Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.059103

ARTICLE

A Blockchain Cross-Chain Transaction Protection Scheme Based on FHE

Hongliang Tian and Zuoqing Li*

College of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China

*Corresponding Author: Zuoqing Li. Email: lizuoqing999@163.com

Received: 28 September 2024; Accepted: 26 December 2024; Published: 06 March 2025

ABSTRACT: Low data encryption efficiency and inadequate security are two issues with the current blockchain cross-chain transaction protection schemes. To address these issues, a blockchain cross-chain transaction protection scheme based on Fully Homomorphic Encryption (FHE) is proposed. In the proposed scheme, the functional relationship is established by Box-Muller, Discrete Gaussian Distribution Function (DGDF) and Uniform Random Distribution Function (URDF) are used to improve the security and efficiency of key generation. Subsequently, the data preprocessing function is introduced to perform cleaning, deduplication, and normalization operations on the transaction data of multi-key signature, and it is classified into interactive data and asset data, so as to perform different homomorphic operations in the FHE encryption stage. Ultimately, in the FHE encryption stage, homomorphic multiplication and homomorphic addition are used targeted for the interactive data and asset data, thereby reducing the computational complexity and enhancing the FHE encryption efficiency. The significance of the proposed scheme is proved by experimental results: Firstly, the multi-key generation function and its specific sampling method and transformation ensure the security and efficiency of key generation. Data preprocessing can also accelerate the FHE encryption process by eliminating invalid data and redundancy, so the FHE encryption efficiency is significantly improved. Secondly, the FHE encryption method based on discrete logarithm problem enhances the security of transaction data and can effectively resist multiple attacks. In addition, the preprocessed data also has good performance in capacity storage. The proposed scheme has significant impacts on key indicators such as encryption efficiency and security, it provides a new reference for blockchain cross-chain transaction protection technology and has an important impact on the security improvement of various cross-chain transaction data.

KEYWORDS: Blockchain; cross-chain transactions; fully homomorphic encryption (FHE)

1 Introduction

The Interledger Protocol (ILP) proposed by Adrian et al. [1] in 2012 laid the foundation for cross-chain transactions between different types of blockchains. However, due to the public transparency of the blockchain, cross-chain transaction data is still easily stolen or tampered with by malicious attackers [2,3]. The particularly famous Poly Network hacking attack led to the theft of $610 million in cryptocurrencies [4]. Similarly, the Binance Coin (BNB) incubated by the cryptocurrency exchange Binance was hacked, and two million BNBs worth approximately $566 million were stolen [5]. Multiple cross-chain transaction security incidents have caused a huge stir in the global cryptocurrency market. It not only makes users' assets on various platforms face huge risks of loss but also makes the whole industry question the security of cross-chain transactions.

The protection scheme for cross-chain transaction data involves knowledge in many fields, such as homomorphic encryption technology [6], multi-signature technology [7,8], multi-party computation [9],

and verifiable random function [10]. To solve the problems of transaction data protection schemes in cross-chain transactions, technologies such as symmetric encryption [11], asymmetric encryption [12], hash function [13], and fully homomorphic encryption [14] have been proposed respectively. However, symmetric encryption and asymmetric encryption need to decrypt the transaction data before the calculation operation, and the hash function is an irreversible conversion of the transaction data. The fully homomorphic encryption technology can perform computational operations in the encrypted state without decrypting transaction data.

Therefore, according to the advantages of fully homomorphic encryption technology, the blockchain cross-chain transaction protection scheme based on fully homomorphic encryption (FHE) is proposed. To improve security and encryption efficiency in the cross-chain transaction process, structured functions such as the FHE function, data preprocessing function, multi-key generation function, and others are written into the smart contract [15–17]. The main contributions can be summarized as follows:

1) The Discrete Gaussian Distribution Function (DGDF) is introduced to generate the private key, and the relationship with the Uniform Random Distribution Function (URDF) is established by the Box-Muller transform. The private key generated by this scheme is faster and more secure.

2) Adding data preprocessing operations, such as cleaning, deduplication, and normalization of cross-chain transaction data. It can reduce the time of the encryption stage and improve the overall efficiency.

3) The FHE algorithm can ensure the security of cross-chain transaction data during transmission, storage, and calculation. The homomorphic multiplication encryption and homomorphic addition encryption are implemented on the classified data, improving the encryption efficiency and capacity.

2 Related Works

Encryption technology combined with blockchain is widely used in medicine [18–21], the Internet of Things [22], government information sharing [23], and other fields. It shows the powerful decentralization ability of blockchain technology, thus avoiding the problem of a single point of failure. However, in a scenario where different blockchain networks need to realize the interoperability of transaction data, the original independent blockchain networks can communicate and trade with each other to achieve the free circulation of data and assets. This situation is most common in the financial field. For example, there are transaction initiators and transaction receivers for the exchange of Bitcoin and Ether coins. These users belong to different blockchain networks, so they need to use cross-chain technology with high security and fast speed for transactions.

FHE encryption technology has realized data privacy protection in many application scenarios [24,25], but at present, algorithm efficiency, security, application expansion, and other fields in fully homomorphic encryption technology are still the focus of attention. In [26], by combining FHE and SE technology, the functions of information sharing and secure search are realized, and the cloud storage security of sensitive data is improved. However, the encryption and decryption time of the index value of the FHE encryption algorithm is not evaluated. In [27], FHE is divided into polynomial operation and non-polynomial operation, and the concept of ciphertext calculation conversion is introduced into Ethereum to realize the conversion between different ciphertext calculation types. However, the efficiency of the FHE encryption phase has not improved.

Summarizing the above research on the combination of FHE encryption scheme and blockchain technology, low encryption efficiency is a common phenomenon. Based on the current research on blockchain technology and FHE technology, we apply the FHE encryption technology to blockchain cross-chain transaction protection for the first time and propose the blockchain cross-chain transaction protection

scheme based on FHE. The feasibility of the proposed scheme is verified by a large number of experiments, and the problems of low encryption efficiency and insufficient security involved in the above research are solved by introducing functions such as data preprocessing and multi-key generation.

3 Proposed Scheme

Fig. 1 shows the blockchain cross-chain transaction protection model based on FHE technology. Fig. 2 shows the sequence diagram of the cross-chain transaction protection model based on FHE technology. In the blockchain cross-chain transaction protection scheme based on FHE technology, the security protection of cross-chain transaction data is achieved by the multi-key generation function, data preprocessing function, and FHE encryption function. These functions are written into the smart contract and deployed in the blockchain network. The users' information on cross-chain transactions is processed by the multi-key generation function, which generates multiple public-private key pairs and returns them to the users. The transaction information is encrypted by the users using the private key and decrypted using the public key. The encrypted transaction information is classified into interactive data and asset data by data preprocessing. The interactive data is encrypted by homomorphic multiplication, and the asset data is encrypted by homomorphic addition. The generated transaction ciphertext data is stored in the InterPlanetary File System (IPFS), and the IPFS system generates a unique content identifier (CID). After verifying the CID number, it is packaged and uploaded to the blockchain cross-chain network. Cross-chain transactions between different blockchains are realized by gateway direct connection. In this way, the centralized institutions are eliminated, and the risk of transaction data leakage can be avoided. The receiver submits an access request to the IPFS system by the acquired public key and obtains the transaction ciphertext data after obtaining the license. The ciphertext data is decrypted by the public key to obtain the original transaction data.

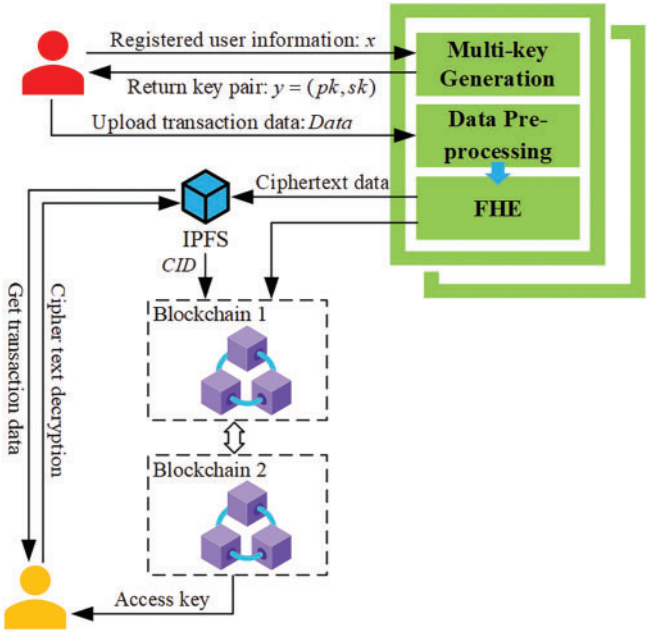

Figure 1: Blockchain cross-chain transaction protection model based on FHE technology

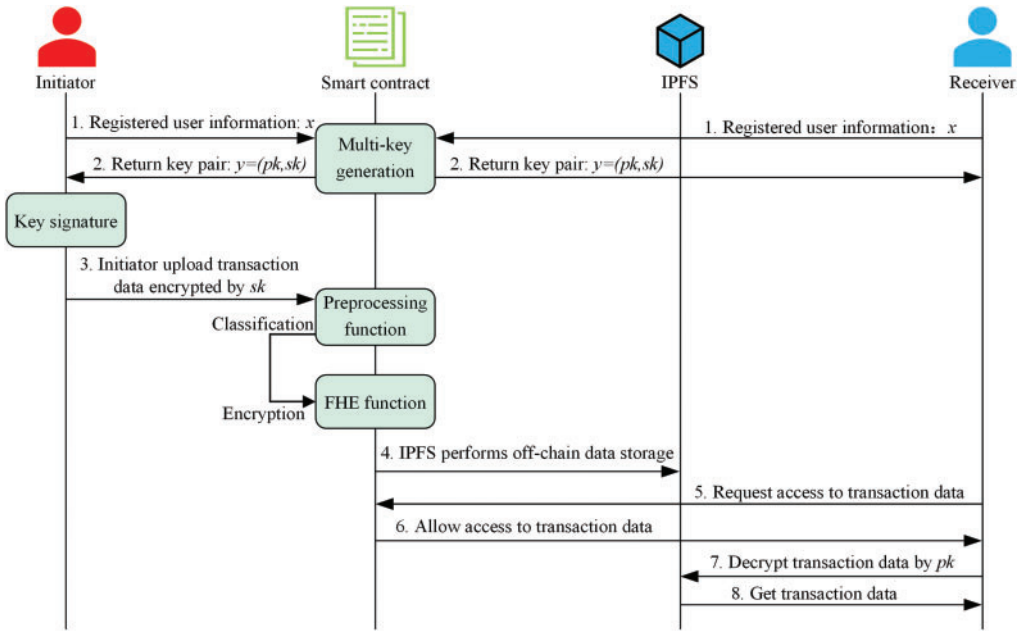

Figure 2: Sequence diagram of cross-chain transaction protection model based on FHE technology

3.1 Multi-Key Generation and Encryption

The key pairs in cross-chain transactions are usually generated by URDF. However, the attackers may guess or crack the key pairs by the statistical properties of the URDF. The generation of private keys is more uniform by introducing the DGDF. It can create more randomness and uncertainty in this way, improving the security of the whole system.

Due to the correlation between the private key *sk* and the public key *pk*, the Box-Muller transform is used to establish the relationship between DGDF and URDF as shown in Fig. 3.

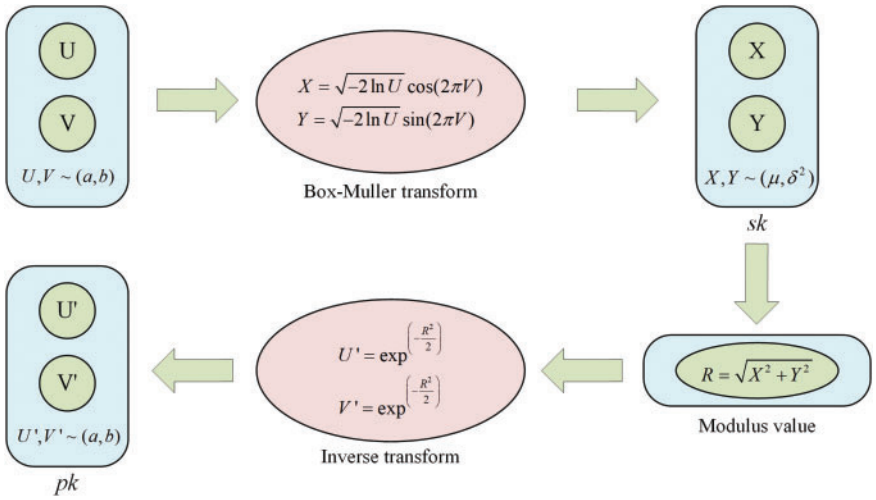

Figure 3: Box-Muller transform

According to the Box-Muller transformation, any U and V belong to the uniform random distribution variables, which can be converted into discrete Gaussian distribution variables by Eq. (1). Where the mean value is 0, the standard deviation is 1, X and Y are random variables that obey the DGDF.

$$\begin{aligned} X &= \sqrt{-2\ln U}\cos(2\pi V) \\ Y &= \sqrt{-2\ln U}\sin(2\pi V) \end{aligned} \tag{1}$$

The mapping random variable of URDF is generated by the DGDF, and the inverse transformation of the Box-Muller transformation is required. Therefore, continue to calculate the modulus values of X and Y, as shown in Eq. (2).

$$R = \sqrt{X^2 + Y^2} \tag{2}$$

Due to the modulus R having a linear relationship with U and V, U and V can be recovered from R, such as U' and V' in Eq. (3), the Box-Muller transformation is a one-way transformation. U' and V' are different from U and V, but they still obey the URDF.

$$\begin{aligned} U' &= \exp\left(-\frac{R^2}{2}\right) \\ V' &= \exp\left(-\frac{R^2}{2}\right) \end{aligned} \tag{3}$$

Finally, the private key is obtained by X and Y, and the public key is obtained by U' and V', that is, the key pairs $y = (pk, sk)$ required for cross-chain transactions.

3.1.1 Multi-Key Generation

As shown in Fig. 4, a total of n users participated in the cross-chain transactions. The users upload their personal information x to the blockchain node for registration, and the multi-key generation function in the smart contract returns the key pair $y = (pk, sk)$. Users can only obtain the output of their personal information by secure multi-party computation, but they cannot obtain the input and output information of other users.

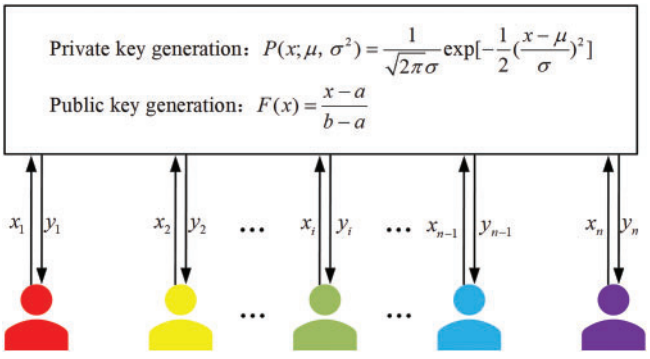

Figure 4: Multi-key generation

Private key generation: This *sk* is obtained by sampling the DGDF once, and *sk* is used to sign and encrypt the cross-chain transaction data. The DGDF is shown in Eq. (4).

$$P\left(x;\mu,\sigma^2\right)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{4}$$

In (4), μ is the mean value, σ is the standard deviation, $\mu, \sigma \in R$, and $\sigma > 0$. For any $x \in Z$, Eq. (4) holds. The Alias Method is used to sample the DGDF, so that each random number x corresponds to a *sk*, and the length is set to 2048 bits. The specific steps are shown in Fig. 5. The implementation process of private key generation is shown in Algorithm 1.

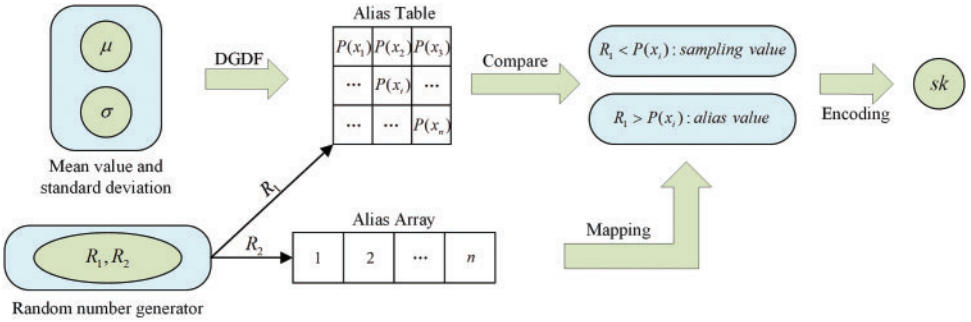

Figure 5: Private key generation

1) Calculating the probability distribution: According to the mean value μ and standard deviation σ, the probability of each random number x may be calculated.

2) Construction of alias table and alias array: The probability distribution of the DGDF is normalized so that the sum of all probabilities is 1. According to the Alias Method, the probability distribution is divided into two parts, the Alias Table, and the Alias Array. The Alias Table stores the received probability and the alias value for each x possible value, and the Alias Array records the index number of the alias value for each *sk*.

3) Generate random numbers: Two random numbers are generated by the random number generator, one is used to select the alias value index number, and the other is used to determine whether to accept the index number.

4) Sampling: According to the generated random number, the receiving probability and the alias value index number are found in the Alias Table. If the random number is less than the receiving probability, the sampling value is used as the private key. Otherwise, the alias value of the sampling value is sampled as the private key.

5) Private key generation: The value is mapped to the corresponding position in the key space, and the sampled value is encoded as the *sk*.

6) Verify the private key: Verify the attributes of the *sk*, including statistical characteristics, key length, and other attributes, to ensure that *sk* meets the requirements of specific applications.

Algorithm 1: Private-key generation

Input: x, μ, σ
Output: sk

```
function  Discrete-Gaussian(x, μ, σ)
          pre_sk==Discrete-Gaussian(x, μ, σ)
if        pre_sk==true  then
          sk==Alias-Method(pre_sk)
return    sk
else
return    pre_sk
end if
return    sk
end function
```

Public key generation: The URDF is sampled once to generate pk, and pk is transmitted to the receiver to decrypt the transaction data. The URDF is shown in Eq. (5).

$$F(x) = \frac{x-a}{b-a} \tag{5}$$

In (5), a, b is the interval endpoint value, $x \in R$, and $a < x < b$. The pk is generated by sampling the URDF, and the length is also set to 2048 bits to enhance the randomness of the pk. The specific steps are shown in Fig. 6. The implementation process of public key generation is shown in Algorithm 2.

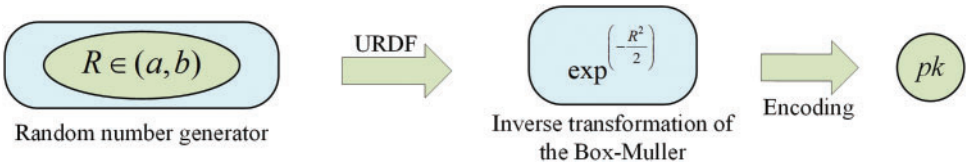

Figure 6: Public key generation

1) Determine the distribution parameters: Determine the parameter range (a, b) of the URDF to determine the value range of the pk.

2) Generate random numbers: Use the random number generator to generate a series of random numbers and use these random numbers as samples of the URDF.

3) Public key generation: The generated random number is mapped to the key space and converted to the pk by the mapping function.

4) Hash function: To ensure security, pk is hashed by the hash function to increase its anti-collision and irreversibility.

5) Verifying the public key: Verifying the attributes of the pk encrypted by the hash function to ensure that pk meets the requirements of specific applications.

Algorithm 2: Public-key generation

Input: *x*

Output: *pk*

```
function  Uniformly-Rand(x)
          pre_pk==Uniformly-Rand(x)
if        pre_pk==true  then
          pk==Rand(pre_pk)
return    pk
else
return    pre_pk
end if
return    pk
end function
```

3.1.2 Multi-Key Encryption

In the blockchain cross-chain transaction network, the multi-key scheme is used to encrypt the cross-chain transaction data. The implementation process of multi-key encryption is shown in Algorithm 3.

In Fig. 7, if Alice wants to send a cross-chain transaction *m* to Bob, the multi-key encryption steps are as follows:

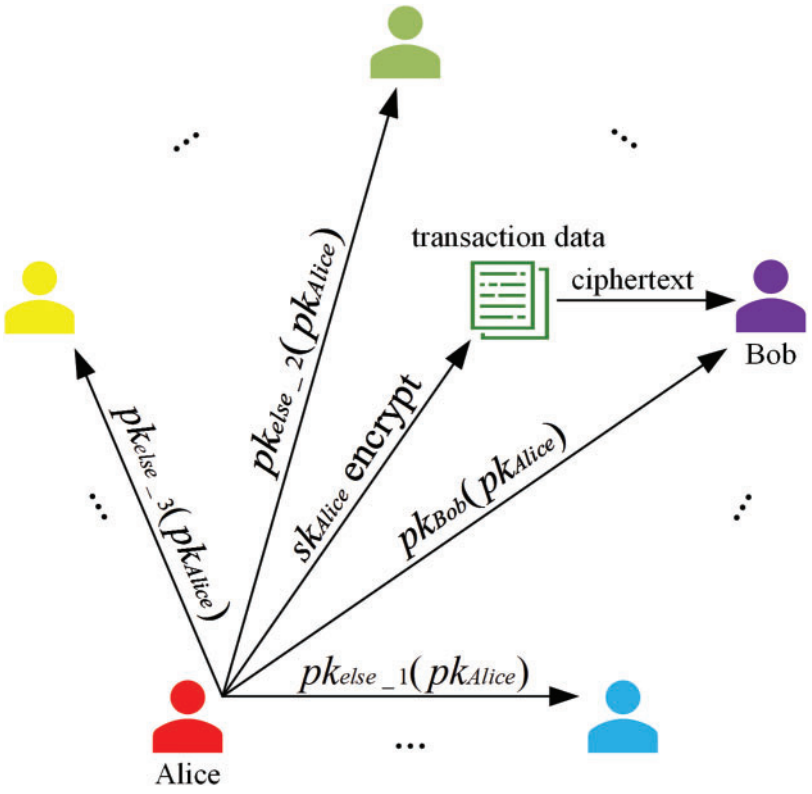

Figure 7: Multi-key encryption

1) Alice encrypts the transaction data m by sk_{Alice} and only pk_{Alice} can decrypt the encrypted data.

2) Alice uses Bob's pk_{Bob} to encrypt pk_{Alice} and generate the encrypted multi-key $pk_{Bob}(pk_{Alice})$ so that Bob can decrypt pk_{Alice} by his sk_{Bob}.

3) Alice sends m and $pk_{Bob}(pk_{Alice})$ encrypted by the sk_{Alice} to Bob.

4) Bob decrypts $pk_{Bob}(pk_{Alice})$ to obtain pk_{Alice} by his sk_{Bob} and then uses pk_{Alice} to decrypt m to obtain transaction data.

5) In the meantime, Alice also sends the encrypted $pk_{else}(pk_{Alice})$ to other users in the blockchain cross-chain network. Other users do not need to decrypt the transaction data but only need to act as witnesses of cross-chain transactions to verify the authenticity of the transaction data.

Algorithm 3: Multi-key encryption

Input: *Transactions,* sk_1, sk_2, pk_1
Output: *Ciphertext, key*

```
function Multi-key encryption(Transactions, sk1)
    Ciphertext==sk1(Transactions)
    if Ciphertext==true then
        key==sk2(pk1)
        return key
    else
        return Ciphertext
    end if
    return Ciphertext, key
end function
```

3.2 Data Preprocessing

Fig. 8 is the data preprocessing process. The FHE encryption phase includes homomorphic multiplication and homomorphic addition. To improve the efficiency of the two calculations, cross-chain transaction data needs to perform data preprocessing operations before uploading them to the local blockchain. Since homomorphic multiplication is suitable for processing data with diverse data types and frequent interactions in the transaction process, such as transaction negotiation information, transaction requests, and transaction verification, such data is defined as interactive data. Homomorphic multiplication can directly multiply the encrypted data, and the result of the operation is still the encrypted value, which can save the time of the FHE decryption stage. The homomorphic addition operation is suitable for processing data such as digital currency, which is defined as asset data. Homomorphic addition can perform addition calculations on two or more encrypted asset data in an encrypted state. The result of decryption is the sum of the original asset data, which is very convenient for the calculation of asset data in the FHE encryption and decryption phase.

The data preprocessing process is divided into three stages: data cleaning, data deduplication, and data normalization, the implementation process is shown in Algorithm 4. The three stages have a sequential relationship, and the following is the specific implementation process.

1) Data cleaning: The invalid data doped in cross-chain transaction data is deleted by data cleaning. The asset data is defined as the float type, where the values of 0 and null are judged, and deleted if they appear. Asset data with a single transaction amount exceeding 2000 will be set as a security warning.

2) Data deduplication: The repeated asset data values are deleted to avoid misjudgment as a double-flower attack or replay attack while reducing the complexity of asset data analysis. The subsequent submitted asset data is compared with the original data. When repeated asset data appear, the original data is retained and the subsequent submitted data is deleted. To ensure the efficiency of the data deduplication process, the total number of asset data is limited to 100.

3) Data normalization: The amount of transaction data is limited to [0, 2000] to ensure the security of asset data. The asset data is scaled to ensure that the results are within the range of [0, 2], which helps to eliminate the influence of unit dimensions among the data and enhances data comparability.

Figure 8: Data preprocessing

Algorithm 4: Data preprocessing

Input: *Transactions*

Output: *Interactive, Asset*

```
function Cleaning(Transactions), Deduplication(Asset), Normalization(Asset)
    Cleaned_data==Cleaning(Transactions)
    if Cleaned_data==true then
        Deduplicated_data==Deduplication(Cleaned_data)
        if Deduplicated_data==true then
            Normalized_data==Normalization(Deduplicated_data)
            return Interactive, Asset
        else
            return Deduplicated_data
        end if
    else
        return Cleaned_data
    end if
    return Interactive, Asset
end function
```

3.3 FHE Encryption

Cross-chain transaction information is classified into interactive data and asset data, which will be encrypted separately by the FHE function in the smart contract. The 2048-bit key is used to encode the original data, which can more effectively resist the brute force mechanism of the quantum computer. The original cross-chain transaction information is encoded to generate plaintext data. The plaintext data is then encrypted into ciphertext data by FHE. The final ciphertext data is stored in the IPFS system. Fig. 9 shows the FHE classification encryption process and the implementation process of the FHE function is shown in Algorithm 5.

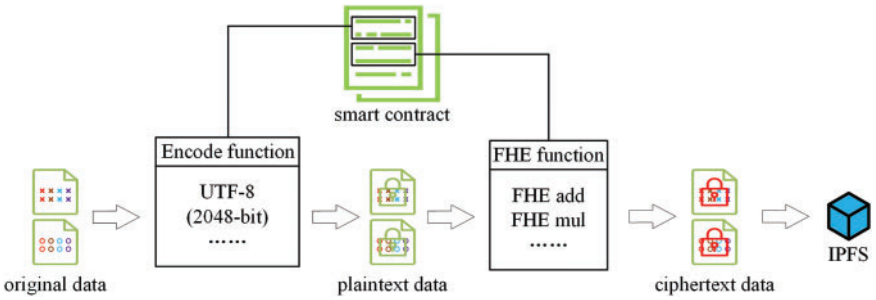

Figure 9: FHE classification encryption

1) Homomorphic multiplication encryption: In Fig. 10, the cross-chain interactive data is encoded using UTF-8 to convert the string data into a 2048-bit digital format. For encrypted strings, homomorphic multiplication can be performed directly. Due to the properties of homomorphic multiplication encryption, the result of two encrypted strings is still an encrypted value. As shown in Eq. (6), $E(\cdot)$ is the encryption function, a and b are the transaction data.

$$E(a) * E(b) = E(a * b) \tag{6}$$

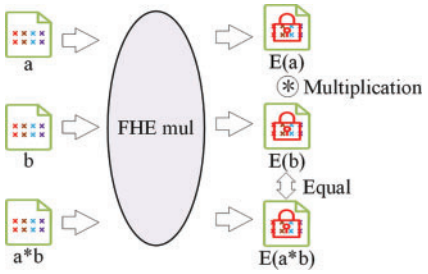

Figure 10: Homomorphic multiplication

2) Homomorphic addition encryption: In Fig. 11, the key pairs are used to encrypt the cross-chain asset data to obtain the encrypted asset data. In the encrypted state, homomorphic addition operations can be performed on two or multiple encrypted asset data, and the result after decryption will still be the sum of the original asset data. As shown in Eq. (7), $E(\cdot)$ is the encryption function, a and b are the transaction data.

$$E(a) + E(b) = E(a + b) \tag{7}$$

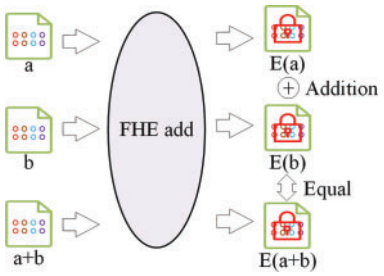

Figure 11: Homomorphic addition

Algorithm 5: FHE algorithm

Input: *Interactive, Asset*

Output: *En-Interactive, En-Asset*

1. **function** *Mul-FHE(Interactive), Add-FHE(Asset)*
2. *Pre-Interactive==Mul-FHE(Interactive)*
3. *Pre-Asset==Add-FHE(Asset)*
4. **if** *Pre-Interactive==true & Pre-Asset==true* **then**
5. *En-Interactive==Mul-FHE(Pre-Interactive)*
6. *En-Asset==Add-FHE(Pre-Asset)*
7. **return** *En-Interactive, En-Asset*
8. **else**
9. **return** *Pre-Interactive, Pre-Asset*
10. **end if**
11. **return** *En-Interactive, En-Asset*
12. **end function**

4 Experiment and Analysis

The experiment is implemented on Ubuntu 22.04.2 LTS 64-bit, CPU dual-core, the memory is 2 GB, and the hard disk is 100 GB. The comprehensive tests include multi-key generation, FHE encryption and decryption time, security analysis, and cross-chain transaction performance.

4.1 Multi-Key Generation

During the multi-key generation stage, the time and capacity consumed for generating keys in cross-chain transactions are tested. The results are illustrated in Figs. 12 and 13.

The time consumed by different key generation methods is illustrated in Fig. 12. When the key length gradually increases, the time fluctuation of multi-signature is the largest [28]. The time of the proposed scheme remains stable, reaching 0.005953 s when the length is 2048 bits. This is because literature [28] uses the off-chain multi-signature technology to aggregate all transactions in the block and generates the public key through one-way address aggregation and KMS aggregation, resulting in a long consumption time. There is no key aggregation step in the proposed scheme. At the same time, the DGDF can accelerate the speed of key generation by parallel processing, making the process of generating private keys more efficient. According to the results, the proposed scheme can quickly provide key pairs with higher security within the time allowed for cross-chain transaction execution.

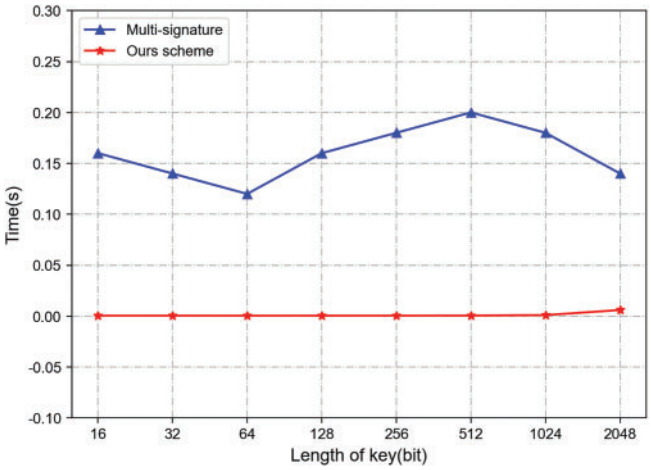

Figure 12: Key generation time

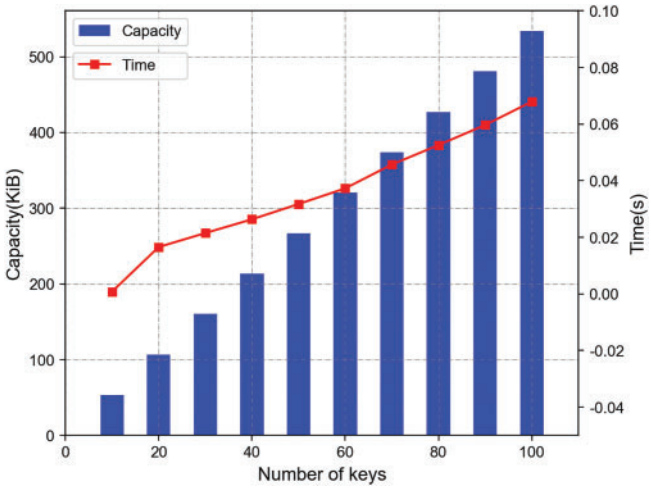

Figure 13: Multi-key generation capacity and time

Fig. 13 shows the capacity and time consumption of the 2048-bit key pairs, which is proportional to the number of key pairs. When the key pairs are 10, the proposed scheme takes 0.000812648 s. When the key pairs are 100, the proposed scheme takes 0.067942201 s. According to the experimental results of multi-key generation, for every 10 key pairs added, the time consumption of the proposed scheme increases by 0.007458839 s. Moreover, a key pair only occupies the capacity of 5.3379 KiB and will be stored in the IPFS system for backup, which can reduce the blockchain storage pressure while ensuring security.

4.2 FHE Encryption and Decryption

The FHE scheme is compared with the existing encryption schemes such as Elliptic Curve Cryptography (ECC), Rivest-Shamir-Adleman (RSA), and Paillier [29]. The encryption time and decryption time are analyzed, and the results are illustrated in Figs. 14 and 15.

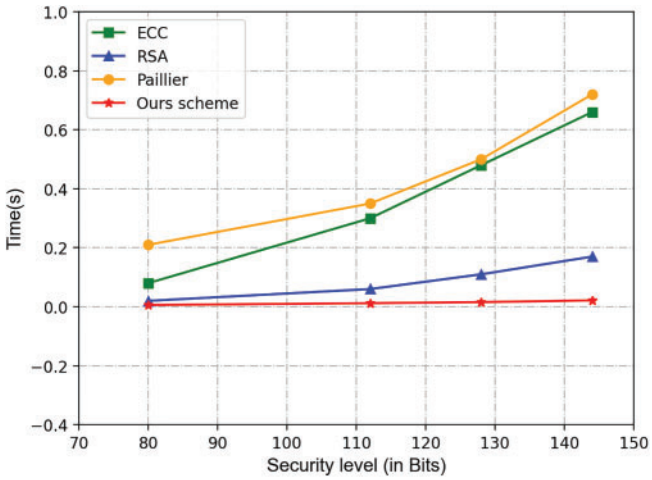

Figure 14: FHE encryption time

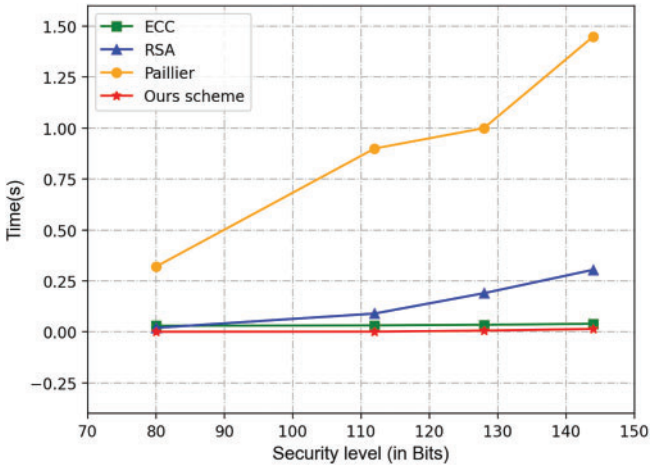

Figure 15: FHE decryption time

It can be seen from Fig. 14 that the encryption time curve of the proposed scheme is stable at 0.01570812 s. The time curves of ECC, RSA, and Paillier schemes show an upward trend, among which the Paillier scheme takes the longest time. The specific reason is the computational complexity. ECC, RSA, Paillier, and other encryption schemes involve some long calculation steps or complex algorithm logic in the encryption process, which makes the encryption time longer. With the increase in the amount of data, more computing resources and time are needed to complete the encryption operation. The FHE scheme avoids these problems through reasonable data preprocessing and efficient classified encryption calculation, thus performing better in encryption time, such as the homomorphic addition encryption of asset data does not require carry operation, and the response speed is quicker. When the amount of data is large, the effect of the data preprocessing stage is more obvious, which will save more time.

In Fig. 15, the FHE scheme still has excellent performance in the decryption stage. The decryption time of the Paillier scheme increases the most, while the ECC and RSA schemes are relatively stable. The proposed FHE scheme has the shortest decryption time, which remains stable at 0.006230373 s. Due to the existence of

data preprocessing, the algorithms and processes in the decryption phase can quickly and accurately restore the transaction data, unlike other schemes that may have some situations that are not suitable for blockchain transaction data. Avoiding data confusion and unnecessary calculations can improve decryption efficiency. Therefore, the FHE scheme is designed according to the characteristics of blockchain cross-chain transaction data, which can better adapt to the encryption and decryption requirements of transaction data.

4.3 Security Analysis

The security of the proposed FHE scheme is proved by the discrete logarithm problem. The results show that the scheme can meet the security requirements of blockchain cross-chain transactions. The discrete logarithm problem is shown in Eq. (8).

$$g^x = y \,(\mathrm{mod}\quad z) \tag{8}$$

In (8), g is the primitive root, x is the exponent, y is the integer, and z is the prime number. The calculation process of the discrete logarithm problem is complex, which means it has a wide range of applications in cryptography.

1) Define the model of security: The private key cannot be calculated from the public key by the attacker, and conversely, the public key cannot be calculated from the private key.

2) Establishing the model of difficulty: The security model of the FHE scheme is based on the discrete logarithm problem. The relevant definitions and symbols in the model are shown in Table 1. The $sk = (n, d)$ is generated by the DGDF, and the $pk = (n, e)$ is generated by the URDF, where n is the product of two large prime numbers p and q, d and e are integers that are relatively prime with $(p-1)(q-1)$. sk is calculated by the FHE encryption function to solve pk, which is the same as the discrete logarithm problem. On the contrary, it is also difficult to solve sk by calculating pk.

Table 1: Model name and symbol

Name	Symbol
Discrete logarithm	$g^x = y \,(\mathrm{mod}\, z)$
Homomorphic multiplication encryption	$FHE\_Mul\,()$
Homomorphic addition encryption	$FHE\_Add\,()$
Interactive data	m_1
Asset data	m_2
Encrypted data	c

3) Process of proof: It is assumed that the attackers can decrypt the cross-chain transaction data c encrypted by sk, that is, $c = m \,(\mathrm{mod}\, sk)$. The cross-chain transaction data c is fully homomorphically encrypted by the FHE scheme, which is as follows:

Homomorphic multiplication encryption: $c\_mul = FHE\_Mul\,(c_1)\,(\mathrm{mod}\, sk)$

Homomorphic addition encryption: $c\_add = FHE\_Add\,(c_2)\,(\mathrm{mod}\, sk)$

Assume that the attackers can recover the transaction data m by calculating c, that is, $m = c \,(\mathrm{mod}\quad pk)$. If the previous step is established, the attackers can effectively calculate c, then c_mul and c_add can also be calculated to obtain interactive data m_1 and asset data m_2.

However, this contradicts the difficulty of the discrete logarithm problem. Therefore, the assumption is not true, and attackers cannot obtain useful information from the ciphertext.

4) Quantum attacks analysis: Assume that the quantum computer attacks the blockchain cross-chain transaction network protected by FHE encryption, as shown in Fig. 16.

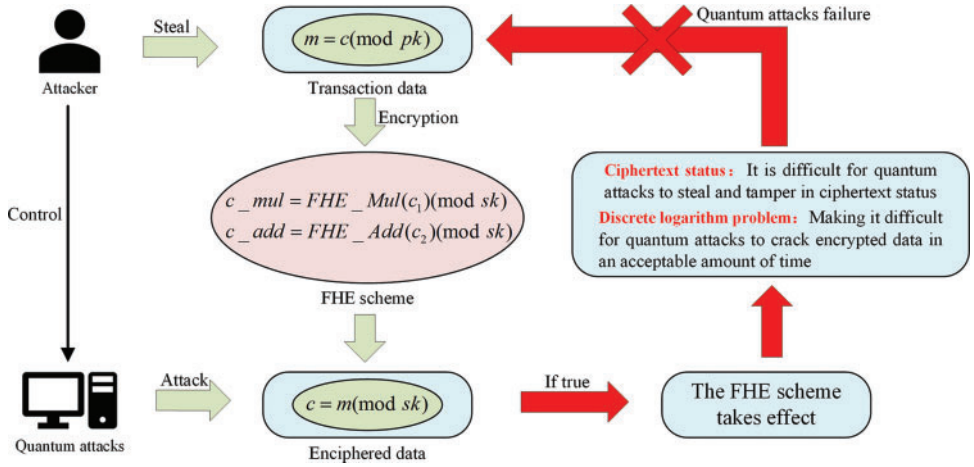

Figure 16: Quantum attacks diagram

The FHE scheme mainly relies on the following two points to resist quantum attacks:

(1) In solving such mathematical problems, quantum computers do not have significant advantages over traditional computers, that is, they cannot effectively crack the FHE scheme in an acceptable time, thus ensuring the security of encrypted data.

(2) The characteristic of the FHE scheme is to allow direct calculation of encrypted data without decryption. This means that the data remains encrypted throughout the processing, and there is no risk of exposing plaintext due to decryption operations. Quantum attacks usually need to obtain the plaintext of the data to carry out effective operations such as stealing and tampering, while the FHE scheme makes it impossible for the attacker to easily obtain the plaintext information, thus resisting quantum attacks.

5) Conclusion: The proposed cross-chain transaction protection scheme based on FHE has higher security and can improve the ability of cross-chain transaction systems to resist attacks. The FHE scheme is compared with the existing blockchain transaction protection scheme, and the results are shown in Table 2.

Table 2: Encryption scheme comparison

Scheme	Centralization	Encryption speed	Decryption speed	Security
ECC	Middle	Middle	Middle	Strong
RSA	Middle	Middle	Middle	Strong
Paillier	Low	Slow	Slow	Weak
FHE	Low	Fast	Fast	Strong

4.4 Cross-Chain Transaction Performance

After adding multi-key generation, data preprocessing, and fully homomorphic encryption, the performance of cross-chain transactions is tested again. Including cross-chain transaction time, throughput, and delay, the results are shown in Figs. 17 and 18.

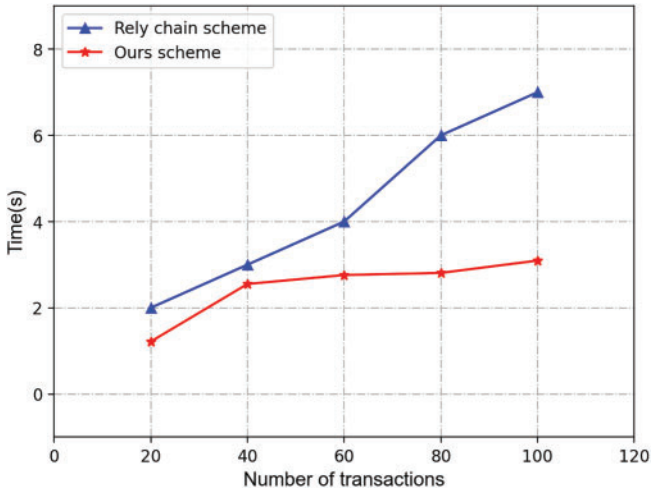

Figure 17: Cross-chain transaction time

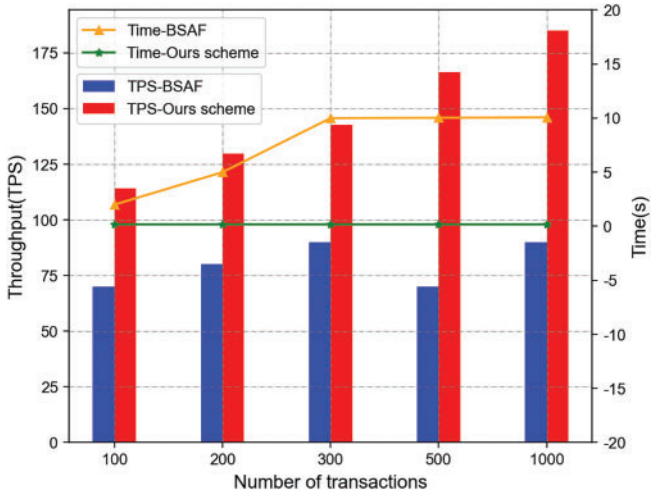

Figure 18: Throughput and delay

The comparison of cross-chain transaction time is illustrated in Fig. 17. With the increase of cross-chain transaction volume, the time of both schemes is increasing. In the relay chain scheme [30], for every 20 transactions, the cross-chain time increases by about 1 s, and the time increase is significant. The proposed scheme has a slight increase in time. The reason is that the proposed scheme is optimized in the multi-key generation, data preprocessing, and FHE encryption and decryption stages, which shortens the time of the whole cross-chain transaction and effectively reduces the complexity.

The throughput and delay of blockchain cross-chain transactions are illustrated in Fig. 18. The results show that the proposed scheme has higher throughput and lower delay compared with the blockchain-based secure access framework (BSAF) [31]. In Fig. 17, it can be concluded that the time of the whole cross-chain transaction process is shortened. The proposed scheme can complete the same transaction volume in a shorter time, improving throughput, delay, and other performance metrics.

5 Conclusions

The cross-chain transaction encryption scheme performance is enhanced by multi-key generation, data preprocessing, and fully homomorphic encryption. The proposed FHE encryption scheme can be applied to various cross-chain transaction scenarios. In financial transactions, the FHE encryption scheme can make the data computable in the encrypted state, effectively protect the security of financial transaction data, and is suitable for various financial transaction activities. In supply chain management, the FHE scheme can ensure that data such as contract terms negotiation and goods price calculation are processed safely in encrypted state, and enhance the security and efficiency of supply chain data management. In IoT data sharing, the FHE scheme can protect data such as device control instructions and data usage statistics, so that data can be calculated when encrypted, effectively protect IoT data privacy, and promote secure and efficient data sharing between IoT devices. In addition, to achieve the larger capacity of data encryption, the next step will supplement the high-frequency cross-chain transaction scenario and study the encryption scalability and data expansion issues in the fully homomorphic encryption algorithm.

Acknowledgement: We would like to acknowledge the editors and reviewers for their comments.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Hongliang Tian; data collection, analysis and interpretation of results and draft manuscript preparation: Zuoqing Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Z. Q. Li, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Adrian HB, Stefan T. Interledger: creating a standard for payments. In: Proceedings of the 25th International Conference Companion on World Wide Web; 2016; Montreal, QC, Canada. p. 281–2. doi:10.1145/2872518.2889307
2. Nicolas K, Wang Y, Giakos GC, Wei B, Shen H. Blockchain system defensive overview for double-spend and selfish mining attacks: a systematic approach. IEEE Access. 2021;9:3838–57. doi:10.1109/ACCESS.2020.3047365.
3. Aman K, Nitin J. An approach of blockchain to enhance supply chain transparency. In: 2023 6th International Conference on Information Systems and Computer Networks (ISCON); 2023; Mathura, India. p. 1–5. doi:10.1109/ISCON57294.2023.10112036
4. Tian HB, Ye W. Risk management policies and analysis of cross-chain digital assets. J Front Comput Sci Technol. 2023;17(9):2219–28.
5. David VT, Antonio B, Tomaso A. FTX's down fall and Binance's consolidation: the fragility of centralised digital finance. Phys A: Stat Mech Appl. 2023 Sep;625:129044. doi:10.1016/j.physa.2023.129044.

6. Zhang YS, Jiang JJ, Dong XW, Wang LM, Xiang Y. BeDCV: blockchain-enabled decentralized consistency verification for cross-chain calculation. IEEE Trans Cloud Comput. 2023 Jul;11(3):2273–84. doi:10.1109/TCC.2022.3196937.
7. Uganya G, Bommi RM, Muthu PK, Vijayaraj N. Revised elliptic curve cryptography multi-signature scheme (RECC-MSS) for enhancing security in electronic health record (EHR) system. J Intell Fuzzy Syst. 2023;45(6):11993–2012.
8. Wang Z, Li J, Chen XB, Li CY. A secure cross-chain transaction model based on quantum multi-signature. Quant Inf Process. 2022;21(8):279.
9. Ma YH, Zhang L, Wu XY, Li M. Multi-party cross-chain transaction scheme based on distributed key generation and attribute-based encryption. Jisuanji Yanjiu Yu Fazhan/Comput Res Develop. 2023;60(11):2534–44.
10. Shu FX, Lei K. Vger: a VRF based cross-chain mechanism for blockchains. J Phys: Conf Ser. 2021;1780:012038. doi:10.1088/1742-6596/1780/1/012038.
11. Li HG, Tian HB, Zhang FG, He JJ. Blockchain-based searchable symmetric encryption scheme. Comput Electr Eng. 2019;73:32–45.
12. Pillai BG, Dayanand Lal N. Blockchain-based searchable asymmetric encryption scheme in cloud environment. In: 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC); 2023; Dharwad, India. p. 1–6. doi:10.1109/ICAISC58445.2023.10201090.
13. Sharma D, Saxena M. Different cryptographic hash functions for security in the blockchain. In: 2023 International Conference on Data Science and Network Security (ICDSNS); 2023; Tiptur, India. p. 1–6. doi:10.1109/ICDSNS58469.2023.10245326.
14. Yang YT, Liu DL, Liu PH, Zeng P, Xiao S. BFVBlockchainvoting: blockchain-based electronic voting systems with BFV full homomorphic encryption. Tongxin Xuebao/J Commun. 2022;43(9):100–11. doi:10.11959/j.issn.1000-436x.2022172.
15. He DJ, Wu R, Li XJ, Chan S, Guizani M. Detection of vulnerabilities of blockchain smart contracts. IEEE Internet Things J. 2023;10(14):12178–85. doi:10.1109/JIOT.2023.3241544.
16. Rym K, Wafa N, Narjes BR. A distributed multi-key generation protocol with a new complaint management strategy. Lect Notes Bus Inf Process. 2023;464:150–64. doi:10.1007/978-3-031-30694-5.
17. Wang T. Application of blockchain-based data pre-processing algorithm in motion analysis system. Int J Glob Energy Issues. 2023;45(6):503–23. doi:10.1504/IJGEI.2023.133805.
18. Mamta, Gupta BB, Li KC, Leung VCM, Psannis KE, Yamaguchi S. Blockchain-assisted secure fine-grained searchable encryption for a cloud-based healthcare cyber-physical system. IEEE/CAA J Automatica Sinica. 2021;8:1877–90. doi:10.1109/JAS.2021.1004003.
19. Nguyen GN, Vieta NHL, Elhoseny M, Shankar K, Gupta BB, Abd El-Latif AA. Secure blockchain enabled Cyber-physical systems in healthcare using deep belief network with ResNet model. J Parallel Distr Comput. 2021;153:150–60. doi:10.1016/j.jpdc.2021.03.011.
20. Wang BW, Wang N, Zhang YX, Xu ZH, Zhang JH. Deletion and recovery scheme of electronic health records based on medical certificate blockchain. Comput Mater Contin. 2023;76(1):849–59. doi:10.32604/cmc.2023.039749.
21. Kumar R, Kumar J, Khan AA, Zakria, Ali H, Bernard CM, et al. Blockchain and homomorphic encryption based privacy-preserving model aggregation for medical images. Comput Med Imaging Graph. 2022;102(5):102139. doi:10.1016/j.compmedimag.2022.102139.
22. Yu CY, Mei NS, Du C, Luo HT, Lian Q. IoT data sharing scheme based on blockchain and homomorphic encryption. In: 2023 Fifth International Conference on Blockchain Computing and Applications (BCCA); 2023; Kuwait, Kuwait. p. 554–60. doi:10.1109/BCCA58897.2023.10338934.
23. Zhang L, Mao JR, An YT, Zhang TS, Ma JX, Feng C, et al. A systematic review of blockchain technology for government information sharing. Comput Mater Contin. 2022;74(1):1161–81. doi:10.32604/cmc.2023.032452.
24. Ma ZF, Wang JY, Gai KK, Duan PF, Zhang YQ, Luo SS. Fully homomorphic encryption-based privacy-preserving scheme for cross edge blockchain network. J Syst Archit. 2023;134:102782. doi:10.1016/j.sysarc.2022.102782.

25. Sivaranjani R, Patruni MR, Srinivas J, Ashok KD, Sajjad SJ, Youngho P. Privacy-preserving electronic medical record sharing for iot-enabled healthcare system using fully homomorphic encryption, IOTA, and masked authenticated messaging. IEEE Trans Ind Inform. 2024;20(9):10802–13. doi:10.1109/TII.2024.3397343.
26. Hamsanandhini S, Balasubramanie P. IoT data encryption and phrase search-based efficient processing using a Fully Homomorphic-based SE (FHSE) scheme. Pervasive Mob Comput. 2024;103:101952. doi:10.1016/j.pmcj.2024.101952.
27. Wu XH, Wang J, Zhang TB. Integrating fully homomorphic encryption to enhance the security of blockchain applications. Future Gener Comput Syst. 2024;161:467–77. doi:10.1016/j.future.2024.07.015.
28. Li XL, Ma ZF, Luo SS. Blockchain-oriented privacy protection with online and offline verification in cross-chain system. In: 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS); 2022; Huaihua, China. p. 177–81. doi:10.1109/ICBCTIS55569.2022.00048.
29. Yadav AK. Significance of elliptic curve cryptography in blockchain IoT with comparative analysis of RSA algorithm. In: 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS); 2021; Greater Noida, India. p. 256–62. doi:10.1109/ICCCIS51004.2021.9397166.
30. Wu CH, Wang JQ, Xiong HL, Yi WL, Zhao YD. A secure cross-chain mechanism based on relay chain and smart contract encryption scheme. In: 2023 11th International Conference on Information Systems and Computing Technology (ISCTech); 2023; Qingdao, China. p. 87–91. doi:10.1109/ISCTech60480.2023.00023
31. Duan L, Xu WY, Ni W, Wang W. BSAF: a blockchain-based secure access framework with privacy protection for cloud-device service collaborations. J Syst Archit. 2023;40:102897. doi:10.1016/j.sysarc.2023.102897.