The Homomorphic Encryption Scheme of Security Obfuscation

Gong Gao-xiang[1,2], Yuan Zheng[2], and Feng Xiao[1]

[1] Communication Engineer Institute, Xidian University, Shanxi 710071, China
[2] Beijing Electronic Science & Technology Institute, Beijing 100070, China

Abstract. In the cloud storage service, according to the data on the cloud computing safety protection problem, the paper presents secure obfuscating homomorphism encryption scheme. Constructing a point function obfuscation that based on perfectly one way probability hash function in scheme, construction depends on hash function and the computational difficulty problems, then use the computational difficulty problems, to realize the encrypted homomorphism function, also guarantee the function of the point function obfuscator at the same time, the scheme raises the security of the encrypted data. This paper provides the security proof of the scheme, shows that the scheme is feasible.

Keywords: obfuscation, computational difficulty, point function obfuscation, homomorphic encryption, perfect one-way hash function.

1 Introduction

The idea of Fully Homomorphic Encryption (FHE) was proposed by Rivest, Adleman and Dertouzosin in 1978. The existence construction has long been one of the important open questions in cryptography. Unfortunately, there has been no progress in cryptographic circle after the issue was put forward for more than 30 years. The construction of fully homomorphic encryption problem research has achieved significant breakthrough in 2009. Gentry constructed the first fully homomorphic encryption scheme [2] based on ideal lattice in 2009. The security of the scheme is based on the computational difficulty of two problems: Bounded distance coding issues over ideal lattice and spare subset sum problem. More detail about construction of the scheme is given in Gentry's dissertation [3]. A very simple homomorphic encryption scheme based on integer is proposed by Dijk, Gentry and Halevi, whose security is based on intractability of the approximate GCD and spare subset sum. A recent article [5] about fully homomorphic mainly improve the first fully homomorphic encryption scheme.

Obfuscation is a new research direction in the last few decades. Say in brief, obfuscation is a kind of algorithm, inputting a program of Boolean system circuit (now the research of obfuscation mainly focus on circuit and Turing machine, while the obfuscation of this paper mainly focus on circuit.) through the obfuscator, then output a

T. Tan et al. (Eds.): IGTA 2013, CCIS 363, pp. 127–135, 2013.

new program of Boolean system circuit which has the same function with the old one. Although the output circuit program on the function is same with the original program, but we find it difficult to read, i,e, we are not easy to recognize and understand. Obfuscator behaves like a "black box" in this concept, in some degree, obfuscator will not leak any information on obfuscation program besides the input and output information.

The idea of this paper is to realize the obfuscation of homomorphic encryption through a point function obfuscator. According to one of the way of constructing point function obfuscation——perfect one-way hash function, the essence of perfect one-way hash function in [6] and [7] is point function obfuscator. The way of construction is based on hash function and computational difficulty problems. And now homomorphic encryption schemes are constructed based on computational difficulty problem. As both depend on computational difficulty problem (that is they contact a common bridge). Then we'll learn the security obfuscation of homomorphic encryption according to the point function obfuscator constructed below.

In the rest of this paper, Section 2 presents the preliminaries of point function obfuscator; the point function is constructed in Section 3; a security obfuscation scheme of homomorphic encryption is realized by this obfuscator is introduced in Section 4; finally, we prove the scheme's obfuscation character and security analysis in Section 5.

2 Preliminaries

2.1 The Introduction of Virtual Black-Box Obfuscation

For circuit C , a probabilistic polynomial time O is a virtual black-box obfuscator. It meets three conditions as follows:

----(**Functional**) For any $n \in N, C \in C_n, O(C)$ is a circuit, and has the same computing function with C .

----(**Polynomial slowdown**) There exists a polynomial q , that for any $n \in N, C \in C_n, |O(C)| \le q(|C|)$.

---- (**virtual black-box**) For any PPT opponent A and polynomial p ,there is a PPT simulator S. Then for all large enough $n \in N$ and $C \in C_n$:

$$\left| \underset{A,O}{pr}[A(O(C))=1] - \underset{S}{pr}\left[S^C\left(1^{|C|}\right)=1\right] \right| \le 1/p(n)$$

The obfuscator is effective if the obfuscator runs in polynomial time.

The black-box obfuscation in polynomial slowdown showing the complexity of the circuit and the virtual black-box (VBB) performance together provide a strong protection for the circuit security after obfuscated. The obfuscated circuit acts as a "black box", in a sense, and it don't divulge any information about the circuit except its input and output behavior. More precisely, any valid adversary who can access to the obfuscation circuit can only get through an effective simulator which enters into the scheme through an oracle channel.

2.2 The Introduction of Semantic Perfect One-Way

For any non-unified PPT A and polynomial p, an ensemble $H=\{H^n\}_{n\in N}$, it is called semantic perfect one-way if it meets three conditions as follows:

$$\text{Completeness: } \forall k\in K_n, x\in\{0,1\}^n, r\in R_n, V(x,H_k(x,r))=1$$

Collision resistance: For any non-unified PPT A:

$$\Pr[k\leftarrow K_n,(x_1,x_2,y)\leftarrow A(k): x_1\neq x_2\wedge V(x_1,y)=V(x_2,y)=1]<u(n)$$

Secrecy: There exists a non-unified PPT S, for sufficiently large n, any k and x:

$$\left|\Pr[r\leftarrow R_n, b\leftarrow A(k,H_k(x,r)):b=1]-\Pr\left[r\leftarrow R_n, b\leftarrow S^{F_x}(k):b=1\right]\right|\leq 1/p(n)$$

Where F_x is a point function in x.

Note that semantic perfect one-way adopt to a simple way to virtual black box, the performance requirements of the obfuscation point function in the definition [8]. Therefore, a function which meets this definition is a point function obfuscation (possess approximate functional computing).But it is not real in the opposite direction. When semantic perfect one-way implies the virtual black box performance, the completeness and collision resistance in H will imply the approximate function of computing. On the other hand, the obfuscation of a point function will not be a perfect one-way function, in an adverse way, because the approximate functional doesn't limit on collision selection.

Definition 1 (Homomorphic encryption) [9] the encryption scheme $E=(KeyGen_\varepsilon, Encrypt_\varepsilon, Decrypt_\varepsilon, Evaluate_\varepsilon)$ is called homomorphism encryption scheme, if for every function f of a kind of specific function F_ε, The output ciphertext of $Evaluate_\varepsilon$ meet correctness requirements. Let such set of function F_ε equal $c_\varepsilon(\lambda)$.

Correctness: For any given security parameter λ, $KeyGen_\varepsilon(\lambda)$ output any key pairs $KeyGen_\varepsilon(\lambda)$, any $f\in F_\varepsilon$, any given plaintext $m_1,m_2,\ldots,m_t$ and the corresponding ciphertext $\vec{c}=(c_1,c_2,\ldots,c_t)$, and $c_i\leftarrow Encrypt_\varepsilon(pk,m_i)$, as $c\leftarrow Evaluate_\varepsilon(pk,f,\vec{c})$, then $Decrypt_\varepsilon(sk,c)=f(m_1,\ldots,m_t)$ established.

3 Construct k, rt^k Obfuscators

The construction process is as follows. Supposing p is a large safe prime, that is to say, $p = aq + 1$, where a is a little integer （For simplicity, we assume that $a = 2$). In Z_p^*, supposing Q is a subset of q time.(In other words, Q is a group of square model p).After input m and secret random input $k \in_R Q$, oracle hash function H computes $r = h(m)$ first, here h is a collision resistance hash function. Then output $H(m,k) = k, rt^k$ (The calculation result is model p). Authentication algorithm V is simple; given an input m and a hash value $\langle a,b \rangle$, calculate $x = h(m)$ and accept it if $rt^a = b$.

The further description of this obfuscator is as follows:

(Construct k, rt^k point obfuscators) Suppose $g = \{G\}$ is a group overall, each G_n is the group whose prime order is p . We define an obfuscator O, for point in domain Z_p^* , there is under type: $C_x \xrightarrow{O} c(k, rt^k)$, $k \xleftarrow{U} G^*$ is a random generator of G_n and $c(k, rt^k)$ is a circuit which input r ,check whether $xt^a = rt^a$.

Under the strong variant of the decided Diffie-Hellman assumption, this construct is safe. This construct and the point obfuscator in [10] is semantic security, which based on logarithm of intractable problems over finite fields.

The introduction of point function obfuscation was proposed in [9] for the first time. More detailed introduction of point function obfuscation and two types of point function obfuscator constructed are given in [12] and [13]. The obfuscators above are constructed reference to the perfect one-way hash function.

We analyze this construction based on the strong Diffie-Hellman assumption variant which is used to reveal, and this construction meet the oracle security of random input and prior message.

Assumption 1: The Diffie-Hellman Indistinguishability Assumptions: Let k be a security parameter. Let $p = 2q + 1$ be a randomly chosen k -bit safe prime and let $g \in_R Q$ (where Q is the group of squares modulus p).

DHI Assumption I: Let $a,b,c \in_R Z_q^*$, then $\langle g^a, g^b, g^{ab} \rangle \approx \langle g^a, g^b, g^c \rangle$。

DHI Assumption II: For any well-spread distribution ensemble $\{X_q\}$ where the domain of X_q is Z_q^*, for a drawn from X_q and for $b,c \in_R Z_q^*$ we have $\langle g^a, g^b, g^{ab} \rangle \approx \langle g^a, g^b, g^c \rangle$。

DHI Assumption III: For any uninvertible function f and for $a,b,c \in_R Z_q^*$ we have $\left\langle f(a), g^b, g^{ab} \right\rangle \approx \left\langle f(a), g^b, g^c \right\rangle$。

1. It can be seen that Assumption III implies Assumption II, and Assumption II implies Assumption I. We were unable to show implications in the other direction.

2. While these assumptions are considerably stronger than the standard Diffie-Hellman assumption (For p, g, g^a, g^b, it is only assumed that g^{ab} cannot be computed), they seem consistent with the current knowledge on the Diffie-Hellman problem. In particular, the assumption in the past is explicitly and implicitly. It is not hard to see that it is equivalent to the semantic security of the ELGamal encryption scheme, because both of them are based on logarithm of intractable problems over finite fields.

Although Assumption II and III look quite strong, we were unable to contradict them. We propose the viability of these assumptions as an open question. To gain assurance in the plausibility of these assumptions, we remark that it is a common practice to use Diffie-Hellman key exchange modulo a large prime.

For the analysis of the construction, we first consider a somewhat simplified version, where the collision resistant hash function h is omitted and the input is assumed to be taken from Z_p^*.

Theorem1

1. If DHI Assumption I holds then the function $H(m,k) = k, rt^k$, together with its verification algorithm, are an oracle hashing scheme for random inputs.

2. If DHI Assumption II holds then the function $H(m,k) = k, rt^k$, together with its verification algorithm, are an oracle hashing scheme.

3. If DHI Assumption III holds then the function $H(m,k) = k, rt^k$, together with its verification algorithm, are a strong oracle hashing scheme.

4 Achieve a Security Obfuscator of Homomorphic Scheme by Obfuscator

A homomorphic encryption scheme is showed at first in this part (scheme 1), next we obfuscate this scheme (scheme 2). The two schemes is equal in the function, comparatively speaking, the readability of scheme 2 is worse (that is to say, “Hard to identify”).

Scheme 1: A simple homomorphic encryption scheme (multiplication homomorphism)

(1) Select a primitive element g , $g \in Z_p^*$;

(2) Select an integer d randomly, $1 \le d \le p-2$,compute $t = g^d (\bmod p)$;

(3) Encryption transformation: For any plaintext $m \in Z_p^*$,secretly choose an integer $k, k \in_R Q$, then compute $E(m) = \left(g^k, mt^k\right)$;

(4) Evaluate transformation: After obfuscation encrypt for any plaintext message m_1, m_2,

$$E(m_1) \cdot E(m_2)\left(g^k, m_1 t^k\right)\left(g^k, m_2 t^k\right) = \left(g^{k_1+k_2}, \left((m_1 \cdot m_2)t^{k_1+k_2}\right)\right) = E(m_1 \cdot m_2)$$

(5) Decryption transformation: Let

$$c_1 = g^{k_1+k_2} \bmod p, c_2 = (m_1 \cdot m_2)t^{k_1+k_2}, t = g^d (\bmod p),$$

$$c_2\left(c_1^d\right)^{-1} \equiv (m_1 \cdot m_2)t^{k_1+k_2}\left(\left(g^{k_1+k_2}\right)^d\right)^{-1} \equiv (m_1 \cdot m_2)g^{d(k_1+k_2)}\left(g^{-d(k_1+k_2)}\right)(\bmod p) \equiv (m_1 \cdot m_2)$$

Notice: p , g are the public elements, t is the encryption key and d is the confidential decryption key .

Scheme 2: We achieve security obfuscation of scheme 1 by the obfuscator which constructed in part 3 and retain the homomorphic characteristic of the scheme in the same time. The scheme as follows:

(1) Select a primitive element g , $g \in Z_p^*$;

(2) Select an integer d randomly, $1 \le d \le p-2$,compute $t = g^d (\bmod p)$;

(3) Encryption transformation: For any plaintext $m \in Z_p^*$,secret choose an integer $k, k \in_R Q$,compute $r = h(m), E(r) = \left(g^k, rt^k\right)$;

(4) Evaluate transformation: After obfuscation encrypt for any plaintext message m_1, m_2,

$$E(r_1) \cdot E(r_2) = \left(g^k, h(m_1)t^k\right)\left(g^k, h(m_2)t^k\right) = \left(g^{k_1+k_2}, \left(\left(h(m_1) \cdot h(m_2)\right)t^{k_1+k_2}\right)\right) = E(r_1 \cdot r_2)$$

(5) Decryption transformation: Let

$$c_1 = g^{k_1+k_2} \bmod p \quad , \quad c_2 = \left(h(m_1) \cdot h(m_2)\right)t^{k_1+k_2} \quad , \quad t = g^d (\bmod p) \quad ,$$

$$c_2\left(c_1^d\right)^{-1} \equiv \left(h(m_1) \cdot h(m_2)\right)t^{k_1+k_2}\left(\left(g^{k_1+k_2}\right)^d\right)^{-1}$$

$$\equiv \left(h(m_1) \cdot h(m_2)\right)g^{d(k_1+k_2)}\left(g^{-d(k_1+k_2)}\right)(\bmod p) \equiv \left(h(m_1) \cdot h(m_2)\right)$$

Notice: p , g are the public elements, t is the encryption key and d is the confidential decryption key .

According to the two schemes above, different ciphertext multiply after they encrypt different messages, which united equal to encryption of a set of different message. Then we find the set of message can be decryption by the decryption key. So the correctness of homomorphic encryption scheme is verified.

5 Proof and Analyze the Security of Secure Obfuscation of Homomorphic Encryption Scheme

In this part, we verify the correctness of homomorphic, then we prove the homomorphic encryption scheme of obfuscation security above is obfuscation secure and the mainly method is reducibility certification.

5.1 The Correctness of Homomorphic（Mainly for Homomorphic Encryption Scheme）

(1)There exists a pair of key (t,d), t is the encryption key and d is the confidential decryption key.

(2) After obfuscation encrypt for any plaintext message m_1, m_2,

$$c_{11} = g^k \bmod p, c_{21} = (m_1)t^{k_1} \text{ and } c_{12} = g^k \bmod p, c_{22} = (m_2)t^{k_2}$$

(3)According to evaluate transformation $c = \left(g^{k_1+k_2}, ((m_1 \cdot m_2)t^{k_1+k_2})\right)$;

(4) $c_2\left(c_1^d\right)^{-1} \equiv (m_1 \cdot m_2) g^{d(k_1+k_2)}\left(g^{-d(k_1+k_2)}\right)(\bmod p) \equiv (m_1 \cdot m_2)$

The correctness verification of scheme 2 is the same as above, it can be set up according to the correctness verification of scheme 1, so the correctness of scheme 2 can be set up, i,e, the homomorphism encryption scheme after obfuscation will not change homomorphism.

5.2 Obfuscation Proof of the Scheme

Through the reducibility certification, it actually can be summed up in obfuscation security certification of the point function obfuscator in part 3.Now we divide it into two steps: the first , certificating this obfuscator meet three characteristics of the definition of obfuscation; the second, certificating this construction is perfect one-way hash function of the probability.

Proof: The following will prove that this function satisfies the definition of obfuscator. (i) As we compute $r = h(m)$ at first, for a circuit C (the characteristic is mt^a), the result of obfuscation is $O(C)$, the two on the function of the computing can not be distinguished (keeping function).(ii)It is cleared that the polynomial slowdown performance is established.(iii)]For any adversary A of probability polynomial time, there are a probability polynomial time S and a negligible function α, so for all the circuits C : $\left|\Pr[A(O(f))=1] - \Pr\left[S^C\left(1^{|C|}\right)\right]\right| \leq \alpha(|C|)$ (Characteristics of weak virtual black box).

This obfuscator is efficient if this obfuscator O runs in polynomial time. After the hash function processes the message, it reaches the difficulty identify characteristics of obfuscation, thus this construction is established to meet the obfuscator.

We will prove that this obfuscator is semantic perfect one-way. An ensemble $H = \{H^n\}_{n\in N}$, there is a decided polynomial time algorithm V. There exist a private key β in the key space, $x \in \{0,1\}^n$, k is belong to a randomelement.

Because $h(x)\beta^k = H_\beta(x,k)$,so $V(x, H_\beta(x,k)) = 1$ 。 (Completeness established).For any nonuniform probability polynomial time A and n which is large enough, there exists a negligible function $u(n)$ and message $x_1 \neq x_2$:

$\Pr[\beta \in K_n, (x_1, x_2, y) \leftarrow A(\beta) : V(x_1, y) = V(x_2, y) = 1] < u(n)$ (Collision resistance established)

For any nonuniform probability polynomial time A and polynomial p, there is a nonuniform probability polynomial time simulator S, for large enough n, any β and message x, input a $k \in R_n$ randomly:

$$\left|\Pr[b \leftarrow A(\beta, H_\beta(x,k)) : b = 1] - \Pr[b \leftarrow S^{F_x}(\beta) : b = 1]\right| \leq 1/p(n)$$

Here F_x is a point function of independent variable x, therefore, the construction is semantic perfect one-wayness.

In more detail, let H be a semantic POW function. To obfuscate F_x, sample a seed, k, and random string, r, for H and output the obfuscation, $O(F_x) = k, H_k(x,r)$. The new function, $O(F_x)$, simply computes the predicate $V(., H_k(x,r))$. It can be shown that O is an obfuscator for the class of point functions. Completeness and collision resistance on H imply computational approximate functionality while semantic perfect one-wayness implies the virtual-black box property. On the other hand, an obfuscation of point functions may not be a POW function because approximate functionality does not rule out collisions chosen in an adversarial way.

In the definition of obfuscation, for the obfuscation of point function, we take a simple way and find that the semantic perfect one-way accord with the performance of virtual black box. Hence it is a obfuscation of a point function if this function meets the semantic perfect one-wayness (The approximate function of computing).But it is not established in the opposite side.

6 Conclusion

In order to improve security, this paper in view of the existing homomorphic encryption scheme and propose homomorphic scheme of obfuscation security. This scheme is safer than other scheme and is difficult to identify. This article is the first time adds obfuscation characteristic to homomorphic public-key encryption and it will greatly protect the encryption data of user in cloud. Also this scheme can use in electronic ballot, which can improve the fairness of election. For the scheme, there are other purposes to be discovered.

References

[1] Rivest, R.L., Adleman, L., Dertouzos, M.L.: On Data Banks and Privacy Homeomorphisms. In: Foundations of Secure Computation, pp. 169–177 (1978)

[2] Gentry, C.: Fully Homomrphic Encryption Using Ideal Lattices. In: ACM STOC 2009, pp. 169–178 (2009)

[3] Gentry, C.: A Fully Homomorphic Encryption Scheme. Ph.D.Thesis, Stanford University (2009)

[4] van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)

[5] Gentry, C., Halevi, S.: Implementing Gentry's Fully-Homomorphic Encryption Scheme. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 129–148. Springer, Heidelberg (2011)

[6] Canetti, R.: Towards Realizing Random Oracles: Hash Functions that Hide All Partial Information. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 455–469. Springer, Heidelberg (1997)

[7] Canetti, R., Micciancio, D., Reingold, O.: Perfectly one-way probabilistic hash functions. In: 30th ACM Symposium on Theory of Computing, pp. 131–140. ACM Press (1998)

[8] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (Im)possibility of Obfuscating Programs. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 1–18. Springer, Heidelberg (2001)

[9] Zhou, Y.: Homomrphic cryptography research progress. China cryptography development report 2010, p. 169. Publishing House of Electronics Industry, Beijing (2010)

[10] Bitansky, N., Canetti, R.: On Strong Simulation and Composable Point Obfuscation. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 520–537. Springer, Heidelberg (2010)

[11] Lynn, B.Y.S., Prabhakaran, M., Sahai, A.: Positive Results and Techniques for Obfuscation. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 20–39. Springer, Heidelberg (2004)

[12] Wee, H.: On Obfuscation Point Funtions. In: Proceedings of the 37th ACM Symposium on Theory of Computing, pp. 523–532 (2005)

[13] Canetti, R., Dakdouk, R.R.: Obfuscating Point Functions with Multibit Output. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 489–508. Springer, Heidelberg (2008)

[14] Naor, M., Reingold, O.: The Brain can Compute Pseudo-Random Functions, Or Efficient Cryptographic Primitives Based on the Decisional Diffie-Hellman Assumption (manuscript)