Available online at www.sciencedirect.com

ScienceDirect

Cognitive Systems Research 52 (2018) 543–549

Cognitive Systems RESEARCH

www.elsevier.com/locate/cogsys

A public key size homomorphic encryption scheme based on the sum of sparse subsets and integers

Jing Yang *, Mingyu Fan, Guangwei Wang

School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Received 2 June 2018; received in revised form 6 July 2018; accepted 10 July 2018
Available online 10 August 2018

Abstract

The paper proposes a homomorphic encryption scheme with public key size based on summation integer of sparse subset. The full-homomorphic encryption scheme that applies the batch processing technology to the integer can homomorphically process and encrypt a plaintext vector in a ciphertext to improve the efficiency of the original scheme, yet its size of the public key is $\tilde{O}(\lambda^8)$. In an effort to reduce the size of public key for this scheme, we combine quadric form of public key elements and ciphertext compression to present SomeWhat homomorphic public key scheme, which reduces the security of public key scheme into the approximate integer GCD problem, thereby converting the homomorphic encryption scheme into full homomorphic encryption scheme. For the proposed homomorphic encryption scheme with public key size based on summation integer of sparse subset, the public key size for improve scheme is $\tilde{O}(\lambda^{5.5})$, a smaller size. Lastly, the scheme is proved to be semantically secure.

Keywords: Full homomorphic encryption; Batch processing; Size of public key; Ciphertext compression; Quadric form

1. Introduction

In 1979, Rivest et al. suggested the concept of "privacy homomorphism" according to the characteristic of multiplicative homomorphism in RSA public key encryption system, i.e., it can directly operate the ciphertext instead of operating it after decrypting the ciphertext. The cryptological scholars made numerous studies in the past three decades, yet they failed to suggest the schemes with characteristic of full homomorphism, i.e. the schemes cannot operate the ciphertext with arbitrary complexity to realize corresponding operation for the plain text. In 2009, The IBM researcher Graig Gentry used ideal lattice to construct the first full homomorphic encryption scheme (Arunkumar et al., 2013; Faig et al., 2017), whose structure is in the several steps as follows: Firstly, constructing a "SomeWhat homomorphic encryption scheme", as the ciphertext is added with noise whose degree increases with additive or multiplicative homomorphic operations for ciphertext, and only when the noise degree is lower than a certain threshold can the decryption be correctly conducted, thus this scheme can only realize addition or multiplication operation with limited times for the ciphertext; secondly, compressing the decryption circuit to reduce the complexity of decoding algorithm to obtain the bootstrapping, and using the re-encryption technique to refresh the ciphertext to lower the ciphertext noise, thereby making the ciphertext noise within allowable scope, and these refreshed ciphertexts can be subjected to homomorphic operation, refreshing the ciphertext relentlessly can realize homomorphic operation with infinite times for ciphertext, thereby realizing a full homomorphic encryption scheme. As the homomorphic encryption is widely applied in

* Corresponding author.
E-mail address: jingjinggy68@163.com (J. Yang).

https://doi.org/10.1016/j.cogsys.2018.07.011
1389-0417/

protecting security and privacy of user data in the cloud computing environment, ciphertext retrieval and processing and multiparty computation, the full homomorphic encryption has become the hotspot of the research in cryptological domain since Gentry made breakthrough in full homomorphic encryption.

2. Symbols definition

2.1. Symbols and parameters definition

For a real number $\lceil x \rceil$, $\lfloor x \rfloor$, {FLD3} and $\lfloor x \rceil$ respectively represent upper, lower and near roundness for the real number x. For a real number z and an integer p, $q_p(z)$ and $r_p(z)$ or $[z]_p$ are respectively used to represent the quotient and remainder obtained by dividing z by p. A safety parameter λ is given, and the parameters we need to use are: the bit length γ for integer x_i in public key; bit length η of private key p; bit length ρ of noise r_i; number τ of x_i in the public key; and the secondary noise parameter ρ' in encrypting process.

The paper adopts Greek letters to represent parameters. Wherein, λ is the security parameter. The real numbers and integers are represented by lowercase English letters. For a real number z, $\lceil z \rceil$ represents upper integer, i.e. $\lceil z \rceil \in [z, z+1)$, $\lfloor z \rfloor$ represents lower integer, i.e. $\lfloor z \rfloor \in (z-1, z]$, $\lfloor z \rceil$ represents the nearest integer, i.e. $\lfloor z \rceil \in \left(z-\frac{1}{2}, z+\frac{1}{2}\right]$, $[z]$ represents taking the integral part of z, $\{z\}$ represents taking the decimal part of z, i.e. $\{z\} = z - [z]$. For a real number z and an integer p, $q_p(z)$ and $r_p(z)$ respectively represent the quotient and remainder obtained by dividing z by p, i.e. $q_p(z) = \lfloor z/p \rceil$, $r_p(z) = z - q_p(x) \cdot p$. It is obvious that $r_p(z) \in (-p/2, p/2]$. The paper adopts $[z]_p$ or $z \bmod p$ to represent modeling z with p.

2.2. Summation for sparse subset

According to the full homomorphic encryption scheme proposed by Gentry in his doctoral thesis, the full homomorphic encryption is to obtain bootstrapping by compressing the decryption circuit based on the SomeWhat homomorphic encryption scheme, then to realize the full homomorphic encryption through bootstrapping conversion. The BDGHV scheme is also constructed according to such idea. Here we only give an outline of the optimized SomeWhat homomorphic encryption scheme: KeyGen (1^λ): A prime number set $p_0, \ldots, p_{l-1}$ is generated, where, the bit length of p_i is η, π represents their product. A noiseless public key element $x_0 = q_0 * \pi$ is defined, where, q_0 meets following condition: $q_0 \leftarrow \mathbb{Z} \cap [0, 2^\gamma/\pi)$, not including prime number factor and smaller than 2^{λ^2}. The integers x_i, x'_i and Π_i distributed uniformly and independently in $\mathbb{Z} \cap [0, q_0)$ are generated, and when $0 \leqslant j \leqslant l-1$, the following is met:

When $1 \leqslant i \leqslant \tau$, $x_i \bmod p_j = 2r_{i,j}$, where $r_{i,j} \leftarrow \mathbb{Z} \cap (-2^{\varphi'-1}, 2^{\varphi'-1})$.

When $0 \leqslant i \leqslant l-1$, $x'_i \bmod p_j = 2r'_{i,j} + \delta_{i,j}$, where $r'_{i,j} \leftarrow \mathbb{Z} \cap (-2^{\varphi}, 2^{\varphi})$.

When $0 \leqslant i \leqslant l-1$, $\Pi_i \bmod p_j = 2\varpi_{i,j} + \delta_{i,j} \cdot 2^{\varphi'+1}$, where $\varpi_{i,j} \leftarrow \mathbb{Z} \cap (-2^{\varphi}, 2^{\varphi})$. So, public key $pk = \langle x_0, (x_i)_{0 \leqslant i \leqslant \tau}, (x'_i)_{0 \leqslant i \leqslant l-1}, (\Pi_i)_{0 \leqslant i \leqslant l-1} \rangle$, and private key $sk = (p_j)_{0 \leqslant j \leqslant l-1}$.

Encrypt $(pk, m \in \{0,1\}^l)$: Selecting random integer vector $\boldsymbol{b} = (b_i)_{1 \leqslant i \leqslant \tau} \in (-2^\alpha, 2^\alpha)^\tau$ and $\boldsymbol{b} = (b'_i)_{0 \leqslant i \leqslant l-1} \in (-2^{\alpha'}, 2^{\alpha'})^l$, calculate cyphertext $c = \left[\sum_{i=0}^{l-1} m_i \cdot x'_i + \sum_{i=0}^{l-1} b'_i \cdot \Pi_i + \sum_{i=1}^{\tau} b_i \cdot x_i\right]$.

Decrypt(sk, c): Output $\mathbf{m} = (m_0, \ldots, m_{l-1})$, where $m_j \leftarrow [c]_{p_j} \bmod 2$.

Add(pk, c_1, c_2): Output $c_1 + c_2 \bmod x_0$.

Mult(pk, c_1, c_2): Output $c_1 \cdot c_2 \bmod x_0$.

Wherein, the parameters in the scheme should meet the following constraint conditions: $\varphi \geqslant 2\lambda$ to avoid the brutal force attack on noise; $\eta \geqslant \alpha' + \varphi' + 1 + \log_2(l)$ to guarantee correctness of decryption; $\eta \geqslant \rho \cdot \Theta(\lambda \log^2 \lambda)$ to support homomorphic operation in evaluating "compression of decryption circuit"; $\gamma = \omega(\eta^2 \cdot \log \lambda)$ to prevent the lattice-based attack; $\rho' \geqslant \rho + \lambda$ and $\alpha' \geqslant \alpha + \lambda$ to guarantee semantic security of scheme; $\alpha \cdot \tau \geqslant \gamma + \lambda$ and $\tau \geqslant l \cdot (\rho' + 2) + \lambda$ to be able to apply the leftover hash lemma in proving the semantic security of scheme; to meet the above constraints, following parameters can be selected: $\varphi = 2\lambda$, $\varphi' = 3\lambda$, $\eta = \tilde{O}(\lambda^2)$, $\gamma = \tilde{O}(\lambda^5)$, $\alpha = \tilde{O}(\lambda^2)$, $\alpha' = \tilde{O}(\lambda^2)$, $l = \tilde{O}(\lambda^2)$, $\tau = \tilde{O}(\lambda^3)$, where, λ is the security parameter.

According to the above parameter setting, the size of public key is $\tilde{O}(\lambda^8)$. As a matter of fact, according to the description in literature Malarkodi, Arunkumar, and Venkataraman (2013), to prevent lattice-based attack, the bit length γ of public key x_i should be at least of 2^{23} bits. While now, the size of public key is at least of 2^{46} bits, and such a big public key are impossible to be applied in any actual cryptosystem.

2.3. Ciphertext size

To decrease the size of ciphertext in the process of encrypting, metatype x_0 is accumulated. However, such processing cannot be done in calculating ciphertext, as even only one multiplication of ciphertext is executed, the result would be far greater than x_0, and if model treatment is conducted, a big x_0 times would be added or deducted, which would lead to the intolerable errors.

In calculation of ciphertext, to reduce the size of ciphertext, we add more elements like $x'_i = q'_i p + r'_i$ to the public key, where, r'_i is selected from interval $[-2^\rho, 2^\rho]$ like usual noise, while q'_i is far greater than the value of q among