Homomorphic Encryption and Its Security Application

Xun-yi Ren¹, Lin-juan Chen¹, Hai-shan Wan²

¹College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China;
²Shandong Zhongfu Information Industry INC, Jinan, Shandong 250101, China
renxy@njupt.edu.cn

Abstract
Homomorphic encryption scheme allows one to directly operate ciphertext without being able to decrypt and it has a wide application prospect. In this article we describe the typical encryption schemes including single-key homomorphic encryption scheme and public key homomorphic encryption scheme and then elaborate its application in the field of security, including the applications in database security, secure multi-party computation and the Internet of things. Finally, we discuss the research direction of homomorphic encryption.

Keywords: homomorphic encryption, privacy protection, ideal lattice, secure multi-party computation

1. Introduction
With the rapid development of information technology, problems like the storage security of mass data, sensitive data against theft and anti-tampering have drawn increasing attention of people. For some important or sensitive data, users are eager to be able to complete its storage and transmission through ciphertext and operate on the database information without divulging the contents. Data encryption methods can be applied into different environments, where a common problem is that the ciphertext data can not operated. In other words, when to do sorting, search, statistics, average and summation on some fields of the ciphertext data, we must first decrypt the fields and then encrypt the decrypted plaintext after the operations. As a result, the time and space overhead is increased. What’s more, it’s unable for users to operate some important or sensitive data without knowing the contained information. If mathematical and conventional operations can be done on ciphertext data, obviously the problems above can be solved and the time and space overhead during encryption and decryption would be decreased, and the database efficiency would be improved. Homomorphic encryption technology is an effective way to solve all this problems.

2. Homomorphic Encryption Technology
Homomorphic encryption, originally called a privacy homomorphism, allowing direct operation of the ciphertext, was introduced by Rivest, Adleman and Dertouzous¹ in 1978. However, due to its unsafe with known-plaintext attack, it was further improved by Domingo² later. Homomorphic encryption technology was first used for statistical data encryption and the homomorphism of the algorithm ensures that user can operate on sensitive data without disclosing information.

2.1 Basic Ideal of Homomorphic Encryption
Homomorphic encryption technology is based on the algebraic theories and the basic idea is as follows.
Suppose $E_{k_1}$ and $D_{k_2}$ are respectively encryption function and decryption function. Explicit data
space is a finite set of elements \( \{M_1, M_2, \ldots, M_n\} \), \( \alpha \) and \( \beta \) are operations. Then the function \( (E_{\alpha}, D_{\beta}, \alpha, \beta) \) is called a secret homomorphism if
\[
\alpha(E_{\alpha}(M_1), E_{\alpha}(M_2), \ldots, E_{\alpha}(M_n)) = E_{\alpha}(\beta(M_1, M_2, \ldots, M_n))
\]
and
\[
D_{\beta}(\alpha(E_{\alpha}(M_1), E_{\alpha}(M_2), \ldots, E_{\alpha}(M_n))) = \beta(M_1, M_2, \ldots, M_n).
\]

2.2 the Process to Implement Secret Homomorphism

a) Select safe large primes \( p \) and \( q \) and then calculate \( m = p \cdot q \) (keep \( m \) secret).

b) Select safe parameter \( n \) (select the appropriate size according to the need).

c) Plaintext space \( T = \mathbb{Z}_m \) (all non-negative integers less than \( \mathbb{Z} \)). Ciphertext space \( T' = (\mathbb{Z}_p \times \mathbb{Z}_q)n \).

d) Select two prime numbers \( r_p \) and \( r_q \), where \( r_p \in \mathbb{Z}_p \) and \( r_q \in \mathbb{Z}_q \).

e) Determine the encryption key \( K = (p, q, r_p, r_q) \).

f) Encryption algorithm \( E_k(x) \) is as follows.

Suppose a plaintext \( x \in \mathbb{Z}_m \). Randomly divide \( x \) into \( n \) pieces of \( x_1, x_2, \ldots, x_n \), \( x = \sum_{i=1}^{n} x_i \mod m \).

\[
E_k(x) = \left[ \left[ x_1 r_p^i \mod p, x_1 r_q^i \mod q \right], \left[ x_2 r_p^i \mod p, x_2 r_q^i \mod q \right], \ldots, \left[ x_n r_p^i \mod p, x_n r_q^i \mod q \right] \right], \quad \text{where} \quad x_i \in \mathbb{Z}_m, \ i = (1, 2, \ldots, n).
\]

g) Decryption algorithm \( D_k(x) \) is as follows.

1) Calculate
\[
\left[ \left[ x_1 r_p^{-i} \mod p, x_1 r_q^{-i} \mod q \right], \left[ x_2 r_p^{-i} \mod p, x_2 r_q^{-i} \mod q \right], \ldots, \left[ x_n r_p^{-i} \mod p, x_n r_q^{-i} \mod q \right] \right], \quad \text{where} \quad r_p^{-i} \text{ and } r_q^{-i}
\]
are respectively the multiplicative inverse of the corresponding power of \( r_p \mod p \) and \( r_q \mod q \).

2) Calculate \( \sum_{i=1}^{n} \left[ x_i \mod p, x_i \mod q \right] = \left[ \sum_{i=1}^{n} x_i \mod p, \sum_{i=1}^{n} x_i \mod q \right] \).

3) Use the Chinese Remainder Theorem to calculate \( D_k(x) = (sxq^{-1} + xpp^{-1}) \mod m \), where \( qq^{-1} = l \mod p, pp^{-1} = l \mod q \).

3. Homomorphic Encryption scheme

3.1 Single Key Homomorphic Encryption scheme

Sander and Tschudin defined the additive and multiplicative homomorphic encryption mechanism in integer scope (HES)\(^{13}\) ensuring that the result after encrypting the two variables is identical to the result before encryption. Sander and Tschudin proposed move cryptogram system with HES, having following insufficiencies. First, no cryptogram system is established with additive homomorphism, multiplication homomorphism or mixed multiplication homomorphism. Then, only some restricted functions are proved to be suitable for HES (such as rational polynomial functions).

3.2 Public Key Homomorphic Encryption scheme

As probabilistic algorithms constructed out of certain mathematical problems, most public key mechanisms are not homomorphic. For instance, ElGamal\(^{[4]}\) algorithm can reach multiplication homomorphism only when the two encrypted random numbers selected are the same. But some public-key mechanisms are exceptions. For instance, RSA encryption scheme is multiplicatively homomorphic. And Naccache-Stern\(^{[5]}\) encryption scheme is additively homomorphic. Alternatively, you can create a additive homomorphism scheme based on lattice or linear code\(^{[6,7]}\).
Lattice public key cryptogram, the most typical representation of quantum cryptography, recently become the hotspot of cryptogram research once again. New theories, new methods, new results are emerging. The lattice-based scheme by Melchor, Castagnos and Gaborit\(^7\) also allow multiplications, though with exponential expansion in ciphertext size. Such schemes have a different flavor from the more “conventional” schemes above, because ciphertexts implicitly contain an “error” that grows as ciphertexts are added together.

A technique that they call “interval obfuscation” can be viewed as a symmetric homomorphic encryption scheme. It uses a secret integer modulus \(M\) and a secret integer \(s\) that is relatively prime to \(M\). A ‘0’ is encrypted as \(s \cdot x \mod M\) for some \(x \in \{1, a\}\), where \(a\) is a “small” integer, while a ‘1’ is encrypted as \(s \cdot x \mod M\) for some \(x \in \{b + 1, a + b\}\), where \(b\) is a “large” integer (but still small in comparison to \(M\)). The recipient decrypts \(c\) by setting \(c' \leftarrow c / s^d \mod M\). One can view their scheme as using a one-dimensional ideal lattice.

Based on these studies, IBM researcher Craig Gentry\(^8\) published a paper in STOC in September 2009 where he constructed a fully homomorphic encryption scheme using ideal lattice. An encryption algorithm is called a fully homomorphic encryption algorithm if the corresponding operation for multiplication and addition can be found. In other words, its meaning is that we can construct corresponding encryption operation for any complex plaintext operation. It can offer search on encrypted data. A user stores an encrypted file on a remote server and the server is only allowed to search the files that satisfy the boolean constraints but not to decrypt the files. The advantage with ideal lattice replacing integers of this scheme is that ideal lattice has maney bases. Some good bases can be used as private keys and some bad for public which are only useful for encryption but valueless for decryption. So a public-key encryption scheme can be built base on ideal lattice. On the other hand, that whether integer \(p\) can be used only valuable for encryption is not clear. And the polynomial ring of ideal lattice is ideal for meeting both additive homomorphism and multiplication homomorphism. Besides, the operations involved in ideal lattice are very simple (for a small integer \(p\), the multiplication operation in \(\mathbb{Z} / (x^n - 1)\)). The time NTRU lattice uses to attack is by the \(n\) index so \(n\) can be reasonably reduced hundreds of times. There are two reasons using ideal lattice to build Homomorphic Encryption scheme. 1. The circuit complexity of the decryption algorithm based on typical ideal lattice is very low, especially compared to RSA and ElGamal. 2. The polynomial ring corresponding to ideal lattice is next to ideal, from where addition and multiplication operations inherit. However, the scheme hasn’t been put into practical application for its synchronization efficiency to be improved.

4. Applications of Homomorphic Encryption Technology

4.1 Applications in Database Security

Dense state database technology is the amalgamation of cryptography and database technology\(^9\), in which the stored data gets more secure confidentiality when using database technology. According to different encryption granularity\(^10\), encryption control of database can be divided into encryption by file, encryption by records, encryption by fields and encryption by data items. However, it has brought new problems. As the encrypted database usually no longer have the original characteristics, traditional database operations such as queries and updates will not be used to dense state database. Indexing mechanism Cui Guohua et al\(^11\) doesn’t fundamentally solve the problem as part of the information must be plaintext. In addition, database encryption will have bad effects on the efficiency of queries and operations. In dense state database, systems usually need to get original information through decryption to provide information services for users. But from another perspective, database encryption will leads to the exposure of large number of user data. The research and algorithm implementation of secret homomorphism technology by Yang Yong et al\(^12\) and application of secret homomorphism technology in database security by Wang Xiaofeng et al\(^13\) fundamentally solve the problem with proposing the concept of homomorphism. But that how to concretely implement is not discussed. Thus, homomorphism encryption technology is the primary research point dense state database faces.
4.2 Applications in Secure Multi-party Computation

For a agreed function $f : D^* \leftarrow D^*$, n persons with mutual mistrust confidentially enter $x_1, x_2, \ldots, x_n$ in a special way and correctly calculate $f(x_1, x_2, \ldots, x_n)$ without disclosing any information about confidentially input $x_1, x_2, \ldots, x_n$ except the function value. The concept above is namely secure multi-party computation that was first mentioned by A.C. Yao [14] in 1982. The previous work about data comparison of protecting private information is as follows. Yao gave a protocol on comparing the size but the efficiency was very low. Compare equal issue was not discussed. Document [15] showed a effective and fair protocol and proved the security. But compare equal issue was also not involved. Though document [16] showed some protocols about compare equal issue, the security of the protocols was not proved. Based on inadequate consideration of the "fairness" of previous protocols, QinJing et al [17] proposed a protocol that can solve both compare equal issue and the fairness in document. Li Shundong et al. [18] proposed a more efficient protocol with oblivious transfer protocol and self-defined special functions to solve the two sides compare issue and gave a proof of the security and efficiency analysis. However, all the protocols above are for both sides and each own one data. There are no comparison protocols of multi-side data. Liu Wen, Luo Shou-Shan and Chen Ping [19] proposed a secure multi-party data sorting protocol, based on ElGamal homomorphic cryptosystem to solve the expansion of the “Millionaire”, and used the definition of security in secure multi-party computation to confirm the correctness and safety of the protocol in the semi-honest model. It lays the foundation for new electronic transactions like secure electronic tendering and auctions and secure online trading. Qiu Mei, Luo Shou-Shan, Liu Wen and Chen Ping [20] proposed another scheme based on RSA homomorphic cryptosystem to solve the secure multi-party data sorting issue. The correctness security and efficiency of the protocol were analyzed in the semi-honest model.

4.3 Applications in the Internet of Things

4.3.1 Data processing and privacy protection

That the storage and processing of mass data is one of the key issues the Internet of Things facing. Users’ confidential information and private data will be more and more with the continuous development and application of the Internet of Things. One of the problems users are most concerned about is how to keep confidential information unknown by others (including service providers). But service providers are most concerned about how to process the data accurately and efficiently and extract the contained valuable information with keeping user data unknown by others (including themselves). All these problems are difficult to achieve by traditionally encryption schemes.

Homomorphic encryption is a new technology to the problem of data processing and privacy protection. It can process encrypted data without decryption and get the same effect with directly processing the raw data. Users can pass the data to be processed to the cloud server in ciphertext form. The server can directly process the ciphertext data without being decrypted by the user and then return the results back to the user. The user can get the processed plaintext data by homomorphically encrypting the received results.

Li et al [21] proposed a combination of disruption technology and encryption technology in a data mining process, where disruption technology can ensure the efficiency of privacy protection algorithm and accompanied encryption technology further ensure the privacy of data. He also proposed a tripartite and multi-party safe distance comparison protocol in Euclidean space with the combination of homomorphic encryption technology [22], which can directly support the data mining methods based on Euclidean distance and its extensions, such as K-neighbor, K-means etc, which are versatile and universally significant in the fields of classification mining, clustering mining and Web mining.

Yu Di [23] proposed a new classification algorithm with the combination of homomorphic encryption and order-preserving encryption, supporting privacy protection. With the premise of ensuring the results of data mining, the algorithm solved mathematical manipulation and numerical comparison of the ciphertext, met the needs of privacy protection and reduced the algorithm complexity of communications.
and computation. In the paper, a new clustering method was also proposed by changing the clustering steps in the condition of vertical distribution of the data, which was applied to the K-center clustering algorithm to realize effective clustering mining as well as the protection of data privacy.

4.3.2 Information Retrieval

With the development of the Internet of things, more and more encrypted data is stored in the cloud server. The retrieval of encrypted data would become an urgent problem when the encrypted data stored in the server side reach a certain size. Current information retrieval algorithms, including linear search, public-key search and safety index, can retrieve encrypted data quickly. But they are only suitable for small-scale data retrieval and very expensive. Complete homomorphism-based data retrieval methods, being able to directly retrieve the encrypted data, can not only ensure the retrieved data against statistical analysis but also do basic addition and multiplication without changing the corresponding plaintext order so that user data security is protected and the retrieval efficiency is improved.

4.3.3 Copyright Protection - Digital Watermarking

There must be many digital products circulating in the network with the Internet of things being widely used in the business. That whether the copyright of these digital products can be effectively protected from violation would directly affect the development and applications of the Internet of things in e-business. Currently, information hiding and digital watermarking technology are relatively mature and have been widely used in the Internet. However, that how to deal with the security challenges of data hiding and digital watermarking systems in complex network environment is a problem to be solved.

Homomorphic encryption-based digital watermarking scheme by Zhi Li et al.[24] can effectively resist unauthorized detection attacks. The working principle is shown in Figure 1. In the scheme watermark signal is encrypted by homomorphic encryption technology and then embedded into the original vector. The vector containing a watermark should be homomorphically decrypted before the detection of watermark signal to ensure that the undetected watermark signal has no obvious correlation with the vector containing a watermark. After decrypting the vector containing a watermark, authorized users can determine the existence of the watermark and then extract the watermark through calculating the correlation between the decrypted vector and the watermark signal.

4.4 Applications in other fields

As cloud computing will become increasingly popular in the future, homomorphic encryption technology would make companies able to store sensitive information in a remote server to avoid the occurrence of leakage from local host and remain the use and search of information. Users can also use a search engine to query and get results without worrying about that your query records would leave in the engine. With the analysis of storage security technology in cloud storage applications and common
encryption retrieval methods and related technologies, in connection with the needs of encrypted storage, Huang Yongfeng, et al.[25]. Proposed a fully homomorphic encryption-based retrieval method with the combination of his research results, and it can improve the retrieval efficiency to some extent.

With the combination of homomorphic encryption, threshold encryption and secure multi-party computation, Ge et al.[26] proposed the collusion-resistance protocol that is used in distributed association rule mining. Because of the use of threshold homomorphic encryption technology[27], compared with other documents, the protocol can be used in not only quasi-credit environment but also malicious environment.

5. Conclusion

Homomorphic encryption mechanism has a good prospect in the fields of secure multi-party computing, the Internet of things and cloud computing. But the research of this method is currently still in its infancy. In the future further research can be started from the following aspects:

1. Homomorphic encryption mechanism, a partially safe encryption mechanism, needs a lot work to be better improved with full security analysis. Further study of the fusion of homomorphic encryption and other encryption mechanisms (such as the combination of encryption functions) to get more secure encryption system is a focus of future study.

2. As homomorphism encryption technology is generally based on public key cryptography, the complexity of the algorithm is usually higher than other shared-key –based encryption technology and general disruption technology. Therefore, how to effectively reduce the algorithm complexity and promote the efficiency is an important direction in future research.

3. To create a more efficient algorithm to achieve “calculation hiding” with the combination of homomorphism encryption is a hotspot of future research.

4. Currently common used homomorphism encryption technology generally can only achieve two operations of addition and multiplication. How to further expand the type of operations is worthy of studying.

6. Acknowledgement

Supported by National Natural Science Foundation of China (61073188), China Postdoctoral Science Foundation (20100471355), A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001)

7 References


Li Feng, “Privacy for data mining method”, Shanghai: Shanghai Jiao Tong University, China, 2008.


Li Feng, “Privacy for data mining method”, Shanghai: Shanghai Jiao Tong University, China, 2008.


