Tsinghua Science and Technology

Volume 19 | Issue 5 Article 7

2014

Targeted Fully Homomorphic Encryption Based on a Double Decryption Algorithm for Polynomials

Yatao Yang
Beijing Electronic Science and Technology Institute, Beijing 100070, China.

Shuang Zhang
Beijing Electronic Science and Technology Institute, Beijing 100070, China. Communication Engineering Institute, Xidian University, Xi'an 710071, China.

Junming Yang
Beijing Electronic Science and Technology Institute, Beijing 100070, China.

Jia Li
Beijing Electronic Science and Technology Institute, Beijing 100070, China.

Zichen Li
Beijing Electronic Science and Technology Institute, Beijing 100070, China. Communication Engineering Institute, Xidian University, Xi'an 710071, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/13 pp478-485
Volume 19, Number 5, October 2014

Targeted Fully Homomorphic Encryption Based on a Double Decryption Algorithm for Polynomials

Yatao Yang, Shuang Zhang*, Junming Yang, Jia Li, and Zichen Li

Abstract: Several public-key encryption schemes used to solve the problem of ciphertext data processing on the fly are discussed. A new targeted fully homomorphic encryption scheme based on the discrete logarithm problem is presented. Public-key encryption cryptosystems are classified to examine homomorphic encryption. Without employing techniques proposed by Gentry such as somewhat homomorphic and bootstrapping techniques, or relinearization technique proposed by Brakerski et al., a new method called "Double Decryption Algorithm" is employed in our cryptography to satisfy a fully or targeted fully homomorphic property. Inspired by ElGamal and BGN cryptography, we obtain the desired fully homomorphic property by selecting a new group and adding an extra component to the ciphertext. Proof of semantic security is also demonstrated.

Key words: targeted fully homomorphic encryption; discrete logarithm problem; exponential function method; power function method

1 Introduction

With the rapid development of cloud computing and big data processing, homomorphic encryption has become a key technology for protecting user privacy and processing ciphertext data. As a result, fully homomorphic encryption is in urgent demand.

In 2009, the first fully homomorphic encryption scheme, which is based on ideal lattice and is of great theoretical significance, was proposed by Gentry[1,2]. However, there are several problems with the application of this scheme. First, it simulates the algorithm with circuits, which is referred to as circuit homomorphism. Any arithmetic operation can be decomposed into basic operations such as "+" and "×", whereas it is difficult or even impossible to transform complex arithmetic operations into circuit operations. Second, a new security assumption, i.e., the Sparse Subset Sum Problem (SSSP) assumption, whose security has not been verified, has been introduced at the stage of squashing the decryption circuit. In addition, Lee[3] indicated that the SSSP is not hard. Third, Gentry's idea is to reduce the complexity of the decryption circuit as much as possible. However, the size of the public key and the complexity of the encryption circuits are increasing significantly. As a result, CPUs cannot perform such complex operations. Assuming that Moore's Law has no limit, the processing power required to perform fully homomorphic encryption needs at least 30 years of development. All of these problems must be overcome to realize a fully homomorphic encryption design.

Other related work has applied other technology. In 2010, a fully homomorphic encryption over the integer was presented by Dijk et al., i.e., the DGHV scheme[4]. The complexity of the DGHV scheme has been improved, however, efficiency and security remain a bottleneck of this scheme. In 2013, the

• Yatao Yang, Shuang Zhang, Junming Yang, Jia Li, and Zichen Li are with Beijing Electronic Science and Technology Institute, Beijing 100070, China. E-mail: yy2008@163. com; 790301590@qq.com; 393770284@qq.com; memory163@163.com; lizc2020@163.com.
• Shuang Zhang and Zichen Li are also with Communication Engineering Institute, Xidian University, Xi'an 710071, China.
* To whom correspondence should be addressed.
Manuscript received: 2014-07-14; revised: 2014-07-28; accepted: 2014-08-20

Chinese Remainder Theorem was applied by Cheon et al.[5] to improve the efficiency of the DGHV scheme. However, the security problem remains unresolved. A fully homomorphic encryption scheme without bootstrapping has also been designed[6]. However, the algorithm demonstrates low efficiency. In fact, a fully homomorphic encryption scheme for polynomials with fixed form can be widely adopted in some areas, such as cryptograph database retrieval and security multiparty computation. In this paper, traditional homomorphic public-key encryption schemes are studied and methods for the design of a homomorphic map are provided. On this basis, a new targeted fully homomorphic encryption scheme is proposed. The efficiency of the proposed scheme is its main advantage, and its security is based on the discrete logarithm problem without introducing new unverified assumptions. In addition, the proposed scheme is unique in both construction and targeted application.

2 Preliminaries

Let ε be a cryptosystem, and $c_1, c_2, \cdots, c_l$ are ciphertexts that correspond to the plaintexts $m_1, m_2, \cdots, m_l$. Thus, we have following definitions.

Definition 1 (additively homomorphic): If $\text{Decrypt}(c_1 \bigotimes c_2) = m_1 + m_2$, where $\bigotimes$ represents some arithmetic operation, then it is additively homomorphic.

Definition 2 (multiplicatively homomorphic): If $\text{Decrypt}(c_1 \bigotimes c_2) = m_1 \cdot m_2$, where $\bigotimes$ represents some arithmetic operation, then it is multiplicatively homomorphic.

Definition 3 (linearly homomorphic): If $\text{Decrypt}(c_1 \bigotimes c_2 \bigotimes \cdots \bigotimes c_l) = m_1 + m_2 + \cdots + m_l$, where $\bigotimes$ represents some arithmetic operation, then it is linearly homomorphic.

Definition 4 (exponentially homomorphic): If $\text{Decrypt}(c_1 \bigotimes c_2 \bigotimes \cdots \bigotimes c_l) = m_1 \cdot m_2 \cdot \cdots \cdot m_l$, where $\bigotimes$ represents some arithmetic operation, then it is exponentially homomorphic.

Definition 5 (targeted fully homomorphic): If one can obtain some fixed arithmetic operation, i.e., a targeted polynomial on the plaintext by performing an arithmetic operation on the ciphertext that corresponds to the plaintext, then it is targeted fully homomorphic.

Definition 6 (fully homomorphic): A scheme is fully homomorphic if it is homomorphic for any polynomial evaluation.

Definition 7 (homogeneous polynomial): Let f be a polynomial whose coefficients are integers. If

$$f(x_{s,i}) = \sum_{s=1}^{l} \left(\prod_{i=1}^{k} x_{s,i} \right),$$

where k is constant, then f is a homogeneous polynomial with degree k and item l. Furthermore, $\sigma = (k, l)$ is the attribute of the homogeneous polynomial.

Any polynomial can be computed from several homogeneous polynomials.

Definition 8 (equal-length polynomial): Let f be a polynomial whose coefficients are integers. If

$$f(x_{s,i}) = \prod_{s=1}^{k} \left(\sum_{i=1}^{l} x_{s,i}^{d_i} \right),$$

where k is constant, then f is an equal-length polynomial with length l and degree k. Furthermore, we denote $\sigma = (k, l)$ is the attribute of the equal-length polynomial.

Note that polynomials with proper form can be computed from several equal-length polynomials.

Definition 9 (discrete logarithm problem): Let G be a finite multiplicative group, and g is a generator of G. Then the discrete logarithm problem is, given any $h \in G$, to find $0 \leqslant x < G$ with $h = g^x$.

3 Idea of Homomorphism

This section analyzes all homomorphic encryptions in the literature and discusses methods for constructing homomorphic maps.

A large number of public-key encryption schemes were proposed after public-key encryption was presented by Diffie and Hellman in 1976. It is worth noting that the majority of the proposed schemes are homomorphic encryption schemes. We discuss three main classes of homomorphic encryption schemes and show how a homomorphic map can be designed properly.

3.1 RSA class of homomorphic encryption schemes

The RSA class of homomorphic encryption schemes includes RSA[7], Rabin[8], Williams[9,10], and so on. Among such schemes, the RSA cryptosystem is based on the Euler Theory, and the Rabin cryptosystem is based on quadratic residues. The flaw of the Rabin cryptosystem is that four plaintexts are generated by the decryption algorithm, which results in a significant amount of work for the decrypter. The Williams cryptosystem avoids this problem and can perform decryption properly. The security of all of the above cryptosystems is based on the big integer factoring

problem. In the following, the RSA cryptosystem is used to demonstrate the homomorphic property.

keygen($\cdot$): takes as input a security parameter λ which denotes the bit length of the module n. Select two large primes p and q, compute $n = pq$ and $\varphi(n) = (p-1)(q-1)$. Select randomly e with $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$, and then compute d, which meets the condition $e \cdot d = 1 \bmod \varphi(n)$. Then the public key is (n, e), and the secret key is d.

Encrypt($\cdot$): takes as input the public key and a plaintext $m \in Z_n$. This outputs a ciphertext $c = m^e \pmod n$.

Decrypt($\cdot$): takes as input the secret key and a ciphertext c and outputs $m = c^d \pmod n$.

Theorem 1 (multiplicatively homomorphic property) The RSA cryptosystem is multiplicatively homomorphic.

Proof Let m_1 and m_2 be two plaintexts, where c_1 and c_2 are two ciphertexts that corresponds to m_1 and m_2, respectively, i.e., $c_1 = m_1^e \pmod n$ and $c_2 = m_2^e \pmod n$.

$c_1 \cdot c_2 = m_1^e \cdot m_2^e \pmod n = (m_1 \cdot m_2)^e \pmod n$.

$\text{Decrypt}(\text{sk}, c_1 \cdot c_2) = (c_1 \cdot c_2)^d \pmod n = (m_1 \cdot m_2)^{e \cdot d} \pmod n = m_1 \cdot m_2$.

Thus, the RSA cryptosystem is multiplicatively homomorphic. ■

The homomorphic property lies in the power function's homorphism, e.g., let $y_1 = x_1^a$ and $y_2 = x_2^a$ be two power functions. Then $y_1 y_2 = (x_1 x_2)^a$, which maintains the operation. This generates the multiplicatively homomorphic property. Similarly, one can obtain $\text{Decrypt}(\text{sk}, c_1 \cdot c_2) = m_1 \cdot m_2$ in both Rabin's and Williams' cryptosystems with $c_1 = m_1^e \pmod n$ and $c_2 = m_2^e \pmod n$ (as shown in Table 1), these cryptosystems are multiplicatively homomorphic. Thus, we obtain the following theorem.

Theorem 2 One can obtain a multiplicative homomorphic encryption scheme by placing the plaintext at the location of the base.

Table 1 RSA class of homomorphic cryptosystems.

Crypto-system	Encryption function	Homomorphism
RSA	$c = m^e \pmod n$[7]	Multiplicatively homomorphic
Rabin	$c = m^e \pmod n$[8]	Multiplicatively homomorphic
Williams	$c = m^e \pmod n$[9,10]	Multiplicatively homomorphic

3.2 Paillier class of homomorphic encryption schemes

The Paillier class of homomorphic encryption schemes includes Paillier[11], Damgard-Jurik[12], Naccache-Stern[13], Okamotto-Uchiyama[14], BGN[15], etc. Paillier[11] based his cryptosystem on the n-th composite residues, and Damgard and Jurik[12] proposed a tweaked scheme. Okamotto-Uchiyama cryptosystem is constructed on the log function over the p-Sylow subgroup. The BGN and Naccache-Stern cryptosystems are designed on the multiplicative cyclic group. Here we then take the Paillier cryptosystem to demonstrate this type of homomorphism.

keygen($\cdot$): Select two large primes p and q. Compute $n = pq$ and $B = \{u \in Z_{n^2}^* | \text{order}(u) = kn\}$. Choose randomly $g \in B$. The public key is (n, g) and the secret key is (p, q).

Encrypt($\cdot$): takes as input the public key and a plaintext $m \in Z_n$. Choose randomly $r \in Z_n^*$. This outputs ciphertext $c = g^m r^n \pmod{n^2}$.

Decrypt($\cdot$): takes as input the secret key and a ciphertext c. Let $S_u = \{u < n^2 | u \equiv 1 \bmod n\}$ and $L(u) = \dfrac{u-1}{n}$. Compute $\lambda(n) = (p-1)(q-1)$. This outputs $m = \dfrac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \pmod n$.

Theorem 3 (additively homomorphic property) The Paillier cryptosystem is additively homomorphic.

Proof Let m_1 and m_2 be two plaintexts, where c_1 and c_2 are two ciphertexts that correspond to m_1 and m_2, respectively, i.e., $c_1 = g^{m_1} r_1^n \pmod{n^2}$ and $c_2 = g^{m_2} r_2^n \pmod{n^2}$.

$c_1 \cdot c_2 = (g^{m_1} r_1^n) \cdot (g^{m_2} r_2^n) \pmod{n^2} = g^{m_1 + m_2} (r_1 r_2)^n \pmod{n^2}$.

$\text{Decrypt}(\text{sk}, c_1 \cdot c_2) = \dfrac{L((c_1 \cdot c_2)^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \pmod n = m_1 + m_2$.

Therefore, the Paillier cryptosystem is additively homomorphic. ■

The Paillier class of cryptosystems demonstrates additive homomorphism that is based on the property of the exponential function $y = a^x$. For example, let $y_1 = a^{x_1}$ and $y_2 = a^{x_2}$ be two exponential functions. Then, $y_1 y_2 = a^{x_1 + x_2}$, which maintains the operation. This generates an additively homomorphic property. In Table 2, it is easy to see that, in the Damgard-Jurik, Naccache-Stern, Okamotto-Uchiyama, and BGN cryptosystems, there is a factor in the ciphertext. With c_1 and c_2, one can easily obtain $\text{Decrypt}(\text{sk}, c_1 \cdot c_2) = m_1 + m_2$; thus,

Table 2 Paillier class homomorphic cryptosystems.

Crypto-system	Encryption function	Homomorphism
Paillier	$c = g^m r^n (\mathrm{mod}\ n^2)$[11]	Additively homomorphic
Damgard-Jurik	$c = g^m r^{n^s} (\mathrm{mod}\ n^{s+1})$[12]	Additively homomorphic
Naccache-Stern	$c = g^m (\mathrm{mod}\ n)$[13]	Additively homomorphic
Okamotto-Uchiyama	$c = g^m h^r (\mathrm{mod}\ n)$[14]	Additively homomorphic
BGN	$c = g^m r^n (\mathrm{mod}\ n^2)$[15]	Additively homomorphic

they are additively homomorphic. Thus, we present the following theorem.

Theorem 4 One can obtain additive homomorphism by placing the plaintext at the location of the exponent.

3.3 ElGamal class of homomorphic encryption schemes

The ElGamal homomorphic encryption schemes include the ElGamal cryptosystem[16] and the Elliptic Curves Cryptosystem (ECC)[17]. The ElGamal cryptosystem is formed on the finite multiplicative group, and the ECC is constructed on the finite multiplicative group on an elliptic curve. The security of these schemes is based on the intractability of the discrete logarithm problem. Here, we analyze this type of homomorphic map and the ElGamal cryptosystem.

keygen($\cdot$): Choose large prime p, and let α be the generator of Z_p^*. Choose randomly an integer d with $1 < d < p-2$, and then compute $\beta = \alpha^d \bmod p$. The public key is (α, β) and the secret key is d.

Encrypt($\cdot$): takes as input the public key and a plaintext $m \in Z_p^*$. Choose randomly and secretly an integer k with $1 \leqslant k \leqslant p-2$, and then compute $c_1 = \alpha^k \bmod p$ and $c_2 = m\beta^k \bmod p$. This outputs ciphertext $c = (c_1, c_2)$.

Decrypt($\cdot$): takes as input the secret key and a ciphertext c. This outputs $m = c_2 \cdot (c_1^d)^{-1} (\mathrm{mod}\ p)$.

Theorem 5 (multiplicatively homomorphic property) The ElGamal cryptosystem is multiplicatively homomorphic.

Proof Let m and m' be two plaintexts, where c and c' are two ciphertexts that correspond to m and m', respectively, i.e.,

$$c = (c_1, c_2) = (\alpha^k \bmod p, m\beta^k \bmod p)$$

and

$$c' = (c_1', c_2') = (\alpha^{k'} \bmod p, m'\beta^{k'} \bmod p),$$

$$c \cdot c' = (c_1 \cdot c_1', c_2 \cdot c_2') =$$

$$(\alpha^k \alpha^{k'} \bmod p, mm'\beta^k \beta^{k'} \bmod p) =$$

$$(\alpha^{k+k'} \bmod p, mm'\beta^{k+k'} \bmod p),$$

$$\mathrm{Decrypt}(\mathrm{sk}, c \cdot c') =$$

$$c_2 c_2' \cdot ((c_1 c_1')^d)^{-1} (\mathrm{mod}\ p) =$$

$$(mm'\beta^{k+k'})(\alpha^{(k+k')d})^{-1} \bmod p =$$

$$m \cdot m'.$$

Therefore, the ElGamal cryptosystem is multiplicatively homomorphic. ■

The homomorphic property lies in the integer multiplication and exponential function; therefore, we call this hybrid class of encryption schemes homomorphic encryption. The main idea is to conceal the plaintext with another factor (i.e., the exponential function) so as to generate homomorphism. Table 3 shows that in the ElGamal and ECC cryptosystems the plaintext is concealed by an extra factor and that they are multiplicatively homomorphic. According to this line of thinking, we have the following theorem.

Theorem 6 One can obtain homomorphism by approperly concealing plaintext with some other factors.

Besides the above three classes, there are other means to design homomorphic cryptosystem, such as by using a bilinear map[15] and matrix operations[6] in linear error correcting code cryptosystems.

4 Proposed Targeted Fully Homomorphic Encryption Scheme

The RSA cryptosystem is multiplicatively homomorphic, and the BGN cryptosystem is additively homomorphic. Here, we describe the proposed public key encryption system.

Table 3 ElGamal class of homomorphic cryptosystems.

Crypto-system	Encryption function	Homomorphism
ElGamal	$c = (\alpha^k \bmod p, m\beta^k \bmod p)$[16]	Multiplicatively homomorphic
ECC	$c = (l \cdot G, (x, y) = l \cdot P, mx)$ where $P = a \cdot G$[17]	Multiplicatively homomorphic

4.1 The double decryption algorithm based scheme

The proposed scheme resembles the RSA and the BGN schemes and, by introducing a new component to the ciphertext, the proposed scheme can be targeted fully homomorphic. Since its decryption algorithm consists of two algorithms, we refer to the proposed scheme as the double decryption algorithm based scheme. The proposed scheme is as follows.

keygen($\cdot$): Choose two large primes p and q, and then compute $n = pq$. α is a generator of a multiplicative cyclic group of order n. Then compute $\beta = \alpha^p \bmod n$. The public key is (α, β) and the secret key is p.

Encrypt($\cdot$): takes as input the public key and a plaintext $m \in Z_n^*$. Choose randomly and secretly an integer k with $\beta \neq 1 \bmod n$. If this condition is not satisfied, then choose integer k again. Then, compute $c_1 = \alpha^k \bmod n$, $c_2 = m\beta^k \bmod n$, and $c_3 = \beta^{k+m} \bmod n$. This outputs ciphertext $c = (c_1, c_2, c_3)$.

Decrypt($\cdot$): takes as input the secret key and a ciphertext c. The decryption is chosen between the two algorithms as follows.

$\mathrm{Dec}_1(c) = c_2 \cdot (c_1^p)^{-1} (\bmod n)$.

$\mathrm{Dec}_2(c)$: This performs the following steps.

Step 1: $c_0 \equiv c_3 \cdot (c_1^p)^{-1} \bmod n \equiv \beta^m \bmod n = (\alpha^p)^m \bmod n$.

Step 2: Let $g = \alpha^p$. To recover m, it is sufficient to compute the discrete log of c_0 base g. Since $0 \leqslant m < n$. this computation is expected to require $\tilde{O}(\sqrt{n})$ time using Pollard's lambda method[18].

The $\mathrm{Dec}_2(c)$ algorithm takes polynomial time in n. Thus, the double decryption algorithm based scheme can only be used to encrypt short message. Clearly, the best way to employ this scheme is by means of a "digtial envelop".

4.2 Targeted fully homomorphic property

The cryptosystem proposed in Section 4.1 is designed according to the principles disscussed in Section 2. Here, we will analyze its homomorphic property with the following theorem.

Theorem 7 (targeted fully homomorphic property) The proposed cryptosystem is targeted fully homomorphic for a fixed form of polynomials and is fully homomorphic with several interactions.

Proof Let m and m' be two plaintexts, where c and c' are two ciphertexts that correspond to m and m', respectively, i.e.,

$$c = (c_1, c_2, c_3) = (\alpha^k \bmod n, m\beta^k \bmod n, \beta^{k+m} \bmod n)$$

and

$$c' = (c_1', c_2', c_3') = (\alpha^{k'} \bmod n, m'\beta^{k'} \bmod n, \beta^{k'+m'} \bmod n),$$

$$c \cdot c' = (c_1 \cdot c_1', c_2 \cdot c_2', c_3 \cdot c_3') = (\alpha^k \alpha^{k'} \bmod n, mm'\beta^k \beta^{k'} \bmod n, \beta^{k+m} \beta^{k'+m'} \bmod n) = (\alpha^{k+k'} \bmod n, mm'\beta^{k+k'} \bmod n, \beta^{k+k'+m+m'} \bmod n),$$

$$\mathrm{Decrypt}(c \cdot c') = \mathrm{Decrypt}(c_1 \cdot c_1', c_2 \cdot c_2', c_3 \cdot c_3') = \mathrm{Decrypt}(\alpha^{k+k'} \bmod n, mm'\beta^{k+k'} \bmod n, \beta^{k+k'+m+m'} \bmod n).$$

If one selects $\mathrm{Dec}_1(c)$, then

$$\mathrm{Dec}_1(c \cdot c') = \mathrm{Dec}_1(\alpha^{k+k'} \bmod n, mm'\beta^{k+k'} \bmod n) = (mm'\beta^{k+k'})(\alpha^{(k+k')p})^{-1} \bmod n = m \cdot m'.$$

Thus, it is exponentially homomorphic. If one chooses $\mathrm{Dec}_2(c)$, then

$$\mathrm{Dec}_2(c \cdot c') = \mathrm{Dec}_2(c_1 \cdot c_1', c_3 \cdot c_3') = \mathrm{Dec}_2(\alpha^{k+k'} \bmod n, \beta^{k+k'+m+m'} \bmod n).$$

Therefore, one can easily obtain the plaintext $m + m'$ using Pollard's lambda method. Namely, by choosing $\mathrm{Dec}_2(c)$ the proposed cryptosystem is linearly homomorphic.

From the above analysis, anyone can obtain the sum or product of several plaintexts from the corresponding ciphertexts. For example, let $m^1, m^2, \cdots, m^k$ be plaintexts and let $c^1, c^2, \cdots, c^k$ be the ciphertexts that correspond to $m^1, m^2, \cdots, m^k$, respectively. First, one can compute $c^1 \cdot c^2 \cdot \cdots \cdot c^k = (c_1^1 \cdot c_1^2 \cdot \cdots \cdot c_1^k, c_2^1 \cdot c_2^2 \cdot \cdots \cdot c_2^k, c_3^1 \cdot c_3^2 \cdot \cdots \cdot c_3^k)$, and then choose $\mathrm{Dec}_1(c)$ or $\mathrm{Dec}_2(c)$ according to the polynomial operated on the plaintexts, i.e., we can always obtain the desired plaintext by choosing the proper decryption algorithm.

One can easily obtain targeted fully homomorphic encryption when the polynomial has proper form, such as a homogeneous polynomial or an equal-length polynomial. It should be noted that targeted fully

homomorphic encryption can be fully homomorphic with the interaction, regardless of the form of the operation polynomial. This can be analyzed using two conditions.

First, we consider the polynomial a homogeneous polynomial. The user sends the attribute of the homogeneous polynomial $f(x_{s,i}) = \sum_{s=1}^{l}(\prod_{i=1}^{k} x_{s,i})$ to the untrusted server or proxy, and the untrusted server or proxy can compute and send $c_s = c^1 \cdot c^2 \cdot \cdots \cdot c^k = (c_1^1 \cdot c_1^2 \cdot \cdots \cdot c_1^k, c_2^1 \cdot c_2^2 \cdot \cdots \cdot c_2^k, c_3^1 \cdot c_3^2 \cdot \cdots \cdot c_3^k)$ to the user, where c^i are the ciphertexts that correspond to $x_{s,i}$. Then, the user performs $\mathrm{Dec}_1(c_1)$ for every $s = 1, 2, \cdots, l$ to obtain $x_s = \prod_{i=1}^{k} x_{s,i}$ (i.e., targeted fully homomorphic), re-encrypts $x_s = \prod_{i=1}^{k} x_{s,i}$ to obtain c'_s, and sends c'_s to the server or proxy. The sever or proxy can now obtain and return $c'_1 \cdot c'_2 \cdot \cdots \cdot c'_l$ to the user. Finally, the user can easily get $f(x_{s,i})$. Here, fully homomorphic encryption is achieved with the help of this interaction between the user and the server or proxy.

Second, we consider the polynomial an equal-length polynomial. The user and the server share the attribute $\sigma = (k, l)$ of $f(x_{s,i}) = \prod_{s=1}^{k}\left(\sum_{i=1}^{l} x_{s,i}^{d_i}\right)$. Similarly, it is targeted fully homomorphic, and the user can obtain $f(x_{s,i})$ by several interactions with the server. Thus, it is fully homomorphic.

Furthermore, any polynomial can be decomposed into several homogeneous polynomials and equal-length polynomials. Therefore, the scheme can be fully homomorphic whatever the operation polynomial is.

In conclusion, the proposed cryptosystem is targeted fully homomorphic for a fixed form of polynomials and fully homomorphic by several interactions with the server. ■

5 Semantic Security

The security of our scheme is based on the intractability of the discrete logarithm problem. To recover the secret key from the public key, one must solve the discrete logarithm problem. Obtaining the plaintext m from c_1 and c_2 or c_1 and c_3 without the secret key is also a discrete logarithm problem. The best known attack methods against the discrete logarithm problem include Shanks' algorithm[19], the Pollard-ρ algorithm[20], and the Polig-Hellman algorithm[21-23]. Here, we provide a simple proof of the semantic security of the proposed scheme based on the above discrete logarithm problem.

Theorem 8 (semantic security) The proposed targeted fully homomorphic encryption scheme is semantically secure if the discrete logarithm problem is intractable.

Proof Assume that our scheme is not semantically secure, i.e., there exists a polynomial time algorithm in which A (the adversary) who was given $(m_0, m_1, c = \mathrm{Encrypt}(\mathrm{pk}, m \in \{m_0, m_1\}))$ (without loss of generality, we assume that $m_0 < m_1$), can distinguish $E(\mathrm{pk}, m_0)$ and $E(\mathrm{pk}, m_1)$ with non-negligible probability.

We then construct a polynomial time machine B (the challenger) with the help of A as a black box that can distinguish $E(\mathrm{pk}, m_0)$ and $E(\mathrm{pk}, m_1)$ with non-negligible probability.

Assume a ciphertext $c = \mathrm{Encrypt}(\mathrm{pk}, b) = (c_1, c_2, c_3) = (\alpha^k \bmod n, b\beta^k \bmod n, \beta^{k+b} \bmod n)$, which is either $\mathrm{Encrypt}(\mathrm{pk}, 0)$ or $c = \mathrm{Encrypt}(\mathrm{pk}, 1)$. When A requests a challenge ciphertext, B computes, $c' = \{\alpha^k \bmod n,\ (b(m_1 - m_0) + m_1)\beta^k \bmod n,\ \beta^{k+b(m_1-m_0)+m_1} \bmod n\}$ randomizes c' into $c'' = \{\alpha^k \cdot \alpha^{k'} \bmod n,\ (b(m_1 - m_0) + m_1)\beta^k \beta^{k'} \bmod n,\ \beta^{k+b(m_1-m_0)+m_1}\beta^{k'} \bmod n\}$ and then sends c'' to A. If $c = \mathrm{Encrypt}(\mathrm{pk}, 0)$, then $c' = \mathrm{Encrypt}(\mathrm{pk}, m_0)$. If $c = \mathrm{Encrypt}(\mathrm{pk}, 1)$, then $c' = \mathrm{Encrypt}(\mathrm{pk}, m_1)$.

Adversary A returns an answer $c'' = \mathrm{Encrypt}(\mathrm{pk}, m_0)$ or $c'' = \mathrm{Encrypt}(\mathrm{pk}, m_1)$ that can distinguish $E(\mathrm{pk}, m_0)$ from $E(\mathrm{pk}, m_1)$. Thus, challenger B can judge $c = \mathrm{Encrypt}(\mathrm{pk}, 0)$, or $c = \mathrm{Encrypt}(\mathrm{pk}, 1)$, i.e., challenger B obtains $b = 0$ or $b = 1$. In the above interaction, the challenger solves the problem with the help of A.

Thus, our targeted fully homomorphic encryption scheme is semantically secure. ■

6 Conclusions

In this paper, methods for constructing homomorphic cryptosystems have been discussed. We have presented the design of a new targeted fully homomorphic encryption scheme that applies three components to ciphertexts and two alternative algorithms for decryption. The proposed scheme is the first and most efficient targeted fully homomorphic cryptosystem. The flaw of the proposed scheme is that we can only prove its semantic security. In addition, constructing a CCA1

security fully homomorphic cryptosystem remains an open problem.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61370188); Beijing Higher Education Young Elite Teacher Project; Fundamental Research Funds for the Central Universities (Nos. 2014CLJH09 and 2014GCYY05); Research Funds of Information Security Key Laboratory of Beijing Electronic Science and Technology Institute. The authors thank the anonymous referees for their helpful comments.

References

[1] C. Gentry, Fully homomorphic encryption using ideal lattices, in *Proc. STOC'09 Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, New York, USA, 2009, pp. 169-178.

[2] C. Gentry, A fully homomorphic encryption scheme, http://crypto stanfordedu/craig, 2009.

[3] M. S. Lee, On the sparse subset sum problem from Gentry-Halevi's implementation of fully homomorphic encryption, http://eprint.iacr.org/2011/567.pdf, 2011.

[4] M. V. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, Fully homomorphic encryption over the integers, presented at the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Riviera, French, 2010.

[5] J. H. Cheon, J. S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, Batch fully homomorphic encryption over the integers, presented at the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, 2013.

[6] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, Fully homomorphic encryption without bootstrapping, http://eprint.iacr.org/2011/277, 2011.

[7] R. Revist, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, vol. 26, no. 1, pp. 96-99, 1983.

[8] M. O. Rabin, Digital signatures and public-key encryptions as intractable as factorization, MIT, Technical Report, MIT/LCS/TR-212, 1979.

[9] H. C. Williams, A modification of the RSA public-key encryption procedure, *IEEE Trans. on Inform. Theory*, vol. 40, no. 6, pp. 726-729, 1980.

[10] H. C. Williams, Some public-key crypto-functions as intractable as factorization, in *the Proceedings of CRYPTO 84*, New York, USA, 1984.

[11] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, presented at the International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech, 1999.

[12] I. Damgard and M. Jurik, A generalisation, a simplification, and some applications of Paillier's probabilistic public-key system, presented at the 4th International Workshop on Practice and Theory in Public Key Cryptosystems, Cheju Island, Korea, 2001.

[13] D. Naccache and J. Stern, A new public-key cryptosystem based on higher residues, presented at the 5th ACM Conference on Computer and Communications Security, Riviera, French, 1998.

[14] T. Okamoto and S. Uchiyama, A new public-key cryptosystem as secure as factoring, presented at the International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, 1998.

[15] D. Boneh, E. J. Goh, and K. Nissim, Evaluating 2-DNF formulas on ciphertexts, presented at the Second Theory of Cryptography Conference, Cambridge, MA, USA, 2005.

[16] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. on Information Theory*, vol. 31, no. 4, pp. 469-472, 1985.

[17] J. M. Alfred and A. V. Scott, Elliptic curve cryptosystems and their implementation, *Journal of Cryptology*, vol. 6, no. 4, pp. 209-224, 1993.

[18] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.

[19] S. R. Blackburn and E. Teske, Baby-step giant-step algorithms for non-uniform distributions, *Lecture Notes in Computer Science*, vol. 1838, pp. 153-168, 2000.

[20] J. H. Cheon, J. Hong, and M. Kim, Speeding up the pollard rho method on prime fields, *Lecture Notes in Computer Science*, vol. 5350, pp. 471-488, 2008.

[21] S. C. Polig and M. E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, *IEEE Transaction on Information Theory*, vol. 24, no. 1, pp. 106-110, 1978.

[22] M. Jersak, K. Richter, and R. Ernst, Performance analysis for complex embedded applications, *International Journal of Computational Science and Engineering*, vol. 1, no. 1/2, pp. 33-49, 2005.

[23] Q. N. T. Do and F. K. Hussain, A hybrid approach for the personalisation of cloud-based e-governance services, *International Journal of High Performance Computing and Networking*, vol. 7, no. 3, pp. 205-214, 2013.

Yatao Yang received the PhD degree from Beijing University of Posts and Telecommunications in 2009 and is currently a master's supervisor of Beijing Electronic Science and Technology Institute. Up to now, he has published more than 30 papers. His interest includes homomorphic cryptosystems and design of cryptographic protocol and algorithm.

Shuang Zhang is studying for a master's degree of cryptography at Xidian University. He received his BEng degree from Zhengzhou University of Light Industry. His research interests include homomorphic cryptosystems and post quantum cryptography.

Junming Yang is studying for a master's degree of cryptography at Beijing Electronic Science and Technology Institute. He got his BEng degree from Anhui Jianzhu University. His research interests include homomorphic cryptosystems and FPGA implementation of cryptographic algorithms.

Jia Li received the PhD degree from Academy of Mathematics and System Sciences in the Chinese Academy of Sciences in 2009. Up to now, she has published more than 10 papers. Her research interests include algebraic curves and surfaces and homomorphic cryptosystems.

Zichen Li received the PhD degree from Beijing University of Posts and Telecommunications in 1999. Up to now, he has published more than 100 papers. His research interests include post quantum cryptography, homomorphic cryptosystems, digital signature, and design of cryptographic protocol and algorithm.