SYMMETRIC KEY ENCRYPTION SCHEMES WITH TWO DIFFERENT MOD VALUES

EMİN AYGÜN, ERKAM LÜY AND ZEKERIYA Y. KARATAS

Abstract. In this article, we construct two new symmetric encryption schemes which use two different mod values. Our main purpose is to construct a modified new homomorphic encryption scheme which is fast and secure. Hence, we mislead the attackers by using two different mod values. We do encryption according to mod N and decryption according to mod N_1 where N is a multiple of N_1. We examine homomorphic properties of these encryption schemes. We further discuss the security of each scheme.

1. Introduction

It is very well known that making any arbitrary operation on encrypted data is very important for privacy. This fact implies the concept of homomorphism in cryptosystems. The idea of privacy homomorphism was first introduced in 1978 by Rivest, Adleman and Dertouzos in [1]. In 1982, S. Goldwasser and S. Micali constructed Goldwasser-Micali cryptosystem in [3], and later a generalization of this system, Pailler cryptosystem, was given in [4] in 1999. This cryptosystem is homomorphic with respect to addition. The famous RSA and El-Gamal cryptosystems are homomorphic with respect to multiplication, [2].

None of the cryptosystems mentioned above satisfy the feature of being homomorphic with respect to two operations. They are homomorphic with respect to a single operation, only addition or only multiplication. Up to 2009, it was not known whether a cryptosystem which is homomorphic to both addition and multiplication exists. We call such schemes Fully Homomorphic Encryption(FHE) Schemes.

In 2009, with a breakthrough result, Gentry introduced first FHE Scheme in [5]. After Gentry's paper, many cryptographers studied this system to improve it and make it more practical and more secure.

Additionally, we know that encryption schemes which are homomorphic with respect to addition have some applications in real life. Nowadays, homomorphic cryptosystems are very useful especially in electronic voting and cloud computing. So, we introduce a cyptosystem to contribute to electronic voting and cloud computing with a new idea with more security.

2010 *Mathematics Subject Classification.* 11T71.

Key words and phrases. Homomorphic Encryption, Large Integer Factorization, Chinese Remainder Theorem.

In this paper, we suggest two new symmetric key encryption schemes, one is multiplicative homomorphic, and the other is additional homorphic. We use different two mod values to deceive any attacks. We do encryption with respect to mod N and decryption according to mod N_1 with $N_1|N$. We purpose that even if attacker receive the secret key k, still more work should be done in order to break the system.

We first start with Scheme 1.

2. Scheme 1

The algorithm is as follows.

Keygen:

(1) Choose $2r$ prime numbers p_i and q_i, where $1 \leq i \leq r$, such that for each i, p_i and q_i are distinct primes.
(2) Compute $f_i = p_i q_i$, for $1 \leq i \leq r$.
(3) Compute $N_1 = \text{lcm}(f_1, \ldots, f_r)$.
(4) Compute $d = \Phi(N_1)$ where Φ denotes Euler Phi function.
(5) Pick a secret key k such that $\gcd(k, d) = 1$.
(6) Compute $N = k^2 \left(\prod_{i=1}^{r} f_i \right)$.
(7) Public key is $\{N\}$, and secret key is $\{k, f_i\}$.

Encryption:

(1) Take the output.
(2) Compute $N_1 = \text{lcm}(f_1, \ldots, f_r)$.
(3) Determine your plaintext $M \in Z_{N_1}$.
(4) Compute $C \equiv M^k \pmod{N}$.
(5) Send C as the ciphertext of M.

Decryption:

(1) Compute inverse l of k according to mod d.
(2) Compute $C^l \equiv M^{kl} \equiv M \pmod{N_1}$.

Theorem 1. *The algorithm given above works.*

Proof. Since $\gcd(k, d) = 1$, we can find l such that $kl \equiv 1 \pmod{\Phi(N_1)}$. Hence, there exist an integer b such that $kl = 1 + b\Phi(N_1)$. By construction, $N_1 = w_1 w_2 \ldots w_r$ where w_i's are distinct prime numbers. So, $\Phi(N_1) = \Phi(w_1) \times \Phi(w_2) \times ... \times \Phi(w_r)$ since Euler Phi function is multiplicative.

Let w_i be one of primes in decomposition of N_1. Assume first that $\gcd(M, w_i) = 1$. Then, by Euler's Theorem $M^{\Phi(w_i)} \equiv 1 \pmod{w_i}$. Hence, by raising both sides of this congruence to the power $b\Phi(w_1)\Phi(w_2)\ldots\Phi(w_{i-1})\Phi(w_{i+1})\ldots\Phi(w_r)$ and then multiplying both sides by M yields $M^{1+b\Phi(w_1)\Phi(w_2)\ldots\Phi(w_r)} \equiv M^{kl} \equiv M \pmod{w_i}$.

Next, suppose that $\gcd(M, w_i) = w_i$. Hence, we have $M \equiv 0 \pmod{w_i}$, which implies $M^{kl} \equiv M \equiv 0 \pmod{w_i}$. Thus, we again get $M^{kl} \equiv M \pmod{w_i}$.

Hence, for every i, we have $M^{kl} \equiv M \pmod{w_i}$. So, we have the following system of equations.

$$\begin{aligned} M^{kl} &\equiv M \pmod{w_1} \\ M^{kl} &\equiv M \pmod{w_2} \\ &\vdots \\ M^{kl} &\equiv M \pmod{w_r} \end{aligned}$$

By Chinese Remainder Theorem, this system has a unique solution $M^{kl} \equiv M \pmod{N_1}$. Now, note that $C \equiv M^k \pmod{N}$, $N_1|N$ and $M \in Z_{N_1}$, hence $C \equiv M^k \pmod{N_1}$ as well. Thus, we get $C^l \equiv M^{kl} \equiv M \pmod{N_1}$ which implies that the algorithm works.

□

Homomorphic Property of Scheme 1

Theorem 2. *The Scheme 1 is homomorphic with respect to multiplication.*

Proof. Assume that $C_1 \equiv M_1^k \pmod{N}$ and $C_2 \equiv M_2^k \pmod{N}$, where $M_i \in Z_{N_1}$ for $i = 1, 2$. We clearly have $C_1C_2 = M_1^k M_2^k = (M_1M_2)^k \pmod{N}$. So, if we decrypt the ciphertext C_1C_2 with $k^{-1} \equiv l \pmod{\Phi(N_1)}$ as in our algorithm, we obtain $M_1M_2 \pmod{N_1}$ which shows that Scheme 1 is homomorphic with respect to multiplication and completes the proof.

□

Our main purpose in this paper is to construct a secure encryption scheme against plaintext - key recovery attack via using different mod values. For this reason, we present two different schemes with different properties. The following is the security of Scheme 1.

Security of Scheme 1

Assume that the attacker knows one correct ciphertext and one correct corresponding plaintext, C_1 and M_1, respectively. Assume also that the attacker knows the mod value N. So, the attacker can construct the congruence

$$C_1 \equiv M_1^k \pmod{N} \tag{1}$$

Here, the attacker wants to obtain the secret key k. Hence, the attacker must solve the congruence (1). But this problem is a well-known problem which called Discrete Logarithm Problem (DLP). Because of the difficulty of this problem, the security is very high. However, we actually want to increase the security more with a new idea. In this method, even if the attacker solves the DLP and gets the secret key k, the attacker could not obtain the inverse of it which is used in decryption algorithm.

To increase the security, we suggest to use two different mod values for misleading the attacker. While decrypting a ciphertext, we use l which is inverse of k with respect to mod $\Phi(N_1)$. This idea was also used in famous cryptosystem RSA. So, we examine what happens if the attacker gets the

secret key k and tries to compute its inverse with respect to mod $\Phi(N)$. Note here that the attacker does not know N_1, hence the attacker can not compute $\Phi(N_1)$.

Assume that attacker gets k and wants to compute its inverse. Hence the attacker must compute $\Phi(N)$. But for computing $\Phi(N)$, the attacker must factor N. This problem is also one of well-known difficult problems - Large Integer Factorization Problem. Assume next that the attacker factors N and computes $\Phi(N)$. So, now the aim for the attacker will be obtaining the inverse of k in $\pmod{\Phi(N)}$. But it is a well-known fact that k has an inverse in $\pmod{\Phi(N)}$ if and only if $\gcd(k, \Phi(N)) = 1$.

But we know that $N = k^2 \prod_{i=1}^{r} f_i$ where $f_i = p_i q_i$ with p_i and q_i are distinct primes for each i. So, $N = k^2 p_1 q_1 p_2 q_2 \dots p_r q_r$ and hence $\Phi(N) = \Phi(k^2 p_1 q_1 p_2 q_2 \dots p_r q_r)$. Assume that $k = h_1^{t_1} h_2^{t_2} \dots h_y^{t_y}$ where h_j's are distinct primes for $1 \leq j \leq y$.

Firstly, if $h_j \in \{p_1, q_1, p_2, q_2, \dots p_r, q_r\}$ for every j, then obviously $\gcd(k, \Phi(N)) > 1$.

Assume now that $h_{j_0} \notin \{p_1, q_1, p_2, q_2, \dots p_r, q_r\}$ for some j_0. Hence, $\Phi\left(h_{j_0}^{2t_{j_0}}\right) = \left(h_{j_0}^{2t_{j_0}-1}\right)(h_{j_0}-1)$ divides $\Phi(N)$, hence we get $\gcd(k, \Phi(N)) > 1$ in this case as well.

So, the attacker will not be able to obtain the inverse of k by using N, which definitely misleads the attacker and increases the security.

Before starting Scheme 2, we want to give a simple example of Scheme 1.

Example 1.

Keygen:

(1) Let $r = 3$ and $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $q_1 = 3$, $q_2 = 5$, $q_3 = 7$.
(2) Compute $f_1 = 6$, $f_2 = 15$ and $f_3 = 35$.
(3) Compute $\text{lcm}(6, 15, 35) = 210$.
(4) Compute $\Phi(210) = 48$.
(5) Let secret key be $k = 5$ since $gcd(5, 48) = 1$.
(6) Compute $N = 5^2 \prod_{i=1}^{3} f_i = 5^2 \times 6 \times 15 \times 35 = 78750$.
(7) Output: Public Key $\{N = 78750\}$ and Secret Key $\{k = 5, f_1 = 6, f_2 = 15, f_3 = 35\}$.

Encryption:

(1) Take the output.
(2) Compute $\text{lcm}(6, 15, 35) = 210$.
(3) Let our message be $M = 20 \in Z_{210}$.
(4) Compute ciphertext as $C \equiv 20^5 \equiv 50000 \pmod{78750}$.
(5) Send $C = 50000$ as ciphertext of $M = 20$.

Decryption:

(1) Compute inverse of k, $l = 29$ with respect to $\pmod{48}$.

(2) Compute $M = 50000^{29} \equiv 20 \pmod{210}$.

Note that $\Phi(78750) = 18000$ and $\gcd(k = 5, \Phi(N) = 18000) = 5$ so k does not have inverse with respect to $\pmod{N}$. So the attacker can not obtain k^{-1} with respect to $\pmod{N}$.

3. Scheme 2

As we see in previous section, Scheme 1 is homomorphic with respect to multiplication. Now, we want to construct an additional homomorphic encryption scheme similar to Scheme 1 as additional homomorphic encryption schemes can be useful in electronic voting.

The algorithm for Scheme 2 is as follows.

Keygen:

(1) Choose $2r$ prime numbers p_i and q_i, where $1 \leq i \leq r$, such that for each i, p_i and q_i are distinct primes.
(2) Compute $f_i = p_i q_i$, for $1 \leq i \leq r$.
(3) Compute $N_1 = \text{lcm}(f_1, \ldots, f_r)$.
(4) Pick a secret key k such that $\gcd(k, N_1) = 1$.
(5) Compute $N = k\left(\prod_{i=1}^{r} f_i\right)$.
(6) Public key is $\{N\}$, and secret key is $\{k, f_i\}$.

Encryption:

(1) Take the output.
(2) Compute $N_1 = \text{lcm}(f_1, \ldots, f_r)$.
(3) Determine your plaintext $M \in Z_{N_1}$.
(4) Compute $C \equiv kM \pmod{N}$.
(5) Send C as the ciphertext of M.

Decryption:

(1) Compute inverse l of k with respect to mod N_1.
(2) Compute $lC \equiv lkM \equiv M \pmod{N_1}$.

Theorem 3. *The algorithm given above works.*

Proof. It can be easily shown that this algorithm works by using the similar arguments in the proof of Theorem 1.

□

Homomorphic Property of Scheme 2

Theorem 4. *The Scheme 2 is homomorphic with respect to addition.*

Proof. Assume that $C_1 \equiv kM_1 \pmod{N}$ and $C_2 \equiv kM_2 \pmod{N}$, then we have $C_1 + C_2 \equiv k(M_1 + M_2) \pmod{N}$. So, if we decrypt the ciphertext $C_1 + C_2$ with $k^{-1} \equiv l \pmod{N_1}$ as in our algorithm, we obtain $M_1 + M_2$, which completes the proof.

□

Security of Scheme 2

Different from previous scheme, the attacker can receive k easily if attacker knows C, M, N_1. However, as in before, the attacker can not obtain the inverse of k with respect to $\pmod N$ as $\gcd(k, \Phi(N)) > 1$. Hence we can say that Scheme 2 is strongly secure as well.

4. Conclusion

In this paper, we introduce two new symmetric key encryption schemes which use two different mod values. Each scheme is homomorphic with respect to a single operation. We also show that these schemes are strongly secure. They are very simple and useful for applications like electronic voting.

References

1. R.Rivest, L.Adleman ve M.L.Dertouzos, *On data banks and privacy homomorphisms, Foundations of Secure Computation*, 169-170, 1978.
2. Alice Silverberg, *Fully Homomorphic Encryption for Mathematicians*, sponsored by DARPA under agreement numbers FA8750-11-1-0248 and FA8750- 13-2-0054. 2013.
3. S. Goldwasser ve S. Micali, *Probabilistic encryption and how to play mental poker keeping secret all partial information*, Proceedings of the 14th ACM Symposium on Theory of Computing, 365-377, 1982.
4. P.Pailler, *Public-Key Cryptosystems Based on Composite degree Residuosity Classes*, Advances in Cryptology, EUROCRYPT, 223-238, 1999.
5. Craig Gentry, *A Fully Homomorphic Encryption Scheme*, Ph.D. Thesis, 2009, Stanford University.

Emin Aygun
Department of Mathematics, Erciyes University, Kayseri, 38200, TURKEY
Email: eaygun@erciyes.edu.tr

Erkam Luy
Department of Mathematics, Erciyes University, Kayseri, 38200, TURKEY
Email: erkamluy@erciyes.edu.tr

Zekeriya Y. Karatas
Department of Mathematics, Tuskegee University, Tuskegee, AL 36088, U.S.A.
Email: zkaratas@mytu.tuskegee.edu