ANALYSIS OF HOMOMORPHIC ENCRYPTION METHODS IN CLOUD COMPUTING SYSTEMS

Karimov Abdukodir Abdisalomovich
Tashkent University of Information Technologies, teacher
@mail: karimovabduqodir041@gmail.com

Olimov Iskandar Salimboyevich
Tashkent University of Information Technologies, teacher
@mail: iskandar.olimov@mail.ru

ABSTRACT: This article discusses gamma encryption in detail. In addition, a comparative analysis of commonly used types of gamma encryption, i.e. partially homomorphic and fully homomorphic encryption, is presented. RSA, El-Gamal, Paillier, Goldwasser-Micali, Boneh-Goh-Nissim and Gentry Homomorphic encryption systems were analysed and the results presented.

Keywords: Homomorphic encryption, RSA algorithm, Either addition or multiplication, Data privacy, transferred to the commercial cloud.

1. INTRODACTION

Homomorphic encryption is a form of encryption that allows computation with encrypted texts, i.e. the result of an open text data operation is the same as the result of another encrypted text operation.

Exposing sent encrypted texts to cloud computing systems without decrypting them saves time and money. This means that the use of homomorphic encryption systems is highly efficient [1].

Homomorphic encryption can be used to store external resources and ensure confidentiality of computing. This allows data to be encrypted and transferred to the commercial cloud for processing.

In highly regulated industries, such as healthcare, homomorphic encryption can be used to provide new services by removing privacy barriers to data sharing. Predictive analytics in healthcare, for example, is difficult to implement because of privacy concerns about medical data, but these privacy concerns are mitigated if the predictive analytics provider works with encrypted data.

Homomorphic encryption is a form of encryption with the added ability to compute encrypted data without the use of a secret key. The result of this calculation remains encrypted. Homomorphic encryption can be seen as an extension of symmetric or public key cryptography. Homomorphism refers to homomorphism in algebra: the encryption and decryption functions can be seen as a homomorphism between the plaintext and the encrypted text.

2. TYPES OF HOMOMORPHIC CRYPTOLOY

Homomorphic encryption includes several types of encryption schemes that can perform various calculations on the encrypted data. Some common types of homomorphic encryption are partially homomorphic, partially homomorphic, level fully homomorphic and fully homomorphic. Calculations are represented as both logical and arithmetic cycles. Partially homomorphic encryption involves schemes that support evaluation when only one type of operation, such as addition or multiplication, is involved. Some homomorphic encryption schemes can evaluate two types of operations. Gradient fully homomorphic encryption supports limited (predefined) operations. Fully homomorphic encryption can be applied to all operations and is a fully implementable version of homomorphic encryption [2].

3. PARTIAL HOMOMORPHIC ENCRYPTION

Using the notations of the publicly available data encryption function, we can write: If the plaintext x is encrypted with public key e in the RSA algorithm. If $\varepsilon(x) = m^e modm$, then the homomorphism property is:

$\varepsilon(x_1) * \varepsilon(x_2) = x_1^e x_2^e modm = (x_1 x_2)^e modm = \varepsilon(x_1 * x_2)$.

In the El-Gamal algorithm. In this cryptosystem, a cyclic group G is equal to a public key (G,q,g,h) when its order is q and its basis is g. Here $h = g^x$ and x is the secret key. In this case, the data encryption function m is $\varepsilon(m) = (g^r, m * h^r)$ for $r \in \{0, \dots, q-1\}$. Homomorphic property for this algorithm:

$\varepsilon(m_1) * \varepsilon(m_2) = (g^{r_1} m_1 * h^{r_1})$=$(g^{r_2}, m_2 * h^{r_2}) = (g^{r_1+r_2}, (m_1 * m_2) h^{r_1+r_2}) = \varepsilon(m_1 * m_2)$.

1. Key generation
a) Choose a prime number q (e.g. 37)
b) Choose a prime number g:g<q (e.g. 19)
c) Choose a random number x:1<x<q-1 (e.g. 5)
d) Calculate h=gx mod qh=g^x modq - our random number, 〖19〗 ^5 mod37=22

Public key: kp=(q,g,h) (we have: (37, 19, 22))
Private key: ks=x (we have: 5)

2. Encryption with key $kp = (q, g, h)$ m to encode a number (e.g. 3)
a). A random number $y: 1 < y < q - 1$ (e.g. 12) is chosen.

б). Calculate $c_1 = g^y modq$ and $c_2 = m * h^y modq$ (in our case $c_1 = 10$ and $c_2 = 4$)

Then m number of ciphers: $E(m, kp) = (c_1, c_2)$ In our case, after encryption 3 equals: (10, 4)

3. Decrypt with key ks=x

Decrypt cryptogram (c_1, c_2) (in our case (10,4))

Calculate $m = D\big((c_1, c_2), x\big) = c_2 * c_1^{q-1-x} modq$

(In our case, $4 * 10^{37-1-5} mod37 = 4 * 10^{31} mod37 = 3$!!!))

In the Goldwasser-Micali algorithm. In this algorithm, if the public key for given numbers $r \in$ {0, …., m-1} consists of modulo m and a non-quadratic residue x, then the encryption function of bit b is $\varepsilon(b) = x^b r^2 modm$.

Homomorphic property for this algorithm:

$$\varepsilon(b_1) * \varepsilon(b_2) = (x^{b_1} r_1^2\, x^{b_2} r_2^2) modm = (x^{b_1+b_2} (r_1 r_2)^2) modm = \varepsilon(b_1 \oplus b_2).$$

In Benalo‘s algorithm. In this algorithm, if the public key is m modulo and the base is g, then for given numbers $r \in \{0, \dots, m-1\}$ the message encryption function x is $\varepsilon(x) = g^x r^c modm$. Then the homomorphism property is equal to:

$$\varepsilon(x_1) * \ \varepsilon(x_2) modm = (g^{x_1} r_1^c)(g^{x_2} r_2^c) modm = g^{x_1+x_2} (r_1 r_2)^c = \varepsilon(x_1 + x_2 modm$$

In the Pailer algorithm. In this algorithm, if the public key consists of a modulus m and a base g, for given numbers $r \in$ {0, …., m-1} the cipher function of the message x is $\varepsilon(x) = g^x r^m modm^2$ will Then the homomorphism property is:

$$\varepsilon(x_1) * \varepsilon(x_2) = (g^{x_1} r_1^m) g^{x_2} r_2^m) modm^2 = g^{x_1+x_2} (r_1 r_2)^m modm^2 = \varepsilon(x_1 + x_2)$$

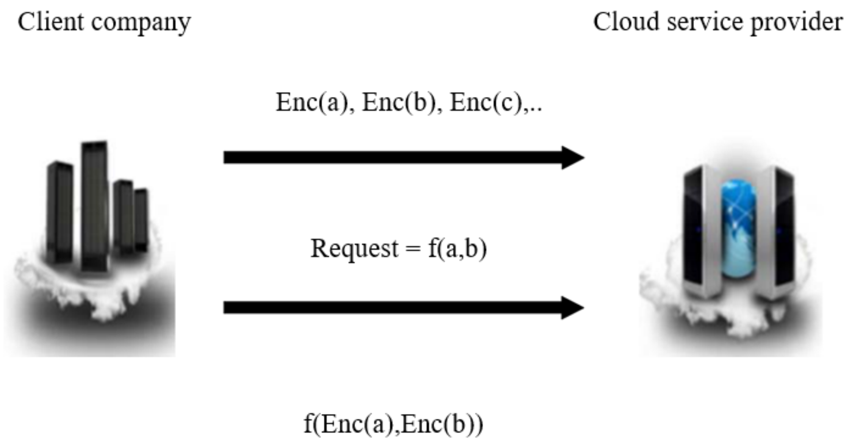
Multiplicative homomorphic encryption (based on RSA cryptosystem)

If the numbers p and q are prime, then let n=p*q. The numbers e and d satisfying the equality $e * d \equiv 1 (mod\varphi)(n)$ are defined. While n and e are public keys, d is a private key. In this case, in RSA algorithm, if encryption parity is $C = M^e modn$, decryption parity is $M = C^e modn$.

Homomorphism: if x_1 and x_2 are open texts, then.

$$\varepsilon(x_1) * \varepsilon(x_2) = (x_1^e x_2^e) modm = (x_1 x_2)^e modm = \varepsilon(x_1 * x_2)$$

This homomorphic encryption only uses the multiplication operation. Full gamma encryption is required to perform all types of computation in the cloud. In 2009, IBM offered a fully homomorphic encryption system known as Gentry. This method performs a variable number of additions and multiplications and can therefore perform any type of encryption on the ciphertext. The security of cloud computing systems requires software tools capable of fully homomorphic encryption that prevent the disclosure of sensitive information between different nodes. A general overview of the



use of full homomorphic encryption in cloud computing systems is shown in Figure 1 [3].

Figure 1: Application of fully homomorphic encryption in cloud computing

Table 1. Analysis of partially homomorphic and fully homomorphic encryption [4]

Parameter	Partial HE	Fully HE
Type of operation	Either addition or multiplication	Both
Computation	Limited number of computations	Unlimited
Computational efforts	Requires less effort	Requires more effort
Performance	Faster and more compact	Slower
Versatility	Low	High
Ciphertext size	Small	Large
Example	Unpadded RSA, ElGamal	Gentry Scheme

In general, a comparative analysis and characteristics of common homomorphic encryption algorithms are presented in Table 1 [5].

Table 2. Comparative analysis and characteristics of homomorphic encryption algorithms

Features	*Homomorphic encryption systems*					
	RSA	Paillier	El-Gamal	Goldwass er-Micali	Boneh- Goh-Nissim	Gentr y
Platform	Cloud computing					
Homo-morphic encryption type	Multiplica tion	additio n	Multiplicati on	addition, Just one bit encrypts	Not limited number addition but one multiplicatio n	full
Data privacy	Provided in communication and storage processes					
Safety is used	Cloud service provider					
Keys are used	Clients (different keys are used for encryption and decryption)					

4. CONCLUSIONS

Homomorphic encryption algorithms are analysed and the following results are obtained:

- An analysis of types of homomorphic encryption algorithms is performed;
- The homomorphic encryption in El-Gamal algorithm is implemented;
- The property of homomorphism for Goldwasser-Micali algorithm is considered;
- The homomorphism property of Benalo algorithm is considered;
- Multiplicative homomorphic encryption is calculated based on RSA cryptosystem.

5. REFERENCES

1. Saja, J.M. 2021. Privacy preserving algorithm using Chao-Scattering of partial homomorphic encryption. Journal of Physics Conference Series.
2. Khowla, K.M., Usman, B., Rimsha, K.M. 2022. Homomorphic Encryption Technologies for Cloud Computing. The 8th International Conference on Next Generation Computing. Republic of Korea.
3. Waleed T., Al-Sit1, Hani Al-Zoubi. 2019. Cloud Security based on the Homomorphic Encryption. International Journal of Advanced Computer Science and Applications.
4. https://www.researchgate.net/figure/Homomorphic-encryption-schemes_tbl5_352211017
5. https://crypto.stanford.edu/craig/craig-thesis.pdf