Lattice-based strongly-unforgeable forward-secure identity-based signature scheme with flexible key update

Xiangsong Zhang[1], Zhenhua Liu[2,3*]
[1] School of Science, Xi'an Technological University, Xi'an 710032 - China
[e-mail: zhualiu@hotmail.com]
[2] School of Mathematics and Statistics, Xidian University, Xi'an 710071 - China
[3]Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin, Guangxi 541004 - China
[e-mail: zh_liu@mail.xidian.edu.cn]
*Corresponding author: Zhenhua Liu

Received April 26, 2015; revised July 19, 2015; accepted December 13, 2016; published May 31, 2017

Abstract

Forward-secure signature is a specific type of signature, which can mitigate the damage caused by the signing key exposure. Most of the existing forward-secure (identity-based) signature schemes can update users' secret keys at each time period, achieve the existential unforgeability, and resist against classical computer attacks. In this paper, we first revisit the framework of forward-secure identity-based signatures, and aim at supporting flexible key update at multi time period. Then we propose a post-quantum forward-secure identity-based signature scheme from lattices and use the basis delegation technique to provide flexible key update. Finally, we prove that the proposed scheme is strongly unforgeable under the short integer solution (SIS) hardness assumption in the random oracle model.

Keywords: Forward security, digital signature, lattice-based cryptography, key update, strong unforgeability

This work is supported by the National Natural Science Foundation of China under Grants No.61472470, 61472309, 61100229 and 61173151, the National Natural Science Foundation of Shaanxi Province under Grants No.2014JM2-6091 and 2015JQ1007, the China Scholarship Council under Grants No. 201208610019, the Fundamental Research Funds for the Central Universities under Grants No. K5051270003, and the Scientific Research Plan Project of Education Department of Shaanxi Province under Grants No.12JK0852.

https://doi.org/10.3837/tiis.2017.05.027 **ISSN : 1976-7277**

1. Introduction

Identity-based signature (IBS), introduced by Shamir [1] in 1984, is a type of digital signature system in which a publicly known string identifying a user is used as a public key. The public string or identity can include an email address, a telephone number, or a physical IP address. A trusted third party, called a key generator center (KGC), generates a secret key according to the identity by using the system master secret key, and distributes the secret key to the corresponding user. Then the user can utilize her or his secret key to produce a signature for any message. Thus, identity-based signatures eliminate the need for certificates as used in a traditional public key infrastructure and reduce the cost of public key certificate management. Since then, identity-based signature has been extensively studied, and a large number of schemes have been published, such as [2-4].

Generally speaking, when considering the security of digital signature schemes, we usually refer to the existential unforgeability against adaptive chosen-message attacks [5]. Existential unforgeability (EUF) can guarantee that an adversary who is given signatures for a few messages of his choice could not produce a signature for a new message. However, it is required to use a stronger security property called strong unforgeability (SUF) in a variety of applications, e.g. signcryption [6], encryption secure against chosen-ciphertext attacks, group signature, authenticated group key exchange. The reason is that strong unforgeability can ensure the adversary cannot even produce a fresh signature for a previously signed message. In other words, suppose that an adversary obtains a message-signature pair $(\mathfrak{m}, \mathfrak{s})$ along with other message-signature pairs of his choice. A signature scheme is said to be strongly unforgeable [7-9] if the adversary cannot produce a new signature $\mathfrak{s}'$ for $\mathfrak{m}$.

At the same time, the security of digital signature schemes is usually studied under the assumption that secret keys are not exposed and are absolutely secure. However, in fact, key compromise seems inevitable or more likely to occur when mobile and unprotected devices are used in many cryptographic systems. When an adversary intrudes a user's storage space, it can steal her or his secret keys and perform any cryptographic operation. It is obvious that secret key exposure will directly threaten the security of digital signature schemes. To reduce the damage of key exposure, Anderson [10] firstly introduced forward security property for digital signatures. In a forward-secure signature scheme, the whole lifetime is divided into d time periods which are labeled from 1 to d. At the end of time period i, a user can self-update her or his current secret key sk_i to compute a new secret key sk_{i+1} for the next time period i+1 by using a one-way function. Then the old key sk_i is deleted and the new secret key sk_{i+1} is used to produce signatures at the time period i+1. In such a way, the secret key of a user is changed with different time period, but the public key is unchanged during the whole lifetime. Each signature is associated with one time period and the validity of time period needs to be verified during signature verification. As a result, compromise of the current secret key sk_i does not enable an adversary to forge signatures pertaining to the past j ($j < i$). This can be useful to mitigate the damage caused by key exposure without requiring distribution of keys. Until now, many forward-secure signature schemes [11-18] have been proposed including forward-secure identity-based signature (FSIBS) schemes [19-21].

Furthermore, most of the above mentioned forward-secure signature schemes are based on bilinear pairings. Their security rests on the hardness of the discrete logarithm problems and its variants. However, it is well known that, if in future quantum computer is realized, discrete logarithm problem can be solved by Shor's algorithm [22]. In order to resist against quantum computation attacks, post-quantum cryptography has been paid much attention in the field of cryptology. One of the most attractive post-quantum cryptosystems is lattice-based cryptography, which stems from its provable security guarantees, well studied theoretical underpinnings, and simplicity and potential efficiency. Recently, inspired by the breakthrough result of Ajtai in 1996 [23], lattice-based cryptography has been rapidly developing [24-35].

Our contribution. In this paper, we mainly focus on three properties of identity-based signature: forward security, strong unforgeability, and post-quantum security. Firstly, the existing forward-secure identity-based signature schemes can only evolve users' secret key period-by-period, and we revise the framework to provide flexible key update at multi time period. Secondly, we present a forward-secure identity-based signature scheme with flexible key update by using the basis delegation technique from lattices. Finally, the proposed scheme is proven to be strongly unforgeable under the small integer solution hardness assumption in the random oracle model. In addition, we show that there exists a flaw in the security proof of Zhang et al.'s forward-secure identity-based signature scheme from lattices [36], i.e. any challenger can solve an instance of short integer solution problem without the need of the adversary. The reason is that the challenger knows the initial trapdoor of lattice and is able to compute new trapdoors of any extended lattices by the basis delegation technique.

Organization. The rest of this paper is organized as follows. Some preliminaries are presented in Section 2. The revised framework of forward-secure identity-based signatures is proposed in Section 3. Our forward-secure identity-based signature scheme over lattices and its security proof are presented in Section 4. Some concluding remarks are given in Section 5. In appendix, Zhang et al.'s forward-secure identity-based signature scheme over lattices and its security proof are reviewed and analyzed.

2. Lattices

In this section, we will briefly review some fundamental backgrounds about lattice technique used in this paper.

We will use integer lattices, namely discrete subgroups of $\mathbb{Z}^m$. The specific lattices contain $q\mathbb{Z}^m$ as a sub-lattice for some prime q that is much smaller than the determinant of the lattice.

Definition 1 For a prime number q, $A_0 \in \mathbb{Z}_q^{n\times m}$ and $u \in \mathbb{Z}_q^n$, define:

$$\Lambda_q(A_0) := \{e \in \mathbb{Z}^m \ s.t.\ \exists s \in \mathbb{Z}_q^n, A_0^{\mathrm{T}} \cdot s = e \bmod q\}$$
$$\Lambda_q^{\perp}(A_0) := \{e \in \mathbb{Z}^m \ s.t.\ A_0 \cdot e = 0 \bmod q\}$$
$$\Lambda_q^{u}(A_0) := \{e \in \mathbb{Z}^m \ s.t.\ A_0 \cdot e = u \bmod q\}$$

Observe that if $t \in \Lambda_q^u(A_0)$ then $\Lambda_q^u(A_0) = \Lambda_q^{\perp}(A_0) + t$ and hence $\Lambda_q^u(A_0)$ is a shift of $\Lambda_q^{\perp}(A_0)$.

2.1 Hard problem from lattices

We recall the short integer solution (SIS) problem, which may be seen as an average-case problem related to the family of lattices described above.

Definition 2 An instance of the SIS $_{n,m,q,\beta}$ problem is a uniformly random matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ for any positive integers n, $m = poly(n)$, $q = poly(n)$, and a real norm bound

$\beta = poly(n)$. The goal is to find a non-zero integer vector $e \in \mathbb{Z}^m$ such that $\| e \| \le \beta$ and $A_0 \cdot e = 0 \in \mathbb{Z}_q^n$, i.e., $e \in \Lambda_q^{\perp}(A_0)$.

Gentry, Peikert and Vaikuntanathan showed in [26] that the SIS $_{n,m,q,\beta}$ problem is as hard (on the average) as approximating certain worst-case problems on lattices to within small factors.

Theorem 1 (Worst-case to Average-case Reduction) For any polynomial-bounded $m, \beta = poly(n)$ and for any prime $q \ge \beta \cdot \omega(\sqrt{n \log n})$, the average-case problem SIS $_{n,m,q,\beta}$ is as hard as approximating the shortest independent vectors problem (SIVP) problem in the worst case to within certain $\gamma \ge \beta \cdot \tilde{O}(\sqrt{n})$ factors.

2.2 The Gram-Schmidt norm of a basis

Let S be a set of vectors $S = \{s_1, \cdots, s_k\}$ in $\mathbb{R}^m$. We use the following notation:

- $\|S\|$ denotes the L_2 length of the longest vector in S, i.e. $\| S \| := \max_{1 \le i \le k} \| s_i \|$.
- $\overline{S} := \{\overline{s}_1, \cdots, \overline{s}_k\} \subset \mathbb{R}^m$ denotes the Gram-Schmidt orthogonalization of the vectors $s_1, \cdots, s_k$ taken in that order.

We refer to $\|\overline{S}\|$ as the Gram-Schmidt norm of S.

Micciancio and Goldwassser [25] showed that a full-rank set S in a lattice Λ can be converted into a basis T for Λ with an equally low Gram-Schmidt norm.

Lemma 1 ([25], Lemma 7.1) Let Λ be an m-dimensional lattice. There is a deterministic polynomial-time algorithm that, given an arbitrary basis of Λ and a full-rank set $S = \{s_1, \cdots, s_m\}$ in Λ, returns a basis T of Λ satisfying

$$\|\overline{T}\| \le \|\overline{S}\| \text{ and } \|T\| \le \|S\| \sqrt{m} / 2.$$

In cryptography, we typically hand over a "bad" basis with long vectors, as the public key, and keep a "good" (short) basis as our secret key. This principle goes back to Ajtai [24], who showed how to sample an essentially uniform matrix $A_0 \in \mathbb{Z}_q^{n \times m}$ with an associated basis T_{A_0} of $\Lambda_q^{\perp}(A_0)$ with low Gram-Schmidt norm. The most recent improvement for generating such a matrix A_0 together with a short trapdoor basis T_{A_0} is due to Alwen and Peikert [27].

Theorem 2 ([27], Theorem 3.2) Let $q \ge 3$ be odd and $m := \lceil 6n \log q \rceil$. There is a probabilistic polynomial-time algorithm **TrapGen** (q, n) that outputs a pair ($A_0 \in \mathbb{Z}_q^{n \times m}$, $T_{A_0} \in \mathbb{Z}^{m \times m}$) such that A_0 is statistically close to a uniform matrix in $\mathbb{Z}_q^{n \times m}$ and T_{A_0} is a basis for $\Lambda_q^{\perp}(A_0)$ satisfying

$$\|\overline{T}_{A_0}\| \le O(\sqrt{n \log q}) \text{ and } \| T_{A_0} \| \le O(n \log q)$$

with all but negligible probability in n.

Let $\sigma_{TG} = O(\sqrt{n \log q})$ denote the maximum Gram-Schmidt norm of a basis produced by **TrapGen** (q, n).

2.3 Discrete Gaussians

We briefly recall Gaussian distributions over lattices [26].

Definition 3 For any positive parameter $\sigma \in \mathbb{R}$ and any vector $c \in \mathbb{R}^m$, define:

$$\rho_{\sigma,c}(x) = \exp\left(-\pi \frac{\| x - c\|^2}{\sigma^2}\right).$$

For an m-dimensional lattice Λ, define the discrete Gaussian distribution $\mathfrak{D}_{\Lambda,\sigma,c}$ over Λ (centered at c) as

$$\forall y \in \Lambda, \mathfrak{D}_{\Lambda,\sigma,c}(y) = \frac{\rho_{\sigma,c}(y)}{\sum_{x\in\Lambda} \rho_{\sigma,c}(x)}.$$

2.4 Sampling a vector

Gentry, Peikert and Vaikuntanathan [26] proposed an efficient Gaussian sampling algorithm---**SamplePre** algorithm. The following lemma captures standard properties of these sampling distributions.

Lemma 2 Let $q \geq 2$ and let A_0 be a matrix in $\mathbb{Z}_q^{n\times m}$ with $m > n$. Let T_{A_0} be a basis for $\Lambda_q^{\perp}(A_0)$ and $\sigma \geq \|\overline{T}_{A_0}\| \cdot \omega(\sqrt{\log m})$. Then for $u \in \mathbb{Z}_q^n$:

(1) There is a PPT algorithm **SampleDom** (1^n) that samples an x from the distribution $\mathfrak{D}_{\mathbb{Z}^m,\sigma,0}$ over $D_n = \{x \in \mathbb{Z}^m : \| x\| \leq \sigma\sqrt{m}\}$, for which the distribution of $A_0 \cdot x$ is uniform over $\mathbb{Z}_q^n$.

(2) There exists a PPT algorithm **SampleGaussian** (A_0, T_{A_0}, σ) that returns $x \in \Lambda_q^{\perp}(A_0)$ drawn from a distribution statistically close to $\mathfrak{D}_{\Lambda_q^{\perp}(A_0),\sigma,0}$.

(3) There is a PPT algorithm **SamplePre** $(A_0, T_{A_0}, u, \sigma)$ that returns $x \in \Lambda_q^{u}(A_0)$ sampled from a distribution statistically close to $\mathfrak{D}_{\Lambda_q^{u}(A_0),\sigma,0}$, where $\Lambda_q^{u}(A_0)$ is not empty.

2.5 Basis delegation technique

When trapdoor delegation is required, one cannot simply hand over the resulting basis as it leaks information about the original trapdoor, and thus one needs to use algorithm **RandBasis** to obtain a randomized offspring with a new, random trapdoor. The following lemma shows the property of **RandBasis** algorithm [29].

Lemma 3 On input a basis T_0 of the lattice $\Lambda_q^{\perp}(A_0)$ of dimension m and a Gaussian parameter $\sigma \geq \|\overline{T}_0\| \cdot \omega(\sqrt{\log n})$, the polynomial time algorithm **RandBasis** (T_0, σ) outputs a basis T' of $\Lambda_q^{\perp}(A_0)$ with $\|\overline{T'}\| \leq \sigma \cdot \sqrt{m}$. The basis is independent of T_0 in the sense that for any two bases T_0, T' of $\Lambda_q^{\perp}(A_0)$ and $\sigma \geq \max\{\|\overline{T}_0\|, \|\overline{T'}\|\} \cdot \omega(\sqrt{\log n})$, **RandBasis** (T_0, σ) is within negligible statistical distance of **RandBasis** (T', σ).

Now, we recall Agrawal et al.'s basis delegation technique [31], which allows one to use a short basis of a given lattice to derive a new short basis of a related lattice in a secure way and does not increase the dimension of the underlying lattices. This technique includes **SampleR** algorithm, **BasisDel** algorithm, and **SampleRwithBasis** algorithm.

Definition 4 (1) A matrix R in $\mathbb{Z}^{m\times m}$ is $\mathbb{Z}_q$-invertible, if $R \bmod q$ is invertible as a matrix in $\mathbb{Z}^{m\times m}$.

(2) $\sigma_R := \overline{L}_{TG} \cdot \omega(\sqrt{\log m}) = \sqrt{n \log q} \cdot \omega(\sqrt{\log m})$.

(3) The distribution $\mathfrak{D}_{m\times m}$ on matrices in $\mathbb{Z}^{m\times m}$ is defined as $(\mathfrak{D}_{\mathbb{Z}^m,\sigma_R,0})^m$ conditioned on the resulting matrix being $\mathbb{Z}_q$-invertible.

Algorithm **SampleR**(1^m):

(1) Input a standard basis T of the lattice $\mathbb{Z}^m$.

(2) For $i = 1,\cdots,m$, do $r_i \leftarrow$ **SampleGaussian**($\mathbb{Z}^m, T, \sigma_R, 0$).

(3) Output R if $R = (r_1,\cdots,r_m)$ is $\mathbb{Z}_q$-invertible, otherwise repeat step (2).

The above algorithm samples matrices in $\mathbb{Z}^{m\times m}$ from a distribution that is statistically close to $\mathfrak{D}_{m\times m}$, and step (2) needs to be repeated fewer than two times in expectation for prime q.

Algorithm **BasisDel**(A, R, T_A, σ):

(1) Input a rank n matrix $A \in \mathbb{Z}_q^{n\times m}$, a $\mathbb{Z}_q$-invertible matrix $R \in \mathbb{Z}^{m\times m}$ sampled from $\mathfrak{D}_{m\times m}$, a basis T_A of $\Lambda_q^{\perp}(A)$, and a parameter $\sigma \in \mathbb{R}$.

(2) Let $T_A = \{a_1,\cdots,a_m\} \subseteq \mathbb{Z}^m$. Compute $T'_B := \{Ra_1,\cdots,Ra_m\} \subseteq \mathbb{Z}^m$. It is obvious that T'_B is a set of independent vectors in $\Lambda_q^{\perp}(B)$, where $B := AR^{-1} \in \mathbb{Z}_q^{n\times m}$.

(3) Convert T'_B into a basis T''_B of $\Lambda_q^{\perp}(B)$ by using Lemma 1, in which the algorithm takes as input T'_B and an arbitrary basis of $\Lambda_q^{\perp}(B)$, and outputs a basis T''_B whose Gram-Schmidt norm is no more than that of T'_B.

(4) Output the resulting basis $T_B \leftarrow$ **RandBasis**(T''_B, σ) for $\Lambda_q^{\perp}(B)$.

The following theorem shows the property of the random basis T_B produced by algorithm **BasisDel** for $\Lambda_q^{\perp}(AR^{-1})$.

Theorem 3 Suppose that R is sampled from $\mathfrak{D}_{m\times m}$ and σ satisfies $\sigma > \|\overline{T}_A| \cdot \sigma\sqrt{m} \cdot \omega(\log^{3/2} m)$ .Then T_B is distributed statistically close to the distribution **RandBasis**(T, σ), where T is an arbitrary basis of $\Lambda_q^{\perp}(AR^{-1})$ satisfying $\|\widetilde{T}\| \le \sigma / \omega(\sqrt{m})$. If R is a product of l matrices sampled from $\mathfrak{D}_{m\times m}$, then the bound on σ degrades to $\sigma > \|\overline{T}_A| \cdot (\sigma_R \sqrt{m}\omega(\log^{1/2} m))^l \cdot \omega(\log m)$.

Algorithm **SampleRwithBasis**(A):

(1) Generate $(B, T_B) \leftarrow$ **TrapGen**(q, n), where a random matrix B has rank n in $\mathbb{Z}_q^{n\times m}$ and T_B is a basis of $\Lambda_q^{\perp}(B)$ such that $\|\overline{T}_B\| \le \overline{L}_{TG} = \sigma_R / \omega(\sqrt{\log m})$.

(2) Let $A=(a_1,\cdots,a_m)\in\mathbb{Z}_q^{n\times m}$. For $i=1,\cdots,m$, do:

(a) Sample $r_i \leftarrow$ **SamplePre**(B,T_B,a_i,σ_R) in $\mathbb{Z}^m$, where $Br_i=a_i \bmod q$ and r_i is sampled from a distribution statistically close to $\mathfrak{D}_{\Lambda_q^{a_i}(B),\sigma_R,0}$.

(b) Repeat step (a) until r_i is $\mathbb{Z}_q$ linearly independent of $r_1,\cdots,r_{i-1}$.

(3) Output $R=(r_1,\cdots,r_m)\in\mathbb{Z}^{m\times m}$ and T_B, where R has rank m over $\mathbb{Z}_q$.

According to the construction, we have $BR = A \bmod q$, i.e. $B=AR^{-1} \bmod q$. Furthermore, we have the following property.

Theorem 4 Let $m>2n\log q$ and $q>2$ a prime. For all but at most a q^{-n} fraction of rank n matrices $A\in\mathbb{Z}_q^{n\times m}$, algorithm **SampleRwithBasis**(A) outputs a matrix $R\in\mathbb{Z}^{m\times m}$, which is sampled from a distribution statistically close to $\mathfrak{D}_{m\times m}$, and a short basis T_B of $\Lambda_q^{\perp}(AR^{-1})$ such that $\|\widetilde{T}_B\|\le\sigma_R/\omega(\sqrt{m})$ with overwhelming probability.

3. Formal definition and security model

In this section, we give the revisited definition of the syntax and the security model of forward-secure identity-based signature.

Firstly, we give the syntax of forward-secure identity-based signature scheme. Our syntax of forward-secure identity-based signature scheme is slightly different from previous one [19-21]. In our **Update** algorithm, any user can evolve his secret keys from the current time period $j\le d-1$ to the next any time period $i(j<i\le d)$, not just to the next time period $i=j+1\le d$ in one step. Consider a scenario where a manager can sign some documents by using his secret key, which is updated every day. In the next one week, the manager will get an annual leave of seven days and go traveling. In this case, the manager goes back and runs our **Update** algorithm only one time to generate his new secret key, whereas he has to execute previous **Update** algorithm [19-21] seven times according to original forward secure signature schemes. This case happens to the other forward-secure signature schemes [11-18]. Thus, in order to provide more flexible key update, we will revisit the **Update** algorithm in the framework of forward-secure identity-based signature. A revisited forward-secure identity-based signature (FSIBS) scheme consists of the following five algorithms:

- **Setup**(λ): This algorithm is run by key generation center (KGC) on input a security parameter λ and the total number of time periods d, and generates public parameters pp and master secret keys msk. Then the public parameters pp are published and the master secret key msk is kept to itself by KGC.
- **Extract**$(pp,msk,\mathfrak{u})$: Given pp, msk, and a user with identity $\mathfrak{u}$, this algorithm generates an initial secret key $sk_{\mathfrak{u},1}$ for the user $\mathfrak{u}$. KGC will use this algorithm to generate initial secret keys for all users participating in the system and distribute the initial secret keys to their respective owners via secure channels.
- **Update**$(pp,\mathfrak{u},i,sk_{\mathfrak{u},j})$: Given pp, a current secret key $sk_{\mathfrak{u},j}$ of a user $\mathfrak{u}$ at the current time period $j\le d-1$, this algorithm computes an update secret key $sk_{\mathfrak{u},i}$ for the user $\mathfrak{u}$

at an update time period $i(j < i \le d)$. The user $\mathfrak{u}$ can execute this algorithm by itself.

- **Sign**($pp, sk_{\mathfrak{u},i}, \mathfrak{m}$): Given pp, a message $\mathfrak{m}$, and a current secret key $sk_{\mathfrak{u},i}$ of a user $\mathfrak{u}$ at the current time period $i \le d$, this algorithm outputs a signature $\mathfrak{s}$. The user $\mathfrak{u}$ can make a signature by running this algorithm.
- **Verify**($pp, \mathfrak{u}, \mathfrak{m}, \mathfrak{s}, i$): Given pp, a candidate signature $\mathfrak{s}$, a message $\mathfrak{m}$, a user $\mathfrak{u}$ and the time period $i \le d$, this algorithm outputs *accept* if $\mathfrak{s}$ is a valid signature of the user $\mathfrak{u}$ on the message $\mathfrak{m}$ at the current time period i, and outputs *reject* otherwise.

Next, we give the formal security definition---strong unforgeability under adaptive chosen identity and message attacks (SUF-ID-CMA) for forward-secure identity-based signatures, which is viewed as a combination of strong unforgeability with existential unforgeability under adaptive chosen identity and message attacks (EUF-ID-CMA) for forward-secure identity-based signatures [20]. More precisely, the security is defined using the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$:

- **Setup**. The challenger $\mathcal{C}$ runs the Setup algorithm. It gives the adversary $\mathcal{A}$ the resulting public parameters pp and keeps the master secret key msk by itself.
- **Queries**. The adversary $\mathcal{A}$ adaptively makes a number of different queries to the challenger $\mathcal{C}$. Each query can be one of the following.
  ---**Extract queries**. $\mathcal{A}$ can request a secret key of any user $\mathfrak{u}$ at any time period $i \le d$. For $i = 1$, $\mathcal{C}$ responds by running **Extract**($pp, msk, \mathfrak{u}$) and forwards the initial secret key $sk_{\mathfrak{u},1}$ to $\mathcal{A}$. And for $1 \le j < i \le d$, $\mathcal{C}$ returns the resulting $sk_{\mathfrak{u},i} \leftarrow$ **Update** $(pp, \mathfrak{u}, i, sk_{\mathfrak{u},j})$ to $\mathcal{A}$. Especially, for $i = 1$, we can also view $sk_{\mathfrak{u},1} \leftarrow$ **Update** $(pp, \mathfrak{u}, 1, sk_{\mathfrak{u},0})$, where $sk_{\mathfrak{u},0} := msk$.
  ---**Sign queries**. $\mathcal{A}$ can ask for a signature of any user $\mathfrak{u}$ on any message $\mathfrak{m}$ for any time period $i \le d$. $\mathcal{C}$ responds by first running **Update** ($pp, \mathfrak{u}, i, sk_{\mathfrak{u},j}$) to obtain the secret key $sk_{\mathfrak{u},i}$ of $\mathfrak{u}$ at the time period i, and then running **Sign**($pp, sk_{\mathfrak{u},i}, \mathfrak{m}$) to obtain a signature $\mathfrak{s}$, which is forwarded to $\mathcal{A}$.
- **Forgery**. $\mathcal{A}$ outputs a user with identity $\mathfrak{u}^*$, a message $\mathfrak{m}^*$, a time period $i^* \le d$ and a candidate signature $\mathfrak{s}^*$. $\mathcal{A}$ succeeds if the followings hold true:
  (1) **Verify**($pp, \mathfrak{u}^*, \mathfrak{m}^*, i^*, \mathfrak{s}^*$)=*accept*.
  (2) $\mathcal{A}$ has not made extract queries on $\mathfrak{u}^*$ at any time period $i \le i^*$.
  (3) $(\mathfrak{u}^*, \mathfrak{m}^*, i^*, \mathfrak{s}^*)$ is not among the tuples generated during the sign queries,

The advantage of an adversary $\mathcal{A}$ in the above game is defined as

$$Adv_{\mathcal{A}} = \Pr[\mathcal{A}\ succeeds],$$

where the probability is taken over all coin tosses made by the challenger and the adversary.

A forward-secure identity-bases signature scheme is SUF-ID-CMA secure, if for any adversary $\mathcal{A}$, its advantage is negligible in the security parameter.

4. The proposed scheme and its security

In Zhang et al.'s forward-secure signature scheme [36], the **Update** algorithm can only update a secret key from a time period $i-1$ to the next time period i, i.e., the interval of time period $\Delta t = 1$. So do the **Update** algorithms in the existing forward-secure identity-bases signature schemes [19-21]. In this section, we will improve the **Update** algorithm to provide more flexible key update, i.e. the interval of time period $\Delta t \ge 1$, and then prove that the improved scheme is strongly unforgeable under adaptively chosen identity and message attacks in the random oracle model. Furthermore, we will show that there exists a flaw in the security proof of Zhang et al.'s scheme [36] in appendix.

4.1 The proposed forward-secure identity-based signature scheme

- **Setup**(n): On input a security parameter n, set the parameters m, q, divide the whole lifetime into d time periods, and set two series of Gaussian parameters $\overline{\sigma} = (\sigma_1, \cdots, \sigma_d)$ and $\overline{\delta} = (\delta_1, \cdots, \delta_d)$. Next do:

  (1) Use algorithm **TrapGen** (q, n) to generate a uniformly random $n \times m$ matrix $A_0 \in \mathbb{Z}_q^{n\times m}$ with a corresponding short basis T_{A_0} for $\Lambda_q^{\perp}(A_0)$ such that $\|\widetilde{T}_{A_0}\| \le \mathcal{O}(\sqrt{n \log q})$.

  (2) Define two hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}^{m\times m}$, where the output is distributed as $\mathfrak{D}_{m\times m}$ [31], and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^n$.

  (3) Publish the public parameters $pp = \left(A_0,\ H_1,\ H_2\right)$, and keep master secret key $msk = (T_{A_0})$ secret.

- **Extract**($pp, msk, \mathfrak{u}$): On input public parameter pp, a master key msk, a user with identity $\mathfrak{u}$ and an initial time period $i = 1$, KGC does:

  (1) Let $R_{\mathfrak{u}\|1} = H_1(\mathfrak{u} \,\|\, 1)$ and compute $A_{\mathfrak{u}\|1} = A_0 \cdot R_{\mathfrak{u}\|1}^{-1} \bmod q$.

  (2) Evaluate

  $$sk_{\mathfrak{u}\|1} \leftarrow \textbf{BasisDel}(A_0, R_{\mathfrak{u}\|1}, T_{A_0}, \sigma_1).$$

  (3) Send a trapdoor $sk_{\mathfrak{u}\|1}$ of $\Lambda_q^{\perp}(A_{\mathfrak{u}\|1})$ to the user over a secure channel.

- **Update**($pp, \mathfrak{u}, i, sk_{\mathfrak{u}\|j}$): On input public parameter pp, the current time period $i \le d$, and $sk_{\mathfrak{u}\|j}$ which denotes the signing secret key associated with the previous time period $j < i$, the user with identity $\mathfrak{u}$ performs the following steps to update his signing secret key:

  (1) Compute $R_{\mathfrak{u}\|j} = H_1(\mathfrak{u}\| j)\cdots H_1(\mathfrak{u}\|1)$ and $A_{\mathfrak{u}\|j} = A_0 \cdot R_{\mathfrak{u}\|j}^{-1} \bmod q$ as the public key at time period j with respect to signing secret key $sk_{\mathfrak{u}\|j}$.

  (2) Let $R_{j\to i} = H_1(\mathfrak{u}\| i)\cdots H_1(\mathfrak{u}\| j+1)$, and compute

  $$sk_{\mathfrak{u}\|i} \leftarrow \textbf{BasisDel}(A_{\mathfrak{u}\|j}, R_{j\to i}, sk_{\mathfrak{u}\|j}, \sigma_i).$$

  Note that $sk_{\mathfrak{u}\|i}$ is a short basis of $\Lambda_q^{\perp}(A_{\mathfrak{u}\|i})$, where $A_{\mathfrak{u}\|i} = A_{\mathfrak{u}\|j} \cdot R_{j\to i}^{-1} = A_0 \cdot R_{\mathfrak{u}\|i}^{-1} \bmod q$ and $R_{\mathfrak{u}\|i} = H_1(\mathfrak{u}\| i)\cdots H_1(\mathfrak{u}\|1)$. Obviously, when $j = i-1$, our update algorithm is degraded to Zhang et al.'s key update algorithm.

- **Sign**($pp, sk_{\mathfrak{u}\|i}, \mathfrak{m}$):On input public parameters pp and a message $\mathfrak{m} \in \{0,1\}^*$, the signing user $\mathfrak{u}$, whose signing secret key is $sk_{\mathfrak{u}\|i}$ at the current time period $i \le d$, computes $y = H_2(\mathfrak{u}\| i\| \mathfrak{m}) \in \mathbb{Z}_q^n$ and evaluates
$$e_i \leftarrow \textbf{SamplePre}(A_{\mathfrak{u}\|i}, sk_{\mathfrak{u}\|i}, y, \delta_i),$$
Note that $A_{\mathfrak{u}\|i} \cdot e_i = y \bmod q$ and e_i is distributed as $\mathfrak{D}_{\Lambda_q^y(A_{\mathfrak{u}\|i}),\delta_i}$. Finally, the signer outputs a signature e_i.
- **Verify**($pp, \mathfrak{u}, \mathfrak{m}, i, e_i$): On input public parameters pp, a user with identity $\mathfrak{u}$, an index of time period i, a message $\mathfrak{m}$ and a candidate signature e_i, the algorithm outputs *accept* if and only if
$$0 < \| e_i \| \le \delta_i \cdot \sqrt{m} \text{ and } A_{\mathfrak{u}\|i} \cdot e_i = y \bmod q,$$
where $A_{\mathfrak{u}\|i} = A_0 \cdot R_{\mathfrak{u}\|i}^{-1} \bmod q$, $R_{\mathfrak{u}\|i} = H_1(\mathfrak{u}\| i) \cdots H_1(\mathfrak{u}\|1)$, and $y = H_2(\mathfrak{u}\| i\| \mathfrak{m})$. Otherwise, it outputs *reject*.

4.2 Security proof

Now, we give the proper security reduction for the proposed scheme.

Theorem 5 In the random oracle model, the proposed forward-secure identity-based signature scheme is strongly unforgeable under adaptively chosen identity and message attacks, provided that the SIS hard problem assumption holds.

Proof. Assume that for the proposed scheme there exists an adversary $\mathcal{A}$, which makes at most Q_{H_1} times H_1 oracle queries, Q_{H_2} times H_2 oracle queries, Q_E extract queries, and Q_S signing queries, and has the advantage ε in time t. According to the adversary, we will build an algorithm $\mathcal{C}$ that solves an instance of SIS (Definition 2) with probability at least ε' and in time at most t', contradicting the SIS hard problem assumption.

The algorithm $\mathcal{C}$ will be given a random matrix $A_0 \in \mathbb{Z}_q^{n\times m}$. To use $\mathcal{A}$ to find a nonzero integer vector $e \in \mathbb{Z}^m$ such that $A_0 \cdot e = 0 \bmod q$ and $\| e \| \le \beta$, $\mathcal{C}$ must simulate a challenger for $\mathcal{A}$. Such a simulation can be created in the following way:

Setup. $\mathcal{C}$ prepares system public parameters for $\mathcal{A}$ as follows.

(1) Select d uniform random integer $Q_1^*, \cdots, Q_d^* \in [Q_{H_1}]$, where Q_{H_1} is the maximum number of queries of H_1 that $\mathcal{A}$ can make.

(2) Sample d random matrices $R_1^*, \cdots, R_d^* \sim \mathfrak{D}_{m\times m}$ by running $R_i^* \leftarrow \textbf{SampleR}(1^m)$ for $i = 1, \cdots, d$.

(3) Choose a random $w \in [d]$ and set $A \leftarrow A_0 R_w^* \cdots R_1^*$. The matrix A is uniform in $\mathbb{Z}_q^{n\times m}$ since all R_i^* are invertible $\bmod q$ and A_0 is uniform in $\mathbb{Z}_q^{n\times m}$.

(4) Pick two hash functions as random oracles, $H_1 : \{0,1\}^* \to \mathbb{Z}_q^{m\times m}$ and $H_1 : \{0,1\}^* \to \mathbb{Z}_q^{m\times m}$.

(5) Publish the public parameters $pp = \left(A,\ H_1,\ H_2\right)$.

H_1 **random oracle queries**. $\mathcal{A}$ may adaptively query the random oracle H_1 on any identity $\mathfrak{u}$ and any time period i of its choice. To respond consistently to these queries, $\mathcal{C}$ maintains a list $L_1 = \{(\mathfrak{u}, i, H_1(\mathfrak{u} \| i), *, *)\}$ which is initially empty, and the simulator simply returns the same output on the same input without incrementing the query counter Q_{H_1}. $\mathcal{C}$ answers the Q-th such query as follows.

(1) For $Q = Q_i^*$, set $H_1(\mathfrak{u} \| i) := R_i^*$, store the tuple $(\mathfrak{u}, i, R_i^*, -, -)$ in L_1, and return R_i^* as the oracle $H_1(\mathfrak{u} \| i)$'s value.

(2) For $Q \neq Q_i^*$, compute $A_i = A \cdot (R_{i-1}^* \cdots R_2^* R_1^*)^{-1} \in \mathbb{Z}_q^{n \times m}$ (if $i = 1$, then set $A_1 = A$), run **SampleRwithBasis** $(A_i) \to (R_i, T_B)$ (where $R_i \sim \mathfrak{D}_{m \times m}$ and a short basis T_B for $\Lambda_q^{\perp}(B)$ such that $B = A_i \cdot R_i^{-1} \bmod q$), save the tuple $(\mathfrak{u}, i, R_i, B, T_B)$ in L_1, and return R_i as the value of $H_1(\mathfrak{u} \| i)$.

Extract oracle queries. $\mathcal{A}$ makes adaptively key extraction queries on arbitrary identities $\mathfrak{u}$ and any time period $i \le d$. To respond consistently to these queries, $\mathcal{C}$ maintains a list $L_2 = \{(\mathfrak{u}, i, A_{\mathfrak{u} \| i}, sk_{\mathfrak{u} \| i})\}$ which is initially empty, and the simulator simply returns the same output on the same input. $\mathcal{C}$ answers an initial or update key query on $(\mathfrak{u} \| i)$ as follows.

(1) Let $j \in [i]$ be the oldest time period such that $H_1(\mathfrak{u} \| j) \neq R_j^*$. This implies that $H_1(\mathfrak{u} \| 1) = R_1^*, \cdots, H_1(\mathfrak{u} \| j-1) = R_{j-1}^*$. If $H_1(\mathfrak{u} \| j) = R_j^*$ for all $j = 1, \cdots, i$, then the simulator aborts and fails.

(2) Retrieve the stored tuple $(\mathfrak{u}, j, R_j, B, T_B)$ from L_1. This tuple was created when responding to a query for $H_1(\mathfrak{u} \| j)$. Assume that a key extraction query on $(\mathfrak{u} \| j)$ is preceded by a hash query on all identity and previous time period, i.e. $(\mathfrak{u} \| 1), \cdots, (\mathfrak{u} \| j-1)$. By construction

$$B = A(R_{j-1}^* \cdots R_1^*)^{-1} \cdot H_1(\mathfrak{u} \| j)^{-1} \bmod q$$

and T_B is a short basis for $\Lambda_q^{\perp}(B)$.

Note that $A_{\mathfrak{u} \| j} = A \cdot H_1(\mathfrak{u} \| 1)^{-1} \cdots H_1(\mathfrak{u} \| j)^{-1} = B$, and therefore T_B is a trapdoor for $\Lambda_q^{\perp}(A_{\mathfrak{u} \| j})$, i.e. the secret key $sk_{\mathfrak{u} \| j} = T_B$. Save $(\mathfrak{u}, j, A_{\mathfrak{u} \| j}, sk_{\mathfrak{u} \| j})$ in L_2.

(3) Run **Update**($pp, \mathfrak{u}, i, sk_{\mathfrak{u} \| j}$) to generate an update secret key $sk_{\mathfrak{u} \| i}$ for $\mathfrak{u}$ from the trapdoor secret key T_B for the identity and time period tuple $(\mathfrak{u} \| j)$, and save $(\mathfrak{u}, i, A_{\mathfrak{u} \| i}, sk_{\mathfrak{u} \| i})$ in L_2. Specially, if $j = i$, evaluate **RandBasis** $(sk_{\mathfrak{u} \| j}, \sigma_i) \to sk_{\mathfrak{u} \| i}$. Then send the resulting secret key $sk_{\mathfrak{u} \| i}$ to the adversary.

Note that when $i = 1$, such a query is viewed as an initial secret key query for $(\mathfrak{u} \| 1)$. If random oracle $H_1(\mathfrak{u} \| 1) = R_1^*$, then the simulator aborts and fails. Otherwise, by construction $B = A \cdot H_1(\mathfrak{u}\|1)^{-1} \bmod q$ and T_B is a short basis for $\Lambda_q^{\perp}(B)$. Then return an initial secret key $sk_{\mathfrak{u}\|1} \leftarrow$ **RandBasis** (T_B, σ_1) and save $(\mathfrak{u}, 1, A_{\mathfrak{u}\|1} = B, sk_{\mathfrak{u}\|1})$ in L_2.

H_2 **random oracle queries**. $\mathcal{A}$ may adaptively query the random oracle H_2 on any identity $\mathfrak{u}$, any time period i and any message $\mathfrak{m}$ of its choice. To respond consistently to these queries, $\mathcal{C}$ maintains a list $L_3 = \{(\mathfrak{u}, i, \mathfrak{m}, e_i, H_2(\mathfrak{u}\| i\| \mathfrak{m}))\}$ which is initially empty, and the simulator simply returns the same output on the same input. $\mathcal{C}$ answers such query as follows.

(1) Look up $(\mathfrak{u}, i, A_{\mathfrak{u}\| i}, *)$ in L_2 (if necessary, look up $(\mathfrak{u}, i, H_1(\mathfrak{u}\| i), *, *)$ in L_1 and compute $A_{\mathfrak{u}\| i} = A \cdot H_1(\mathfrak{u}\| i)^{-1} \cdots H_1(\mathfrak{u}\|1)^{-1}$).

(2) Run **SampleDom** $(1^n) \rightarrow e_i$, compute $H_2(\mathfrak{u}\| i\| \mathfrak{m}) = A_{\mathfrak{u}\| i} \cdot e_i \bmod q$, store the tuple $(\mathfrak{u}, i, \mathfrak{m}, e_i, H_2(\mathfrak{u}\| i\| \mathfrak{m}))$ in L_3, and return $A_{\mathfrak{u}\| i} \cdot e_i \bmod q$ as the oracle $H_2(\mathfrak{u}\| i\| \mathfrak{m})$'s value.

Signing oracle queries. When running the adversary $\mathcal{A}$, signing queries can occur. Suppose $\mathcal{A}$ asks for a signature on identity $\mathfrak{u}$ at time period i for a message $\mathfrak{m}$. $\mathcal{C}$ answers these queries as follows. $\mathcal{C}$ looks up $(\mathfrak{u}, i, \mathfrak{m}, e_i, A_{\mathfrak{u}\| i} \cdot e_i \bmod q)$ in L_3, and returns e_i as the signature (if necessary, query H_2 random oracles on $(\mathfrak{u}, i, \mathfrak{m})$ in advance).

Challenge. Finally, $\mathcal{A}$ produces a forged signature e^* for $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$ on which it wishes to be challenged. We require that $\mathfrak{u}^*$ has not been requested in any preceding and subsequent extract oracle queries. If $w \neq i^*$ and $H_1(\mathfrak{u}\| j) \neq R_j^*$ for all $j = 1, \cdots, i^*$, then the simulator aborts and fails. Otherwise, i.e. $w = i^*$ and $H_1(\mathfrak{u}\| j) = R_j^*$ for all $j = 1, \cdots, i^*$, recall that $A = A_0 \cdot R_w^* \cdots R_1^*$. Then by definition

$$A_{\mathfrak{u}\| i^*} = A \cdot (R_1^*)^{-1} \cdots (R_w^*)^{-1} = A_0 \in \mathbb{Z}_q^{n \times m}.$$

Furthermore, we have

$$A_{\mathfrak{u}^*\| i^*} \cdot e^* \bmod q = H_2(\mathfrak{u}^*\| i^*\| \mathfrak{m}^*).$$

Now without loss of generality, we assume that before outputting its forgery e^*, $\mathcal{A}$ queries H_2 random oracle on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$ and $\mathcal{C}$ returns $A_{\mathfrak{u}^*\| i^*} \cdot e_{\mathfrak{m}^*}$, i.e.

$$H_2(\mathfrak{u}^*\| i^*\| \mathfrak{m}^*) = A_{\mathfrak{u}^*\| i^*} \cdot e_{\mathfrak{m}^*} \bmod q.$$

Therefore,

$$A_{\mathfrak{u}^*\| i^*} \cdot e_{\mathfrak{m}^*} = H_2(\mathfrak{u}^*\| i^*\| \mathfrak{m}^*) = A_{\mathfrak{u}^*\| i^*} \cdot e^* \bmod q,$$

i.e. $A_0 \cdot e_{\mathfrak{m}^*} = A_0 \cdot e^* \bmod q$ and $A_0 \cdot (e_{\mathfrak{m}^*} - e^*) = 0 \bmod q$. $\mathcal{C}$ outputs $e = e_{\mathfrak{m}^*} - e^* \neq 0$ as a solution of SIS instance. It remains to show that $e^* \neq e_{\mathfrak{m}^*}$. There are two cases to consider:

(1) If $\mathcal{A}$ queried a signature on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$, it would receive a signature $e_{\mathfrak{m}^*}$. Because e^*

is viewed as a forged signature on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$, we have $e^* \neq e_{\mathfrak{m}^*}$.

(2) If $\mathcal{A}$ did not query a signature on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$, then for the query to H_2 on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$, $\mathcal{C}$ sampled $e_{\mathfrak{m}^*} \leftarrow \textbf{SampleDom}(1^n)$, stored a tuple $(\mathfrak{u}^*, i^*, \mathfrak{m}^*, e_{\mathfrak{m}^*}, A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*})$, and returned $H_2(\mathfrak{u}^* \| i^* \| \mathfrak{m}^*) = A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*} \bmod q$ to $\mathcal{A}$. By the preimage min-entropy property of the hash family, the min-entropy of $e_{\mathfrak{m}^*}$ given $A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*} \bmod q$ is $\omega(\log n)$. Thus, the signature $e^* \neq e_{\mathfrak{m}^*}$ with overwhelming probability $1 - 2^{-\omega(\log n)}$ [26].

This completes the description of the simulation. It remains to analyze the probability of $\mathcal{A}$ not aborting. For the simulation to complete without fail (write $\neg\texttt{abort}$), we require that all key extract queries on $(\mathfrak{u} \| i)$ have $H_1(\mathfrak{u} \| j) \neq R_j^*$ for some $j \in [i]$ and that $w = i^*$ and $H_1(\mathfrak{u} \| j) = R_j^*$ for all $j = 1, \cdots, i^*$ in the forgery stage. According to the analysis of successful probability in Agrawal et al.'s hierarchical identity-based encryption [31], we can obtain that $\Pr[\neg\texttt{abort}] \geq Q_{H_1}^{-d} / d - negl(n)$, where $negl(n)$ is negligible. Furthermore, if $\mathcal{A}$ has advantage $\varepsilon > 0$, then $\mathcal{C}$ has advantage at least $\varepsilon / (dQ_{H_1}^d) - negl(n)$ in solving the SIS problem instance. This completes our proof.

4.3 The flexibility of key update algorithm

In the proposed forward-secure identity-based signature scheme from lattices, our key update algorithm can provide greater flexibility than those of the existing forward- secure identity-based signature schemes [19-21]. As shown in **Table 1**, for the existing scheme, any user runs the key update algorithms one time and updates her/his secret key from $sk_{\mathfrak{u} \| j}$ at the time period $j \leq d-1$ to $sk_{\mathfrak{u} \| j+1}$ at the next time period $i = j+1 \leq d$. Furthermore, if she/he needs to update her/his secret key to the next multi time period i $(j < i \leq d)$, she/he have to execute the key update algorithm $i - j$ *times*. When each running cost is expensive, the total costs will increase linearly. Fortunately, any user can run our key update algorithm *only one time* to update her/his secret key from the current time period $j \leq d-1$ to the next any time period i $(j < i \leq d)$.

Table 1. Comparison of number of times for running key update algorithm

Schemes	From time period j to $j+1$	From time period j to i
Existing schemes [19-21]	One time	$i - j$ **times**
Our scheme	One time	One time

5. Conclusions

We have revisited the definition of forward-secure identity-based signatures to provide flexible key update, and proposed an identity-based signature scheme from lattices. Furthermore, the proposed scheme is shown to have the properties as follows: forward security with flexible key update, strong unforgeability, and post-quantum security based on lattices. In addition, it is indicated that there exists a serious drawback in the security proof of Zhang et al.'s scheme in appendix. Finally, we remark that a construction of efficient lattice-based forward-secure identity-based signature, which can achieve the strong unforgeability in the standard model, will be our future work.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of Advances in Cryptology---CRYPTO 1984*, LNCS, vol. 0196, pp. 47-53, August 19-22, 1984. Article (CrossRef Link)

[2] K. Paterson, J. Schuldt, "Efficient identity based signatures secure in the standard model," in *Proc. of 11th Australasian Conference Information Security and Privacy---ACISP 2006*, LNCS, vol. 4058, pp. 207-222, July 3-5, 2006. Article (CrossRef Link)

[3] E. Kiltz, G. Neven, "Identity-based signatures, " in *Proc. of Cryptology and Information Security Series on Identity-based Cryptography*, vol. 2, IOS Press, pp. 31-44, 2008. Article (CrossRef Link)

[4] P. Yang, Z. Cao, X. Dong, "Fuzzy identity based signature with applications to biometric authentication," *Computers and Electrical Engineering*, vol.37, no. 4, pp. 532-540, July, 2011. Article (CrossRef Link)

[5] S. Goldwasser, S. Micali, R. Rivest, "A digital signature scheme secure against adaptive chosen-messages attacks," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 281-308, April, 1988. Article (CrossRef Link)

[6] J. An, Y. Dodis, T. Rabin, "On the security of joint signature and encryption, " in *Proc. of Int. Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology---EUROCRYPT 2002*, LNCS, vol. 2332, pp. 83-107, April 28-May 2, 2002. Article (CrossRef Link)

[7] C. Sato, T. Okamoto, E. Okamoto, "Strongly unforgeable ID-based signatures without random oracles, " in *Proc. of 5th Int. Conference of Security Practice and Experience--- ISPEC 2009*, LNCS, vol. 5451, pp. 35-46, April 13-15, 2009. Article (CrossRef Link)

[8] M. Rückert, "Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles," in *Proc. of 3rd Int. Workshop on Post-Quantum Cryptography-PQCrypto 2010*, LNCS, vol. 6061, pp. 182-200, May 25-28, 2010. Article (CrossRef Link)

[9] Z. Liu, Y. Hu, X. Zhang, F. Li, "Efficient and strongly unforgeable identity-based signature scheme from lattices in the standard model," *Security and Communication Networks*, vol.6, pp.69-77, January, 2013. Article (CrossRef Link)

[10] R. Anderson, "Two remarks on public key cryptology (invited lecture)," in *Proc. of 4th ACM Conference on Computer and Communications Security---CCS 1997*, April 1-4, 1997. Article (CrossRef Link)

[11] M. Bellare, S. Miner, "A forward secure digital signature scheme," in *Proc. of 19th Annual Int. Cryptology Conference on Advances in Cryptology--- CRYPTO 1999*, LNCS, vol. 1666, pp. 431-448, August 15-19, 1999. Article (CrossRef Link)

[12] M. Abdalla, L. Reyzin, "A new forward-secure digital signature scheme," in *Proc. of 6th Int. Conference on the Theory and Application of Cryptology and Information Security Advances in Cryptology---ASIACRYPT 2000*, LNCS, vol. 1976, pp. 116-129, December 3-7, 2000. Article (CrossRef Link)

[13] G. Itkis, L. Reyzin, "Forward-secure signatures with optimal signing and verifying," in *Proc. of 21st Annual Int. Cryptology Conference Advances in Cryptology-CRYPTO 2001*, LNCS, vol. 2139, pp. 499-514, August 19-23, 2001. Article (CrossRef Link)

[14] T. Maklin, D. Micciancio, S. Miner, "Efficient general forward-secure signatures with an unbounded number of time periods," in *Proc. of Int. Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology--- EUROCRYPT 2002*, LNCS, vol. 2332, pp. 400-417, April 28-May 2, 2002. Article (CrossRef Link)

[15] X. Boyen, H. Shacham, E. Shen, B. Waters, "Forward-secure signatures with untrusted update," in *Proc. of 13th ACM Conference on Computer and Communications Security---CCS 2006*, pp. 191-200, October 30-November 3, 2006. Article (CrossRef Link)
[16] A. Hülsing, C. Busold, J. Buchmann, "Forward secure signatures on smart cards, " in *Proc. of 19th Int. Conference on Selected Areas in Cryptography---SAC 2012*, LNCS, vol. 7707, pp. 66-80, August 15-16, 2012. Article (CrossRef Link)
[17] M. Abdalla, F. Hamouda, D. Pointcheval, "Tighter reductions for forward-secure signature schemes, " in *Proc. of 16th Conference on Practice and Theory in Public-Key Cryptography---PKC 2013*, LNCS, vol. 7778, pp.292-311, February 26-March 1, 2013. Article (CrossRef Link)
[18] J. Yu, F. Kong, X. Cheng, R. Hao, G. Li, "One forward-secure signature scheme using bilinear maps and its applications," *Information Sciences*, vol. 279, pp. 60-76, September, 2014. Article (CrossRef Link)
[19] Y. Liu, X. Yin, L. Qiu, "ID-based forward secure signature scheme from the bilinear pairings," in *Proc. of the Int. Symposium on Electronic Commerce and Security---ISECS 2008*, pp. 179-183, August 3-5, 2008. Article (CrossRef Link)
[20] J. Yu, R. Hao, F. Kong, X. Cheng, J. Fan, Y. Chen, "Forward-secure identity-based signature: security notions and construction," *Information Sciences*, vol.181, no.3, pp. 648-660, 2011. Article (CrossRef Link)
[21] N. Ebri, J. Baek, A. Shoufan, Q. Vu, "Forward-secure identity-based signature: new generic constructions and their applications," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1, pp. 32-54, 2013. Article (CrossRef Link)
[22] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, October,1997. Article (CrossRef Link)
[23] M. Ajtai, "Generating hard instances of lattices problems (extended abstract)," in *Proc. of the 28th Annual ACM Symposium on the Theory of Computing---STOC 1996*, pp. 99-108, May 22-24, 1996. Article (CrossRef Link)
[24] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. of 26th Int. Colloquium on Automata, Languages and Programming---ICALP 1999*, LNCS, vol. 1644, pp. 1-9, July 11-15, 1999. Article (CrossRef Link)
[25] D. Micciancio, S. Goldwasser, "*Complexity of lattice problems: a cryptographic perspective*," Kluwer Academic Publishers, vol. 671, 2002. Article (CrossRef Link)
[26] C. Gentry, C. Peikert, V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of 40th Annual ACM Symposium on Theory of Computing---STOC 2008*, pp. 197-206, May 17-20, 2008. Article (CrossRef Link)
[27] J. Alwen, C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535-553, April, 2011. Article (CrossRef Link)
[28] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. of the 41st Annual ACM Symposium on Theory of Computing---STOC 2009*, pp. 169-178, May 31-June 2, 2009. Article (CrossRef Link)
[29] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Proc. of 29th Annual Int. Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology---EUROCRYPT 2010*, LNCS, vol. 6110, pp. 523-552, May 30-June 3, 2010. Article (CrossRef Link)
[30] S. Agrawal, D. Boneh, X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. of 29th Annual Int. Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology---EUROCRYPT 2010*, LNCS, vol. 6110, pp. 553-572, May 30-June 3, 2010. Article (CrossRef Link)
[31] S. Agrawal, D. Boneh, X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. of 30th Annual Cryptology Conference on Advances in Cryptology---CRYPTO 2010*, LNCS, vol. 6223, pp. 98-115, August 15-19, 2010. Article (CrossRef Link)

[32] Y. Yao, Z. Li, "A novel fuzzy identity based signature scheme based on the short integer solution problem," *Computers and Electrical Engineering*, vol. 40, no. 6, pp. 1930-1939, August, 2014. Article (CrossRef Link)
[33] S. Garg, C. Gentry, S. Halevi, "Candidate multilinear maps from ideal lattices," in *Proc. of 32nd Annual Int. Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology---EUROCRYPT 2013*, LNCS, vol. 7881, pp. 1-17, May 26-30, 2013. Article (CrossRef Link)
[34] C. Yang, S. Zheng, L. Wang, M. Tian, L. Gu, Y. Yang, "A fuzzy identity-based signature scheme from lattices in the standard model," *Mathematical Problems in Engineering*, vol. 2014, Article ID 391276, 10 pages, 2014. Article (CrossRef Link)
[35] L. Ducas, D. Micciancio, "Improved Short Lattice Signatures in the Standard Model," in *Proc. of 34th Annual Cryptology Conference on Advances in Cryptology---CRYPTO 2014*, LNCS, vol. 8616, pp. 335-352, August 17-21, 2014. Article (CrossRef Link)
[36] X. Zhang, C. Xu, C. Jin, R. Xie, "Efficient forward secure identity-based shorter signature from lattice," *Computers and Electrical Engineering*, vol. 40, no. 6, pp. 1963-1971, August, 2014. Article (CrossRef Link)

Appendix

Recently, Zhang et al. [36] proposed a forward-secure identity-based signature scheme from lattices, and claimed its existential unforgeability under the short integer solution hardness assumption. In the appendix section, we show that there is one serious drawback in Zhang et al.'s security proof, i.e. a challenger can solve an instance of SIS problem without the help of an adversary. The reason is that the challenger knows the initial trapdoor of lattice and is able to compute new trapdoors of any extended lattices by the basis delegation technique.

A.1 Review of Zhang et al.'s scheme

- **Setup**(n): On input a security parameter n, set the parameters m, q, divide the whole lifetime into d time periods, and set two series of Gaussian parameters $\bar{\sigma}=(\sigma_1,\cdots,\sigma_d)$ and $\bar{\delta}=(\delta_1,\cdots,\delta_d)$. Next do:
  (1) Use algorithm **TrapGen** (q,n) to generate a uniformly random $n\times m$ matrix $A_0\in\mathbb{Z}_q^{n\times m}$ with a corresponding short basis T_{A_0} for $\Lambda_q^{\perp}(A_0)$ such that $\|\tilde{T}_{A_0}\|\le\mathcal{O}(\sqrt{n\log q})$.
  (2) Define two hash functions $H_1:\{0,1\}^*\to\mathbb{Z}^{m\times m}$, where the output is distributed as $\mathfrak{D}_{m\times m}$, and $H_2:\{0,1\}^*\to\mathbb{Z}_q^n$.
  (3) Publish the public parameters $pp=\left(A_0,\ H_1,\ H_2\right)$, and keep master secret key $msk=(T_{A_0})$ secret.
- **Extract**($pp, msk, \mathfrak{u}$): On input public parameter pp, a master key msk, a user with identity $\mathfrak{u}$ and an initial time period $i=1$, KGC does:
  (1) Let $R_{\mathfrak{u}\|1}=H_1(\mathfrak{u}\|1)$ and compute $A_{\mathfrak{u}\|1}=A_0\cdot R_{\mathfrak{u}\|1}^{-1} \bmod q$.
  (2) Evaluate $sk_{\mathfrak{u}\|1}\leftarrow$ **BasisDel** $(A_0,R_{\mathfrak{u}\|1},T_{A_0},\sigma_1)$.
  (3) Send a trapdoor $sk_{\mathfrak{u}\|1}$ of $\Lambda_q^{\perp}(A_{\mathfrak{u}\|1})$ to the user over a secure channel.

- **Update**($pp, \mathfrak{u}, i, sk_{\mathfrak{u}\|i-1}$):On input public parameter pp, the current time period $i \le d$, and $sk_{\mathfrak{u}\|i-1}$ which denotes the signing secret key associated with the previous time period $i-1$, the user with identity $\mathfrak{u}$ performs the following steps to update his signing secret key:
  (1) Compute $R_{\mathfrak{u}\|i-1} = H_1(\mathfrak{u}\|i-1)\cdots H_1(\mathfrak{u}\|1)$ and $A_{\mathfrak{u}\|i-1} = A_0 \cdot R_{\mathfrak{u}\|i-1}^{-1} \bmod q$ as the public key in time period $i-1$ with respect to signing secret key $sk_{\mathfrak{u}\|i-1}$.
  (2) Let $R_i = H_1(\mathfrak{u}\|i)$, and compute $sk_{\mathfrak{u}\|i} \leftarrow$ **BasisDel** $(A_{\mathfrak{u}\|i-1}, R_i, sk_{\mathfrak{u}\|i-1}, \sigma_i)$.
  Note that $sk_{\mathfrak{u}\|i}$ is a short basis of $\Lambda_q^{\perp}(A_{\mathfrak{u}\|i})$, where
  $$A_{\mathfrak{u}\|i} = A_{\mathfrak{u}\|i-1} \cdot R_i^{-1} = A_0 \cdot R_{\mathfrak{u}\|i}^{-1} \bmod q \text{ and } R_{\mathfrak{u}\|i} = H_1(\mathfrak{u}\|i)\cdots H_1(\mathfrak{u}\|1).$$
- **Sign**($pp, sk_{\mathfrak{u}\|i}, \mathfrak{m}$):On input public parameters pp and a message $\mathfrak{m} \in \{0,1\}^*$, the signing user $\mathfrak{u}$, whose signing secret key is $sk_{\mathfrak{u}\|i}$ at the current time period $i \le d$, computes $y = H_2(\mathfrak{u}\|i\|\mathfrak{m}) \in \mathbb{Z}_q^n$ and evaluates $e_i \leftarrow$ **SamplePre** $(A_{\mathfrak{u}\|i}, sk_{\mathfrak{u}\|i}, y, \delta_i)$, Note that $A_{\mathfrak{u}\|i} \cdot e_i = y \bmod q$ and e_i is distributed as $\mathfrak{D}_{\Lambda_q^y(A_{\mathfrak{u}\|i}), \delta_i}$. Finally, the signer outputs a signature e_i.
- **Verify**($pp, \mathfrak{u}, \mathfrak{m}, i, e_i$): On input public parameters pp, a user with identity $\mathfrak{u}$, an index of time period i, a message $\mathfrak{m}$ and a candidate signature e_i, the algorithm outputs *accept* if and only if
  $$0 < \|e_i\| \le \delta_i \cdot \sqrt{m} \text{ and } A_{\mathfrak{u}\|i} \cdot e_i = y \bmod q,$$
  where $A_{\mathfrak{u}\|i} = A_0 \cdot R_{\mathfrak{u}\|i}^{-1} \bmod q$, $R_{\mathfrak{u}\|i} = H_1(\mathfrak{u}\|i)\cdots H_1(\mathfrak{u}\|1)$, and $y = H_2(\mathfrak{u}\|i\|\mathfrak{m})$. Otherwise, it outputs *reject*.

A.2 Zhang et al.'s security proof

In this section, we briefly review the key points in their security proof. For further details, please refer to the literature [36].

Given parameters q, n, m, β, find a nonzero integer vector $e \in \mathbb{Z}^m$ such that
$$A_{\mathfrak{u}^*\|i^*} \cdot e = 0 \bmod q \text{ and } \|e\| \le \beta,$$
$$\text{where } A_{\mathfrak{u}^*\|i^*} = A_0 \cdot (R_{\mathfrak{u}^*\|i^*})^{-1} \in \mathbb{Z}_q^{n\times m},\ A_0 \in \mathbb{Z}_q^{n\times m} \text{ and } R_{\mathfrak{u}^*\|i^*} \in \mathbb{Z}^{m\times m}.$$

Zhang et al. used the method of proof by contradiction. Assume that there exists a forger or adversary $\mathcal{A}$ that can forge a signature in the proposed scheme with non-negligible advantage ε. Then a challenger $\mathcal{C}$ will solve an instance of short integer solution problem (as above) with a non-negligible probability ε' by using the ability of the adversary $\mathcal{A}$ .In the simulation, $\mathcal{C}$ is viewed as a simulator who setups the system public parameters and responds to the adversary $\mathcal{A}$'s queries.

- In the phase of setup, $\mathcal{C}$ firstly runs the trapdoor algorithm **TrapGen**(q,n) to generate $A_0 \in \mathbb{Z}_q^{n\times m}$ with corresponding trapdoor $T_{A_0} \in \mathbb{Z}_q^{m\times m}$. Then $\mathcal{C}$ sends the public parameter $pp = A_0$ to $\mathcal{A}$ and keeps the master secret key $msk = T_{A_0}$ by itself.
- In the phase of queries, $\mathcal{C}$ randomly guesses the challenged time period $i^*(1 \le i^* \le d)$ and the challenged identity $\mathfrak{u}^*$ ($\mathfrak{u}^*$ is the l-th query). Then $\mathcal{A}$ may query the random oracles H_1 on $(\mathfrak{u}, i)$ and H_2 on $(\mathfrak{u}, i, \mathfrak{m})$. In order to answer consistently, $\mathcal{C}$ maintains four lists in its local storage, called L_1 list, L_2 list, L_3 list, and L_4 list, respectively. Moreover, $\mathcal{C}$ uses the known trapdoor T_{A_0} to make response to $\mathcal{A}$'s queries, including UserkeyExt queries, Signing secret key queries, Sign queries, and Breakin queries, under the condition of security definition.
- In the phase of forgery, $\mathcal{A}$ outputs a valid signature e^* on a user with identity $\mathfrak{u}^*$, a time period i^*, and a message $\mathfrak{m}^*$. That is to say,
$$A_{\mathfrak{u}^* \| i^*} \cdot e^* = H_2(\mathfrak{u}^* \| i^* \| \mathfrak{m}^*) \bmod q \tag{1}$$
Note that before forging a signature, $\mathcal{A}$ may query the random oracle H_2 on $(\mathfrak{u}^*, i^*, \mathfrak{m}^*)$. Then $\mathcal{C}$ samples $e_{\mathfrak{m}^*} \leftarrow$ **SampleDom**(1^n), stores a tuple $(\mathfrak{u}^*, i^*, \mathfrak{m}^*, e_{\mathfrak{m}^*}, A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*})$ into L_4, and returns $A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*}$ to $\mathcal{A}$. Here, it implies that
$$H_2(\mathfrak{u}^* \| i^* \| \mathfrak{m}^*) = A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*} \bmod q \tag{2}$$
- In the phase of solving the SIS problem instance, by combining the formulas (1) and (2), $\mathcal{C}$ has
$$A_{\mathfrak{u}^* \| i^*} \cdot e^* = A_{\mathfrak{u}^* \| i^*} \cdot e_{\mathfrak{m}^*} \bmod q \Leftrightarrow A_{\mathfrak{u}^* \| i^*} \cdot (e^* - e_{\mathfrak{m}^*}) = 0 \bmod q.$$
Thus, $\mathcal{C}$ can output $e = e^* - e_{\mathfrak{m}^*}$ as a solution of the instance of SIS problem as above. Here, $e^* \ne e_{\mathfrak{m}^*}$ holds except with negligible probability $2^{-\omega(\log n)}$.

A.3 Cryptanalysis of Zhang et al.'s security proof

In Zhang et al.'s security proof, the reduction seems reasonable at first glance. However, in fact, the challenger $\mathcal{C}$ can solve the instance of SIS problem by itself, without the need of $\mathcal{A}$. The reasons are listed as follows.

(1) In the beginning of security proof, the challenger $\mathcal{C}$ is assumed to know the trapdoor T_{A_0} since $\mathcal{C}$ runs the trapdoor algorithm **TrapGen**(q,n) to generate $A_0 \in \mathbb{Z}_q^{n\times m}$ with corresponding trapdoor $T_{A_0} \in \mathbb{Z}_q^{m\times m}$.

(2) And then the simulator $\mathcal{C}$ can easily answer $\mathcal{A}$'s all queries by using the known trapdoor T_{A_0}.

(3) Finally, the challenger $\mathcal{C}$ can evaluate
$$T_{A_{\mathfrak{u}^* \| i^*}} \leftarrow \textbf{BasisDel}(A_0, R_{\mathfrak{u}^* \| i^*}, T_{A_0}, \sigma_{i^*})$$
by using the known trapdoor T_{A_0}, and solve the instance of SIS problem

$$A_{u^* \| i^*} \cdot e = 0 \bmod q$$

by using the trapdoor $T_{A_{u^* \| i^*}}$ of $\Lambda_q^{\perp}(A_{u^* \| i^*})$.

Thus, the analysis shows Zhang et al.'s proof is incorrect.

Xiangsong Zhang is a lecturer of Xi'an Technological University, China. She received her B.S. degree in pure mathematics from Henan Normal University in 2004. She received her M.S. and Ph.D. degrees in applied mathematics from Xidian University, in 2007 and 2011, respectively. Her research interests lie in the fields of applied cryptography and optimization.

Zhenhua Liu is a professor and supervisor of M.S. students of Xidian University. He received his B.S. degree in computational mathematics from Henan Normal University in 2000. He received his M.S. and Ph.D. degrees in applied mathematics and cryptography from Xidian University, in 2003 and 2009, respectively. His research interests include applied cryptography and cloud security.