Journal of Engineering Science and Technology Review 16 (4) (2023) 109 - 118

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

A Novel Network Data Encryption Method Based on TrustZone

Hongyan Xu* **and Yaodong Yuan**

Department of Health Administration, Zhengzhou Shuqing Medical College, Zhengzhou 450064, China

Received 23 May 2023; Accepted 15 August 2023

Abstract

To ensure the security of network data transmission, a network data encryption method based on TrustZone was analyzed. The hardware layer was used to provide hardware support for network data encryption. The framework layer was provided with a trusted operating kernel and a common kernel with a rich execution environment through the kernel layer. The framework layer used the key security management unit on the client-application-side to create, write, read, and delete the security key file. The network data encryption and decryption unit on the client-application-side constructed a session connection between the client application side and the trusted application side under the kernel provided by the kernel layer by calling the hardware of the hardware layer. Moreover, the network data encryption and decryption instructions were transmitted to the trusted-application-side network data encryption and decryption unit. The encryption of the memristor neural network and the cubic chaotic map were employed. The network data encryption and decryption results were presented to the user through the application layer. Experiments results prove that, the interval of the optimal key chaotic sequence must be controlled between [−1,1]. The ASCII code value of the network data character sequence before encryption has an obvious distribution pattern to encrypt the network data to be transmitted under the wireless sensor network scenario. The value of the 0-1 balance index of the encrypted network data of this study's method is the closest to 0, and the highest values are approximately 0.0088, 0.0079, and 0.0067. The method in this study can effectively encrypt and decrypt network data. Under different attack types, this method has high integrity of encrypted network data and a high value of avalanche effect. Thus, it has effective anti-attack performance.

Keywords: TrustZone; network data; encryption method; memristor; neural network; cubic chaotic map

1. Introduction

In the digital age, the rapid growth of the Internet has created the need to transmit and store large amounts of sensitive information.

Networks can help people transmit different data types and provide convenience; however, they are challenged by different types of attacks, affecting data transmission security [1]. The existing network data security transmission products include intrusion detection systems and firewall systems; however, these products only start to stop the discovered security risks after discovering network data security threats. These products tend to lead to network data leakage and problems, such as tampering, affecting the security of network data transmission. The lack of perfect defense capabilities provides convenience for attackers and increases the difficulty of finding traces of attacks [2].

Encryption in network communication has become a common solution to protect user privacy and data integrity. As an important means of protecting data security, network data encryption encrypts and transforms data, thereby allowing only the legitimate receiver to decrypt these data; thus, network data encryption can effectively prevent attacks, such as illegal eavesdropping, data tampering, and hijacking. The development of network data encryption technology provides a crucial guarantee for strengthening data security [3]. Moreover, hardware security is increasingly emphasized and constantly undergoing technological innovation. As a hardware-based security extension, TrustZone technology is widely used in computing devices. TrustZone technology realizes the isolation of security-sensitive information by dividing the processor and storage into secure and nonsecure areas to provide high security [4].

Nowadays, the wide variety of network applications and the frequent occurrence of hacker attacks make people recognize the harm of network data leakage; moreover, the concern about the security of network data transmission gradually increases [5]. Network data leakage affects the personal property security of individuals, increases the economic loss of enterprises, and leads to the loss of national military secrets of the state [6]. Therefore, this study investigates the network data encryption method based on TrustZone to address the above scenarios. We hope to design a secure and efficient solution by combining TrustZone technology and network data encryption to improve data transmission and storage while ensuring information security. This study is expected to contribute to the development of the network data encryption field, provide a reliable and efficient security solution, and promote the further development of data security in the Internet era.

In this study, TrustZone was used to design the technical architecture of the network data encryption method and the session process of network data encryption and decryption from the client application (CA) side and trusted application (TA) side. The key to network data encryption and decryption is securely managed to realize the network data encryption method in this study. The proposed method was verified through comparative experiments to analyze the application effect of the proposed method.

*E-mail address: ihangspace@outlook.com

ISSN: 1791-2377
doi:10.25103/jestr.164.14

2. Literature review

In recent years, to enhance the security of network data transmission, findings in the field of network data encryption have received attention. Ju et al. [7] used convolutional neural networks to classify network data. They obtained encrypted and unencrypted data and stored them in designated hardware storage devices. An asymmetric searchable encryption algorithm was obtained by fusing symmetric encryption schemes and searchable algorithms. The anti-information leakage performance was improved, the symmetric searchable encryption algorithm was used to encrypt and process the unencrypted data obtained by classification, and the network data encryption was completed. This method has an average encryption time of approximately 0.68s and high encryption efficiency. However, this method must store the encrypted network data in a designated hardware storage device, which has high development costs and poor universality. Teng et al. [8] added random interference information to the traditional Advanced Encryption Standard (AES) algorithm and improved the column mixing operation and key arrangement in the AES algorithm. They also proposed an improved AES algorithm and used it to generate keys and encrypt network data. Thus, network data security is improved. The proposed improved AES algorithm can effectively encrypt network data and speed up the efficiency of network data decryption. However, it does not consider the security of the key in the network environment. The transmitted key may be tampered with if the key for encrypting network data needs to be transmitted. Thus, the security of the key cannot be ensured, and the network data can be affected. For encrypted security, Chen et al. [9] obtained the Triple Data Encryption Standard (3DES) algorithm by cascading the three-layer DES algorithm. The 3DES algorithm was used in generating a key with a length of 56 bits to encrypt network data and avoid the avalanche effect. This method can effectively resist tampering attacks and improve the security of network data transmission. However, this method cannot resist replay attacks. During network data transmission, network data processing requests must be repeatedly sent to block previous operations, thereby threatening the timeliness of network data transmission. Harn et al. [10] designed a lightweight aggregate data encryption method to reduce key size. Multiple short-length pairwise shared keys were used to encrypt network data. This method can effectively encrypt network data, and the encryption speed is fast. However, this method needs to receive keys frequently, increases the energy consumption of key transmission, and reduces the security and stability of the channel. Thus, the security of network data transmission is affected. Arroyo et al. [11] changed the plaintext format of important information in the network data through the Polybius cipher to obtain the ciphertext. The Huffman coding algorithm was used to compress the ciphertext. The ciphertext was embedded into the ciphertext through the least significant bit algorithm of steganography. In the original data, network data encryption was completed. This method can effectively encrypt network data, and the error of network data encryption is small. However, this method does not perform unified management on the encrypted network data, thereby affecting the efficiency of subsequent network data decryption.

TrustZone technology belongs to the Advanced RISC Machine (ARM) processor core and security extension components, which provide protection and isolation for hardware chips. It can establish a programmable environment to prevent the confidentiality and integrity of network data from being attacked. Moreover, it uses a bus to divide the hardware and software resources of the architecture, thereby obtaining two execution environments [12]. These two environments, namely, the rich execution environment (REE) and the secure trusted execution environment (TEE), are parallel. The two worlds run on the same core by extending the ARM core [13]. TrustZone technology can effectively isolate REE and TEE, perform network data encryption and key management operations in TEE, resist various types of attacks, such as tampering and replay attacks, and improve the security of network data transmission. The development costs are low. Therefore, the network data encryption method based on TrustZone is studied to improve the integrity and confidentiality of network data transmission.

3. Methodology

3.1 Technical Architecture of the Network Data Encryption Method

A network data encryption method is designed using TrustZone. This method mainly includes two functional units: the key security management unit and the network data encryption and decryption unit. The technical architecture of the network data encryption method is shown in Fig. 1.

The hardware layer provides hardware support for network data encryption methods while offering the hardware-level isolation of TrustZone. TrustZone technology switches the secure and non-secure states of the ARM core according to the call command of the security monitoring mode, ensuring the hardware-level isolation of the technical architecture when sharing a set of hardware devices [14]. Moreover, a safe and reliable trusted execution for an upper-layer application environment is provided.

The kernel layer provides the trusted operation kernel of the TEE for the network data encryption method and the common kernel of the REE. The Linux kernel provides rich operations for the REE, whereas the OP-TEE kernel provides trusted operations for the TEE. These two cores connect the framework layer to the hardware layer. This layer has state-switching and session-building functions.

The framework layer belongs to the core layer of the data encryption method, which mainly includes two parts: the CA side and the TA side. The security key file is created in the key security management unit on the CA side, including directory files and secure encrypted network data files. The directory file is used to store and manage the information of all security network data files, and the specified security network data files can be quickly found. Writing the security key file allows storing the key generated by the encryption algorithm in the specified security key file. Reading the security key file allows the required keys in the security key file to be read [15]. Deleting the security key file deletes the keys that the user does not need in the security key file. The key security management unit on the TA side is responsible for realizing the functions of the key security management unit on the CA side. The network data encryption and decryption unit on the CA side is responsible for providing specific functional interfaces for the application layer, constructing a session connection between CA and TA, exchanging network data, switching states simultaneously, and entering the TEE. The network data encryption and decryption unit on the CA side transmits the network data encryption and decryption instructions and the network data to the network data encryption and decryption unit on the TA side [16]. The encryption algorithm of the network and the cubic chaotic map completes the encryption and decryption of network data and returns the encryption and decryption results to the network data encryption and decryption unit on the CA side.

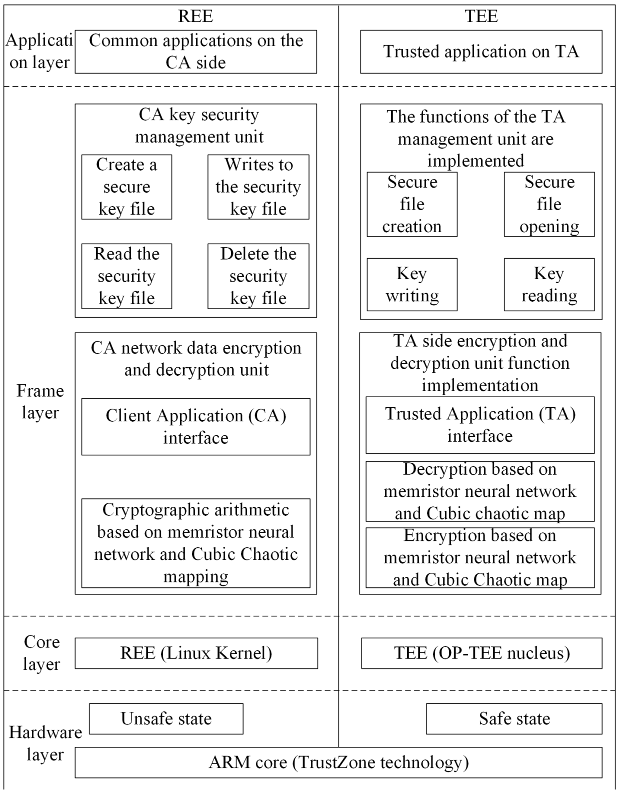

REE
TEE
Application layer
Common applications on the CA side
Trusted application on TA
Frame layer
CA key security management unit
Create a secure key file
Writes to the security key file
Read the security key file
Delete the security key file
The functions of the TA management unit are implemented
Secure file creation
Secure file opening
Key writing
Key reading
CA network data encryption and decryption unit
Client Application (CA) interface
Cryptographic arithmetic based on memristor neural network and Cubic Chaotic mapping
TA side encryption and decryption unit function implementation
Trusted Application (TA) interface
Decryption based on memristor neural network and Cubic chaotic map
Encryption based on memristor neural network and Cubic Chaotic map
Core layer
REE (Linux Kernel)
TEE (OP-TEE nucleus)
Hardware layer
Unsafe state
Safe state
ARM core (TrustZone technology)

Fig. 1. Technical architecture of the network data encryption method based on TrustZone

The application layer provides users with function entry and inputs network data encryption and decryption instructions. The application layer performs network data encryption and decryption operations by calling the functional interface in the framework layer. It also returns the corresponding network data encryption and decryption results.

3.2 Conversation flow between the CA side and the TA side of network data encryption and decryption

In the network data encryption method based on TrustZone under the TrustZone technology, the TEE and the REE use the CA and the TA sides in the framework layer to complete the network data exchange. The specific session flow is shown in Fig. 2.

The conversation process between the CA and the TA sides is as follows: First, the context of network data encryption and decryption is constructed starting from the common application of CA in the REE [17]. Then, the session is opened, and the session command is initiated. Finally, the feedback on the network data encryption and decryption session result is obtained. Similarly, the TEE has a corresponding session opening function. It also realizes the specific functions of the network data encryption and decryption unit and returns the network data encryption and decryption session results to the REE side. In the conversation process between the CA and the TA sides, the TEE supplicant, which is the link between the two environments, realizes the network data exchange between the REE and the TEE. After the session connection between the CA side and the TA side is established, the CA side can send the network data encryption and decryption command to the TA side. After receiving the network data encryption and decryption command, the TA side first checks whether the command is safe. If the command is a security command, then this command is executed. Moreover, the memristor neural network and the cubic chaotic map encryption method are used to encrypt and decrypt network data, and the network data encryption and decryption results are returned to the CA side.

The network data encryption and decryption request instruction is implemented in the RPC in the OP-TEE core; that is, it is implemented by remote calling [18]. In the OP-TEE core, the REE and the TEE can initiate network data encryption and decryption requests, and the hardware provided by the hardware layer is used to complete the switching between REE and TEE. In the network data encryption method based on TrustZone, if the network data encryption function or key management function is required, the CA side in the REE has an external interface, which is used to realize the network data encryption and decryption between the CA side and the TA side. For instruction transmission, the TA side implements network data encryption and decryption or key management functions, and the network encryption and decryption results or key management results are returned to the CA side. Therefore, all actual operations are completed in the TEE, thereby ensuring the security of network data encryption and decryption and key management.

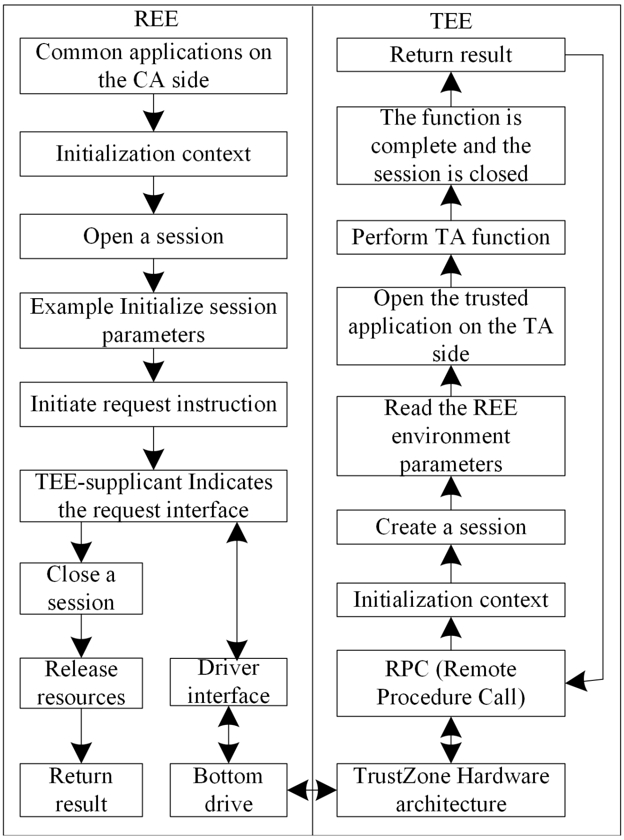

REE
TEE
Common applications on the CA side
Initialization context
Open a session
Example Initialize session parameters
Initiate request instruction
TEE-supplicant Indicates the request interface
Close a session
Release resources
Return result
Driver interface
Bottom drive
Return result
The function is complete and the session is closed
Perform TA function
Open the trusted application on the TA side
Read the REE environment parameters
Create a session
Initialization context
RPC (Remote Procedure Call)
TrustZone Hardware architecture

Fig. 2. Session flow between the CA and the TA

3.3 Key Security Management for Network Data Encryption and Decryption

The security key management unit on the CA side in the framework layer is responsible for creating, writing, reading, and deleting security key files. The specific steps for creating a security key file are as follows:

Step 1: Obtain session control instructions for encrypting and decrypting network data on the TA side.
Step 2: Obtain the context information of encryption and decryption of network data on the TA side according to the session control instruction [19].
Step 3: Check the operation authority for creating the security key file.
Step 4: Obtain the *tee_pobj* structure that is combined with the context information and created by the securely encrypted network data file.
Step 5: Fill in the parameters of the security key file creation in the *tee_pobj* structure.
Step 6: Add the *tee_pobj* structure variable to the secure encrypted network data queue [20].
Step 7: Create a secure encrypted network data file according to the secure encrypted network data queue.
Step 8: Create directory files for securely encrypted network data.
Step 9: Set the file number for the created directory file. The name of the directory file is the set file number.
Step 10: In the directory file, write the created secure encrypted network data file to complete the creation of the security key file.

The directory file is used to find the required security key file quickly and accurately before the security key file can be written and read. The network data encryption and decryption results are obtained by reading the specified encryption and decryption key.

3.4 Implementation of the network data encryption method

The network data encryption unit on the TA side in the framework layer realizes the encryption and decryption of network data by calling the initial key in the security key management unit on the CA side using the memristor neural network and the cubic chaotic map encryption method. In the memristor neural network, the network data that need to be encrypted are input, and the network data are encrypted once. The weight value used by the memristor neural network to encrypt the network data once is the security key in the security key management unit on the TA side. The key chaotic sequence for the secondary encryption of network data is generated to complete the secondary encryption using the weight update result of the memristor neural network as the initial value of the cubic chaotic map.

The nonlinear window function for encrypting network data with a memristor is

$$f(x)=4\left[\frac{w}{H}-\left(\frac{w}{H}\right)^{2}\right] \tag{1}$$

The network data that need to be encrypted are x; the width of the T_iO_2 between the two platinum contacts is H; the width of the doped region is w.

The resistance value of the memristor when encrypting network data is $\varepsilon(t)$, where the network data encryption time is t. The calculation formula of $\varepsilon(t)$ is as follows:

$$\varepsilon(t)=\varepsilon_{\min}+\frac{\varepsilon_{\max}-\varepsilon_{\min}}{\exp\left(\bar{\mu}\frac{\varepsilon_{\min}}{D^{2}}q(t)+1n\frac{\varepsilon_{\max}-\varepsilon_{\min}}{\varepsilon_{0}-\varepsilon_{\min}}\right)+1} \tag{2}$$

The minimum and maximum resistances of the memristor are $\varepsilon_{\min}$ and $\varepsilon_{\max}$, respectively; the current passing through the memristor is $q(t)$; the average mobility is $\bar{\mu}$; the initial resistance of the memristor is ε_0.

Let the conductance of the memristor be $G(t)$. The calculation formula is as follows:

$$G(t)=\frac{1}{f\left[\varepsilon(t)\right]}=\frac{\exp\left(\bar{\mu}\frac{\varepsilon_{\min}}{H^{2}}q(t)+\ln\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_{0}-\varepsilon_{on}}\right)+1}{f\left[\varepsilon_{\min}\cdot\exp\left(\bar{\mu}\frac{\varepsilon_{\min}}{H^{2}}q(t)+\ln\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_{0}-\varepsilon_{on}}\right)+\varepsilon_{\max}\right]} \tag{3}$$

The formula for the change in conductance ΔG can be obtained using the differential of *G* to *t*. The formula is as follows:

$$\frac{dG}{dt}=\frac{\bar{\mu}\frac{\varepsilon_{\min}}{H^{2}}i(t)\exp\left(\bar{\mu}\frac{\varepsilon_{\min}}{H^{2}}q(t)+\ln\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_{0}-\varepsilon_{on}}\right)\cdot\left(\varepsilon_{\max}-\varepsilon_{\min}\right)}{f\left[\varepsilon_{\min}\cdot\exp\left(\bar{\mu}\frac{\varepsilon_{\min}}{H^{2}}q(t)+\ln\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_{0}-\varepsilon_{on}}\right)+\varepsilon_{\max}\right]^{2}} \tag{4}$$

Among them, $\frac{dq(t)}{dt}=i(t)$; in the case of time variation $\Delta t\to 0$, $\frac{dG}{dt}\approx\Delta G$. Then, the variation $\Delta\omega$ of the neural network weight can be replaced by ΔG.

The Chebyshev type II polynomial $C(t)$ is taken as the activation function of the memristor neural network. In this case, the primary encryption result y of the network data output by the memristor neural network is as follows:

$$y=\sum_{i=1}^{n-1}x\omega_i C_i(t) \tag{5}$$

The number of neurons in the output layer is n; the weight and activation function of the i neuron are ω_i and $C_i(t)$; the value of ω_i is the security key in the security key file created by the key management unit on the CA side.

The one-time encryption error of the network data is as follows:

$$e_j=\frac{\sum_{j=1}^{h}(\hat{y}_j-y_j)}{2} \tag{6}$$

The number of network data samples is h; the expected and actual primary encryption results of the j network data sample x_j are $\hat{y}_j$ and y_j.

ΔG is used to replace $\Delta\omega$. In this case, when the current is small, the change in G is also small, which is very close to 0; when the current is large, the change in G is also large. The updated formula of $\Delta\omega$ can be deduced through the memristor equation. The formula is as follows:

$$\Delta\omega_i=\frac{\bar{\mu}\frac{\varepsilon\min}{H^2}\exp\left(\bar{\mu}\frac{\varepsilon\min}{H^2}E(t)+1n\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_0-\varepsilon_{on}}\right)\cdot(\varepsilon_{\max}-\varepsilon_{\min})\cdot\eta\cdot e_j\cdot C_i(t)}{f\left[\varepsilon_{\min}\cdot\exp\left(\bar{\mu}\frac{\varepsilon\min}{H^2}q(t)+1n\frac{\varepsilon_{off}-\varepsilon_{on}}{\varepsilon_0-\varepsilon_{on}}\right)+\varepsilon_{\max}\right]} \tag{7}$$

The learning rate is η; the integral of e_j is $E(t)$, $E(t)=\int e_j(t)dt$.

The updated formula of ω_i can be obtained through $\Delta\omega_i$. The formula is as follows:

$$\hat{\omega}_i=\omega_i+\Delta\omega_i \tag{8}$$

The weights before and after updating are ω_i and $\hat{\omega}_i$.

ω_i is taken as the initial value of the one-dimensional cubic chaotic map. In this case, the formula of the one-dimensional cubic chaotic map is as follows:

$$\hat{\omega}_{k+1}=a\hat{\omega}_k^3-b\hat{\omega}_k \tag{9}$$

The parameters are a and b; the number of iterations is k.

Let the chaotic series of the secondary encryption key of network data be zk, $zk=r\hat{\omega}_k+r$, where the encryption coefficient is r. Then,

$$\begin{cases}\hat{\omega}_k=\frac{z_k}{r}-1\\ \hat{\omega}_{k+1}=\frac{z_{k+1}}{r}-1\end{cases} \tag{10}$$

Formula (10) is introduced into Formula (9). Formula (9) is integrated to obtain

$$z_{k+1}=\frac{az_k^3}{r^2}-\frac{4bz_k^2}{r}+9z_k \tag{11}$$

The value of r is 2^{L-1}, where the word length of the computer is L, to ensure that the iteration results of z_k are all integers.

The integrated Formula (11) contains two fixed solutions, 0 and $\frac{3r}{2}$, indicating that when the initial key is 0 or $\frac{3r}{2}$, the key chaotic sequence z_k of the secondary encryption of network data is 0. Formula (11) must be improved to avoid $z_k=0$. The improved formula is as follows:

$$z_k=\begin{cases}\frac{az_k^3}{r^2}-\frac{4bz_k^2}{r}+9z_k+1 & z_k\in 0\bigcup z_k=\frac{3r}{2}\\ \frac{az_k^3}{r^2}-\frac{4bz_k^2}{r}+9z_k-1 & z_k=r\\ \frac{az_k^3}{r^2}-\frac{4bz_k^2}{r}+9z_k & z_k\in\left(0,\frac{3r}{2}\right)\bigcup\left(\frac{3r}{2},r\right)\end{cases} \tag{12}$$

The memristor neural network and cubic chaotic map encryption method are used, and the specific steps for encrypting network data are as follows:

Step 1: Select the initial key in the key security management unit on the CA side. In the memristor neural network in the network data encryption and decryption unit on the TA side, input the network data transmitted through the session connection between the CA side and the TA side and output the network data. The data are encrypted once.

The updated result $\hat{\omega}$ of the weight ω is obtained according to the encrypted result of the first encryption, and $\hat{\omega}$ is the initial key of the cubic chaotic map.

Step 2: Use the cubic chaotic map to perform $k+l$ iterations on $\hat{\omega}$ to generate the chaotic sequence $Z=\{z_1,z_2,\cdots,z_l\}$ of $\hat{\omega}$, where the number of chaotic sequences is l, $l=M\times N$. The rows and columns of the network data plaintext characters are M and N.

Step 3: Build an index matrix U and scramble the network data plaintext character matrix. Convert $Z=\{z_1,z_2,\cdots,z_l\}$ into $M\times N$ chaotic matrix, sort $Z=\{z_1,z_2,\cdots,z_l\}$ in descending order according to row and column order, and obtain U. Process the network data plaintext character matrix x' according to U and obtain the network data plaintext character matrix x'' after scrambling.

Step 4: In the initial key selection, use the cubic chaotic map to iterate $\hat{\omega}$ for $k+8l$ times and generate the chaotic sequence $V=\{v_1,v_2,\cdots,v_{8l}\}$ of $\hat{\omega}$, which is the initial key.

Step 5: For bit reorganization, change the plaintext characters of the network data in x'' to binary form s, convert V to chaotic matrix $M\times 8N$, arrange the plaintext characters in descending order, obtain the index array R, use R to sort the plaintext characters of the network data in s again, and obtain new network data plaintext; sequence O is the plaintext sequence of the network data after bit reorganization [19].

Step 6: For the key sequence generation, use the cubic chaotic map to iterate $\hat{\omega}$ for $k+l$ times, generate two chaotic random sequences β_1 and β_2 of $\hat{\omega}$, randomly select the network data plaintext character $O_{i'}$ in x'', select $\hat{\omega}_{i'1}$ and $\hat{\omega}_{i'2}$ in $\hat{\omega}$, and add $\hat{\omega}_{i'1}$ and $\hat{\omega}_{i'2}$ to β_1 and β_2. Let the length of the chaotic sequence be $l+1$. β_1 and β_2 are processed to improve the security of the encryption method. The processing formula is as follows:

$$\begin{aligned}\beta_1' &= floor(\beta_1\times 10^{14})\bmod\hat{\omega}_{i'1}\\ \beta_2' &= ceil(\beta_2\times 10^{14})\bmod\hat{\omega}_{i'2}\end{aligned}\tag{13}$$

The rounding down function is $floor$; the rounding function is $ceil$; the remainder operation symbol is $\bmod$.

Step 7: For the diffusion, add the network data plaintext sequence O after bit reorganization during the first round of the network data diffusion encryption process. In the case of $l=1$, the first round of diffusion formula is as follows:

$$\psi_1(1)=\mathrm{mod}(O(1)+\beta_1'(2))\oplus(\beta_1'(1)+O_{i'})\tag{14}$$

In case $1<l\le l_{\max}$, the first-round diffusion formula is as follows:

$$\psi_1(l)=\mathrm{mod}(O(l)+\beta_1'(l+1))\oplus\mathrm{mod}(\beta_1'(l)+\psi(l-1))\tag{15}$$

The maximum value of l is $l_{\max}$.

During the second round of diffusion, the results of the first round of diffusion must be added. In the case of $l=l_{\max}$, the formula for the second round of diffusion is as follows:

$$\begin{aligned}\psi_2(1)=\ &\mathrm{mod}(\psi_1(l_{\max})+\beta_2'(l_{\max}))\oplus\mathrm{mod}(\beta_2'(l_{\max}+1)\\ &+\psi_1(1))\end{aligned}\tag{16}$$

In the case of $1\le l\le l_{\max}$, the two-round diffusion formula is as follows:

$$\psi_2(l)=\mathrm{mod}(\psi_1(l)+\beta_2'(l))\oplus\mathrm{mod}(\beta_2'(l)+\psi_2(l+1))\tag{17}$$

ψ_2 is the final network data encryption result.

The decryption algorithm is the inverse operation of the encryption algorithm, and the key used for decrypting network data is the same as the key used for encryption.

4. Experimental results analysis

The wireless sensor network scenario built by network simulation software is taken as the experimental object. The wireless sensor network scenario contains 30 sensor nodes, and the initial energy of each sensor node, which is 150 J, is consistent. The topological structure of the wireless sensor network scenario is shown in Fig. 3.

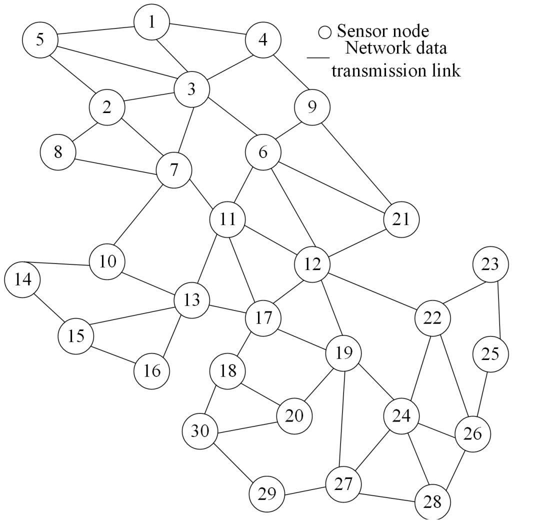
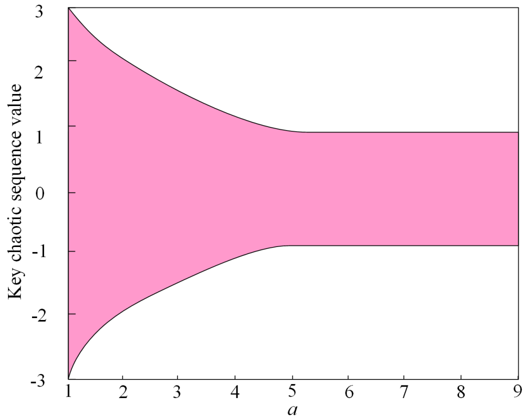
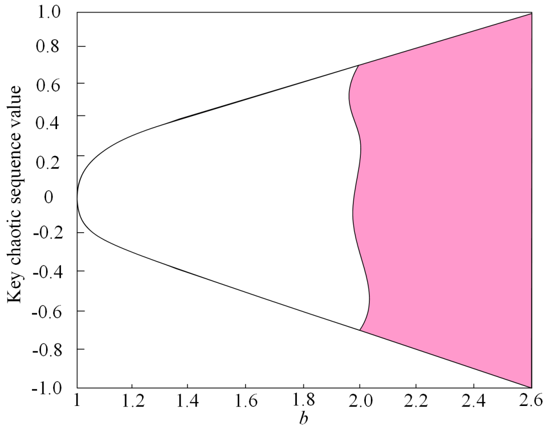


Fig. 3. Topological structure of the wireless sensor network scenario

The values of parameters a and b in the cubic chaotic map in the method of this study directly affect the encryption effect of subsequent network data. Therefore, the optimal values of a and b must be analyzed, and the value interval of the optimal key chaotic sequence needs to be controlled within [−1,1]. The analysis results are shown in Fig. 4. We can determine by analyzing Fig. 4(a) that the value range of the network data encryption key chaotic sequence output by the cubic chaotic map in this method gradually shrinks when the value of a continues to increase. When the value of a in the network data is approximately 5, the value interval of the encryption key chaotic sequence is approximately [−1,1]. This finding shows that the network data encryption effect of the method in this study is the best. We can determine by analyzing Fig. 4(b) that the network data encryption key chaotic sequence output by the cubic chaotic map in this method appears chaotic when the value of B is approximately 2. This finding indicates that the key chaotic sequence generated at this time is good. In particular, the network data encryption effect of the method in this study is the best. Comprehensive analysis shows that when the value of a is 5 and the value of b is 2, the network data encryption effect of this method is the best.



(a) The best value analysis result of a



(b) The best value analysis result of b

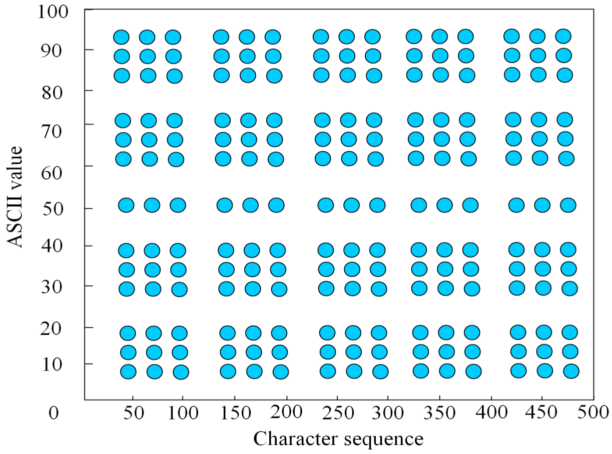
Fig. 4. Analysis results of the optimal values of parameters a and b

The method in this study is used to encrypt the network data that need to be transmitted in the wireless sensor network scenario. The network data encryption results are shown in Fig. 5.

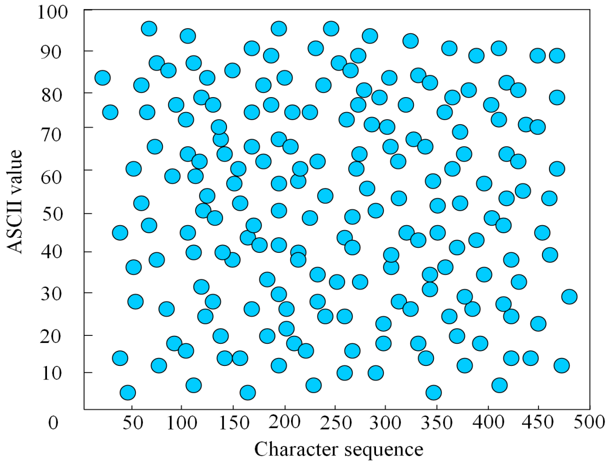
The analysis of Fig. 5(a) shows that the ASCII code values of the network data character sequences that need to be encrypted have clear distribution rules. The analysis of Fig. 5(b) shows that the ASCII code values of the network data character sequences encrypted by the method in this study are relatively uniform. No rule exists. Thus, the distribution rule of the ASCII code value of the original network data character sequence is completely concealed. The experiment proves that the method in this study can effectively encrypt network data.

The balance of the network data encryption by this method is measured by analyzing the proportion of network data plaintext characters and ciphertext characters in the ASCII value before and after encryption. The analysis results are shown in Fig. 6.

We can determine by analyzing Fig. 6(a) that the distribution of plaintext characters in the ASCII code value in the unencrypted network data is not uniform. The proportion of plaintext characters in the network data in the ASCII code value is relatively high when the ASCII code value is lower than approximately 40. This finding indicates that many plaintext characters exist at this time. Thus, this part of plaintext characters may have valid network data prone to leakage. We can determine by analyzing Fig. 6(b) that after being encrypted by the method in this study, the ciphertext characters in the network data in the ASCII code value show a uniform distribution, evenly distributed between 0.003 and 0.008. Moreover, no number of occurrences exists in different ASCII code values. A large number of ciphertext characters indicate that the characters of the network data encrypted by the method in this study have a good balance; that is, the anti-probabilistic statistical attack effect is good.

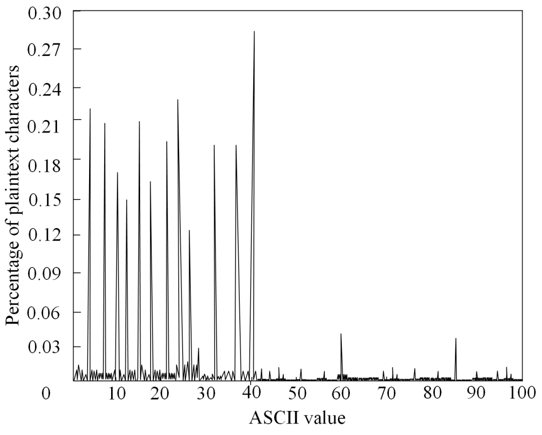


(a) Distribution of the ASCII code values of the network data character sequences to be transmitted

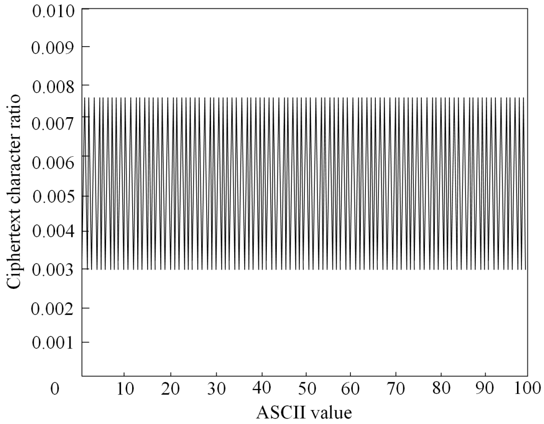


(b) Distribution of the ASCII code values of the encrypted network data character sequences

Fig. 5. Network data encryption results of the method in this study



(a) Proportion of plaintext characters in the ASCII code value before encryption



(b) Proportion of the encrypted ciphertext characters in the ASCII code value

Fig. 6. Analysis results of the balance of the network data encryption by the method in this study

The use of completeness γ_1 and avalanche effect γ_2 for measuring the network data encryption effect of this method shows that the closer the values of γ_1 and γ_2 are to 1, the better the confusion and diffusion effects of the network data encryption by this method are; that is, the encryption effect is improved. The calculation formula of γ_1 is as follows:

$$\gamma_1 = 1 - \frac{1}{\theta_{in}\theta_{out}} \tag{18}$$

The network data input by the encryption method is θ_{in} bit; the output is θ_{out} bit.

The calculation formula of γ_2 is as follows:

$$\gamma_2 = 1 - \frac{2}{\rho(x)\times\theta_{in}\theta_{out}} \sum_{\hat{j}=0}^{\theta_{out}-1} \left(\hat{j}\sigma_{\hat{i}\hat{j}} - \frac{\theta_{out}\rho(x)}{2} \right) \tag{19}$$

The number of characters in the input network data sample is $\rho(x)$; the differential Hamming weight is $\hat{j}$; the number of $\hat{j}$ in the network data sample between the output of only changing the $\hat{i}$ bit and the original output is $\sigma_{\hat{i}\hat{j}}$.

When analyzing different attack types, the method in this study encrypts the effect of network data. The analysis results are shown in Table 1. The analysis of Table 1 shows that the network data encrypted by the method in this study can effectively resist different types of network attacks. In different types of network attacks, the integrity value of the network data encrypted by the method in this study and the value of the avalanche effect is close to 1, which is relatively close. The experiment proves that the method in this study can effectively defend against different types of network attacks. Under different types of network attacks, the integrity and avalanche effect of the encrypted network data by this method are relatively high, indicating that this method has good confusion and diffusion effects on the encrypted network data. In particular, the network data encryption effect is good. Moreover, the network data encrypted by the method in this study have good integrity and timeliness.

Table 1. Analysis results of network data encryption in different attack types

Attack type	**Attack result**	**Result analysis**	γ_1	γ_2
Tampering attacks on write operations	Failure	The number of characters in the network data calculated by the returned data is consistent with that in the original network data packet, indicating that the network data packet is not tampered with.	1	0.999
Tampering attacks on read operations	Failure	The secure TEE returns the number of characters in the read data. The number is consistent with that in the original network packet, indicating that the network packet is not tampered with.	1	0.998
Replay attack on write operations	Failure	Network data write request packet write technology, consistent with the actual write count, indicates that the network data successfully resist replay attacks and that network data packets have timeliness.	1	0.997
Read operation replay attack	Failure	In secure TEE, the random number in the network data response packet is consistent with the actual random number, indicating that the network data successfully withstand replay attacks and that the network data packet has timeliness.	0.999	0.999
Channel attack of write operation	Failure	The attacker cannot obtain the key for encryption and decryption of network data.	1	0.996
Read operation channel attack	Failure	The attacker cannot enter the secure TEE.	1	0.998
Write theft attack	Failure	The attacker cannot generate network data encryption and decryption keys and decrypt network data.	1	0.997
Read operation steal attack	Failure	The attacker cannot enter the secure TEE.	0.999	0.999
Forge attacks on write operations	Failure	The illegitimate identity of the requested visitor is confirmed successfully, and access to network data is denied.	1	0.996
Forgery attacks for read operations	Failure	Access to network data is denied in a secure TEE.	1	0.998

If the network data encryption effect is good, the distribution of 0 and 1 in the network data binary ciphertext should be uniform. The 0-1 balance index is used to measure the statistical attack resistance effect of the network data encryption method in this study. The calculation formula of λ is as follows:

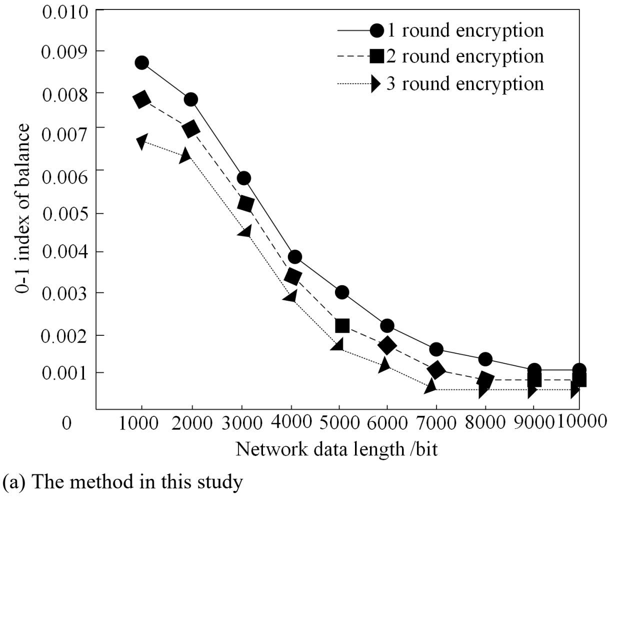
$$\lambda = \frac{|\zeta_0 - \zeta_0|}{\zeta}. \tag{20}$$

The total number of 0 and 1 s in the binary ciphertext of the network data is ζ; the numbers of 0 and 1 s in the binary ciphertext of the network data are ζ_0 and ζ_1. The closer the value of the 0-1 balance index λ is to 0, the better the statistical attack resistance effect of the network data encryption method in this study is.

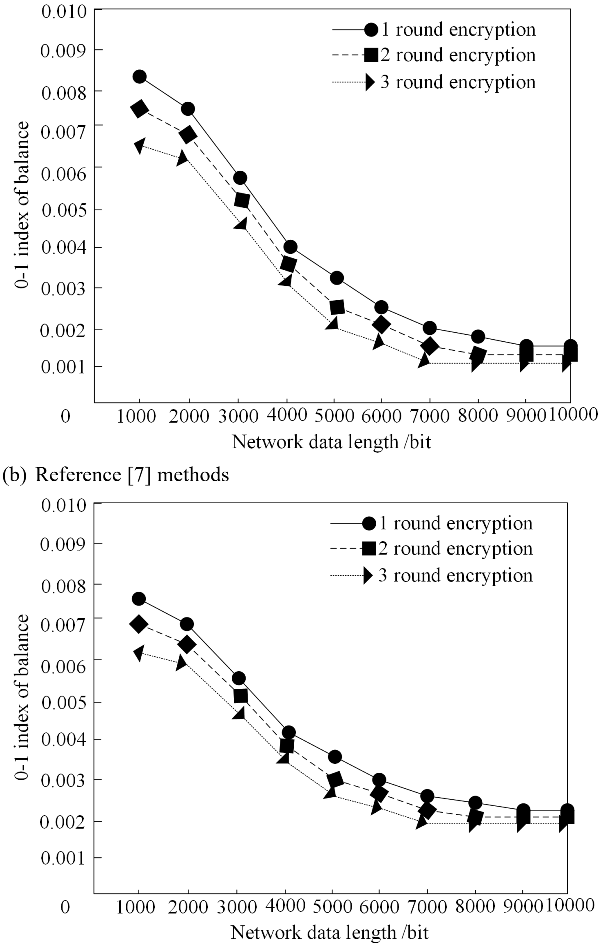
The simulation method of the effective data encryption of documents based on a convolutional neural network in reference [7] and the advanced encryption method of data security based on an improved AES algorithm in reference [8] are used as the comparative methods for verifying the effect of the attack resistance of the network data encryption of the three methods with different network data lengths. The statistical attack-resisting effect under the application of the three methods is shown in Fig. 7.

The result of Fig. 7 shows that with the increase in network data length, the value of the 0-1 balance index of network data encryption shows a downward trend when the number of encryption rounds differs; that is, the longer the network data length is, the better the statistical attack defense effect is. The higher the number of encryption rounds is, the smaller the 0-1 balance index value of the method in this study is; that is, the statistical attack resistance effect is improved. When the number of encryption rounds differs, the 0-1 balance index values of the network data encrypted by this method are relatively close to 0, and the highest values are approximately 0.0088, 0.0079, and 0.0067. This finding shows that the network data ciphertext encrypted by this method performs well. The 0-1 balance index value indicates that the attacker cannot steal sufficient information in the encrypted network data; that is, the statistical attack resistance effect is good.

The 0-1 balance index values of the network data encrypted by the method of reference [7] are close to 0.001, indicating that they fail to approach 0. Moreover, the highest values are approximately 0.0084, 0.0076, and 0.0064. This finding indicates that the ciphertext of the network data encrypted by the method in reference [7] has a relatively good 0-1 balance. The possibility of the attacker stealing the information in the encrypted network data effectively is small, i.e., the statistical attack resistance effect is relatively good. The 0-1 balance index values of the network data encrypted by the method of reference [8] are close to 0.002 and relatively far from 0. Moreover, the highest values are approximately 0.0084, 0.0076, and 0.0064. In particular, the statistical attack-defense effect is relatively good. The 0-1 balance index values of the network data encrypted by the method of reference [8] are close to 0.002 and far from 0. Moreover, the highest values are approximately 0.0078, 0.0069, and 0.0062. This finding indicates that the ciphertexts of the network data encrypted by the method of reference [8] have a relatively poor 0-1 balance. Moreover, the attacker can steal valid information in the encrypted network data. The attacker has a high possibility of stealing valid information within the encrypted network data, i.e., the statistical attack is relatively poorly defended.



(a) The method in this study



(b) Reference [7] methods

(c) Reference [8] methods

Fig. 7. Analysis of the results of the attack-defense performance statistics

5. Conclusion

This study investigates the network data encryption method based on TrustZone to ensure that the data transmitted by users in the network are not at risk of being leaked. TrustZone is used to solve incomplete isolation, high cost, and poor consistency. After experimental verification, the following conclusions are obtained:

(1) The interval of the optimal key chaotic sequence must be controlled between [−1,1]. The analysis of the optimal values of parameters a and b in the cubic chaotic mapping indicates that the method of this study encrypts the data in the network excellently when the value of a is 5 and the value of b is 2.

(2) The ASCII code value of the network data character sequence before encryption has an obvious distribution pattern to encrypt the network data to be transmitted under the wireless sensor network scenario set up in the experiment. After the encryption method of this study is applied to the ASCII code value of the network data character sequence, the distribution uniformity increases, and no pattern to follow exists. Thus, the effective encryption of the network data is realized.

(3) The balance of this study's method is verified to ensure that the encrypted network data can be seen. Encryption before the part of the plaintext characters may exist in the effective network data. The encryption method of this study can be easily applied to network data leakage

problems. In different ASCII code values, no ciphertext characters appear frequently. These findings indicate that the method of network data encryption in this study is balanced, and the anti-probabilistic statistical attack effect is superior.

(4) The completeness γ_1 and the avalanche effect γ_2 are used to measure the encryption effect of the network data of this study's method. The results indicate that the completeness value and the avalanche effect value of the encrypted network data of this study's method are close to 1 under different types of network attacks. This finding indicates that this study's method encrypts network data with good obfuscation and diffusion and that the completeness and timeliness of the data are still intact and preserved. The overall encryption effect is also good.

(5) The method of literature [7] and the method of literature [8] are used for comparison to verify the attack-resistance effect of the network data encryption of the three methods when the network data lengths vary. The results indicate that the value of the 0-1 balance index of the encrypted network data of this study's method is the closest to 0, and the highest values are approximately 0.0088, 0.0079, and 0.0067. Thus, the method in this study has a good overall balance and a good effect of statistical attack resistance.

The above experimental conclusions indicate that the method in this study can effectively isolate the REE and TEE of the network data encryption, provide users with a complete network data encryption function, improve the effect of network data encryption, and successfully defend against various types of network attacks used by attackers. As a result, the leakage of important network data is avoided, and the security of network data transmission is improved. However, the method in this study still has certain shortcomings. In particular, the real network environment may face many variables and challenges. However, the experimental process in this study is mainly conducted based on an idealized environment and model. Therefore, further empirical studies and tests in realistic scenarios are needed in the future to continue improving the methodology of this study. The methodology of this study is expected to provide a reference for the development of computer network communication.

6. Acknowledgment

This study was supported by a grant from the Backbone Teacher Training Program of Zhengzhou Shuqing Medical College (No. 2022zygg04).

References

[1] B. Cho, "Policy-based in-network security management using p4 network dataplane programmability," *Converg. Secur. J.*, vol. 20, no. 5, pp.3-10, Dec. 2020,

[2] M. C. Nkuna, E. Esenogho, and R. Heymann. "Integrating smartphone network architecture and data security techniques to mitigate sharp practices in non-profit organizations," *J. Commun.*, vol. 15, no. 10, pp.755-767, Oct. 2020.

[3] P. Kumar and A. Kumar Bhatt, "Enhancing multi-tenancy security in the cloud computing using hybrid ECC-based data encryption approach," *IET Commun*., vol. 14, no. 18, pp. 3212-3222, Nov. 2020.

[4] H. Yang, "Feature mining method of equipment support data based on attribute classification," *Ordnan. Mater. Sci. Eng.*, vol. 43, no. 6, pp. 124-128, Dec. 2020.

[5] G. A. Al-Rummana, G. Shinde, and A. Al-Ahdal, "MapReduced based: a new stream cipher technique for data encryption," *Int. J. Eng. Adv. Technol.,* vol. 9, no. 5, pp. 763-769, Jun. 2020.

[6] W. L. Tai, Y. F. Chang, and W. H. Huang, "Security analyses of a data collaboration scheme with hierarchical attribute-based encryption in cloud computing," *Int. J. Netw. Secur.*, vol. 22, no. 2, pp. 212-217. Mar. 2020.

[7] L. Ju and J. B. Zhao, "Simulation of document valid data encryption based on convolutional neural network," *Comput. Simulat.*, vol. 39, no. 10, pp.440-444, Oct. 2022.

[8] L. Teng, H. Li, S. Yin, and Y. Sun, "A modified advanced encryption standard for data security," *Int. J. Netw. Secur.*, vol. 22, no. 1, pp.112-117, Jan, 2020.

[9] M. Chen, "Accounting data encryption processing based on data encryption standard algorithm," *Complexity*, vol. 2021, Jun. 2021, Art. no. 7212688.

[10] L. Harn, C. F. Hsu, Z. Xia, and Z. He, "Lightweight aggregated data encryption for wireless sensor networks (WSNS)," *IEEE Sens. Lett.*, vol. 5, no. 4, Apr. 2021, Art no. 6000704.

[11] J. Arroyo, C. P. Barbosa, M. V. Aborde, F. B. Yara, and A. Delima, "An improved image steganography through least significant bit embedding technique with data encryption and compression using Polybius cipher and Huffman coding algorithm," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp.3376-3383, May. 2020.

[12] H. Xia and W. Yang, "Security access solution of cloud services for trusted mobile terminals based on trustzone," *Int. J. Netw. Secur.,*, 22(2), 2020, pp.201-211.

[13] W. Dai, Q. Wang, Z. Wang, X. Lin, and H. Jin, "Trustzone-based secure lightweight wallet for hyperledger fabric," *J. Parallel. Distr. Com.*, vol. 149, no. 6, pp.66-75, Mar. 2021.

[14] M. M. Abdul Mounim and A. N. Mohammed, "Review on chaotic theory using DNA encoding with image encryption," *Inform. J. Appl. Mach. Electr. Electron. Comput. Sci. Commun. Syst.* vol. 2, no. 1, pp.14-19. Mar. 2021.

[15] L. Meng, S. Yin, C. Zhao, H. Li, and Y. Sun, "An improved image encryption algorithm based on chaotic mapping and discrete wavelet transform domain," *Int. J. Netw. Secur.*, vol. 22, no. 1, pp.155-160, Jan. 2020.

[16] G. Dhanya and J. Jayakumari, "Speech scrambling based on chaotic mapping and random permutation for modern mobile communication systems," *APTIKOM J. Comput. Sci. Inf. Tec.*, vol. 2, no. 1, pp.20-25, Mar. 2017.

[17] R. Rishabh and T. P. Sharma, "Device classification based data encryption for Internet of Things," *Int. J. of High Perform. Comput. Netw.*, vol. 16, no. 1, pp. 36-42, Sep. 2020.

[18] Y. X. Ma, H. S. Li, and Q. Huang, "Regression prediction model for low delay query security of information network data," *Electron. Eng. Des.*, vol. 31, no. 3, pp.155-158, Feb. 2023.

[19] H. Y. Lin and M. Y. Hsieh, "A dynamic key management and secure data transfer based on m-tree structure with multi-level security framework for internet of vehicles," *Connect. Sci.*, vol. 34, no. 1, pp.1089-1118, Dec. 2022.

[20] C. Yang and C. Li, "Design of key management protocols for internet of things," *Int. J. Netw. Secur.*, vol. 22, no. 3, pp.476-485, May. 2020.

[21] H. J. Ding, P. Pinson, Z. C. Hu, J. H. Wang, and Y. H. Song, "Optimal offering and operating strategy for a large wind-storage system as a price maker," *IEEE T. Power Syst.*, vol. 32, no. 6, pp. 4904-4913, Nov. 2017