Information Security Journal: A Global Perspective

ISSN: 1939-3555 (Print) 1939-3547 (Online) Journal homepage: https://www.tandfonline.com/loi/uiss20

Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption

Masaya Yasuda

To cite this article: Masaya Yasuda (2017) Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption, Information Security Journal: A Global Perspective, 26:2, 85-103, DOI: 10.1080/19393555.2017.1293199

To link to this article: https://doi.org/10.1080/19393555.2017.1293199

Published online: 17 Mar 2017.

INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE
2017, VOL. 26, NO. 2, 85–103
http://dx.doi.org/10.1080/19393555.2017.1293199

Taylor & Francis
Taylor & Francis Group

Secure Hamming distance computation for biometrics using ideal-lattice and ring-LWE homomorphic encryption

Masaya Yasuda

Institute of Mathematics for Industry, Kyushu University, Kyushu, Fukuoka, Japan

ABSTRACT

With widespread development of biometrics, concerns about security and privacy are rapidly increasing. Homomorphic encryption enables us to operate on encrypted data without decryption, and it can be applied to construct a privacy-preserving biometric system. In this article, we apply two homomorphic encryption schemes based on ideal-lattice and ring-LWE (Learning with Errors), which both have homomorphic correctness over the ring of integers of a cyclotomic field. We compare the two schemes in applying them to privacy-preserving biometrics. In biometrics, the Hamming distance is used as a metric to compare two biometric feature vectors for authentication. We propose an efficient method for secure Hamming distance. Our method can pack a biometric feature vector into a single ciphertext, and it enables efficient computation of secure Hamming distance over our packed ciphertexts.

KEYWORDS

Biometric template protection; homomorphic encryption; packing method; privacy-preserving biometrics; secure hamming distance

1. Introduction

Homomorphic encryption is encryption with the property that it can support operations on encrypted data. This encryption provides a powerful tool to perform meaningful computations while preserving the data confidentiality. In particular, this encryption has been expected to be adopted in cloud computing. Depending on possible operations, we can classify homomorphic encryption schemes into three types. (i) *Additive or multiplicative schemes* can support only additions or multiplications on encrypted data (e.g., the Paillier scheme (Paillier, 1999) is an additive scheme). (ii) *Fully homomorphic encryption (FHE) schemes* can support arbitrary operations. Since Gentry's breakthrough (Gentry, 2009), a number of FHE schemes have been proposed (e.g., (Alperin-Sheriff & Peikert, 2014; Brakerski, Gentry, & Vaikuntanathan, 2012; Brakerski & Vaikuntanathan, 2011a, 2011b; Coron, Mandal, Naccache, & Tibouchi, 2011; Ducas & Micciancio, 2015; Gentry & Halevi, 2011; Gentry, Halevi, & Smart, 2012; Halevi & Shoup, 2014)), but current FHE schemes are yet impractical [it is reported by Ducas and Micciancio (2015) that it requires about half a second to refresh a noisy ciphertext]. (iii) *Somewhat homomorphic encryption (SHE) schemes* have been proposed as a building block for the FHE construction in the literature since Gentry's work. Although SHE can support only a limited number of both additions and multiplications, it is much faster and more compact than FHE. Hence it is now receiving attention in research on applications with SHE.

1.1. Biometrics and homomorphic encryption

Biometrics is identification of clients by their physical characteristics, such as fingerprints, iris, vein, and DNA. Compared to the commonly used ID/password authentication, it has the advantage that clients do not need to remember long and complex passwords. The use of biometrics is rapidly expanding. However, at the same time, concerns about security and privacy are also increasing. In biometrics, it is most important to protect *templates*, stored biometric data, because once they are leaked, templates can be neither revoked nor replaced. A number of template protection techniques have been proposed, and the basic approach is to transform raw templates by a certain conversion and register the transformed templates in a database. At present, there are three main

CONTACT Masaya Yasuda yasuda@imi.kyushu-u.ac.jp Institute of Mathematics for Industry, Kyushu University, 744 Motooka Nishi-ku, Kyushu, Fukuoka 819-0395, Japan.

approaches: (i) *feature transformation*, (ii) *biometric cryptosystem*, and finally, (iii) *homomorphic encryption approaches* [e.g., see Jain, Nandakumar, and Nagar, (2008)]. Each approach has both advantages and limitations. In the homomorphic encryption approach, biometric feature data are protected by encryption, and the similarity of two feature data is measured on encrypted data. Namely, the similarity calculation is performed on encrypted data, different from the conventional encryption approach in which only data on communication channels are encrypted, but the similarity calculation is performed on plaintexts after decryption. This approach also makes a biometric system "cryptographically secure" as long as the secret key is managed by a trusted party. There exist a number of algorithms to extract a feature vector from a biometric image. For example, from a scanned iris image, the algorithm of Daugman (2003) generates a binary vector of 2048 bits, called an iris vector. In 2013, it was announced by Fujitsu Laboratories Ltd. (2013) that it had developed a new extraction algorithm for matching feature vectors of 2048 bits from palm vein images. In Daugman (2003 and Fujitsu Laboratories Ltd. (2013), the Hamming distance is used to compare two extracted feature vectors for authentication. If the Hamming distance is smaller than a predefined threshold, then the authentication is acceptable.

1.2. Our contributions

In this article, we apply SHE schemes to privacy-preserving biometrics. At present, there are four SHE variants: ideal-lattice–based (Gentry, 2009; Gentry & Halevi, 2011; Smart & Vercauteren, 2010), integers-based (Cheon et al., 2013; Cheon & Stehl´E, 2015; Coron et al., 2011; Van Dijk, Gentry, Halevi, & Vaikuntanathan, 2010), (ring-)LWE-based (Brakerski et al., 2012; Brakerski & Vaikuntanathan, 2011a, 2011b), and finally, NTRU-based schemes (L´opez-Alt, Tromer, & Vaikuntanathan, 2012). Among the four variants, we focus on the ideal-lattice and ring-LWE schemes because both types of schemes have homomorphic correctness over the ring of integers of a cyclotomic field, which is useful to construct privacy-preserving protocols. Our main contributions are the following three points: (a) Given a degree parameter n, let $R = \mathbb{Z}[x]/(x^n + 1)$ denote the base ring of the ideal-lattice and the ring-LWE schemes. We propose a method for efficient, secure computation of the Hamming distance. Our method transforms a biometric feature vector into a certain polynomial of R, and encrypts the polynomial to pack the feature vector into a single ciphertext. The homomorphic correctness over R enables us efficiently to compute a secure Hamming distance by a few homomorphic operations over our packed ciphertexts. (b) We select suitable key parameters of the ideal-lattice and the ring-LWE schemes for secure Hamming distance of 2,048-bit feature vectors. By our implementation, we compare performance and ciphertext sizes of the two schemes. We also give a comparison with related work in the homomorphic encryption approach, and show that our packing method over the SHE schemes gives faster performance and shorter ciphertext sizes than the state-of-the-art prior work. (c) Moreover, we apply our packing method to construct a challenge-response protocol over the ring-LWE scheme (it is hard to construct such a protocol over the ideal-lattice scheme). Our challenge-response protocol has sufficient security against replay attacks, and it also has practical performance.

Notation. The symbols $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$ denote the ring of integers, the field of rational numbers, the field of real numbers, and the field of complex numbers, respectively. The finite field with p elements is denoted by $\mathbb{F}_p$. For integers z and d, let $[z]_d$ denote the reduction of z modulo d in $[-d/2, d/2)$ (cf. $z(\mathrm{mod} d)$denotes the reduction in$[0, d)$). Denote by $\lfloor q \rceil$ the rounding of $q \in \mathbb{Q}$ to the nearest integer, and by $[q]$ the fractional part of q. These notations can be extended to vectors and matrices. For a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$, let $\| \mathbf{a} \|$ denote the Euclidean norm. Let $\| \mathbf{a}\|_1$ and $\| \mathbf{a}\|_\infty$ denote the 1-norm and the ∞-norm, respectively.

2. Preliminaries

2.1. Lattices

Given an integer n, let $\mathbf{B} \in \mathbb{R}^{n \times n}$ be a matrix and $\mathbf{b}_i \in \mathbb{R}^n$ denote its ith row for $1 \leq i \leq n$. We denote by

$$\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} m_i \mathbf{b}_i : m_i \in \mathbb{Z} \text{ for } 1 \le i \le n \right\}$$

the set of all integral linear combinations of the $\mathbf{b}_i$'s. The set $\mathcal{L}(\mathbf{B})$ is a subgroup of $\mathbb{R}^n$. We say that $\mathcal{L}(\mathbf{B})$ is a (full-rank) *lattice* of dimension n if all $\mathbf{b}_i$'s are linearly independent over $\mathbb{R}$. In this case, the matrix $\mathbf{B}$ is called a *basis* of the lattice. Every lattice has infinitely many lattice bases; If $\mathbf{B}_1$ and $\mathbf{B}_2$ are two bases, then there exists a unimodular matrix $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ satisfying $\mathbf{B}_1 = \mathbf{U} \times \mathbf{B}_2$. Since $\det(\mathbf{U}) = \pm 1$, the absolute value $|\det(\mathbf{B})|$ is invariant for any basis $\mathbf{B}$ and it is denoted by $\det(L)$. Given a basis $\mathbf{B}$, we let

$$\mathcal{P}(\mathbf{B}) = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in [-1/2, 1/2) \text{ for } 1 \le i \le n \right\}$$

denote its associated half-open parallelepiped. Then the volume of $\mathcal{P}(\mathbf{B})$ precisely equals to $\det(L)$. Every lattice L has a unique Hermite normal form basis $\mathrm{HNF}(L) = (b_{ij}) \in \mathbb{R}^{n \times n}$, where $b_{ij} = 0$ for all $i < j$, $b_{jj} > 0$ for all j, and $b_{ij} \in [-b_{jj}/2, b_{jj}/2)$ for all $i > j$. Given any basis of L, the basis $\mathrm{HNF}(L)$ is easily computable by Gaussian elimination.

2.2. Lattice reduction and Hermite factor

Given a lattice basis, lattice reduction aims to output a new basis $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n]^t$ of L with short and nearly orthogonal vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. Lattice reduction gives a powerful tool to break lattice-based cryptosystems. The *Hermite factor* δ of a lattice reduction algorithm is defined by

$$\delta = \| \mathbf{b}_1 \| / \det(L)^{1/n}$$

with the output basis $[\mathbf{b}_1, \ldots, \mathbf{b}_n]^t$ (assume that $\mathbf{b}_1$ is the shortest among the $\mathbf{b}_i$'s). This factor gives an index to measure the output quality of the algorithm. In particular, the output quality becomes better as δ is smaller. Here we introduce two practical lattice reduction algorithms: (i) LLL is a polynomial-time algorithm (Lenstra, Lenstra, & Lov´Asz, 1982). Gama-Nguyen's experimental results (Gama & Nguyen, 2008, Fig. 4) show that the Hermite factor of LLL is practically 1.02^n on average in high dimension $n \ge 100$. (ii) BKZ is a blockwise generalization of LLL with subexponential complexity (Schnorr & Euchner, 1994; Schnorr & Hörner, 1995). At present, no good upper bound on the complexity is known. BKZ uses a blockwise parameter κ, and larger κ improves the output quality but increases the running time. In practice, $\kappa \approx 20$ can achieve the best time/quality compromise. It follows from Gama and Nguyen (2008, Sec. 5.2) that the Hermite factor with $\kappa = 20$ is practically 1.0128^n on average. Currently, BKZ 2.0 is the state-of-the-art implementation with recent techniques by Gama, Nguyen, and Regev (2010). Chen and Nguyen (2011) analyzed the behavior of BKZ 2.0 with $\kappa \ge 50$, with which we can predict both the output quality and the running time for large κ.

3. Homomorphic encryption schemes

In this section, we present the construction of the ideal-lattice and the ring-LWE schemes, and introduce their homomorphic correctness. For a 2-power integer n, let $R = \mathbb{Z}[x]/(f_n(x))$ denote the polynomial ring modulo $f_n(x) = x^n + 1$. Note that $f_n(x)$ is irreducible over $\mathbb{Q}$ if and only if n is of 2-power. Since a map

$$R \to \mathbb{Z}^n, v(x) = v_0 + \cdots + v_{n-1} x^{n-1} \mapsto \mathbf{v} = (v_0, \ldots, v_{n-1}) \tag{1}$$

gives an isomorphism as $\mathbb{Z}$-modules, we can regard an element of R as both a polynomial $v(x) \in R$ and a vector $\mathbf{v} \in \mathbb{Z}^n$. Both ideal-lattice and ring-LWE schemes make use of R as a basic ring for construction. In the ideal-lattice scheme, we take an element $v(x) \in R$, called a *generating polynomial*, and consider the ideal lattice L given by

$$R \supset (v(x)) \simeq L \subset \mathbb{Z}^n \tag{2}$$

under the isomorphism (1), where let $(v(x))$ denote the principal ideal generated by $v(x)$. We take the rotation basis $\mathbf{V}$ of L as the secret key, and the Hermite normal form basis $\mathbf{B}$ as the public key ($\mathbf{V}$ and $\mathbf{B}$ are called good" and bad" bases of L, respectively). In contrast, the key generation of the ring-LWE scheme is simple. We first sample a small element s of R as the secret key. Given a prime q, we then take p_1 uniformly chosen from $R_q = R/qR$, and set (p_0, p_1) as the public key with

$p_0 = -(p_1 s + te)$, where let t denote the size of the plaintext and e a small element of R.

3.1. Construction of ideal-lattice scheme

Here we present the construction of the ideal-lattice scheme of Gentry-Halevi's implementation (Gentry & Halevi, 2011) of Gentry's original FHE scheme (Gentry, 2009). While Gentry-Halevi (Gentry & Halevi, 2011) present a scheme with 1-bit plaintext space, we give a scheme with general plaintext space. The following parameters are required for the construction:

- n = the degree parameter of 2-power
- α = the maximal bit length of coefficients of a generating polynomial $v(x)$
- β = the plaintext space size (cf. $\beta = 2$ is used in Gentry and Halevi (2011))

3.1.1. Key generation

Given randomly chosen integers v_i's with $|v_i| \leq 2^\alpha$, we take $\mathbf{v} = (v_0, \ldots, v_{n-1})$ to define a generating polynomial $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$. Take the ideal lattice L given by (2), and set $\mathbf{V} := \mathrm{rot}(\mathbf{v})$ as in Yasuda, Yokoyama, Shimoyama, Kogure, and Koshiba, (2014, Sec. 2.1) (the rows of $\mathbf{V}$ give a basis of L). By the extended Euclidean-GCD algorithm, we obtain $w(x) = \sum_{i=0}^{n-1} w_i x^i \in R$ satisfying

$$w(x) \times v(x) \equiv d (\mathrm{mod}\, f_n(x)), \tag{3}$$

where $d = \det(L)$ is the determinant. Set $\mathbf{W} := \mathrm{rot}(\mathbf{w})$ with $\mathbf{w} = (w_0, \ldots, w_{n-1})$, and it satisfies

$$\mathbf{W} \times \mathbf{V} = \mathbf{V} \times \mathbf{W} = d \cdot \mathbf{I}_n,$$

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. We give the following definition and lemma given by Gentry and Halevi (2011, Sec. 3).

Definition 1 [Goodness of $v(x)$]: *We say that $v(x)$ is good if the Hermite normal form basis $\mathbf{B} := \mathrm{HNF}(L)$ of L has the special form*

$$\mathbf{B} = \begin{pmatrix} d & 0 & 0 & \cdots & 0 \\ -r & 1 & 0 & \cdots & 0 \\ & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ & 0 & 0 & \cdots & 1 \end{pmatrix}. \tag{4}$$

Lemma 1: *A generating polynomial $v(x)$ is good if and only if L contains a vector of the form $(-r', 1, 0, \ldots, 0)$ for some r'. Moreover, if $v(x)$ is good, then*

$$\begin{cases} r & := w_1/w_0 \equiv w_2/w_1 \equiv \cdots \equiv -w_0/w_{n-1} (\mathrm{mod}\, d), \\ r^n & \equiv -1 (\mathrm{mod}\, d). \end{cases}$$

To check whether $v(x)$ is good, we test that $r := w_1/w_0 (\mathrm{mod}\, d)$ satisfies $r^n \equiv -1 (\mathrm{mod}\, d)$. Otherwise, we generate another $v(x)$. Set $\mathbf{V}, \mathbf{W}$ (resp. $\mathbf{B}$) as the secret key (resp. the public key). However, due to the form (4), we only need to set $\mathrm{SK} = w_i$ as the secret key and $\mathrm{PK} = (d, r, n, \beta)$ as the public key, where w_i is one entry of $\mathbf{w}$ with $\gcd(w_i, \beta) = 1$.

3.1.2. Encryption

To encrypt a plaintext $b \in \mathbb{Z}/\beta\mathbb{Z} = \{0, 1, \ldots, \beta - 1\}$ with $\mathrm{PK} = (d, r, n, \beta)$, we choose a noise vector $\mathbf{u} = (u_0, u_1, \ldots, u_{n-1})$ with $u_i \in \{0, \pm 1\}$ chosen as 0 with some probability P and as ± 1 with probability $(1 - P)/2$ each. Then a fresh" ciphertext (i.e., a non-operated ciphertext) of b is given by

$$Enc(b, \mathrm{PK}) = \left[b + \beta \sum_{i=0}^{n-1} u_i r^i \right]_d.$$

Set $\mathbf{a} := \beta \mathbf{u} + b\mathbf{e}_1 = (\beta u_0 + b, \beta u_1, \ldots, \beta u_{n-1})$ with $\mathbf{e}_1 = (1, 0, \ldots, 0)$, and let $a(x) \in R$ denote its corresponding polynomial under (1). Then $Enc(b, \mathrm{PK}) = [a(r)]_d$ and $(Enc(b, \mathrm{PK}), 0, \ldots, 0)$ equals to

$$\mathbf{a} (\mathrm{mod}\, \mathbf{B}) := \mathbf{a} - \left(\left\lceil \mathbf{a} \times \mathbf{B}^{-1} \right\rfloor \times \mathbf{B} \right) \in \mathrm{P}(\mathbf{B}).$$

Definition 2 (Masked plaintext): *We call the vector $\mathbf{a}$ [or the polynomial $a(x)$] the masked plaintext of a ciphertext.*

Remark 1. We should choose the probability P to make it hard to recover the noise vector $\mathbf{u}$ from a ciphertext. Against exhaustive-search and birthday attacks, we need to set P satisfying (i) $2^{(1-P)n} \cdot \binom{n}{Pn} > 2^{2\lambda}$ by (Gentry & Halevi, 2011, Sec. 5.2), where let λ denote the security parameter. Moreover, Gentry-Halevi (Gentry & Halevi, 2011) consider the hybrid attack, whose method is to choose a random subset of the powers of r including all the noise coefficients and search for a small vector in this

low-dimensional lattice (e.g., dimension 200). Then it is sufficient to set P satisfying (ii) $\left(\frac{n}{200}\right)^{Pn} \geq 2^{\lambda}$ against the hybrid attack. For $\lambda = 80$ and $n \geq 1024$, the two conditions (i) and (ii) are satisfied if we set $P = 1/3$, which we fix in this article.

3.1.3. Homomorphic operations

Given two ciphertexts $ct_1 = Enc(b_1, \mathrm{PK})$ and $ct_2 = Enc(b_2, \mathrm{PK})$, the homomorphic addition " $\dot{+}$ " and multiplication " $*$ " are defined by

$$\begin{cases} ct_1 \dot{+} ct_2 & := [ct_1 + ct_2]_d, \\ ct_1 * ct_2 & := [ct_1 \cdot ct_2]_d. \end{cases}$$

3.1.4. Decryption

To decrypt $Enc(b, \mathrm{PK})$ with secret key matrices $\mathbf{V}$ and $\mathbf{W}$, we first recover the masked plaintext $\mathbf{a}$ by computing

$$\begin{aligned} \mathbf{a} &= \mathbf{c} (\mathrm{mod} \mathbf{V}) \\ &= \mathbf{c} - \left(\lceil \mathbf{c} \times \mathbf{V}^{-1} \rfloor \times \mathbf{V} \right) = [\mathbf{c} \times \mathbf{W}/d] \times \mathbf{V} \end{aligned}$$

with $\mathbf{c} = (Enc(b, \mathrm{PK}), 0, \ldots, 0)$. It follows from Gentry and Halevi (2011, Sec. 6) that we can recover $\mathbf{a}$ if every entry in $\mathbf{a} \times \mathbf{W}$ is less than $d/2$ in absolute value. Then we output $b = a_0 (\mathrm{mod}\, \beta)$ for $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$.

In Gentry and Halevi (2011, Sec. 6.1), Gentry-Halevi propose an optimized decryption in the case $\beta = 2$. We can extend their method for general β. Let $\mathbf{a} = \beta \mathbf{u} + b\mathbf{e}_1$ be the masked plaintext of $Enc(b, \mathrm{PK})$. A similar argument in Gentry and Halevi (2011, Sec. 6.1) shows

$$[\mathbf{c} \times \mathbf{W}]_d = \mathbf{a} \times \mathbf{W} = \beta \mathbf{u} \times \mathbf{W} + b \cdot \mathbf{w}$$

if every entry in $\mathbf{a} \times \mathbf{W}$ is less than $d/2$ in absolute value. Since $[\mathbf{c} \times \mathbf{W}]_d$ is equal to

$$\left([Enc(b, \mathrm{PK}) \cdot w_0]_d, \ldots, [Enc(b, \mathrm{PK}) \cdot w_{n-1}]_d \right),$$

we have $[Enc(b, \mathrm{PK}) \cdot w_i]_d \equiv b \cdot w_i (\mathrm{mod}\ \beta)$ for any $0 \leq i \leq n-1$. It is sufficient to keep one entry w_i of $\mathbf{w}$ with $\gcd(w_i, \beta) = 1$ as SK, and then we can recover b by

$$b = [Enc(b, \mathrm{PK}) \cdot \mathrm{SK}]_d \cdot \mathrm{SK}^{-1} (\mathrm{mod}\ \beta). \tag{5}$$

Note that there always exists w_i satisfying $\gcd(w_i, \beta) = 1$ if we take $\beta = 2^k$ for some $k \geq 1$.

3.2. Security and correctness of ideal-lattice scheme

As described by Gentry and Halevi (2011), the security of the ideal-lattice scheme relies on the computational hardness of the following lattice problem:

Definition 3 (Bounded distance decoding problem, BDDP): Given a (public) basis $\mathbf{B}$ of a lattice L and a vector $\mathbf{c}$ close to a lattice point $\mathbf{b}$ of L, the problem is to find the lattice point $\mathbf{b}$. Given a parameter $\gamma > 1$, the problem γ-BDDP is to find the target lattice point $\mathbf{b}$ under

$$\mathrm{dist}(L, \mathbf{c}) := \min_{\mathbf{z} \in L} \{ \| \mathbf{z} - \mathbf{c} \| \} \leq \frac{\det(L)^{1/n}}{\gamma}.$$

Given two ciphertexts $Enc(b_1, \mathrm{PK})$ and $Enc(b_2, \mathrm{PK})$, let $\mathbf{a}_1$ and $\mathbf{a}_2$ denote the corresponding masked plaintexts. Then two operated ciphertexts,

$$\begin{cases} (Enc(b_1, \mathrm{PK}) \dot{+} Enc(b_2, \mathrm{PK}), 0, \ldots, 0), \\ (Enc(b_1, \mathrm{PK}) * Enc(b_2, \mathrm{PK}), 0, \ldots, 0), \end{cases}$$

are equal to $\mathbf{a}_1 + \mathbf{a}_2 (\mathrm{mod}\, \mathbf{B})$ and $\mathbf{a}_1 \times \mathbf{a}_2 (\mathrm{mod}\, \mathbf{B})$, respectively, where " $\times$ " is the multiplication over R. This shows that the homomorphic operations correspond to the ring structure of R (i.e., the scheme has homomorphic correctness over R). However, homomorphic operations make the size of the noise vector $\mathbf{u}$ of the corresponding masked plaintext $\mathbf{a}$ larger. That is, it is only possible to add and multiply ciphertexts before the size of the noise vector grows beyond the decryption range. Then the size of the decryption range is the most important to avoid decryption failure. Under $|v_i| \leq 2^{\alpha}$, Gentry and Halevi (2011, Sec. 7) estimate experimentally that the decryption range is roughly equal to 2^{α} and it succeeds to decrypt a ciphertext if the corresponding masked plaintext $\mathbf{a}$ satisfies $\| \mathbf{a} \| \leq 2^{\alpha}$. In contrast, the decryption range was studied theoretically by Yasuda et al. (2014), and we introduce the result (Yasuda et al., 2014, Corollary 3.2) as follows:

Theorem 1 (Theoretical decryption range): *Given a generating polynomial* $v(x) = \sum_{i=0}^{n-1} v_i x^i \in R$, *assume*

$$\begin{aligned} (\clubsuit) \ : \ T &= |v_{n-1}| = 2^{\alpha}(1 + \varepsilon_{n-1}) \text{ and} \\ v_i &= T\varepsilon_i \text{ for } 0 \leq i \leq n-2 \end{aligned}$$

for some constants ε_i *with* $0 \le i \le n-1$. *If* $0 < \varepsilon_{n-1} < \frac{1}{4n}$ *and* $|\varepsilon_i| < \frac{1}{4n}$ *for* $0 \le i \le n-2$, *then the decryption of a ciphertext succeeds if the corresponding masked plaintext* $\mathbf{a}$ *satisfies either*

$$\| \mathbf{a}\|_1 < \frac{11 \cdot 2^{\alpha-1}}{13} \quad \text{or} \quad \| \mathbf{a}\|_\infty < \frac{11n \cdot 2^{\alpha-1}}{19n-6}. \tag{6}$$

3.3. Construction of ring-LWE scheme

Here we give the construction of the ring-LWE scheme proposed by Brakerski and Vaikuntanathan (2011b) (strictly speaking, the security of Brakerski and Vaikuntanathan (2011b) relies on the polynomial-LWE assumption, which is a simplified version of the ring-LWE of Lyubashevsky, Peikert, and Regev (2010)). The construction requires the following four parameters [see also Yasuda, Shimoyama, Kogure, Yokoyama, and Koshiba (2015, Sec. 2)]:

- n = the degree parameter of 2-power
- q = a prime with $q \equiv 1 (\mathrm{mod} 2n)$, defining the ring $R_q = R/qR = \mathbb{F}_q[x]/(f_n(x))$ for the ciphertext space
- t = an integer with $t<q$, defining the ring $R_t = (\mathbb{Z}/t\mathbb{Z})[x]/(f_n(x))$ of the plaintext space
- σ = the parameter to define a discrete Gaussian distribution $\chi = D_{\mathbb{Z}^n,\sigma}$ (we select each entry in a vector of length n by sampling from a Gaussian distribution $N(0,\sigma)$, and then round it to the nearest integer)

3.3.1. Key generation

We choose an element $s \leftarrow \chi$ [then $s \in R$ by (1)]. We sample a uniformly random element $p_1 \in R_q$ and an error $e \leftarrow \chi$. 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.

3.3.2. Encryption

For a plaintext $m \in R_t$ and the public key $\mathrm{pk} = (p_0, p_1)$, we sample $u, f, g \leftarrow \chi$ and compute a fresh ciphertext by

$$\begin{aligned} Enc(m, \mathrm{pk}) &= (c_0, c_1) \\ &= (p_0 u + tg + m, p_1 u + tf), \end{aligned} \tag{7}$$

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

3.3.3. Homomorphic operations

While the encryption generates a ciphertext with only two ring elements of R_q, the homomorphic multiplication defined below makes the ciphertext length longer. Therefore we need to define homomorphic operations for ciphertexts of any length. Let $\mathrm{ct} = (c_0, \ldots, c_\xi)$, $\mathrm{ct}' = (c_{0'}, \ldots, c_{\eta'})$ be two ciphertexts. The homomorphic addition is computed by $ct \dot{+} ct' = (c_0 + c_{0'}, \ldots, c_{\max(\xi,\eta)} + c'_{\max(\xi,\eta)})$ by padding with zeros if necessary. The homomorphic subtraction is also computed by component-wise subtraction. The homomorphic multiplication is computed by $ct * ct' = (\hat{c}_0, \ldots, \hat{c}_{\xi+\eta})$, where all elements $\hat{c}_i$'s are determined by (z is just a symbolic variable)

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

3.3.4. Decryption

For a ciphertext $\mathrm{ct} = (c_0, \ldots, c_\xi)$, the decryption with $\mathrm{sk} = s$ is computed by $Dec(\mathrm{ct}, \mathrm{sk}) = [\tilde{m}]_q (\mathrm{mod}\, t) \in R_t$, where $\tilde{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$. For the secret key vector $\mathbf{s} := (1, s, s^2, \ldots)$, we can rewrite $Dec(ct, \mathrm{sk}) = [\langle ct, \mathbf{s} \rangle]_q (\mathrm{mod}\, t)$.

3.4. Security and correctness of ring-LWE scheme

The security of the ring-LWE scheme relies on the following assumption (the following assumption is a simplified version of the ring-LWE assumption of Lyubashevsky et al. (2010)):

Definition 4 (Polynomial-LWE assumption): *Given parameters* (n, q, t, σ), *the polynomial-LWE assumption* $\mathrm{PLWE}_{n,q,\chi}$ *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 $b_i = a_i s + e_i$.

Remark 2. The ring-LWE scheme is IND-CCA1 secure under the polynomial-LWE assumption. In contrast, the ideal-lattice scheme is proven to be not IND-CCA1 secure (Loftus, May, Smart, &

Vercauteren, 2012), but IND-CPA secure under that the ideal-coset problem (ICP) is hard (Gentry, 2009) (ICP is the decisional version of BDDP).

For $ct = (c_0, c_1)$ given by (7), we have

$$\begin{aligned}\langle ct, \mathbf{s}\rangle &= (p_0 u + tg + m) + s \cdot (p_1 u + tf) \\ &= m + t \cdot (g + sf - ue)\end{aligned} \tag{8}$$

over 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 $[\langle ct, \mathbf{s}\rangle]_q = m + t \cdot (g + sf - ue)$ over R. In this case, we can recover the correct plaintext m by mod t-operation, which shows the decryption mechanism for any *fresh* ciphertexts. For two fresh ciphertexts ct_1 and ct_2, we clearly have

$$\begin{cases} \langle ct_1 \dot{+} ct_2, \mathbf{s}\rangle &= \langle ct_1, \mathbf{s}\rangle + \langle ct_2, \mathbf{s}\rangle \\ \langle ct_1 * ct_2, \mathbf{s}\rangle &= \langle ct_1, \mathbf{s}\rangle \cdot \langle ct_2, \mathbf{s}\rangle \end{cases} \tag{9}$$

by the construction. The above equations imply that the homomorphic correctness over R holds. Specifically, we have

$$\begin{cases} Dec(ct \dot{+} ct', \mathrm{sk}) &= m + m' \\ Dec(ct * ct', \mathrm{sk}) &= m \times m' \end{cases} \tag{10}$$

where two ciphertexts ct, ct' correspond to plaintexts m, m', respectively. The correctness of the SHE scheme holds under the following condition:

Lemma 2 (Condition for successful decryption): *For a ciphertext ct, the decryption $Dec(ct, \mathrm{sk})$ recovers the correct plaintext if for the secret key vector $\mathbf{s}$ it satisfies*

$$\| \langle ct, \mathbf{s}\rangle \|_\infty < q/2.$$

In the inequality, we regard $\langle ct, \mathbf{s}\rangle$ as the element of R calculated by the right-hand sides of (8) and (9), and for $a(x) = \sum_{i=0}^{n-1} a_i x^i \in R$, let $\| a(x)\|_\infty = \max|a_i|$ denote the ∞-norm of its coefficient representation. Specifically, for a fresh ciphertext ct, the ∞-norm $\| \langle ct, \mathbf{s}\rangle\|_\infty$ is given by $\| m + t(g + sf - ue)\|_\infty$ with $t \in \mathbb{Z}, m \in R_t \subset R$ and $e, f, g, u, s \leftarrow \chi = D_{\mathbb{Z}^n, \sigma}$.

4. Secure Hamming distance

In this section, we give a method for efficient computation of secure Hamming distance. This method is applicable to privacy-preserving biometrics. Let $\mathbf{T} = (t_0, \ldots, t_{\ell-1})$ and $\mathbf{Q} = (q_0, \ldots, q_{\ell-1})$ denote template and queried binary vectors, respectively (e.g., set $\ell = 2048$). Take a 2-power integer $n \geq \ell$. Our idea for efficient computation is to transform $\mathbf{T}$ and $\mathbf{Q}$ into two elements $F_1(\mathbf{T})$ and $F_2(\mathbf{Q})$ of $R = \mathbb{Z}[x]/(x^n + 1)$, given by

$$F_1(\mathbf{T}) := \sum_{i=0}^{\ell-1} t_i x^i \text{ and } F_2(\mathbf{Q}) := -\sum_{j=0}^{\ell-1} q_j x^{n-j}. \tag{11}$$

Moreover, we encrypt $F_1(\mathbf{T})$ and $F_2(\mathbf{Q})$ over either the ideal-lattice or the ring-LWE scheme. Since both schemes have homomorphic correctness over the ring R, one homomorphic multiplication computes

$$\begin{aligned} F_1(\mathbf{T}) \times F_2(\mathbf{Q}) &= -\sum_{i=0}^{\ell-1} t_i q_i x^n + (\text{other terms}) \\ &= \langle \mathbf{T}, \mathbf{Q}\rangle + (\text{nonconstant ter}ms) \end{aligned} \tag{12}$$

on encrypted data due to $x^n = -1$ in R. This idea is derived from the *convolution* of polynomials. Then the inner product $\langle \mathbf{T}, \mathbf{Q}\rangle$ can be extracted from the constant term of the decryption result. This method can be applied to secure Hamming distance. In the following, we consider each case of the ideal-lattice and the ring-LWE schemes.

4.1. Ideal-lattice scheme case

As described in Section 3.1, the ideal-lattice scheme can encrypt an element of $\mathbb{Z}/\beta\mathbb{Z}$. In contrast, we give a method to encrypt vectors $\mathbf{T}$ and $\mathbf{Q}$. However, our method causes a problem that it is not straightforward to apply the optimized decryption (5). To solve the problem, we need to take a generating polynomial $v(x)$ satisfying a certain condition. Let us begin to give our packing method over the ideal-lattice scheme.

Definition 5 (Packed encryption over the ideal-lattice scheme): *Given key parameters (n, α, β) of the ideal-lattice scheme, we assume $n \geq \ell$.*

(1) A packed ciphertext $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ for $\mathbf{T} = (t_0, \ldots, t_{\ell-1})$ is given by

$$[F_1(\mathbf{T})(r)+\beta u_1(r)]_d=\left[\sum_{i=0}^{\ell-1}t_i r^i+\beta u_1(r)\right]_d \tag{13}$$

using $\mathrm{PK}=(d,r,n,\beta)$, where $u_1(x)$ is a noise polynomial and $F_1(\mathbf{T})(r)$ the value substituted $x=r$ for $F_1(\mathbf{T})$. Note that the masked plaintext of the packed ciphertext $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T},\mathrm{PK})$ is given by $F_1(\mathbf{T})+\beta u_1(x)\in R$.

(2) A packed ciphertext $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q},\mathrm{PK})$ for $\mathbf{Q}=(q_0,\ldots,q_{\ell-1})$ is given by

$$[F_2(\mathbf{Q})(r)+\beta u_2(r)]_d=\left[-\sum_{i=0}^{\ell-1}q_i r^{n-i}+\beta u_2(r)\right]_d \tag{14}$$

where $u_2(x)$ is a noise polynomial. As in the above case, the masked plaintext of $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q},\mathrm{PK})$ is given by $F_2(\mathbf{Q})+\beta u_2(x)\in R$.

Proposition 1 (Secure inner product): *Let* ct *be a ciphertext given by* $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T},\mathrm{PK})*Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q},\mathrm{PK})$. *Let* $\mathbf{a}=(a_0,\ldots,a_{n-1})\in R\simeq\mathbb{Z}^n$ *denote the masked plaintext corresponding to* ct. *Then* $a_0\equiv\langle\mathbf{T},\mathbf{Q}\rangle(\mathrm{mod}\,\beta)$.

Proof. The homomorphic correctness over R shows that $\mathbf{a}=(F_1(\mathbf{T})+\beta u_1(x))\times(F_2(\mathbf{Q})+\beta u_2(x))$ underthe isomorphism (1). Hence $\mathbf{a}\equiv F_1(\mathbf{T})\times F_2(\mathbf{Q})(\mathrm{mod}\,\beta)$. Furthermore it follows by (12) that the constant term of $F_1(\mathbf{T})\times F_2(\mathbf{Q})$ is equal to $\langle\mathbf{T},\mathbf{Q}\rangle$.

Let *ct* and $\mathbf{a}$ be as in Proposition 1. By Proposition 1, we only need to recover $a_0(\mathrm{mod}\beta)$ to obtain the inner product $\langle\mathbf{T},\mathbf{Q}\rangle$. However, we cannot apply the optimized decryption (5) for *ct* directly, since its masked plaintext $\mathbf{a}$ is not of the form $\beta\mathbf{u}+b\mathbf{e}_1$ for some plaintext b and noise vector $\mathbf{u}$. Then we give the following proposition:

Proposition 2 (Decryption of packed ciphertexts): *Let* ct *and* $\mathbf{a}$ *be as in Proposition 1. Given a generating polynomial* $v(x)=\sum_{i=0}^{n-1}v_i x^i\in R$, *we assume* $v_0\notin\beta\mathbb{Z}$ *and* $v_i\in\beta\mathbb{Z}$ *for* $i=1,\ldots,n-1$. *Let* $\mathbf{w}=(w_0,w_1,\ldots,w_{n-1})$ *be the vector generated in Section 3.1, and we take* w_0 *as* SK. *Then we can recover* $a_0(\mathrm{mod}\,\beta)$ *by computing*

$$a_0\equiv[\mathrm{ct}\cdot\mathrm{SK}]_d\cdot v_0\cdot d^{-1}(\mathrm{mod}\,\beta).$$

In particular, if we take $v_0\in 1+\beta\mathbb{Z}$, we can recover $a_0(\mathrm{mod}\,\beta)$ by computing $a_0\equiv[\mathrm{ct}\cdot\mathrm{SK}]_d(\mathrm{mod}\,\beta)$.

Proof. Let $\mathbf{V}=\mathrm{rot}(\mathbf{v})$ and $\mathbf{W}=\mathrm{rot}(\mathbf{w})$ be as in Section 3.1. For $\mathbf{c}=(\mathrm{ct},0,\ldots,0)$, a similar argument as in Section 3.1 shows that $[\mathbf{c}\times\mathbf{W}]_d\times\mathbf{V}=\mathbf{a}\times\mathbf{W}\times\mathbf{V}$, and hence we have (since $\mathbf{W}\times\mathbf{V}=d\cdot\mathbf{I}_n$)

$$\begin{aligned}&([\mathrm{ct}\cdot w_0]_d,\ldots,[\mathrm{ct}\cdot w_{n-1}]_d)\cdot\mathbf{V}\\&\quad=(da_0,\ldots,da_{n-1})\end{aligned} \tag{15}$$

if every entry in $\mathbf{a}\times\mathbf{W}$ is less than $d/2$ in absolute value. By comparing the first entry of the two vectors in (15), we have $[\mathrm{ct}\cdot w_0]_d\cdot v_0\equiv da_0(\mathrm{mod}\,\beta)$ since $v_i\in\beta\mathbb{Z}$ for $i=1,\ldots,n-1$. Hence we can recover $a_0(\mathrm{mod}\,\beta)$ by $a_0\equiv[\mathrm{ct}\cdot w_0]_d\cdot v_0\cdot d^{-1}(\mathrm{mod}\,\beta)$. If $v_0\in 1+\beta\mathbb{Z}$, then $v_0\cdot d^{-1}\equiv 1(\mathrm{mod}\,\beta)$ since $d\equiv v_0^n\equiv 1(\mathrm{mod}\,\beta)$ by the fact that $d=|\det(\mathbf{V})|$ and $v_i\in\beta\mathbb{Z}$ for $i=1,\ldots,n-1$. This completes the proof.

Now we apply our packing method to secure Hamming distance. Assume $n,\beta\geq\ell$. For both the decryption range of Theorem 1 and the successful decryption of Proposition 2, we should take a generating polynomial $v(x)$ satisfying

$$(\spadesuit):\ v_0\in 1+\beta\mathbb{Z},\ v_i\in\beta\mathbb{Z}\ \text{for}\ 1\leq i\leq n-1,\ \text{and}\ (\clubsuit).$$

We also take $\mathrm{SK}=w_0$ as in Proposition 2. The Hamming distance $d_H(\mathbf{T},\mathbf{Q})$ can be computed by $\mathrm{HW}(\mathbf{T})+\mathrm{HW}(\mathbf{Q})-2\langle\mathbf{T},\mathbf{Q}\rangle$, where let $\mathrm{HW}(\mathbf{A})$ denote the Hamming weight of a binary vector $\mathbf{A}$. Set

$$C_1:=\left[\sum_{i=0}^{n-1}r^i\right]_d\ \text{and}\ C_2:=[-C_1+2]_d,$$

which can be computed with $\mathrm{PK}=(d,r,n,\beta)$. We take

$$\begin{aligned}C_1(x):=\sum_{i=0}^{n-1}x^i\ \text{and}\ C_2(x):=-C_1(x)\\+2=1-\sum_{j=1}^{n-1}x^j\in R.\end{aligned} \tag{16}$$

From a similar argument as in the proof of Proposition 1, the homomorphic multiplication $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T},\mathrm{PK})*C_2$ [resp. $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q},\mathrm{PK})*C_1$] corresponds to the masked plaintext $(F_1(\mathbf{T})+$

$\beta u_1(x)) \times C_2(x)$ [resp. $(F_2(\mathbf{Q}) + \beta u_2(x)) \times C_1(x)$], whose constant term modulo β is equal to $\mathrm{HW}(\mathbf{T})$ [resp. $\mathrm{HW}(\mathbf{Q})$]. Then we obtain the following theorem.

Theorem 2 (Secure computation over ideal-lattice scheme): *The encrypted Hamming distance ct_H is given by*

$$C_2 * Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK}) \dot{+} C_1 * Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK}) \dot{+} \left(-2Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK}) * Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})\right). \tag{17}$$

By Proposition 2, we can recover $d_H(\mathbf{T}, \mathbf{Q})$ with SK by

$$d_H(\mathbf{T}, \mathbf{Q}) \equiv [\mathrm{ct}_H \cdot \mathrm{SK}]_d (\mathrm{mod}\, \beta) \tag{18}$$

if the masked plaintext $\mathbf{a}_H$ of ct_H is included in the theoretical decryption range of Theorem 1.

4.2. Ring-LWE scheme case

Compared with the ideal-lattice case, the ring-LWE scheme case is very simple (see Yasuda et al., 2015, Sec. 3) for details).

Definition 6 (Packed encryption over the ring-LWE scheme): *Given key parameters (n, q, t, s), assume $n \geq \ell$, where ℓ is the length of $\mathbf{T}$ and $\mathbf{Q}$. Let (pk, sk) denote the key pair of the ring-LWE scheme. We define two types of packed ciphertexts over the ring-LWE scheme as follows:*

(1) Using pk, a packed ciphertext for $\mathbf{T}$ is defined by

$$ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) := Enc(F_1(\mathbf{T}), \mathrm{pk}).$$

(2) In contrast, a packed ciphertext for $\mathbf{Q}$ is defined by

$$ct^{(2)}_{\mathrm{pack}}(\mathbf{Q}) := Enc(F_2(\mathbf{Q}), \mathrm{pk}).$$

Due to the homomorphic correctness (10) over R, we can efficiently compute secure Hamming distance as well as the ideal-lattice scheme case:

Theorem 3 (Secure computation over ring-LWE scheme): *The encrypted Hamming distance ct'_H is given by*

$$ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) * C_2(x) \dot{+} ct^{(2)}_{\mathrm{pack}}(\mathbf{Q}) * C_1(x) \dot{+} \left(-2ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) * ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})\right), \tag{19}$$

where two constant polynomials $C_1(x)$ and $C_2(x)$ are defined by (16). Let $m = \sum_{i=0}^{n-1} m_i x^i \in R_t$ denote the decryption result $Dec(ct'_H, \mathrm{sk})$. If the correctness for ct'_H is satisfied, then we have

$$m_0 \equiv d_H(\mathbf{T}, \mathbf{Q}) (\mathrm{mod}\, t).$$

Remark 3. Smart and Vercauteren (2010) proposed the polynomial-CRT (Chinese Remainder Theorem) packing method over ring-LWE schemes. It is useful to perform SIMD (Single Instruction–Multiple Data) operations on encrypted data. Their basic idea is as follows. If a polynomial $f(x)$ factorizes into r-factors modulo t such as $f(x) \equiv f_1(x) \times \cdots \times f_r(x) (\mathrm{mod}\, t)$, then the plaintext space ring $R_t = \mathbb{F}_t[x]/(f(x))$ splits as

$$R_t \simeq \mathbb{F}_t[x]/(f_1(x)) \times \cdots \times \mathbb{F}_t[x]/(f_r(x)). \tag{20}$$

Since we pick $f(x)$ to be the special polynomial $x^n + 1$ in this article, the cyclotomic field $R \otimes_{\mathbb{Z}} \mathbb{Q} = \mathbb{Q}[x]/(f(x))$ is a Galois extension over $\mathbb{Q}$ and hence we have

$$\begin{cases} \deg f_1(x) = \ldots = \deg f_r(x) = d, \text{ and} \\ \mathbb{F}_t[x]/(f_1(x)) \simeq \cdots \simeq \mathbb{F}_t[x]/(f_r(x)) = \mathbb{F}_{t^d}. \end{cases}$$

This enables us to operate on r-elements in the field $\mathbb{F}_{t^d}$ in parallel. Specifically, 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 (20), and then encrypt $m \in R_t$. The ciphertext is a packed one of the r-elements $m_1, \ldots, m_r$ [each element m_i is called a slot" by Gentry et al. (2012)]. In contrast, our packing method is essentially different from the CRT method, and hence our method cannot be applied to SIMD operation. However, our method specializes in computing secure inner product, and it is also efficient for secure Hamming distance since our method does not require the transformation (20).

4.3. Privacy-preserving biometric protocol

Here we present a protocol for privacy-preserving biometrics using our packing method. Our protocol involves three parties, a client device $\mathcal{CD}$, a

computation server $\mathcal{CS}$ having a database D, and an authentication server $\mathcal{AS}$. Assume that $\mathcal{AS}$ is a trusted party managing the secret key, and $\mathcal{CS}$ is semi-honest (i.e., *honest-but-curious*) and cannot collude with $\mathcal{AS}$. Our architecture is based on ; that of Hattori et al. (2012), which uses generic 2-DNF (disjunctive normal form) homomorphic encryption such as the BGN scheme (Boneh, Goh, & Nissim, 2005).

4.3.1. Setup phase

$\mathcal{AS}$ generates the key pair (PK, SK) [resp. (pk, sk)] of the ideal-lattice (resp. the ring-LWE) scheme. Then $\mathcal{AS}$ distributes PK (resp. pk) to both $\mathcal{CD}$ and $\mathcal{CS}$.

4.3.2. Enrollment phase

(1) $\mathcal{CD}$ extracts a template vector $\mathbf{T}$ from client's biometric data, encrypts $\mathbf{T}$ using our packing method of *the first type*, and sends the packed ciphertext $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ [resp. $ct^{(1)}_{\mathrm{pack}}(\mathbf{T})$] with client's ID to $\mathcal{CS}$.
(2) $\mathcal{CS}$ stores $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ [resp. $ct^{(1)}_{\mathrm{pack}}(\mathbf{T})$] in the database D as a template with client's ID.

4.3.3. Authentication phase

(1) As above, $\mathcal{CD}$ extracts a queried vector $\mathbf{Q}$ from client's biometric data, encrypts $\mathbf{Q}$ using our packing method of *the second type*, and sends the packed ciphertexts $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})$ [resp. $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$] with client's ID to $\mathcal{CS}$.
(2) $\mathcal{CS}$ extracts $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ [resp. $ct^{(1)}_{\mathrm{pack}}(\mathbf{T})$] corresponding to client's ID from the database D. Then $\mathcal{CS}$ computes the encrypted Hamming distance ct_H of $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ and $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})$ defined by (17) [resp. ct'_H of $ct^{(1)}_{\mathrm{pack}}(\mathbf{T})$ and $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$ defined by (19)], and sends ct_H (resp. ct'_H) to $\mathcal{AS}$.
(3) $\mathcal{AS}$ decrypts ct_H (resp. ct'_H) with SK (resp. sk) to obtain the Hamming distance $d_H(\mathbf{T}, \mathbf{Q})$. Finally, $\mathcal{AS}$ returns the authentication result OK' (resp. NG') if $d_H(\mathbf{T}, \mathbf{Q}) \leq \theta$ (resp. otherwise), where θ denotes a predefined threshold.

All data over $\mathcal{CS}$ are always protected by homomorphic encryption. In other words, $\mathcal{CS}$ cannot learn any information about biometric vectors $\mathbf{T}$ and $\mathbf{Q}$ (cf. in the conventional encryption approach, the Hamming distance computation is performed on plaintexts after decryption, and hence $\mathbf{T}$ and $\mathbf{Q}$ are revealed in $\mathcal{CS}$). Hence we hope that we could use the cloud as $\mathcal{CS}$ for outsourcing storage of templates and computation resources of secure Hamming distance. Since the method to encrypt $\mathbf{T}$ is asymmetric to the method for $\mathbf{Q}$ [e.g., $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ vs $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})$], our protocol is expected to have the advantage that our protocol is harder to spoof than conventional protocols. Specifically, if an adversary steals $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ from $\mathcal{D}$ and sends it to $\mathcal{CS}$ instead of $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})$ in the authentication phase, the authentication will fail with very high probability, since the decryption result is not equal to the Hamming distance between $\mathbf{T}$ and $\mathbf{T}$, and $\mathcal{AS}$ cannot obtain Hamming distance smaller than θ with high probability. The probability depends on FAR (False Acceptance Rate) and FRR (False Rejection Rate). From this discussion, our protocol is secure as long as $\mathcal{AS}$ manages the secret key correctly.

5. Choosing key parameters

In this section, we choose suitable key parameters of both the ideal-lattice and the ring-LWE schemes for the privacy-preserving biometrics protocol described in Section 4.3. We fix $\ell = 2048$ as the length of $\mathbf{T}$ and $\mathbf{Q}$ (this length is used in biometrics, as introduced in Section 1.1).

5.1. Ideal-lattice scheme case

5.1.1. Choosing key parameters

We begin to give key parameters (n, α, β) of the ideal-lattice scheme, suitable for the secure Hamming distance computation (17). Our chosen key parameters are

$$(n, \alpha, \beta) = (4096, 37, 2048). \tag{21}$$

The key parameters are estimated to have more than 80-bit security against exhaustive-search and birthday

attacks against a generating polynomial $v(x) \in R$ satisfying (♠). For example, the exhaustive-search attack takes $(2^{\alpha}/(4n \cdot \beta))^n = (2^{12})^{4096} \gg 2^{80}$.

Our method to choose (21) is as follows. Since $0 \leq d_H(\mathbf{T}, \mathbf{Q}) \leq 2048$, it is sufficient to set $\beta = 2048$ to avoid carry operations. To pack a vector of length $\ell = 2048$ into a single ciphertext by our packing method, we should take $n \geq 2048$, and we consider two cases, $n = 2048$ and 4096 (we finally choose $n = 4096$ for enough security). Let ct_H denote the encrypted Hamming distance given by (17). As in Theorem 2, let $\mathbf{a}_H$ denote the masked plaintext corresponding to ct_H. We let $\mathbf{a}_1$ and $\mathbf{b}_1$ denote the masked plaintexts corresponding to $Enc^{(1)}_{\text{pack}}(\mathbf{T}, \text{PK})$ and $Enc^{(2)}_{\text{pack}}(\mathbf{Q}, \text{PK})$, respectively. Let $\mathbf{a}_2$ and $\mathbf{b}_2$ denote the masked plaintexts corresponding to the multiplied ciphertexts $C_2 * Enc^{(1)}_{\text{pack}}(\mathbf{T}, \text{PK})$ and $C_1 * Enc^{(2)}_{\text{pack}}(\mathbf{Q}, \text{PK})$, respectively. Then it follows by (17) that we have $\mathbf{a}_H = \mathbf{a}_2 + \mathbf{b}_2 - 2\mathbf{a}_1 \times \mathbf{b}_1$. By evaluating the ∞-norm size of $\mathbf{a}_H$, we determine suitable α [for the probability $P = 1/3$, the ∞-norm evaluation of (6) is more suitable than the 1-norm]. For any two elements $\mathbf{a}, \mathbf{b} \in R$, we have

$$\begin{cases} \| \mathbf{a} + \mathbf{b}\|_{\infty} & \leq \| \mathbf{a}\|_{\infty} + \| \mathbf{b}\|_{\infty}, \\ \| \mathbf{a} \times \mathbf{b}\|_{\infty} & \leq n \cdot \| \mathbf{a}\|_{\infty} \cdot \| \mathbf{b}\|_{\infty}. \end{cases} \tag{22}$$

Since we consider two cases, $n = 2048$ and 4096, we have

$$\begin{aligned} \| \mathbf{a}_H\|_{\infty} &\leq \| \mathbf{a}_2\|_{\infty} + \| \mathbf{b}_2\|_{\infty} + 2n \cdot \| \mathbf{a}_1\|_{\infty} \cdot \| \mathbf{b}_1\|_{\infty} \\ &\leq 2n \cdot (2^{11} + 1) + 2n \cdot (2^{11} + 1)^2 \\ &\leq 2^{26} + 2^{35}, \end{aligned}$$

where $\| \mathbf{a}_1\|_{\infty}, \| \mathbf{b}_1\|_{\infty} \leq 2^{11} + 1$ and $\| \mathbf{a}_2\|_{\infty}, \| \mathbf{b}_2\|_{\infty} \leq n \cdot (2^{11} + 1)$ due to $\beta = 2^{11}$. By the ∞-norm evaluation of (6), we estimate that the decryption of ct_H succeeds if we take $\alpha \geq 37$, and then we fix $\alpha = 37$.

5.1.2. Security analysis against lattice reduction attack

By a lattice reduction algorithm with small Hermite factor, it is feasible to recover the plaintext from a ciphertext. This is called the *lattice reduction attack*. The security of the ideal-lattice scheme against the attack is based on the computational hardness of γ-BDDP (see Definition 3) with parameter (where d is the determinant)

$$\gamma = d^{1/n} \approx 2^{\alpha}. \tag{23}$$

Notice that γ is independent of the plaintext size β [security analysis in our previous work (Yasuda, Shimoyama, Kogure, Yokoyama, & Koshiba, 2013) is incorrect at this point], and the approximate value of d is given by Yasuda et al. (2014, Theorem 3.1). BDDP is intuitively the analog of unique-SVP (Shortest Vector Problem), and it can be solved by lattice reduction algorithms with the Hermite factor smaller than γ (see Gama & Nguyen, 2008, for details). Our chosen parameters (21) give $\gamma = 1.00628^n$ by (23) [cf. we have $\gamma = 1.01260^n$ if we choose $(n, \alpha, \beta) = (2048, 37, 2048)$]. Since the Hermite factor of LLL (resp. BKZ with block size $\kappa = 20$) is practically 1.02^n (resp. 1.0128^n) on average, BKZ with $\kappa = 20$ cannot break the case $n = 4096$ but $n = 2048$ (this is the reason why we select not $n = 2048$ but $n = 4096$). As described in Section 2.2, BKZ 2.0 is the state-of-the-art implementation of BKZ with large $\kappa \geq 50$. The approximate Hermite factor achieved by BKZ 2.0 is shown by Chen and Nguyen (2011, Table 2). According to Chen and Nguyen (2011, Table 2), the BKZ blocksize $\kappa \approx 216$ is needed to solve our chosen parameters (21) with $\gamma = 1.00628^n$ (the Hermite factor of BKZ 2.0 with $\kappa = 216$ is predicted as 1.006^n). The approximate running time of BKZ 2.0 is given by

$$z \times \dim \times e, \tag{24}$$

where z is the number of rounds, dim is the lattice dimension, and e is the approximate running time of the enumeration subroutine of block size κ (see Chen & Nguyen, 2011, for details). Since a few rounds are required to achieve enough performance of BKZ 2.0, we assume $z = 1$ for the sake of simplicity. As for the running time e, Chen and Nguyen (2011, Table 3) give an estimated upper bound on the cost of the enumeration subroutine by their experiments. From Chen and Nguyen (2011, Table 3), we have that the enumeration subroutine of BKZ 2.0 with $\kappa = 216$ is

Table 1. Performance and ciphertext sizes of the ideal-lattice and the ring-LWE schemes ("Total" denotes the authentication performance of Section 4.3).

Scheme	Key Gen.	Packed Enc.	Secure Comp.	Dec.	Total
Ideal-lattice (HF = 1.00628^n)	870 ms	19.89 ms (19 KBytes)†	9.05 ms* (19 KBytes)‡	9.08 ms	38.02 ms
Ring-LWE (HF = 1.00491^n)	1.89 ms	3.65 ms (31 KBytes)†	5.31 ms (46.5 KBytes)‡	3.47 ms	12.43 ms

HF = Hermite factor. *It is about twice faster than Yasuda et al. (2013) due to the transformation (28). †Denotes the size of the packed ciphertext $Enc^{(1)}_{\rm pack}(\mathbf{T}, {\rm PK})$ or $Enc^{(2)}_{\rm pack}(\mathbf{Q}, {\rm PK})$ [resp. $ct^{(1)}_{\rm pack}(\mathbf{T})$ or $ct^{(2)}_{\rm pack}(\mathbf{Q})$] over the ideal-lattice (resp. the ring-LWE) scheme. ‡Denotes the size of the encrypted Hamming distance ct_H or ct'_H.

Table 2. Comparison with previous work.

Work (vector size)	Performance of secure Hamming	Size increase rate* (ciphertext size)	Homomorphic encryption scheme
SCiFI (900-bit)	310 ms[(a)]	2048 times (230 KBytes)	Paillier-1024† (additive scheme)
ESORICS2011 (2048-bit)	150 ms[(b)]	1024 times (262 KBytes)	DGK-1024† (additive scheme)
KN13 (2048-bit)	58 s [(c)]	(total bandwidth 400 KBytes)	BGN(MNT curve) (SHE scheme)
Ours (2048-bit)	**5.31 ms**[(d)]	**About 120 times (31 KBytes)**	**ring-LWE-2048 (SHE scheme)**
Ours (2048-bit)	**9.05 ms**[(d)] [faster than Yasuda et al. (2013)]$^\#$	**About 80 times (19 KBytes)**	**ideal-lattice-4096 (SHE scheme)**

$^\#$Due to the transformation (28). *The ratio of (encrypted vector size)/(plain vector size), †Either **T** or **Q** can be protected by homomorphic encryption due to additive schemes. (*a*) On an 8 core machine of 2.6 GHz AMD Opteron processors with 1 GByte memory. (*b*) On an Intel Core 2 Duo 2.13 GHz with 3 GByte memory. (*c*) On an Intel 3.2 GHz with around 200 KByte memory usage. (*d*) On an Intel Xeon X3480 at 3.07 GHz with 16 GByte memory.

Table 3. Performance and sizes of ciphertexts in our challenge-response protocol.

Parameter	Challenge (29)	Response (30)	Comp. s (31)	Dec.	Total
(27) (HF = 1.00533^n)	3.65 ms (32.8 KBytes)	3.12 ms (32.8 KBytes)	5.31 ms (49.2 KBytes)	3.47 ms	15.55 ms

estimated at $2^{110}\sim 2^{130}$ nodes. Since it is also reported that one node requires about 200 clock cycles, we estimate that $e \geq 200 \times 2^{110} \approx 2^{117.6}$ clock cycles for BKZ 2.0 with $\kappa = 216$. Therefore we estimate that the running time of BKZ 2.0 with $\kappa = 216$ is about $2^{12} \times 2^{117.6} = 2^{129.6}$ cycles in dimension $n = 4096$. This shows that the key parameters (21) have more than 80-bit security (with enough margin) against the lattice reduction attack.

5.2. Ring-LWE scheme case

As in the ideal-lattice scheme case, we need to carefully select key parameters (n, q, t, σ) of the ring-LWE scheme for both enough correctness and security.

5.2.1. Correctness and security

Let ct'_H be the encrypted Hamming distance given by (19). By Lemma 2, the correctness of ct'_H requires satisfying $\| \langle ct'_H, \mathbf{s}\rangle\|_\infty < q/2$, where $\mathbf{s} = (1, s, s^2, \ldots)$ is the secret key vector. It follows by (9) that $\langle ct'_H, \mathbf{s}\rangle$ is equal to

$$\langle ct^{(1)}_{\rm pack}(\mathbf{T}), \mathbf{s}\rangle \cdot C_2(x) + \langle ct^{(2)}_{\rm pack}(\mathbf{Q}), \mathbf{s}\rangle \cdot C_1(x) - 2\langle ct^{(1)}_{\rm pack}(\mathbf{T}), \mathbf{s}\rangle \cdot \langle ct^{(2)}_{\rm pack}(\mathbf{Q}), \mathbf{s}\rangle.$$

Let U denote an upper bound of the ∞-norm size $\| \langle ct, \mathbf{s}\rangle\|_\infty$ for any fresh ciphertext $ct \in (R_q)^2$. Then the above expression gives $\| \langle ct'_H, \mathbf{s}\rangle\|_\infty \leq 2nU + 2nU^2$ by (22) and the fact that $\| C_1(x)\|_\infty = \| C_2(x)\|_\infty = 1$ and $\| \langle ct, \mathbf{s}\rangle\|_\infty \leq U$ for the two fresh ciphertexts $ct = ct^{(1)}_{\rm pack}(\mathbf{T})$ and $ct^{(2)}_{\rm pack}(\mathbf{Q})$. As in Naehrig, Lauter, and Vaikuntanathan (2011), we take U to be $2t\sigma^2\sqrt{n}$, which is the experimental estimation given in the proof of Naehrig et al. (2011, Lemma 3.3). Then we have $2nU + 2nU^2 \approx 8n^2t^2\sigma^4$. Therefore we estimate that the correctness for ct'_H is satisfied if

$$16n^2t^2\sigma^4 < q. \tag{25}$$

The security of the ring-LWE scheme relies on $\mathrm{PLWE}_{n,q,\chi}$ (Definition 4). According to Lindner and Peikert (2011, Sec. 4), there are two efficient attacks against the general LWE problem, namely, the distinguishing attack of Micciancio and Regev (2007), and the decoding attack proposed by Lindner

and Peikert (2011). The analysis of Lindner and Peikert (2011) shows that the decoding attack is always better than the distinguishing one, but the two attacks seem to have similar performance for practical advantages such as $\varepsilon = 2^{-32}$ and 2^{-64}. Therefore we only need to consider the security against the distinguishing attack as in Naehrig et al. (2011). According to the analysis of Lindner and Peikert (2011) on the distinguishing attack, for given key parameters (n, q, t, σ), we obtain the relation

$$c \cdot q/\sigma = 2^{2\sqrt{\lg(q)\cdot\lg(\delta)}} \tag{26}$$

between n, q and the targeted Hermite factor δ, where c is the constant determined by the attack advantage ε, and we assume $c = 2.657$ corresponding to $\varepsilon = 2^{-32}$ as in Naehrig et al. (2011).

5.2.2. Chosen key parameters and their security level

Here we give key parameters (n, q, t, σ). As in Naehrig et al. (2011), we take $\sigma = 8$ to make the ring-LWE scheme secure against the combinatorial attack. We also set $t = 2048$, which is enough to compute the Hamming distance between $\mathbf{T}$ and $\mathbf{Q}$ of length 2048. We also need $n \geq 2048$ in order to pack $\mathbf{T}$ or $\mathbf{Q}$ into a single ciphertext by our packing method (Definition 6). When we take $n = 2048$, the equality (25) implies that the correctness for ct'_H is satisfied if $q > 2^{60}$. Then let us fix

$$(n, q, t, \sigma) = (2048, 61 - bit, 2048, 8). \tag{27}$$

By (26), the Hermite factor is calculated as $\delta = 1.00491^n$. Then it requires block size $\kappa = 286$ to attack the key parameters (27) with BKZ 2.0 [according to Chen and Nguyen (2011, Table 2), this κ gives the Hermite factor 1.0050^n]. We see from Chen and Nguyen (2011, Table 3) that the approximate running time e of the enumeration subroutine of BKZ 2.0 with $\kappa = 286$ is estimated at $2^{175} \sim 2^{205}$ nodes. Then we estimate by (24) that it requires $2^{11} \times 2^{175} = 2^{186}$ cycles to attack the key parameters (27) by BKZ 2.0. Therefore the key parameters (27) can make the ring-LWE scheme to be more than 80-bit security with enough margin.

6. Experimental evaluation

In this section, we give detailed information about our experimental results.

6.1. Implementation results

In Table 1, we give implementation results on performance and ciphertext sizes of the ideal-lattice and the ring-LWE schemes. For our chosen key parameters (21) [resp. (27)], we implemented the ideal-lattice (resp. the ring-LWE) scheme with our packing method for secure Hamming distance. Our experiments ran on an Intel Xeon X3480 at 3.07 GHz with 16 GByte memory. We used our software library written in assembly language x86_64. In particular, we implemented Karatsuba multiplication and Montgomery reduction algorithms for efficient multiplication. Our software library with assembly-level optimizations has the advantage of fast performance of 64-bit × 64-bit → 128-bit multiplication, which requires only four clocks on our PC environment. When we implement all operations in standard C programs (without assembly-level optimizations), it needs 16 clocks, and then our performance is about 4 times slower.

6.1.1. Ideal-lattice scheme case

The key generation (Yasuda et al., 2013, Algorithm 1) ran in about 870 ms [in Yasuda et al. (2013, Section 3.3), we improve the key generation of the ideal-lattice scheme for efficiency]. We made use of a precomputed table

$$M = \left\{1, [r]_d, [r^2]_d, \ldots, [r^{n-1}]_d\right\}$$

for efficiency. The packed encryption $Enc^{(1)}_{\mathrm{pack}}(\mathbf{T}, \mathrm{PK})$ or $Enc^{(2)}_{\mathrm{pack}}(\mathbf{Q}, \mathrm{PK})$ took 19.89 ms, the secure Hamming distance computation (17) took 9.05 ms [it is about *twice faster* than Yasuda et al.'s (2013) results due to the transformation (28) below], and finally, the decryption (18) took 9.08 ms. Hence the total time of the authentication phase of Section 4.3 is 38.02 ms. For the secure Hamming distance computation (17), we implemented

$$-\frac{1}{2}\{(2ct_1 - C_1) * (2ct_2 - C_2) - C_1 * C_2\}, \tag{28}$$

with $ct_1 = Enc^{(1)}_{\text{pack}}(\mathbf{T}, \text{PK})$ and $ct_2 = Enc^{(2)}_{\text{pack}}(\mathbf{Q}, \text{PK})$. The computation (28) mainly requires only one homomorphic multiplication if we precompute C_1, C_2 and $C_1 * C_2$. Any ciphertext of the ideal-lattice scheme can be represented as an integer included in $[-d/2, d/2)$. Since $d \approx 2^{n\alpha}$ by Yasuda et al. (2014, Theorem 3.1), the ciphertext size is about 4096×37 bits $\approx$ 19 KBytes. The precomputed table M has size 4096×19 KBytes $\approx$ 77.8 MBytes.

6.1.2. Ring-LWE scheme case

The key generation (excluding the prime generation) ran in 1.89 ms, the packed encryption $ct^{(1)}_{\text{pack}}(\mathbf{T})$ or $ct^{(2)}_{\text{pack}}(\mathbf{Q})$ took 3.65 ms, the secure Hamming distance computation (19) took 5.31 ms, and finally, the decryption took 3.47 ms. Then the total time of the authentication phase of Section 4.3 is 12.43 ms. As in the ideal-lattice scheme case, for the encrypted Hamming distance ct'_H given by (19), we implemented

$$-\frac{1}{2}\{(2ct'_1 - C_1(x)) * (2ct'_2 - C_2(x)) - C_1(x) * C_2(x)\},$$

where $ct'_1 = ct^{(1)}_{\text{pack}}(\mathbf{T})$ and $ct'_2 = ct^{(2)}_{\text{pack}}(\mathbf{Q})$. The size of $\text{pk} = (p_0, p_1) \in R_q^2$ is $2n \cdot \lg(q) \approx 31$ KBytes, and the size of $\text{sk} = s \in R_q$ is $n \cdot \lg(q) \approx 16$ KBytes. A fresh ciphertext has two elements of R_q, and hence its size is $2n \cdot \lg(q) \approx 31$ KBytes. However, the ciphertext ct'_H consists of three elements of R_q, and hence its size is $3n \cdot \lg(q) \approx 46.5$ KBytes [we did not use the relinearization technique of Naehrig et al. (2011, Sec. 3.2.3)].

6.2. Related work using homomorphic encryption

Schoenmakers and Tuyls (2006) proposed secure computations for privacy-preserving biometric authentication using the Paillier scheme (Paillier, 1999). Osadchy, Pinkas, Jarrous, and Moskovich, (2010) designed a new face recognition algorithm to propose an efficient secure face identification system, called SCiFI, with the Paillier scheme and the oblivious transfer protocol. In the SCiFI system, a feature vector extracted from a face image is represented as a binary vector of 900 bits, and the Hamming distance is used for matching. Their implementation showed that it took 310 ms to compute secure Hamming distance on an 8-core machine of 2.6 GHz AMD Opteron processors. Blanton and Gasti (2011) developed secure protocols for iris and fingerprints. Their method is similar to SCiFI, but they make use of the DGK scheme (Damgård, Geisler, & Krøigård, 2008), which is an additive one with shorter ciphertexts than the Paillier scheme. In their protocol, an iris feature is represented as a binary vector of 2048 bits and the Hamming distance is used as in SCiFI. Their implementation showed that it took 150 ms on an Intel Core 2 Duo 2.13 GHz to compute secure Hamming distance. Kulkarni and Namboodiri (2013) gave a secure biometric authentication protocol using the BGN scheme (Boneh et al., 2005), which can support additions and one-depth multiplications on encrypted data. Their protocol is secure against a semi-honest adversary, but it took 58 s on an Intel 3.2 GHz with around 200 KBytes memory usage for secure matching.

6.3. Comparison with related work

In Table 2 we give a comparison of our work to several previous works, with respect to performance and ciphertext sizes [in particular, we update the previous table of Yasuda et al. (2015, Table 3)]. Note that all schemes in Table 2 are estimated to have more than 80-bit security. While Blanton and Gasti (2011) and Osadchy et al. (2010) can protect either $\mathbf{T}$ or $\mathbf{Q}$ by using additive schemes, our SHE work can protect both of them, and it is also faster and shorter due to our packing method. For example, our method over the ideal-lattice scheme is about 16 times faster and about 14 times shorter than that of Blanton and Gasti (2011) when we ignore the difference in PC performance. The BGN scheme is used by Kulkarni and Namboodiri (2013) for secure Hamming distance, but its performance (i.e., 58

s) is much slower than ours. The BGN scheme is based on bilinear pairings on elliptic curves, and it requires pairing computations for one homomorphic multiplication [Lauter et al. (Naehrig et al., 2011, Sec. 1.2) describe that the homomorphic multiplication over the BGN scheme is slower than lattice-based schemes]. Furthermore, we cannot use our packing method over the BGN scheme because it has no homomorphic correctness over $R = \mathbb{Z}[x]/(x^n + 1)$, and hence it needs 2048 homomorphic multiplications for secure Hamming distance of two feature vectors of 2048 bits (cf. our work mainly requires only one homomorphic multiplication). This is why the BGN scheme gives much slower performance than ours. Finally, we compare the ideal-lattice and the ring-LWE schemes. The ring-LWE scheme gives faster performance (the security level of the ring-LWE is also higher), but the ideal-lattice scheme gives shorter ciphertexts. This is due mainly to the different homomorphic encryption schemes.

7. Challenge-response protocol

Challenge-response authentication is a family of protocols in which one party presents a question, called a *challenge*, and another party provides a valid answer, called a *response*, to be authenticated successfully. Challenge-response authentication gives resistance against replay attacks, in which an adversary intercepts data of a legitimate client passing on communication and sends the captured data for authentication. In this section, we construct a challenge-response protocol only over the ring-LWE scheme (it is hard to construct such a protocol over the ideal-lattice scheme). The difficulty is to combine a challenge-response protocol with our packing method described in Section 4.2, so as not to reduce efficiency of the secure Hamming distance computation (19). Our solution is to take monomials of the form cx^k as challenges. The main reason is that such a monomial is invertible over the ring R_t of the plaintext space if c is chosen to be prime to t, and the inverse element is computable as $-c^{-1}x^{n-k} \in R_t$.

7.1. Construction over ring-LWE scheme

Here we give our challenge-response protocol in the authentication phase of Section 4.3 (i.e., setup and enrollment phases are the same as in Section 4.3). Let (n,q,t,σ) and (pk, sk) denote the key parameters and the key pair of the ring-LWE scheme, respectively. As in Section 4.3, we assume that the computation server $\mathcal{CS}$ is semi-honest and the authentication server $\mathcal{AS}$ is honest.

7.1.1. Challenge

In accordance with every authentication request from the client device $\mathcal{CD}$, the authentication server $\mathcal{AS}$ randomly generates $0 \le c \le t-1$ and $0 \le k \le n-1$ with $(c, t) = 1$. Then $\mathcal{AS}$ encrypts the monomial $cx^k \in R_t$ and sends

$$ct_{\mathrm{chl}} := \mathrm{Enc}(cx^k, \mathrm{pk}) \tag{29}$$

to both $\mathcal{CD}$ and $\mathcal{CS}$ as a challenge.

7.1.2. Response

Given the challenge $ct_{\mathrm{chl}} = (c_0, c_1) \in (R_q)^2$, the client device $\mathcal{CD}$ extracts a queried biometric feature vector $\mathbf{Q}$ from a client's biometric image, and computes

$$\begin{aligned} ct_{\mathrm{res}} &:= F_2(\mathbf{Q}) * ct_{\mathrm{chl}} \dot{+} \mathrm{Enc}(0, \mathrm{pk}) \\ &= (F_2(\mathbf{Q}) \cdot c_0 + d_0, F_2(\mathbf{Q}) \cdot c_1 + d_1), \end{aligned} \tag{30}$$

where $Enc(0, pk) = (d_0, d_1)$ is randomly generated over $\mathcal{CD}$. This gives a ciphertext of $F_2(\mathbf{Q}) \cdot cx^k \in R_t$ due to the homomorphic correctness over R_t. Then $\mathcal{CD}$ sends ct_{res} to $\mathcal{CS}$ as a response.

7.1.3. Secure distance computation

Given a template $ct^{(1)}_{\mathrm{pack}}(\mathbf{T})$ stored in a database, the computation server $\mathcal{CS}$ computes [cf. (19)]

$$\begin{aligned} ct := ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) * (C_1(x) * ct_{\mathrm{chl}}) \dot{+} ct_{\mathrm{res}} * C_2(x) \\ \dot{+} (-2ct^{(1)}_{\mathrm{pack}}(\mathbf{T})) * ct_{\mathrm{res}}, \end{aligned}$$

where $C_1(x)$ and $C_2(x)$ are the polynomials (16) (note that the challenge ct_{chl} is sent to $\mathcal{CS}$ from $\mathcal{AS}$). In the case $c = 1$ (i.e., monomials of the form x^k are used as challenges), the computation becomes simple and it is given by

$$ct := ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) * C_1(x) \dot{+} ct_{\mathrm{res}} * C_2(x) \dot{+} (-2ct^{(1)}_{\mathrm{pack}}(\mathbf{T})) * ct_{\mathrm{res}} \tag{31}$$

Note that $C_1(x) * ct_{\mathrm{chl}}$ is a ciphertext of $\sum_{i=0}^{n-1} x^i \times x^k \in R$, whose x^k-coefficient is equal to that of $C_1(x)$. Hence ct_{chl} is not required in (31), and $\mathcal{AS}$ does not need to send it to $\mathcal{CS}$ in this case. We remark that ct is a ciphertext of

$$d_H(\mathbf{T}, \mathbf{Q}) \cdot cx^k + (\textit{the other terms}) \in R_t \tag{32}$$

by a similar argument in Section 4 [e.g., one homomorphic multiplication $ct^{(1)}_{\mathrm{pack}}(\mathbf{T}) * ct_{\mathrm{res}}$ gives a ciphertext of $F_1(\mathbf{T}) \times (F_2(\mathbf{Q}) \cdot cx^k) = \langle \mathbf{T}, \mathbf{Q} \rangle \cdot cx^k + \cdots]$. Then $\mathcal{CS}$ sends ct to the authentication server $\mathcal{AS}$.

7.1.4. Decryption

The authentication server $\mathcal{AS}$ decrypts the ciphertext ct with the secret key sk to obtain the polynomial (32). Using c and k (only $\mathcal{AS}$ knows), $\mathcal{AS}$ picks up the x^k-coefficient $c \cdot d_H(\mathbf{T}, \mathbf{Q}) (\mathrm{mod} t)$ and multiplies $c^{-1} (\mathrm{mod} t)$ to obtain the Hamming distance $d_H(\mathbf{T}, \mathbf{Q})$ for authentication.

7.2. Security analysis against replay attacks

As in Section 4.3, all data over $\mathcal{CS}$ are encrypted in our challenge-response protocol. Thereby our challenge-response protocol has the same security as in Section 4.3. Here we focus on the security of our challenge-response protocol against replay attacks. We first consider an adversary mounting replay attacks in our challenge-response protocol, and assume that he can eavesdrop both the challenge ct_{chl} and the response ct_{res} in every authentication procedure of legitimate clients. We remark that the adversary cannot learn any plain information from ct_{chl} and ct_{res} [in particular, he cannot obtain c, k, since cx^k is encrypted, and he also cannot obtain $F_2(\mathbf{Q})$ without knowing the noise (d_0, d_1) included in (30)]. However, to generate valid responses for every different challenge, the adversary might try to generate $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$ by performing certain homomorphic operations over ct_{chl} and ct_{res}. Since $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q}) * cx^k = ct_{\mathrm{res}}$ and $ct_{\mathrm{chl}} = \mathrm{Enc}(cx^k, \mathrm{pk})$, a *homomorphic division* is required to obtain $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$. However, the ring-LWE scheme cannot support any homomorphic division, and hence the adversary cannot obtain $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$.

We further consider case where the adversary could manage to obtain $ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$. In this case, given a challenge $ct_{\mathrm{chl}} = \mathrm{Enc}(cx^k, \mathrm{pk})$, the adversary can generate a valid response by a homomorphic multiplication $ct_{\mathrm{res}}{}' := ct^{(2)}_{\mathrm{pack}}(\mathbf{Q}) * ct_{\mathrm{chl}}$, which is a ciphertext of $F_2(\mathbf{Q}) \cdot cx^k \in R_t$. However, while the genuine response ct_{res} is composed of two ring elements in R_q, the fake response $ct_{\mathrm{res}}{}'$ has three ring elements in R_q (one homomorphic multiplication over the ring-LWE scheme between two ciphertexts of length $\xi + 1$ and $\eta + 1$ generates a ciphertext of length $\xi + \eta + 1$). Therefore the computation server $\mathcal{CS}$ can verify whether a response is genuine by checking the number of ring elements of the received response. To obtain a valid response without increasing the number of ring elements, the adversary should find an element $g \in R$ such that $g * ct^{(2)}_{\mathrm{pack}}(\mathbf{Q})$ is a ciphertext of $F_2(\mathbf{Q}) \cdot cx^k$. Then $g = cx^k$ over R_t. Therefore the adversary cannot perform replay attacks without knowing the two random elements c and k. In particular, since $0 \le c \le t - 1$ and $0 \le k \le n - 1$, the success probability of replay attacks is estimated to be about $1/tn$ if we choose t to be prime. In the restrictive case $c = 1$, the success probability is estimated as about $1/n$.

7.3. Selection of key parameters

Here we give key parameters (n, q, t, σ) of the ring-LWE scheme, suitable for our challenge-response protocol. For our implementation, we consider only the restrictive case $c = 1$ (i.e., we consider only monomials of the form x^k in our challenge-response protocol). Our chosen key parameters are

$$(n, q, t, \sigma) = (2048, 64\text{−bit}, 2048, 4). \tag{33}$$

As discussed in Section 5.2, the key parameters enable us to avoid decryption failure. Note that we select $\sigma = 4$ instead of $\sigma = 8$, since we set q to be a prime slightly smaller than a 64-bit integer (this q

enables efficient implementation). Furthermore, by the relation (26), the Hermite factor δ of the key parameters (33) is calculated as $\delta = 1.00533^n$. Thereby, as discussed in Section 5.2, we estimate that the key parameter (33) has more than 80-bit security with enough margin. In the key parameters, the success probability of replay attacks is estimated as about $1/n \approx 10^{-3.3}$.

7.4. Implementation results

In Table 3 we summarize our implementation results for our challenge-response protocol. Our PC environment and implementation methods are the same as in Section 6.1. For the parameters (33), it took 3.65 ms to generate a challenge ct_{chl} given by (29), and 3.12 ms to generate a response ct_{res} given by (30). Furthermore, it took 5.31 ms to compute the secure Hamming distance computation (31), and the decryption took 3.47 ms. The size of $\mathrm{pk} = (p_0, p_1) \in R_q^2$ is $2n \cdot \lg(q) \approx 32.8$ KBytes, and the size of $\mathrm{sk} = s \in R_q$ is $n \cdot \lg(q) \approx 16.4$ KBytes. Each of two ciphertexts ct_{chl} and ct_{res} has two ring elements in R_q, and its size is $2n \cdot \lg(q) \approx 32.8$ KBytes. In contrast, the ciphertext ct given by (31) has three ring elements, and its size is about 49.2 KBytes.

8. Conclusion

We applied the ideal-lattice and the ring-LWE SHE schemes to construct privacy-preserving biometric protocols. Due to our packing method over the schemes, our protocols give faster performance and shorter sizes than the state-of-the-art prior work using homomorphic encryption. In particular, the ring-LWE scheme gives faster performance but larger ciphertext sizes than the ideal-lattice scheme. We also applied our packing method over the ring-LWE scheme to construct a challenge-response protocol. Our challenge-response protocol is tolerant against replay attacks, and it gives practical performance.

Author note

This work is a fully revised version of works by Yasuda et al. (Yasuda et al., 2013; Yasuda, Shimoyama, Kogure, Yokoyama, & Koshiba, 2014). Specifically, we gave a theoretical evaluation of the decryption range of the ideal-lattice scheme (Section 3.2), and showed how to select suitable key parameters for privacy-preserving biometrics (Section 5.1). We rewrote the security level of our chosen parameters. We gave informative implementation results to compare the two schemes, and gave a comparison with typical work (Section 6). We proposed a challenge-response protocol and gave implementation results (Section 7).

Funding

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant number 16H02830.

References

Alperin-Sheriff, J., & Peikert, C. (2014). Faster bootstrapping with polynomial error. In *Advances in Cryptology–CRYPTO 2014* (Vol. 8616, pp. 297–314). New York, NY: Springer.

Blanton, M., & Gasti, P. (2011). Secure and efficient protocols for iris and fingerprint identification. In *European Conference on Research in Computer–ESORICS 2011* (Vol. 6879, pp. 190–209). New York, NY: Springer.

Boneh, D., Goh, E.-J., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. In *Theory of Cryptography–TCC 2005* (Vol. 3378, pp. 325–341). New York, NY: Springer.

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2012). (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference–ITCS 2012* (pp. 309–325). New York, NY: ACM.

Brakerski, Z., & Vaikuntanathan, V. (2011a, October). *Efficient fully homomorphic encryption from (standard) LWE*. Presented at 52nd Annual IEEE Symposium on Foundations of Computer Science–FOCS 2011, Palm Springs, CA.

Brakerski, Z., & Vaikuntanathan, V. (2011b). Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *Advances in Cryptology–CRYPTO 2011* (Vol. 6841, pp. 505–524). New York, NY: Springer.

Chen, Y., & Nguyen, P. Q. (2011). BKZ 2.0: Better lattice security estimates. In *Advances in Cryptology–ASIACRYPT 2011* (Vol. 7073, pp. 1–20). New York, NY: Springer.

Cheon, J. H., Coron, J.-S., Kim, J., Lee, M. S., Lepoint, T., Tibouchi, M., & Yun, A. (2013). Batch fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2013* (Vol. 7881, pp. 315–335). New York, NY: Springer.

Cheon, J. H., & Stehl´E, D. (2015). Fully homomophic encryption over the integers revisited. In *Advances in Cryptology–EUROCRYPT 2015* (Vol. 9056, pp. 513–536). New York, NY: Springer.

Coron, J.-S., Mandal, A., Naccache, D., & Tibouchi, M. (2011). Fully homomorphic encryption over the integers with shorter public keys. In *Advances in Cryptology–CRYPTO 2011* (Vol. 6841, pp. 487–504). New York, NY: Springer.

Damg°ard, I., Geisler, M., & Krøig°ard, M. (2008). Homomorphic encryption and secure comparison. *International Journal of Applied Cryptography*, *1*(1), 22–31. doi:10.1504/IJACT.2008.017048

Daugman, J. (2003). The importance of being random: Statistical principles of iris recognition. *Pattern Recognition*, *36*(2), 279–291. doi:10.1016/S0031-3203(02)00030-4

Ducas, L., & Micciancio, D. (2015). FHEW: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology–EUROCRYPT 2015* (Vol. 9056, pp. 617–640). New York, NY: Springer.

Fujitsu Laboratories Ltd. (2013). *Press release: Fujitsu develops world's first authentication technology to extract and match 2,048-bit feature codes from palm vein images*. Tokyo, Japan: Public and Investor Relations Division, Fujitsu Limited.

Gama, N., Nguyen, P., & Regev, O. (2010). Lattice enumeration using extreme pruning. In *Advances in Cryptology–EUROCRYPT 2010* (Vol. 6110, pp. 257–278). New York, NY: Springer.

Gama, N., & Nguyen, P. Q. (2008). Predicting lattice reduction. In *Advances in Cryptology–EUROCRYPT 2008* (Vol. 4965, pp. 31–51). New York, NY: Springer.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41th Symposium on Theory of Computing–STOC 2009* (pp. 169–178). New York, NY: ACM.

Gentry, C., & Halevi, S. (2011). Implementing Gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology–EUROCRYPT 2011* (Vol. 6632, pp. 129–148). New York, NY: Springer.

Gentry, C., Halevi, S., & Smart, N. P. (2012). Homomorphic evaluation of the AES circuit. In *Advances in Cryptology–CRYPTO 2012* (Vol. 7417, pp. 850–867). New York, NY: Springer.

Halevi, S., & Shoup, V. (2014). Algorithms in HElib. In *Advances in Cryptology–CRYPTO 2014* (Vol. 8616, pp. 554–571). New York, NY: Springer.

Hattori, M., Matsuda, N., Ito, T., Shibata, Y., Takashima, K., & Yoneda, T. (2012). Provably-secure cancelable biometrics using 2-DNF evaluation. *Journal of Information Processing*, *20*(2), 496–507.

Jain, A. K., Nandakumar, K., & Nagar, A. (2008). Biometric template security (review article). *EURASIP Journal on Advances in Signal Processing*, *2008*(113), 1–17. doi:10.1155/2008/579416

Kulkarni, R., & Namboodiri, A. (2013, June). *Secure Hamming distance based biometric authentication*. Presented at the 2013 International Conference on Biometrics–ICB 2013, Madrid, Spain.

López-Alt, A., Tromer, E., & Vaikuntanathan, V. (2012). On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the 44th Symposium on Theory of Computing–STOC 2012* (pp. 1219–1234). New York, NY: ACM.

Lenstra, A., Lenstra, H., & Lov´Asz, L. (1982). Factoring polynomials with rational coefficients. *Mathematische Annalen*, *261*(4), 515–534. doi:10.1007/BF01457454

Lindner, R., & Peikert, C. (2011). Better key sizes (and attacks) for LWE-based encryption. In *RSA Conference on Topics in Cryptology–CT-RSA 2011* (Vol. 6558, pp. 319–339). New York, NY: Springer.

Loftus, J., May, A., Smart, N. P., & Vercauteren, F. (2012). On CCA-secure somewhat homo-morphic encryption. In *Selected Areas in Cryptography–SAC 2011* (Vol. 7118, pp. 55–72). New York, NY: Springer.

Lyubashevsky, V., Peikert, C., & Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010* (Vol. 6110, pp. 1–23). New York, NY: Springer.

Micciancio, D., & Regev, O. (2007). Worst-case to average-case reduction based on Gaussian measures. *SIAM Journal on Computing*, *37*(1), 267–302. doi:10.1137/S0097539705447360

Naehrig, M., Lauter, K., & Vaikuntanathan, V. (2011). Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop–CCSW 2011* (pp. 113–124). New York, NY: ACM.

Osadchy, M., Pinkas, B., Jarrous, A., & Moskovich, B. (2010). SCiFI - A system for secure face identification. In *2010 IEEE Symposium on Security and Privacy* (pp. 239–254). New York, NY: ACM.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology–EUROCRYPT 1999* (Vol. 1592, pp. 223–238). New York, NY: Springer.

Schnorr, C.-P., & Euchner, M. (1994). Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, *66*(1), 181–199. doi:10.1007/BF01581144

Schnorr, C.-P., & Hörner, H. H. (1995). Attacking the Chor-Rivest cryptosystem by improved lattice reduction. In *Advances in Cryptology–EUROCRYPT 1995* (Vol. 921, pp. 1–12). New York, NY: Springer.

Schoenmakers, B., & Tuyls, P. (2006). Efficient binary conversion for Paillier encrypted values. In *Advances in Cryptology–EUROCRYPT 2006* (Vol. 4004, pp. 522–537). New York, NY: Springer.

Smart, N. P., & Vercauteren, F. (2010). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography–PKC 2010* (Vol. 6056, pp. 420–443). New York, NY: Springer.

Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2010* (Vol. 6110, pp. 24–43). New York, NY: Springer.

Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., & Koshiba, T. (2013). Packed homomorphic encryption based on ideal lattices and its application to biometrics. In *Modern Cryptography and Security Engineering–MoCrySEn 2013* (Vol. 8128, pp. 55–74). New York, NY: Springer.

Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., & Koshiba, T. (2014). Practical packing method in somewhat homomorphic encryption. In *Data Privacy Management–DPM 2013 and Autonomous Spontaneous Security–SETOP 2013* (Vol. 8247, pp. 34–50). New York, NY: Springer.

Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., & Koshiba, T. (2015). New packing method in somewhat homomorphic encryption and its applications. *Security and Communication Networks*, *8*, 2194–2213. doi:10.1002/sec.v8.13

Yasuda, M., Yokoyama, K., Shimoyama, T., Kogure, J., & Koshiba, T. (2014). On the exact decryption range for Gentry–Halevi's implementation of fully homomorphic encryption. *Journal of Mathematical Cryptology*, *8*(3), 305–329. doi:10.1515/jmc-2013-0024

Biographies

Masaya Yasuda received his B. S. degree in mathematics from Kyoto University, Japan in 2002. He received his M. S. degree and Ph.D. in mathematical sciences from University of Tokyo, Japan in 2004 and 2007, respectively. He had been engaged in research on cryptography at FUJITSU LABORATORIES Ltd. and FUJITSU Ltd. since 2007. Since 2015, he has been engaged in mathematical cryptography at Institute of Mathematics for Industry, Kyushu University. His recent interests are security evaluation of elliptic curve and lattice-based cryptography, and also practical applications of homomorphic encryption.