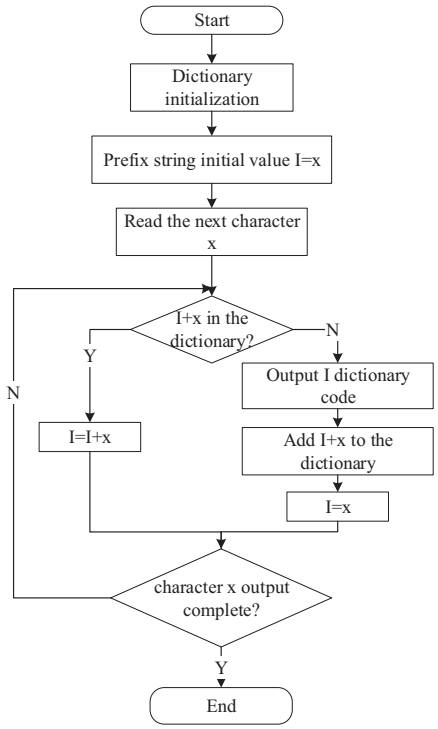

Fig.1 LZW algorithm coding flow chart

III. HYBRID ENCRYPTION SCHEME IMPLEMENTATION

In this paper, a hybrid encryption algorithm using LZW compression is proposed on the basis of reference [10]. It combines the advantages of ECC encryption with high security and AES encryption with high efficiency.

A. Encryption process

The plaintext message P is grouped by 1M, and the group is recorded as d . If $d \bmod 2 \equiv 1$: the plaintext message block $d \bmod 2 \equiv 1$ is encrypted by the AES encryption module, and the ciphertext message block is obtained, the LZW compression module is used to obtain the density of the sub-encryption module. Then use the packet clear text message block of $d \bmod 2 \equiv 0$ to encode to $E_p(a,b)$ Using the ECC encryption module to obtain another part of the ciphertext message block, and then compress the ECC encrypted ciphertext message block by using LZW to add the AES encryption. After the ciphertext message block. Finally, get some ciphertext. This step is the core part of the hybrid encryption algorithm, as shown in Fig.2.

To ensure security and integrity, the MAC address and the AES key are encoded into $E_p(a,b)$ and encrypted by the ECC encryption module to obtain a ciphertext including the MAC address and the AES key. This method can ensure the security of the MAC address when the receiver performs verification

and the potential security problem of overcoming the key distribution of the symmetric encryption algorithm.

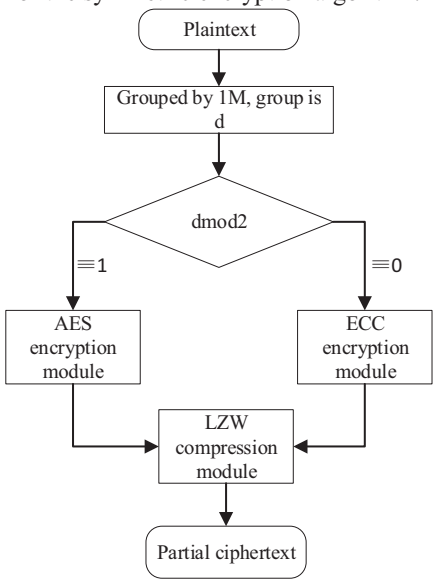

Fig.2 Core encryption module

Connect two parts of ciphertext to the receiver. The full encryption process is as follows:

*1)*The plaintext message P is grouped according to the 1M packet and then encrypted by the AES key and the ECC public key, respectively, and compressed to obtain a ciphertext message.

$$Q_1 = \sum_{d=1}^{n} \{Co[\mathrm{En}(P_{d \bmod 2 \equiv 1})_{k_{AES}}] \| [\mathrm{En}(P_{d \bmod 2 \equiv 0})_{k_{ECCpub}}]\} \quad (1)$$

2) Encrypt the AES key and MAC address information using the ECC public key.

$$Q_2 = En(k_{AES}, Add_{MAC})_{k_{ECCpub}} \quad (2)$$

*3)*Connect Q_1 and Q_2 compress to get all ciphertext messages Q.

$$Q = Co(Q_1 \| Q_2) \quad (3)$$

B. Decryption process

After receiving the ciphertext message Q, the receiver decomposes Q into two parts, Q_1 and Q_2. Firstly, the ECC decryption module is used to decrypt the ciphertext Q_2 containing the MAC address and the AES key, then the MAC address and the AES key are obtained. The ciphertext of the message is then grouped according to the compression module, and the group is denoted as f. If $f \bmod 2 \equiv 1$, the ciphertext is decrypted using the obtained AES key after decompression. If $f \bmod 2 \equiv 0$, decompress and use the decryption module of ECC to decrypt. After the end, the stitching is plain text. The decryption process for ciphertext Q_1 is the core decryption module, as shown in Fig.3.

The complete decryption process is as follows:

1) Decompress Q to get Q_1 and Q_2.

2) Decrypt the ECC private key k_{ECCpri} to Q_2 obtain the AES key and MAC address information.

*3)*Verify the MAC address information obtained in step2.

*4)*Decrypt the different compressed data blocks with the AES key or the ECC private key obtained in step 2 to obtain the plaintext P.

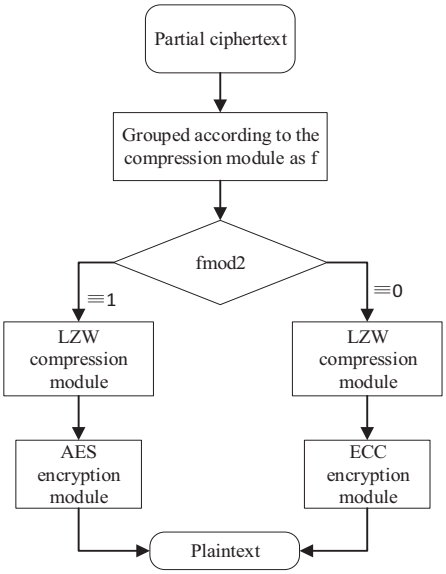

Fig.3 Core decryption module

IV. COMPARE PLAN

In terms of speed and efficiency, this paper simulates the method proposed in [10] and the improved method in document [10] by using 1 M, 10 M, 20 M and 30 M data packets, and compares the key generation time, encryption time, decryption time and total complexity of running time.

The two algorithms have no difference in key generation time. The key generation time is shown in Table 1.

TABLE I
COMPARISON OF THE KEY GENERATION TIME OF THE LITERATURE [10] AND THE PROPOSED ALGORITHM

Packet size /M	Literature [10] key generation time /ms	Proposed algorithm Key generation time /ms
2	1	1
10	1	1
20	1	1
30	1	1
40	1	1
50	1	1

In terms of encryption time, the algorithm proposed in this paper is obviously less than the algorithm proposed in [10] for 2M, 10M, 20M, 30M, 40M, 50M data packets, as shown in Fig.4. It is easy to see that there is a positive correlation between the encryption time and the size of the data packet. When the packet size is 2M, there is no significant difference in the encryption time between the two encryption algorithms. However, as the packet size increases, the encryption speed of the algorithm proposed in [10] is slowed down.

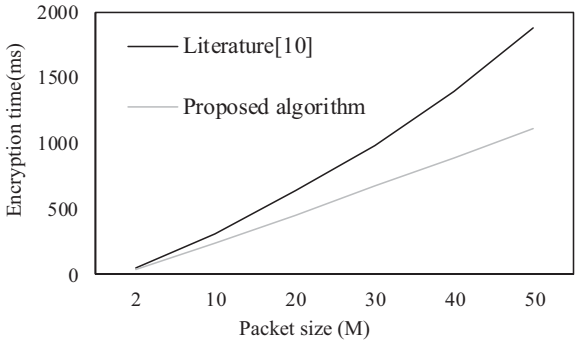

Fig.4 Comparison of encryption time

In terms of decryption time, the algorithm proposed in this paper takes less time than the algorithm proposed in [10]. As shown in Fig.5. Based on the analysis of the relationship between the decryption time and the packet size, it can be seen that there is a positive correlation between the two. And when the data packet is larger than 10M, the decryption speed of the algorithm proposed in [10] is slowed down, and the algorithm proposed in this paper has no significant difference between the decryption speed and the decryption speed of less than 10M data packet.

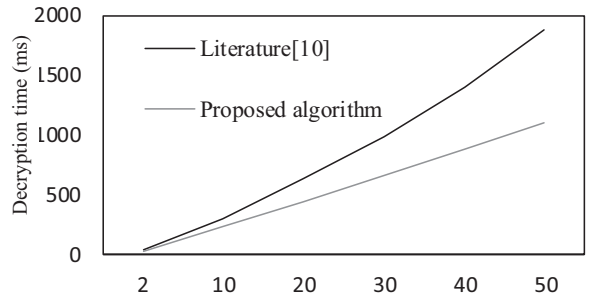

Fig.5 Comparison of decryption time

In terms of the total complexity of running time, the total complexity of the running time of the proposed algorithm for data of different packet sizes is lower than that proposed in [10]. As shown in Fig.6. Compared with the complexity of the total time of operation in the reference [10], the algorithm is obviously much lower. And when the data packet is larger than 20M, the total complexity of the running time of the algorithm proposed in [10] is slowed down, and the total complexity of the running time of the proposed algorithm is basically unchanged.

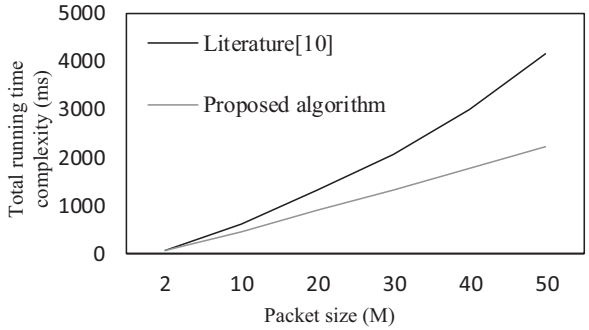

Fig.6 Comparison of total running time complexity

In terms of security, reference [10] adopts ECC encryption method for all plaintext, which can achieve a good level of security. In this paper, ECC and AES are used to encrypt messages simultaneously. Although the security of AES is much lower than that of ECC algorithm, this paper only encrypts some plaintext messages with AES and uses ECC to encrypt another part of plaintext messages and AES keys with a higher security level. Even if the attacker breaks the ciphertext message encrypted by AES, only about one-half of the plaintext message can be obtained. Therefore, the security of the algorithm proposed in this paper is not very different from that of the algorithm proposed in reference [10]. Combining with the simulation results, the improved algorithm in this paper can perform faster computation on the premise of ensuring the same security as that in reference [10].

V. Conclusions

With the advent of the 5G era, the network system based on wireless sensors will increase day by day, and the data generated in each network will also express more abundant meanings, which will lead to a sharp rise in the size of communication packets. At this time, the data encryption efficiency of wireless sensor and related networks should be effectively improved. The hybrid algorithm proposed in this paper combines the characteristics of symmetric cryptography which is easy to calculate and asymmetric cryptography which has high secrecy. It improves the efficiency of encryption and decryption while ensuring security. Compared with the existing algorithms, it effectively reduces the encrypting and decrypting time and the total running time complexity.

Acknowledgment

This paper has been supported by China's Post-doctoral Science Found under Grant 2018M643727, National Natural Science Foundation of China under Grant 51705030, Shanxi province Natural Science Funding 2019JM-099 ,Youth Teacher Fund of Xi'an University of Posts & Telecommunications No.401-205020001, Shaanxi Natural Science Fundamental Research Project: 2019JM-099,Soft Project of the Ministry of Industry and Information Technology: 2019R28.

References

[1] Z. Yingbing and L. Yongzhen, "The design and implementation of a symmetric encryption algorithm based on DES," *2014 IEEE 5th International Conference on Software Engineering and Service Science*, Beijing, 2014, pp. 517-520.

[2] N. Floissac and Y. L'Hyver, "From AES-128 to AES-192 and AES-256, How to Adapt Differential Fault Analysis Attacks on Key Expansion," *2011 Workshop on Fault Diagnosis and Tolerance in Cryptography*, Nara, 2011, pp. 43-53.

[3] K. Balasubramanian, "Variants of RSA and their cryptanalysis," *2014 International Conference on Communication and Network Technologies*, Sivakasi, 2014, pp. 145-149.

[4] D. Kobrle and R. Lorencz, "Optimization of elliptic curve operations for ECM using double & add algorithm," *2015 Fourth International Conference on e-Technologies and Networks for Development (ICeND)*, Lodz, 2015, pp. 1-4.

[5] S. Vidhya and T. Sasilatha, "Performance analysis of black hole attack detection scheme using MD5 algorithm in WSN," *2014 International Conference on Smart Structures and Systems (ICSSS)*, Chennai, 2014, pp. 51-54.
[6] Y. Ren, V. Oleshchuk and F. Y. Li, "A distributed data storage and retrieval scheme in unattended WSNs using Homomorphic Encryption and secret sharing," *2009 2nd IFIP Wireless Days (WD)*, Paris, 2009, pp. 1-6.
[7] H. Jeon, J. Choi, S. W. McLaughlin and J. Ha, "Channel Aware Encryption and Decision Fusion for Wireless Sensor Networks," in *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 619-625, April 2013.
[8] S. Ben Othman, A. Trad and H. Youssef, "Performance evaluation of encryption algorithm for wireless sensor networks," *2012 International Conference on Information Technology and e-Services*, Sousse, 2012, pp. 1-8.
[9] S. Ben Othman, A. Trad and H. Youssef, "Performance Evaluation of EC-ElGamal Encryption Algorithm for Wireless Sensor Networks," *Wireless Mobile Communication and Healthcare*, Paris, 2012, pp. 271-285.
[10] Liang W, Weixin W and Lin Z,"A new key agreement algorithm for sensor networks," *Transducer and Microsystem Technologies*, vol.32, no.7, 2013, pp.129-131.