Fully Homomorphic Encryption and Its Applications

Xiaojun Liu [1]*, Chunxia Tu [2]
1.School of Transportation, Huanggang Normal University, Hubei Huanggang, China
*E-mail:18623582@qq.com
2.School of Computer, Huanggang Normal University, Hubei Huanggang, China

***Abstract*—The concept of fully homomorphic encryption was first introduced by Rivest et al. in 1970s.How to construct such schemes is a hard problem for cryptographers. Until 2009, Gentry presented the first fully homomorphic schemes based on ideal lattice, which is a breakthrough in this field. Subsequently, many cryptographers made meaningful work on the research of fully homomorphic encryption schemes, and promoted the development of fully homomorphic encryption to be practical. This paper introduces the research progress of the full homomorphic encryption, including full homomorphic encryption scheme and its optimization. Private data bank, encrypted search, cloud computing, Multi-party security calculations, electronic voting, digital watermarking and other applications are taken as examples to introduce the significant value of application of fully homomorphric encryption.**

***Index Terms*—cloud computing; information security; fully homomorphic encryption; cryptography**

1. INTRODUCTION

Cloud computing offers many services to users, including the option to offload storage and computation of large amounts of data to the cloud. However to take advantage of cloud computing, users must trust and share their data with the cloud service providers. One way to ensure data privacy is to encrypt data before uploading to the cloud. However, if users wish to compute on the data, the data must be downloaded and decrypted, making the main advantages of using cloud services redundant. One solution for providing secure cloud computing on untrusted public clouds is the use of homomorphic encryption: a method of encryption which allows computations on encrypted data, without the need to fully decrypt the data on the cloud. Partially homomorphic encryption schemes have been known for many years, offering the ability to carry out a certain type of operation on ciphertexts without decryption, for example addition or multiplication, such as the additively homomorphic Paillier [1] or the multiplicatively homomorphic ElGamal [2] cryptosystems.

II. PREPARATION KNOWLEDGE

A. Homomorphic encryption

Fully Homomorphic encryption public key scheme contains four algorithms: key generation algorithm($KeyGen$), encryption algorithm(Enc), decryption algorithm (Dec)and ciphertext calculation algorithm($Evaluate(evk, f, c_1, \dots, c_t)$). Among them the $Evaluate$ algorithm is the core, because the purpose of the homomorphic encryption is to calculate the ciphertext. The other three algorithms are the cornerstone, providing encryption and decryption functions.

$\boldsymbol{KeyGen}$: $(pk, sk) <- KeyGen\ (k)$, Select a parameter K, generate the program's public key pk, private key sk.

$\boldsymbol{Enc}$: $c \leftarrow Encpk\ (m)$,Give a plaintext m, use the public key pk to encrypt the plaintext m, and get the ciphertext c。

$\boldsymbol{Dec}$: $m \leftarrow Dec_{sk}(c)$, Enter the private key sk and ciphertext c to decrypt the operation, the output plaintext m。

Enter the encrypted public key evk and t input functions,a set of ciphertexts $\vec{c} = (c_1, c_2, \dots, c_t)$, Where c_i is the encrypted ciphertext corresponding to plaintext m_i

The output is $c^* = Evaluate(evk, f, \vec{c})$, and is satisfied:

$$Dec(sk, c^*) = f(m_1, m_2, \dots m_t)$$

Homomorphic encryption schemes must satisfy the following basic calculations of ciphertext:

$Add(pk, c_1, c_2)$
$Mult(pk, c_1, c_2)$

B. Homomorphic decryption

As ciphertext noise grows in ciphertext calculation (multiplication noise growth is particularly significant), resulting in ciphertext calculation is bounded, beyond which ciphertext decryption may fail. Therefore, the control of ciphertext noise has become a key issue to achieve fully homomorphic encryption. Gentry uses an important technique to solve this problem:Homomorphic decryption, that is, encrypts the ciphertexts and the corresponding keys bit by bit, and then inputs the decryption circuits. The homomorphic execution decryption circuit in the evaluate algorithm outputs a new ciphertext Called ciphertext update), the ciphertext decryption or the original plaintext. If the new ciphertext noise also allows for a multiplication, after each ciphertext calculation, through the homomorphic decryption update ciphertext can be the next calculation, the recursive process can be unlimited ciphertext calculations, In order to achieve the fully homomorphic encryption.

Homomorphic decryption process is as follows:

Suppose each element of $Encrypt(pk_1, m) \rightarrow c_1, Encrypt(pk_2, sk_{1j}) \rightarrow \overrightarrow{sk_1}, \overrightarrow{sk_1}$ is a ciphertext encrypted with pk_2 for each binary digit in sk_1.

Homomorphic decryption algorithm is

$recypt(pk_2, Dec, \overrightarrow{sk_1}, c_1)$:

$encrypt(pk_2, c_{1j}) \rightarrow \overrightarrow{c_1}$,

$evaluate(pk_2, Dec, \overrightarrow{sk_1}, \overrightarrow{c_1}) \rightarrow c_2$.

Where: c_1 is the double encryption of m, the inner encryption is under pk_1 and the outer encryption is under pk_2 ; c_2 is the result of executing the homomorphic decryption circuit and is the ciphertext encrypted by m at pk_2.

The wonders of homomorphic decryption lie in decrypting the inner layer to get plaintext (with noise removed), then plaintext under the new key (again introducing new noise), as long as the new noise introduced allows one more The purpose of multiplication, homomorphic decryption is reached.

C. DGHV

In June 2010, Dijk , Gentry , Halevi and Vaikuntanathan published an article entitled Fully Homomorphic Encryption over the Integers[3]. The scheme replaces the ideal lattice with an integer loop, and uses the sum-of-products multiplication on the integer ring to replace the ideal lattice. The program has the concept of simple advantages, easier to understand.

Dijk et al. Give a symmetric encryption scheme.

$KeyGen_\varepsilon$: The key is a prime number $\in (2^{\eta-1}, 2^\eta]$,

$Encrypt_\varepsilon$: Enter plaintext $m \in \{0,1\}$, key p, output ciphertext $c \leftarrow m + pq + 2r$, among them q, r are randomly selected integers, and $m + 2r < p/2$,

$Decrypt_\varepsilon$: Enter the ciphertext c , key p , after calculating $m \leftarrow (c \bmod p) \bmod 2$, output plaintext m.

When noise $m + 2r$ is less than $p/2$, the above scheme can be correctly decrypted . This is a symmetric homomorphic encryption scheme that can be modified to a public-key encryption scheme. Specifically, add a number of 0 ciphertexts to the public key, that is $x_i = q_i p + 2r_i$, q_i and r_i are also selected according to the above solution, and the modified public key encryption scheme is as follows:

$KeyGen_\varepsilon$: $K_p = \langle x_0, x_1, \dots, x_\tau, \rangle, K_s = p$,

$Encrypt_\varepsilon$: Enter plaintext $m \in \{0,1\}$, public key K_p, output ciphertext $c \leftarrow m + 2r + \sum x_i$,

$Decrypt_\varepsilon$: Enter ciphertext$c$, key K_s, output $m \leftarrow (c \bmod p) \bmod 2$。

After that, Dijk et al revised the above scheme into a somewhat scheme by adding parameters and adding $Evaluate$ algorithms.

2. Application of homomorphic encryption

This section briefly summarizes the practical applications of homomorphic encryption. The application of homomorphic encryption in ciphertext calculation is the most basic and important application of homomorphic encryption. This application can make us do not need to trust the cloud service providers, and take advantage of the computing power and storage capacity of cloud service providers The implementation of computing and storage tasks, that is, computing outsourcing and storage outsourcing, can basically solve the problem of confidentiality protection of data in cloud computing and cloud storage and has important theoretical and practical significance for the popularization of cloud services. In addition, homomorphic encryption also has an extremely wide range of applications in other areas:

1). Private data banks. Rivest, Adleman, and Dertouzous [4-5] proposed homomorphic encryption when it came to the concept of homomorphic encryption that could be used to build private data banks. The so-called private data bank is the user can encrypt their own data stored in an untrusted server, then you can query the server for the information you need, the server generates a user's public key encryption query results, the user can decrypt the result Get the information they need, and the server does not know the user specific query content.

2). Encrypted search (encrypted search)[5-6]. With the existing circuit isomorphic encryption algorithm can achieve such an encrypted search process [29]: The user selects a fully homomorphic encryption scheme for the program to generate a public key pk. Pk encrypts the content to be searched b1, b2, ..., bn (bi can be a single bit of the user's query content) to generate the corresponding ciphertext c1, c2, ..., cn. Assume that circuit C represents the search engine query function, search engine using homomorphism calculation:

c * i = Evaluate (pk, Ci, (c1, c2, ..., cn)),

Where Ci is a sub-circuit of circuit set C for calculating the ith bit of the output; Evaluate () is a certain algorithm. The search server sends these results to the user, who decrypts ci with the private key sk to get Ci (b1, b2, ..., bn). These values constitute the user's search answer, and search engines do not know the true content of the user's search. Of course, if we can design algebraic homomorphism encryption algorithm, the significance of this encrypted search will be greater.

3) Multi-party security calculations[7-8]. The so-called multi-party secret calculation means that n (n≥2) participants P1, P2, ..., Pn own the secret data x1, x2, ..., xn respectively and they want to jointly calculate the function f (x1, x2, But are not willing to disclose their own confidential data. Multiple secure computing is the key technology of network privacy protection, which has important theoretical and practical significance in cryptography and information security. Many real-world games (such as poker games) can be described by a multi-party secure computing protocol. Many cryptographic protocols (such as secret sharing protocols, key distribution protocols and inadvertent transmission protocols) can be considered as a special Multi-party security computing agreement. Therefore, multi-party cryptographic computation is also a hot issue in cryptography research, and the homomorphic encryption algorithm is a powerful tool for constructing multi-party secure computing protocols.

4) Digital water mark. Digital watermarking technology refers to the signal processing method to

embed hidden tags in digital multimedia data, such tags are usually invisible and can only be extracted by a dedicated detector or reader.How to deal with data hiding in complex network environment And the security challenge of digital watermarking system is an urgent problem to be solved at present.One of the main security attacks against digital watermarking is unauthorized detection attacks, in which an attacker detects an unauthenticated watermarked carrier , To determine whether the watermark exists, and then guess or decipher the meaning of the watermark, and even remove the watermark in the carrier and embed a forged watermark. Watermarking [8] based on the homomorphic encryption digital watermarking scheme can effectively resist this attack This scheme first encrypts the watermark signal and the original carrier by using the homomorphic encryption system, and then embeds the encrypted watermark into the original carrier.When the user checks the watermark, the carrier containing the watermark must first be subjected to homomorphic decryption so that There is no significant correlation between the decrypted watermark signal and the watermarked vector. After decrypting the watermarked carrier, the watermark can be judged by determining the correlation between the decrypted carrier and the watermark signal, and then the watermark can be extracted.

5) Electronic voting. Electronic voting has the advantages that the traditional voting methods cannot match in terms of the quick and accurate counting of votes, the saving of manpower and expenses, the convenience of voting, etc. The design of secure electronic voting system is a typical application of the same homomorphic encryption. References [9] described a simple electronic voting scheme:

a) Voters encrypt their votes if there is a homomorphic function Enck (x1 + x2) = Enck (x1) × Enck (x2) Ci = Enck (Mi) where Mi∈ {0,1};
b) The voting center collects the ciphertext encrypted voters' votes Ci. The voting center calculates the ciphertext Ci based on the homomorphism of the homomorphic encryption scheme C = C1 × C2 × ... × Cn, After the election results C = Enck (M1 + M2 + ... + Mn);
c) Only a trusted institution that has a decryption key can decrypt the encrypted election result and publish the election result In the above process, the ballot box collection and counting completely operate on the encrypted ballot data without the need of Using any decryption key, so that either subject or agency can fulfill the responsibility of the teller, whether trusted or not.

6) other aspects of the application. Homomorphic encryption has applications in all aspects of cryptography, such as key distribution, inadvertent transmission, zero-knowledge proofing, and the like. In addition to this there are other applications such as Goldreich and Ostrovsky who studied the application of homomorphic encryption in software protection; Ostrovsky and Skeith envisaged using the homomorphic encryption algorithm to achieve the following obfuscation Function: Alice first writes a program P and encrypts E (P) and sends it to Bob, Bob provides the program with an input x, runs P (x) with the encrypted program E (P) It has also been widely used in proxy re-encryption [10]. Homomorphic encryption can be used to construct zero-knowledge proof protocols and it can also be used to construct key authentication negotiation protocols and proxy re-encryption Encryption protocol.

CONCLUSION

Fully homomorphic encryption in cloud computing, e-commerce and other important applications in practice so that it has been more and more attention by cryptographers. On the one hand, breakthroughs in the study of fully homomorphic encryption continue to meet the needs of search and processing of encrypted data on the Internet. On the other hand, the application needs in turn promote the development of the theory. The first fully isomorphic encryption scheme was proposed in 2009, but due to the limitations of efficiency and key storage, the scheme is far from practical requirements. Later, in order to break through these limitations, researchers constantly make new work in this direction, making the fully homomorphic encryption gradually practical. However, there is still a long way to go in the research on the security and practicability of the fully homomorphic encryption scheme.

REFERENCES

[1] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,”*in EUROCRYPT*, 1999, pp. 223–238.
[2] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.
[3] Diffie W, Hellman M E. “New directions in cryptography”. *IEEE Trans on Information Theory*, 1976, 22(6): 644-654.
[4] Elgamal T. “A public-key cryptosystem and a signature scheme based on discrete logarithms”. *IEEE Trans on Information Theory*, 1985, 31(4): 469-472.
[5] Gentry C, “Halevi S.Implementing Gentry’s fully-homomorphic en-cryption scheme” // *Advances in Cryptology*. Berlin: Springer, 2011: 129-148.
[6] Gentry C, Halevi S. “Fully homomorphic encryption without squashingusing depth-3 arithmetic circuits” //Proc of the 52nd IEEE *Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press*, 2011: 107-109.
[7] Rivest R, Adleman L, “Dertouzos M. On data banks andprivacy homomorphisms”. *Foundations of SecureComputation*,1978,4(11):169-180
[8] Li Zhi, Zhu Xinglei, Lian Yong, et al. “Constructing securecontent dependent watermarking scheme using homomorphicencryption” // *Proc of the 2007 IEEE Int Conf on Multimedia and Exposition (ICME 2007)*.Piscataway,NJ:IEEE,2007:627-630
[9] C Moore , M O'Neill , E O'Sullivan ,Y Doroz. “Practical Homomorphic Encryption: A Survey”. *IEEE International Symposium on Circuits & Systems* , 2014 :2792-2795
Gentry C. “A fully homomorphic encryption scheme”. *Stanford, CA: Stanford University*, 2009.