2023 17th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-5348-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IMCOM56909.2023.10035630

A Study on Partially Homomorphic Encryption

Jihyeon Ryu
Department of Computer Science and Engineering
Sungkyunkwan University
Suwon, Korea
jhryu@security.re.kr

Keunok Kim
Department of Electrical and Computer Engineering
Sungkyunkwan University
Suwon, Korea
kimkeunok@gmail.com

Dongho Won*
Department of Computer Science and Engineering
Sungkyunkwan University
Suwon, Korea
dhwon@security.re.kr

***Abstract*—Recently, data experts can obtain a large amount of data with the development of the Internet. When computing such data, cloud services that do not use the personal device's memory are becoming popular. However, storing sensitive data as a source in the cloud carries the risk of hijacking. To compensate for this, homomorphic encryption, which encrypts and stores sensitive data, and can safely operate in an encrypted state, is being studied. In this paper, we analyze four methods of partially homomorphic encryption among homomorphic encryption methods. We compare and analyze the key size and ciphertext size of four partially homomorphic encryptions, Paillier, ElGamal, ASHE, and Symmetria.**

***Index Terms*—Homomorphic Encryption, Cryptosystem, Security Strength**

I. Introduction

Nowadays, a lot of data is available on the Internet, and the ways in which this data is statistically processed and streamlined are increasing. In particular, as such data increases, the demand for cloud computing that uses external computing power rather than personal devices is increasing. However, large amounts of data can also contain sensitive data.

Storing sensitive data in the cloud raises privacy concerns. In general, if it is encrypted and stored in the cloud, calculations cannot be performed in the cloud. In general, encrypted data needs to be decrypted, calculated, and then encrypted again. At this time, the data is not protected in the decrypted state. A method to compensate for this is the homomorphic encryption method. A homomorphic encryption method has been proposed to protect data in the operating state.

Homomorphic encryption is divided into Fully Homomorphic Encryption (FHE) and Partially Homomorphic Encryption (PHE) methods. FHE can perform all addition and multiplication operations in an encrypted state. Conversely, PHE represents a system that allows for one addition or multiplication operation.

Because FHE is computationally expensive, it cannot be used in devices with small memory. Also, since the operation speed is slow, it cannot be used even when fast speed is important. Classic and well-used PHEs such as ElGamal [1], Benaloh [2], and Paillier [3] have more reasonable computational speed. However, the size of their key and the size of the ciphertext are important issues for devices with small memory.

We describe the size of the secure key pair and the size of the ciphertext for the use of various partial homomorphic encryption methods. We compare the PHE of the four methods Paillier [3], ElGamal [1], ASHE [4], and Symmetria [5]. We analyze their security strength and show that they can work safely on small devices.

We describe the paper as follows. Section II introduces the typical encryption method. In section III, we describe the types of partial homomorphism ciphers. Section IV analyzes the key pair size and ciphertext size suitable for partially homomorphic encryption, and we conclude in Section V.

II. Preliminaries

In this section, we introduce popular encryption algorithms that use similar difficulties to partially homomorphic encryption. Representative encryption algorithms include RSA [6], ECC [7], and AES [9].

A. RSA

The RSA cipher is a cryptography named after the author, created by Ron Rivest, Adi Shamir, and Leonard Adleman in 1978 [6], and is one of the public key cryptography systems. RSA is known as the first algorithm capable of digital signature as well as encryption. The digital signature function of RSA is used for e-commerce that requires authentication. It is an encryption method based on the fact that the product of two large numbers is difficult to factorize.

When there are public key $n = p * q$ (where p and q are large prime numbers) and e, private key d (such that $d * e = 1 \mod \phi(n)$), and a plaintext message m, RSA encryption and decryption works as follows.

1) Encryption

$$c = m^e \mod n \tag{1}$$

2) Decryption

$$m = c^d \mod n \tag{2}$$

This work was supported by an Institute of Information & Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00558, Development of National Statistical Analysis System using Homomorphic Encryption Technology)

978-1-6654-5348-6/23/$31.00 ©2023 IEEE

B. ECC

Elliptic curve cryptography (ECC) is a public key cryptography based on elliptic curve theory. The encryption method using elliptic curves was independently proposed by Neil Koblitz and Victor Miller in 1985 [7], [8]. The most representative advantage of elliptic curve cryptography over existing public key cryptography such as RSA and Elgamal cryptography is that it provides a similar level of security while using a shorter key. Due to these advantages, it is currently used as a cryptographic method of the Bitcoin blockchain. It is an encryption method based on the fact that the product of two large numbers is difficult to factorize.

Basically, the ECC encryption method operates on the coordinate plane $y^2 = x^3 + ax + b$, which is the x value and the y value of the elliptic curve, where a and b are constant, and satisfy the equation $4a^3 + 27b^2 \neq 0$. At this time, the private key d is generated with a random number generator with a prime number smaller than P, and the public key Q is $Q(x, y) = dG(x_0, y_0)$, that is, by addition operation on the elliptic curve. When we know G and Q it is difficult to know the value d. Given plaintext m, ECC encryption and decryption works as follows.

1) Encryption

$$generate\ random\ value\ k \tag{3}$$

$$public\ key\ Q = dG \tag{4}$$

$$y_1 = kG \tag{5}$$

$$y_2 = m + kQ \tag{6}$$

$$c = (y_1, y_2) \tag{7}$$

2) Decryption

$$m = y_2 - (d * y_1) \tag{8}$$

$$= y_2 - (d * (kG)) \tag{9}$$

$$= m + kQ - k * (dG) \tag{10}$$

C. AES

AES stands for Advanced Encryption Standard and is an encryption method established by the National Institute of Standards and Technology (NIST) in 2001 [9]. On November 26, 2001, AES was promulgated as the US Federal Information Processing Standard (FIPS-197), and it is widely used worldwide due to its high safety and speed. Due to these advantages, it is also widely used for ransomware. It consists of a block cipher format consisting of the following four steps.

- Substitute Bytes (SB): Blocks are exchanged in the form of Byte units using S-box.
- Shift Rows (SR): Shifts one row to another.
- Mix Columns (MC): Changes all bytes in a column by replacing each Byte in a column using a circular matrix.
- Add Round Key (ARK): XORs a part of the extended key and the current block by bit.

For encryption, we calculate 9 rounds of the above four steps, and the last round performs the remaining operations except for the Mix Columns operation.

When decryption, we use the Inverse S-box in the SB stage and use the ARK key in the reverse order. SR is also substituted in the opposite direction, and the inverse of the MC recursive matrix is used.

III. Partially Homomorphic Encryption

Although there are various partial homomorphic encryption methods, in this section we discuss Paillier's encryption [3] and ElGamal [1], which are representative partial homomorphic encryption, and the recently proposed ASHE [4] and Symmetria [5]. The details are as follows.

A. Paillier

The Paillier cryptosystem is an asymmetric encryption algorithm for public key cryptography invented and named after Pascal Paillier in 1999 [3]. Additive homomorphic encryption is possible and was first proposed in Eurocrypt. The cryptosystem is an additive homomorphic encryption. A secret key was conceived using the Carmichael function, and $[[w]]$ was defined in [2].

The public key $n = p * q$, and the private key λ means $\lambda(n)$ to which the Carmichael function λ is applied. At this time, assuming that g has a non-zero multiple of n as its order, encryption and decryption can be performed as follows.

1) Encryption

$$C = g^m r^n \qquad mod\ n^2 \tag{11}$$

2) Decryption

$$m = L(C_\lambda)/L(g_\lambda) \qquad mod\ n \tag{12}$$

such that $C_\lambda = C^\lambda\ mod\ n^2$, $g_\lambda = g^\lambda\ mod\ n^2$ and $L(u) = (u - 1)/n$.

B. ElGamal

ElGamal encryption is a public key encryption method based on Diffie-Hellman key exchange, devised in 1985 by Taher ElGamal [1]. It is an encryption method based on the problem that it is difficult to obtain x of the discrete log equation $\beta = \alpha^x$ when α and β are known.

The public key is prime number p, α, and β, where $\beta = \alpha^x$ is satisfied. x becomes the private key. Assuming that the message is m, ElGamal's encryption and decryption methods are as follows.

1) Encryption

$$Select\ random\ integer\ k \in [0, p-1] \tag{13}$$

$$c_1 = \alpha^k \qquad mod\ p \tag{14}$$

$$c_2 = \beta^k m \qquad mod\ p \tag{15}$$

2) Decryption

$$s = c_1^x = \beta^k \qquad mod\ p \tag{16}$$

$$m = c_2 s^{-1} \qquad mod\ p \tag{17}$$

TABLE I
COMPARABLE STRENGTHS [14]

Security strength	Symmetric key algorithms	DSA	RSA	ECC
≤ 80	2TDEA	L = 1024, N = 160	k = 1024	f = 160 - 223
112	3TDEA	L = 2048, N = 224	k = 2048	f = 224 - 255
128	AES-128	L = 3072, N = 256	k = 3072	f = 256 - 383
192	AES-192	L = 7680, N =384	k = 7680	f = 384 - 511
256	AES - 256	L = 15360, N = 512	k = 15360	f = 512+

1) EC-ElGamal: EC-ElGamal is the ElGamal method using the difficulty of elliptic curve cryptography. When the public key is $A = aP$, the private key is a, and the message is m, encryption and decryption are performed as follows.

1) Encryption

$$Select\ random\ integer\ k \tag{18}$$

$$K = kP \tag{19}$$

$$C = kA + m \tag{20}$$

2) Decryption

$$S = aK \tag{21}$$

$$m = C - S \tag{22}$$

2) CRT-ElGamal: CRT-ElGamal is the ElGamal method using the Chinese Remainder Theorem [11], [12]. n is prime, and $h = g^r \mod n$ is public key, r is private key, for generator g and $i = 1, ..., t$, d_i that $gcd\ (d_i, d_j) = 1$ is satisfied, and the generator and d_i are public. When $m = \{m_1, ..., m_t\}$ is given as a message, encryption and decryption are performed as follows.

1) Encryption

$$Select\ random\ integer\ k_i \quad i = 1, ..., t \tag{23}$$

$$C_{1_i} = g^{k_i} \mod n \tag{24}$$

$$C_{2i} = h^{k_i} g^{m_i} \mod n \tag{25}$$

2) Decryption

$$m_i = CRT^{-1}[(log_g * C_{2i} C_{1i}^{-r} \mod n)] \tag{26}$$

$$that\ CRT^{-1}[C_i] = \sum_{i=1}^{t} C_i \frac{d}{d_i} (\frac{d}{d_i}^{-1} \mod d_i) \bmod d \tag{27}$$

C. ASHE

ASHE, which claims that the speed of additive homomorphic encryption is faster than Paillier's encryption system, is an acronym for a new additively symmetric homomorphic encryption scheme [4]. It was proposed at the 2016 OSDI conference by Papadimitriou et al.

$F_k(x)$ is a pseudo random function that uses k and x as inputs, and it is assumed that the set I is mapped to Z_n. Encryption and decryption of ASHE is performed as follows.

1) Encryption

$$Enc_k(m, i) = ((m - F_k(i) + F_k(i-1)) \mod n, \{i\}) \tag{28}$$

2) Decryption

$$Dec_k(c, S) = (c + \sum_{i \in S} (F_k(i) - F_k(i-1))) \bmod n \tag{29}$$

D. Symmetria

Symmetria is a symmetric encryption method opposite to Paillier, ElGamal, and ASHE described above [5]. Symmetria proposed in VLDB 2020 conference by Savvides et al., is created considering both symmetric additive homomorphic encryption (SAHE) and symmetric multiplicative homomorphic encryption (SMHE), and is more cost-effective than the Paillier encryption.

$F_k(x)$ is a pseudo random function that uses value k and x as inputs, and it is assumed that the set I is mapped to Z_n. Symmetria SAHE encryption and decryption are performed as follows.

1) Encryption

$$c = (m + F_k(r)) \mod N,\ [r],\ \phi \tag{30}$$

At this time, the form of c is $\{v, I_p, I_n\}$.

2) Decryption

$$m = (v + \sum_{r_1 \in I_p} F_k(r_1) + \sum_{r_2 \in I_n} F_k(r_2)) \mod N \tag{31}$$

$F_k(x)$ is a pseudo random function that uses value k and x as inputs, and it is assumed that the set I is mapped to Z_n. Symmetria SMHE encryption and decryption are performed as follows.

1) Encryption

$$c = (m * g^{F_k(r)}) \mod N,\ [r],\ \phi \tag{32}$$

At this time, the form of c is $\{v, I_p, I_n\}$.

2) Decryption

$$m = (v * \prod_{r_1 \in I_p} g^{-F_k(r_1)} + \prod_{r_2 \in I_n} g^{F_k(r_2)}) \mod N \tag{33}$$

IV. ANALYSIS OF PHE

We estimate the secure key size and ciphertext size based on FIPS [13], NIST 800–57 [14] for the introduced PHEs, Paillier, ElGamal, ASHE, and Symmetria. On this basis, we

can choose a PHE that can be used even for low memory devices.

Paillier says that encryption is not secure when we know p and q of $n = p * q$. This is similar to RSA safety. Since EC-ElGamal and CRT-ElGamal are used based on large prime numbers, they are the basis for choosing the key size similar to the safety of ECC. As for both ASHE and Symmetria, are similar to AES, safety increases in proportion to the size of n. Table I provides details.

V. Conclusion

In this paper, we analyzed a partially homomorphic encryption for use when performing an encryption operation in the cloud.We introduced the four methods of partial homomorphism, Paillier, Elgamal, ASHE, and Symmetria, compared their security strength. The results of our research can also be used to prevent path tracing by making it work by encrypting it when operating on low-memory semiconductors.

Acknowledgment

This work was supported by an Institute of Information & Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00558, Development of National Statistical Analysis System using Homomorphic Encryption Technology)

References

[1] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms." IEEE transactions on information theory, 31(4), pp. 469–472, 1985.

[2] J. D. C. Benaloh, "Verifiable secret-ballot elections." Yale University, 1987.

[3] P. Paillier. "Public-key cryptosystems based on composite degree residuosity classes." In International conference on the theory and applications of cryptographic techniques, pp. 223–238, 1999.

[4] A. Papadimitriou, R. Bhagwan, N. Chandran, R. Ramjee, A. Haeberlen, H. Singh, A. Modi, S. Badrinarayanan, "Big data analytics over encrypted datasets with seabed." In 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 587–602, 2016.

[5] S. Savvides, D. Khandelwal, P. Eugster, "Efficient confidentiality-preserving data analytics over symmetrically encrypted datasets." Proceedings of the VLDB Endowment, 13(8), pp. 1290–1303, 2020.

[6] R. L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM, 21(2), pp. 120–126, 1978.

[7] N. Koblitz, "Elliptic curve cryptosystems." Mathematics of computation, 48(177), pp. 203–209, 1987.

[8] V. S. Miller, "Use of elliptic curves in cryptography." In Conference on the theory and application of cryptographic techniques, pp. 417–426, 1985.

[9] J. Daemen, V. Rijmen, "AES proposal: Rijndael". 1999.

[10] N. Koblitz, "Elliptic curve cryptosystems." Mathematics of computation, 48(177), pp. 203–209, 1987.

[11] M. T. Ibn Ziad, A. Alanwar, Y. Alkabani, M. W. El-Kharashi and H. Bedour, "Homomorphic Data Isolation for Hardware Trojan Protection." IEEE Computer Society Annual Symposium on VLSI, pp. 131–136, 2015.

[12] Y. Hu, W. J. Martin, B. Sunar, "Enhanced flexibility for homomorphic encryption schemes via CRT." In Applied Cryptography and Network Security (ACNS), 2012.

[13] C. F. Kerry, C. R. Director, "Federal Information Processing Standards Publication (FIPS PUB) 186–4" Digital Signature Standard (DSS), 2013.

[14] E. Barker, Q. Dang, "Recommendation for Key Management Part1: General."NIST Special Publication 800–57 part 1, revision 4. NIST, Tech. Rep, 16. 2016.