Secure Pattern Matching using Somewhat Homomorphic Encryption

Masaya Yasuda
FUJITSU LABORATORIES LTD., 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, 211-8588, Japan
yasuda.masaya@jp.fujitsu.com

Takeshi Shimoyama
FUJITSU LABORATORIES LTD., 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, 211-8588, Japan
shimo-shimo@jp.fujitsu.com

Jun Kogure
FUJITSU LABORATORIES LTD., 1-1, Kamikodanaka 4-chome, Nakahara-ku, Kawasaki, 211-8588, Japan
kogure@jp.fujitsu.com

Kazuhiro Yokoyama
Department of Mathematics, Rikkyo University, Nishi-Ikebukuro, Tokyo 171-8501, Japan
kazuhiro@rikkyo.ac.jp

Takeshi Koshiba
Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura, Saitama, 338-8570, Japan
koshiba@mail.saitama-u.ac.jp

ABSTRACT

The basic pattern matching problem is to find the locations where a pattern occurs in a text. Recently, secure pattern matching has been received much attention in various areas, including privacy-preserving DNA matching and secure biometric authentication. The aim of this paper is to give a practical solution for this problem using homomorphic encryption, which is public key encryption supporting some operations on encrypted data.

In this paper, we make use of the somewhat homomorphic encryption scheme presented by Lauter, Naehrig and Vaikuntanathan (ACM CCSW 2011), which supports a limited number of both additions and multiplications on encrypted data. In their work, some message encoding techniques are also presented for enabling us to efficiently compute sums and products over the integers. Based on their techniques, we propose a new packing method suitable for an efficient computation of multiple Hamming distance values on encrypted data. Our main extension gives two types of packed ciphertexts, and a linear computation over packed ciphertexts gives our desired results.

We implemented the scheme with our packing method. Our experiments ran in an Intel Xeon at 3.07 GHz with our software library using inline assembly language in C programs. Our optimized implementation shows that the packed encryption of a text or a pattern, the computation of multiple Hamming distance values over packed ciphertexts, and the decryption respectively take about 3.65 milliseconds (ms), 5.31 ms, and 3.47 ms for secure exact and approximate pattern matching of a binary text of length 2048. The total time is about 12.43 ms, which would give the practical performance in real life. Our method gives both faster performance and lower communication than the state-of-the-art work for a binary text of several thousand bits in length.

CCSW'13, November 8, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517488.2517497.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

Theory, Algorithms

Keywords

pattern matching, somewhat homomorphic encryption, packing method, the Hamming distance

1. INTRODUCTION

The recent development of cloud storage and computing easily allows users to outsource their data to cloud services. On the other hand, new privacy concerns for both individuals and business have risen (see [13] for example). With homomorphic encryption, users send their data in encrypted form to the cloud, and the cloud still can perform computations on encrypted data. Since all data in the cloud are in encrypted form, the confidentiality of users' data is preserved irrespective of any actions in the cloud. Therefore homomorphic encryption would give a powerful tool to break several barriers to the adoption of cloud services for various uses.

1.1 Homomorphic Encryption Schemes

Homomorphic encryption schemes proposed before 2000 can only support simple operations such as either additions or multiplications on encrypted data (see [15, 23, 34] for examples), and hence applications of these schemes are very

limited. Actually, typical applications of additively homomorphic encryption such as the Paillier scheme [34] are electronic voting and e-cash. The first scheme supporting both additions and multiplications is the BGN scheme [6] proposed in 2005, which is based on pairings over elliptic curves. However, the BGN scheme can handle a number of additions but depth-one multiplications.

In 2009, Gentry in [19] proposed a concrete construction of a fully homomorphic encryption (FHE) scheme supporting arbitrary computations on encrypted data. After Gentry's breakthrough, FHE is expected to be applied to various areas including cloud computing, and a number of new schemes, improvements, and implementations have been proposed (see [7, 8, 12, 18, 21, 22, 30] for examples of recent work). At present, there are the following four main variant schemes:

- ideal lattices based schemes [19, 20, 36],
- integers based schemes [12, 14, 17],
- the learning with errors (LWE) based schemes [8, 9, 10, 18], and
- the NTRU encryption based schemes [30].

However, despite of rapid developments in FHE, currently known schemes are impractical, and hence it is believed to need a long way for the practical use (see [14, 20] for the performance of a "pure" FHE scheme, and also [22] for that of a "leveled" FHE scheme).

In this paper, we rather focus on somewhat homomorphic encryption (SHE), which is well known as a building block for the FHE construction. Sometimes, it is abbreviated as "SwHE" (note that we do not consider the BGN scheme as an SHE scheme in this paper). Although SHE can only support a limited number of both additions and multiplications, it is much faster and more compact than FHE. Therefore SHE can give a practical solution for wider applications than additively homomorphic encryption, and it is coming to attention to research on applications with SHE schemes (see [28] for typical work, and also [5, 16] for recent work).

1.2 Pattern Matching Problems

As the application scenario, we consider the following pattern matching problems (for simplicity, we only consider binary vectors as in the work of [24]):

Definition 1. Given a binary text $T \in \{0,1\}^k$ of length k and a binary pattern $P \in \{0,1\}^\ell$ of length ℓ with $k \geq \ell$ (for example, $T =$ '01010' and $P =$ '010').

(a) The exact pattern matching problem is to find the locations where P occurs in T, namely, to find the set

$$S_{\mathrm{MP}} = \left\{0 \leq i \leq k-\ell \mid T^{(i)} = P\right\}$$

of matching positions, where for $T = (t_0, \ldots, t_{k-1})$, let

$$T^{(i)} = (t_i, t_{i+1}, \ldots, t_{i+\ell-1})$$

denote its i-th sub-vector of length ℓ.

(b) The approximate pattern matching problem is to find the locations where the Hamming distance between $T^{(i)}$ and P is less than a pre-defined threshold $\tau \leq \ell$, namely, to find the set

$$S_{\mathrm{AMP}} = \left\{0 \leq i \leq k-\ell \mid d_H(T^{(i)}, P) \leq \tau\right\}$$

of approximate matching positions, where $d_H(A, B)$ denotes the Hamming distance between two binary vectors A and B. Note that we clearly have $S_{\mathrm{MP}} = S_{\mathrm{AMP}}$ when we set $\tau = 0$.

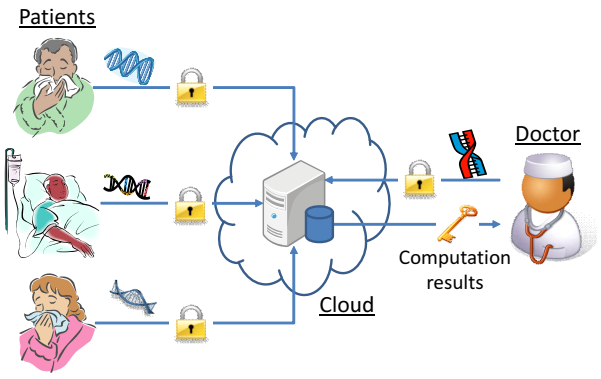

Figure 1: Analysis of personal DNA sequences privately to the cloud using homomorphic encryption

Pattern matching is one of the fundamental tools in computer science, and it can be used in many areas including text processing, database search, data mining, computational biology and network security. Recently, pattern matching with privacy-preserving has been received much attention, for examples,

- privacy-preserving DNA matching [2, 26, 38],
- secure biometric authentication [4, 33], and
- anomaly detection in RFID [27].

1.2.1 *Application Scenario and Some Remarks*

We introduce an application scenario of secure patten matching for analyzing personal DNA sequences, which is shown in Figure 1; A doctor knows a DNA sequence pattern P causing a certain disease, and patients want the doctor to analyze whether they would get the disease or not. Since the occurrence rate and matching positions of the pattern P in DNA sequences are required in DNA analysis, the above pattern matching problems are deeply related with our application scenario. For analysis of a lot of patients, the doctor uses the cloud for an outsourced storage and a computation resource. For the privacy protection from the cloud, the doctor generates keys of homomorphic encryption, and distributes the public key to patients. Then each patient can encrypt his or her own DNA sequence text T by the public key, and upload the encrypted text to the cloud. For privately computing in the cloud, the doctor encrypts the pattern P, and sends the encrypted pattern to the cloud. The cloud computes all Hamming distance values $d_H(T^{(i)}, P)$ on encrypted data, and return the encrypted results to the cloud. Finally, the doctor decrypts the encrypted results with the secret key to obtain all Hamming distance values, and uses them to identify which patient would get the disease in the future. Due to the use

of homomorphic encryption, all computations in the cloud are performed on encrypted data, and hence the confidentiality of the DNA sequences in the cloud is protected by homomorphic encryption.

Remark 1. At present, it is difficult to compare Hamming distance values $d_H(T^{(i)}, P)$ with a threshold τ on encrypted data using homomorphic encryption. Therefore, in the above scenario, we consider to compute all Hamming distance values on encrypted data in the cloud. For pattern matching problems, the set S_{MP} or S_{AMP} of matching positions can be obtained after the decryption. Therefore more information than matching positions can be revealed to a decryptor (in some cases the entire text string may be revealed to a decryptor), but there is no problem in the above scenario since we assume the doctor as a trusted party with the secret key. In this work, we give an efficient computation of multiple Hamming distance values on encrypted data. Hence we remark that this work gives a building block for secure pattern matching (for example, a much more useful application would be required to compute the minimum Hamming distance between a pattern and string texts from a database).

1.3 Our Contributions

We use the SHE scheme presented by Lauter, Naehrig, and Vaikuntanathan in the work of [28], which is slightly modification of the scheme proposed by Brakerski and Vaikuntanathan in [9, Section 3.2]. The scheme is based on the ring-LWE assumption, and its (somewhat) homomorphic correctness holds over the ring of the plaintext space

$$R_t = \mathbb{Z}[x]/(x^n + 1, t),$$

where n is the lattice dimension and t is the modulus parameter (see [28, Section 3.3] for the correctness). For secure exact and approximate pattern matching, we give an efficient method to compute multiple Hamming distance values

$$d_H(T^{(i)}, P) \text{ for } 0 \leq i \leq k - \ell \tag{1}$$

on encrypted data.

New Packing Method. Once we encrypt a binary text or pattern bit by bit, we have difficulty on both the encrypted data size and the performance. In contrast, our method transforms a binary vectors of length less than n into a certain polynomial in the ring R_t, and then packs it in a single ciphertext, which enables us to considerably reduce both the size and the performance. The idea of our packing method is based on the message encoding techniques presented in [28, Section 4], whose aim is to give an efficient computation of sums and products over the integers. Our extension

1. gives two types of packed ciphertexts for a text T and a pattern P, and
2. makes use of the homomorphic property over the ring R_t for computing multiple values (1) at the same time.

Concretely speaking, a certain linear computation over packed ciphertexts of T and P enables us to efficiently compute a polynomial in the ring R_t with each x^i-coefficient equal to the value (1) on encrypted data, and hence only one time decryption gives our desired multiple values (1). In particular, our packing method requires that the SHE scheme only can support depth-one multiplications, which condition gives the practical performance (see [28, Table 2] for implementation results of the SHE scheme).

Optimal Implementation. To demonstrate the efficiency of our packing method, we implemented the SHE scheme for four lattice dimensions $n = 2048, 4096, 8192$ and 16348 in order to obtain various security levels (as the lattice dimension is larger, the security level is higher). Note that all these four parameters are estimated to achieve 80-bit security with an enough margin according to the security analysis in the work of [28]. Furthermore, compared to the implementation of [28, Section 5] with the computer algebra system MAGMA, ours is considerably optimized since we used our software library using inline assembly language in C programs for all computations in the base ring of the ciphertext space.

Remark 2. Our packing method makes use of the structure of the special ring $\mathbb{Z}[x]/(x^n + 1)$, and it can be applied to the BGV scheme of [8], and the scheme recently proposed by Fan and Vercauteren [18] (the performances in these schemes are estimated to be almost the same as in the scheme of [28] used in this paper). However, it cannot be applied to the BGN scheme [6] since it is based on pairings over elliptic curves. Furthermore, as noted in [28, Section 1.2], the homomorphic multiplication in the BGN scheme is much slower than that in lattice-based schemes since it requires pairing computations for the homomorphic multiplication in the BGN scheme. Therefore we remark that the SHE scheme of [28] with our packing method is estimated to be much faster than the BGN scheme for computing multiple values (1) on encrypted data.

1.4 Comparison with Related Work

At present, there are two basic approaches to realize secure pattern matching, namely,

- the multi-party computation (MPC) approach, and
- the homomorphic encryption approach, on which we focus in this work.

1.4.1 MPC Approach

There are a variety of previous work for secure pattern matching based on MPC (see [24, 25, 39] for recent work). In 2012, Baron et al. in [3] presented an efficient two-parties protocol of secure pattern matching for more expressive search queries including single character wildcards and substring pattern matching. They called their protocol "5PM (5ecure Pattern Matching)", which is the first one to have sub-linear communication complexity in circuit size (in general, MPC has linear communication complexity). Their main improvement is to use additively homomorphic encryption schemes in computing linear operations such as the inner product and the matrix multiplication. In the semi-honest model (resp. the malicious model), 5PM requires $O((k + \ell)\lambda)$ bandwidth and 2 rounds (resp. $O((k + \ell)\lambda^2)$ bandwidth and 8 rounds) of communication, where k is the text length, ℓ is the pattern length, and λ is the security parameter. In [3, Section 8], they also reported the performance of their implementation of 5PM. In particular, it took 0.67 (resp. 6.81 and 64.76) seconds on an Intel dual quad-core 2.93 GHz with 8 GB memory to compute secure

Table 1: A comparison of our work with 5PM and FHE (k: text length, ℓ: pattern length, λ: security parameter, n: lattice dimension of the SHE scheme, $n = 2048$ gives the best performance in our implementation)

	Communication cost		Computation cost	Possible types of pattern matching
	Bandwidth	Rounds		
5PM [3] (MPC-based)	$O((k+\ell)\lambda)$	2 (semi-honest)	practical in using $k \leq 1000 \sim 10000$	exact, approximate, wildcards, substring
	$O((k+\ell)\lambda^2)$	8 (malicious)		
FHE	$O(\ell)$	2	impractical	any
Our work (SHE scheme)	$O(\lceil \ell/n \rceil)$	2	faster than 5PM	exact, approximate

single character wildcards and substring pattern matching for a pattern of 100 characters and a text of 1,000 characters (resp. a text of $10,000$ and $100,000$ characters), in using the Paillier scheme [34] of 1024 bit key length and using the DNA alphabet as a character (the performance of 5PM increases linearly in the text size).

1.4.2 Homomorphic Encryption Approach

Using FHE, we can theoretically construct the following protocol on any pattern matching; A client encrypts its pattern P of length ℓ using an FHE scheme and only sends the encrypted pattern to a server in whose database an encrypted text is stored. The server computes appropriate pattern matching on encrypted data, and only sends the encrypted result to the client having the secret key. Finally, the client decrypts the encrypted data to obtain the desired result. Such a protocol only requires $O(\ell)$ bandwidth and 2 rounds of communication (assume to use the bit-wise encryption), and hence FHE has the advantage of communication cost compared to 5PM, but its performance is estimated to be far from practical as described in §1.1.

Comparison.

In our work, we use the SHE scheme of [28] with our packing method. The lattice dimension $n = 2048$ gives us the best performance with the enough security (more than 80-bit security level). In this setting, we can pack a binary vector of length less than $n = 2048$ in a single ciphertext. Therefore it requires $O(\lceil \ell/n \rceil) = O(\lceil \ell/2048 \rceil)$ bandwidth and 2 rounds of communication as in the FHE case. Furthermore, our implementation results show that the packed encryption of a pattern, the computation of multiple Hamming distance values (1) on encrypted data, and the decryption respectively take about 3.65 ms, 5.31 ms, and 3.47 ms on an Intel Xeon X3480 at 3.07 GHz with 16 GB memory for secure exact and approximate pattern matching of a binary text of length less than $n = 2048$ (see §4.2 below for details). Hence the total cost is about 12.43 ms, which is about 50 times faster than 5PM when we ignore the difference of the PC performance and implementation levels (cf. in 5PM, it took 670 ms for a text of length 1000). Our software library with assembly-level optimizations has the main advantage of the fast processing performance of the 64-bit $\times$ 64-bit $\rightarrow$ 128-bit multiplication operation, which needs about 4 clocks. When we implement the operations in standard C programs, it is estimated to need about $4 \times 4 = 16$ clocks and hence we estimate that our implementation results would be $4 \sim 5$ times slower, which is still at least 10 times faster than 5PM. Hence we estimate that our work is faster than 5PM on the same PC with the same implementation. Furthermore, for a longer binary text, we can use our method by dividing it into binary vectors of length less than n, and then the performance increases linearly in the text size as in 5PM. We remark that compared to the case of packing in a single ciphertext using a large lattice dimension n, the case of dividing into blocks of length $n = 2048$ would be faster, but in this case we have difficulty in handling the boundary between blocks. Due to the space restriction, we will not discuss how to handle the boundary.

In Table 1, we give a comparison of our work with the related work described above. Although our work does not support various pattern matching, our work has much less communication cost than 5PM and FHE. Furthermore, our work give the faster performance than 5PM for basic secure pattern matching.

Remark 3. The authors in [1] presented several efficient protocols for the secure and private outsourcing of linear algebra computations including approximate pattern matching by the Hamming distance (see [1, Section 5.3] for details). In their framework, users are not interested in keeping data permanently on an external resource like the cloud, instead, users only want to temporarily use its superior external computation power. Their framework is different from ours, and hence we do not give a comparison with their work.

Notation. The symbols $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ denote the ring of integers, the field of rational numbers, and the field of real numbers, respectively. For a prime number p, the finite field with p elements is denoted by $\mathbb{F}_p$. For two integers z and d, let $[z]_d$ denote the reduction of z modulo d included in the interval $[-d/2, d/2)$ (the reduction of z modulo d included in the interval $[0, d)$ is denoted by $z \bmod d$ as usual). For a vector $\vec{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$, let $||\vec{a}||_\infty$ denote the ∞-norm defined by $\max_i |a_i|$. We let $\langle \vec{a}, \vec{b} \rangle$ denote the inner product of two vectors $\vec{a}$ and $\vec{b}$. Finally, we let $\lg(q)$ denote the logarithm value of an integer q with base 2.

2. PRELIMINARIES

In this section, we give the basic construction of the SHE scheme of [28], which is slightly modification of Brakerski and Vaikuntanathan's scheme [9, 10] (see also [9, Section 3.2] for the construction). The security of the scheme is based on the polynomial LWE assumption as defined below, which is a simplified version of the ring-LWE assumption of [31] (see [9, Definition 1] for details).

Definition 2. For a security parameter λ, let $f(x) = x^n + 1$ be the cyclotomic polynomial for an integer $n = n(\lambda)$ of 2-power. Let $q = q(\lambda)$ be an integer and set $R = \mathbb{Z}[x]/(f(x))$ and $R_q = R/qR$. Let $\chi = \chi(\lambda)$ be a distribution over the

ring R. Then the polynomial LWE assumption $\mathsf{PLWE}_{n,q,\chi}$ (in the Hermite normal form) 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 $s \leftarrow \chi$ 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 Construction of the SHE scheme

For the construction, we need the following four parameters (the notation in this section is almost same as in [28, Section 3]):

n : an integer of 2-power, which defines the base ring $R = \mathbb{Z}[x]/(f(x))$ with the cyclotomic polynomial $f(x) = x^n + 1$ of degree n as in Definition 2. This parameter is called the lattice dimension of the scheme.

q : a prime number with $q \equiv 1 \bmod 2n$, which defines the base ring $R_q = R/qR = \mathbb{F}_q[x]/(f(x))$ of a ciphertext space.

t : an integer with $t < q$ to determine a plaintext space $R_t = \mathbb{F}_t[x]/(f(x))$.

σ : the parameter to define a discrete Gaussian error distribution $\chi = D_{\mathbb{Z}^n,\sigma}$ with the standard deviation σ, namely, 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. In practice, we choose relatively small value such as $\sigma = 4 \sim 8$.

Remark 4. Under the condition $q \equiv 1 \bmod 2n$, the polynomial $f(x) = x^n + 1$ factors modulo q into linear terms

$$f(x) \equiv \prod_{i \in (\mathbb{Z}/2n\mathbb{Z})^*} (x - \zeta^i) \bmod q,$$

where let ζ denote a fixed primitive $2n$-th root of unity. There are some improvements using this factorization (see [22, A.3] for their "double CRT representation" technique). However, this condition is not necessarily required just for the construction of the scheme (but it seems to be necessary for the security discussion in [9, Theorem 1]).

2.1.1 Key Generation

We first choose an element $R \ni s \leftarrow \chi$. We then sample a uniformly random element $a_1 \in R_q$ and an error $R \ni e \leftarrow \chi$. Set $\mathsf{pk} = (a_0, a_1)$ with $a_0 = -(a_1 s + t \cdot e)$ as the public key and $\mathsf{sk} = s$ as the secret key.

2.1.2 Encryption

For a plaintext $m \in R_t$ and the public key $\mathsf{pk} = (a_0, a_1)$, the encryption samples $R \ni u, f, g \leftarrow \chi$ and compute the "fresh" ciphertext given by

$$\begin{aligned}\mathsf{Enc}(m, \mathsf{pk}) &= (c_0, c_1) \in (R_q)^2 \\ &= (a_0 u + tg + m, a_1 u + tf),\end{aligned}$$

where $m \in R_t$ is considered as an element of R_q in the natural way due to the condition $t < q$.

2.1.3 Decryption

For a ciphertext $\mathsf{ct} = (c_0, \ldots, c_\xi) \in (R_q)^{\xi+1}$, the decryption with the secret key $\mathsf{sk} = s$ is computed by

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

where $\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$ (note that the homomorphic multiplication defined below makes the ciphertext length longer, and let $\xi + 1$ denote the ciphertext length). For the vector $\vec{s} = (1, s, s^2, \ldots)$, we can also write

$$\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}) = [\langle \mathsf{ct}, \vec{s} \rangle]_q \bmod t.$$

For the sake of simplicity, we call $\vec{s}$ the secret key vector.

2.1.4 Homomorphic Operations

Let $\mathsf{ct} = (c_0, \ldots, c_\xi) \in (R_q)^{\xi+1}$, $\mathsf{ct}' = (c'_0, \ldots, c'_\eta) \in (R_q)^{\eta+1}$ be two ciphertexts (let $\xi+1, \eta+1$ denote the length of ciphertexts $\mathsf{ct}, \mathsf{ct}'$, respectively). The homomorphic addition "$\dot{+}$" is computed by component-wise addition of ciphertexts, namely, we have

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

The homomorphic subtraction is computed by component-wise subtraction. Furthermore, the homomorphic multiplication "$*$" is computed by

$$\mathsf{ct} * \mathsf{ct}' = (\hat{c}_0, \ldots, \hat{c}_{\xi+\eta}),$$

where we consider ciphertexts $\mathsf{ct}, \mathsf{ct}'$ as elements of $R_q[z]$ by an embedding map

$$\iota : (R_q)^r \ni (v_0, \ldots, v_{r-1}) \mapsto \sum_{i=0}^{r-1} v_i z^i \in R_q[z]$$

for any $r \geq 1$, and compute

$$\sum_{i=0}^{\xi+\eta} \hat{c}_i z^i = \left(\sum_{i=0}^{\xi} c_i z^i\right) \cdot \left(\sum_{i=0}^{\eta} c'_i z^i\right) \in R_q[z].$$

Here we give the lemma on the security of the SHE scheme (see [9, Proposition 1] for the security, and also [9, Theorem 1] for a connection of the average-case hardness of $\mathsf{PLWE}_{n,q,\chi}$ with the worst-case hardness of ideal lattice problems).

Lemma 1 (security). Given the parameters (n, q, t, σ), the scheme constructed above is provably secure in the sense of IND-CPA under the polynomial LWE assumption $\mathsf{PLWE}_{n,q,\chi}$ with $\chi = D_{\mathbb{Z}^n,\sigma}$.

2.2 Correctness of the SHE Scheme

By correctness, we mean that the decryption can recover the operated result over plaintexts after some homomorphic operations over ciphertexts. For the scheme constructed in §2.1, it follows from the proof of [28, Lemma 3.3] that the homomorphic operations over ciphertexts correspond to the ring structure of the plaintext space R_t, namely, we have

- (Addition) $\mathsf{Dec}(\mathsf{ct} \dot{+} \mathsf{ct}', \mathsf{sk}) = m + m' \in R_t$, and
- (Multiplication) $\mathsf{Dec}(\mathsf{ct} * \mathsf{ct}', \mathsf{sk}) = m \times m' \in R_t$

for ciphertexts $\mathsf{ct}, \mathsf{ct}'$ corresponding to plaintexts m, m', respectively. However, the scheme merely gives an SHE scheme (not an FHE scheme), and its correctness holds under the following condition (see the proof of [28, Lemma 3.3]):

Lemma 2 (successful decryption). For a ciphertext ct, the decryption $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$ recovers the correct result if $\langle \mathsf{ct}, \vec{s} \rangle \in R_q$ does not wrap around mod q, namely, if the condition

$$||\langle \mathsf{ct}, \vec{s} \rangle||_\infty < \frac{q}{2}$$

is satisfied, where for $a = \sum_{i=0}^{n-1} a_i x^i \in R_q$ let

$$||a||_\infty = \max_{0 \le i \le n-1} |a_i|$$

denote the ∞-norm of its coefficient representation.

3. NEW PACKING METHOD

In this section, we give a brief review of the packing method proposed in [28], and present a new packing method for secure pattern matching.

The packing method of [28]. For the reduction of both the size and the performance in the SHE scheme, Lauter, Naehrig and Vaikuntanathan in [28] introduce some message encoding techniques. Their main technique is to encode integers in a single ciphertext so that it enables us to efficiently compute their sums and products over the integers (see [28, Section 4.1] for details). Their idea is to partition an integer M of at most n bits into a binary vector $(M_0, \ldots, M_{n-1})$, create a polynomial given by (where n is the lattice dimension parameter described in §2.1)

$$\mathsf{pm}(M) = \sum_{i=0}^{n-1} M_i x^i$$

of degree $(n-1)$, and finally encrypt M as

$$\mathsf{ct}_{\mathrm{pack}}(M) = \mathsf{Enc}\left(\mathsf{pm}(M), \mathsf{pk}\right),$$

where we consider the polynomial $\mathsf{pm}(M)$ as an element of R_t for sufficiently large t. Note that we have

$$\mathsf{pm}(M)|_{x=2} = M$$

for any integer M of at most n bits, where $a(x)|_{x=2}$ denotes the value substituted $x = 2$ for a polynomial $a(x)$. For two integers M and M' of at most n bits, the homomorphic addition of $\mathsf{ct}_{\mathrm{pack}}(M)$ and $\mathsf{ct}_{\mathrm{pack}}(M')$ gives the polynomial addition $\mathsf{pm}(M) + \mathsf{pm}(M')$ on encrypted data by the correctness of the scheme as described in §2.2 and it also gives the integer addition $M + M'$ since

$$\mathsf{pm}(M) + \mathsf{pm}(M')\big|_{x=2} = M + M'.$$

However, the integer multiplication $M \cdot M'$ causes a problem since the polynomial multiplication $\mathsf{pm}(M) \cdot \mathsf{pm}(M')$ has larger degree than n in general. Their solution is to encode integers of at most n/d bits if we need to perform d times homomorphic multiplications over ciphertexts. Therefore their solution is useful in computing low degree multiplications such as the standard deviation.

3.1 Our Method for Secure Pattern Matching

While their techniques give efficient computations over the integers, we present a new packing method suitable for an efficient computation of multiple Hamming distance values (1) on encrypted data. The idea of our packing method is based on theirs. However, unlike their technique, we give two types of packed ciphertexts in order to make use of the ring structure of the plaintext space R_t. Note that our method presented below specializes in the ring structure of R with the property $x^n = -1$ in R.

3.1.1 Definition of Our Packing Method

Now, let us define our packing method.

Definition 3. Assume $n \ge k \ge \ell \ge 1$. Two types of packed ciphertexts for a text T and a pattern P are defined as follows:

(A) For a binary text $T = (t_0, t_1, \ldots, t_{k-1})$ of length k, let $m_{\mathrm{txt}}(T)$ be the packed message defined by

$$m_{\mathrm{txt}}(T) = \sum_{i=0}^{k-1} t_i x^i.$$

Then the packed encryption of the text T is defined by

$$\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}) = \mathsf{Enc}(m_{\mathrm{txt}}(T), \mathsf{pk}),$$

where we consider $m_{\mathrm{txt}}(T)$ as an element of R_t for sufficiently large t. This packed encryption is the same as the encoding technique of [28].

(B) For a binary pattern $P = (p_0, p_1, \ldots, p_{\ell-1})$ of length ℓ, let $m_{\mathrm{ptn}}(P)$ be the packed message given by

$$m_{\mathrm{ptn}}(P) = -\sum_{j=0}^{\ell-1} p_j x^{n-j}.$$

Similarly to the above case, the packed encryption of the pattern P is defined by

$$\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}) = \mathsf{Enc}(m_{\mathrm{ptn}}(P), \mathsf{pk}).$$

This is new packed encryption in our method.

As seen from the above definition, we can pack a text T or a pattern P of length less than n in a single ciphertext. Therefore, compared to the bit-wise encryption, our packing method enables us to considerably reduce the encrypted data size. In §3.1.3 below, we shall give a comparison with the other known packing methods.

3.1.2 Computations over Packed Ciphertexts

Combinations of two types (A) and (B) of packed ciphertexts give us some efficient computations. In the following, we give the basic result (actually, we define our packing method so that the following result holds):

Proposition 1 (inner product). Let ct be a ciphertext given by the homomorphic multiplication $\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}) * \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk})$. Let

$$m = \sum_{i=1}^{n-1} m_i x^i \in R_t$$

denote the decryption result $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk})$ of the ciphertext ct. If the correctness for the ciphertext ct is satisfied, then we have

$$m_i \equiv \langle T^{(i)}, P \rangle \bmod t \text{ for } 0 \le i \le k - \ell. \tag{2}$$

In other words, the homomorphic multiplication over packed ciphertexts gives us multiple inner products (2) on encrypted data for sufficiently large t.

Proof. The homomorphic multiplication of two packed ciphertexts $\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk})$ and $\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk})$ corresponds to the multiplication of two elements $m_{\mathrm{txt}}(T)$ and $m_{\mathrm{ptn}}(P)$ in the ring R_t (see §2.2 for the correctness of the scheme). Since $x^n = -1$ in the ring R_t, we have

$$\begin{aligned} m_{\mathrm{txt}}(T) \cdot m_{\mathrm{ptn}}(P) &= \left(\sum_{i=0}^{k-1} t_i x^i\right) \cdot \left(-\sum_{j=0}^{\ell-1} p_j x^{n-j}\right) \\ &= -\sum_{j=0}^{\ell-1} \sum_{h=0}^{k-j-1} t_{h+j} p_j x^{n+h} \ (h = i - j) \\ &= \sum_{h=0}^{k-\ell} \sum_{j=0}^{\ell-1} t_{h+j} p_j x^h \\ &\quad + \text{polynomials of deg} \geq k - \ell + 1. \end{aligned}$$

From the above equation, we see that the x^h-coefficient of the polynomial $m_{\mathrm{txt}}(T) \cdot m_{\mathrm{ptn}}(P) \in R_t$ is equal to the inner product

$$\langle T^{(h)}, P \rangle = \sum_{j=0}^{\ell-1} t_{h+j} p_j \text{ for } 0 \leq h \leq k - \ell.$$

Therefore the result of this proposition holds under the correctness for the ciphertext ct. □

In the following, we apply Proposition 1 to compute multiple Hamming distance values (1) over packed ciphertexts for secure exact and approximate pattern matching:

Theorem 1 (matching computation). Let $\mathsf{ct}_{\mathrm{spm}}$ be a ciphertext given by the homomorphic operation

$$\begin{aligned} \mathsf{ct}_{\mathrm{spm}} = \mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}) * C_\ell \dot{+} \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}) * C'_k \\ \dot{+} (-2\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk})) * \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \end{aligned} \tag{3}$$

where for k, ℓ, let

$$C_\ell = -\sum_{j=0}^{\ell-1} x^{n-j} \text{ and } C'_k = \sum_{i=0}^{k-1} x^i$$

(the homomorphic operations on C_ℓ and C'_k are computed as in §2.1.4). Let

$$m = \sum_{i=0}^{n-1} m_i x^i \in R_t$$

denote the decryption result $\mathsf{Dec}(\mathsf{ct}_{\mathrm{spm}}, \mathsf{sk})$ of the ciphertext $\mathsf{ct}_{\mathrm{spm}}$. If the correctness for the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ is satisfied, then we have

$$m_i \equiv d_H(T^{(i)}, P) \bmod t \text{ for } 0 \leq i \leq k - \ell.$$

In other words, the homomorphic operation (3) over packed ciphertexts gives us multiple Hamming distance values (1) for sufficiently large t.

Proof. As in the proof of Proposition 1, the homomorphic multiplication of the packed ciphertext $\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk})$ and the element C_ℓ (resp. $\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk})$ and C'_k) corresponds to the multiplication of two polynomials $m_{\mathrm{txt}}(T)$ and C_ℓ (resp. $m_{\mathrm{ptn}}(P)$ and C'_k) in the ring R_t. Furthermore, by a similar argument of the proof of Proposition 1, we see that the x^i-coefficient of the polynomial $m_{\mathrm{txt}}(T) \cdot C_\ell \in R_t$ (resp. $m_{\mathrm{ptn}}(P) \cdot C'_k$) is equal to the Hamming weight

$$\mathrm{HW}(T^{(i)}) \text{ (resp. } \mathrm{HW}(P)) \text{ for } 0 \leq i \leq k - \ell,$$

where $\mathrm{HW}(A)$ denotes the Hamming weight of a binary vector A. For two binary vectors A and B, the Hamming distance $d_H(A, B)$ can be computed by

$$\mathrm{HW}(A) + \mathrm{HW}(B) - 2\langle A, B \rangle.$$

By the above arguments and Proposition 1, we have that

$$\begin{aligned} m_i &\equiv \mathrm{HW}(T^{(i)}) + \mathrm{HW}(P) - 2\langle T^{(i)}, P \rangle \\ &\equiv d_H(T^{(i)}, P) \bmod t \end{aligned}$$

for $0 \leq i \leq k - \ell$ under the correctness for the ciphertext $\mathsf{ct}_{\mathrm{spm}}$. This completes the proof of Theorem 1. □

The result of Theorem 1 tells us that when we precompute two elements C_ℓ and C'_k, it only requires two homomorphic additions and three homomorphic multiplications to compute multiple Hamming distance values (1) (in particular, see the set S_{AMP} of approximate matching positions in Definition 1). Furthermore, in our implementation, we transform the homomorphic operation (3) for the efficiency, and only one time homomorphic multiplication enables us to compute multiple values (1) (see §4.2 below for details).

Remark 5. In the case $k = \ell$, the computation (3) in Theorem 1 gives us only a single Hamming distance value over packed ciphertexts, which computation is often used in biometric authentication to measure the similarity of two biometric feature vectors (see [4, 33] for work on secure biometric authentication). Please see [40, 41] for our work on an efficient computation of a single Hamming distance value in SHE schemes with our packing method proposed in this work. Furthermore, we can extend our method to compute multiple Euclidean distance values, but we will not refer to it in this paper due to lack of space.

3.1.3 Comparison with the Other Known Methods

Smart and Vercauteren in [37] propose the polynomial-CRT (Chinese Remainder Theorem) packing method, which is very useful to perform SIMD (Single Instruction - Multiple Data) operations on encrypted data (recently, Brakerski, Gentry and Halevi in [8] extend the SIMD notions to the standard LWE based scheme of [10] using the packing method of [35]). The basic idea of the polynomial-CRT packing method is as follows: If the cyclotomic polynomial $f(x)$ factorizes into r-factors modulo t such as

$$f(x) \equiv f_1(x) \times \cdots \times f_r(x) \bmod t,$$

then the plaintext space ring $R_t = \mathbb{F}_t[x]/(f(x))$ splits into a product of finite fields (note that the following map gives an isomorphism as rings, not only as $\mathbb{F}_t$-modules):

$$R_t \simeq \mathbb{F}_t[x]/(f_1(x)) \times \cdots \times \mathbb{F}_t[x]/(f_r(x)). \tag{4}$$

Since we pick $f(x)$ to be the cyclotomic polynomial $x^n + 1$ in this work, the number field $R \otimes_{\mathbb{Z}} \mathbb{Q} = \mathbb{Q}[x]/(f(x))$ is a Galois extension over $\mathbb{Q}$ and hence we have that $\deg f_1(x) = \ldots = \deg f_r(x) = d$ and $\mathbb{F}_t[x]/(f_1(x)) \simeq \cdots \simeq \mathbb{F}_t[x]/(f_r(x)) = \mathbb{F}_{t^d}$. This enables us to operate on r-elements in the field $\mathbb{F}_{t^d}$ in parallel. Concretely speaking, if we operate r-elements $m_1, \ldots, m_r$ with $m_i \in \mathbb{F}_{t^d}$, we first transform the r-fold plaintext vector $(m_1, \ldots, m_r) \in (\mathbb{F}_{t^d})^r$ to an element $m \in R_t$ by the isomorphism (4) using polynomial-CRT, and then encrypt $m \in R_t$ as described in §2.1.2, whose ciphertext is a packed one of the r-elements $m_1, \ldots, m_r$ (each element m_i

is called a "slot" in the work of [21, 22]). In particular, the polynomial-CRT packing method is used in the work [22] to evaluate the AES circuit homomorphically by the BGV scheme of [8].

Our packing method presented in this paper is essentially different from the polynomial-CRT method, and our method cannot be applied for SIMD operation. However, our method specializes in secure pattern matching computation, and it is much more efficient for multiple Hamming distance values (1) since our method does not need the polynomial-CRT transformation using the isomorphism (4). Furthermore, it would be more interesting by combining the polynomial-CRT method and ours, which would be our future work.

3.2 Our Protocol

Using the SHE scheme with our packing method, we can construct a protocol, which can be applied to the scenario of §1.2.1. Our protocol involves, a user having a binary text T of length k, a client having a binary pattern P of length ℓ, and finally the cloud (in the scenario of §1.2.1, the user can be considered as a patient, and the client as a doctor). In the protocol, the client would like to find the set S_{AMP} of approximate matching positions as described in Definition 1, with preserving privacy of both the text T and the pattern P. We use the cloud as both an outsourced data storage and a computation resource for multiple Hamming distance values (1). In the following, we give a protocol using the SHE scheme with our packing method for the above scenario (we assume $n \geq k \geq \ell \geq 1$ and $t \geq n$ for the sake of simplicity):

Setup Phase. For suitable parameters (n, q, t, σ) of the SHE scheme, the client generates the public key pk and the secret key sk as described in §2.1, and distributes only pk to the user (note that only the client knows sk).

Text Sending Phase. The user encrypts the binary text T of length k according to the type (A) of Definition 3, and only sends a pair

$$(\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}), k)$$

of the packed ciphertext $\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk})$ of the text and the text size k to the cloud.

Matching Phase.

1. The client encrypts the binary pattern P of length ℓ according to the type (B) of Definition 3, and only sends a pair

   $$(\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \ell)$$

   of the packed ciphertext $\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk})$ of the pattern and the pattern size ℓ to the cloud.

2. The cloud computes a ciphertext $\mathsf{ct}_{\mathrm{spm}}$ defined by the equation (3) from

   $$(\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}), k) \text{ and } (\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \ell),$$

   and sends the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ to the client.

3. The client decrypts the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ with sk, and obtain its decryption result $m = \sum_{i=0}^{n-1} m_i x^i \in R_t$. Finally, the client checks whether m_i is less than a predefined threshold τ for $0 \leq i \leq k - \ell$, and outputs the set

   $$S = \{0 \leq i \leq k - \ell \mid m_i \leq \tau\},$$

   which is equal to the set S_{AMP} by Theorem 1. As remarked in Remark 1 of §1.2.1, this protocol gives the client more information than matching positions.

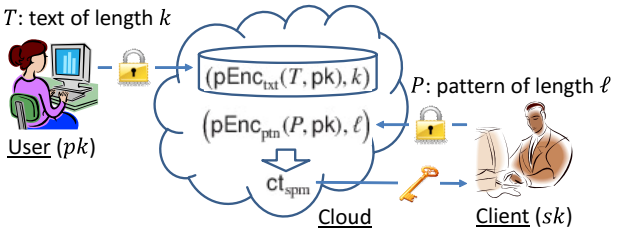

Figure 2: Our protocol for the scenario of §1.2.1

Feature of the protocol. The protocol constructed above has the following feature:

- Confidentiality to the cloud. The cloud does not learn any information about client's and user's data since all data in the cloud are in encrypted form and the cloud does not have the secret key.
- Availability. Since the protocol uses public key encryption, any user easily can send text data to the cloud with keeping the data confidentiality, and hence client can perform pattern matching computations among multiple users.

4. IMPLEMENTATION

In this section, we describe our choice of parameters of the SHE scheme, and the details of our implementation of the scheme with our packing method.

4.1 Parameters of the SHE Scheme

According to an application, key parameters (n, q, t, σ) of the SHE scheme should be appropriately chosen in order to achieve both the enough security and the correctness. In this work, we need to evaluate secure pattern matching computation (3) in Theorem 1 homomorphically. The method to choose parameters in this section is based on the work of [28] (see [28, Appendix A] for details of their method). For the sake of simplicity, we assume $n \geq k \geq \ell \geq 1$, and only consider the case where we pack a binary text of length k and a binary pattern of length ℓ in a single ciphertext with our packing method.

Correctness. For the correctness of the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ given by the equation (3), we need to satisfy the condition

$$||\langle \mathsf{ct}_{\mathrm{spm}}, \vec{s} \rangle||_\infty < \frac{q}{2} \tag{5}$$

by Lemma 2. Furthermore, it follows from the correctness of the SHE scheme that the element $\langle \mathsf{ct}_{\mathrm{spm}}, \vec{s} \rangle$ is equal to

$$\langle \mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}), \vec{s} \rangle \cdot C_\ell + \langle \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \vec{s} \rangle \cdot C'_k$$
$$-2 \langle \mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}), \vec{s} \rangle \cdot \langle \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \vec{s} \rangle$$

in the ring R_q (see also the proof of [28, Lemma 3.3]). When we set U to be an upper bound of the ∞-norm size $||\langle \mathsf{ct}, \vec{s}\rangle||_\infty$ for any fresh ciphertext $\mathsf{ct} \in (R_q)^2$, the above equation on the element $\langle \mathsf{ct}_{\mathrm{spm}}, \vec{s}\rangle$ gives an inequality

$$|| \langle \mathsf{ct}_{\mathrm{spm}}, \vec{s}\rangle ||_\infty \leq 2nU + 2nU^2 \tag{6}$$

by the fact that $||C'_k||_\infty = ||C_\ell||_\infty = 1$ and

$$|| \langle \mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk}), \vec{s}\rangle ||_\infty, || \langle \mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk}), \vec{s}\rangle ||_\infty \leq U.$$

Note that we also use the fact that we have

- $||a + b||_\infty \leq ||a||_\infty + ||b||_\infty$, and
- $||a \cdot b||_\infty \leq n \cdot ||a||_\infty \cdot ||b||_\infty$

for any two elements $a, b \in R_q$. As in the work of [28], we take U to be the value $2t\sigma^2\sqrt{n}$, which is an experimental estimation (see the proof of [28, Lemma 3.3] for details). Then we have

$$||\langle \mathsf{ct}_{\mathrm{spm}}, \vec{s}\rangle||_\infty \leq 2nU + 2nU^2 \approx 8n^2t^2\sigma^4$$

by the inequality (6). Therefore, by the inequality (5), we estimate that the correctness for the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ is satisfied if

$$16n^2t^2\sigma^4 < q, \tag{7}$$

which condition gives a lower bound of the prime q for the correctness.

Remark 6. The authors in [22] give an alternative analysis on the upper bound U. In [22], they estimate that a product of two polynomials randomly chosen with Gaussian distributions of standard deviation σ_a, σ_b, respectively, has the ∞-norm size at most $16\sigma_a\sigma_b$ with probability greater than $1 - 2^{-50}$, since the probability that both coefficients exceed 4 times of their standard deviation is roughly 2^{-50} due to the value $\mathrm{erfc}(4) \approx 2^{-25}$ of complementary error function. According to their estimation, we have

$$U = 2 \cdot 16\sigma^2 \cdot t = 32t\sigma^2,$$

which is smaller than $2t\sigma^2\sqrt{n}$ if $n \geq 256$. Since we take $n \geq 2048$ in our implementation, the upper bound $U = 2t\sigma^2\sqrt{n}$ used in this paper has larger margin for the correctness of the SHE scheme.

Security. By Lemma 1, the security of the scheme relies on the polynomial LWE assumption $\mathsf{PLWE}_{n,q,\chi}$ in Definition 2. The security analysis in the work of [28] is based on the methodology of Lindner and Peikert in [29] for the general LWE problem. As in the work of [28], we still use their security analysis. According to [29, Section 4], there are two efficient attacks against the general LWE problem, namely,

- the distinguishing attack of [32], and
- the decoding attack proposed by [29].

The analysis of [29] shows that the decoding attack is always better than the distinguishing attack, but the two attacks seem to have a similar performance for practical advantages $\varepsilon = 2^{-32}$ and 2^{-64}. Therefore we only consider the security against the distinguishing attack as in the work of [28].

The security of lattice-based cryptographic schemes can be measured by the root Hermite factor. According to the analysis of [29] on the distinguishing attack, for given parameters (n, q, t, σ) of the SHE scheme, we have the relation (see also [28, Appendix A])

$$c \cdot q/\sigma = 2^{2\sqrt{n \cdot \lg(q) \cdot \lg(\delta)}} \tag{8}$$

between n, q and δ, where c is the constant determined by the attack advantage ε, and we assume $c = 3.758$ corresponding to $\varepsilon = 2^{-64}$ for higher security than the work of [28], in which $\varepsilon = 2^{-32}$ is considered.

Chosen parameters. As in the work of [28], we set $\sigma = 8$ to make the scheme secure against combinatorial style attacks as noted in [16, Appendix D] (cf. the authors in [22] used $\sigma \approx 4$ for evaluating the AES circuit homomorphically in the leveled FHE scheme of [8]). We also set $t = n$ as the modulus parameter of the plaintext space, which is enough to compute multiple Hamming distance values (1) for secure pattern matching of a binary text of length less than n. For obtaining various security levels of the scheme, we take four lattice dimension parameters $n = 2048, 4096, 8192$ and 16384 (as the lattice dimension is larger, the security of the scheme becomes higher). Then for each lattice parameter n, the parameter q is determined by the condition (7), and then the root Hermite factor δ is computed by the relation (8).

Example 1. When we take $n = 2048$, the condition (7) tells us that the correctness for the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ is satisfied if $q > 2^{60}$, and hence we take a prime q of 61-bit. Furthermore, for given $(n, q, \sigma) = (2048, 61\text{-bit}, 8)$, the root Hermite factor $\delta = 1.00499$ is computed by the relation (8).

As a summary, we take the following four parameters of the SHE scheme for secure pattern matching computation (3) in Theorem 1:

(i) $(n, q, t, \sigma) = (2048, 61\text{-bit}, 2048, 8)$, $\delta = 1.00499$,

(ii) $(n, q, t, \sigma) = (4096, 65\text{-bit}, 4096, 8)$, $\delta = 1.00266$,

(iii) $(n, q, t, \sigma) = (8192, 69\text{-bit}, 8192, 8)$, $\delta = 1.00141$,

(iv) $(n, q, t, \sigma) = (16384, 73\text{-bit}, 16384, 8)$, $\delta = 1.00075$.

According to the state-of-the-art security analysis of Chen and Nguyen [11] for lattice-based cryptographic schemes, a root Hermite factor smaller than $\delta = 1.0050$ is estimated to have more than 80-bit security level with an enough margin. Therefore the above four parameters are estimated to have 80-bit security level against the distinguishing attack with advantage $\varepsilon = 2^{-64}$, and also against the more powerful decoding attack. In particular, note that the parameter (i) is similar as the parameter $(n, q, t, \sigma) = (2048, 58\text{-bit}, 1024, 8)$ with $\delta = 1.0046$ included in [28, Table 1 in Section 5], which is estimated to have more than 120-bit security level according to the security analysis of [29] (see also the below remark).

Remark 7. Different from the work of [11], the authors in [29] simply estimate the running times for the NTL implementation of the BKZ algorithm, which is one of the most practical lattice reduction algorithms against lattice problems. They also derive a relation of the expected running time t_{Adv} (as in [29]) of the distinguishing attack with the root Hermite factor δ as follows:

$$\lg(t_{\mathsf{Adv}}) = \frac{1.8}{\lg(\delta)} - 110. \tag{9}$$

For chosen parameters (i)-(iv) of the SHE scheme, we give the expected running time t_{Adv} computed by the relation (9) in Table 2. However, their security analysis seems no longer state-of-the-art due to the old NTL implementation. Therefore we remark that the data in Table 2 are just at the reference level, but the data tell us a rough standard of the security level of each parameter. In particular, according to Table 2, the parameter (iv) is estimated to have much more than 1000-bit security level against the distinguishing attack with advantage $\varepsilon = 2^{-64}$.

Table 2: The running time t_{Adv} of the distinguishing attack computed by the relation (9)

	δ	t_{Adv}
(i)	1.00499	140-bit
(ii)	1.00266	400-bit
(iii)	1.00141	775-bit
(iv)	1.00075	1554-bit

4.2 Implementation Results

For the above four parameters (i)-(iv), we implemented the SHE scheme with our packing method for secure pattern matching computation (3) in Theorem 1. Note that we have four parameters of the scheme in order to correspond various security levels. Our experiments ran on an Intel Xeon X3480 at 3.07 GHz with 16 GB memory, and we used our software library using inline assembly language x86_64 in C programs for all computations in the base ring $R_q = \mathbb{F}_q[x]/(x^n + 1)$ of the ciphertext space. Furthermore, our C code was complied using gcc 4.6.0 on Linux. In particular, for efficient multiplication in the ring R_q, we implemented the Montgomery reduction algorithm, and

- the Karatsuba multiplication algorithm for the parameter (i),
- the multiplication algorithm using the FFT (Fast Fourier Transform) method for parameters (ii)-(iv).

Actually, our experiments shows that the Karatsuba method is about twice faster than the FFT method for the parameter (i). On the other hand, the FFT method is faster than the Karatsuba method for the parameters (ii)-(iv), and in particular, it is about 5 times faster for the largest parameter (iv).

In Table 3, we summarize the performances and the sizes of the SHE scheme with our packing method. As described in §3.1, our packing method can pack a binary vector of length less than n in a single ciphertext, and the computation (3) in Theorem 1 enables us to compute multiple Hamming distance values (1) over packed ciphertexts, whose values are required for secure exact and approximate pattern matching as described in Definition 1. Furthermore, for the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ of secure pattern matching, we implemented the homomorphic operation

$$-\frac{1}{2}\{\big(2\mathsf{pEnc}_{\mathrm{txt}}(T,\mathsf{pk}) - C'_k\big) * \big(2\mathsf{pEnc}_{\mathrm{ptn}}(P,\mathsf{pk}) - C_\ell\big) - C'_k * C_\ell\} \in (R_q)^3, \qquad (10)$$

which is equal to the equation (3) in Theorem 1. While the computation (3) requires two homomorphic additions and three homomorphic multiplications, the computation (10) mainly requires only one homomorphic multiplication when we precompute two elements C_ℓ, C'_k and the element $C'_k * C_\ell$, which only require considerably lower cost than the homomorphic multiplication. We also note that in our implementation, we did not use the relinearization technique described in [28, Section 3.2.3], which reduces ring elements of a ciphertext from three to two elements.

We describe details of the performances and the sizes only for the parameter (i) in Table 3 (see [28, Section 1.2 or Section 5] for a comparison). Note that it gives us the better performance than the other parameters (ii)-(iv) since the prime parameter q is smaller than 64-bit and hence it gives the fastest performance in our implementation.

- The size of the public key $\mathsf{pk} = (a_0, a_1) \in R_q^2$ is $2n \cdot \lg(q) \approx 31.2$ KB, and the size of the secret key $\mathsf{sk} = s \in R_q$ is $n \cdot \lg(q) \approx 15.6$ KB. A fresh ciphertext has two elements in the ring R_q, and hence its size is $2n \cdot \lg(q) \approx 31.2$ KB. Therefore the size of packed ciphertexts $\mathsf{pEnc}_{\mathrm{txt}}(T, \mathsf{pk})$ and $\mathsf{pEnc}_{\mathrm{ptn}}(P, \mathsf{pk})$ is about 31.2 KB, respectively, irrespective of the text size k and the pattern size ℓ if $k, \ell < n$. In contrast, the ciphertext $\mathsf{ct}_{\mathrm{spm}}$ has three ring elements, and its size is about 46.8 KB.
- The key generation (excluding the prime generation) ran in about 1.89 ms, the packed encryption of a text or a pattern of length less than $n = 2048$ took about 3.65 ms (see Definition 3), the secure pattern matching computation (3) took about 5.31 ms, and finally the decryption took about 3.47 ms. Hence the total time is about

  $$3.65 + 5.31 + 3.47 = 12.43 \text{ ms}$$

  for secure pattern matching of length less than $n = 2048$.
- For further information, it took about 0.0016 ms (4,837 clock cycles) to compute one polynomial addition, and it also took about 1.56 ms (4,793,850 clock cycles) to compute one polynomial multiplication in the ring R_q. We remark that performances of the homomorphic addition and the homomorphic multiplication depend on the number of ring elements in ciphertexts.

5. CONCLUSIONS AND FUTURE WORK

For basic secure pattern matching, we proposed a packing method to efficiently compute multiple Hamming distance values (1) on encrypted data in the SHE scheme of [28]. Our implementation showed that our packing method has the advantage on both the performance and the communication cost compared to the state-of-the-art work including the MPC approach. Furthermore, for secure pattern matching of several thousands length, our method gives very practical performance and it would be useful in real life.

Since our packing method specializes in secure pattern matching computations, our future work is to extend it for various computations. As described in §3.1.3, it would also be technically interesting to combine our packing method with the polynomial-CRT method in order to perform SIMD operations as in the work of [22]. Furthermore, unlike this work, we would like to use our packing method in homomorphic symmetric-key encryption such as the scheme presented

Table 3: The performances and the sizes of the SHE scheme with our packing method for secure pattern matching computation (3) in Theorem 1 (our packing method can pack a binary vector of length n in a single ciphertext, and the computation (3) over packed ciphertexts enables us to compute multiple Hamming distance values (1), which are required for pattern matching problems of length less than n in Definition 1)

Parameters (n, q, t, σ)	Ciphertext space ring R_q addition	multiplication	Packed encryption	Secure pattern matching (3)	Decryption	Total time
(i) (2048, 61-bit, 2048, 8) $\delta = 1.00499$	0.0016 ms	1.56 ms	3.65 ms (31.2 KB)	5.31 ms (46.8 KB)	3.47 ms	12.43 ms
(ii) (4096, 65-bit, 4096, 8) $\delta = 1.00266$	0.034 ms	11.09 ms	23.03 ms (66.6 KB)	34.34 ms (99.8 KB)	22.17 ms	79.54 ms
(iii) (8192, 69-bit, 8192, 8) $\delta = 1.00141$	0.055 ms	23.17 ms	48.07 ms (141.3 KB)	71.25 ms (212.0 KB)	46.35 ms	165.67 ms
(iv) (16384, 73-bit, 16384, 8) $\delta = 1.00075$	0.109 ms	51.97 ms	107.25 ms (299.0 KB)	159.45 ms (448.5 KB)	103.94 ms	370.64 ms

in [9, Section 3.1] (it is expected to be much more efficient than this work using homomorphic public-key encryption), and to apply it to private outsourced database searching applications as in recent work of [1, 5].

6. ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their useful comments. The authors also thank Melissa Chase for her helpful comments to improve the paper.

7. REFERENCES

[1] M.J. Atallah and K.B. Frikken, "Securely outsourcing linear algebra computations", In ACM Symposium on Information, Computer and Communication Security–ASIACCS 2010, ACM, 48-59, 2010.

[2] P. Baldi, R. Baronio, E. De Crisofaro, P. Gasti and G. Tsudik, "Countering gattaca: efficient and secure testing of fully-sequenced human genomes", In ACM Conference on Computer and Communications Security–CCS 2011, ACM, 691-702, 2011.

[3] J. Baron, K. El Defrawy, K. Minkovich, R. Ostrovsky and E. Tressier, "5PM: secure pattern matching", IACR e-print 2012/565, available at http://eprint.iacr.org/2012/698.pdf, 2012 (a preliminary version was presented at Security and Cryptography for Networks–SCN 2012, Springer LNCS 7485, 222-240, 2012).

[4] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification", In European conference on Research in computer–ESORICS 2011, Springer LNCS 6879, 190-209, 2011.

[5] D. Boneh, C. Gentry, S. Halevi, F. Wang and D. Wu, "Private database queries using somewhat homomorphic encryption", In Applied Cryptography and Network Security–ACNS 2013, Springer LNCS 7954, 102-118, 2013.

[6] D. Boneh, E.J. Goh and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts", In Theory of Cryptography–TCC 2005, Springer LNCS 3378, 325-341, 2005.

[7] Z. Brakerski, C. Gentry and S. Halevi, "Packed ciphertexts in LWE-based homomorphic encryption", In Public Key Cryptography–PKC 2013, Springer LNCS 7778, 1-13, 2013.

[8] Z. Brakerski, C. Gentry and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping", In Innovations in Theoretical Computer Science–ITCS 2012, ACM, 309-325, 2012.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages", In Advances in Cryptology–CRYPTO 2011, Springer LNCS 6841, 505-524, 2011.

[10] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE", In Foundations of Computer Science–FOCS 2011, IEEE, 97-106, 2011.

[11] Y. Chen and P. Q. Nguyen, "BKZ 2.0: better lattice security estimates", In Advances in Cryptology–ASIACRYPT 2011, Springer LNCS 7073, 1-20, 2011.

[12] J.H. Cheon, J.-S. Coron, J. Kim, M.S. Lee, T. Lepoint, M. Tibouchi and A. Yun, "Batch fully homomorphic encryption over the integers", In Advances in Cryptology–EUROCRYPT 2013, Springer LNCS 7881, 315-335, 2013.

[13] Cloud Security Alliance (CSA), Security guidance for critical areas of focus in cloud computing, available at https://cloudsecurityalliance.org/csaguide.pdf, December 2009.

[14] J. -S. Coron, A. Mandal, D. Naccache and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public-keys", In Advances in Cryptology–CRYPTO 2011, Springer LNCS 6841, 487-504, 2011.

[15] R. Cramer, R. Gennaro and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme", In Advances in Cryptology–EUROCRYPT 1997, Springer LNCS 1462, 103-118, 1997.

[16] I. Damgård, V. Pastro, N. Smart and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption", In Advances in Cryptology–CRYPTO 2012, Springer LNCS 7417, 643-662, 2012.

[17] M. van Dijk, C. Gentry, S. Halevi and V. Vaikuntanathan, "Fully homomorphic encryption over

the integers," In Advances in Cryptology–EUROCRYPT 2010, Springer LNCS 6110, 24–43, 2010.

[18] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption", IACR e-print 2012/144, available at http://eprint.iacr.org/2012/144, 2012.

[19] C. Gentry, "Fully homomorphic encryption using ideal lattices", In Symposium on Theory of Computing–STOC 2009, ACM, 169-178, 2009.

[20] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme", In Advances in Cryptology–EUROCRYPT 2011, Springer LNCS 6632, 129-148, 2011.

[21] C. Gentry, S. Halevi and N. P. Smart, "Fully homomorphic encryption with polylog overhead", In Advances in Cryptology–EUROCRYPT 2012, Springer LNCS 7237, 465-482, 2012.

[22] C. Gentry, S. Halevi and N. P. Smart, "Homomorphic evaluation of the AES circuit", In Advances in Cryptology–CRYPTO 2012, Springer LNCS 7417, 850-867, 2012.

[23] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all partial information", In Symposium on Theory of Computing–STOC 1982, ACM, 365-377, 1982.

[24] C. Hazay and T. Toft, "Computationally secure pattern matching in the presence of malicious adversaries", In Advances in Cryptology–ASIACRYPT 2010, Springer LNCS 6477, 195-212, 2010.

[25] A. Jarrous and B. Pinkas, "Secure hamming distance based computation and its applications", In Applied Cryptography and Network Security–ACNS 2009, Springer LNCS 5536, 107-124, 2009.

[26] J. Katz and L. Malka, "Secure text processing with applications to private DNA matching", In ACM Conference on Computer and Communications Security–CCS 2010, ACM, 485-492, 2010.

[27] F. Kerschbaum and N. Oertel, "Privacy-preserving pattern matching for anomaly detection in RFID anti-counterfeiting", In Radio Frequency Identification: Security and Privacy Issues–RFIDSec 2010, Springer LNCS 6370, 124-137, 2010.

[28] K. Lauter, M. Naehrig and V. Vaikuntanathan, "Can homomorphic encryption be practical?", In ACM workshop on Cloud computing security workshop–CCSW 2011, ACM, 113-124, 2011.

[29] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption", In RSA Conference on Topics in Cryptology–CT-RSA 2011, Springer LNCS 6558, 319-339, 2011.

[30] A. Lopez-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," In Symposium on Theory of Computing–STOC 2012, ACM, 1219-1234, 2012.

[31] V. Lyubashevsky, C. Peikert and O. Regev, "On ideal lattices and learning with errors over rings", In Advances in Cryptology–EUROCRYPT 2010, Springer LNCS 6110, 1-23, 2010.

[32] D. Micciancio and O. Regev, "Worst-case to average-case reduction based on gaussian measures", SIAM J. Computing 37 (1), 267-302, 2007.

[33] M. Osadchy, B. Pinkas, A. Jarrous and B. Moskovich, "SCiFI - a system for secure face recognition", In IEEE Security and Privacy, IEEE Computer Society, 239-254, 2010.

[34] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes", In Advances in Cryptology–EUROCRYPT 1999, Springer LNCS 1592, pp. 223 - 238, 1999.

[35] C. Peikert, V. Vaikuntanathan and B. Waters, "A framework for efficient and composable oblivious transfer", In Advances in Cryptology–CRYPTO 2008, Springer LNCS 5157, 554-571, 2008.

[36] N. P. Smart and F. Vercauteren, Fully homomorphic encryption with relatively small key and ciphertext sizes, in: Public Key Cryptography - PKC 2010, Springer LNCS 6056, 420-443, 2010.

[37] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations", To appear in Designs, Codes and Cryptography, IACR e-print 2011/133, available at http://eprint.iacr.org/2011/133.pdf, 2011.

[38] J. R. Troncoso-Pastoriza, S. Katzenbeisser and M. Celik, "Privacy preserving error resilient DNA searching through oblivious automata", In ACM Conference on Computer and Communications Security–CCS 2007, ACM, 519-528, 2007.

[39] D. Vergnaud, "Efficient and secure generalized pattern matching via fast fourier transform", In International Conference on Cryptology in Africa–AFRICACRYPT 2011, Springer LNCS 6737, 41-58, 2011.

[40] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama and T. Koshiba, "Packed homomorphic encryption based on ideal lattices and its application to biometrics", In CD-ARES Workshop 2013 (Modern Cryptography and Security Engineering–MoCrySEn 2013), Springer LNCS 8128, 55-74, 2013.

[41] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama and T. Koshiba, "Practical packing method in somewhat homomorphic encryption", To be presented at International Workshop on Data Privacy Management–DPM 2013.