The Homomorphic Subtraction and Division Property of Paillier Cryptosystem

Tanyaporn Sridokmai[1],
Somchai Prakancharoen[2],
and Nalinpat Porrawatpreyakorn[3]
[1]ladytanyaorm@gmail.com, www.kmutnb.ac.th
[2]spk@kmutnb.ac.th, www.kmutnb.ac.th
[3]nalinpatp@kmutnb.ac.th, www.kmutnb.ac.th

Abstract - **In this paper, one example of Paillier's encryption schemes and homomorphic encryption was illustrated. In mathematical details, Subtraction, Multiply, Division binary operation of binary based integer number operands was presented. In particular, the secrecy of encryption and decryption will be shown. Both operands were still encrypted even through another operation was processing.**

Keywords - **Cryptosystem, Paillier Encryption, Homomorphic Encryption**

I. INTRODUCTION

Cryptanalysis is the flip-side of cryptography: it is the science of cracking codes, decoding secrets, violating authentication schemes, and in general, breaking cryptographic protocols. One of the main motivations for a threshold cryptosystem is that it allows one to construct a third-party decryption service in a distributed, secure, without a significant increase in the size or the cost of creating a cipher text compared to a standard cryptosystem. To be at all useful, the third party should not decrypt everything that comes its way and give it to just anybody, but should implement some kind of useful decryption policy. To implement such a policy securely, in addition to selected cipher text security, one needs an additional facility: the ability to attach a label to the cipher text during the encryption process. Such a label is a bit string that contains information that can be used by the third party to determine if the decryption request is authorized, according to its policy and its current state. One can think of the label as being a part of the cipher text, so that changing the label changes the cipher text; security against selected cipher text attack would then imply, in particular, that one cannot subvert the third party's policy by simply swapping labels. RSA and ElGamal are representatives of two different types of asymmetric cryptosystem classes. In 1999, Pascal Paillier [1] proposed an additional, different class of asymmetric cryptosystems. Paillier is a public key cryptosystem which offers an additive homomorphism, making it very useful for privacy preserving applications.

Homomorphic encryption is a format of encryption that allows computations to be carried out on cipher text, so generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. The notion of homomorphic encryption was first introduced by Rivest, Adleman and Dertouzos [2] and was first instantiate for restricted classes of functions, like addition or multiplication.

This paper focus Paillier's work beginning to homomorphic encryption by showing how to encrypt and decrypt messages using this cryptosystem, with the underlying mathematical principles that make the system work clearly outlined.

II. THE PAILLIER CRYPTOSYSTEM

The Paillier Cryptosystem is a modular, public key encryption scheme, created by Pascal Paillier [5, 6, 7].

A. Properties of Scheme

Paillier scheme is the most known, and maybe the most efficient partially homomorphic encryption scheme. Although Paillier interestingly comes back to a standard RSA modulus $N = pq$, it is the direct continuation of the work of Okamoto and Uchiyama. In the following description, $\lambda(N)$ denotes the Carmichael function (in our case, $\lambda(N) = lcm(p-1, q-1)$), and $L_N(\bullet)$ denotes the function defined over the set $SN = \left\{x < N^2 \mid x = 1\, mod\ N\right\}$ as $LN(x) = (x-1)/N.$

B. Threshold Paillier Cryptosystem

A recent popular public key cryptosystem is a system proposed by Paillier [1] in 1999. The Paillier cryptosystem is based on properties of Carmichael function in Z_{N^2}. Security of the cryptosystem is based on the intractability of computing discrete logarithms in Z_{N^2} without the Carmichael number $\lambda(N)$. Below are the steps for the key generation, encryption and decryption used

- **Using The Paillier Cryptosystem**

The following is the description of the Paillier steps that have been produced from [4]:

- **Steps for Key Generation**

1. Choose two large prime numbers p and q randomly and independently of each other.

2. Compute the modulus n ; the product of two primes $n = p.q$ and (λ is Carmichael's function)

3. Select random integer g where $g \in \mathbf{Z}^*_{n^2}$ and g's order is a non-zero multiple of n (since $g = (1+n)$ works and is easily calculated - this is the best choice).

4. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse: $u = L(g\lambda\ modn^2)^1 mod\ n$, where function L is defined as (Lagrange function) $L\ (u) = u-1\backslash\ n$ for $u = 1\, mod\ n$

- The public (encryption) key is $(n,\ g)$.
- The private (decryption) key is (λ).

- **Steps for Encryption**

1. Plaintext is m where $m < n.$
2. Find a random r
3. Let cipher text $c = g^m.\, r^n mod\ n^2.$

- **Steps for Decryption**

1. The cipher text $c < n^2.$
2. Retrieve plaintext
   $m = L\left(c\lambda\ modn^2)/L(g\lambda\ modn^2\right)mod\ n$

Or in same expression
   $m = L\left(c\lambda\ mod\ n^2\right).\ u \bmod n$

Looking at the form of the equation, one will see that the decryption will be done by first removing the random part r^n, then retrieving the exponent $m\ mod\ n^2.$

III. COMPUTATION OF PAILLIER CRYPTOSYSTEM

Example1: Step of paillier cryptosystem

Key Generation

1. p = 5, q = 7,
2. n = pq = 35, n^2 = 1225
3. φ(n) = (p-1)(q-1) = 24,
4. λ(n) = lcm(p-1,q-1) = 12
5. n = 187 , n^2=34969
6. $g = (\alpha n+1)*\beta^n modn2$ = 36. $(\alpha = 1, \beta = 1)$
7. Remember L(μ) = μ -l\n so we also calculate L = $\mu = 1/L(g^\lambda \bmod n^2) \bmod n$ = 3

We get the public key (n,g) = (35, 36) and the private key (p, q, λ) = (5 , 7, 12).

Encryption Step:

1. Plaintext m = 4.
2. r = 2 $\in Z^*_n$
3. Let cipher text c = 36^4 • 2^{35} mod 1225 = 421

Decryption Step:

1. The cipher text 421 < 1225.
2. m = L(421^{12} mod 1225)\ 1 mod 35
3. m = 141\ 1 mod 35.
4. m = 141• 1 mod 35= 4.

IV. HOMOMORPHIC ENCRYPTION

The concept of privacy homomorphism was first introduced by Rivest et. al. [2], that defined privacy homomorphisms as encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands. According to the correspondence between the operation in the cipher text domain and the operation in the plaintext domain, a cryptosystem can be additively homomorphic or multiplicatively homomorphic: in this paper we are interested in the former. Multiplicatively homomorphic cryptosystems allow, in fact, to perform additions, subtractions and multiplications with a known (non-encrypted) factor in the encrypted domain. More extensive processing would be allowed by the availability of an algebraically homomorphic encryption scheme, that is a scheme that is additive and multiplicative homomorphic.

The most common definition is the following. Let M (resp., C) denote the set of the plaintexts (resp., cipher texts).

An encryption scheme is [8] said to be homomorphic if for any given encryption key k the encryption function E satisfied

$$\forall m_1, m_2 \in M,\ \ E(m_1 \square_M m_2) \leftarrow E(m_1) \square_C E(m_2)$$

for some operators $\square_M$ in M and $\square_C$ in C, where means "can be directly computed from," that is, without any intermediate decryption. If (M, $\square_M$)and (C, $\square_C$) are groups ,we have a *group homomorphism*. We say a scheme is *additively homomorphic* if we consider addition operators, and *multiplicatively homomorphic if we* consider multiplication operators.

A lot of such homomorphic schemes have been published that have been widely used in many applications. Note that in some contexts it may be of great interest to have this property not only for one operator but for two at the same time.

Hence, we are also interested in the design of ring/algebraic homomorphisms. Such schemes would satisfy a relation of the form

$$\begin{aligned} \forall m_1, m_2 \in M,\quad & E(m_1 +_M m_2) \leftarrow E(m_1) +_C E(m_2), \\ & E(m_1 \times_M m_2) \leftarrow E(m_1) \times_C E(m_2) \end{aligned}$$

As it will be further discussed, no convincing algebraic homomorphic encryption scheme has been found yet, and their design remains an open problem. Less formally, these definitions mean that, for a field key k, it is equivalent to perform operations on the plaintexts before encryption or on the corresponding ciphertexts after encryption. So we require a kind of commutatively between encryption and some data processing operations. Of course, the schemes we will consider in the following have to be probabilistic ciphers, and we may consider E to behave in a probabilistic way in the above definitions.

V. COMPUTATION

A. The Subtraction Homomorphic Property of the Paillier Cryptosystem

Using by property of additively homomorphic

Encryption:

$$\begin{aligned} c_1 \cdot c_2 &= g^{m_1} x_1^n {r_1}^n \cdot g^{m_2} x_2^n {r_2}^n \bmod n^2 \\ &= g^{m_1+m_2} (x_1 x_2)^n \bmod n^2 \end{aligned}$$

Example2:

For the example we use the following Paillier public key

$$p = 5,\ q = 7,\ n = pq = 35,\ n^2 = 1225,$$
$$\varphi(n) = (p-1)(q-1) = 160,\ \lambda(n) = (p-1, q-1) = 12$$
$$(r_1, r_2) = (2,36),\ g = (\alpha n + 1)\beta^n \bmod n^2 = 36$$
$$(\alpha = 1,\ \beta = 1),\ \mu = 3$$

And say that, for example, we want to encrypt the two messages

(m_1,m_2) = 4, 16.

c_1 =88, c_2 =23

$$c_1 \cdot c_2 = g^{m_1} x_1^n r_1{}^n \cdot g^{m_2} x_2^n r_2{}^n \bmod n^2$$

$$= g^{m_1+m_2}(x_1 x_2)^n \bmod n^2$$

$c_1 \equiv 88\, c_2 \equiv 23$

≡ (88,*(-23)) mod 1225

≡ -799

Decryption:

$m = L\left(c^{\lambda} \bmod n^2\right) \cdot \mu \bmod n$

≡ (-779^{12} mod 1225)* μ mod35

≡ 3

B. The Multiplicatively Homomorphic Property of the Paillier Cryptosystem

The given ciphertexts are valid encryptions of plaintexts $m_i, = Enc(m_i) = g^{m_i} r^{n_i} \bmod n^2$ .The following properties hold.

Raising an encrypted message to the power of a second message results in the multiplication of plaintext messages.

Encryption:

$$E(m_1, pk)^{m_2} = g^{m_1 \times m_2} \left(r_1^{m_2}\right)^n \left(mod\ n^2\right)$$

$$\in Enc(pk, m_1 \times m_2 \bmod n)$$

Next, begin to decrypt

Decryption:

$$D\left[E_{x_1}\left(m_1\right)^{m_2} \bmod n^2\right] = \left(m_1 m_2\right) \bmod n$$

Example3:

The according cipher texts are multiplicatively homomorphic

Encryption:

$$E(m_1, pk)^{m_2} = (g^{m_1} r_1^{\ n})^{m_2} \left(mod\ n^2\right)$$

$$= g^{m_1 m_2} \left(r_1^{m_2}\right)^n \left(mod\ n^2\right)$$

$$= E(m_1 m_2, pk)$$

c_1 = 88, c_2 =23

Computing $E\left(m^1\right) m^2$:

≡ 88^{16}mod 1225

≡ 991^{12} mod 1225

≡ 1156

≡ L(1156)

≡ 1156-1/35

≡ 33 mod 35

Next, begin to decrypt $E\left(m^1\right) m^2$ by applying the function *L(u)* to the previously attained value:

Decryption:

$D(E(m_1, pk)^{m2}(mod\ n^2)) = m_1 m_2 \left(mod\ n\right)$

= 33 *17

= 561

C. The Division Homomorphic Property of the Paillier Cryptosystem

Using multiplicatively homomorphic property and property of Modular Multiplicative Inverse

Example 4:

Encryption:

$m_i, = Enc(m_i) = g^{m_i} r^{n_i} \bmod n^2.$

Since $E\left(m_1\right)$ is still 223 and $E\left(m_2\right) = 23$

$= c_1 \equiv 223, c_2 \equiv 23$

≡ 223^{16}mod 1225

≡ 386^{12} mod 1225

≡ 946

≡ L(946)

≡ 946-1/35

≡ 27 mod 35

Next, begin to decrypt $E\left(m^1\right) m^2$ by applying the function *L(u)* to the previously attained value:

Decryption:

$D(E(m_1, pk)^{m2}(mod\ n^2)) = m_1 m_2 \left(mod\ n\right)$

≡ 27 *3 mod 35

Modular Multiplicative Inverse

≡ 16

The Ciphertext it the same of plantext.

III. CONCLUSIONS

Homomorphic cryptosystems [7] allow for the same level of privacy as any other cryptosystem, while also allowing for operations to be performed on the data without the need to see the actual data. We observe that Paillier scheme is always better than other scheme Although, there is no surprise that RSA is the overall fastest, but Paillier scheme fastest probabilistic homomorphic scheme is faster than RSA in decryption because of finding r. Thus, since Paillier is faster with the same advantages, it is a much better choice. And in the further we are contribute homomorphic Paillier in computation new method other for the high security of cryptography.

REFERENCES

(Arranged in the order of citation in the same fashion as the case of Footnotes.)

[1] Paillier, P, (1999). “Public-key cryptosystems based on composite degree residuosity classes”. Advances in Crypto/ogy (EUROCRYPT'99), Volume 1592 of Lecture Notes in Computer Science, pp. 223-238, Springer, New York, USA.

[2] Rivest, R., Adleman, L., and Dertouzos, M. (1978). “On data banks and privacy_homomorphisms”. Foundations of Secure Computation, pp. 169-177.

[3] Akinwande, M. (2009). “Advances in Homomorphic Cryptosystems”. J. UCS 15.3: pp. 506-522.

[4] O’Keeffe, M. (2008). “The Paillier Cryptosystem”. A Look Into The Cryptosystem and Its Potential Application, college of New Jersey.

[5] Tobias, V. (2006). “Paillier cryptosystem: Amathematical introduction”.

[6] Parmar, P.V. and et al. (2014). “Survey of Various Homomorphic Encryption algorithms and Schemes”. International Journal of Computer Applications 91.8.

[7] Fontaine, C. and Fabien, G. (2007). “A survey of homomorphic encryption for nonspecialists”. EURASIP Journal on Information Security 2007:15.