J Ambient Intell Human Comput (2015) 6:549–555
DOI 10.1007/s12652-014-0239-6

ORIGINAL RESEARCH

How to build a faster private information retrieval protocol?

Wei Zhang · Shuguang Liu · Weidong Zhong · Xiaoyuan Yang

Received: 11 November 2013 / Accepted: 20 April 2014 / Published online: 5 June 2014

Abstract A CPA secure multi-bit somewhat homomorphic encryption scheme based on Learning With Errors over Rings assumption is presented. We use canonical embedding to transform ring elements into vectors over Z_q, and thus decrease encryption and decryption cost. Comparing with GHV scheme appeared in 2010, to encrypt n bits, this scheme can reduce encryption cost from $O(n^{3/2})$ into $O(n\log n)$. Finally, an efficient private information retrieval protocol that employs this scheme is presented.

Keywords Homomorphic encryption · RLWE assumption · Canonical embedding · Private information retrieval

Abbreviations

PIR	Private information retrieval
FHE	Fully homomorphic encryption
SWHE	Somewhat homomorphic encryption
LWE	Learning with errors
RLWE	Learning with errors over rings

1 Introduction

Homomorphic encryption is a powerful cryptographic primitive that allow for a variety of applications. It is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which decrypted matches the result of operations performed on the plaintext. There are many interesting applications including private information retrieval (PIR), electronic voting, database encryption, delegated computation and secure multiparty computation (Chen et al. 2012a, b).

Fully homomorphic encryption (FHE) permits arbitrarily computation on encrypted data (Gentry 2009). During the past 4 years, numerous constructions of FHE involving novel mathematical techniques and a number of applications have appeared (Dijk et al. 2010; Stehle and Steinfeld 2010; Smart and Vercauteren 2010; Brakerski and Vaikuntanathan 2011a; Bogdanov and Lee 2011; Brakerski et al. 2012). However, it seems that most of the available FHE schemes still have a long way to go before they can be used in practice. Comparing with the theoretical perfect but unpractical FHE, somewhat homomorphic encryption (SWHE), which only permits a specific set of operations, seems more efficient, and most of the actual applications only involve SWHE schemes by now.

The main target of this work is to construct an efficient multi-bit somewhat homomorphic encryption scheme. Starting from Regev's Learning With Errors over Rings (RLWE)-based scheme (Regev 2009) and using canonical embedding to improve efficiency, we present a new construction of SWHE scheme that supports a larger plaintext space and faster encryption. Moreover, we provide a Private Block Retrieval (PBR) protocol using this scheme.

2 Related works

Boneh et al. (2005) described a cryptosystem (denoted by BGN) that permits arbitrary numbers of additions and one multiplication, without growing the ciphertext size. Later

W. Zhang (✉) · S. Liu · W. Zhong · X. Yang
Key Lab of Computer Network and Information Security Under CAPF, Xi'an, China
e-mail: zhaangweei@yeah.net

in EUROCRYPT 2010, Gentry et al. (2010) constructed a variant of BGN, called GHV, it is based on Learning With Errors (LWE) assumption, supports a larger message space and has a better message-to-ciphertext expansion ratio than BGN. In GHV, to encrypt m^2 bits, the encryption process has a computation cost of $\tilde{O}(m^3)$.

Aiming at constructing time-efficient schemes that supports larger message spaces, we present a multi-bit SWHE scheme that is basing on RLWE assumption. Comparing with GHV, our scheme are more time-efficient, to encrypt n bits, the total encryption cost is $\tilde{O}(n \log n)$. Such improvement attributes to the combination of the more compact RLWE assumption and canonical embedding. We show how to use this scheme to build an efficient PIR (private information retrieval) protocol.

3 Preliminaries

3.1 Homomorphic encryption schemes

Definition 1 A Homomorphic Encryption scheme (HE) can be described as a 4-tuple of algorithms HE = (KeyGen, Enc, Dec, Eval). The algorithms are probabilistic polynomial-time and satisfy the following properties:

- KeyGen(1^{λ}): given security parameter λ, output (pk, sk, evk), where pk and sk are public key and private key respectively, and evk is the public homomorphic evaluation key.
- Enc(pk, m): given the encryption key pk and a message m, the encryption algorithm outputs a ciphertext c, denoted by $c = \text{Enc}(pk, m)$.
- Dec(sk, c): given a ciphertext c and decryption key sk, output a plaintext m.
- Eval(evk, f, c_1, c_2…, c_l): Given the homomorphic evaluation key evk, a function f and l ciphertexts c_1, c_2…, c_l, output a ciphertext c_f, satisfying $c_f = \text{Enc}(pk, f(\text{Dec}(sk, c_1), \text{Dec}(sk, c_2), \ldots\ldots, \text{Dec}(sk, c_l)))$

This definition is a generic description of homomorphic encryption schemes, and the material of function f is omitted. Generally f can be expressed as a Boolean circuit on field GF(2^n), and only contains ADD and OR operations.

3.2 RLWE assumption

The LWE problem has gained a universal notice since it had been first introduceed by Regev in (2009). In Eurocrypt 2010, Lyubashevsky et al. (2010) analyzed the efficiency of LWE-based cryptosystems. For a standard LWE assumption, obtaining one pseudorandom scalar $b_i \in Z_q$ requires an n-dim inner production computation. They propose a more compact version of LWE called RLWE, that is, *LWE assumptions on a given ring*, where conducting an n-dim inner production can get another n-dim vector. This makes an efficiency improvement by n times.

Definition 2 (*RLWE* assumption) Let $f(x)$ be an n-degree polynomial with integer coefficients, q is a prime, and ring R_q is defined as $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$. Let χ be error distribution on R_q, $s \xleftarrow{\$} R_q$, $a_i \xleftarrow{\$} R_q$, $k = \text{poly}(n)$. For any given k pairs $(a_i, b_i = a_i s + e_i)_{i=1}^{k}$, where e_i is the error vector, then b_i is computationally indistinguishable from any uniformly chosen element in R_q.

Lyubashevsky et al. (2010) have proved that, the *Shortest Independent Vector Problem* (SIVP) or *Shortest Vector Problem* (SVP) in the worst case on ideal lattice can be reduced to RLWE. Their main result can be captured as the following: with error distribution be D_{ξ} and $\xi = \alpha \cdot (nl/\log(nl))^{1/4}$, given l samples, the RLWE problem is at least as hard as SIVP problem in a lattice.

To make the description more clear, we only use RLWE assumption on a special polynomial $R = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ where n is a power of 2 and $q = 1 \bmod 2n$.

3.3 Canonical embedding in polynomial rings

Canonical embedding was first proposed by Minkowski (Lyubashevsky et al. 2010). Let $n = 2^k$, $q = 1 \bmod 2n$ is a prime, and $\omega = \exp(\pi i/n)$, then canonical embedding is defined as a mapping σ from $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$into vector space on complex numbers $\mathbb{C}^n$, that is $a(x) \mapsto (a(\omega^1), a(\omega^3), \ldots, a(\omega^{2n-1})) \in \mathbb{C}^n$. Where $a(x) \in R_q$ and $f(x) = x^n + 1$.

Using canonical embedding, we can map a polynomial in $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$into a Ring vector. When a polynomial is mapped into a vector in $\mathbb{C}^n$, both addition and multiplication can be conducted coordinate-wise, thus making computation more convenient. Especially when q is a prime and $q = 1 \bmod 2n$, ω^{2i-1}, $i = 1, \ldots, n$-1 are just the n roots of $x^n + 1$ in $\mathbb{Z}_q$, so a polynomial $a(x) \in \mathbb{Z}_q[x]/\langle x^n + 1\rangle$can be mapped into an elements in $\mathbb{Z}_q^n$or a n-dim vector on $\mathbb{Z}_q$.

For a given $\sigma(a(x)) = (a(\omega^1), a(\omega^3), \ldots, a(\omega^{2n-1})) \in \mathbb{Z}_q^n$, we can get its preimage $a(x)$ by solving a linear equation set of n variables.

4 Multi-bit homomorphic encryption schemes based on RLWE assumption

4.1 The basic scheme

The first single-bit public key encryption scheme basing on LWE assumption was proposed by Regev in (2009), and

from this scheme, people have promoted some other constructions and applications. The multi-bit version of Regev's scheme can be implemented on RLWE assumption as the following (Rückert and Schneider 2010).

Scheme 1 (RLWE based version of Regev's multi-bit encryption scheme) Parameters: let q be a prime, $q \equiv 1 \mod 2n$, $R = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$, χ is discrete Gauss distribution. A sample that conforms to χ is noted by $e(x) \in R$ with $r \geq 1$. Define a set D_r as $D_r = \left(Z \cap \left\{-\left\lfloor \frac{r}{2} \right\rfloor, \ldots, \left\lceil \frac{r}{2} \right\rceil\right\}\right)/\langle x^n + 1\rangle$

For a positive integer k, define two operations on R^k:

1. Multiplication of two polynomial vectors $\otimes : R^k \times R^k \to R$: For any $\hat{x}, \hat{y} \in R^k$, $\hat{x} \otimes \hat{y} = \sum_{i=1}^{k} x_i y_i$
2. Multiplication of one polynomial vector and one polynomial: for any $\hat{x} \in R^k$, $y \in R$, $\hat{x}y = (x_1 y, \ldots, x_k y) \in R^k$

- Private key: randomly choose $s \xleftarrow{\$} R$, the length of s is $n\log_2 q$ bits.
- Public key: randomly choose a k-dim vector $\hat{a} \xleftarrow{\$} R^k$, choose error vector $\hat{e} \leftarrow \chi_{R,\alpha}^k$, here $\chi_{R,\alpha}^k$ obeys discrete Gaussian distribution on R^k, with expectation 0 and standard deviation $\alpha \leq 1/t\left(\sqrt{nk}\lceil r/2\rceil + 1\right)$. Computing a vector $\hat{b} = \hat{a}s + \hat{e} \in R^k$, and the public key is $(\hat{a}, \hat{b})$. To decrease key length, we could let all of the users share the same $\hat{a}$. So the length of public key is $kn \log_2 q$ bits.
- Encryption: given a plaintext $m \in D_1 = \mathbb{Z}_2[x]/\langle x^n + 1\rangle$, randomly choose $\hat{r} \xleftarrow{\$} D_r^k$, compute a pair (c_0, c_1) as the ciphertext, here $c_0 = \hat{a} \otimes \hat{r} \in R$ and $c_1 = \hat{b} \otimes \hat{r} + m(q-1)/2 \in R$.
- Decryption: compute $c_1 - c_0 s = m(q-1)/2 + \hat{e} \otimes \hat{r} \approx m(q-1)/2$

Correctness of scheme 1 is shown in Rückert and Schneider (2010), and when $\alpha \leq 1/30\sqrt{nk}\lceil r/2\rceil$, the scheme can decrypt correctly.

4.2 A new scheme using canonical mapping

Basing on scheme 1, we use canonical mapping to construct a new scheme.

Scheme 2

- Parameters: Let q be a prime and $q \equiv 1 \mod 2n$, let ω be a root of $x^n + 1$ in $\mathbb{Z}_q$, and $(q\text{-}1)/2$ cannot be divided by ω. The error distribution $\chi_{R,\alpha}^k$ is discrete Gaussian distribution on R^k, with expectation 0 and standard deviation $\alpha \leq 1/t\left(\sqrt{nk}\lceil r/2\rceil + 1\right)$. Definition of D_r and polynomial vector operations are the same with scheme 1.
- Private key: $s \xleftarrow{\$} R$, s.t. $s(0)$ is not a divisor of $(q\text{-}1)/2$. The length of private key is $n\log_2 q$ bits.
- Public key: Randomly choose a k-dim polynomial vector $\hat{a} \xleftarrow{\$} R^k$. Choose error vector $\hat{e} \leftarrow \chi_{R,\alpha}^k$ and set $\hat{b} = \hat{a}s + \hat{e} \in R^k$. To reduce key length, we can let all of the users share the same $\hat{a}$, and the public key is $(\hat{a}, \hat{b})$ which has a length of $kn \log_2 q$ bits.
- Encryption: Encryption has three steps.

1. For any given n-bits plaintext $m \in D_1$, let $m = (m_0, m_1, \ldots, m_{n-1})$ and randomly choose $\hat{r} \xleftarrow{\$} D_r^k$
2. Compute $c_0 = \hat{b} \otimes \hat{r}$, $c_1 = \hat{a} \otimes \hat{r}$. Noticing that c_0, c_1 are two polynomials in R, we can use canonical mapping to change them into vectors in $\mathbb{Z}_q^n$, namely

$$c_0 \mapsto (c_0(\omega), c_0(\omega^3), \ldots, c_0(\omega^{2n-1})) = C_0$$
$$c_1 \mapsto (c_1(\omega), c_1(\omega^3), \ldots, c_1(\omega^{2n-1})) = C_1$$

3. Compute $C_2 = C_0 + \frac{q-1}{2}(m_0, \ldots m_{n-1})$, and output the ciphertext (C_1, C_2).

- Decryption: Also includes three steps.

1. Use the inverse of canonical mapping to change C_1 into a polynomial $c_1(x) = \hat{a} \otimes \hat{r}$;
2. Compute $c_1(x) \cdot s = \hat{a} \otimes \hat{r} \cdot s = \hat{b} \otimes \hat{r} - \hat{e} \otimes \hat{r} \approx c_0(x)$, and transform $c_1(x) \cdot s$ into a vector S;
3. Compute $(C_2 - S) \mod \omega \approx \frac{q-1}{2} m$

Theorem 1 *When the parameters satisfy the aforementioned requirement, Scheme 2 can decrypt correctly.*

Proof Consider the decryption process,

$$\begin{aligned} C_2 - S &= C_0 + \frac{q-1}{2} m - \sigma(c_1(x)s) \\ &= \left((\hat{b} \otimes \hat{r})(\omega), \ldots, (\hat{b} \otimes \hat{r})(\omega^{2n-1})\right) \\ &\quad - \left((\hat{a} \otimes \hat{r}s)(\omega), \ldots, (\hat{a} \otimes \hat{r}s)(\omega^{2n-1})\right) + \frac{q-1}{2} m \end{aligned}$$

We focus on the first item, and case of the other items is analogous. The first item of the above formula is

$$\begin{aligned} &(\hat{a} \otimes \hat{r}s)(\omega) + (\hat{e} \otimes \hat{r})(\omega) - (\hat{a} \otimes \hat{r}s)(\omega) + \frac{q-1}{2} m_0 \\ &\quad = (\hat{e} \otimes \hat{r})(\omega) + \frac{q-1}{2} m_0 \end{aligned}$$

where $(\hat{e} \otimes \hat{r})(\omega)$ is a polynomial about ω in R, and after a module operation, only the constant term remains. Let $\hat{e} = (e_1, \ldots, e_k)$, $\hat{r} = (r_1, \ldots, r_k)$, then $\hat{e} \otimes \hat{r} = \sum_{i=1}^{k} e_i r_i$.

Considering $\hat{e} \leftarrow \chi^k_{R,\alpha}$, on account of Chebyshev's law, for n independent samples that abiding the same Gaussian distribution $X_i \leftarrow N(\mu, \sigma^2)$, $1 \le i \le n$, their summation satisfies $\sum_{i=1}^{n} X_i \leftarrow N(n\mu, n\sigma^2)$, thus $\sum_{i=1}^{k} e_i r_i$ obeys a Normal distribution with expectation 0 and standard deviation $\sqrt{\sum_{i=1}^{k} (r\sqrt{n}\alpha/2)^2} = \sqrt{nk}r\alpha/2 \le \sqrt{nk}\lceil r/2 \rceil \alpha \le 1/t$. According to the truncated inequality of Normal distribution,we have $\Pr\left(\left[\sum_{i=1}^{k} e_i(0)\right] > q/4\right) = \frac{4}{t}\sqrt{\frac{2}{\pi}} e^{-\frac{t^2}{32}}$. When $t \ge 30$, this value can be ignored, so $\Pr([\sum_{i=1}^{k} e_i(0)] \le q/4) \approx 1$. Considering that ω is not a divisor of $(q\text{-}1)/2$, the first item of $(C_2 - S) \mod \omega$is not greater than $\frac{q}{4} + \frac{q-1}{2} m_0$. Thus completes the proof.

Theorem 2 *For any $\varepsilon > 0$ and $m \ge (1 + \varepsilon)(1 + n)$ - log q, if there exists a probabilistic polynomial-time algorithm that can attack the CPA security of scheme 2 with advantage ε, then there exist a poly-time distinguisher V that for any possible private key s, can distinguish distribution $\left\{(\hat{a}, \hat{a}s + \hat{e}) | \hat{a} \xleftarrow{\$} R^k, \hat{e} \leftarrow D_{R,\xi}, s \xleftarrow{\$} R\right\}$and uniform distribution U on $R^k \times R^k$, here $\xi = \alpha \cdot (nk/\log(nk))^{1/4}$.*

Proof We only discuss the first bit m_0 of a plaintext. Suppose there exists a CPA attacker A that can distinguish the ciphertext of $m_0 = 0$ and $m_0 = 1$ with advantage ε. We construct a distinguisher V which can distinguish the following two distributions with advantage at least ε/2: $\left\{(\hat{a}, \hat{a}s + \hat{e}) | \hat{a} \xleftarrow{\$} R^k, a_i(0) = 1, i = 1, \ldots, k, \hat{e} \leftarrow D_{R,\xi}, s \xleftarrow{\$} R, s(0) = 1\right\}$and Uniform distribution U on $R^k \times R^k$. The distinguisher V is constructed as the following:Input of V are two polynomial vectors $(\hat{a}, \hat{b})$in $R^k \times R^k$, satisfying that each constant term of $\hat{a}$is 1. Now V can invoke A to judge that whether $(\hat{a}, \hat{b})$obeys uniform distribution or is a RLWE vector. Using $(\hat{a}, \hat{b})$as private key, V invokes A, the latter generate two message bits m_0, m_1, and send them to V. V randomly choose $i \in \{0, 1\}$, encrypt m_i and send the ciphertext back to A. If A can guess the correct i and return it to V, then V outputs 1, else, outputs 0.Let the challenging ciphertext be (C_1, C_2), if σ is canonical mapping, then the first bit of C_1 and C_2 are $(\hat{a} \otimes \hat{r})(\omega)$and $(\hat{b} \otimes \hat{r})(\omega) + \frac{q-1}{2} m_0$respectively. If $\hat{b}$is chosen randomly and uniformly in R^k, and is independent of $\hat{a}$, then the first bit of the challenging ciphertext is also randomly and uniformly. In this case, the probability of "V outputs 1" is at most 1/2. On the other side, if $\hat{b} = \hat{a}s + \hat{e}$and the parameters are chosen according to the requirement, then by assumption, the probability of A correctly guessing i is $(1 + \varepsilon)/2$, so V can output 1 with the same probability. Thus completes the proof, namely, V can distinguish two distributions with advantage ε/2.

4.3 Homomorphic evaluations

Given two pairs of ciphertexts (C_1, C_2) and (C'_1, C'_2), where

$$C_1 = (c_1(\omega), c_1(\omega^3), \ldots, c_1(\omega^{2n-1}))$$

$$C_2 = \left(c_0(\omega) + \frac{q-1}{2} m_0, c_0(\omega^3) + \frac{q-1}{2} m_1, \ldots, c_0(\omega^{2n-1}) + \frac{q-1}{2} m_{n-1}\right)$$

$$C'_1 = (c'_1(\omega), c'_1(\omega^3), \ldots, c'_1(\omega^{2n-1}))$$

$$C'_2 = \left(c'_0(\omega) + \frac{q-1}{2} m'_0, c'_0(\omega^3) + \frac{q-1}{2} m'_1, \ldots, c'_0(\omega^{2n-1}) + \frac{q-1}{2} m'_{n-1}\right)$$

When computing the sum of two ciphertexts, we could simply add them coordinate-wise, and get $(C_{add1}, C_{add2}) = (C_1 + C'_1, C_2 + C'_2)$

Due to the use of canonical mapping, multiplication of two vectors could also done coordinate-wisely. Let "*" denote the coordinate-wise multiplication of vectors, then $(C_{mult1}, C_{mult2}) = (C_1 * C'_1, C_2 * C'_2)$

We focus on the decryption of the first item. Case of the other items is analogous.

The first item of $C_2 * C'_2$is $c_0(\omega)c'_0(\omega) + \frac{q-1}{2} m_0 c'_0(\omega) + \frac{q-1}{2} m'_0 c_0(\omega) + \frac{(q-1)^2}{4} m_0 m'_0$.

During the decryption process, we need to change $C_1 * C'_1$into a polynomial, multiply it with s^2 and then transform the result into a vector S_{mult}. The first item of S_{mult} is

$$s^2(\omega)c_1(\omega)c'_1(\omega) = \hat{a} \otimes \hat{r}(\omega)s(\omega) \cdot \hat{a} \otimes \hat{r}'(\omega)s(\omega) \quad (4-1)$$

Noticing that

$$c_0(\omega)c'_0(\omega) = [(\hat{a} \otimes \hat{r})(\omega)s(\omega) + (\hat{e} \otimes \hat{r})(\omega)] \times [(\hat{a} \otimes \hat{r}')(\omega)s(\omega) + (\hat{e} \otimes \hat{r}')(\omega)] \quad (4-2)$$

Subtract (4-2) by (4-1), we can get

$$(\hat{e} \otimes \hat{r})(\omega)(\hat{e} \otimes \hat{r}')(\omega) + (\hat{a} \otimes \hat{r})(\omega)s(\omega)(\hat{e} \otimes \hat{r}')(\omega) + (\hat{a} \otimes \hat{r}')(\omega)s(\omega)(\hat{e} \otimes \hat{r})(\omega) = D$$

The last decryption step in scheme 2 is to compute C_2-S, and after homomorphic multiplication, it needs to compute $C_2 * C'_2 - S_{mult}$. Then the first item is

$$D+\frac{q-1}{2}m_0c_0'(\omega)+\frac{q-1}{2}m_0'c_0(\omega)+\frac{(q-1)^2}{4}m_0m_0'$$

where $c_0(\omega)=(\hat{a}\otimes\hat{r})(\omega)s(\omega)+(\hat{e}\otimes\hat{r})(\omega)$, $c_0'(\omega)=(\hat{a}\otimes\hat{r}')(\omega)s(\omega)+(\hat{e}\otimes\hat{r}')(\omega)$. Noticing that besides the first item, all of the other items are multiples of ω, and recalling that ω is not a divisor of $(q\text{-}1)^2/4$, so we can divide the first item by ω, and get the residue:

$$\begin{aligned}&(\hat{e}\otimes\hat{r})(0)(\hat{e}\otimes\hat{r}')(0)+(\hat{a}\otimes\hat{r})(0)s(0)(\hat{e}\otimes\hat{r}')(0)\\&\quad+(\hat{a}\otimes\hat{r}')(0)s(0)(\hat{e}\otimes\hat{r})(0)+\frac{q-1}{2}m_0[(\hat{a}\otimes\hat{r})(0)s(0)\\&\quad+(\hat{e}\otimes\hat{r})(0)]+\frac{q-1}{2}m_0'[(\hat{a}\otimes\hat{r}')(0)s(0)+(\hat{e}\otimes\hat{r}')(0)]\\&\quad+\frac{(q-1)^2}{4}m_0m_0'\end{aligned}$$

Also noticing that $s(0)$ is not a divisor of $(q\text{-}1)/2$, dividing the above formula by s(0) and get the residue, the first item becomes

$$\begin{aligned}&(\hat{e}\otimes\hat{r})(0)(\hat{e}\otimes\hat{r}')(0)+\frac{q-1}{2}m_0(\hat{e}\otimes\hat{r})(0)\\&\quad+\frac{q-1}{2}m_0'(\hat{e}\otimes\hat{r}')(0)+\frac{(q-1)^2}{4}m_0m_0'\\&\quad=\Delta\end{aligned}$$

where $(\hat{e}\otimes\hat{r})(0)$and $(\hat{e}\otimes\hat{r}')(0)$are constant items of $\hat{e}\otimes\hat{r}$and $\hat{e}\otimes\hat{r}'$respectively.

According to the proof of theorem 4.1, $C_2 * C'_2$can be correctly decrypted and thus obtain the multiplication of two plaintexts.

4.4 Efficiency

The advantage of scheme 2 lies in a shorter key length and smaller computation cost, we give a detailed analysis below.

Key length: The length of public key is $kn\log_2 q$ bits. The private key is a polynomial in R with constant item 1, and the length of private key is $n\log_2 q$ bits.

Computation cost:

1. During encryption, the computing cost of polynomial convolution can be reduced through a Fast Fourier Transformation. To encrypt n bits, the total computation cost is $\tilde{O}(n\log n)$.
2. During decryption, it needs to compute the inverse of canonical mapping, then compute a polynomial multiplication and one canonical mapping, finally a vector subtraction. The total computation cost is $\tilde{O}(n\log n)$.
3. Homomorphic addition: The addition of two ciphertexts is simply vector addition, the computation cost is $\tilde{O}(n)$. After an addition, the length of ciphertext is not increased, and accordingly the computation cost of decryption remains the same.
4. Homomorphic multiplication: Multiplication of two ciphertexts only needs to directly compute vector multiplication on $\mathbb{Z}_q^n$coordinate-wise, the computing cost is $\tilde{O}(n\log n)$. After multiplication, the length of ciphertext increase to $4n\log_2 q$ bits, namely doubled. In decryption phase, for each ciphertext element, it needs to solve a linear equation set, then compute one polynomial multiplication and one subtraction, the total computation cost of decryption is $\tilde{O}(n^2)$.

To sum up, we confirm that comparing with scheme 1, scheme 2 has an obvious advantage in efficiency. The key length and computation cost is controlled in a rational bound. We believe that scheme 2 is a practical somewhat homomorphic encryption scheme.

5 Private information retrieval protocol basing on scheme 2

5.1 A PBR protocol

The most representative application of homomorphic encryption is to construct private information retrieval (PIR) protocols (Cachin et al. 1999). Using homomorphic encryption, communication complexity of PIR protocol can be reduced to *poly*(log n) bits, this is a great improvement. Kushilevitz and Ostrovsky (1997) first introduced homomorphic encryption into PIR protocols, their PIR protocol has sublinear time-complexity and exponential communication cost. In 2009, Gentry discussed (2009) how to implement PIR protocol using homomorphic encryption. In 2011, Brakerski and Vaikuntanathan (2011b) presented a generic framework through combining a FHE with a symmetric key encryption scheme. Most of the available PIR protocols refer to single bit retrieval, while in fact, a record in a database is often longer than one bit, thus arise a natural expansion of PIR, namely PBR (Private Block Retrieval) protocols.

We introduce a PBR protocol basing on scheme2. Considering a database that each record of which is more than one bit, we use multi-bit encryption scheme to encrypt index information, thus can reduce the number of ciphertext, and also reduce communication cost.

Suppose there are n records in a database, each has a length of d bits. The initial position of each record is represented by indexes, which has a length of log n bits. Let $SYM=(SYM.KeyGen, SYM.Enc, SYM.Dec)$ be a secure symmetric key encryption scheme, with plaintext space $\{0,1\}^{\log n}$, without lost of generality, assuming that the ciphertext space also be $\{0,1\}^{\log n}$. Let $SWHE=(SWHE.KeyGen, SWHE.Enc, SWHE.Dec, SWHE.Eval)$be a somewhat homomorphic encryption scheme on plaintext space $\{0,1\}^k$, where $k=poly(\log n)$.

Our PBR protocol is comprised of four algorithms:

$PBR = (Setup, Query, Response, Decode)$

The algorithms are defined as the following:

- $Setup(1^{\lambda})$: on inputting the security parameter λ, generate the symmetric key symsk $\leftarrow$ $SYM.Keygen(1^{\lambda})$ and keys of the SWHE scheme (hpk, hevk, hsk) $\leftarrow$ $SWHE.Keygen(1^{\lambda})$, then encrypt symsk with the public key, namely $C_{symsk} \leftarrow SWHE.Enc_{hpk(}$symsk).
  The setup stage output the public parameters Params: = (hpk,hevk,C_{symsk}), and private parameters Setupstate: = (hsk, symsk).
- Query(1^{λ}, setupstate, i): Suppose the i^{th} record is to be required, i$\in${1,…,n}, the user encrypts i by symsk, and generate the query string query, namely query $\leftarrow$ $SYM.Enc_{symsk}$(i).
- Response(1^{λ},DB,params,query): Upon receiving the query string query, database compute the query function h(C_{symsk}), and let resp $\leftarrow$ $SWHE.Eval_{hevk}$(h(C_{symsk})), thus can get the a ciphertext of DB[i]. Where the query function h(x) is defined as

  $$h(x) \stackrel{def}{=} DB[SYM.Dec(x, query)]$$

- Decode(1^{λ},setupstate, qstate, resp): the receiver decrypt resp, and obtain b $\leftarrow$ $SWHE.Dec_{hsk}$(resp)

5.2 Analysis

The above protocol can be implemented using a LWE-based symmetric encryption scheme combining with our multi-bit SWHE scheme. In this implementation, the index has a length of log n bits (here n present the number of records in a database), so the size of query information query is log n bits. According to scheme 2, the response information to a query has $2d$ log$q$ bits. So in the above PBR protocol, to retrieve d bits, the protocol has a communication complexity of $2d\log q + \log n$, communication cost of each bit is 2 log q + (log n)/d, which is a polynomial of the length of q and n. Such a communication complexity is fairly reasonable.

On the other hand, let's consider the computational cost of this protocol. Also according to the SWHE scheme, suppose the decryption algorithm has one multiplication, then to generate response information, the server has a computation cost of $\tilde{O}(n \log n)$, while in user end, computation cost of decryption is $\tilde{O}(n^2)$.

6 Conclusion

In this paper we provide a somewhat homomorphic multi-bit encryption scheme that is basing on RLWE assumption. We use canonical mapping in the process of encryption and homomorphic evaluations. Due to this technique and the comparatively compact RLWE assumption, the new scheme is time-efficient and the number of ciphertext elements will not increase after homomorphic evaluations. Using this scheme, an efficient PIR protocol can be constructed.

Homomorphic encryption scheme is a new hot area in cryptography. There has been abundant works in recent years focusing on scheme construction and application, and new methods and new ideas have appeared continuously. However there still leaves a lot of problems to solve, both in theoretical and practical.

Aiming on performance improvement, we use a new technique to construct scheme, and our scheme is practical due to its computation cost and key length, while because homomorphic multiplication can cause an increase in ciphertext length, the scheme is somewhat but not fully homomorphic. Further studies on controlling ciphertext length and ultimately constructing fully homomorphic encryption schemes will be our target in the future.

Acknowledgments This work was financially supported by the National Natural Science Foundation of China (Grant No. 61272492, 61103230, 61103231).

References

Bogdanov A, Lee C (2011) Homomorphic encryption from codes (2011). Arxiv preprint arXiv:1111.4301. 1, 9, 14

Boneh D, Goh EJ, Nissim K (2005) Evaluating 2-DNF formulas on ciphertexts. TCC 2005, LNCS 3378, pp 325–341

Brakerski Z, Vaikuntanathany V (2011a) Efficient fully homomorphic encryption from (Standard) LWE. In: *Electronic Colloquium on Computational Complexity ECCC,* vol. 18, pp 109–138

Brakerski Z, Vaikuntanathan V (2011b) Fully homomorphic encryption from ring-LWE and security for key dependent messages. Advances in Cryptology-CRYPTO2011, pp 505–524 (1, 9, 13)

Brakerski Z, Gentry C, Vaikuntanathan V (2012) Fully homomorphic encryption without bootstrapping. ITCS, See also http://eprint.iacr.org/2011/277

Cachin C, Micali S, Stadler M (1999) Computationally private information retrieval with polylogarithmic communication. EUROCRYPT 1999:402–444

Chen X, Li J, Ma J, Tang Q, Lou W (2012a) New algorithms for secure outsourcing of modular exponentiations. ESORICS 2012, LNCS 7459, Springer, pp 541–556

Chen X, Li J, Susilo W (2012b) Efficient fair conditional payments for outsourcing computations. IEEE Trans Inf Forensics Secur 7(6):1687–1694

Dijk M, Gentry C, Halevi S, Vaikuntanathan V (2010) Fully homomorphic encryption over the integers. Advances in Cryptology-EUROCRYPT, pp 24–43

Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: Proceedings of STOC, pp 169–178

Gentry C (2009) A fully homomorphic encryption scheme. PhD thesis, Stanford University

Gentry C, Halevi S, Vaikuntanathan V (2010) A simple BGN-type cryptosystem from LEW. In Proceedings of EUROCRYPT'10, LNCS vol 6110. Springer, Heidelberg, pp 506–522

Kushilevitz E, Ostrovsky R (1997) Replicationis not needed: single data base, computationally-private information retrieval. In: FOCS, pp 364–373

Lyubashevsky V, Peikert C, Regev O (2010) On ideal lattices and learning with errors over rings. Eurocryt, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp 1–23

Regev O (2009) On lattices, learning with errors, random linear codes, and cryptography. J ACM 56(6):34, Preliminary version in STOC'05

Rückert M, Schneider M (2010) Estimating the security of lattice-based cryptosystems. http://eprint.icur.org/2010/137.pdf

Smart NP, Vercauteren F (2010) Fully homomorphic encryption with relatively small key and ciphertext sizes. PKC 2010, LNCS 6056, pp 420–443

Stehle D, Steinfeld R (2010) Faster fully homomorphic encryption. ASIACRYPT LNCS 6477, pp 377–394