Improved Homomorphic String Bits Encryption Public-key Cryptosystem based on LWE

Bai Jian
Communication Engineering Institute, Xidian University
Xi'an, Shanxi 710071, China
Beijing Electronic Science and Technology Institute
Beijing 100070, China

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

Li Zichen
Beijing Electronic Science and Technology Institute
Beijing 100070, China

Abstract—**Accompany with the developing of the cloud computing, a public-key cryptosystem which is efficiency and homomorphic will have a wide application. Through analyzing the public-key cryptosystem, which is designed by Oded Regev and based on LWE (Learning with errors), in details, our main results are optimizing this public-key cryptosystem for bits string encryption and designing some good idea to make the optimized public-key cryptosystem satisfy the additive homomorphism and mixed multiplicative homomorphism. And also we give a small example and a time simulation about the improved public-key cryptosystem. The small example shows the cryptosystem can encrypt and decrypt correctly and the time simulation tells us the time of the generation of key and the decryption is the same with the original cryptosystem, but the encryption is more efficiency than the original cryptosystem.**

Keywords-LWE; Lattice; Public-key Cryptosystem; Homomor
-phism; Regev's Cryptosystem

I. INTRODUCTION

Ever since the seminal work of Ajtai[1] connecting the average-case complexity of lattice problems to their complexity in the worst case, basing cryptography on worst-case lattice assumptions has intriguing and fruitful achievements[2-8].

In addition to their unique theory, lattice-based schemes enjoy many advantages. The first is their asymptotic efficiency and simplicity, which usually requires only linear operations on small integers. Secondly, they can resist cryptanalysis from quantum algorithms. The last is the guarantee that their random instances are "as hard as possible". In May 2, 2009, Oded Regev presented the LWE (Learning with Errors) and the first public Cryptosystem[2] based on LWE. Gentry gave an optimized version of the system[3] in which all users share a common matrix $\mathbf{A} \in \mathbf{Z}_q^{n\times m}$ chosen uniformly and randomly. Assuming the worst-case hardness, which approximates the minimum distance in n -dimensional lattices within small poly (n) factors, Peikert also constructs a public-key cryptosystem[4] that are secure.

In this paper, we mainly optimize the public-key cryptosystems presented by Oded Regev. In section 2, some basic definitions are given. And in section 3 we transform the cryptosystem into bits string encryption instead of single bit encryption. We would give the analysis of homomorphic properties about the public-key cryptosystem in section 4. A simple example and an efficiency analysis of our improved scheme are given in section 5.

II. PRELIMINARIES

In this section, we introduce some basic definitions and concepts that will be used throughout the paper.

A. The definitions of homomorphic properties

At first, the definitions of homomorphic properties are explained in the following.

Definition 1[9] Let R and S is rings. We call an (encryption) function $E: R \to S$

1) additively homomorphic, if there is an efficient algorithm PLUS to compute $E(x+y)$ from $E(x)$ and $E(y)$ that does not reveal x and y.
2) mixed multiplicatively homomorphic, if there is an efficient algorithm MIXED-MULT to compute $E(xy)$ from $E(x)$ and y that does not reveal x.
3) multiplicatively homomorphic, if there is an efficient algorithm MULT to compute $E(xy)$ from $E(x)$ and $E(y)$ that does not reveal x and y.

B. The definition of Learning with errors (LWE)

The public-key cryptosystem, we will talk later, is based on the LWE. The definition of LWE is:

Definition 2[3]:For an integer $q = q(n)$ and a distribution χ on $\mathbf{Z}_q$, the goal of the (average-case) learning with error problem $\mathrm{LWE}_{q,\chi}$ is to distinguish (with nonnegligible probability) between the distribution $\mathbf{A}_{\mathbf{s},\chi}$ for some uniform (secret) $\mathbf{s} \leftarrow \mathbf{Z}_q^n$ and the uniform distribution on $\mathbf{Z}_q^n \times \mathbf{Z}_q$ (via oracle access to the given distribution). In other words, if LWE is hard, the collection of distributions $\mathbf{A}_{\mathbf{s},\chi}$ is pseudorandom.

C. The public-key cryptosystem presented by Oded Regev

Now, we give the construction of the public–key cryptosystem presented by Oded Regev. n is the security parameter of the cryptosystem. The cryptosystem is parameterized by two integers m, p and a probability distribution χ on $\mathbf{Z}_p$. The setting of these parameters guarantees both security and correctness. Choose a prime $p \geq 2$ between n^2 and $2n^2$ and let $m = (1+\varepsilon)(n+1)\log p$ where ε is an arbitrary constant and $\varepsilon > 0$. The probability distribution χ is taken to be $\bar{\Psi}_{\alpha(n)}$, where $\alpha(n) = o(1/(\sqrt{n}\log n))$, i.e., $\alpha(n)$ satisfies $\lim_{n\to\infty} \alpha(n)\cdot\sqrt{n}\log n = 0$. For example, we can choose $\alpha(n) = 1/(\sqrt{n}\log n)$. In the following description, all additions are performed in $\mathbf{Z}_p$, i.e., mod p.

Private Key: Choose $\mathbf{s} \in \mathbf{Z}_p^n$ uniformly and randomly. The private key is $\mathbf{s}$.

Public Key: For $i = 1,...,m$, choose m vectors $\boldsymbol{a}_1,...,\boldsymbol{a}_m \in Z_p^n$ independently from the uniform distribution. Also choose elements $e_1,...,e_m \in \mathbf{Z}_p$ independently according to χ. The public key is given by $(\boldsymbol{a}_i, b_i)_{i=1}^m$ where $b_i = <\boldsymbol{a}_i, \boldsymbol{s}> + e_i$.

Encryption: In order to encrypt a bit, we choose a random set S uniformly among all 2^m subsets of $[m]$. The encryption is $(\sum_{i\in S}\boldsymbol{a}_i, \sum_{i\in S} b_i)$ if the bit is 0 or $(\sum_{i\in S}\boldsymbol{a}_i, \left\lfloor \frac{p}{2} \right\rfloor \sum_{i\in S} b_i)$ if the bit is 1.

Decryption: The decryption of a pair $(\boldsymbol{a}, b)$ is 0 if $b - <\boldsymbol{a}, \mathbf{s}>$ is closer to 0 than to $\left\lfloor \frac{p}{2} \right\rfloor \bmod p$. Otherwise, the decryption is 1.

III. Optimize the Public-key Cryptosystem Presented by Oded Regev to Bits String Encryption

In this section we will adapt the cryptosystem to bits string encryption, and also the proof of the correctness and security are presented.

A. The improvement of the cryptosystem presented by Oded Regev

The original cryptosystem is bit encryption. We now want to change it into bits encryption. The way how to optimize is inspired by Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem[4].

In our scheme, the part to generate key is same with the Cryptosystem above. We make change in the parts of encryption and decryption.

There is a bits string $\mathbf{m}^T = \{m_1, m_2, ..., m_n\} \in \{0,1\}^n$. The progresses of encryption and decryption are designed as follows:

Encryption: In order to encrypt a bits string we choose a random set S uniformly among all 2^m subsets of $[m]$. We let the $\mathbf{k} = \left\lfloor \frac{p}{2} \right\rfloor \mathbf{m} + \mathbf{b}'$, and the $\mathbf{b}'^T = \{\sum_{i\in S} b_i, \sum_{i\in S} b_i, ..., \sum_{i\in S} b_i\}$. The $\mathbf{m}$ is encrypted to $(\sum_{i\in S}\boldsymbol{a}_i, \mathbf{k})$.

Decryption: The decryption of a pair $(\boldsymbol{a}, \mathbf{k})$ is 0 if $k_i - <\boldsymbol{a}, \mathbf{s}>$ is closer to 0 than to $\left\lfloor \frac{p}{2} \right\rfloor \bmod p$. Otherwise, the decryption is 1. So we can get m_i. The $\mathbf{m}$ is recovered.

B. Correctness

According to our improvement, $k_i - <\boldsymbol{a}, \mathbf{s}>$ is been computed one by one, the correctness is approximately same to the cryptosystem presented by Oded Regev. So the correctness of single bit encryption is given firstly.

Correctness[2]: Let $\delta > 0$. Assume $\Pr_{e\square\chi^{*k}}\left[|e| < \left\lfloor \frac{p}{2} \right\rfloor / 2 \right] > 1-\delta$, for any $k \in \{0,1,...,m\}$ and χ^{*k}. Then, the probability of decryption error is at most δ. That is, for any bit $c \in \{0,1\}$, if we use the protocol above to choose private and public keys, encrypt c and decrypt the result, the probability of decrypting correctly is at least $1-\delta$.

The proof to the correctness above in details can be gotten in paper [2]. In decrypting we compute $k_i - <\boldsymbol{a}, \mathbf{s}> = m_i \cdot \left\lfloor \frac{p}{2} \right\rfloor + \sum_{i\in S} e_i$, so we can decrypt k_i to m_i. Finally, $\mathbf{m}$ is decrypted successfully.

C. Security

Our improvements are based on the public-key cryptosystem presented by Oded Regev. The improvements only change the steps of the original cryptosystem. The improved scheme is also based on LWE. The proof of security is the same to the public-key cryptosystem presented by Oded Regev.

Security[2]: For any $\varepsilon > 0$ and $m \geq (1+\varepsilon)(n+1)\log p$, if there is a polynomial time algorithm W which can distinguish between encryption of 0 and encryption of 1, then there will be a distinguisher Z which distinguishes between $A_{\mathbf{s},\chi}$ and U for a non-negligible fraction of all possible $\mathbf{s}$.

The detailed proof to the security above can be seeing in the paper [2].

IV. Homomorphic Analysis of the Public-key Cryptosystem Presented by Oded Regev

The homomorphism has three parts, which are additive homomorphism, mixed multiplicative homomorphism and multiplicative homomorphism. The public-key cryptosystem satisfies the additive homomorphism and mixed multiplicative homomorphism, but it isn't suit for the multiplicative homomorphism. The

three parts would be discussed one by one.

A. Additive homomorphism

Firstly, we discuss the additive homomorphism of this cryptosystem. If c_1 and c_2 is the encryption of m_1 and m_2, We have $c_1 = (\boldsymbol{a}, b_1)$ and $c_2 = (\boldsymbol{a}, b_2)$. The sum of c_1 and c_2 is $c = (\boldsymbol{a}, b_1 + b_2)$. We can get $c - 2 \cdot < \boldsymbol{a}, \mathbf{s} >= m_1 \cdot \left\lfloor \frac{p}{2} \right\rfloor + m_2 \cdot \left\lfloor \frac{p}{2} \right\rfloor + 2 \cdot \sum_{i \in S} e_i$. We are not sure that $\Pr_{e \sim \chi^{*k}} \left[2 \cdot |e| < \left\lfloor \frac{p}{2} \right\rfloor / 2 \right] > 1 - \delta$. In order to satisfy the additive homomorphism, we let $b_i = (< \boldsymbol{a}_i, \mathbf{s} > + e_i) / 2$ in the public key generation. Then, $c - < \boldsymbol{a}, \mathbf{s} >= m_1 \cdot \left\lfloor \frac{p}{2} \right\rfloor + m_2 \cdot \left\lfloor \frac{p}{2} \right\rfloor + \sum_{i \in S} e_i$ .If $m_1 = 0$ and $m_2 = 0$, $m = m_1 + m_2 = 0$, we can achieve that Encrypt[m]=Encrypt[m_1]+Encrypt[m_2]. This scene is suit for $m_1 = 0, m_2 = 1$ and $m_1 = 1, m_2 = 0$. When $m_1 = 1$ and $m_2 = 1$, $c - < \boldsymbol{a}, \mathbf{s} >= 2 \cdot \left\lfloor \frac{p}{2} \right\rfloor + \sum_{i \in S} e_i$ which will be decrypted for 1. This will be satisfy the additive homomorphism in the or additive, but it doesn't accord with additive homomorphism in addition modulo 2.

However, if $c - < \boldsymbol{a}, \mathbf{s} >\geq 2 \cdot \left\lfloor \frac{p}{2} \right\rfloor$, c is decrypted to 0. It will satisfy the addition modulo2.

B. Mixed multiplicative homomorphism

The public-key cryptosystem can also satisfy mixed multiplicative homomorphism. The decryption progress of this cryptosystem must add a small judge to satisfy the mixed multiplicative homomorphism. The judge is that whether $b = 0$ must be decided. If $b = 0$, c would be decrypted to 0.

If m_1 is 0 or 1 and $c_2 = (\boldsymbol{a}, b_2)$ is achieved by encrypting m_2, which is also 0 or 1, the result is 0 or b_2 when $c = (\boldsymbol{a}, m_1 \cdot b_2)$. Then we decrypt c as the decrypting system presented above, we can achieve $D(m_1 \cdot E(m_2)) = D(E(m_1 \cdot m_2))$ whether m_1 and m_2 is 0 or 1. So the public-key-cryptosystem satisfies mixed multiplicative homomorphism.

C. Multiplicative homomorphism

The public-key cryptosystem can not fulfill multiplicative homomorphism. The second part of $c = (\boldsymbol{a}, b)$, which is ciphertext of m, is

$$b = \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \sum_{i \in S} b_i = \left\lfloor \frac{p}{2} \right\rfloor \cdot m + \sum_{i \in S} (< \boldsymbol{a}_i, \mathbf{s} > + e_i) \quad .$$

Assuming we have $b_1 = \left\lfloor \frac{p}{2} \right\rfloor \cdot m_1 + < \boldsymbol{a}, \mathbf{s} > + \sum_{i \in S} e_i$ and $b_2 = \left\lfloor \frac{p}{2} \right\rfloor \cdot m_2 + < \boldsymbol{a}, \mathbf{s} > + \sum_{i \in S} e_i$ $b_1 \cdot b_2 = \left\lfloor \frac{p}{2} \right\rfloor \cdot m_1 \cdot \left\lfloor \frac{p}{2} \right\rfloor \cdot m_2 + \left\lfloor \frac{p}{2} \right\rfloor \cdot m_1 \cdot < \boldsymbol{a}, \mathbf{s} > + \left\lfloor \frac{p}{2} \right\rfloor \cdot m_1 \cdot \sum_{i \in S} e_i + < \boldsymbol{a}, \mathbf{s} > \cdot \left\lfloor \frac{p}{2} \right\rfloor \cdot m_2 + < \boldsymbol{a}, \mathbf{s} >^2 + < \boldsymbol{a}, \mathbf{s} > \cdot \sum_{i \in S} e_i + \sum_{i \in S} e_i \cdot \left\lfloor \frac{p}{2} \right\rfloor \cdot m_2 + \sum_{i \in S} e_i \cdot < \boldsymbol{a}, \mathbf{s} > + (\sum_{i \in S} e_i)^2$, in which $< \boldsymbol{a}, \mathbf{s} >^2$, $\sum_{i \in S} e_i \cdot < \boldsymbol{a}, \mathbf{s} >$ and $(\sum_{i \in S} e_i)^2$ can not be committed size. So we can not decrypt when we compute $b_1 \cdot b_2$. Maybe it can satisfy multiplicative homomorphism when make some bigger improvements.

V. Simple Example and Simulation

A. A simple example of the system optimized

A simple example of this public-key system is given below. We define $n = 10, m = 5, p = 991$. The private key is $\boldsymbol{s} = \{487, 204, 195, 460, 872, 975, 613, 57, 154, 743\}$ and the public key is

$\boldsymbol{a}_1 = \{284, 904, 76, 469, 696, 297, 637, 747, 924, 276\}$;

$\boldsymbol{a}_2 = \{692, 285, 287, 261, 796, 102, 327, 891, 810, 825\}$;

$\boldsymbol{a}_3 = \{137, 69, 413, 277, 404, 389, 848, 734, 662, 128\}$;

$\boldsymbol{a}_4 = \{921, 313, 244, 768, 308, 479, 443, 257, 897, 270\}$;

$\boldsymbol{a}_5 = \{840, 763, 681, 638, 691, 186, 167, 905, 952, 920\}$;

$\boldsymbol{b} = \{2230811, 2354217, 1779544, 2273804, 2759153\}$.

$\boldsymbol{m} = \{0, 1, 1, 0, 0\}$ will be encrypted. The second part of the ciphertext is

$\boldsymbol{b}' = \{6838691, 4282364, 4919905, 294630, 8514513\}$.

Then we decrypt the ciphertext and get $\boldsymbol{k} = \{5, 542, 491, 47, 147\}$. At the end, we can achieve the $m' = \{0, 1, 1, 0, 0\}$ after the decryption of the ciphertext.

B. Time Efficiency of the system optimized

An efficiency simulation is configured with an Intel(R) Core(TM)2 CPU(E8400 at 3.00GHz) and 1.96GB RAM. It is designed for $p = 197, n = 60, m = 100$. The size of information is 8bits. We measure the time-consuming of key generating, encryption and decryption for 10times and give the means in table 1.

TABLE I. The Comparison of Time Efficiency Between Regev's Scheme and Our Optimized Scheme

Time(s) / Item / Sch	*Key generating*	*Encryption*	*Decryption*
Regev's scheme	4.218×10^{-4}	1.200×10^{-5}	3.856×10^{-5}
Optimized scheme	4.356×10^{-4}	4.047×10^{-5}	3.913×10^{-5}

The figure one shows the time of Key generating and Decryption is roughly equal. But the time of Encryption in our scheme is less than the Regev's scheme.

And also we simulate the RSA in the same environment. The comparison between RSA and our scheme is showed table 2.

TABLE II. THE COMPARISON OF TIME EFFICIENCY BETWEEN RSA(1024) AND OUR OPTIMIZED SCHEME

Time(s) Item / Sch	*Key generating*	*Encryption*	*Decryption*
RSA (1024)	6.786	0.176	0.139
Optimized scheme	4.356×10-4	4.047×10-5	3.913×10-5

From table 2, we can see the scheme based on lattice is much faster than the RSA for its linear operation.

VI. CONCLUSION

The public-key cryptosystem based on lattice is so attractive because of its good properties. And also the homomorphism becomes hotspot for its advantages. The public-key homomorphism cryptosystem based on lattice will have a foreground in the future. Upon our changes, the public-key cryptosystem can be suit for the additive homomorphism, mixed multiplicative homomorphism. And also we change it into bits string encryption. But it can not satisfy the multiplicative homomorphism. So we have a large work to do on the homomorphic public-key cryptosystem based on lattice.

VII. ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (No.61070219): Research Funds of Information Security Key Laboratory of Beijing Electronic Science & Technology Institute.

REFERENCES

[1] Miklos Ajtai . Generating hard instances of lattice problems (extended abstract). In STOC, pages 99-108, 1996.

[2] Regev O. On lattices, learning with errors, random linear codes, and cryptography[J]. Journal of the ACM, 2009, 56(6):1-40.

[3] Gentry C, Peikert C, and Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions[C]. STOC'08, Victoria, BC, Canada, ACM, 2008:197-206.

[4] Peikert C, Vaikuntanathan V, and Waters B. A framework for efficient and composable oblivious transfer[C]. EUROCRYPT' 2008, 2008, LNCS, 5157:554-571.

[5] David C, Dennis H, Eike K, et al. Bonsai trees, or how to delegate a lattice basis[C]. EUROCRYPT'2010, LNCS, 6110: 523-552.

[6] Lyubashevsky V, Peikert C, and Regev O. On ideal lattices and learning with errors over rings[C]. EUROCRYPT'2010, 2010, LNCS, 6110:1-23.

[7] Peikert C. Public-key cryptosystems from the worst-case shortest vector problem[C]. STOC'09, Maryland, USA, ACM, 2009:333-342.

[8] Agrawal S, Boneh D, and Boyen X. Efficient lattice (H)IBE in the standard model[C]. EUROCRYPT'2010, 2010, LNS, 6110: 553-572.

[9] T.Sander and C. Tschudin. Towards mobile cryptography. In Proceedings of the IEEE Symposium on Security and Privacy. Oakland, CA, 1998. IEEE Computer Society Press.

[10] Craig Gentry. Toward Basing Fully Homomorphic Encryption on Worst-Case Hardness. CRYPTO 2010, LNCS 6223, pp. 116–137, 2010.