EasyChair Preprint

№ 13234

Lattice-Based Puncturable Blind Signature Scheme with Bidirectional Security

Jumei Zhang, Zhenhua Liu and Dongdong Yao

May 12, 2024

Lattice-based puncturable blind signature scheme with bidirectional security

No Author Given

No Institute Given

Abstract. Lattice-based blind signature ensures that users can generate a signature on a message while interacting with the signer without revealing any information about the message, and resists quantum attacks. However, the existing lattice-based blind signature schemes did not fully address the threat of key exposure, lacking in their provision for both forward and backward security. In this paper, we propose a lattice-based puncturable blind signature (PBS) scheme that employs puncturable pseudorandom functions to achieve bidirectional security. The implementation of puncturing technique not only enables fine-grained revocation of signing capabilities, effectively safeguarding against key leakage attacks and thereby ensuring bidirectional security, but also markedly decreases the computational complexity involved in key updates, reducing it from $O(n)$ to $O(1)$. Furthermore, the security of the proposed PBS scheme under the SIS hard assumption is validated in the random oracle model.

Keywords: Lattice-based signature · Blind signature · Key exposure · Puncturable pseudorandom function · Bidirectional Security.

1 Introduction

Blind signature, introduced by Chaum [1], allows a user to generate a signature on a message by interacting with a signer in a way that the signer gains no information about the signed message. This property, known as blindness, effectively preserves privacy in applications such as electronic auctions, e-cash systems, and electronic voting [2–4]. With the continuous development of quantum computers, traditional public-key cryptographic algorithms, which are based on mathematical challenges such as integer factorization and discrete logarithm, can be easily solved by quantum computing. This renders cryptographic algorithms based on these mathematical problems insecure in the quantum era. To address this challenge, cryptographic research is shifting its focus towards the development of quantum-resistant methodologies. In the field of cryptography, the techniques widely considered to be resistant to quantum attacks include hash-based, code-based, multivariate-based, lattice-based, and isogeny-based cryptographic algorithms. Among these, lattice-based cryptography is currently regarded as the most promising.

The progressing of lattice-based cryptography has also given birth to the development of lattice-based blind signature. Rücker et al. [5] introduced the

first lattice-based blind signature scheme, which is built upon Lyubashevsky's identification scheme stemming from the short integer solution (SIS) hardness assumption [6]. Chen et al. [7] used the matrix-vector-blinding technique to create two hierarchical ID-based schemes resilient to quantum attacks with improved efficiency and shorter keys. Fuchsbauer et al. [8] introduced structure-preserving signature on equivalence classes for efficient round-optimal blind signature in the standard model. Zhang et al. [9] combined proxy signature and blind signature to develop a novel lattice-based identity-specific proxy blind signature scheme independent of random oracles. Gao et al. [10] designed an identity-based blind signature scheme resilient to selective identity and chosen-message attacks over the SIS assumption under the random oracle model, ensuring unconditional blindness. Subsequently, Le et al. [11] constructed the first lattice-based blind ring signature scheme, which were provably secure under the SIS hardness assumption in the random oracle model. Furthermore, Alkadri et al. [12] enhanced these methods based on the ring-SIS hardness assumption. Beullens et al. [13] presented a practical, efficient, and round-optimal blind signature scheme based on standard lattice assumption.

In addition to resisting quantum computing attacks, blind signature also needs to consider the risk of key leakage. For instance, if a signer's secret key is compromised or stolen, any individual possessing this secret key can illicitly forge a signature attributed to the said signer [14]. Fortunately, forward security ensures the validity of signature that was signed prior to the disclosure of secret key [15, 16]. In other words, even if the current secret key of the signer is compromised, the previous sessions maintain their security.

Most forward-secure blind signature schemes were based on bilinear groups or integer lattices. Duc et al. [17] presented the first blind signature scheme with forward security based on bilinear pairings, followed by Sherman et al. [18] and Yu et al. [19]. In a recent study, Yang et al.[20] presented a novel forward-secure lattice-based Fiat-Shamir signature scheme that boasts a reduced secret key size and efficient key evolution. In a significant stride, Le et al. [21] proposed the first forward-secure blind signature scheme in the lattice setting, utilizing a binary tree structure for key updates.

1.1 Motivation and Contribution

For digital signature, the best solution to the risk of key disclosure is to have both forward and backward security. In other words, even if the signing secret key is leaked, previous and future signatures are not affected, thus improving the security and stability of the entire signature system. Similar work has been done on digital signature [22] and [23], which are either vulnerable to attacks or require substantial improvements in performance. However, there is no research work on backward security of blind signature. Inspired by Xiang et al.'s bilateral-secure signature [22] that was based on the work of Le et al. [21], this paper gives the definition of bidirectional security, which satisfies both forward and backward security, and proposes a blind signature scheme with bidirectional security based on the standard lattice hypothesis. The proposed scheme uses a

puncturable pseudorandom function to achieve fine-grained retractable key signature capability. Furthermore, the proposed scheme can resist the key exposure attack and ensure its own security. The contributions are as follows.

1. We introduce the concept of *bidirectional security*, which "encapsulates" both forward and backward security.
2. We propose a puncturable blind signature scheme based on lattices. Under the random oracle model and the SIS hardness assumption, the proposed scheme is proved to satisfy *bidirectional security*.
3. Compared to the existing forward-secure anti-key leakage methods, the computational complexity of key update is reduced from O(n) to O(1), significantly improving computational efficiency while maintaining the security of the proposed scheme.

2 Preliminaries

In this section, we provide some notations and introduce some fundamental knowledge.

2.1 Notations

The notations used in this paper are given in Table 1.

Table 1. Parameters of the proposed scheme

Parameters	**Descriptions**
λ	Security parameters
$n, m; q; l$	Dimension parameters; prime modulus; positive integer
$\sigma_1, \sigma_2, \sigma_3$	Standard deviations for different gaussian distributions
$\mathbf{x}, \mathbf{v}; \mathbf{a}, \mathbf{b}, \mathbf{d}; \mathbf{r}_1, \mathbf{r}_2$	Vector; random blinding factors; random vector
$\mathbf{c}, \mathtt{com}$	Commitment
$\mathbf{A}, \mathbf{F}, \mathbf{K}, \mathbf{S}, \mathbf{T}, \mathbf{W}$	Public matrices
$(\mathbf{e}, \mathbf{z})$	The blinded message and its blind signature
$[l]$	The set $\{1, 2, \cdots, l\}$
$\|\| \cdot \|\|$	The norm of a vector or a matrix
$\mathbf{M}[i]$	The i-th column of $\mathbf{M}$

2.2 Lattice and hard assumption

Lattice. Integer lattice is a discrete subgroup of $\mathbb{Z}^m$. Formally, a lattice denoted by $\mathcal{L}$ with the space $\mathbb{Z}^m$ is defined as

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) := \{\sum_{i=1}^{n} \mathbf{b}_i x_i : x_i \in \mathbb{Z}, i = 1, \cdots, n\},$$

where $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}^m$, $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_n] \subseteq \mathbb{Z}^{m \times n}$, $\mathbf{B}$ is the basis of $\mathcal{L}$, n is the rank of $\mathcal{L}$. When $n = m$, we say $\mathcal{L}$ is a full-rank lattice.

Let $\mathbf{A} \in \mathbb{Z}^{m \times n}$ and $\mathbf{u} \in \mathbb{Z}_q^n$, define two lattices:

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{Ae} = 0 \ (\text{mod q})\},$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{Ae} = \mathbf{u} \ (\text{mod q})\}.$$

If $\mathbf{v}$ belongs to the lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A})$, then $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ can be expressed as the sum of $\Lambda_q^{\perp}(\mathbf{A})$ and $\mathbf{v}$. When $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_k\} \subseteq \mathbb{R}^m$, we have $\|\mathbf{S}\| = \max\|\mathbf{s}_i\|$. $\tilde{\mathbf{S}} = \{\tilde{\mathbf{s}}_1, \cdots, \tilde{\mathbf{s}}_k\}$ is the Gram-Schmidt orthogonalization of vectors $\mathbf{s}_i (i \in [1, k])$.

Definition 1. ***The Shortest Independent Vectors Problem, SIVP*** ([24]) Let $\Lambda \subseteq \mathbb{Z}^m$, where the dimension is n and full rank. $\mathbf{B}$ represents a set of basis vectors. If $\mathbf{S}$ is a solution to the SIVPγ problem, then we have $|\mathbf{S}| \leq \gamma(n) \cdot \lambda n(\Lambda(\mathbf{S}))$.

Definition 2. ***Gaussian Distribution*** ([24]) Let $\mathbf{v}$ and s be vectors of the lattice Λ and a positive parameter on $\mathbb{R}$, respectively. $\rho_{s,\mathbf{v}}(\mathbf{x}) = \exp(-\pi\|\mathbf{x} - \mathbf{v}\|^2/\mathrm{s}^2)$ and $\rho_{s,\mathbf{v}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{s,\mathbf{v}}(\mathbf{x})$ are the discrete Gaussian distributions over this lattice with center at vector $\mathbf{v}$ and parameter value s for any $\mathbf{x}$.

For simplicity, we refer to ρ_s and $D_{\Lambda,s}$ as $\rho_{0,s}$ and $D_{\Lambda,s,0}$, respectively. In the case of $s = 1$, we may substitute ρ for ρ_1. Additionally, we can express $D_{s,\mathbf{v}}^m$ and Ds^m as $D_{\mathbb{Z}^m,s,\mathbf{v}}$ and $D_{\mathbb{Z}^m,s}$ correspondingly.

Lemma 1. ([25]) For any $\mathbf{v} \in \mathbb{Z}^m$, if s is equal to α times the norm of $\mathbf{v}$, the probability of $\Pr[D_s^m(\mathbf{x})/D_{s,\mathbf{v}}^m(\mathbf{x}) \leq e^{12/\alpha + 1/(2\alpha^2)}]$ is at least $1 - 2^{-100}$, where α is a positive number and $\mathbf{x}$ is selected from D_s^m.

Trapdoors and Trapdoor Delegation. Alwen et al.[24] proposed an algorithm to generate a compact basis $\mathbf{T_A}$ for the lattice $\Lambda_q^{\perp}(\mathbf{A})$, by sampling a matrix $\mathbf{A}$ from the set of matrices uniformly distributed in $\mathbb{Z}_q^{n \times m}$.

Theorem 1. ([24]) Assuming an odd value q (where $3 \leq q$). One matrix $\mathbf{A}$ approximates a uniformly distributed integer matrix modulo q, while the other matrix $\mathbf{T_A}$ serves as a basis for the orthogonal complement lattice $\Lambda_q(A)$. The generated $\mathbf{T_A}$ satisfies two conditions: its norm $\|\tilde{\mathbf{T}}_\mathbf{A}\|$ is upper bounded by $\mathrm{O}(\sqrt{n \ \log(q)})$, and its norm $\|\mathbf{T_A}\|$ is upper bounded by $\mathrm{O}(n \ \log(q))$. These conditions hold with almost negligible probability when considering parameter n.

Lemma 2. ([25]) Let m, n, q be positive integers, where q is a prime number and $m \geq 2n \log q$. Then, for almost all matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (except for a fraction of only $2q^{-n}$), and for any value of s greater than or equal to $\omega(\sqrt{\log m})$, the distribution of $\mathbf{u} := \mathbf{Ae}(\text{mod } q)$ is very close to being uniformly distributed over $\mathbb{Z}_q^n$. Here, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ represents the random variable drawn from the discrete Gaussian distribution. Additionally, when given that $\mathbf{Ae} = \mathbf{u}(mod\ q)$ holds true, the conditional distribution of $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,s}$ becomes exactly equal to $D_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),s}$.

Lemma 3. ([26]) Consider the matrix $[\mathbf{A}_1\|\mathbf{A}_2\|\mathbf{A}_3]$, denoted as $\mathbf{A}$. Assuming that $\mathbf{T}_{\mathbf{A}_2}$ is a basis for $\Lambda_q^{\perp}(\mathbf{A}_2)$, there exists a deterministic algorithm called ExtBasis. When given the inputs of the matrix $\mathbf{A}$ and the basis $\mathbf{T}_{\mathbf{A}_2}$, ExtBasis algorithm outputs a basis $\mathbf{T}_{\boldsymbol{A}}$ for the lattice $\Lambda_q^{\perp}(\boldsymbol{A})$, such that the norm of $\widetilde{\mathbf{T}}_{\mathbf{A}}$ is equal to the norm of $\widetilde{\mathbf{T}}_{\mathbf{A}_2}$.

Lemma 4. (**Rejection Sampling** [25]) Let V be a given set, and $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ represent probability distributions. Assume the existence of a family of probability distributions indexed by $\mathbf{v} \to V$, denoted as $g_{\mathbf{v}} : \mathbb{Z}^m \to \mathbb{R}$, such that for all $\mathbf{v} \in V$ and $\mathbf{z} \in \mathbb{Z}^m$, there exists a constant $M \in \mathbb{R}$ satisfying the inequality $M \cdot g_{\mathbf{v}}(\mathbf{z}) \geq f(\mathbf{z})$. Given this condition, the output distributions of the following two algorithms cannot be distinguished from each other:

1. $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow g_{\mathbf{v}}$, output $(\mathbf{z}, \mathbf{v})$ with a probability of $f(\mathbf{z})/(M \cdot g_{\mathbf{v}}(\mathbf{z}))$.
2. $\mathbf{v} \leftarrow h$, $\mathbf{z} \leftarrow f$, output $(\mathbf{z}, \mathbf{v})$ with a probability of $1/M$.

Definition 3. ([25]) Given a random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, find a vector $\mathbf{z} \in \mathbb{Z}^m \backslash \{0\}$ such that $\mathbf{Az}{=}0(\text{mod } q)$ and $\|\mathbf{z}\| \leq \beta$.

Lemma 5. ([11, 25]) For $d \gg q^{m/n}$, the $\text{SIS}_{q,n,m,\beta}$ distribution is statistically close to uniform over $\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$. Given $(\mathbf{A}, \mathbf{u})$ from the $\text{SIS}_{q,n,m,\beta}$ distribution, there are many possible solutions $\mathbf{s}$ satisfying $\mathbf{As} = \mathbf{u}$.

Commitment Functions. ([9, 11]) A commitment function $\texttt{com}$ maps a pair of strings $(\mu, \mathbf{d}) \in \{0,1\}^n \times \{0,1\}^n$ (called committed string) to a commitment string $C := \texttt{com}(\mu,\ \mathbf{d}) \in \{0,1\}^n$.

2.3 Puncturable pseudorandom function

As a special form of constrained pseudorandom functions [27–29], Boneh et al. [30] constructed the puncturable pseudorandom functions (PRFs) from standard lattice assumptions. Generally, PRFs has three probabilistic polynomial time (PPT) algorithms PRFs=(F.Setup, F.Puncture, F.Eval), where

- F.Setup$(1^\lambda) \to k$, where $k \in \mathcal{K}$ is the key space.
- F.Puncture$(k, (x, sta)) \to k_x$, where $x \in \mathcal{X}$, $k_x \in \mathcal{K}_p$, sta is the status of x, and $\mathcal{K}_p$ is a punctured key space.
- F.Eval$(k_x, (x', sta) =$ F$(k, (x', sta))$ where $x' \neq x$, $k_x \in \mathcal{K}_{\mathcal{P}}$, $x' \in \mathcal{X}$, and $k_x \leftarrow$ F.Puncture$(k, (x, sta))$. If $x' = x$, then abort.

3 Formal definition of puncturable blind signature

3.1 Syntax of puncturable blind signature

We present a formal framework for the puncturable blind signature, which consists of five algorithms: **Setup**, **KeyGen**, **Punc**, **BSign**, and **Verify**. The framework is illustrated in Figure 1 and described as follows:

- **Setup** $(1^\lambda) \to (pp, vk, sk_{init})$. vk is a verifiable public key and sk_{init} is an initial secret key.
- **KeyGen** $(pp, vk, sk_{init}, i) \to sk_i$. sk_i is the secret key at the time point i, where $i \in [1, \cdots, \|T\|]$ and $\|T\|$ is the total number of time period.
- **Punc** $(sk_{init}, \mu, i) \to sk_i'$. μ is a message that is to be signed, and the secret key sk_i is punctured to sk_i' at the time point i. And then set $sk_i \leftarrow sk_i'$.
- **BSign**$(pp, sk_i, \mu, i) \to (e, z)$. An interactive process requires the cooperation between a user $\mathcal{U}$ and a signer $\mathcal{S}$ as follows.
  1. Blinding:$(\mu, a) \to e$. $\mathcal{U}$ blinds the message μ with a blinding factor a to get e, and then sends it to $\mathcal{S}$.
  2. Signing: $(e, sk_i) \to z$. $\mathcal{S}$ generates a corresponding blind signature z by using the punctured secret key sk_i, and sends z to $\mathcal{U}$.
  3. Unblinding: $(z, a) \to z'$. $\mathcal{U}$ uses the blinded signature z with the blind factor a to derive a valid signature z' for the original message μ.
- **Verify** $(pp, vk, i, e, z) \to \{0, 1\}$. Using the public key vk to verify the signature z of the blinded message e, the algorithm returns 1 if the signature is valid, and 0 otherwise.

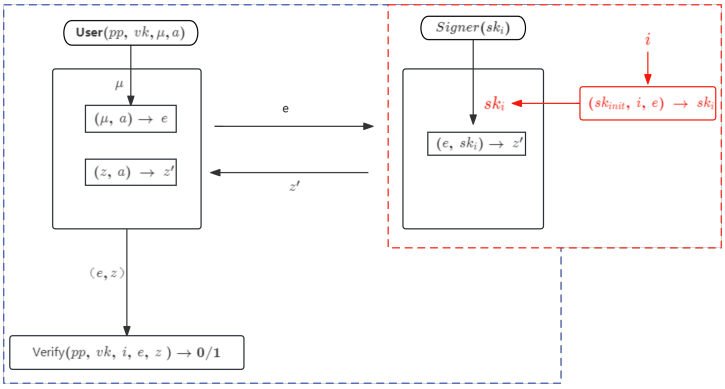

Fig. 1. Puncturable blind signature: the blind signature process is encapsulated by the blue border, whereas the secret key puncturable process is delineated by the red border.

The correctness implies that for any $(pp, vk, sk_{init}) \leftarrow$ **KeyGen**(1^λ) and $(e, z) \leftarrow$ **BSign**(pp, sk_i, e, i), the blindeed message/signature pair are valid, and **Verify** algorithm fails with negligible probability. In other words:

$$\Pr[\textbf{Verify}(pp, vk, i, e, z) = 1] = 1 - \text{negl}(\lambda).$$

3.2 Security Notions for Puncturable Blind Signature

The security of a blind signature scheme encompasses the properties of *one-more unforgeability*, which prevents a malicious user $\mathcal{U}^*$ from producing more

signatures than those requested, and *blindness*, which ensures that a malicious signer $\mathcal{S}^*$ cannot obtain any useful information about the user's message [21, 33, 34, 39–41]. A PBS scheme, which introduces the puncturable property to blind signature, also adheres to these security properties.

Definition 4. *(**Blindness**) Let Adv^{blind}_{PBS} be the advantage of $\mathcal{S}^*$ wins in the blindness game $Blind^{\mathcal{S}^*}_{PBS}$,*

$$Adv^{blind}_{PBS,\mathcal{S}^*} = \Pr[Blind^{\mathcal{S}^*}_{PBS}(b' = b)] \leq 1/2 + negl(\lambda).$$

The PBS scheme is blindness if $Adv^{blind}_{PBS,\mathcal{S}^*}$ is exactly $1/2$.

A blindness game $Blind^{\mathcal{S}^*}_{PBS}$ between a challenger $\mathcal{C}$ and a malicious $\mathcal{S}^*$ consists of three phases as follows.

- **Initialization**. A malicious $\mathcal{S}^*$ picks a security parameter λ, generates the common parameters pp, an initial secret key sk_{init}, and a verifying key vk.
- **Challenge**. $\mathcal{S}^*$ selects two messages, denoted as μ_0 and μ_1, randomly chooses a bit b from $\{0, 1\}$, and then sends a message/signature pair $(\mathbf{e}_b, \mathbf{z}_b)$ corresponding to μ_b to $\mathcal{C}$. The challenger, in turn, sends a pair of messages (μ_b, μ_{1-b}) to $\mathcal{S}^*$. In this interaction, $\mathcal{S}^*$ serves as the role of the signer. Using its knowledge, $\mathcal{S}^*$ ultimately obtains two message/signature pairs: the original $(\mathbf{e}_b, \mathbf{z}_b)$ and a newly created $(\mathbf{e}_{1-b}, \mathbf{z}_{1-b})$ corresponding to μ_{1-b}.
- **Output**. $\mathcal{S}^*$ outputs a bit $b' \in \{0, 1\}$, and wins the game if $b' = b$.

Definition 5. *(**One-More Unforgeability**) Let $Adv^{Omf}_{PBS,\mathcal{U}^*}$ be the advantage of $\mathcal{U}^*$ that gets $l + 1$ valid message/signature pairs.*

$$Adv^{Omf}_{PBS,\mathcal{U}^*} = \Pr[Omf^{U^*}_{PBS}(\text{Verity}(pp, vk, i, \boldsymbol{e}, \boldsymbol{z}) = 1] \leq \text{negl}(\lambda).$$

The PBS scheme is one-more unforgeability if $Adv^{Omf}_{PBS,\mathcal{U}^*}$ is $\text{negl}(\lambda)$.

The one-more unforgeability game $Omf^{U^*}_{PBS}$ consists of three phases [21].

During the time period i, the malicious entity $\mathcal{U}^*$ is permitted to make polynomially bounded random oracle queries and signing queries adaptively. Within the time period i, the forger executes a puncturing query to acquire the new secret key $\mathbf{S}'_i$ corresponding to the time period i.

- **Setup**. The forger $\mathcal{U}^*$ sends security parameter λ to the challenger $\mathcal{C}$. $\mathcal{C}$ runs $\text{Setup}(1^\lambda)$ to generate pp and key pair (vk, sk_{init}), and sends pp and vk back to $\mathcal{U}^*$, while keeping sk_{init} secret.
- **Queries**.
  1. Puncturing key query $Q_K(i)$ during the time period i: if $\bar{i} \leq \|T\| - 1$, the challenger retrieves and provides the new secret key $\mathbf{S}'_i$. Otherwise $\bar{i} = \|T\| - 1$, an empty string is returned as $sk'_{\|T\|-1}$.
  2. Puncturing signature query $Q_S(i, \mathcal{V})$ during the time period i, where $i \leq \|T\| - 1$: the challenger responds with a randomly generated value.

3. Break-out query $Q_B(\bar{i})$, which can only be made once: the challenger sends an empty string, effectively ending the query phase and transiting to the output phase.

– **Forge**. $\mathcal{U}^*$ outputs o forgery message/signature pair $(\mathbf{e}^*, \mathbf{z}^*)$ at the time period $i^* < \bar{i}$. It wins if $Q_S(\mathbf{e}^*, \mathbf{z}^*)$ has never been queried, and $(\mathbf{e}^*, \mathbf{z}^*)$ is valid.

3.3 Bidirectional Security

In the context of digital signature security, forward and backward security are pivotal, ensuring the safety of past and future signature, respectively, despite a current key breach. Nevertheless, modern digital landscapes demand more robust security measures. This paper introduces a novel security attribute: bidirectional security, amalgamating forward and backward security. Significantly, we showcase the proposed puncturable blind signature scheme, leveraging "puncturable" techniques to deliver bidirectional security. This adaptability equips it to address more intricate security challenges. Essentially, if a key is compromised at any point, the proposed scheme can selectively invalidate it, maintaining the integrity of all other keys throughout their lifespan. This enhances the signature system's resilience against a wider array of attacks.

Definition 6. *(**Bidirectional Security**) A puncturable blind signature scheme is bidirectional security if for any malicious user $\mathcal{U}^*$, a valid message/signature pair cannot be forged successfully at the puncture point $j \neq i$. Let $Adv_{PBS}^{Bidi}(\mathcal{U})$ be the advantage of $\mathcal{U}^*$ that gets a valid message/signature pair.*

$$Adv_{PBS}^{Bidi}(\mathcal{U}) := \Pr[Bidi_{PBS}^{U^*}(pp, vk, j, \boldsymbol{e}, \boldsymbol{z}^*) = 1] = negl(\lambda).$$

A puncturable blind signature scheme is bidirectional security if $Bidi_{PBS}^{\mathcal{U}^*}$ is $negl(\lambda)$.

The bidirectional security game $Bidi_{PBS}^{\mathcal{U}^*}$ comprises three distinct phases.

– **Setup**
1. Firstly, the total lifetime of keys, denoted by T, is defined.
2. Then, the challenger $\mathcal{C}$ declares the puncture point t_i.
3. Next, $\mathcal{C}$ selects a security parameter λ and computes the verification key vk and the initial secret key sk_{init}. The public parameters pp and vk are subsequently submitted to the malicious user $\mathcal{U}^*$, while keeping sk_{init} private.
4. Additionally, the set representing the lifetime of the puncturable blind signature is established as $T = \{t_0, t_1, \ldots, t_{T-1}\}$, where each t_j corresponds to a specific secret key for the respective time period, except for the puncture point t_i.

– **Queries**
1. In this phase, the set T is partitioned into three subsets: $T_{\text{pre}} = \{t_0, t_1, \cdots, t_{i-1}\}$, $\{t_i\}$, and $T_{\text{post}} = \{t_{i+1}, \cdots, t_{T-1}\}$.

2. The malicious user $\mathcal{U}^*$ is permitted to perform signature queries adaptively at any time point other than the puncture point t_i. These queries are limited to time period within T_{pre} and T_{post}.
3. During the query phase, $\mathcal{U}^*$ can make up to $\|T\| - 2$ signature queries in a random and sequential manner from either T_{pre} or T_{post}.

– **Forgery**
1. Finally, in the forgery phase, the malicious user $\mathcal{U}^*$ generates a valid message/signature pair $(\mathbf{e}, \mathbf{z}^*)$ at the time period t_k.
2. $\mathcal{U}^*$ is considered successful if the pair satisfies the following conditions:
   (a) The verification process succeeds, i.e., $\text{Verify}(pp, vk, k, \mathbf{e}, \mathbf{z}^*) = 1$, where k is a time period in T but not equal to i.
   (b) The time period k has not been queried before and belongs to either T_{pre} or T_{post}.

4 The puncturable blind signature scheme

We utilize a binary tree structure for key generation. Each time period is allocated to a leaf node, and keys are derived from the root node using the ExtBasis delegation mechanism. The lattice-based puncturable blind signature scheme comprises five main algorithms: PBS={Setup, KeyGen, Punc, BSign, Verify}. Figure 2 depicts the signing process.

The algorithm initializes the matrix $\mathbf{F}_{t_i}$ by concatenating the matrices $\{\mathbf{A}_0, \mathbf{A}_1^{t_1} \cdots \mathbf{A}_l^{t_l}\}$ and subsequently generates the secret key $\mathbf{S}_{t_i} \in \mathbb{Z}^{(l+1)m \times k}$ for a designated time period t, leveraging the trapdoor $\mathbf{T}_{\mathbf{F}_{t_i}}$ and associated parameters σ and $\mathbf{K}$. Subsequently, random vectors $\mathbf{r}_1 \in \mathbb{Z}^{lm}$ and $\mathbf{r}_2 \in \mathbb{Z}^m$ are sampled from discrete Gaussian distributions $D_{\sigma_2}^{lm}$ and $D_{\sigma_2}^m$, respectively. These vectors are concatenated to form $\mathbf{r} \in \mathbb{Z}^{(l+1)m}$ sampled from $D_{\sigma_2}^{(l+1)m}$. The ciphertext $\mathbf{x} \in \mathbb{Z}_n^q$ is then computed through multiplication with $\mathbf{F}_{t_i}$ and reduction modulo n. Finally, the resulting ciphertext $\mathbf{x}$ is transmitted to the user.

– **Setup**$(1^\lambda, 1^l)$: For a security parameter λ and a binary tree depth l, this phase initializes with the selection of a prime q, dimensions n, k, and other parameters. Matrices $\mathbf{K}$ and $\mathbf{A}_i^b$ are randomly generated from $\mathbb{Z}_q^{n \times k}$ and a trapdoor pair $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0})$ is obtained via TrapGen. Additionally, a one-way hash function H and a computationally binding, statistically hiding commitment function `com` are defined. The final output consists of public parameters pp, a public key vk encompassing the matrices, and an initial secret key $\mathbf{S}_{\text{init}}$ set as the trapdoor $\mathbf{T}_{\mathbf{A}_0}$.
– **KeyGen**$(pp, vk, \mathbf{S}_{init})$: This algorithm mainly refer to the method in [21]. As mentioned earlier, based on binary-tree data structure, for any leaf tt, the secret key can be computed by its ancestor's secret key. It finally output the matrix $\mathbf{F}_{\text{t}_\text{i}} = [\mathbf{A}_0 \| \mathbf{A}_1^{t_1} \| \cdots \| \mathbf{A}_l^{t_l}] \in \mathbb{Z}_q^{n \times (l+1)m}$ as public key and $\mathbf{S}_{t_i}$ as secret key which is computed by SampleKey, at the time period $\mathbf{t} = (t_1, \cdots, t_l)$.

- **Punc**$(\mathbf{S}_{init}, t_i, \mathbf{e})$: The puncture algorithm is performed locally by the signer without interactions. In fact, Punc is a key updating algorithm, we can use the puncturable PRF get the new secret key that we required: $\mathbf{S}_{t_i} \leftarrow$ F.Puncture$(\mathbf{S}_{init}, t_i, \mathbf{e}')$. Finally, we have the punctured key $\mathbf{S}_{t_i}$.
- **BSign**$(pp, vk, t_i, \mathbf{e}, \mathbf{z})$: This is an interactive blind signing algorithm between the signer and the user, Figure 2 illustrates the interactive phase. At the time period t_i, the blinded message $\mathbf{e}$ is provided to the signer, who then returns the blinded signature $\mathbf{z}$.
- Verify$(pp, vk, t_i, \mathbf{e}, \mathbf{z})$: For the time $\mathbf{t} = (t_1, \cdots, t_l)$, the algorithm accepts signature $\mathbf{z}$ on the message $\mathbf{e}$, output 1 if and only if $\|\mathbf{z}'\| \leq \sigma_3\sqrt{(1+l)m}$ and $\widehat{\mathbf{e}} = \mathbf{e}'$, where

$$\Sigma = (\mathbf{d}, \mathbf{e}', \mathbf{z}'),\ \mathbf{F}_{\mathrm{t_i}} = [\mathbf{A}_0 \| \mathbf{A}_1^{t_1} \| \cdots \| \mathbf{A}_l^{t_l}] \in \mathbb{Z}_q^{n\times(l+1)m},$$

$$\mathbf{c} := \mathtt{com}(\mu + \mathbf{d}), \quad \widehat{\mathbf{e}} := H(\mathbf{F}_{t_i}\mathbf{z}' - \mathbf{K}\mathbf{e}' \mod q, \mathbf{c}).$$

  Otherwise, return 0.

5 Security analysis of PBS scheme

5.1 Correctness

Theorem 2. *The puncturable blind signature scheme is correct after at most* e^2 *restarts with probability at least* $1 - 2^{-100}$.

The proof of correctness can be accessed in Appendix A.

5.2 Blindness

Theorem 3. *The PBS scheme is blindness, if* com *is a statistically hiding commitment and* H *is a one-way and collision-resistant hash function.*

The proof of blindness can be accessed in Appendix B.

5.3 One-more unforgeability

Theorem 4. (**One-more Unforgeability**) *Suppose there is an adversary* (q_H, q_S, δ)*-forger* $\mathcal{A}$*, which is against one-more unforgeability of the puncturable blind signature with non negligible probability* δ*,* q_H *and* q_S *are the number of queries to the random oracle* H *and blind signing oracle, respectively. Then there exists a polynomial-time algorithm* $\mathcal{B}$*, that can find a solution to* l_2-$\mathrm{SIS}_{q,n,(1+2l)m,\beta}$ problem with $\beta = \max\{(2\sigma_3 + 2\sigma\sqrt{k})\sqrt{(1+l)m}, (2\sigma_3 + \sigma_2)\sqrt{(1+l)m}\}$.

The proof of one-more unforgeability can be accessed in Appendix C.

The signer $\mathbf{S}(pp, S_{init}, t_i)$	**The User** $\mathbf{U}(pp, vk, \mu)$
Phase 1:	
01. $\mathbf{F}_{t_i} := [\mathbf{A}_0 \| \mathbf{A}_1^{t_1} \| \cdots \| \mathbf{A}_l^{t_l}]$	
02. $\mathbf{S}_{t_i} \in \mathbb{Z}^{(l+1)m \times k} \leftarrow \text{SampleKey}(\mathbf{F}_t, \mathbf{T}_{\mathbf{F}_{t_i}}, \sigma, \mathbf{K})$	
03. $\mathbf{r}_1 \in \mathbb{Z}^{lm} \leftarrow D_{\sigma_2}^{lm}$, $\mathbf{r}_2 \in \mathbb{Z}^m \leftarrow D_{\sigma_2}^m$	
04. $\mathbf{r} = (\mathbf{r}_1 \| \mathbf{r}_2) \in \mathbb{Z}^{(l+1)m} \xleftarrow{\$} D_{\sigma_2}^{(l+1)m}$	
05. $\mathbf{x} = \mathbf{F}_{t_i} \cdot \mathbf{r} \in \mathbb{Z}_n^q$	
06. Send $\mathbf{x}$ to the user	
	Phase 2:
	07. $\mathbf{F}_{t_i} := [\mathbf{A}_0 \| \mathbf{A}_1^{t_1} \| \cdots \| \mathbf{A}_l^{t_l}]$
	08. $\mathbf{a} \leftarrow D_{\sigma_2}^{(l+1)m}$, $\mathbf{b} \leftarrow D_{\sigma_2}^k$
	09. $\mathbf{d} \xleftarrow{\$} \{0,1\}^n$, $\mathbf{c} := \mathtt{com}(\mu + \mathbf{d})$ $\mu = \mathbf{F}_{t_i}\mathbf{a} + \mathbf{x} + \mathbf{K}\mathbf{b} (\text{mod q})$
	10. $\mathbf{e}' = \mathtt{H}(\mu, \mathbf{c}) \in (\mathcal{R})_{\mathtt{H}}^k, \mathbf{e} := \mathbf{e}' + \mathbf{b}$
	11. Output $\mathbf{e}$ with probability $\min = \{\frac{D_{\sigma_1}^m(\mathbf{e})}{M_1 \cdot D_{\sigma_1, \mathbf{e}'}^m(\mathbf{e})}, 1\}$
	12. Send $\mathbf{e}$ back to the signer.
Phase 3:	
(13. Puncture: $\mathbf{S}_{t_i} \leftarrow \text{Punc}(\mathbf{S}_{init}, t_i, \mathbf{e})$)	
13. $\mathbf{z} = \mathbf{r} + \mathbf{S}_{t_i}\mathbf{e}$	
14. Output $\mathbf{z}$ with probability $\min = \{\frac{D_{\sigma_2}^{(l+1)m}(\mathbf{e})}{M_2 \cdot D_{\sigma_1, S_{t_i}\mathbf{e}}^{(l+1)m}(\mathbf{z})}, 1\}$	
15. Send $\mathbf{z}$ to the user	
	Phase 4:
	16. $\mathbf{z}' = \mathbf{z} + \mathbf{a}$
	17. If $\mathbf{z}'$ with probability $\min = \{\frac{D_{\sigma_3}^{(l+1)m}(\mathbf{e})}{M_2 \cdot D_{\sigma_3, \mathbf{z}}^{(l+1)m}(\mathbf{z}')}, 1\}$
	(17′. else $\perp$ and restart from Phase 1)
	18. Send $\Sigma = (t_i, \mu, (\mathbf{d}, \mathbf{e}', \mathbf{z}'))$ to the signer.
	19. Output the view: $\Sigma = (t_i, \mu, (\mathbf{d}, \mathbf{e}', \mathbf{z}'))$
Phase 5:	
20. Output: the view $\mathcal{V} = (t_i, \mathbf{r}, \mathbf{e}, \mathbf{z})$	

Fig. 2. The signing process of PBS.

5.4 Bidirectional security

In the puncturable blind signature scheme, a bidirectional security model is established through interactions between a malicious user $\mathcal{U}^*$ and a challenger $\mathcal{C}$ following the selection of a puncturing point by $\mathcal{U}^*$. The essence of the bidirectional security proof lies in the inability of $\mathcal{U}^*$ to forge signature for chosen-message at the non-puncturing time nodes, even after the puncturing event, thus ensuring the PBS system's bidirectional security. More precisely, within this signature scheme, the compromise of the secret key at the i-th time period precludes an adversary from forging signatures for messages before or after this period. The bidirectional security is depicted in Figure 3.

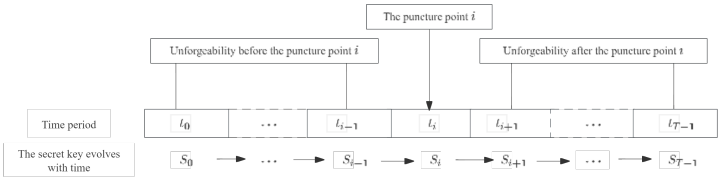

Fig. 3. Bidirectional security

Theorem 5. *The bidirectional security of the puncturable blind signature scheme is defined such that the probability that a malicious user $\mathcal{U}^*$, successfully forges signatures at the non-puncturing time nodes is defined as negligible.*

Proof. According to the definition of bidirectional security, it ensures that even if the key is compromised at a specific point in time, known as the "puncturing point", it will not affect the validity of signatures made before that point and will not reveal any key information from signatures made after. In practice, malicious users will choose to attack at a time that falls before or after this puncture point. The unforgeability of signatures is essentially a reflection of both forward and backward security.

- **Setup** The malicious user $\mathcal{U}^*$ declares the challenged puncture point i (where $i \in [\|T\| - 1]$) and receives the secret key $\mathbf{S}_{t_i}$ from the challenger. Subsequently, the user adaptively selects a node j for signature inquiries (where $j \in [\|T\| - 1]$ and $j \neq i$), with a maximum of $\|T\| - 2$ inquiries allowed.
- **Query**
  1. Based on the blind signature interaction rules outlined in Phase 2 of Appendix A.2, the challenger C returns the corresponding signature.
  2. The user $\mathcal{U}^*$ announces the end of the queries.
- **Forgery** At time node t_k, the user $\mathcal{U}^*$ forges a message/signature pair ($\mathbf{e}$, $\mathbf{z}^*$) such that $\text{Verify}(pp, vk, k, \mathbf{e}, \mathbf{z}^*) = 1$. The time node k must not have been previously queried and belong to either T_{pre} or T_{post}.

1. If $t_k \in T_{\text{pre}}$, due to the inherent One-more Unforgeability property of blind signatures, this ensures the forward security of the PBS scheme.
2. If $t_k \in T_{\text{post}}$, it implies that the adversary has a non-negligible probability of solving the `SIS` problem. This ensures the backward security of the PBS scheme.

5.5 Comparisons

The focus of our work lies in contrasting with existing signature schemes, such as forward-secure blind signature (FBS). In terms of efficiency, a comparison can be easily made by referring to Figure 4.

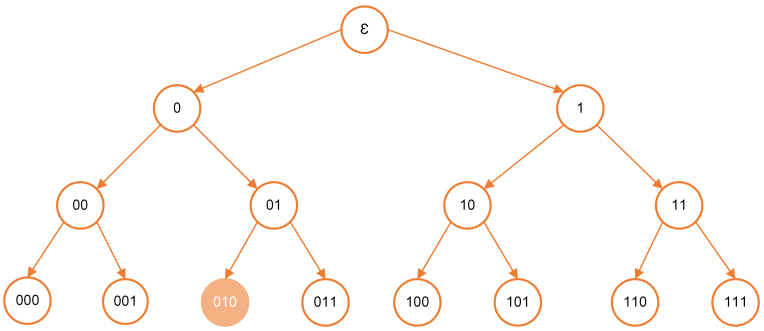

Fig. 4. In the punctured blind signature scheme, when the secret key is leaked at a certain period, we only need to update the leaked secret key independently and in a fine-grained manner

In the FBS scheme, if the adversary acquires the secret key during the time period t_{010}, all subsequent keys must be updated. However, in the proposed punctured blind signature scheme, even if the same scenario occurs, key updates are only necessary at t_{011} due to the precise nature of these updates, thereby significantly reducing the computational complexity from $O(n)$ in the FBS scheme to $O(1)$ in the proposed method.

6 Conclusions

In this paper, we have proposed a puncturable blind signature scheme based on lattices with bidirectional security. Our approach employs puncturable pseudorandom functions for efficient and targeted key revocation, offering computational efficiency and flexible key updates, thus effectively tackling key leak challenges. Future research can build upon this foundation, exploring PBS construction in the standard model, adaptive message signing, and scenarios involving multiple compromised keys.

References

1. Charm, D., Rivets, R. L., Sherman, A. T.,: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R. L., Sherman A. T. (eds) Advances in Cryptology–CRYPTO 1982, pp. 199–203, Springer (1982). https://doi.org/10.1007/978-1-4757-0602-4_18
2. Shim, K. A., An, Y.,: Cryptanalysis of lattice-based blind signature and blind ring signature schemes. IEEE Access, pp. 134427–134434 (2021). https://doi.org/10.1109/ACCESS.2021.3113938
3. Camenisch, J., Neven, G., Shelat, A.,: Simulatable adaptive oblivious transfer. In: Naor M. (ed) Advances in Cryptology–EUROCRYPT 2007, LNCS 4515, pp. 573–590, Springer (2007). https://doi.org/10.1007/978-3-540-72540-4_33
4. Rodriguez-Henriquez, F., Ortiz-Arroyo, D., Garcia-Zamora, C.,: Yet another improvement over the Mu-Varadharajan e-voting protocol. Computer Standards & Interfaces, vol. 29, no. 4, pp. 471–480 (2007). https://doi.org/10.1016/j.csi.2006.11.003
5. Rückert, M.,: Lattice-based blind signatures. In: Abe M. (ed) Advances in Cryptology–ASIACRYPT 2010, LNCS 6477, pp. 413–430, Springer (2010). https://doi.org/10.1007/978-3-642-17373-8_24
6. Lyubashevsky, V.,: Fiat-Shamir with aborts: applications to lattice and factoring-based signatures. In: Matsui M. (ed) Advances in Cryptology–ASIACRYPT 2009, LNCS 5912, pp. 598–616, Springer (2009). https://doi.org/10.1007/978-3-642-10366-7_35
7. Chen, L., Cui, Y., Tang, X., Hu, D., Wan, X.: Hierarchical ID-based blind signature from lattices. In: 2011 Seventh International Conference on Computational Intelligence and Security–CIS 2011, pp. 803–807, IEEE (2011). https://doi.org/10.1109/CIS.2011.182
8. Fuchsbauer, G., Hanser, C., Slamanig, D.: Practical round-optimal blind signatures in the standard model. In: Gennaro R., Robshaw M. (eds) Advances in Cryptology–CRYPTO 2015, LNCS 9216, pp. 233–253, Springer (2015). https://doi.org/10.1007/978-3-662-48000-7_12
9. Zhang, P., Jiang, H., Zheng, Z., Hu, P., Xu, Q.: A new post-quantum blind signature from lattice assumptions. IEEE Access, vol. 6, pp. 27251-27258 (2018). https://doi.org/10.1109/ACCESS.2018.2833103
10. Gao, W., Hu, Y., Wang, B., Xie, J.: Identity-based blind signature from lattices in standard model. In: 12th International Conference on Information Security and Cryptology–Inscrypt 2016, LNCS 10143, pp. 205–218, Springer (2016). https://doi.org/10.1007/978-3-319-54705-3_13
11. Le, H. Q., Duong, D. H., Susilo, W.: A blind ring signature based on the short integer solution problem based on the short integer solution problem. In: You I. (ed) 20th International Conference on International Workshop on Information Security Applications–WISA 2020, LNCS 11897, pp. 92–111, Springer (2020). https://doi.org/10.1007/978-3-030-39303-8_8
12. Alkadri, N. A., El Bansarkhani, R., Buchmann, J.: BLAZE: practical lattice-based blind signatures for privacy-preserving applications. In: Joseph B., Nadia H. (eds) Financial Cryptography and Data Security–FC 2020, LNCS 12059, pp. 484–502, Springer (2020). https://doi.org/10.1007/978-3-030-51280-4_26
13. Beullens, W., Lyubashevsky, V., Nguyen, N. K., Seiler, G.: Lattice-based blind signatures: short, efficient, and round-optimal. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security–CCS 2023, pp. 16–29, ACM (2023). https://doi.org/10.1145/3576915.3616613

14. Lai, Y. P., Chang, C. C.: A simple forward secure blind signature scheme based on master keys and blind signatures. In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications–AINA 2005, vol. 2, pp. 139–144, IEEE (2005). https://doi.org/10.1109/AINA.2005.63
15. Günther, C. G.: An identity-based key-exchange protocol. In: Quisquater, J.J., Vandewalle, J. (eds) Advances in Cryptology–EUROCRYPT 1989, LNCS 434, pp. 29–37, Springer (1990). https://doi.org/10.1007/3-540-46885-4_5
16. Itkis, G., Reyzin, L.: Forward-secure signatures with optimal signing and verifying. In: Kilian, J. (ed) Advances in Cryptology–CRYPTO 2001, LNCS 2139, pp. 332–354, Springer (1999). https://doi.org/10.1007/3-540-44647-8_20
17. Duc, D. N., Cheon, J. H., Kim, K.: A forward-secure blind signature scheme based on the strong RSA assumption. In: Qing, S., Gollmann, D., Zhou, J. (eds) International Conference on Information and Communications Security–ICICS 2003, LNCS 2836, pp. 11–21, Springer (2003). https://doi.org/10.1007/978-3-540-39927-8_2
18. Chow, S. S. M., Hui, L. C. K., Yiu, S. M., Chow, K. P.: Forward-secure multisignature and blind signature schemes. Applied Mathematics and Computation, vol. 168, no. 2, pp. 895-908 (2005). https://doi.org/10.1016/j.amc.2004.09.015
19. Yu, J., Kong, F., Cheng, X., Hao, R., Chen, Y., Li, X.: Forward-secure multisignature, threshold signature and blind signature schemes. Journal of Networks, vol. 5, no. 6, pp. 634-641 (2010). https://doi.org/10.4304/JNW.5.6.634-641
20. Tao, Y., Zhang, R., Ji, Y.: Forward security of Fiat–Shamir lattice signatures. In: Tibouchi, M., Wang, X. (eds) 21st International Conference on Applied Cryptography and Network Security–ACNS 2023, LNCS 13905, pp. 607–633, Springer (2023). https://doi.org/10.1007/978-3-031-33488-7_23
21. Le, H. Q., Duoung, D. H., Susilo, W., Tran, H. T. N., Trinh, V. C., Pieprzyk, J., Plantard, T.: Lattice blind signatures with forward security. In: Liu, J., Cui, H. (eds) 25th Australasian Conference on Information Security and Privacy–ACISP 2020, LNCS 12248, pp. 431–448, Springer (2020). https://doi.org/10.1007/978-3-030-55304-3_1
22. Xiang, T., Li, X., Chen, F., Mu, Y.: Bilateral-secure signature by key evolving. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security–ASIACCS 2016, pp. 523–533, ACM (2016). https://doi.org/10.1145/2897845.2897864
23. Guan, D. J., Lin, D. R., Wang, C.: A forward-secure signature with backward-secure detection. In: Proceedings of the 2008 International Conference on Information Security and Assurance–ISA 2008, pp. 106–110, ACM (2008). https://doi.org/10.1109/ISA.2008.79
24. Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Computing Systems, vol. 48, no. 3, pp. 535-553 (2011). https://doi.org/10.1007/s00224-010-9278-3
25. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T.: Advances in Cryptology–EUROCRYPT 2012, LNCS 7237, pp. 372–381, Springer (2012). https://doi.org/10.1007/978-3-642-29011-4_43
26. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed) Advances in Cryptology–EUROCRYPT 2010, LNCS 6110, pp. 553–572, Springer (2010). https://doi.org/10.1007/978-3-642-13190-5_28
27. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Sako, K., Sarkar, P. (eds) Advances in Cryptology–ASIACRYPT 2013. LNCS 8270, pp. 280–300, Springer (2013). https://doi.org//10.1007/978-3-642-42045-0_15

28. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security–CCS 2013, pp. 669–684, ACM (2013). https://doi.org/10.1145/2508859.2516668
29. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Krawczyk H. (ed) 17th International Conference on Practice and Theory in Public-Key Cryptography–PKC 2014, LNCS 8383, pp. 501–519, Springer (2014). https://doi.org/10.1007/978-3-642-54631-0_29
30. Wang, X., Li, S., Xue, R.: Adaptively secure puncturable pseudorandom functions via puncturable identity-based KEMs. In: Zhou, J. Y., Luo, X. P., Shen, Q. N., Xu, Z. (eds) 21st International Conference on Information and Communications Security–ICICS 2019, LNCS 11999, pp. 501–519, Springer (2019). https://doi.org/10.1007/978-3-030-41579-2_27
31. Lin, D. R., Wang, C. I., Guan, D. J.: A forward-backward secure signature scheme. Journal of Information Science Engineering, vol. 26, no. 6, pp. 2319–2329 (2010). http://www.iis.sinica.edu.tw/page/jise/2010/201011_24.html
32. Micciancio D., Regev O.: Worst-case to average-case reductions based on Gaussian measures. In: 45th Symposium on Foundations of Computer Science–FOCS 2004, pp. 372–381, IEEE (2004). https://doi.org/10.1109/FOCS.2004.72
33. Hauck, E., Kiltz, E., Loss, J., Nguyen, N. K.: Lattice-based blind signatures, Revisited. In: Micciancio, D., Ristenpart, T. (eds) Advances in Cryptology–CRYPTO 2020, LNCS 12171, pp. 500–529, Springer (2020). https://doi.org/10.1007/978-3-030-56880-1_18
34. Schröder, D. and Unruh, D.: Security of blind signatures revisited. Journal of Cryptology, vol. 30, no. 2, pp.470–494 (2017). https://doi.org/10.1007/s00145-015-9225-1
35. Ling, S., Nguyen, K., Wang, H., Xu, Y.: Forward-secure group signatures from lattices. In: 10th International Conference on Post-Quantum Cryptography–PQCrypto 2019, LNCS 11505, pp. 44-64, Springer (2019). https://doi.org/10.1007/978-3-030-25510-7_3
36. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed) Advances in Cryptology–EUROCRYPT 1999, LNCS 1592, pp.223–238, Springer (1999). https://doi.org/10.1007/3-540-48910-X_16
37. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Kaliski, B.S. (ed) Advances in Cryptology–CRYPTO 1997, LNCS 1294, pp. 150–164, Springer (1997). https://doi.org/10.1007/BFb0052233
38. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology, vol. 13, pp. 361–396 (2000). https://doi.org/10.1007/s001450010003
39. Halevi, S., Ishai, Y., Jain, A., Komargodski, I., Sahai, A., Yogev, E.,: In: Takagi, T., Peyrin, T. (eds) Advances in Cryptology–ASIACRYPT 2017, LNCS 10626, pp. 181–211, Springer (2017). https://doi.org/10.1007/978-3-319-70700-6_7
40. Jiang, M., Dung, D. H., Susilo, W.: Puncturable signature: a generic construction and instantiations. In: Atluri, V., Di Pietro, R., Jensen, C.D., Meng, W. (eds) 27th European Symposium on Research in Computer Security–ESORICS 2022, LNCS 13555, pp.507–527, Springer (2022). https://doi.org/10.1007/978-3-031-17146-8_25
41. Li, X., Xu, J., Fan, X., Wang, Y., Zhang, Z.: Puncturable signatures and applications in proof-of-stake blockchain protocols. IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3872–3885 (2020). https://doi.org/10.1109/TIFS.2020.3001738

A Proof for Correctness

Theorem 6. *The punctured blind signature scheme guarantees correctness after a maximum of e^2 restarts, with a probability of at least $1 - 2^{-100}$.*

Proof. Referring to Figure 4, from the user's perspective, we are able to obtain the message/signature pair $(\mathbf{e}', \mathbf{z}')$. It is readily verifiable that:

$$H(\mathbf{F}_t\mathbf{z}' - \mathbf{K}\mathbf{e}' \pmod q, \mathtt{com}(\mu, \mathbf{d})) = \mathbf{e}'.$$

Furthermore, in the absence of restarts during the rejection sampling process, given the relationship $\mu = \mathbf{F}_t\mathbf{a} + \mathbf{x} + \mathbf{K}\mathbf{b} \pmod q$, we have:

$$\mathbf{F}_t\mathbf{a} = \widehat{\mu} - \mathbf{F}_t\mathbf{z} - \mathbf{K}\mathbf{e}'(\text{mod q}) \Rightarrow \mu = \widehat{\mu} - \mathbf{F}_\text{t}\mathbf{z} + \mathbf{K}\mathbf{e}' + \mathbf{x} + \mathbf{K}\mathbf{b}(\text{mod q}),$$

and

$$\begin{aligned}\mu =& \widehat{\mu} - \mathbf{F}_t z + \mathbf{K}\mathbf{e}' + \mathbf{x} + \mathbf{K}\mathbf{b}(mod\ q)\\ =& \widehat{\mu} - \mathbf{F}_t(\mathtt{r} + \mathbf{S}_i\mathtt{e}) + \mathbf{K}\mathbf{e}' + \mathbf{x} + \mathbf{K}\mathbf{b}(mod\ q)\\ =& \widehat{\mu} - \mathbf{F}_t\mathbf{S}_i\mathbf{e} + \mathbf{K}\mathbf{e}' + \mathbf{K}\mathbf{b}(mod\ q)\\ =& \widehat{\mu} - \mathbf{K}\mathbf{e} + \mathbf{K}\mathbf{e}' + \mathbf{K}\mathbf{b}(mod\ q)\\ =& \widehat{\mu}.\end{aligned}$$

Hence $H(\mathbf{F}_t\mathbf{z}' - \mathbf{K}\mathbf{e}'(mod\ q), \mathtt{com}(\mu, \mathbf{d})) = \mathbf{e}'$. Note that, with overwhelming probability, for all $i \in [l]$ we have

$$\frac{D^m_\sigma(\mathbf{e})}{M \cdot D^m_{\sigma,\mathtt{c}}(\mathbf{e})} \leq \frac{e^{1+1/288}}{M},\ \|\mathbf{z}'\| \leq \sigma_3\sqrt{(1+l)m}$$

with probability at least $1 - 2^{-100}$, if $\mathtt{c} = 12\|\sigma\|$. Being used in the rejection sampling, we requires that

$$\frac{D^m_\sigma(\mathbf{e})}{M \cdot D^m_{\sigma,\mathtt{c}}(\mathbf{e})} \leq 1.$$

We can get the best choice $M \approx e^{1+1/288}$. The rejection sampling approach has applied in Phases 3 and 4, to ensure PBS scheme can output a valid signature, after at most $M_2 \cdot M_3 \approx e^2$ restarts.

B Proof for Blindness

Theorem 7. *The proposed PBS scheme is Blindness, if* com *is a statistically hiding commitment and H is a one-way and collision-resistant hash function.*

Proof. As defined in Definition 4 of Blindness, the malicious signer $\mathcal{S}^*$ submits two messages, μ_0 and μ_1, to the challenger. Subsequently, the challenger randomly selects a bit $b \in \{0, 1\}$ and engages in an interaction with $\mathcal{S}^*$. To facilitate the signing of both messages, μ_b and μ_{1-b}, the challenger $\mathcal{C}$ assumes the roles of two users: $\mathcal{U}_b = \mathcal{U}(pp, vk, \mu_b)$ and $\mathcal{U}_{1-b} := \mathcal{U}(pp, vk, \mu_{1-b})$. Throughout

this process, the signer $\mathcal{S}^*$ acquires two message/signature pairs: $(\mathbf{e}_b, \mathbf{z}_b)$ and $(\mathbf{e}_{1-b}, \mathbf{z}_{1-b})$ corresponding to the users $\mathcal{U}_b$ and $\mathcal{U}_{1-b}$, respectively. We contend that the knowledge of $(\mathbf{e}_b, \mathbf{z}_b)$ and $(\mathbf{e}_{1-b}, \mathbf{z}_{1-b})$ bears no correlation to the specific messages μ_b and μ_{1-b}. In essence, this signifies that the signer $\mathcal{S}^*$ remains oblivious to the identity of the user with whom they are interacting.

Indeed, for $\mathcal{A} = (t, \mathbf{r}_b, \mathbf{e}_b, \mathbf{z}_b)$ $(b \in \{0, 1\})$, since $\mathbf{z}_b$ and $\mathbf{z}_{1-b}$ are produced by S^* itself, so we only to analyze the $\mathbf{e}_b$ and $\mathbf{e}_{1-b}$. In Phase 2, $\mathbf{e} = \mathbf{e}' + \mathbf{b}$ and output it with probability min$=(D^m_{\sigma_1}(\mathbf{e})/M_1 D^m_{\sigma_1,\mathbf{e}'}(\mathbf{e}), 1)$, by the rejection sampling, we can make sure the $\mathbf{e}_b$ and $\mathbf{e}_{1-b}$ are independent of the same message being signed.

As for the signature $(\mathbf{d}_b, \mathbf{e}'_b, \mathbf{z}'_b)$, and $(\mathbf{d}_{1-b}, \mathbf{e}'_{1-b}, \mathbf{z}'_{1-b})$, which is similar to $\mathbf{e}_b$ and $\mathbf{e}_{1-b}$, we only to analyze the $\mathbf{z}'_b$ and $\mathbf{z}'_{1-b}$. In Phase 4, uses the rejection sampling, $\mathbf{z}' = \mathbf{z}+\mathbf{a}$ and output it with probability min$=(\mathcal{D}^{(l+1)m}_{\sigma_3}(\mathbf{z}')/M_3 \mathcal{D}^{(l+1)m}_{\sigma_3,\mathbf{z}}, 1)$, by the rejection sampling, we can make sure the $\mathbf{z}_b$ and $\mathbf{z}_{1-b}$ are independent of their corresponding messages being signed.

C Proof for One-more unforgeability

For brevity, we designate the (q_H, q_S, δ)-forger $\mathcal{A}$ as a polynomial-time algorithm $\mathcal{B}$ that violates the one-more unforgeability of our PBS protocol with a significant probability δ, utilizing a maximum of q_H hash queries and q_S sign blind queries. The theorem asserts that the existence of such a forger enables the construction of an SIS problem-solving algorithm.

Theorem 8. (**One-moreUnforgeability**) *Suppose there is an adversary (q_H, q_S, δ)-forger $\mathcal{A}$ which is against one-more unforgeability of our blind signature PBS with non negligible probability δ, q_H and q_S are the number of queries to the random oracle H and blind signing oracle, respectively. Then there exist a polynomial-time algorithm $\mathcal{B}$, that can find a solution to the l_2-SIS$_{q,n,(1+2l)m,\beta}$* problem with $\beta = \max\{(2\sigma_3 + 2\sigma\sqrt{k})\sqrt{(1+l)m}, (2\sigma_3 + \sigma_2)\sqrt{(1+l)m}\}$.

Proof. – **Instance**. Assume that $\mathcal{B}$ wants to solve an instance of the SIS$_{q,n,(1+2l)m,\beta}$ problem

$$\mathbf{F} \cdot \mathbf{V} = 0 \ \log q, \ \|\mathbf{V}\| \leq \beta, \ \mathbf{F} \in \mathbb{Z}^{n \times (1+2l)m} \tag{1}$$

– **Guessing the target**. $\mathcal{B}$ guesses the target time period $\mathbf{t}^*$ that $\mathcal{A}$ wants to attack by choosing randomly $t^* = (t^*_1, \cdots, t^*_l) \xleftarrow{\$} \{0, \cdots, \tau - 1\}$. The success probability of guessing $t^*$is $1/\tau$.

– **Initialize**. $\mathcal{B}$ sets common parameters pp as in the Setup algorithm. $\mathcal{B}$ sets the public key vk to $\mathcal{A}$.

– **Queries**. $\mathcal{B}$ plays the role of signer and interacts with $\mathcal{A}$. $\mathcal{B}$ responds to $\mathcal{A}$ queries as follows:

1. Key update queries $Q_K(t)$ with $t = (t_1, \cdots, t_l)$: If $t \leq t^*$, $\mathcal{B}$ halts the query. Otherwise, it computes an extended basis for a matrix incorporating historical and current key updates, from which it derives all keys in sk_t using the actual key update algorithm.

2. Hash queries $Q_H(\mu, \mathbf{c})$: Upon receiving a hash query, $\mathcal{B}$ checks its hash list $\mathcal{L}_H$. If the query exists, it returns the stored hash value. Otherwise, it selects an unused random value from a predefined set, stores the query-hash pair in $\mathcal{L}_H$, and forwards the hash value to the forger $\mathcal{A}$.
3. Signing queries $Q_S(t, \mu)$: $\mathcal{B}$ constructs a matrix $\mathbf{F}_t$ based on key updates and checks if $t \neq t^*$. If so, it computes an extended basis for $\mathbf{F}_t$ and samples a signing key accordingly. If $t = t^*$, it simply assigns the preset key $\mathbf{S}^*$.
4. Break-out signing queries $Q_B(t)$: When $\mathcal{A}$ makes such a query, if $t \leq t^*$, $\mathcal{B}$ halts. Otherwise, it sets the break-out time $\bar{t}$ to t and responds to $\mathcal{A}$ with the secret key $\mathbf{S}_t$, similar to key update queries.

– **Forge**. Eventually, A outputs a forgery $(t'_1, \mu^*_1, \Sigma^*_1)$. $\mathcal{B}$ checks if $t'_1 = t$ or not. If not, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ accepts the forgery. For the forgery $(t^*, \mu^*_1, \Sigma^*_1)$, we have:
  1. $\Sigma^*_1 = (\mathbf{d}_1, \mathbf{e}'_1, \mathbf{z}'_1)$;
  2. $\mathbf{e}'_1 := H(\mathbf{F}_{t^*} - \mathbf{K}\mathbf{e}'_1 \bmod q,\ \mathrm{com}(\mu^*_1, d'_1))$, where $\mathbf{F}_{t^*} = [\mathbf{A}_0 \| \mathbf{A}_1^{t^*_1} \| \cdots \| \mathbf{A}_l^{t^*_l}] \in \mathbb{Z}^{n \times (1+l)m}$;
  3. $\|\mathbf{z}'_1\| \leq \sigma_3 \sqrt{(1+l)m}$.

  If $\mathbf{e}'_2 = \mathbf{e}'_1$, $\mathcal{B}$ aborts and replays $\mathcal{A}(pp, pk, \rho')$ at most q_H^{qs} times using different random tapes ρ' and different hash queries. If $\mathbf{e}'_2 \neq \mathbf{e}'_1$, then $\mathcal{B}$ returns

$$\mathbf{F}_{t^*}\mathbf{z}'_1 - \mathbf{K}\mathbf{e}'_1,\ \mathtt{com}(\mu^*_1, d'_1),\ \mathbf{F}_{t^*}\mathbf{z}'_2 - \mathbf{K}\mathbf{e}'_2,\ \mathtt{com}(\mu^*_2, \mathbf{d}'_2) \tag{2}$$

  Since the pair in Equation 2 are both coming from the same hash query and com is computationally binding, we have $\mu^*_2 = \mu^*_1, \mathbf{d}'_1 = \mathbf{d}'_2$ and

$$\mathbf{F}_{t^*}\mathbf{z}'_1 - \mathbf{K}\mathbf{e}'_1 = \mathbf{F}_{t^*}\mathbf{z}'_2 - \mathbf{K}\mathbf{e}'_2 (\bmod\ q)$$

  or equivalently,

$$\mathbf{F}_{t^*}(\mathbf{z}'_1 - \mathbf{z}'_2 - \mathbf{S}^*(\mathbf{e}'_1 - \mathbf{e}'_2) = 0 (\bmod\ q) (\bmod\ q)$$

  Set $\widehat{\mathbf{v}} := \mathbf{z}'_1 - \mathbf{z}'_2 - \mathbf{S}^*(\mathbf{e}'_1 - \mathbf{e}'_2)$. Signing interaction (with In particular, we show that if $\mathcal{A}$ can produce a forgery by restarting the signing interaction (with $\mathcal{B}$), then $\mathcal{B}$ is able to find a solution to the l_2-SIS problem given by Equation (1). Indeed, to restart the signing interaction, $\mathcal{A}$ delivers result:=$(\mathbf{a}, \mathbf{b}, \mathbf{e}', \mathbf{c})$ to $\mathcal{B}$. Now $\mathcal{B}$ with its view $\mathcal{V} = (t, \mathbf{r}, \mathbf{e}, \mathbf{z})$, will check whether all

$$\mathbf{e} - \mathbf{b} = \mathbf{e}' = H(x + F_{t^*}\mathbf{a} + K\mathbf{b} (\bmod\ \mathrm{q}), \mathbf{c}) \tag{3}$$

$$\mathbf{e}' = H(\mathbf{F}_{t^*}\mathbf{a} + \mathbf{F}_{t^*}\mathbf{z} - \mathbf{K}\mathbf{e}' (\bmod\ \mathrm{q}), \mathbf{c}) \tag{4}$$

$$\|\mathbf{z} + \mathbf{a}\| > \sigma_3 \sqrt{(1+l)m} \tag{5}$$

hold or not. If all are satisfied, $\mathcal{B}$ restarts the interaction with $\mathcal{A}$. Let assume that afterwards $\mathcal{A}$ successfully produces a valid signature $\widehat{\Sigma} = (\widehat{\mathbf{d}}, \widehat{\mathbf{e}}', \widehat{\mathbf{z}}')$. Let $\widehat{\mathbf{b}} \in D_{\sigma_1}^m$ be such that $\mathbf{e} = \widehat{\mathbf{e}}' + \widehat{\mathbf{b}}$. Then, the following relations have to hold

$$\mathbf{e} - \widehat{\mathbf{b}} = \widehat{\mathbf{e}}' = H(\mathbf{x} + \mathbf{F}_{t^*}\mathbf{a} + K\widehat{\mathbf{b}} (\text{mod q}), \mathbf{c}), \tag{6}$$

$$\widehat{\mathbf{e}}' = H(\mathbf{F}_{t^*}\widehat{\mathbf{z}} - \mathbf{K}\mathbf{e}' (\text{mod q}), \text{com}(\mu^*, \widehat{\mathbf{d}})), \tag{7}$$

$$\|\widehat{\mathbf{z}}\| \leq \sigma_3 \sqrt{(1+l)m}. \tag{8}$$

Now, if $\widehat{\mathbf{e}}' \neq \mathbf{e}'$, then $\mathcal{B}$ aborts. Otherwise, Equations 3 and 6 give $\mathbf{F}_{t^*}\mathbf{a} + \mathbf{F}_{t^*}\mathbf{z} (\text{mod q}) = \mathbf{F}_{t^*}\widehat{\mathbf{z}} (\text{mod q})$. We insert zeros into the corresponding position of $\widehat{\mathbf{v}}$ to get the desired solution $\mathbf{v}$ to the problem given by Equation 1. Obviously, $\mathbf{F} \cdot \mathbf{v} = 0 (\text{mod q})$, and $\|\mathbf{v}\| = \|\widehat{\mathbf{v}}\|$. Let $\widehat{\mathbf{v}} := \mathbf{a} + \mathbf{z} - \widehat{\mathbf{z}}'$, then $\widehat{\mathbf{v}} \neq 0$. This is true as otherwise $\mathbf{a} + \mathbf{z} = \widehat{\mathbf{z}}'$, which implies that $\|\mathbf{z} + \mathbf{a}\| \leq \eta\sigma_3\sqrt{m}$ (by Equation 7.)

This contradicts Equation 4. Again, we have $\mathbf{F}_{t^*} \cdot \widehat{\mathbf{v}}' = 0 (\text{mod q})$, $\widehat{\mathbf{v}} \neq 0$ and $\|\widehat{\mathbf{v}}\| \leq \|\mathbf{a}\| + \|\mathbf{z}\| + \|\widehat{\mathbf{z}}\| \leq (2\sigma_3 + \sigma_2)\sqrt{(1+l)m}$.

To summarise, we have shown that $\mathcal{B}$ can solve the l_2-$\text{SIS}_{q,n,(1+2l)m,\beta}$ problem, with

$$\beta = \max\{(2\sigma_3 + 2\sigma\sqrt{k})\sqrt{(1+l)m},\ (2\sigma_3 + \sigma_2)\sqrt{(1+l)m}\}.$$