Electronics, Communications and Networks
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA241380

A Fully Homomorphic Encryption Scheme Based on Fibonacci Recursive Matrix

Yiqin BAO[a1], Wenbin XU[b], Yulu BAO[c]
[a] *College of Information Engineering of Nanjing XiaoZhuang University, China*
[b] *Jiangsu United Vocational and Technical College Suzhou Branch, China*
[c] *Nanjing Shuangyun Intelligent Data Technology Co., Ltd, China*

Abstract. Homomorphic encryption is widely used in fields such as data aggregation, secure multi-party computation, and federated learning, and is one of the main technologies for addressing data privacy protection. However, existing homomorphic encryption schemes require noise reduction operations to be performed before the noise level of the ciphertext increases to a certain scale, as the noise level of the ciphertext increases with computation. To address these issues, this paper uses the characteristics of third-order Fibonacci to design a homomorphic encryption scheme based on Fibonacci matrix (FM-HE), which solves the noise problem of ciphertext. The experimental results show that under the same environment, the proposed homomorphic encryption scheme (FM-HE) significantly reduces the decryption time compared to typical homomorphic encryption algorithms such as Paillier and CKKS, and it also has fault tolerance performance.

Keywords. Homomorphic encryption; privacy protection; Fibonacci;paillier; CKKS

1. Introduction

With the continuous development of big data and cloud computing technology, a large number of companies and institutions are able to build their own cloud computing platforms and provide users with services such as massive data storage, complex computing, and prediction. The user side sends its own data to the server side, and the server side completes the processing, calculation, prediction and other work of the data according to the corresponding algorithm or model, and sends the corresponding results back to the user side [1]. If the user side provides private data when applying for services from the server side, the server side may attempt to collect this data; On the other hand, the user side may also learn the server-side model, training set, and other information from the calculation results returned by the server-side Therefore, the privacy and data security of both parties have been threatened [2].

Because the leakage of user data may pose a significant threat. In this case, homomorphic encryption (HE) has become the main technology for reducing data leakage and addressing data privacy protection. Wu et al. [3] studied "Federated learning for network attack detection using attention-based graph neural networks", Zhou et al. [4] designed "Latent Vector Optimization-Based Generative Image

[1] Corresponding author: Yiqin BAO, email: 392335241@qq.com

Steganography for Consumer Electronic Applications". In response to the security issue caused by the increase in ciphertext noise during network transmission, this paper proposes a new fully homomorphic encryption scheme based on Fibonacci technology. Through this scheme, Diffie Hellman and Fibonacci transformations are adopted to enhance security and ensure the safety and reliability of the data transmission process.

The main contributions of this article are as follows:

1) Compare and summarize relevant encryption algorithms.

2) A fully homomorphic encryption scheme based on Fibonacci matrix is proposed by combining Diffie-Hellman and Fibonacci transform.

3) Comparing FH-HE with Paillier and CKKS, demonstrating the safety and fault tolerance of FH-HE.

The rest of the paper is organized as follows. The second part introduces the research on related technologies. The third part proposes a homomorphic encryption scheme based on Fibonacci matrix (FM-HE). The fourth part compares and analyzes the experimental results. The fifth part summarizes the conclusion.

2. Related technology

2.1 Related encryption algorithms

The commonly used encryption algorithms currently include hash algorithms (such as MD5, SHA family, Hmac), symmetric encryption algorithms (such as AES), asymmetric encryption algorithms (RSA), Diffie Hellman key negotiation algorithm, elliptical encryption algorithm (ECC), homomorphic encryption, etc. These algorithms have their own characteristics and are suitable for different scenarios:

1) Hash algorithm: Forward fast, irreversible, meaning it is difficult to decrypt plaintext after encryption. Often used for data encryption and data verification to prevent information from being modified.

2) Symmetric encryption algorithm: AES is a commonly used symmetric encryption algorithm, characterized by using the same key for encryption and decryption. Advantages: The algorithm is publicly available, encryption/decryption operations are relatively simple, encryption and decryption are fast and efficient. Disadvantages: Cracking is relatively easy, key management is difficult, and not suitable for distributed systems.

3) Asymmetric Encryption Algorithm (RSA): The RSA algorithm is an asymmetric encryption algorithm that consists of a key pair consisting of a private key and a public key. Advantages: High security, as private keys are not sent over the network, suitable for distributed systems, as each node can have different public and private key pairs. Disadvantages: Slow encryption and decryption speed due to the need for a large amount of computing resources, a large amount of storage space is required to store public and private key pairs.

4) Diffie-Hellman key negotiation algorithm: Diffie-Hellman is a key negotiation algorithm (referred to as DH algorithm), which is based on a mathematical principle and can negotiate a key between two parties without leaking the key. Advantages: High security, disadvantages: No mechanism to resist man in the middle attacks.

5) Homomorphic encryption algorithm: It is a special encryption mode in cryptography that allows us to send encrypted ciphertext to any third party for computation without the need for decryption before computation.

2.2 Homomorphic encryption algorithm

Homomorphic encryption (HE) is a special encryption mode in cryptography that allows us to send encrypted ciphertext to any third party for computation without the need for decryption. The mathematical definition of homomorphic encryption is shown in formula 1.

$$E(m_1)\bigstar E(m_2)=E(m_1\bigstar m_2) \quad \forall\ m_1,m_2 \in M \tag{1}$$

Where E is the encryption algorithm and M is the set of all possible information. If the encryption algorithm E satisfies formula (1), then we say that E conforms to the property of homomorphic encryption in the ★ operation. The current homomorphic encryption algorithms mainly support two types of homomorphic operations: addition and multiplication.

Homomorphic encryption algorithms generally include the following four parts: 1) KeyGen: a key generation algorithm that generates public and private keys; 2) Encryption: Encryption algorithm; 3) Decryption: Decryption algorithm; 4) Homomorphic Property: Homomorphic encryption computation part. As shown in Figure 1.

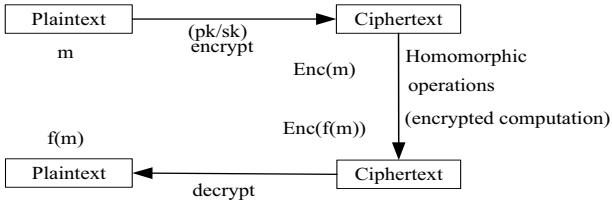

Figure 1. Homomorphic encryption algorithm process

2.2.1 Classification of Homomorphic Encryption Techniques

Homomorphic encryption mainly includes partial homomorphic encryption algorithm, hierarchical homomorphic encryption algorithm, and fully homomorphic encryption algorithm.

1) Partial Homomorphic Encryption (PHE) refers to homomorphic encryption algorithms that have homomorphic properties only for addition or multiplication (one of which). The advantage of PHE is its simple principle and easy implementation, but the disadvantage is that it only supports one type of operation (addition or multiplication).

Paillier algorithm is a typical HE encryption algorithm [5], which is a probabilistic public key encryption system invented by Paillier in 1999. Difficult problem based on composite residual classes. The encryption algorithm process is as follows:

Step1: Randomly select two prime numbers p and q, meeting gcd (p.q, (p-1) (q-1)) =1, ensuring that the lengths of p and q are proportional as much as possible, and gcd is the greatest common divisor;

Step2: Calculate N=p.q and λ=lcm (p−1, q−1), where lcm is the least common multiple;

Step3: Randomly select $g \in Z\ *N^2$, meeting gcd ($L(g^{\lambda} \bmod N^2)$, N) = 1(directly take g=n+1).Where Z represents an integer, and the index represents how many elements are in the set of integers;

$$L(x)=\frac{(x-1)}{N} \qquad \mu = (L\ (g^{\lambda} \bmod N^2))^{-1} \bmod\ N; \tag{2}$$

The public key is (N, g); the private key is (λ, μ).

Step4: Encryption process: for any plaintext message $m \in Z_N$, Choose any random number $r \in Z_N^*$, Calculate ciphertext:

$$c = E(m) = g^m r^N \bmod N^2 \quad (3)$$

Step5: Decryption process: For ciphertext $c \in Z^* N^2$, calculate plaintext m:

$$m = D(c) = \frac{L(c^\lambda \bmod N2)}{L(g^\lambda \bmod N2)} \bmod N = L(c^\lambda \bmod N^2)\,\mu \bmod N \quad (4)$$

Step6: Additive homomorphic computation

For any plaintext m1, m2 $\in Z_N$, r1, r2 $\in Z_N^*$, The corresponding ciphertext c1=E [m1, r1], c2=E [m2, c2], meeting

$$c1 * c2 = E[m1, r1] * E[m2, c2] = g^{m1+m2} * (r1 * r2)^N \bmod N^2 \quad (5)$$

After decryption, we obtained:

$$D[c1 * c2] = D[E[m1, r1] * E[m2, r2] \bmod N^2] = (m1 + m2) \bmod N \quad (6)$$

So c1 * c2=m1+m2 ciphertext product is plaintext.

2) Hierarchical homomorphic encryption algorithms (LHE, Leveled HE) is mainly aimed at solving the problem of being able to support both addition and multiplication at the same time, but the disadvantage is that the number of calculations is limited. A typical representative of this is the BGN encryption algorithm, which can support one multiplication and one addition, and is an upgraded solution for partially homomorphic encryption.

3) Fully Homomorphic Encryption Algorithm (FHE) supports infinite computations of any type on ciphertext. The advantage of FHE is that it supports a large number of operators and has no limit on the number of operations. The disadvantage is that its efficiency is very low and it is currently unable to support large-scale calculations. The schematic diagram of the encryption process is shown in Figure 2.

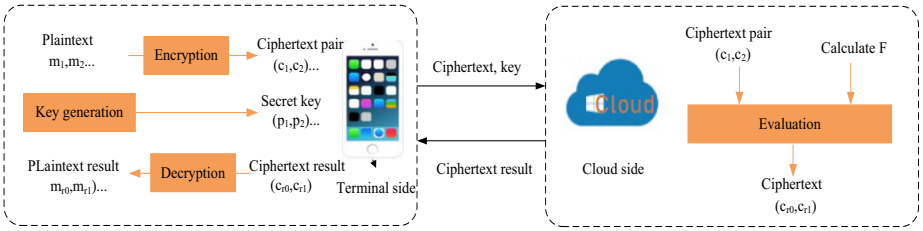

Figure 2. Schematic diagram of FHE encryption process

The CKKS (Cheon Kim Kim Kim Song) algorithm [6] is a fully homomorphic encryption algorithm that proposes an approximate calculation method that can encrypt real numbers. It supports most real-world application environments and has great potential for development. At present, the libraries that support the CKKS solution mainly include HEAAN, SEAL, HElib, PALISADE, CIMERA, and PEGASUSSEAL. The biggest advantage of these libraries is that they are developed by Microsoft and can be used on Windows.

3. Fully Homomorphic Encryption Scheme Based on Fibonacci Matrix (FM-HE)

3.1 Fibonacci matrix and inverse matrix

Due to the long history and unique properties of studying Fibonacci sequences, they are often applied in fields such as computational science, physics, aviation, and military, and have produced excellent results [7-8]. In practice, we use third-order Fibonacci matrices and inverse matrices. The second-order Fibonacci sequence F0=0, F1=1, Fn+1=Fn+Fn-1, n=1,2,3…, The second-order Fibonacci matrix T_2 can be used to represent [9], as shown in formula (7).

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = T_2^n \; n=1,2,3\ldots \tag{7}$$

The third-order Fibonacci sequence F0=1, F_0=1, F_1=1, F_2=2, $F_{n+1}=F_n+F_{n-1}+F_{n-2}$, n=2,3,4…, The third-order Fibonacci matrix T_3 can be used to represent [10], as shown in formula (8).

$$\begin{pmatrix} F_n & F_{n+1}-F_n & F_{n-1} \\ F_{n-1} & F_n-F_{n-1} & F_{n-2} \\ F_{n-2} & F_{n-1}-F_{n-2} & F_{n-3} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^n = T_3^n \; n=3,4,5\ldots \tag{8}$$

Since the determinant $|T_3|=1$, T_3 is reversible. Calculate T_3 and T_3^{-1}, as shown in formula (9).

$$T_3=\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad T_3^{-1}=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix} \tag{9}$$

Using n as a parameter, n=6, 7, 8, 9, 10..., it is easy to calculate T_3^n and $(T_3^n)^{-1}$, as shown in formula (10).

$$T_3^n=\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^n \quad (T_3^n)^{-1}=\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}^n \tag{10}$$

We define $F_n=T_3^n$, $F_n^{-1}=(T_3^n)^{-1}$, Utilizing the unique feature of invertibility and ease of computation of third-order Fibonacci matrices. By using n as a parameter, F_n and F_n^{-1} are applied in the encrypted transmission process to achieve tamper proof and encryption effects. As shown in Figure 3, the sender sends plaintext M data to the receiver. Firstly, it is transformed into G3 (G_1,G_2,G_3) through F_n transformation. Then, it is passed through an intermediary to the receiver. Finally, the receiver restores the plaintext through F_n^{-1} transformation.

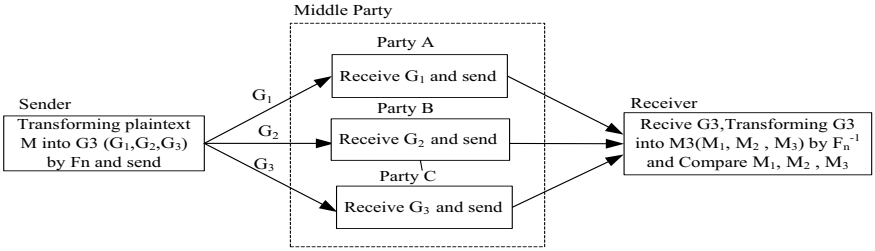

Figure 3. Application process of third-order Fibonacci matrix

3.2 FM-HE encryption process

By combining the trivalent Fibonacci matrix and Diffie-Hellman, the FM-HE homomorphic encryption algorithm is formed. During the FM-HE process, firstly, during encryption, since the encryption key is random each time, it needs to go through the Fibonacci transform F_n, so the ciphertext generated each time is different. If an intermediary obtains the ciphertext, it will be analyzed vaguely; Secondly, during decryption, the Fibonacci matrix inverse transformation F_n^{-1} needs to be performed. Furthermore, by encrypting with the password vector X and decrypting it into the plaintext vector Mn, a vote verification is required to finally determine the plaintext. The FM-HE homomorphic encryption process is shown in Figure 4.

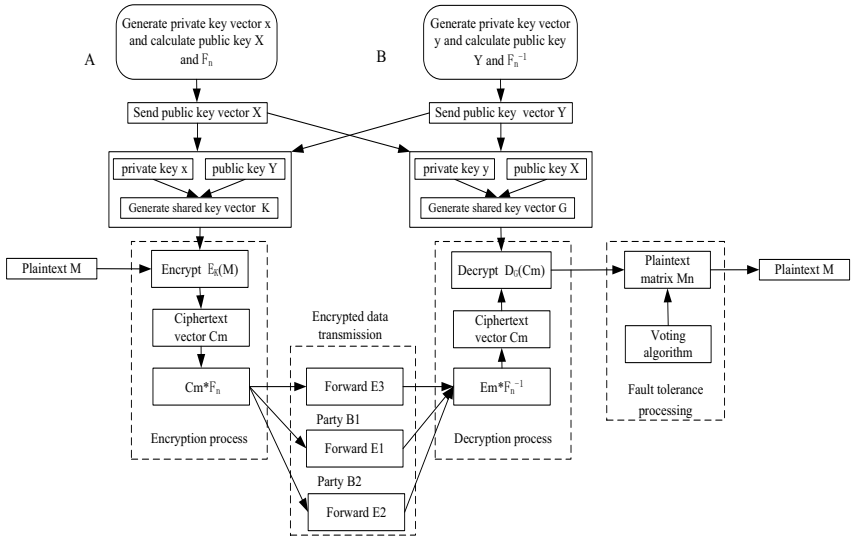

Figure 4. FM-HE encryption process flow diagram

4. Encryption performance testing

To compare encryption performance, we chose to test on a personal computer with an Intel (R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, RAM: 8.00 GB. The encryption performance comparison between FH-HE, Paillier, and CKKS is shown in Table 1:

Table 1. Comparison of Encryption Performance

HE Algorithm	Plaintext (bytes)	Encrypt Time(ms)	Decrypt Time(ms)	Fault tolerance
Pailier	1.5k	11.13	39.05	No
CKKS	1.5k	10.78	17.68	No
FH-HE	1.5k	10.33	13.12	Yes

Table 1 shows the performance comparison between FH-HE, Paillier, and CKKS. From the results, it can be seen that encrypting plaintext of the same size takes about 11ms, but FH-HE has the shortest decryption time of 13.12ms. Not only that, but FH-HE also has fault tolerance compared to the other two homomorphic encryption

methods. Therefore, when encrypting data transmission in machine learning, FH-HE has better encryption performance.

5. Conclusions

This article proposes a fully homomorphic encryption scheme based on Fibonacci matrix, which solves the data transmission security issues of federated learning and others. By using the F_n and F_n^{-1} transformations of the Fibonacci matrix and HD, security has been enhanced and fault tolerance has been improved. The feasibility of the FH-HE scheme has been demonstrated through a comparison of encryption performance. In future research, we will expand the application of FH-HE scheme, and its application in other areas will continue to be studied in the future.

References

[1] Y.-P. Yang, Y. Zhao, J.-M. Zhang, Recent Development of Theory and Applicationon Homomorphic Encryption [J]. Chinese Journal of Electronics & Information Technology, 2021, 43(2):13.

[2] X.-Y. Lu, J.-W. Chen, Y. Feng, W.-Y.Wu, Privacy-preserving Data Classification Protocol Based on Homomorphic Encryption [J]. Computer Science, 2023, 50(8): 321-332.

[3] Jianping, Wu, et al. "Federated learning for network attack detection using attention-based graph neural networks." Scientific Reports 14.1(2024).

[4] Z. Zhou et al., "Latent Vector Optimization-Based Generative Image Steganography for Consumer Electronic Applications," in IEEE Transactions on Consumer Electronics, vol. 70, no. 1, pp. 4357-4366, Feb. 2024, doi: 10.1109/TCE.2024.3354824.

[5] G. Crihan, M. Crăciun, L.Dumitriu, A Comparative Assessment of Homomorphic Encryption Algorithms Applied to Biometric Information. Inventions 2023, 8, 102.

[6] J. Lee, P.-N. Duong, H. Lee, Configurable Encryption and Decryption Architectures for CKKS-Based Homomorphic Encryption. Sensors 2023, 23, 7389.

[7] G.-Y. Lee, S.-H. Cho, The Generalized Pascal Matrix Via the Generalized Fibonacci Matrix and Generalized Pell Matrix. Korean Mathematical Society, 45(2018), 479-491.

[8] C. Shuzhen, W. Zhu, The general term and property of the five order Fibonacci series. Journal of Hainan Normal University (Natural Science), China, 12(2014), 241.

[9] X. Xie, Discussion and application of Fibonacci matrix. Scientific and technological information, 24(2008), 2.

[10] L. Peng, Properties and applications of third-order Fibonacci sequence. Journal of Putian University, 5(2006),5.