A Biometric Key-Binding Scheme Using Lattice Masking *

Yuka SUGIMURA, Masaya YASUDA,
Shigefumi YAMADA, Narishige ABE, Takashi SHINZAKI

FUJITSU LABORATORIES LTD.
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan
{sugimura.yuka, yasuda.masaya, yamada.shige, abe.narishige, shinzaki}@jp.fujitsu.com

Abstract: Template protection technology can protect the confidentiality of a biometric template by certain conversion. We focus on the key-binding approach for template protection. This approach generates a secure template (or a conversion template) from joint data of a user's specific key with a user's template, and the key can be correctly extracted from the secure template only when a queried biometric feature is sufficiently close to the original template. While almost all conventional schemes use the error correcting code (ECC) technique, we present a new technique based on lattices to give a new key-binding scheme. Our proposed scheme can provide several requirements (e.g., diversity and revocability) for template protection, which cannot be provided by ECC-based schemes such as the fuzzy commitment and the fuzzy vault.

1 Introduction

Biometrics is an authentication of users by using their physiological or behavioral features. Examples of the physiological ones include fingerprint, iris, face and vein, while signature, keystroke dynamics, and gait are some of the behavioral ones. Compared to the ID/password authentication, biometrics does not require users to remember their long and complex passwords, and hence the use of biometrics is now expanding in various applications ranging from international border crossings to securing information in databases (e.g., see US-VISIT [Sec07]). However, concerns about the security and the privacy are rapidly increasing at the same time. Especially, it is important to protect *templates* which are enrollment biometric features, since once leaked templates can be neither revoked nor replaced. During the rapid expansion of biometrics, the biometric template protection technology has been actively researched (e.g., see [Cam13, JNN08, RU11]), and its basic method is to store biometric features transformed by certain conversion, instead of storing raw ones. According to [JNN08, Section 3], an ideal biometric template protection scheme should satisfy the following four requirements;

(R-1) *Diversity*: the secure template must not allow cross-matching across databases.
(R-2) *Revocability*: it should be straightforward to revoke a compromised template and reissue a new secure template based on the same biometric data.

*This is a revised paper of our previous technical report [SYY+13]

(R-3) *Security*: it must be computationally hard to obtain the original biometric template from the secure template.
(R-4) *Performance*: the scheme should not degrade the recognition performance (e.g., FAR and FRR) of the system.

At present, there are four main approaches for template protection; *Salting*, *non-invertible transform*, *key-binding*, and finally *key-generation*. Each approach has both advantages and limitations, and it can not achieve an ideal scheme. Here we focus on the key-binding approach. This approach tries to protect a template by monolithically binding it with a user's specific key using cryptographic tools, and it can give various authentication ways.

Our Contribution According to [JNN08, Section 3.3], conventional schemes such as the fuzzy commitment [JW99] and the fuzzy vault [JS02] are based on the ECC technique. Our contribution is to propose a key-binding scheme using a new technique. While ECC-based schemes have difficulty of providing (R-1) diversity and (R-2) revocability (see [JNN08, Section 4.3]), our new scheme can provide both of them (in contrast, bipartite biotokens [SB09] gives a different solution in the key-binding approach, and it can support only revocability using the re-encoding methodology for revocable biotokens).

Our Strategy and Sketch of Our Scheme A number of techniques have been proposed in privacy-preserving data mining (PPDM). Among them, we take up the randomization method [AP08, Chapter 2] (or called the random masking method), in which sufficiently large noises are added to raw values so that individual values cannot be recovered but only statistics of the entire values can be (approximately) obtained. However, the method cannot be simply applied to biometrics since small errors between two biometric features should be permitted for authentication in biometrics. For the obstacle, we modify the randomization method by taking "lattice points" as noises (we call this new method *lattice masking*). A lattice is a set of infinite points in $\mathbb{R}^N$ spaced with sufficient regularity that one can shift any point onto any other point by symmetry of the arrangement (e.g., see [NV09] for lattices). Fix one lattice L, which gives an additive subgroup of $\mathbb{R}^N$. Given a pair (T, K) of a template and a user's specific key, we choose a random lattice point $r \in L$ to obtain a "masked" data $H := (T, K) + r$. Similarly, a queried biometric feature Q is transformed to a masked data $H' := (Q, 0) + r'$ by independently choosing a random lattice point $r' \in L$ (note that r' is independent of r). The difference

$$H - H' = (T - Q, K) + (r - r')$$

still includes the random lattice point $(r - r') \in L$ as a noise. Our trick for error tolerance is to "clear off this lattice point using a certain mathematical map in the theory of lattices". Note that this procedure is correctly performed only if T is sufficiently close to Q, and hence the correct key K can be extracted only in this case. Furthermore, as long as the lattice L is hidden to any attackers, the template security of our scheme relies only on the randomness of added noises as well as the original randomization method.

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 p, the finite field with p elements is denoted by $\mathbb{F}_p$. For $q \in \mathbb{R}$, let $\lceil q \rfloor$ denote the rounding of q to the nearest integer. This notation is extended to vectors and matrices in the natural way.

2 New Key-Binding Scheme

Before presenting a new key-binding scheme using the lattice masking method, we briefly review the theory of lattices and the framework of the key-binding approach.

Definitions and Properties on Lattices Let us first give basic definitions and properties on lattices [NV09, Chapter 2]; Fix a positive integer N. Let $\mathbf{B} \in \mathbb{R}^{N \times N}$ be a matrix, and $\vec{b}_i \in \mathbb{R}^N$ denote its i-th row for $i = 1, \ldots, N$. Denote by $\mathcal{L}(\mathbf{B}) = \left\{ \sum_{i=1}^{N} m_i \vec{b}_i : m_i \in \mathbb{Z} \right\}$ the set of all integral linear combinations of the $\vec{b}_i$'s, which gives an additive subgroup of $\mathbb{R}^N$. We say that the subgroup $\mathcal{L}(\mathbf{B})$ is a (full-rank) *lattice* of dimension N if all the vectors $\vec{b}_1, \ldots, \vec{b}_N$ are linearly independent (over $\mathbb{R}$). In this case, we also say that the matrix $\mathbf{B}$ is a *basis* of the lattice. Note that every lattice has infinitely many lattice bases (if $\mathbf{B}_1$ and $\mathbf{B}_2$ are two bases, there exists a unimodular matrix $U \in \mathrm{GL}_N(\mathbb{Z})$ satisfying $\mathbf{B}_1 = U \times \mathbf{B}_2$). For a given basis $\mathbf{B}$, we let $\mathcal{P}(\mathbf{B}) = \left\{ \sum_{i=1}^{N} x_i \vec{b}_i : x_i \in \left[-\frac{1}{2}, \frac{1}{2} \right) \right\}$ denote its associated half-open parallelepiped, which is called the *fundamental domain* of the lattice determined by the basis $\mathbf{B}$. For the construction of our key-binding scheme, we make use of the map "$\mathrm{mod}\mathbf{B}$" defined by

$$\vec{a} \bmod \mathbf{B} := \vec{a} - \left(\lceil \vec{a} \times \mathbf{B}^{-1} \rfloor \times \mathbf{B} \right) \in \mathcal{P}(\mathbf{B})$$

for $\vec{a} \in \mathbb{R}^N$. Note that the vector $\lceil \vec{a} \times \mathbf{B}^{-1} \rfloor \times \mathbf{B} \in \mathbb{R}^N$ gives a lattice point of $\mathcal{L}(\mathbf{B})$ since $\lceil \vec{a} \times \mathbf{B}^{-1} \rfloor \in \mathbb{Z}^N$ (specifically, it gives the nearest lattice point to the input vector $\vec{a}$). In particular, for any two vectors $\vec{v}, \vec{w} \in \mathbb{R}^N$, this map has the following two properties;

$$\text{(P-1) } (\vec{v} + \vec{w}) \bmod \mathbf{B} = \vec{v} \bmod \mathbf{B} \text{ if } \vec{w} \in \mathcal{L}(\mathbf{B}), \text{ and (P-2) } \vec{v} \bmod \mathbf{B} = \vec{v} \text{ if } \vec{v} \in \mathcal{P}(\mathbf{B}).$$

Framework of Key-Binding Approach Here we review the framework of the key-binding approach; The key-binding approach involves two parties, namely, a user and an authentication server. In the following, we show the specific authentication flow;

-**Enrollment:** An association, called a *helper data* H, is generated from both user's template T and specific key K using cryptographic tools, and then it is stored in a database of the server as the secure template (i.e., the conversion template).
-**Authentication:** The correct key K can be extracted from the helper data H only when user's queried biometric feature Q is sufficiently close to the original template T. Then a validity check is performed using the extracted key to output a decision.

The main advantage of this approach is that instead of providing a "match/non-match" decision, the system can authenticate a user using the extracted key K in various ways such as digital signature, document encryption/decryption and an authentication system without ID. The template security of this approach relies on the computational hardness of the following problem (it is related with (R-3) security in terms of the template protection); ($\bigstar$) "Given only the helper data H, can we recover either the key K or the original template T without any knowledge of user's biometric feature data?"

2.1 Construction of Our Scheme

As setup parameters, we need four integers (n, m, ℓ, k) with $k < N := n + m + \ell$. Different from the fuzzy vault scheme, our scheme handles an "order-variance" vector of length n as both a biometric template T and a queried biometric feature Q. In addition, a user's specific key K is also represented as a vector of length m. In order to distinguish the fuzzy vault scheme, we write $\vec{T}, \vec{Q}$, and $\vec{K}$ for the template T, the queried feature Q, and the key K, respectively. Then let us give the construction of our key-binding scheme;

1. **Setup Phase:** The authentication server generates N linearly independent vectors $\vec{v}_i \in \mathbb{R}^N$ for $1 \le i \le N$ (in practice, choose $\vec{v}_i \in \mathbb{Z}^N$ or $\mathbb{Q}^N$), and fix the lattice $L = \mathcal{L}(\mathbf{V})$ of dimension N with a basis $\mathbf{V} = (\vec{v}_1, \ldots, \vec{v}_N)^T$.

2. **Enrollment Phase:** The authentication server first sends k randomly chosen lattice points $\vec{b}_1, \ldots, \vec{b}_k \in L$ to the user. The user binds his biometric template $\vec{T} \in \mathbb{R}^n$ with his own specific key $\vec{K} \in \mathbb{R}^m$ into the vector $(\vec{T}, \vec{K}, \vec{0}) \in \mathbb{R}^N$ (note that the last vector $\vec{0} \in \mathbb{R}^\ell$ of length ℓ plays a important role to decide whether the key extraction is successful in the next authentication phase). To conceal this vector $(\vec{T}, \vec{K}, \vec{0})$, the user independently chooses k random integers (not general real numbers) $r_1, \ldots, r_k \in \mathbb{Z}$, and generates

$$\vec{H} = (\vec{T}, \vec{K}, \vec{0}) + \sum_{i=1}^{k} r_i \vec{b}_i \in \mathbb{R}^N. \tag{1}$$

Then the user sends only this vector $\vec{H}$ of length N to the authentication server.

3. **Authentication Phase:** The authentication server sends k randomly chosen lattice points $\vec{b}'_1, \ldots, \vec{b}'_k \in L$ to the user. Given his queried biometric feature $\vec{Q} \in \mathbb{R}^N$, the user independently chooses k random integers $r'_1, \ldots, r'_k \in \mathbb{Z}$ (our scheme has the advantage of taking $(\vec{b}'_i, r'_i)$'s quite different from $(\vec{b}_i, r_i)$'s chosen in the enrollment), generates

$$\vec{H}' = (\vec{Q}, \vec{0}, \vec{0}) + \sum_{i=1}^{k} r'_i \vec{b}'_i \tag{2}$$

as in equation (1) in order to conceal the queried biometric feature $\vec{Q}$, and sends only $\vec{H}'$ to the authentication server. After receiving $\vec{H}'$, the authentication server computes

$$\vec{z} = (\vec{H} - \vec{H}') \bmod \mathbf{V} \in \mathbb{R}^N \tag{3}$$

using the whole basis $\mathbf{V}$ of the lattice L, which only the authentication server knows. If all the last ℓ entries of $\vec{z}$ are equal to zero (this condition is equivalent to that the template $\vec{T}$ is sufficiently close to the queried feature $\vec{Q}$), then the authentication server can extract the correct key $\vec{K} \in \mathbb{R}^m$ from the $(n+1)$-th entry to the $(n+m)$-th entry of $\vec{z}$ with high probability (the probability can be controlled by the parameter ℓ). Then the authentication server can check the authentication validity by the extracted key.

REMARK. Given a masked template $\vec{H}$ generated by (1). If $k \le \ell$, then it is possible to uniquely determine integers r_i's from the last ℓ entries of $\vec{H}$ given k lattice points

$\vec{b}_i$'s, since the last ℓ entries of $(\vec{T}, \vec{K}, \vec{0})$ are all zero. Then it requires $k > \ell$. A similar attack is also possible for a masked query $\vec{H}'$, and hence we should take $k > \ell + m$ so that the queried feature $\vec{Q}$ can not be recovered from $\vec{H}'$. Furthermore, once a legitimate value of $\vec{Q}$ is known, the difference $\vec{H} - (\vec{Q}, \vec{0}, \vec{0}) = (\vec{T} - \vec{Q}, \vec{K}, \vec{0}) + \sum_{i=1}^{k} r_i \vec{b}_i$ has ℓ zeros in the last ℓ entries and n "small" values in the first n entries. In this case, a similar attack may recover the key $\vec{K}$ if $k \leq n + \ell$. Then, as a summary, we should take $k > \max(m, n) + \ell = N - \min(m, n)$. In other words, the number k of lattice points $\vec{b}_i$'s (or $\vec{b}'_i$'s) is required to be somewhat large for security. Even though we set suitable k, the current construction is vulnerable against the replay attack since there is no procedure to detect the attack. A countermeasure is to introduce the challenge-response mechanism in our scheme, but it is our future work.

Principle of Error Tolerance The error tolerance of our scheme depends only on the shape of the fundamental domain $\mathcal{P}(\mathbf{V})$, which is uniquely determined by the whole basis $\mathbf{V}$. Specifically, the key $\vec{K}$ can be correctly extracted if the condition

$$(\vec{T} - \vec{Q}, \vec{K}, \vec{0}) \in \mathcal{P}(\mathbf{V}) \tag{4}$$

is satisfied. This condition means that two vectors $\vec{T}, \vec{Q}$ are sufficiently close under the assumption that the key $\vec{K}$ is included in the appropriate range depending on the basis $\mathbf{V}$. Here we shall describe the principle of error tolerance in our scheme; The difference vector between the masked template $\vec{H}$ (i.e., the helper data) and the masked query $\vec{H}'$ is given by

$$\vec{D} := \vec{H} - \vec{H}' = (\vec{T} - \vec{Q}, \vec{K}, \vec{0}) + \sum_{i=1}^{k} (r_i \vec{b}_i - r'_i \vec{b}'_i),$$

and its last sum gives a lattice point of L since $\vec{b}_i, \vec{b}'_i \in L$. By the property (P-1), we have $\vec{z} = \vec{D} \bmod \mathbf{V} = (\vec{T} - \vec{Q}, \vec{K}, \vec{0}) \bmod \mathbf{V}$, irrespective of the lattice noise $\sum (r_i \vec{b}_i - r'_i \vec{b}'_i)$ in $\vec{D}$. In addition, the property (P-2) tells $\vec{z} = (\vec{T} - \vec{Q}, \vec{K}, \vec{0})$ if the condition (4) is satisfied. This means that in case where two vectors $\vec{T}, \vec{Q}$ are sufficiently close compared to the shape of $\mathcal{P}(\mathbf{V})$, we have $\vec{z} = (\vec{T} - \vec{Q}, \vec{K}, \vec{0})$ and hence the authentication server can extract the correct key $\vec{K}$ from the vector $\vec{z}$ calculated by the expression (3). Furthermore, the authentication server can verify whether the condition (4) is satisfied only by checking the last ℓ entries of $\vec{z}$. This verification procedure may cause a false detection, but the false probability would become negligible when the decision vector length ℓ is taken to be sufficiently long (we expect that $\ell \geq 10$ would be enough for verification without fail).

Advantages of Our Scheme In our scheme, any masked template can not allow cross-matching if different lattices are used among systems; Given two masked templates $\vec{H}_1 = (\vec{T}, \vec{K}, \vec{0}) + \vec{q}_1$ and $\vec{H}_2 = (\vec{T}, \vec{K}, \vec{0}) + \vec{q}_2$ created from the same template $\vec{T}$ and key $\vec{K}$ by using different lattice points $\vec{q}_1 \in L_1$ and $\vec{q}_2 \in L_2$. In this case, the difference $\vec{H}_1 - \vec{H}_2 = \vec{q}_1 - \vec{q}_2$ still seems to be random, and it enables an attacker to obtain neither $\vec{q}_1$ nor $\vec{q}_2$. Hence our scheme can provide (R-1) diversity. If a masked template is compromised, the system can revoke the masked template by changing the lattice, and it can make another masked template from the original template by using a new lattice, which shows

that our scheme can provide (R-2) revocability (i.e., our revoked templates can not reveal any information while the fuzzy vault [JS02] allows to recover the raw template by cross-matching). Although the fuzzy vault scheme hides biometric features among random chaff points, our scheme directly transforms biometric features into random elements. Then our masked templates are always uniform irrespective of biometric features. Therefore statistical analysis does not give an efficient attack against our scheme. Furthermore, in the fuzzy vault, when a legitimate user is authenticated, a "plain" query feature Q is sent to a system. In addition, for authentication, the system computes the intersection between the x-coordinate of a vault and Q, which is approximately equal to the original template T. Then the original template T is exposed temporarily, which might be glanced by an attacker. In contrast, only a "masked" query is sent to a system in our scheme. Furthermore, the system can compute neither the plain queried feature $\vec{Q}$ nor the original template $\vec{T}$ from any masked template $\vec{H}$ and any masked query $\vec{H}'$. Hence any attacker can glance neither $\vec{T}$ nor $\vec{Q}$ during the authentication procedure of our scheme.

2.2 Security Analysis of Our Scheme

Template Security The hardness of the template security problem ($\bigstar$) of our scheme relies on the security of the original randomization method. Specifically, given a masked template $\vec{H}$ obtained by the equation (1), a brute-force attack for obtaining the original template $\vec{T}$ or/and user's specific key K is to find the random noise $\sum_{i=1}^{k} r_i \vec{b}_i \in \mathbb{R}^N$ (for the attacker, this noise looks like just an N dimensional vector with large coefficients), and it is computationally hard if the noise has sufficient randomness depending only on user's random generator for the r_i's. Even in the case where the lattice points $\vec{b}_i$'s are leaked to the attacker, the computational time for finding the integers r_i's is roughly equal to ε^k, where ε denotes the maximal value of the r_i's (the value ε is determined by the entropy of the random integers r_i's). Given a security parameter λ, if it requires λ-bit security level for the template security, we must take k and ε satisfying $\varepsilon^k \geq 2^\lambda$. Conversely speaking, given k and λ, it requires (λ/k)-bit of entropy for the random integers r_i's.

Security against Impersonation Attack Next we consider an impersonation attack where an adversary tries to illegally log into a biometric system (or illegally obtain a user's specific key). Given a masked template $\vec{H}$ of joint data $(\vec{T}, \vec{K}, \vec{0})$, a brute-force method of the impersonation attack is to try to repeatedly input a queried feature vector $\vec{Q}$ sufficiently close to $\vec{T}$. Let $B \subset \mathbb{R}^n$ denote the maximal bounded domain of the template feature vectors $\vec{T}$ which any user can input. Due to the successful condition (4) for error tolerance, the number of candidates of queried feature vectors $\vec{Q}$ sufficiently close to $\vec{T}$ is approximately equal to $\#(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n)$ since $(\vec{Q}, \vec{0}, \vec{0}) \in (\vec{T}, \vec{K}, \vec{0}) + \mathcal{P}(\mathbf{V})$ can be successful for impersonation. Then the success probability of the brute-force impersonation attack is estimated to be $\#(B/(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n))^{-1}$. Therefore λ-bit security level requires

$$\#(B/(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n)) \geq 2^\lambda,$$

and it enforces us to control the balance between B and $\mathbf{V}$ for security. Simply speaking, we need to have larger B or smaller $\mathcal{P}(\mathbf{V})$ for higher security, but smaller $\mathcal{P}(\mathbf{V})$ can not

provide sufficient error tolerance.

Security against Insider Attack We further consider the worst case where the whole basis $\mathbf{V}$ is leaked to an attacker (or simply, we consider an insider attack where the authentication server is assumed to be malicious). In this case, the attacker can perform the validity check by computing (3), and he can extract the key K embedded in the masked template $\vec{H}$ by repeatedly trying to input a queried feature vector $\vec{Q}$ as well as the above impersonation attack. However, unlike the impersonation attack, we can consider an another attack using the map $\mathrm{mod}\mathbf{V}$ directly to a masked template $\vec{H}$; The $\mathrm{mod}\mathbf{V}$ procedure for $\vec{H}$ gives the relation $\mathcal{P}(\mathbf{V}) \ni \vec{H} \bmod \mathbf{V} = (\vec{T}, \vec{K}, \vec{0}) - \vec{q}, \quad \exists q \in L$, irrespective of the lattice noise $\sum_{i=1}^{k} r_i \vec{b}_i \in L$ included in $\vec{H}$ (i.e., the lattice point $\vec{q}$ is determined by $(\vec{T}, \vec{K}, \vec{0})$). Then the attacker only has to find the lattice point $\vec{q}$ for obtaining raw data $\vec{T}$ and $\vec{K}$ since $(\vec{T}, \vec{K}, \vec{0}) = \vec{q} + (\vec{H} \bmod \mathbf{V})$. Furthermore, the point $\vec{q}$ is the nearest lattice point to $(\vec{T}, \vec{K}, \vec{0})$. Hence the number of the candidates of the lattice point $\vec{q}$ is approximately equal to $\#(B/(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n))$, which equals the approximate number of lattice points included in the input domain B. Then the complexity of this attack is roughly estimated to be $\#(B/(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n))$, which is the same as in the above impersonation attack.

As a summary, given a security parameter λ (e.g., consider $\lambda = 80$), it requires both

$$\varepsilon^k \geq 2^\lambda \text{ and } \#(B/(\mathcal{P}(\mathbf{V}) \cap \mathbb{R}^n)) \geq 2^\lambda \tag{5}$$

to make our scheme secure against the above attacks. As seen from (5), the security depends mainly on the parameter n and the size of $\mathcal{P}(\mathbf{V})$ (larger n and smaller $\mathcal{P}(\mathbf{V})$ can give higher security). In particular, as described above, the whole basis $\mathbf{V}$ is not a decryption key but just a verification key (i.e., the whole basis $\mathbf{V}$ does not help an adversary to recover the raw template $\vec{T}$ from a masked template $\vec{H}$).

REMARK. Lattice reduction algorithms can compute a basis $\mathbf{B} = [\vec{b}_1, \ldots, \vec{b}_N]$ of a given lattice L with short and nearly orthogonal vectors $\vec{b}_1, \ldots, \vec{b}_N$. Lattice reduction is often used to solve several lattice problems such as SVP (shortest vector problem) and CVP (closest vector problem) [NV09]. Given lattice points $\vec{b}_i$'s or $\vec{b}'_i$'s, lattice reduction may help an adversary to recover the verification key matrix $\mathbf{V}$ (or a similar matrix). Since our scheme can not allow to recover the raw template $\vec{T}$ even with the verification matrix $\mathbf{V}$, lattice reduction can not help an adversary to break our scheme.

3 Concluding Remarks

We proposed a new scheme for the key-binding approach using the lattice masking technique, and the error tolerance range can be easily controled by choice of a lattice basis. Our proposed scheme is based on the randomization method, and processing performance of our scheme is considerably faster than general cryptographic techniques. Our scheme also has a number of advantages for template protection compared to ECC-based schemes such as the fuzzy vault. Specifically, our scheme can provide both (R-1) diversity and (R-2) revocability without any help of other approaches or auxiliary inputs such as passwords

(cf. bipartite biotokens [SB09] can provide (R-2) but cannot achive (R-1)). Furthermore, our scheme has no decryption key but only a verification key for authentication. In particular, if we set suitable parameters, even an administrator with the verification key can not recover the raw template from a secure template, and hence our scheme is secure against the insider attack. With respect to (R-4) performance, the error tolerance in our scheme depends only on the shape of $\mathcal{P}(\mathbf{V})$ and hence the choice of the verification key matrix $\mathbf{V}$ may break the balance between FAR and FRR (here we give only a theoretical analysis and a practical one is our future work). Furthermore, while the fuzzy vault scheme can be applied to various modalities including fingerprint due to the *order-invariance* property, our scheme has restrictive applications such as an implementation of keystroke dynamics (see our previous technical report [SYY+13]) due to that our scheme can handle only *order-variant* vectors. Therefore our future work is to modify our scheme in order to correspond various biometric modalities like the fuzzy vault. A solution for the obstacle is to use ideal lattices, which are special lattices with additional properties (unfortunately, due to space restriction, we can not describe ideal lattices in detail).

References

[AP08] Charu C. Aggrawal and S. Yu Philip. *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[Cam13] Patrizio Campisi. *Security and Privacy in Biometrics*. Springer, 2013.

[JNN08] Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric Template Security. *EURASIP J. Adv. Signal Process*, 2008:113:1–113:17, jan 2008.

[JS02] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006 (a preliminary version was presented at ISIT 2002).

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999.

[NV09] Phong Q. Nguyen and Brigitte Valle. *The LLL algorithm: survey and applications*. Springer Publishing Company, Incorporated, 2009.

[RU11] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):1–25, 2011.

[SB09] Walter J. Scheirer and Terrance E. Boult. Bipartite biotokens: Definition, implementation, and analysis. In *Advances in Biometrics*, pages 775–785. Springer, 2009.

[Sec07] Homeland Security. Privacy impact assessment for the biometric storage system, March 28, 2007.

[SYY+13] Yuka Sugimura, Masaya Yasuda, Shigefumi Yamada, Narishige Abe, and Takashi Shinzaki. A proposal of key binding technology using lattice masking (in Japanese). *IEICE Technical Report*, 113(137):297–304, 2013.