DE GRUYTER
OPEN

TATRA
MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2016-0031
Tatra Mt. Math. Publ. **67** (2016), 69–83

EFFICIENT SECURE MATRIX MULTIPLICATION OVER LWE-BASED HOMOMORPHIC ENCRYPTION

Dung Hoang Duong — Pradeep Kumar Mishra — Masaya Yasuda

ABSTRACT. Homomorphic encryption enables various calculations while preserving the data confidentiality. In this paper, we apply the somewhat homomorphic encryption scheme proposed by Brakerski and Vaikuntanathan (CRYPTO 2011) to secure matrix multiplication between two matrices. To reduce both the ciphertext size and the computation cost, we propose a new method to pack a matrix into a single ciphertexts so that it also enables efficient matrix multiplication over the packed ciphertexts. Our packing method generalizes Yasuda et al.'s methods (Security Comm. Networks 2015 and ACISP 2015), which are for secure inner product. We also implement our methods and give a comparison with previous packing methods.

1. Introduction

Homomorphic encryption is a form of encryption that can support meaningful operations on encrypted data (without decryption). This encryption has been expected to give a powerful tool for data protection in cloud computing. The concept of homomorphic encryption was first introduced by Rivest et al. in [7, 1978], and the first fully homomorphic encryption (FHE) scheme that supports arbitrary computations on encrypted data was constructed by Gentry [4]. However, currently known FHE schemes are yet impractical. In contrast, somewhat homomorphic encryption (SHE) schemes (e.g., the Boneh-Goh-Nissim (BGN) scheme [1], and the building block for an FHE scheme), which support only a limited number of additions and multiplications on encrypted data, have wider applications and take a lot of attention from various communities.

2010 Mathematics Subject Classification: 94A60.
Keywords: somewhat homomorphic encryption, packing methods, secure inner product, secure matrix multiplication.
This work was supported by JSPS KAKENHI Grant Numbers 16K17644 and 16H02830. This work was also supported by CREST, JST.

We use the SHE scheme proposed by Brakerski and Vaikuntanathan [3]. The security of the scheme relies on the computational hardness of the polynomial version of the learning with errors (LWE) problem, a simplified version of the ring-LWE assumption of [6]. Over the SHE scheme, Lauter et al. [5] proposed a method to pack an integer of large size into a single ciphertext so that it enables to efficiently compute secure sums and products over the integers. After that, Yasuda et al. [9] proposed a new packing method for secure multiple inner products, which can be applied to secure Hamming distance and pattern matching computations. While their packing method is efficient only for vectors with small size entries, they modified their method for large size entries [10]. Their modified method can be applied to secure statistical analysis such as covariance and correlation between two variables with integers of practical size.

In this paper, we propose several packing methods for secure matrix multiplication, using the idea of Yasuda et al.'s methods [9], [10]. More specifically, for matrices with binary entries, our method is based on [9], and for non-binary entries, it is based on [10]. Our main ingredient for packing a matrix $\mathbf{A}$ is to deploy the entries of $\mathbf{A}$ into a single polynomial by using the packing methods of [9], [10], and then encrypt the polynomial over the SHE scheme. Our matrix packing method requires only one homomorphic multiplication over our packed ciphertexts for secure matrix multiplication. In Section 2, we review the SHE scheme proposed by Brakerski and Vaikuntanathan. In Section 3, we present previous packing method over the SHE scheme. In Section 4, we present our packing methods for secure matrix multiplication. In Section 5, we give implementation results and discuss about a comparison with previous methods.

2. Preliminaries

In this section, we briefly recall the construction and the homomorphic correctness of the somewhat homomorphic encryption scheme proposed by Brakerski and Vaikuntanathan [3]. Let n be a 2-power integer defining the base ring $R = \mathbb{Z}[x]/(x^n + 1)$. Let q be a prime number with $q \equiv 1 \bmod 2n$ defining the ring $R_q = R/qR = \mathbb{F}_q[x]/(x^n + 1)$, which gives the base ring of ciphertext space. According to [2], the condition $q \equiv 1 \bmod 2n$ is not necessary for the construction and the security but for the efficiency of the scheme. Let t be a positive integer with $t < q$ defining the plaintext space $R_t = R/tR$. Let σ be the parameter for defining a discrete Gaussian error distribution $\chi = D_{\mathbb{Z},\sigma}$ (specifically, we select each entry in an n-dimensional vector by sampling from a Gaussian distribution $N(0, \sigma)$ and then round it to the nearest integer).

The security of the scheme (constructed in Section 2.1 below) relies on the following polynomial-LWE assumption, which is a simplified version of the ring-LWE assumption of Lyubashevsky, Peikert and Regev [6] (the following assumption is independent of the parameter t, see [3, Proposition 11] for details).

Definition 1 (Polynomial-LWE)**.** Given (n, q, t, σ), the polynomial-LWE assumption is that it is infeasible to distinguish the following two distributions:

(1) One samples (a_i, b_i) uniformly from $(R_q)^2$.
(2) One first draws $s \leftarrow \chi = D_{\mathbb{Z}^n,\sigma}$ uniformly and then samples $(a_i, b_i) \in (R_q)^2$ by sampling $a_i \leftarrow R_q$ uniformly, $e_i \leftarrow \chi$ and setting $b_i = a_i s + e_i$.

2.1. Somewhat homomorphic encryption scheme

We here present the construction of the public-key SHE scheme. Specifically, the below construction is a variant of [5, Section 3.2].

- **Key generation:** Choose an element $R \ni s \leftarrow \chi$, sample a uniformly random element $p_1 \in R_q$ and an error $R \ni e \leftarrow \chi$. Set $\mathrm{pk} = (p_0, p_1)$ with $p_0 = -(p_1 s + te)$ as the public key and $\mathrm{sk} = s$ as the secret key.
- **Encryption:** For a plaintext $m \in R_t$, sample $R \ni u, f, g \leftarrow \chi$ and compute the fresh ciphertext

$$\mathrm{Enc}(m, \mathrm{pk}) = (c_0, c_1) = (p_0 u + tg + m, p_1 u + tf) \in (R_q)^2, \tag{1}$$

  where the plaintext $m \in R_t$ is regarded as an element of R_q, since $t < q$.
- **Homomorphic operations:** Given two (fresh or operated) ciphertexts $\mathrm{ct} = (c_0, c_1, \ldots, c_\xi)$ and $\mathrm{ct}' = (c'_0, c'_1, \ldots, c'_\eta)$ (note that the length of a ciphertext increases by the homomorphic multiplication defined below). Then the homomorphic addition "$\dot{+}$" is computed by component-wise addition

$$\mathrm{ct} \dot{+} \mathrm{ct}' = \left(c_0 + c'_0, \ldots, c_{\max(\xi,\eta)} + c'_{\max(\xi,\eta)}\right)$$

  by padding with zeros if $\xi \neq \eta$, and the homomorphic multiplication "$*$" is defined by $\mathrm{ct} * \mathrm{ct}' = (\tilde{c_0}, \ldots, \tilde{c}_{\xi+\eta})$ with

$$\sum_{i=0}^{\xi+\eta} \tilde{c_i} z^i = \left(\sum_{i=0}^{\xi} c_i z^i\right)\left(\sum_{j=0}^{\eta} c'_j z^j\right).$$

  Here z denotes a symbolic variable.
- **Decryption:** For a ciphertext $\mathrm{ct} = (c_0, c_1, \ldots, c_\xi)$, the decryption with secret key $\mathrm{sk} = s$ is computed by

$$\mathrm{Dec}(\mathrm{ct}, \mathrm{sk}) = [\tilde{m}]_q \bmod t \in R_t,$$

where $\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$ and $[z]_q$ denotes the reduction of z under modulo q in the interval $[-q/2, q/2)$. For the secret key vector $\mathbf{s} = (1, s, s^2, \ldots)$, we can simply rewrite $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk}) = [\langle \mathrm{ct}, \mathbf{s} \rangle]_q \mod t$.

2.2. Homomorphic correctness

The "homomorphic correctness" means that the decryption procedure can correctly recover the operated result over plaintexts after performing several homomorphic operations over ciphertexts. For given ciphertexts ct_1 and ct_2 corresponding to the plaintexts m_1 and m_2, one has from the construction that

$$\begin{cases} \langle \mathrm{ct}_1 \dot{+} \mathrm{ct}_2, \mathbf{s} \rangle &= \langle \mathrm{ct}_1, \mathbf{s} \rangle + \langle \mathrm{ct}_2, \mathbf{s} \rangle, \\ \langle \mathrm{ct}_1 * \mathrm{ct}_2, \mathbf{s} \rangle &= \langle \mathrm{ct}_1, \mathbf{s} \rangle \cdot \langle \mathrm{ct}_2, \mathbf{s} \rangle. \end{cases}$$

It follows from [5, Theorem 3.3] that homomorphic operations over ciphertexts correspond to the ring structure of the plaintext space R_t. Specially we have

$$\begin{cases} \mathrm{Dec}(\mathrm{ct}_1 \dot{+} \mathrm{ct}_2, \mathrm{sk}) &= m_1 + m_2 \in R_t, \\ \mathrm{Dec}(\mathrm{ct}_1 * \mathrm{ct}_2, \mathrm{sk}) &= m_1 \times m_2 \in R_t. \end{cases}$$

Remark 2. Given a fresh ciphertext $\mathrm{ct} = (c_0, c_1)$ generated by (1), we have

$$\begin{aligned} \langle \mathrm{ct}, \mathbf{s} \rangle &= (p_0 u + tg + m) + s \cdot (p_1 u + tf) \\ &= m + t \cdot (g + sf - ue) \end{aligned} \tag{2}$$

in the ring R_q since $p_0 + p_1 s = -te$. If the value $m + t \cdot (g + sf - ue)$ does not wrap around mod q (all errors $e, f, g, u \leftarrow \chi$ must be sufficiently small), we have $[\langle \mathrm{ct}, \mathbf{s} \rangle]_q = m + t \cdot (g + sf - ue)$ in the base ring R. Hence we can recover the correct plaintext m by (mod t)-operation, which shows the decryption mechanism for any fresh ciphertexts (see [3] for more details on the homomorphic correctness).

The following Lemma gives us the condition for the homomorphic correctness of the SHE scheme and for choosing suitable parameters (n, q, t, σ) in order to avoid decryption failure of a ciphertext.

Lemma 3 (Condition for successful decryption)**.** *For a ciphertext* ct*, the decryption* $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$ *recovers the correct result if it satisfies the condition*

$$\| \langle \mathrm{ct}, \boldsymbol{s} \rangle \|_\infty < \frac{q}{2}.$$

Here for $a = \sum_{i=0}^{n-1} a_i x^i \in R$*, let* $\|a\|_\infty = \max |a_i|$ *denote the* ∞*-norm of its coefficient representation.*

3. Previous packing methods

In this section, we briefly review the previous packing methods proposed by Lauter et al. [5] and Yasuda et al. [9], [10] over the SHE scheme described in the previous section.

3.1. Packing method by [5] for large integers

Lauter et al. [5] introduced a method to pack a large integer into a single ciphertext, and it enables efficient computation of sums and products over packed ciphertexts. Specially, a message M of up to n bits is broken into a binary vector $(m_0, \ldots, m_{n-1})$ and associated with a polynomial

$$\mathrm{pm}(M) = \sum_{i=0}^{n-1} m_i x^i,$$

of degree less than or equal to $n-1$ and finally encrypts M as $\mathrm{ct}(M) := \mathrm{Enc}(\mathrm{pm}(M), \mathrm{pk})$. Note that $\mathrm{pm}(M)|_{x=2} = M$. Moreover, for given plaintexts M, M', the homomorphic addition $\mathrm{ct}(M) \dot{+} \mathrm{ct}(M')$ gives the polynomial addition $\mathrm{pm}(M) + \mathrm{pm}(M')$ on encrypted data under the correctness [5, Lemma 3.3]. However, the integer multiplication causes a problem as polynomial multiplication $\mathrm{pm}(M) \cdot \mathrm{pm}(M')$ has degree larger than n. Hence, their method can only encodes integers of at most (n/d)-bit when it requires d homomorphic multiplications. Their method is effective in computing low degree multiplications.

3.2. Packing method by [9] for secure inner product

Yasuda et al. [9] proposed two types of packed ciphertexts as follows: For an integral vector $A = (a_0, \ldots, a_{m-1})$ of length $m \leq n$, set

$$\mathrm{pm}^{(1)}(A) = \sum_{i=0}^{m-1} a_i x^i \quad \text{and} \quad \mathrm{pm}^{(2)}(A) = -\sum_{i=0}^{m-1} a_i x^{n-i}, \tag{3}$$

and then packed ciphertexts are defined as $\mathrm{ct}^{(i)}(A) := \mathrm{Enc}(\mathrm{pm}^{(i)}(A), \mathrm{pk})$ for $i = 1, 2$. The first type is same as Lauter et al.'s method [5], but the second one enables efficient secure computation of inner product. Specially, for two vectors A and B of same length $m \leq n$, only one homomorphic multiplication over $\mathrm{ct}^{(1)}(A)$ and $\mathrm{ct}^{(2)}(B)$ can give the inner product $\langle A, B \rangle$ on encrypted data; see [9, Proposition 1]. This computation is effective for secure distance computations such as the Euclidean and the Hamming distances; see [9] for more details.

3.3. Modification by [10] for large integer entries

When handling with small integers (e.g., 1 or 2 bits), the packing method in [9] is effective and sufficient for secure statistics. However, for integers of practical bit-size (e.g., even for 16 or 32 bit), the packing method enforces to set the parameter t of the plaintext space R_t to be considerably large, and it then causes slow performance of the SHE scheme. Yasuda et al. [10] modified the packing method in [9] for large-size integers as follows.

Given an integer m, let $A = (a_0, \ldots, a_{m-1})$ be a vector of length m with entries a_i less than p bits. For a chosen integer $r > 0$ (e.g., $r = 2$), we write each integral entry a_i in the base-r representation, namely

$$a_i = \sum_{u=0}^{d-1} a_{i,u} r^u \quad \text{with} \quad a_{i,u} \in \{0, 1, \ldots, r-1\},$$

where $d = \lceil \log_r 2^p \rceil$. We associate to A the following two polynomials:

$$\mathrm{pm}^{(1)}_{m,p,r}(A) = \sum_{i=0}^{m-1} \left(\sum_{u=0}^{d-1} a_{i,u} x^u \right) x^{2id}, \tag{4}$$

$$\mathrm{pm}^{(2)}_{m,p,r}(A) = -\sum_{i=0}^{m-1} \left(\sum_{u=0}^{d-1} a_{i,u} x^u \right) x^{n-2id}, \tag{5}$$

and then packed ciphertexts are defined as $\mathrm{ct}^{(i)}_{m,p,r}(A) := \mathrm{Enc}(\mathrm{pm}^{(i)}_{m,p,r}(A), \mathrm{pk})$ for $i = 1, 2$. Then we have the following theorem [10, Theorem 1] on secure inner product between A and B with large entries.

Theorem 4. *Let $A = (a_0, \ldots, a_{m-1})$ and $B = (b_0, \ldots, b_{m-1})$ be two integral vectors of same length m with entries less than p bits. Assume that $n \geq 2md$ and $t > m(r-1)^2 d$, where $d = \lceil \log_r 2^p \rceil$ for a fixed positive integer r. Let*

$$\mathrm{ct} = \mathrm{ct}^{(1)}_{m,p,r}(A) * \mathrm{ct}^{(2)}_{m,p,r}(B),$$

and let $\mathrm{Dec}(\mathrm{ct}, \mathrm{pk}) = \sum_{i=0}^{n-1} m_i x^i \in R_t$ denote the decryption result. Then under the condition of Lemma 3 for the ciphertext ct, the sum $\sum_{i=0}^{2d-1} m_i r^i \in \mathbb{Z}$ gives the inner product $\langle A, B \rangle$.

4. Our packing methods for secure matrix multiplication

In this section, we propose packing methods to efficiently compute secure matrix multiplication. Notice that by using the packing methods of Yasuda et al. [9], [10] described in the previous section, one needs m^2 secure inner product computations (i.e., m^2 homomorphic multiplications) to compute a matrix

multiplication of two $m \times m$ matrices. In this section, for each of the packing method proposed by Yasuda et al., we propose two methods to reduce the number of computations: the first one requires m *homomorphic multiplications* and the second one requires *only one homomorphic multiplication*.

4.1. Binary matrix multiplication

Let $\mathbf{A}$ be an $m \times m$ matrix with binary entries. Let $A_1, \ldots, A_m$ denote the row vectors of $\mathbf{A}$ and $A_1^T, \ldots, A_m^T$ the column vectors of $\mathbf{A}$. In order to compute the matrix multiplication $\mathbf{AB}$ of two binary matrices $\mathbf{A}$ and $\mathbf{B}$, one needs to compute the inner products $\langle A_i, B_j^T \rangle$ for $i, j = 1, \ldots, m$. We pack each row $A_i = (a_{i,0}, \ldots, a_{i,m-1})$ and each column $A_j^T = (a_{j,0}, \ldots, a_{j,m-1})$ of $\mathbf{A}$ as in (3), namely:

$$\mathrm{pm}^{(1)}(A_i) := \sum_{u=0}^{m-1} a_{i,u} x^u, \quad \mathrm{pm}^{(2)}(A_j^T) := -\sum_{v=0}^{m-1} a_{j,v} x^{n-v}. \tag{6}$$

It follows from Section 3.2 that the constant term of $\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T)$ is the inner product $\langle A_i, B_j^T \rangle$ (note that "$\times$" denotes the multiplication in the ring $R = \mathbb{Z}[x]/(x^n + 1)$). We define the following two types of polynomials in R associated to a given matrix $\mathbf{A}$:

$$\begin{aligned}
\mathbf{Pol}^{(1)}(\mathbf{A}) &:= \mathrm{pm}^{(1)}(A_1) + \cdots + \mathrm{pm}^{(1)}(A_m) x^{m(m-1)} \\
&= \sum_{i=1}^{m} \mathrm{pm}^{(1)}(A_i) x^{(i-1)m}, \\
\mathbf{Pol}^{(2)}(\mathbf{A}) &:= \mathrm{pm}^{(2)}(A_1^T) + \cdots + \mathrm{pm}^{(2)}(A_m^T) x^{m^2(m-1)} \\
&= \sum_{j=1}^{m} \mathrm{pm}^{(2)}(A_j^T) x^{(j-1)m^2}.
\end{aligned}$$

Define the packed ciphertexts for a given matrix $\mathbf{A}$ to be

$$\mathrm{ct}_{\mathrm{mat}}{}^{(i)}(\mathbf{A}) := \mathrm{Enc}(\mathbf{Pol}^{(i)}(\mathbf{A}), \mathrm{pk}), \quad \text{for} \quad i = 1, 2.$$

In order to get the matrix multiplication $\mathbf{AB}$, we need to obtain the inner products $\langle A_i, B_j^T \rangle$ for $i, j = 1, \ldots, m$. Here we present our two approaches.

Theorem 5 (The first packing method). *Assume that $n \geq m^2$. For each $j = 1, \ldots, m$, let*

$$\mathrm{ct}_j = \mathrm{ct}_{\mathrm{mat}}^{(1)}(\boldsymbol{A}) * \mathrm{ct}^{(2)}(B_j^T)$$

and let $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk}) \in R_t$ denote the decryption result. Then under the condition of Lemma 3 for the ciphertext ct_j, for each $j = 1, \ldots, m$, the inner product $\langle A_i, B_j^T \rangle$ is the coefficient of $x^{(i-1)m}$ in $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk})$.

P r o o f. It follows from Section 3.2 (and clear from definition) that

$$\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) = \langle A_i, B_j^T \rangle + \text{other terms of degree } (n - v + u),$$

with $u \neq v$ and $u, v \in \{0, \ldots, m-1\}$, and so the constant term of $\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) \in R$ gives us the inner product of A_i and B_j^T. Now one has that

$$\begin{aligned}\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk}) &= \mathbf{Pol}^{(1)}(\mathbf{A}) \times \mathrm{pm}^{(2)}\left(B_j^T\right) \\ &= \sum_{i=1}^{m} \mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) x^{(i-1)m}.\end{aligned}$$

For a fixed index i, we have

$$\begin{aligned}\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}\left(B_j^T\right) x^{(i-1)m} = &\langle A_i, B_j^T \rangle x^{(i-1)m} \\ &+ \text{other terms of degree } n - v + u + (i-1)m,\end{aligned}$$

with $u \neq v$ and $u, v \in \{0, \ldots, m-1\}$. Since the exponents of x is modulo n and therefore $n - v + u + (k-1)m$ is never equal to $(i-1)m$ for $u \neq v$ and $i, k \in \{1, \ldots, m\}$. This implies that the terms of degree $(i-1)m$ in $\mathbf{Pol}^{(1)}(\mathbf{A}) \times \mathrm{pm}^{(2)}(B_j^T)$ is exactly the term of degree $(i-1)m$ in $\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) x^{(i-1)m}$. Hence the inner product $\langle A_i, B_j^T \rangle$ is the coefficient of $x^{(i-1)m}$ in $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk})$. □

Remark 6. With this approach, after m homomorphic multiplications over the SHE scheme, we get the resulting matrix multiplication $\mathbf{AB}$.

Theorem 7 (The second packing method)**.** *Assume that $n \geq m^3$. For each $j = 1, \ldots, m$, let*

$$\mathrm{ct} = \mathrm{ct}_{\mathrm{mat}}^{(1)}(\mathbf{A}) * \mathrm{ct}_{\mathrm{mat}}^{(2)}(\mathbf{B})$$

and let $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk}) \in R_t$ *denote the decryption result. Then under the condition of Lemma 3 for the ciphertext* ct*, for each i and j, the inner product $\langle A_i, B_j^T \rangle$ is the coefficient of $x^{(j-1)m^2+(i-1)m}$ in* $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$.

P r o o f. The proof is similar to that of Theorem 5. Over the ring R, we have

$$\begin{aligned}\mathrm{Dec}(\mathrm{ct}, \mathrm{sk}) &= \mathbf{Pol}^{(1)}(\mathbf{A}) \times \mathbf{Pol}^{(2)}(\mathbf{B}) \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} \mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) x^{(j-1)m^2+(i-1)m}.\end{aligned}$$

As in the proof of Theorem 5, the term of degree $(j-1)m^2 + (i-1)m$ in $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$ is exactly the term of degree $(j-1)m^2 + (i-1)m$ in $\mathrm{pm}^{(1)}(A_i) \times \mathrm{pm}^{(2)}(B_j^T) x^{(j-1)m^2+(i-1)m} \in R$. Hence the coefficient of $x^{(j-1)m^2+(i-1)m}$ in $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$ gives the inner product $\langle A_i, B_j^T \rangle$. □

Remark 8. In this approach, we need to do only one homomorphic multiplication on packed ciphertexts for secure matrix multiplication.

4.2. Non-binary matrix multiplication

In this subsection, we adapt the method in Y a s u d a et al. [10] described in Section 3.3 to propose packing methods for multiplying integral matrices with non-binary entries. Let $\mathbf{A}$ and $\mathbf{B}$ be two $m \times m$ matrices whose entries are integers of less than p bits. Let $A^{(1)}, \ldots, A^{(m)}$ and $B^{(1)}, \ldots, B^{(m)}$ be the rows of $\mathbf{A}$ and $\mathbf{B}^T$, respectively. For $i = 0, \ldots, m-1$, we write $A^{(i)} = (a_0^{(i)}, \ldots, a_{m-1}^{(i)})$ and $B^{(i)} = (b_0^{(i)}, \ldots, b_{m-1}^{(i)})$. For a chosen integer $r > 0$, we write each integral entry $a_k^{(i)}$ in the base-r representation, namely

$$a_k^{(i)} = \sum_{u=0}^{d-1} a_{ku}^{(i)} r^u \quad \text{with} \quad a_{ku}^{(i)} \in \{0, 1, \ldots, r-1\},$$

where $d = \lceil \log_r 2^p \rceil$, as in Section 3.3 (we took $r = 2$ in our experiments, and then $d = p$). We pack $a_k^{(i)}$ as

$$a_k^{(i)}(x) := \sum_{u=0}^{d-1} a_{ku}^{(i)} x^u \in R = \mathbb{Z}[x]/(x^n + 1).$$

We associate to each row $A^{(i)}$ and column $B^{(j)}$ of $\mathbf{A}$ and $\mathbf{B}$, respectively the following polynomials in the ring R:

$$\mathrm{pm}^{(1)}_{m,p,r}(A^{(i)}) = \sum_{k=0}^{m-1} a_k^{(i)}(x) x^{2kd}, \tag{7}$$

$$\mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) = -\sum_{l=0}^{m-1} b_l^{(j)}(x) x^{n-2ld}, \tag{8}$$

We define the following polynomials in R associated to $\mathbf{A}$ and $\mathbf{B}$:

$$\begin{aligned}
\mathbf{Pol}^{(1)}(\mathbf{A}) &:= \mathrm{pm}^{(1)}_{m,p,r}(A^{(1)}) + \cdots + \mathrm{pm}^{(1)}_{m,p,r}(A^{(m)}) x^{(m-1)2md} \\
&= \sum_{i=1}^{m} \mathrm{pm}^{(1)}_{m,p,r}(A^{(i)}) x^{(i-1)2md}, \\
\mathbf{Pol}^{(2)}(\mathbf{B}) &:= \mathrm{pm}^{(2)}_{m,p,r}(B^{(1)}) + \cdots + \mathrm{pm}^{(2)}_{m,p,r}(B^{(m)}) x^{(m-1)2m^2 d} \\
&= \sum_{j=1}^{m} \mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) x^{(j-1)2m^2 d}.
\end{aligned}$$

Define the packed ciphertexts for a given matrix $\mathbf{A}$ to be

$$\mathrm{ct}_{\mathrm{mat}}{}^{(i)}(\mathbf{A}) := \mathrm{Enc}(\mathbf{Pol}^{(i)}(\mathbf{A}), \mathrm{pk}), \quad \text{for } i = 1, 2\,.$$

As in the case of binary matrix multiplication, we here present two approaches.

Theorem 9 (The first packing method). *Assume that $n \geq 2md(m+1)$. For each $j = 1, \ldots, m$, let*

$$\mathrm{ct}_j = \mathrm{ct}_{\mathrm{mat}}(\mathbf{A}) * \mathrm{ct}^{(2)}_{m,p,r}(B^{(j)}),$$

and let $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk}) \in R_t$ denote the decryption result. Then under the condition of Lemma 3 for the ciphertext ct_j, for each $j = 1, \ldots, m$, the inner product $\langle A^{(i)}, B^{(j)} \rangle$ is the sum of the terms of degree greater or equal to $(i-1)2md$ and less than $(i-1)2md + 2d$ in $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk})$ evaluated at $x = r$.

P r o o f. Fix i and j. Over the ring R, we have that

$$\begin{aligned}\mathrm{pm}^{(1)}_{m,p,r}(A^{(i)}) \times \mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) = &\underbrace{\sum_{k=0}^{m-1} a_k^{(i)}(x) b_k^{(j)}(x)}_{I(i)} \\ &+ \underbrace{\left(- \sum_{k=0}^{m-1} \sum_{l=0}^{k-1} a_k^{(i)}(x) b_l^{(j)}(x) x^{n+(k-l)2d} \right)}_{II(i)} \\ &+ \underbrace{\left(- \sum_{k=0}^{m-1} \sum_{l=k+1}^{m-1} a_k^{(i)}(x) b_l^{(j)} x^{n+(k-l)2d} \right)}_{III(i)}.\end{aligned}$$

One gets that the degrees of terms in $I(i), II(i)$ and $III(i)$ are in the intervals $[0, 2d-2]$, $[2d, 2md-2]$ and $[n-(m-1)2d, n-2]$, respectively. It follows that the sum of terms of degree less than $2d$ in $\mathrm{pm}^{(1)}_{m,p,r}(A^{(i)}) \times \mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) \in R$ evaluated at $x = r$ gives us the inner product $\langle A^{(i)}, B^{(j)} \rangle$. Now for each $j = 1, \ldots, m$

$$\begin{aligned}\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk}) &= \mathbf{Pol}^{(1)}(\mathbf{A}) \times \mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) \\ &= \sum_{i=1}^{m} \mathrm{pm}^{(1)}_{m,p,r}(A^{(i)}) \times \mathrm{pm}^{(2)}_{m,p,r}(B^{(j)}) x^{(i-1)2md}.\end{aligned}$$

It then follows that for every $i = 1, \ldots, m$ the inner product $\langle A^{(i)}, B^{(j)} \rangle$ is the sum of terms of degree greater than or equal to $(i-1)2md$ and less than $(i-1)2md + 2d$ in $\mathrm{Dec}(\mathrm{ct}_j, \mathrm{sk})$ evaluated at $x = r$. □

Theorem 10 (The second packing method). *Assume that $n \geq 2m^3d + 2md + 2d$. Let*

$$\mathrm{ct} = \mathrm{ct}^{(1)}_{\mathrm{mat}}(\mathbf{A}) * \mathrm{ct}^{(2)}_{\mathrm{mat}}(\mathbf{B})$$

and let $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk}) \in R_t$ *denote the decryption result. Then under the condition of Lemma 3 for the ciphertext* ct*, for each* i *and* j*, the inner product* $\langle A^{(i)}, B^{(j)} \rangle$ *is the sum of terms of degree greater than or equal to* $(i-1)2md + (j-1)2m^2d$ *and less than* $(i-1)2md + (j-1)2m^2d + 2d$ *in* $\mathrm{Dec}(\mathrm{ct}, \mathrm{sk})$ *evaluated at* $x = r$*.*

P r o o f. The proof follows easily from that of Theorems 7 and 9. □

5. Implementation and comparison

In this section, we report our implementation results. Specifically, we implemented previous methods for inner product and our methods for secure matrix multiplication. In the next subsection, let us describe how to select parameters of the SHE scheme for each method.

5.1. Selection of parameters

Our selection of parameters is followed from [5, Section 3.2.2]. In this subsection, we describe how to select parameters (n, q, t, σ) of the SHE scheme. We divide our description into two cases where the entries of matrices $\mathbf{A}$ and $\mathbf{B}$ are binary and non-binary.

5.1.1. Case of binary matrix multiplication

Here we assume that the entries of two $m \times m$ matrices $\mathbf{A}$ and $\mathbf{B}$ are binary. As in [5], we fix $\sigma = 8$. For the matrix size m, it is sufficient to take the plaintext modulus as $t = m + 1$ since every entry of $\mathbf{AB}$ is not greater than m. In our experiments, we take $m = 16$ and 32 for practical use, and hence we set $t = 17$ and 33, respectively. Let ct_1 and ct_2 denote the two ciphertexts obtained by packing a part of or whole $\mathbf{A}$ and $\mathbf{B}$, respectively. For example, in our first packing method, we have

$$\mathrm{ct}_1 = \mathrm{ct}^{(1)}_{\mathrm{mat}}(\mathbf{A}) \quad \text{and} \quad \mathrm{ct}_2 = \mathrm{ct}^{(2)}_{\mathrm{pack}}(B_j^T) \quad \text{for some} \quad 1 \le j \le m.$$

In every packing method, we need to multiply $\mathrm{ct} := \mathrm{ct}_1 * \mathrm{ct}_2$ over ciphertexts for secure matrix multiplication $\mathbf{AB}$. In order to avoid decryption failure of the ciphertext ct, it requires $\|\langle \mathrm{ct}, \mathbf{s} \rangle\|_\infty < q/2$ by Lemma 3. Let U denote an upper bound of the ∞-norm size $\|\langle \mathrm{ct}', \mathbf{s} \rangle\|_\infty$ for any fresh ciphertexts $\mathrm{ct}' \in (R_q)^2$. We clearly have

$$\|\langle \mathrm{ct}, \mathbf{s} \rangle\|_\infty = \|\langle \mathrm{ct}_1, \mathbf{s} \rangle \cdot \langle \mathrm{ct}_2, \mathbf{s} \rangle\|_\infty \le nU^2$$

by the well-known fact that $\|a \cdot b\|_\infty \le n\|a\|_\infty \|b\|_\infty$ for $a, b \in R = \mathbb{Z}[x]/(x^n + 1)$. According to the experimental estimation of [5], we may take $U = 2t\sigma^2\sqrt{n}$ in practice. Therefore it suffices to take a prime q satisfying

$$2n(2t\sigma^2\sqrt{n})^2 = 8n^2t^2\sigma^4 \le q. \tag{9}$$

With respect to the degree parameter n, the previous method [9] requires $n \geq m$, our first method $n \geq m^2$, and our second method $n \geq m^3$. In particular, we take $n = (2^5)^3 = 32768$ in our second method for $m = 32 = 2^5$. In this case, we approximately have $q \geq 2^{3+30+10+12} = 2^{55}$ by (9) since $t \approx 2^5$. Then we always fix one prime q of 60-bit for the binary case. Note that with such q, we can avoid decryption failure of ct in every method since our second method requires the largest n and q. Furthermore, since the ciphertext modulus q is less than 64-bit, we estimate that the performance over a 64-bit machine would not change when we use smaller q. We remark that for such q, we need $n \geq 2048$ for more than 80-bit security (see [5] for more details on the security).

5.1.2. Case of non-binary matrix multiplication

In this case, we assume that the entries of two $m \times m$ matrices $\mathbf{A}$ and $\mathbf{B}$ are less than p-bit. In our experiments, we took $r = 2$ and hence $d = p$ in Theorems 4, 9 and 10. We fix $\sigma = 8$ as in the binary case. Let ct_1 and ct_2 denote the two ciphertexts obtained by packing a part of or whole $\mathbf{A}$ and $\mathbf{B}$, respectively. For example, in our second packing method, we have

$$\mathrm{ct}_1 = \mathrm{ct}^{(1)}_{\mathrm{mat}}(\mathbf{A}) \quad \text{and} \quad \mathrm{ct}_2 = \mathrm{ct}^{(2)}_{\mathrm{mat}}(\mathbf{B}).$$

In every method, each coefficient of the decryption polynomial of $\mathrm{ct} = \mathrm{ct}_1 * \mathrm{ct}_2$ is not greater than mp (see [10, Section 3.2] for details). Hence, in every method, it is sufficient to take $t = mp+1$ as the plaintext modulus. For the degree parameter n, the previous method requires $n \geq 2mp$ by Theorem 4, our first method $n \geq 2mp(m+1)$ by Theorem 9, and our second method $n \geq 2p(m^3+m+1)$ by Theorem 10. In our experiments, we take $m = 16$ and $p = 10$, and hence we fix $t = mp + 1 = 161 < 2^8$. In this setting, we take

$$n = 2^{13} = 8192 \geq 2mp(m+1) \quad \text{and} \quad n = 2^{17} = 131072 \geq 2p(m^3+m+1)$$

in our first and second method, respectively. When we set $n = 2^{17}$, by inequality (9), it requires $q \geq 8n^2t^2\sigma^4 \approx 2^{3+34+16+12} = 2^{65}$ in order to avoid decryption failure of ct. In our experiments, we took one prime q of 70-bit in every method for a margin. For such q, it requires $n \geq 2048$ for more than 80-bit security as in the binary case, and hence we set $n = 2048 \geq 2mp$ in the previous method.

5.2. Implementation results

We implemented the SHE scheme with our packing methods for secure matrix multiplication. Our experiments ran on an Intel Core i7-4790 CPU with 3.60 GHz and 8.00 GB RAM, using [PARI] library [8] (version 2.7.5) in C programs.

TABLE 1. Performance of secure matrix multiplication of $m \times m$ matrices with binary entries.

(In seconds)	m	$(n, \log_2(q), t, \sigma)$	Encryption	Secure Matrix Mul.	Decryption	Total time
Previous method [9]	16	$(2048, 60, 17, 8)$	0.3987	11.5704	3.5352	15.5043
	32	**(2048,60,33,8)**	**0.7950**	**47.8745**	**14.063**	**62.7325**
Our first method	16	$(2048, 60, 17, 8)$	0.0150	0.3064	0.2170	0.5384
	32	**(2048,60,33,8)**	**0.0160**	**0.5780**	**0.3592**	**0.9532**
Our second method	16	$(4096, 60, 17, 8)$	0.0439	0.0438	0.0328	0.1205
	32	**(32768,60,33,8)**	**0.1814**	**0.5220**	**0.4036**	**1.1070**

TABLE 2. Performance of secure matrix multiplication of 16×16 matrices with entries less than $p = 10$ bits.

(In seconds)	$(n, \log_2(q), t, \sigma)$	Encryption	Secure Matrix Mul.	Decryption	Total time
Previous method [10]	$(2048, 70, 161, 8)$	1.4454	9.4483	3.6581	14.5518
Our first method	$(8192, 70, 161, 8)$	1.5996	1.5797	0.7624	3.9417
Our second method	$(131072, 70, 161, 8)$	4.2024	2.0834	0.9797	7.2655

5.2.1. Binary case

Our chosen parameters and implementation results for the binary case are given in Table 1. As shown in Table 1, our two methods are much faster than the previous method [9]. Specifically, our first method is about $2m$ times faster than the previous method [9]. For $m = 16$, our second method is several times faster than the first one. However, for $m = 32$, our first method is slightly faster than the second one in total. This is due to the fact that our second method requires $n \geq m^3 = 32^3 = 32768$ while our first method requires only $n \geq m^2 = 32^2 = 1024$ (as in [9], we took $n \geq 2048$ for enough security). Then we conclude that our second method is still efficient for $m \leq 32$, but our first method becomes much more efficient for $m \geq 64$ in the binary case.

Remark 11. In our experiments, we took m of the special form $m = 2^k$ to make it easier to choose parameters of the SHE scheme (recall that the degree parameter n should be chosen as a 2-power integer). However, performance of secure matrix multiplication does not depend on the form of m. Furthermore, for m between $m = 32$ and 64, we estimate from Table 1 that our first method would be slightly faster than our second one.

5.2.2. Non-binary case

In Table 2, we give a performance comparison of secure matrix multiplication of 16×16 matrices (i.e., $m = 16$) with entries less than $p = 10$ bits. Our chosen parameters for the SHE scheme and implementation results are given in Table 2. As we can see from the Table 2, our first method is around 5-times faster and second method is around 2-times faster than the previous method. In this case we are not using same parameter for our both the methods as in the previous method. Our first and second methods require $n = 8192 \geq 2mp(m+1)$ and $n = 131078 \geq mp(m^3 + m + 1)$, respectively. Because of very large n in our second method, our first method is faster than the second method. However, in sense of computation efficiency, our second method is better than the first method because we need m-homomorphic multiplications for our first method and just only one homomorphic multiplication for our second method. Once we increase the size of entries and dimension of matrices our first method will be much faster.

Remark 12. When we fix m and take larger p, our first method becomes much faster than our second one since our first method permits us to take smaller parameters of the SHE scheme. Therefore, unfortunately, our second method has advantage on performance only in case of very small m and p such as $(m, p) = (16, 5)$. More specifically, for $m \geq 32$, our first method is faster than our second one for any p since our first method is faster even in the binary case from Table 1.

Acknowledgements. We are grateful to the anonymous referee for their helpful comments.

REFERENCES

[1] BONEH, D.—GOH, E.—NISSIM, K.: *Evaluating* 2-DNF *formulas on ciphertexts*, in: Theory of Cryptography—TCC '05, Lecture Notes in Comput. Sci., Vol. 3378, Springer-Verlag, Berlin, 2005, pp. 325–341.

[2] BRAKERSKI, A., LANGLOIS, A., PEIKERT, C.—REGEV, O.—STEHLÉ, D.: *Classical hardness of learning with errors*, in: Symposium on Theory of Computing—STOC '13, ACM New York, NY, USA, 2013, pp. 575–584.

[3] BRAKERSKI, A.—VAIKUNTANATHAN, V.: *Fully homomorphic encryption from ring-*LWE *and security for key dependent messages*, in: Advances in cryptology–CRYPTO '11, (P. Rogaway, ed.), Lecture Notes in Comput. Sci., Vol. 6841, Springer, Heidelberg, 2011. pp. 505–524.

[4] GENTRY, C.: *Fully homomorphic encryption using ideal lattices*, in: Proceedings of the 41th Symposium on Theory of Computing—STOC '09, ACM New York, NY, USA, 2009, pp. 169–178.

[5] LAUTER, K.—NAEHRIG, M.—VAIKUNTANATHAN, V.: *Can homomorphic encryption be practical ?*, in: Proceedings of the 3rd ACM workshop on Cloud computing security workshop—CCSW '11 , ACM New York, NY, USA, 2011, pp. 113–124.

[6] LYUBASHEVSKY, V.—PEIKERT, C.—REGEV, O.: *On ideal lattices and learning with errors over rings*, in: Advances in cryptology—EUROCRYPT'10, Lecture Notes in Comput. Sci., Vol. 6110, Springer-Verlag, Berlin, 2010, pp. 1–23.

[7] RIVEST, R.—SHAMIR, A.—ADLEMAN, L.: *A method for obtaining digital signatures and public-key cryptosystems*, Commun. ACM **21** (1978), no.(2), 120–126.

[8] The PARI Group, Bordeaux, PARI/GP. http://pari.math.u-bordeaux.fr/doc.html

[9] YASUDA, M.—SHIMOYAMA, T.—KOGURE, J.—YOKOYAMA, K.—KOSHIBA, T.: *New packing method in somewhat homomorphic encryption and its applications*, Security and Communication Networks, **8** (2015), 2194–2213.

[10] YASUDA, M.—SHIMOYAMA, T.—KOGURE, J.—YOKOYAMA, K.—KOSHIBA, T.: *Secure statistical analysis using* RLWE*-based homomorphic encryption*, ACISP'15 (E. Foo and D. Stebila, eds.), Lectute Notes in Comput Sci., Vol. 9144, Springer-Verlag, Berllin, 2015, pp. 471–487.

Received September 20, 2016

Institute of Mathematics for Industry and
Graduate School of Mathematics
Kyushu University
744 Motooka, Nishi-ku
Fukuoka 819-0395
JAPAN

E-mail: duong@imi.kyushu-u.ac.jp
p-mishra@math.kyushu-u.ac.jp
yasuda@imi.kyushu-u.ac.jp