Advances in Computer Science Research, volume 83

8th International Conference on Social Network, Communication and Education (SNCE 2018)

Survey on Homomorphic Encryption Technology

Chunxia Tu[1, a *]

[1]School of Computer, Huanggang Normal University, Hubei Huanggang, China

[A]18623582@qq.com

* The corresponding author

Keywords: Cloud Computing; Information Security; Full Homomorphic Encryptionist

Abstract. Cloud computing technology has rapidly evolved over the last decade, offering an alternative way to store and work with large amounts of data. However data security remains an important issue particularly when using a public cloud service provider. The recent area of homomorphic cryptography allows computation on encrypted data, which would allow users to ensure data privacy on the cloud and increase the potential market for cloud computing. This paper reviewed the development of homomorphic encryption. It summarized two kinds of typical fully homomorphic encryption scheme, and analyzed the design method of fully homomorphic encryption algorithm. It also introduced the applications of the homomorphic encryption to the protection of the data confidentiality in cloud computing and to other fields. It pointed out the key problems of fully homomorphic encryption that needed to be researched at present. It can provide reference for further study of homomorphic encryption.

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.

Preparation Knowledge

Homomorphic encryption. Full 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 $\vec{c} = (\mathrm{c}_1, \mathrm{c}_2, \dots, \mathrm{c}_t)$, Where c_i is the encrypted ciphertext corresponding to plaintext m_i.The output is

$c^* = Evaluate(evk, f, \vec{c})$, and is satisfied: $Dec(sk, c^*) = f(m_1, m_2, \ldots m_t)$ .Homomorphic encryption schemes must satisfy the following basic calculations of ciphertext:

$Add(pk, c_1, c_2)$

$Mult(pk, c_1, c_2)$

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 full 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 full 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.

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 $p \in (2^{\eta-1}, 2^\eta]$;

$Encrypt_\varepsilon$: 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 $m \leftarrow (c \; mod \; p) mod 2$, 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:

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

$Encrypt_\varepsilon$: 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 $m \leftarrow (c \; mod \; p) mod 2$。

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

Related Work

The concept of homomorphic encryption was first proposed by Rivest et al. In 1978 entitled "On data banks and privacy homomorphic", allowing users to directly perform specific algebraic operations on ciphertexts, obtaining results that the data is still encrypted, Do the same thing and encrypt the result. The public-key cryptosystem was proposed by Diffie et al. [4] in 1976, and different cryptosystems were used to separate encryption and decryption. This laid the foundation for the study of homomorphic encryption, and then many outstanding homomorphic cryptosystems continued to emerge. In 1978, Rivest et al. Constructed a well-known public-key cryptosystem RSA by using number theory. The security of this algorithm depends on the difficulty of large integer decomposition, which has multiplicative homomorphism but does not possess additive homomorphism.

In response to this flaw, Rivest et al. Proposed another Rivest scheme that satisfies the homomorphisms of additive homomorphism and multiplication simultaneously. The security of Rivest also depends on the difficulty of large integers. Experimental results show that this scheme has serious security problems. After some scholars put forward a better MRS algorithm. The first public key cryptosystem based on discrete logarithm difficulties, El Gamal, proposed in 1984, has the property of multiplicative homomorphism [5]. An encryption algorithm GM algorithm that satisfies additive homomorphisms was proposed by Goldwasse and Micali Sex is based on quadratic residual problems; an improved probabilistic homomorphic encryption system was introduced in 1994 by Benaloh [6] Proposed that at present the program has been applied in practice. In 1998, Okamoto and Naccache proposed the OU and NS systems based on additive homomorphism [8, 9], respectively. Both systems can achieve multiple additive homomorphisms.

The first system of additive homomorphic cryptographic cryptography based on the remainder of congruent congruent classes was proposed in 1999 [1], and the system also supports multiple addition homomorphisms. In 2001, Dam-gard et al. [10] promoted the Paillier system and proposed the DJ system. In 2005, the first BGN system that supports simultaneous multiple homomorphisms and one-time multiplicative homomorphisms was proposed by Boneh et al. [11], which is the most recent work from the fully homomorphic encryption scheme. In the same year, domestic scholars also published some achievements in homomorphic cryptography. Our scholars proposed the homomorphic encryption mechanism on the real number range to Guangli et al. [12], but it has not been applied in practice. In the process of exploration, domestic scholars have also applied homomorphic encryption technology to cloud computing, multi-party computing, anonymous access, e-commerce and other fields and have achieved a wide range of achievements. Li et al. [13] contrasted the DGHV and CAFED schemes and proposed an improved homomorphic encryption algorithm. At the same time, the algorithm was used to upload, download, update, delete and retrieve data in the cloud storage environment. In 2013, Peng et al. [14] proposed a general transitive signature scheme based on homomorphic encryption based on the problems of large integer decomposition, discrete logarithm and bilinear math. This scheme supports the features of ciphertext operation , To achieve the general model can be passed signatures and verification. In 2014, Yang et al. [15] proposed a homomorphic encryption scheme to prevent SQL injection attacks, and achieved the requirement information when the important information was kept secret. In the more than thirty years of the concept of homomorphic encryption, various encryption schemes have been put forward. However, most of these schemes are based on semi-homomorphic encryption. Several minority full-homomorphic encryption schemes can not be obtained due to security problems Practical application. After the semi-homomorphic encryption scheme is gradually matured, many scholars begin to study the fully homomorphic encryption scheme.

Conclusion

Full 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. However, there is still a long way to go in the research on the security and practicability of the fully homomorphic encryption scheme. At present, the problems to be solved are mainly shown in the following aspects:

a) How to choose an appropriate homomorphic encryption scheme for efficient retrieval of ciphertext data on the cloud server under the premise of ensuring data security;

b) How to solve the noise problem in the fully homomorphic encryption scheme, which has complicated operation and low operation efficiency;

c) How to make the Homomorphic Encryption scheme more and more practical in the premise of ensuring the security of the Homomorphic Encryption Scheme.

Acknowledgements

This paper was supported by the Scientific Research Project of Hubei Provincial Department of Education, 2018

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]. Van Dijk M, Gentry C, Halevi S. Fully homomorphic encryption overthe integers[C]/ / Advances in Cryptology. Berlin: Springer, 2010: 24-43.

[4]. Diffie W, Hellman M E. New directions in cryptography[J]. IEEETrans on Information Theory, 1976, 22(6) : 644-654.

[5]. Elgamal T. A public-key cryptosystem and a signature scheme based on discrete logarithms[J]. IEEE Trans on Information Theory, 1985, 31(4) : 469-472.

[6]. Ajtai M, Dwork C. A public-key cryptosystem with worst-case equiva-lence[C]/ / Proc of the 29th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 1997: 284-293.

[7]. Okamoto T, Uchiyama S. A new public-key cryptosystem as secure as factoring[C]/ / Advances in Cryptology. Berlin: Springer, 1998: 308-318.

[8]. Naccache D, Stern J. A new public key cryptosystem based on higher residues[C]/ / Proc of the 5th ACM Conference on Computer and Communications Security. New York: ACM Press, 1998: 59-66.

[9]. Xiang Guangli, Chen Xinmeng, Ma Jie, et al. Homomorphic encryptionscheme in the range of the real[J]. Computer Engineering and Applications, 2005, 41(20) : 12-14.

[10]. Damgard I, Jurik M. A generalization, a simplification and some applications of Pailler’s probabilistic public-key system[C]/ / Proc of the 4th International Workshop on Practice and Theory in Public KeyCryptography. Berlin: Springer, 2001: 119-136.

[11]. Boneh D, Goh E J, Nissim K. Evaluating 2-DFN formulas on cipher-texts[C]/ / Proc of the 2nd International Conference on Theory of Cryptography. Berlin: Springer, 2005: 325-341.

[12]. Benaloh J. Dense probabilistic encryption[C]/ /Proc of Workshop on Selected areas of Cryptography. 1994: 120-128.

[13]. Li Jian, Chen Sicong, Song Danjie. Security structure of cloud storage based on homomorphic encryption scheme[C]/ / Proc of the 2nd Inter-national Conference on Cloud Computing and Intelligent Systems. Washington DC: IEEE Computer Society, 2012: 224-227.

[14]. Gentry G. A fully homomorphic encryption scheme[D]. Stanford:Stanford University, 2009.

[15]. Smart N P, Vercauteren F. Fully homomorphic encryption with rela-tively small key and cipher sizes[C]/ / Proc of the 13th InternationalConference on Public Key Cryptography. Berlin: Springer, 2010: 420-443.

[16]. Rivest R,Adleman L, Dertouzos M.On data banks andprivacy homomorphisms [J]. Foundations of SecureComputation,1978,4(11):169-180

[17]. Gentry C.A fully homomorphic encryption scheme [D].Stanford,CA:Stanford University,2009