Semigroup-homomorphic Signature

Heng Guo[1,2], Kun Tian[1,3*], Fengxia Liu[3], Zhiyong Zheng[1,3,4]

[1]School of Mathematics, Renmin University of China, Beijing, China.
[2]Institute of Interdisciplinary Science, Renmin University of China, Beijing, China.
[3]Great Bay University, Dongguan, Guangdong Province, China.
[4]Institute of Mathematics, Henan Academy of Sciences, Zhengzhou, Henan Province, China.

*Corresponding author(s). E-mail(s): tkun19891208@ruc.edu.cn;
Contributing authors: guoheng@ruc.edu.cn; shunliliu@gbu.edu.cn; zhengzy@ruc.edu.cn;

Abstract

In 2002, Johnson et al. posed an open problem at the Cryptographers' Track of the RSA Conference: how to construct a secure homomorphic signature on a semigroup, rather than on a group. In this paper, we introduce, for the first time, a semigroup-homomorphic signature scheme. Under certain conditions, we prove that the security of this scheme is based on the hardness of the Short Integer Solution (SIS) problem and is tightly secure. Furthermore, we extend it to a linearly semigroup-homomorphic signature scheme over lattices, and this scheme can also ensure privacy.

Keywords: Semigroup, homomorphic signature, lattice, tightly secure

1 Introduction

The primary purpose of homomorphic signature schemes was to provide authentication services for network coding and to effectively mitigate pollution attacks[1]. However, the significance of these schemes extends well beyond this initial goal. Due to their capability to perform computations on authenticated data, they have proven to be of unique importance across a variety of application scenarios. The idea of homomorphic signature schemes was initially proposed by R. Rivest [2] during a lecture at

Cambridge University in 2000. In 2002, Johnson et al. [3] formalized the definition and conducted a security analysis of these schemes. Following this, a range of different homomorphic signature schemes has been developed. When categorizing these schemes according to the type of homomorphic operation they support, they are typically divided into linearly homomorphic signatures, homomorphic signature scheme for polynomial functions, and fully homomorphic signatures.

The first homomorphic signature scheme to be proposed was the linearly homomorphic signature scheme. In 2007, Zhao et al. put forward a linearly homomorphic signature scheme [1]. This scheme enables arbitrary linearly combinations of signed data, which is highly beneficial for verifying the integrity of received messages. Moreover, it plays a crucial role in effectively safeguarding applications based on network coding against pollution attacks. Following this pioneering work, a plethora of efficient and practical linearly homomorphic signature schemes [4–10] have been introduced. Researchers have since dedicated their efforts to enhancing these schemes, with a particular focus on improving efficiency, bolstering security, and strengthening privacy protection.

In addition to linearly homomorphic signatures, more flexible homomorphic signature schemes for polynomial functions and fully homomorphic signatures have also been developed. In 2011, Boneh et al. proposed the first homomorphic signature scheme for polynomial functions, which allows multivariate polynomial computations on signed data [11]. There have not been many improvements to homomorphic signature schemes for polynomial functions. Currently, only Hiromasa et al. [12], Arita et al. [13] and Catalano et al. [14] have made meaningful contributions to the efficiency and security of these schemes. In 2009, Gentry et al. proposed the first fully homomorphic encryption scheme [15], which attracted widespread attention from cryptographers and subsequently led many researchers to investigate fully homomorphic signatures. In 2014, Gorbunov et al. proposed the first leveled fully homomorphic signature scheme [16]. Subsequently, the research works in [17–21] have made continuous improvements in efficiency, functionality, and security.

According to the definition of homomorphic signature in reference [3], the message space $\mathcal{M}$ is not required to be a group, nor does the signature algorithm have to be a group homomorphism. Furthermore, in certain group homomorphic algorithms, when the signature of the identity element is known or the signature of the inverse of a specific message can be readily computed, homomorphic operations can be exploited to generate the signature of any message. This situation would render the signature system ineffective. If the message space is a semigroup instead of a group, the signature is then called a semigroup-homomorphic signature, as described in reference [3]. Consequently, in addressing the above-mentioned issue, a semigroup-homomorphic signature is comparatively secure.

In light of this, Johnson et al.[3] posed the following open question at the Cryptographers' Track of the RSA Conference: Is it possible to find a concrete example of a secure semigroup-homomorphic signature rather than a group-homomorphic signature? To be more specific, let $\mathcal{M} = \{0,1\}^*$ be the message space, and define the operation on $\mathcal{M}$ as $\|$, which represents the concatenation of bit strings. For any $\mathbf{x}, \mathbf{y} \in \{0,1\}^*$, $\mathbf{x} \parallel \mathbf{y} \in \mathcal{M}$. It is clear that $\mathcal{M}$ forms a semigroup under $\|$. Next, let the signature space be Σ, with an operation defined on Σ as $\otimes$, and let the signature algorithm be **Sign** : $\mathcal{M} \to \Sigma$. The question is whether it is possible to construct a semigroup-homomorphic algorithm **Sign**, such that the following holds:

$$\mathbf{Sign}(\mathbf{x} \parallel \mathbf{y}) = \mathbf{Sign}(\mathbf{x}) \otimes \mathbf{Sign}(\mathbf{y}), \quad \text{for any } \mathbf{x}, \mathbf{y} \in \{0,1\}^*.$$

However, if we take such a semigroup $\mathcal{M} = \{0,1\}^*$ as the message space, there will be a fatal flaw: once the signatures of 0 and 1 are given, then through the homomorphic algorithm, the signatures of any message $x \in \{0,1\}^*$ will be obtained, which makes the signature system meaningless. Therefore, it is necessary to consider new semigroups.

Our Contribution. In this paper, we propose the first semigroup-homomorphic signature scheme. Under certain conditions, we show that the security of this scheme is based on the hardness of the Short Integer Solution (SIS) problem, and we prove that it is tightly secure. Furthermore, we extend the scheme to a linearly semigroup-homomorphic signature scheme based on lattices, which also ensures privacy

Overview of our techniques.

- **Determining the Message Space and Signature Space.**

  We know that homomorphic mappings preserve algebraic structures. Homomorphic signature algorithms can be regarded as homomorphic mappings between two algebraic structures. Therefore, if the message space forms a semigroup under a certain operation, the signature space will also form at least a semigroup under a corresponding operation.

  As mentioned earlier, we cannot choose the semigroup $(\{0,1\}^*, \|)$; instead, we need to select a different semigroup. If we write $\{0,1\}^*$ as $(\mathbb{Z}_2)^*$, then in general, $(\mathbb{Z}_p)^*$ denotes the set of elements where each position is taken from $\mathbb{Z}_p$. Let $\mathbb{Z}_+ = \{0, 1, 2, \dots\}$. Then $(\mathbb{Z}_+)^*$ denotes the set of elements where each position is taken from $\mathbb{Z}_+$.

  It is evident that $((\mathbb{Z}_p)^*, \|)$ and $((\mathbb{Z}_+)^*, \|)$ are both semigroups. When p is sufficiently large, if the homomorphic signature uses these two semigroups as the message space, it avoids the situation where the signature of all messages can be derived from the signatures of a few messages through homomorphic operations.

  To avoid additional requirements (such as p being sufficiently large), in our construction, we choose $((\mathbb{Z}_+)^*, \|)$ as the message space for our homomorphic signature scheme. Finally, we choose $((\mathbb{Z}^n)^*, \|)$ as the signature space of the homomorphic signature scheme. Similarly, here $(\mathbb{Z}^n)^*$ denotes the set where each position of the elements comes from $\mathbb{Z}^n$. In fact, this is a natural choice.

- **Signature generation for the semigroup-homomorphic signature.**

  We primarily leverage the trapdoor generation algorithm **TrapGen**(q, h, n) of Peikert et al. [22] for key pair generation and the pre-image sampling algorithm **SamplePre**$(A, T_A,\ \mathbf{u},\ \sigma)$ of Gentry et al.[23] for signature generation. Let $\mathbf{x} \in ((\mathbb{Z}_+)^*, \|)$, where $\mathbf{x} = (x_1, ..., x_k)$ and $x_i \in \mathbb{Z}_+$ for $1 \leq i \leq k$. To sign the message $\mathbf{x}$, we perform a sub-signature operation on each component x_i of the message $\mathbf{x}$

to obtain $\boldsymbol{\sigma}_i$, and then apply a homomorphic operation to $\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_k$. The resulting signature for the message $\mathbf{x}$ is $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 \parallel \ldots \parallel \boldsymbol{\sigma}_k$.

Next, we describe how to generate sub-signatures. First, we select a collision-resistant hash function $\mathbf{h} : \mathbb{Z}_+ = \{0,1\}^* \to \{0,1\}^k$. Then, we apply the hash function $\mathbf{h}$ to the component x_i of the message $\mathbf{x}$, obtaining a k-dimensional bit-string $\mathbf{v}_i$. Drawing on the idea from [24], we randomly and independently choose k linearly independent vectors $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 ..., \boldsymbol{\alpha}_k$ in $\mathbb{Z}_q^h$(which are part of the public key). Then, we set $\boldsymbol{\beta}_i = \sum_j^k \mathbf{v}_{ij} \boldsymbol{\alpha}_j$. Finally, by using the sampling algorithm **SamplePre**(A,T_A, $\boldsymbol{\beta}_i$, σ), we can obtain a sub-signature $\boldsymbol{\sigma}_i$.

- **Linear operations on semigroup-homomorphic signatures.**

  Based on the definition and security model of the linearly homomorphic signature scheme provided in [11], we can similarly define the linearly semigroup-homomorphic signature scheme and its security model (see Section 2). In our linearly semigroup-homomorphic signature scheme, to ensure the completeness of linearly operations, we define the signature of the empty string $\varnothing$ as $\aleph$, which satisfies that for any $\mathbf{x} \in (\mathbb{Z}_+)^*$, $\mathbf{x} \parallel \varnothing = \varnothing \parallel \mathbf{x} = \mathbf{x}$ and for any $\boldsymbol{\sigma} \in (\mathbb{Z}^n)^*$, $\boldsymbol{\sigma} \parallel \aleph = \aleph \parallel \boldsymbol{\sigma} = \boldsymbol{\sigma}$. Next, we define $0 \cdot \mathbf{x} = \varnothing$ and $0 \cdot \boldsymbol{\sigma} = \aleph$ for any $\mathbf{x} \in (\mathbb{Z}_+)^*$, $\boldsymbol{\sigma} \in (\mathbb{Z}^n)^*$. If $c \in \mathbb{Z}_p = \{0, 1, 2, ..., p-1\}$ and $c \neq 0$, $\mathbf{x} \in (\{0,1\}^*, \parallel)$, we define

  $$c \cdot \mathbf{x} = \underbrace{\mathbf{x} \parallel \cdots \parallel \mathbf{x}}_{c}.$$

  Therefore, if $\langle f \rangle = (c_1, \ldots, c_k)$, where $c_i \in \mathbb{Z}_p$, then for any $\mathbf{x}_1, \ldots, \mathbf{x}_k \in (\mathbb{Z}_+)^*$, we have

  $$f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = c_1 \cdot \mathbf{x}_1 \parallel \cdots \parallel c_k \cdot \mathbf{x}_k = \underbrace{\mathbf{x}_1 \parallel \cdots \parallel \mathbf{x}_1}_{c_1} \parallel \cdots \parallel \underbrace{\mathbf{x}_k \parallel \cdots \parallel \mathbf{x}_k}_{c_k}.$$

  Similarly, we can define the linear operations on the signature space $(\mathbb{Z}^n)^*$. For the sake of readability, we will not repeat it here.

- **Signature generation for the linearly semigroup-homomorphic signature.**

  In essence, the only difference between the linearly semigroup-homomorphic signature we constructed and the semigroup-homomorphic signature constructed above is that in the linearly semigroup-homomorphic signature, each data set is bound to a unique label vector $\boldsymbol{\tau}$. Therefore, the label vector needs to be properly handled during the signature process. Drawing on the idea from [24], we map the label vector $\boldsymbol{\tau}$ to a diagonal matrix H whose diagonal elements are ±1. Then, by means of the algorithm **NewBasis**(A, T_A, H)[24], we generate a new basis T_B, where T_B is the basis of the lattice $\Lambda_q^{\perp}(B)$ and $B = AH^{\top}$. The advantage of this algorithm is that the dimensions of the newly generated lattice and basis remain unchanged. Finally, the way of generating the signature is consistent with the signature generation process of the above semigroup-homomorphic signature.

Paper Organization. In Section 2, we introduce the definition of semigroup-homomorphic signature and their security model, followed by the definition and security model of linearly semigroup-homomorphic signature. In Section 3, we present some commonly used notations and preliminary knowledge. In Section 4, we propose our semigroup-homomorphic signature and prove its correctness and security. In Section 5, we introduce linearly semigroup-homomorphic signature and prove their correctness, unforgeability, and privacy. The final section concludes the paper and discusses some open problems.

2 Semigroup-Homomorphic Signature: Definition and Security model

2.1 Semigroup-Homomorphic Signature

A signature scheme is defined by a message space $\mathcal{M}$, a set of private keys $\mathcal{K}$, a set of public keys $\mathcal{K}'$, a signature generation algorithm **Sign** : $\mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ (which may be randomized), and a verification algorithm **Verify** : $\mathcal{K}' \times \mathcal{M} \times \mathcal{Y} \to \{0, 1\}$. The scheme must satisfy the condition that for all messages $x \in \mathcal{M}$, when (k, k') are matching private and public keys, it holds that $\textbf{Verify}(k', x, \textbf{Sign}(k, x)) = 1$. We often omit the private and public keys, using $\textbf{Sign}(x)$ in place of $\textbf{Sign}(k, x)$ and $\textbf{Verify}(x, s)$ instead of $\textbf{Verify}(k', x, s)$, provided this causes no ambiguity. Additionally, for a binary operation $\odot : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ and a subset $S \subseteq \mathcal{M}$, we define $\text{span}_{\odot}(S)$ as the smallest set T such that $S \subseteq T$ and $x \odot y \in T$ for all $x, y \in T$.

In 2002, Johnson et al.[3] first formally defined homomorphic signatures and the associated security model. Below, based on [3], we provide the original definition of homomorphic signature.

Definition 1. *(homomorphic signature,[3]) Let* ***Sign*** $: \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ *and* ***Verify*** $: \mathcal{K}' \times \mathcal{M} \times \mathcal{Y} \to 1$ *be a signature scheme, and let* $\odot : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ *be a binary operation. We say that* ***Sign*** *is homomorphic with respect to* $\odot$ *if there exists an efficient family of binary operations* $\otimes_{k'} : \mathcal{Y} \times \mathcal{Y} \to \mathcal{Y}$ *such that for all* x, x', y, y' *satisfying* $\textbf{\textit{Verify}}(x, y) = \textbf{\textit{Verify}}(x', y') = 1$*, the following condition holds:*

$$y \otimes_{k'} y' = \textbf{\textit{Sign}}(k, x \odot x').$$

A homomorphic signature scheme is called a semigroup-homomorphic signature scheme when the message space forms a semigroup, not a group. Below, we provide the definition of a semigroup-homomorphic signature.

Definition 2. *In the description of the homomorphic signature mentioned above, if the message space* $(\mathcal{M}, \odot)$ *is a semigroup rather than a group, then we refer to this homomorphic signature as a semigroup-homomorphic signature.*

For the homomorphic signature scheme we construct, we introduce a definition of existential unforgeability under the adaptive chosen-message attack with a fixed message range (EUF-CMA-FMR), which can be viewed as a selective version of the existential unforgeability definition introduced in [3].

We introduce a weaker security concept named EUF-CMA-FMR because there are some fundamental obstacles in achieving the standard concept in [3]. The obstacles are as follows: According to the definition of our signature algorithm, in order to avoid trivial forgeries, we require that only messages $\mathbf{x} \in \mathbb{Z}_+$ can be signed with the private key. Therefore, there is a certain range limitation on the messages for signature queries. This motivates the introduction of our security concept, EUF-CMA-FMR, in this paper.

Definition 3. *(EUF-CMA-FMR) We say that a homomorphic signature scheme $\mathcal{S}$ is (t, q, ϵ)-secure under the adaptive chosen-message attacks within a fixed message range , if each adversary, after being given the public key, makes at most q adaptive chosen-message queries to a non-empty subset $\mathcal{M}'$ of the message space $\mathcal{M}$, and runs in time at most t, with its advantage satisfying $\mathrm{Adv}(\mathcal{A}) \leq \epsilon$. The advantage of an adversary $\mathcal{A}$ is defined as the probability that, after queries on the messages $x_1, ..., x_q \in \mathcal{M}'$, $\mathcal{A}$ outputs a valid signature (x', y') on some message $x' \notin \mathrm{span}_{\odot}(x_1, ..., x_q)$. In other words,*

$$\mathrm{Adv}(\mathcal{A}) = \Pr[\mathcal{A}^{\boldsymbol{Sign}(k,\cdot)} = (x', y') \wedge \boldsymbol{Verify}(x', y') = 1 \wedge x' \notin \mathrm{span}_{\odot}(x_1, ..., x_q)].$$

Remark 1: To ensure that the subset $\mathcal{M}'$ has enough messages to provide for the adversary's queries, it is generally required that the number of elements in $\mathcal{M}'$ is much greater than the maximum number of queries, i.e., $|\mathcal{M}'| \gg q$

2.2 Linearly Semigroup-Homomorphic Signature

Inspired by the definition of linearly homomorphic signature schemes in [5, 11], we present the following definition of a linearly semigroup-homomorphic signature scheme.

Definition 4. *(Linearly semigroup-homomorphic signature) A linearly semigroup-homomorphic signature scheme is a tuple of probabilistic, polynomial-time algorithms $\mathcal{LHS}$ =**(Setup, Sign, Combine, Verify)** with the following functionality:*

- ***Setup**(n, pp). Given a security parameter n and additional public parameters pp , the public parameter pp defines a maximum data-set size k, a message space $(\mathcal{M}, \oplus)$, and a signature space $(\Sigma, \otimes)$, where the message space is a semigroup, not a group. The algorithm then outputs a public key pk and a secret key sk.*

- ***Sign**$(sk, \tau, \mathbf{v})$. Given a secret key sk, data-set tag $\tau \in \{0,1\}^n$, and a vector $\mathbf{v} \in \mathcal{M}$, this algorithm outputs a signature $\boldsymbol{\sigma}$.*

- ***Combine**$(pk, \tau, \{(c_i, \boldsymbol{\sigma}_i)_{i=1}^l\})$. Given a public key pk, a data-set tag $\tau \in \{0,1\}^n$, and a set of tuples $(c_i, \boldsymbol{\sigma}_i)_{i=1}^l$ where $c_i \in R$ and $\boldsymbol{\sigma}_i \leftarrow$ **Sign**$(sk, \tau, \mathbf{v}_i)$, with R being an* *integer ring, this algorithm outputs a signature $\boldsymbol{\sigma}$, which is meant to be a signature on $\sum_{i=1}^{l} c_i \mathbf{v}_i$.*

- ***Verify***$(pk, \tau, \mathbf{y}, \boldsymbol{\sigma})$. *On input a public key pk, an data-set tag $\tau \in \{0,1\}^n$, a vector $\mathbf{y} \in \mathcal{M}$, and a signature $\boldsymbol{\sigma}$, this algorithm outputs either 0 (reject) or 1 (accept).*

*We require that for each (pk, sk) output by **Setup**(n, pp), we have:*

*(1) For all τ and $\mathbf{y} \in \mathcal{M}$, if $\boldsymbol{\sigma} \leftarrow$ **Sign**$(sk, \tau, \mathbf{y})$ then **Verify**$(pk, \tau, \mathbf{y}, \boldsymbol{\sigma}) = 1$.*

*(2) For all $\tau \in \{0.1\}^n$ and all sets of triples $\{(c_i, \boldsymbol{\sigma}_i, \mathbf{v}_i)\}_{i=1}^{l}$, if **Verify**$(pk, \tau,$ $mathbfv_i, \boldsymbol{\sigma}_i) = 1$, for all i, then*

$$\textbf{\textit{Verify}}(pk, \tau, \sum_{i=1}^{l} c_i \mathbf{v}_i, \textbf{\textit{Combine}}(pk, \tau, \{(c_i, \boldsymbol{\sigma}_i)\}_{i=1}^{l})) = 1.$$

Since our linearly semigroup-homomorphic signature scheme is a direct extension of the semigroup homomorphism signature scheme constructed in Section 4, we will similarly present the definition of unforgeability under adaptive chosen - message attacks within a fixed message range for the linearly semigroup - homomorphic signature, abbreviated as Unforgeability-FMR.

Definition 5. *(Unforgeability-FMR) A linearly semigroup-homomorphic signature scheme $\mathcal{LHS}$ = (**Setup**, **Sign**, **Combine**, **Verify**) is unforgeable if the advantage of any probabilistic, polynomial-time adversary $\mathcal{A}$ in the following security game is negligible in the security parameter n:*

Setup: *The challenger runs **Setup**(n, pp) to obtain (pk, sk), and gives pk to $\mathcal{A}$.*
***Queries**: The challenger selects a sufficiently large subset $\mathcal{M}' \subseteq \mathcal{M}$, where $|\mathcal{M}'| \gg qk$. Proceeding adaptively, $\mathcal{A}$ specifies a sequence of subsets*

$$V_i \subseteq \mathcal{M}^k, \quad V_i = \mathrm{span}_{\oplus}\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \ldots, \mathbf{v}_{ik}\}, \quad \mathbf{v}_{ij} \in \mathcal{M}',$$

where $1 \leq i \leq q_s$, and $1 \leq j \leq k$. For each i, the challenger chooses a tag τ_i uniformly from $\{0,1\}^n$ and gives to $\mathcal{A}$ the pair $(\tau_i, \{\boldsymbol{\sigma}_{ij}\})$, where the signatures are computed as:

$$\boldsymbol{\sigma}_{ij} \leftarrow \textbf{\textit{Sign}}(sk, \tau_i, \mathbf{v}_{ij}), \quad j = 1, \ldots, k.$$

***Output**: $\mathcal{A}$ outputs $\tau^* \in \{0,1\}^n$, $\mathbf{y}^* \in \mathcal{M}$, and a signature $\boldsymbol{\sigma}^*$*

*The adversary wins if **Verify**$(sk, \tau^*, \mathbf{y}^*, \boldsymbol{\sigma}^*) = 1$, and either*

*(1) $\tau^* \neq \tau_i$ for all i (**type I forgery**), or*
*(2) $\tau^* = \tau_i$ for some i but $\mathbf{y}^* \notin V_i$ (**type II forgery**).*

The advantage of $\mathcal{A}$ is defined to be the probability that $\mathcal{A}$ wins the security game.

The following is a review of the definition of weakly context hiding [11]

Definition 6. *(Weakly context hiding) A linearly semigroup-homomorphic signature scheme $\mathcal{LHS}$ =(**Setup**, **Sign**, **Combine**, **Verify**) is weakly context hiding if the advantage of any probabilistic, polynomial-time adversary $\mathcal{A}$ in the following security game is negligible in the security parameter n:*

***Setup**: The challenger runs **Setup**(n, pp) to obtain (pk, sk) and gives pk and sk to $\mathcal{A}$.*

***Challenge**: A outputs $(V_0, V_1, f_1, ..., f_s)$, where $V_b = span_{\oplus}\{\mathbf{v}_{b1}, ..., \mathbf{v}_{bk}\}$, $\{\mathbf{v}_{b1}, ..., \mathbf{v}_{bk}\} \subseteq \mathcal{M}$, $b \in \{0,1\}$. The functions $f_1, ..., f_s$ are R-linear functions from $\mathcal{M}^k$ to $\mathcal{M}$, and they satisfy*

$$f_i(\mathbf{v}_{01}, ..., \mathbf{v}_{0k}) = f_i(\mathbf{v}_{11}, ..., \mathbf{v}_{1k}), 1 \leq i \leq s.$$

*In response, the challenger generates a random bit $b \in \{0,1\}$ and a random tag $\tau \in \{0,1\}^n$ and signs the vector subsets V_b using the tag τ. Next, for $i = 1, ..., s$, the challenger uses the algorithm **Combine** to derive signatures $\boldsymbol{\sigma}_i$ on $f_i(\mathbf{v}_{b1}, ..., \mathbf{v}_{bk})$ and sends $\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_s$ to $\mathcal{A}$*

***Output**: $\mathcal{A}$ outputs a bit b'.*

The adversary $\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ is the probability that $\mathcal{A}$ wins the game.

2.3 Tight security

At the end of this section, we provide a brief introduction to the important concept of tight security in cryptography.

In the framework of provable security, reduction is a key concept for evaluating the security of signature schemes. The definition is as follows.

Definition 7. *If an adversary $\mathcal{A}$ breaks the scheme $\mathcal{S}$ with (t, ϵ) in the defined security model, then there exists an algorithm $\mathcal{B}$ that breaks a certain computational problem P with (t', ϵ'), where $\epsilon' = \epsilon/\theta$ and $t' = t + o(t)$, and $\theta \geq 1$.The parameter θ is used to measure the tightness of the reduction. In this context, the security parameter is denoted as λ, and the number of adversarial queries is Q. When $\theta = \mathcal{O}(1)$, it is called a tight reduction; when $\theta = poly(\lambda)$, it is an almost tight reduction; when $\theta = poly(Q)$, it is a loose reduction.*

The use of tight reductions is of great significance. From a practical perspective, a tighter reduction allows for shorter security parameters, thereby improving efficiency. Theoretically, a tight reduction indicates that the difficulty of the two computational problems is close.

3 Preliminaries

Notation. In our discussion, we employ the notations $\mathcal{O}$, $\widetilde{\mathcal{O}}$, and ω to characterize the growth rates of functions. Consider two functions f and g that depend on the variable n. We say that the relationship $f(n) = \mathcal{O}(g(n))$ is valid precisely when there exist a positive constant c and an integer N such that for every integer n greater than N, the inequality $f(n) \leq c \cdot g(n)$ is satisfied.

Similarly, we define the notation $f(n) = \widetilde{\mathcal{O}}(g(n))$ to represent a scenario where, for a specific positive constant c', the function $f(n)$ satisfies the relation $f(n) = \mathcal{O}(g(n) \cdot \log^{c'} n)$. As for the notation $f(n) = \omega(g(n))$, it is established if and only if there is an integer N such that for every positive constant c and all integers n that are greater than N, the inequality $g(n) \leq c \cdot f(n)$ holds.

Let the security parameter be denoted as n. When there exists a positive constant c such that $f(n) = \mathcal{O}(n^c)$, we write $f(n) = \text{poly}(n)$. On the contrary, if for every positive constant c, the function $f(n)$ has the property that $f(n) = \mathcal{O}(n^{-c})$, then $f(n)$ is referred to as negligible and is symbolized as $\text{negl}(n)$. In the case where the probability of an event taking place is $1 - \text{negl}(n)$, we state that the event occurs with overwhelming probability.

We use the capital letter $\mathbb{Z}$ to denote the ring of all integers. Meanwhile, $\mathbb{Z}_q$ stands for the ring of integers modulo q (where $q \geq 2$), and $\mathbb{R}^n$ represents the n-dimensional Euclidean space. In our notation, we employ capital letters like A, B, C, etc., to represent matrices, and bold lowercase letters such as $\mathbf{a}, \mathbf{b}, \mathbf{c}$, etc., to denote vectors.

Suppose $A = (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in \mathbb{R}^{h \times n}$. The norm of matrix A is defined in the following way: $\|A\| = \max_{1 \leq i \leq n} \|\mathbf{a}_i\|$, where $\|\mathbf{a}_i\|$ represents the l_2-norm of the vector $\mathbf{a}_i$.

Moreover, we denote $\widetilde{A} = (\widetilde{\mathbf{a}}_1, \ldots, \widetilde{\mathbf{a}}_n)$ as the result of applying the Gram - Schmidt orthogonalization process to matrix A. The Gram-Schmidt orthogonalization is defined as follows:

$$\widetilde{\mathbf{a}}_1 = \mathbf{a}_1, \quad \widetilde{\mathbf{a}}_i = \mathbf{a}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{a}_i, \widetilde{\mathbf{a}}_j \rangle}{\langle \widetilde{\mathbf{a}}_j, \widetilde{\mathbf{a}}_j \rangle} \widetilde{\mathbf{a}}_j, \quad 2 \leq i \leq n,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product within the Euclidean space.

For an arbitrary distribution $\mathcal{D}$, the notation $x \sim \mathcal{D}$ indicates that the variable x is distributed according to $\mathcal{D}$, and $x \leftarrow \mathcal{D}$ represents the operation of sampling a random value in accordance with this distribution. Given a set $\mathcal{X}$, the expression $x \overset{\$}{\leftarrow} \mathcal{X}$ is used to signify the act of uniformly and randomly selecting an element x from $\mathcal{X}$. For any probabilistic polynomial-time (PPT) algorithm Alg, we use $y \leftarrow Alg(x)$ to convey that the algorithm takes x as input and subsequently yields the output y.

Definition 8. *(*Lattice*[24]) Let $\Lambda \subset \mathbb{R}^n$ be a non-empty subset. Λ is called a lattice if:*

(1) it is an additive subgroup of $\mathbb{R}^n$;

(2) there exists a positive constant $\lambda = \lambda(\Lambda) > 0$ such that

$$\min\{\| \mathbf{x} \| \, | \mathbf{x} \in \Lambda, \mathbf{x} \neq 0\} = \lambda.$$

λ is called the minimum distance.

A full-rank n-dimensional lattice can also be expressed as a linear combination of a set of basis vectors $B = \{\mathbf{b}_1, ..., \mathbf{b}_n\} \subset \mathbb{R}^n$:

$$\Lambda = \mathcal{L}(B) = \{B \cdot \mathbf{x} = \sum_{i=1}^{n} x_i \mathbf{b}_i | x = (x_1, ..., x_n)^\top \in \mathbb{Z}^n\}.$$

We call Λ^* the dual lattice of Λ if $\Lambda^* = \{\mathbf{y} \in \mathbb{R}^n | < \mathbf{y}, \mathbf{x} > \in \mathbb{Z} \quad \text{for all } \mathbf{x} \in \Lambda\}$.

Definition 9. *(*q-ary lattices*, [26]) Let $A \in \mathbb{Z}_q^{n\times m}$, $\mathbf{u} \in \mathbb{Z}^n$. The following two q-ary lattices are defined as:*
(1)$\Lambda_q^\perp = \{\mathbf{x} \in \mathbb{Z}^m | A \cdot \mathbf{x} \equiv 0(\text{mod}q)\}$
(2)$\Lambda_q^\mathbf{u} = \{\mathbf{y} \in \mathbb{Z}^m | A \cdot \mathbf{y} \equiv \mathbf{u}(\text{mod}q)\}$
The set $\Lambda_q^\mathbf{u}$ is a coset of $\Lambda_q^\perp$ since $\Lambda_q^\mathbf{u} = \Lambda_q^\perp + \mathbf{t}$ for any $\mathbf{t}$ such that $A \cdot \mathbf{t} \equiv \mathbf{u}(\text{mod}q)$.

The short integer solution (SIS) problem is formally described as follows.

Definition 10. *(*Short integer solution *[27]) Let n, m, q be positive integers, with $m = \text{poly}(n)$. Let $A \in \mathbb{Z}_q^{n\times m}$ be a uniformly distributed random matrix over $\mathbb{Z}_q$, and let $\beta \in \mathbb{R}$ such that $0 < \beta < q$. The SIS problem is to find a short integer solution $\mathbf{x}$ satisfying the following condition:*

$$A \cdot \mathbf{x} \equiv 0(\text{mod}q), \quad \text{and} \quad \mathbf{x} \neq 0, \| \mathbf{x} \| \leq \beta.$$

We write the above SIS problem as $\text{SIS}_{q,n,m,\beta}$ or $\text{SIS}_{q,\beta}$.

Ajtai, in [28], established that solving the SIS problem is at least as hard as solving a related worst-case problem. Later, Gentry et al., in [23], presented an improved reduction, detailed as follows.

Theorem 1. *(*Worst-case to average-case reduction *[23]) For any polynomial bounded $m = poly(n)$, and any $\beta > 0$, if $q \geq \beta \cdot \omega(\sqrt{n \log n})$, then solving the average-case problem $\boldsymbol{SIS}_{q,\beta}$ is at least as hard as solving the worst-case problem $SIVP_\gamma$ on any n-dimensional lattice for $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$.*

Definition 11. *(* Gaussian distributions*[27]) Let s be a positive real number and $\mathbf{c} \in \mathbb{R}^n$ be a vector. The Gaussian function centered at $\mathbf{c}$ with parameter s is defined as: $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{\frac{-\pi}{s^2}\|\mathbf{x}-\mathbf{c}\|^2}$. The discrete Gaussian measure $\mathcal{D}_{\Lambda,s,\mathbf{c}}$ defined on the lattice Λ is given by:*

$$\mathcal{D}_{\Lambda,s,\mathbf{c}} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)},$$

where $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(x)$.

Definition 12. *(* Statistical distance*[27]) Let $M \subset \mathbb{R}^n$ be a finite or countable set, and let X and Y be two discrete random variables taking values in M. The statistical* *distance between X and Y is defined as:*

$$\triangle(X,Y) = \frac{1}{2}\sum_{a\in M} |P\{X=a\} - P\{Y=a\}|.$$

When the statistical distance between two distributions is less than a negligible value, we say that the two distributions are statistically close.

Definition 13. *(* Min-entropy*[16]) For a random variable X, its min-entropy is defined as:*

$$H_\infty(X) = -\log\left(\max_{x\in X}\Pr[X=x]\right).$$

The average conditional min-entropy of a random variable X conditional on a correlated variable Y is defined as:

$$H_\infty(X|Y) = -\log\left(\mathbb{E}_{y\in Y}\left\{\max_{x\in X}\Pr[X=x|Y=y]\right\}\right).$$

The optimal probability of an unbounded adversary guessing X given the correlated value Y is $2^{-H_\infty(X|Y)}$.

Lemma 1. *([16]) Let X, Y be arbitrarily random variables where the support of Y lies in $\mathcal{Y}$, Then*

$$H_\infty(X|Y) > H_\infty(X) - \log(|\mathcal{Y}|).$$

Micciancio and Goldwasser [29] proved that a full-rank set S in the lattice Λ can be transformed into a basis T such that both have similarly low Gram-Schmidt norms.

Lemma 2. *([29],* Lemma 7.1*) Let Λ be an n-dimensional lattice. There is a deterministic polynomial-time algorithm that, given an arbitrary basis of and a full-rank set $S = \{s_1, ..., s_n\}$ in Λ, returns a basis T of Λ satisfying*

$$\| \widetilde{T} \| \leq \| \widetilde{S} \| \textit{ and } \| T \| \leq \| S \| \frac{\sqrt{n}}{2}.$$

Ajtai [30], and later Alwen and Peikert [22], presented a method for sampling a matrix $A \in \mathbb{Z}_q^{n\times m}$ that is statistically close to uniform, along with an associated basis S_A for the lattice $\Lambda_q^\perp(A)$, which has a low Gram-Schmidt norm. The result below is derived from Theorem 3.2 in [22], where the parameter δ is set to $1/3$. The theorem guarantees the construction of a matrix A that is statistically close to uniformly distributed in $\mathbb{Z}_q^{n\times m}$, accompanied by a short basis. Given that m is significantly larger than n, the matrix A is rank n with overwhelming probability. Therefore, the theorem can be interpreted as stating that A is statistically close to a uniform rank n matrix from $\mathbb{Z}_q^{n\times m}$.

Theorem 2. *([22]) Let $q \geq 3$ be odd, n be a positive integer, and let $m := \lceil 6n \log q \rceil$. There is a probabilistic polynomial-time algorithm* ***TrapGen****(q, n, m) that outputs a pair $(A \in \mathbb{Z}_q^{n\times m}, T_A \in \mathbb{Z}^{n\times n})$ such that A is statistically close to a uniform rank n matrix in $\mathbb{Z}_q^{n\times m}$, and T_A is a basis for $\Lambda_q^{\perp}(A)$ satisfying*

$$\parallel \widetilde{T_A} \parallel \leq \mathcal{O}(\sqrt{n \log q}) \quad \textit{and} \quad \parallel T_A \parallel \leq \mathcal{O}(n \log q).$$

Gentry et al.[23] propose an algorithm capable of sampling from a discrete Gaussian distribution utilizing an arbitrary short basis.

Lemma 3. *(*Sampling from discrete Gaussian *[23]) Let $q \geq 2$, $A \in \mathbb{Z}_q^{n\times m}$ with $m > n$ and let T_A be a basis for $\Lambda_q^{\perp}(A)$ and $s \geq \widetilde{T_A} \cdot \omega(\sqrt{\log m})$. Then for $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{Z}_q^n$:*

1. *There is a probabilistic polynomial-time algorithm* ***SampleGaussian****$(A, T_A, s, \mathbf{c})$ that outputs $\mathbf{x} \in \Lambda_q^{\perp}(A)$ drawn from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\perp}(A),s,\mathbf{c}}$*
2. *There is a probabilistic polynomial-time algorithm* ***SamplePre****$(A, T_A, \mathbf{u}, s)$ that outputs $\mathbf{x} \in \Lambda_q^{\mathbf{u}}(A)$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(A),s}$, whenever $\Lambda_q^{\mathbf{u}}(A)$ is not empty.*

When $s \geq \omega(\sqrt{\log n})$, we denote the Gaussian sampling algorithm over the integer lattice $\mathbb{Z}^n$ as **SampleDom**$(1^n, s)$. That is, when $\mathbf{x} \overset{\$}{\leftarrow}$ **SampleDom**$(1^n, s)$, $\mathbf{x}$ is statistically close to the distribution $\mathcal{D}_{\mathbb{Z}^n,s}$. And these preimages have a conditional min-entropy of $\omega(\log n)$[23].

Lemma 4. *([23]) Let n and h be positive integers, and q be a prime, such that $n \geq 2h \lg q$. Then for all but a $2q^{-h}$ fraction of all $A \in \mathbb{Z}_q^{h\times n}$ and for any $s \geq \omega(\sqrt{\log n})$, the distribution of the $\alpha = A \cdot \mathbf{x}(\mathrm{mod} q)$ is statistically close to uniform over $\mathbb{Z}_q^h$, where $\mathbf{x} \overset{\$}{\leftarrow}$* ***SampleDom****$(1^n, s)$.*

Definition 14. *(* Smoothing parameter*[31]) For any n-dimensional lattice Λ and any given $\epsilon > 0$, the smoothing parameter of the lattice is defined as*

$$\eta_\epsilon(\Lambda) = \min\left\{ s > 0 \mid \rho_{\frac{1}{s}}(\Lambda^*) < 1 + \epsilon \right\}.$$

For the vast majority of matrices $A \in \mathbb{Z}_q^{h\times n}$, there exists a negligible value ϵ such that the smoothing parameter $\eta_\epsilon(\Lambda_q^{\perp}(\mathbf{A}))$ is smaller than $\omega(\sqrt{\log n})$:

Lemma 5. *([23]) Let $q \geq 3$, h and n be positive integers satisfying $n \geq 2h \lg q$. Then there exists a negligible function $\epsilon(n)$ such that $\eta_\epsilon(\Lambda_q^{\perp}(A)) < \omega(\sqrt{\log n})$ for all but at most a q^{-h} fraction of A in the $\mathbb{Z}_q^{h\times n}$.*

Theorem 3. *([32]) $\boldsymbol{t}_i \in \mathbb{Z}^m$ and $\boldsymbol{x}_i$ are mutually independent random variables sampled from a Gaussian distribution $D_{\boldsymbol{t}_i+\Lambda,\sigma}$ over $\boldsymbol{t}_i + \Lambda$ for $i = 1, 2, \cdots, k$ in which Λ is* *a lattice and $\sigma \in \mathbb{R}$ is a parameter. Let $\boldsymbol{c} = (c_1, \cdots, c_k) \in \mathbb{Z}^k$ and $g = \gcd(c_1, \cdots, c_k)$, $\boldsymbol{t} = \sum_{i=1}^{k} c_i \boldsymbol{t}_i$. If $\sigma > \|\boldsymbol{c}\| \eta_\epsilon(\Lambda)$ for some negligible number ϵ, then $\boldsymbol{z} = \sum_{i=1}^{k} c_i \boldsymbol{x}_i$ statistically closes to $D_{\mathbf{t}+g\Lambda, \|\mathbf{c}\|\sigma}$.*

The subsequent lemma demonstrates that the lengths of vectors obtained through sampling from a discrete Gaussian distribution are predominantly concentrated within a specific bound.

Lemma 6. *([31]) Let Λ be an n-dimensional lattice, and T be a basis of the lattice Λ. If $s \geq \| \widetilde{T} \| \cdot \omega(\sqrt{\log n})$, then for any $\mathbf{c} \in \mathbb{R}^n$, we have:*

$$\Pr\{\| \mathbf{x} - \mathbf{c} \| > s\sqrt{n} : \mathbf{x} \leftarrow \mathcal{D}_{\Lambda, s, \mathbf{c}}\} \leq \mathrm{negl}(n).$$

Agrawal et al.[22] proposed a new lattice basis delegation algorithm, **BasisDel**(A, R, T_A, s), in which the lattice dimension remains unchanged during the delegation process. Inspired by their **BasisDel** algorithm, Guo et al.[24] developed a new algorithm **NewBasis**(A, H, T_A), for generating trapdoors in new lattices. This algorithm is a special case of **BasisDel**(A, R, T_A, s) algorithm, where the matrix R is replaced by an orthogonal matrix H. The detailed conclusions are as follows.

Theorem 4. *([24]) Let $q \geq 3$ be odd, h be a positive integer, and let $n := \lceil 6h \log q \rceil$. There exists a deterministic polynomial-time algorithm denoted as **NewBasis**(A, H, T_A) Its inputs are: a matrix $A \in \mathbb{Z}_q^{h \times n}$ of rank h, a orthogonal matrix $H \in \mathbb{Z}^{n \times n}$, and T_A which is a basis of $\Lambda_q^{\perp}(A)$. The output of this algorithm is a basis T_B of the lattice $\Lambda_q^{\perp}(B)$, where $B = AH^{\top}$, and $\| \widetilde{T_B} \| < \| \widetilde{T_A} \|$, $\| T_B \| < \| T_A \| \sqrt{n}/2$*

The following three lemmas establish fundamental properties of orthogonal matrices and Gaussian distributions over lattices.

Lemma 7. *([24]) Let $A \in \mathbb{R}^{n \times n}$ be an n-dimensional full-rank matrix, and $H \in \mathbb{R}^{n \times n}$ be an orthogonal matrix. Then,*

$$\| HA \| = \| A \| \quad \text{and} \quad \| \widetilde{HA} \| = \| \widetilde{A} \| .$$

Lemma 8. *([24]) Let H be an orthogonal matrix over $\mathbb{Z}^{n \times n}$, that is, $HH^{\top} = I_n$. If $\mathbf{x} \sim \mathcal{D}_{\mathbb{Z}^n, \sigma}$, then $H\mathbf{x} \sim \mathcal{D}_{\mathbb{Z}^n, \sigma}$.*

Lemma 9. *([24]) Let $A \in \mathbb{Z}_q^{n \times m}$, $s > 0$, $\mathbf{u} \in \mathbb{Z}^m$, and $\Lambda = \Lambda_q^{\boldsymbol{u}}(A)$. If $\mathbf{x}$ is sampled from $\mathcal{D}_{\mathbb{Z}^m, s}$ conditioned on $A\mathbf{x} \equiv \boldsymbol{u} \pmod q$, then the conditional distribution of $\mathbf{x}$ is $\mathcal{D}_{\Lambda, s}$.*

4 Semigroup homomorphic signature

4.1 Construction

Before presenting our semigroup homomorphic signature, we first need to define our message space and signature space. As introduced in the Introduction section, we choose $\mathcal{M} = ((\mathbb{Z}_+)^*, \|)$ as the message space and $\Sigma = ((\mathbb{Z}^n)^*, \|)$ as the signature space, both of which are semigroups without inverses. Note: if $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1\boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_k \in (\mathbb{Z}^n)^*$, we also treat $\boldsymbol{\sigma}$ as a matrix $[\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, ..., \boldsymbol{\sigma}_k] \in \mathbb{Z}^{n\times k}$. If $\mathbf{x} \in (\mathbb{Z}_+)^*$, without loss of generality, assume $\mathbf{x} = x_1x_2 \cdots x_k \in (\mathbb{Z}_+)^*$. We define $|\mathbf{x}|$ as the length of $\mathbf{x}$, that is, $|\mathbf{x}| = k$.

Our semigroup-homomorphic signature scheme, $\mathcal{SH} = (\mathbf{Gen}, \mathbf{Sign}, \mathbf{Verify})$ works as follows:

$\mathbf{Gen}(1^n)$: Input the secure parameters n. Let $q = \text{poly}(n)$, $k = \text{poly}(n)$ and $q \geq (kn)^2$, $h = \lfloor \frac{n}{6\log q} \rfloor$, and $V = k\sqrt{2n\log q} \cdot \log n$. The algorithm generates a pair of public key and secret key as follows:

(1) Compute $(A, T_A) \longleftarrow \mathbf{TrapGen}(q, h, n)$, where $A \in \mathbb{Z}_q^{h\times n}$ and T_A is a basis for the lattice $\Lambda_q^{\perp}(A)$.
(2) Select a secure collision-resistant hash function $\mathbf{h} : \mathbb{Z}_+ = \{0, 1\}^* \to \{0, 1\}^k$. Note that the hash function in our context is regarded as a deterministic algorithm rather than an idealized random oracle.
(3) Sample $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k \xleftarrow{\$} \mathbb{Z}_q^h$. We require that $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ be linearly independent. If not, resampling should be carried out, and this process can be easily accomplished within polynomial time.
(4) Output the public key $pk = (A, \mathbf{h}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$ and the secret key $sk = T_A$.

$\mathbf{Sign}(sk, \mathbf{x})$: Input the secret key $sk = T_A$, the message $\mathbf{x} = x_1, ..., x_{|\mathbf{x}|} \in (\mathbb{Z}_+)^*$. the signing algorithm proceeds as follows:

(1) Compute $\mathbf{v}_i = \mathbf{h}(x_i) = (\mathbf{v}_{i1}, ..., \mathbf{v}_{ik})$ and $\boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{v}_{ij} \cdot \boldsymbol{\alpha}_j (\text{mod} q)$, for $i = 1, ..., |\mathbf{x}|$
(2) Compute $\boldsymbol{\sigma}_i \leftarrow \mathbf{SamplePre}(A, T_A, \boldsymbol{\beta}_i, V)$, for $i = 1, ..., |\mathbf{x}|$.

Output signature $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1\boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{|\mathbf{x}|} \in (\mathbb{Z}^n)^*$.

$\mathbf{Verify}(pk, \mathbf{x}, \boldsymbol{\sigma})$: Input the public key $pk = (A, \mathbf{h}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$, $\mathbf{x} = x_1x_2 \cdots x_{|\mathbf{x}|} \in \mathcal{M}$ and $\boldsymbol{\sigma}$. The verification process is as follows:

(1) Compute $\mathbf{v}_i = \mathbf{h}(x_i)$, $\boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{v}_{ij} \cdot \boldsymbol{\alpha}_j (\text{mod} q)$, for $i = 1, .., |\mathbf{x}|$. Let

$$B = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{|\mathbf{x}|}] \quad \text{and} \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}_1\boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{|\mathbf{x}|} = [\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, ..., \boldsymbol{\sigma}_{|\mathbf{x}|}].$$

(2) Output 1 if all of the following two conditions are satisfied; otherwise, output 0.
1) $\|\boldsymbol{\sigma}\| \leq V \cdot \sqrt{kn}$; ($\| \cdot \|$ represents the matrix norm.)
2) $A \cdot \boldsymbol{\sigma} (\text{mod} q) = B$. (Note: $A \cdot \boldsymbol{\sigma} (\text{mod} q) = [A \cdot \boldsymbol{\sigma}_1, ..., A \cdot \boldsymbol{\sigma}_{|\mathbf{x}|}] (\text{mod} q)$)

Remark 2: In our signature scheme, in order to avoid trivial forgeries, we actually sign only the messages $x_1, ..., x_k \in \mathbb{Z}_+$ with the private key, while other messages $\mathbf{x} \in (\mathbb{Z}_+)^*$, which are not signed with the private key, are derived through homomorphic operations, if $\mathbf{x} \in \mathrm{span}_{\|}\{x_1, ..., x_k\}$.

4.2 Correctness

We will now prove that the signature scheme we constructed is correct.

Theorem 5. *The above signature scheme $\mathcal{HS}$ satisfies correctness with overwhelming probability.*

Proof. Assume that $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{|\mathbf{x}|} \in (\mathbb{Z}^n)^*$ is the signature of the message $\mathbf{x} = x_1 x_2 \cdots x_{|\mathbf{x}|} \in (\mathbb{Z}_+)^*$, i.e.,

$$\boldsymbol{\sigma}_i \leftarrow \mathbf{SamplePre}(A, T_A, \boldsymbol{\beta}_i, V), \quad \text{where} \quad \boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{v}_{ij} \cdot \boldsymbol{\alpha}_j (\mathrm{mod} q), \quad 1 \leq i \leq |\mathbf{x}|.$$

From Theorem 2, it follows that $\| \widetilde{T}_A \| \leq \mathcal{O}(\sqrt{h \log q})$ with overwhelming probability. Since $V = \sqrt{2n \log q} \cdot \log n$, it follows that

$$\frac{V}{\| \widetilde{T}_A \|} \geq k \cdot \sqrt{\frac{2n}{h}} \cdot \log n \geq \sqrt{\log n},$$

which implies $V \geq \| \widetilde{T}_A \| \cdot \omega(\sqrt{\log n})$. Therefore, by Lemma 6, we have $\| \boldsymbol{\sigma}_i \| \leq V \cdot \sqrt{n}$ with overwhelming probability. Thus,

$$\| \boldsymbol{\sigma} \| = \max_{1 \leq i \leq |x|} \| \boldsymbol{\sigma}_i \| \leq V \cdot \sqrt{n} \leq V \cdot \sqrt{k \cdot n}.$$

Finally, according to Lemma 3 we have $A \cdot \boldsymbol{\sigma}_i (\mathrm{mod} q) = \boldsymbol{\beta}_i$, for $i = 1, ..., |\mathbf{x}|$. Thus

$$A \cdot \boldsymbol{\sigma} (\mathrm{mod} q) = [A \cdot \boldsymbol{\sigma}_1, ..., A \cdot \boldsymbol{\sigma}_{|\mathbf{x}|}] (\mathrm{mod} q) = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{|\mathbf{x}|}] = B$$

This completes the proof of the theorem. □

4.3 Homomorphism

We will now prove that the signature scheme satisfies homomorphism.

Theorem 6. *The above signature scheme $\mathcal{HS}$ is homomorphic with respect to the operation $\|$. That is, for any $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}_+)^*$, we have*

$$\boldsymbol{Sign}(pk, \mathbf{x} \| \mathbf{y}) = \boldsymbol{Sign}(pk, \mathbf{x}) \| \boldsymbol{Sign}(pk, \mathbf{y}).$$

Proof. For any $\mathbf{x}, \mathbf{y} \in (\mathbb{Z}_+)^*$, assume that

$$\begin{cases} \boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_{11}\boldsymbol{\sigma}_{12}\cdots\boldsymbol{\sigma}_{1|\mathbf{x}|} \leftarrow \text{Sign}(pk, \mathbf{x}) \\ \boldsymbol{\sigma}_2 = \boldsymbol{\sigma}_{21}\boldsymbol{\sigma}_{22}\cdots\boldsymbol{\sigma}_{2|\mathbf{y}|} \leftarrow \text{Sign}(pk, \mathbf{y}), \end{cases}$$

where

$$\begin{cases} \boldsymbol{\sigma}_{1i} \leftarrow \mathbf{SamplePre}(A, T_A, \boldsymbol{\beta}_{1i}, V), \quad \boldsymbol{\beta}_{1i} = \sum_{l=1}^{k} \mathbf{h}(x_i)_l \cdot \boldsymbol{\alpha}_l (\text{mod} q), \quad i = 1, \dots, |\mathbf{x}|, \\ \boldsymbol{\sigma}_{2j} \leftarrow \mathbf{SamplePre}(A, T_A, \boldsymbol{\beta}_{2j}, V), \quad \boldsymbol{\beta}_{2j} = \sum_{l=1}^{k} \mathbf{h}(\mathbf{y}_j)_l \cdot \boldsymbol{\alpha}_l (\text{mod} q), \quad j = 1, \dots, |\mathbf{y}|. \end{cases}$$

From the verification algorithm, we know that:

$$\begin{cases} A \cdot \boldsymbol{\sigma}_{1i}(\text{mod} q) = \boldsymbol{\beta}_{1i}, i = 1, ..., |\mathbf{x}|, \\ A \cdot \boldsymbol{\sigma}_{2i}(\text{mod} q) = \boldsymbol{\beta}_{2i}, i = 1, ..., |\mathbf{y}|. \end{cases}$$

Let $B_1 = [\boldsymbol{\beta}_{11}, \dots, \boldsymbol{\beta}_{1|\mathbf{x}|}]$, $B_2 = [\boldsymbol{\beta}_{21}, \dots, \boldsymbol{\beta}_{2|\mathbf{y}|}]$, $\mathbf{z} = \mathbf{x} \parallel \mathbf{y}$, and $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 \parallel \boldsymbol{\sigma}_2$. Next, calculate $\boldsymbol{\beta}_i = \sum_{l=1}^{k} \mathbf{h}(\mathbf{z}_i)_l \cdot \boldsymbol{\alpha}_l (\text{mod}\, q)$, for $1 \le i \le |\mathbf{z}| = |\mathbf{x}| + |\mathbf{y}|$. Let $B = [\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_{|\mathbf{z}|}]$, then $B = [B_1, B_2]$.

Since

$$\begin{cases} A \cdot \boldsymbol{\sigma}_1(\text{mod} q) = B_1, \\ A \cdot \boldsymbol{\sigma}_2(\text{mod} q) = B_2. \end{cases}$$

Thus, $A \cdot \boldsymbol{\sigma} = A \cdot [\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2](\text{mod} q) = [B_1, B_2]$. Clearly, $\parallel \boldsymbol{\sigma} \parallel < V \cdot \sqrt{k \cdot n}$.

Therefore, according to the verification algorithm,

$$\mathbf{Verify}(pk, \mathbf{z}, \boldsymbol{\sigma}) = \mathbf{Verify}(pk, \mathbf{x} \parallel \mathbf{y}, \boldsymbol{\sigma}_1 \parallel \boldsymbol{\sigma_2}) = 1.$$

Therefore,

$$\mathbf{Sign}(pk, \mathbf{x} \parallel \mathbf{y}) = \mathbf{Sign}(pk, \mathbf{x}) \parallel \mathbf{Sign}(pk, \mathbf{y}).$$

□

4.4 Security

In this section, we will prove that our semigroup-homomorphic signature scheme satisfies EUF-CMA-FMR and is tightly secure. Before presenting the main conclusion, we need to rule out a possible forgery.

Let $\mathbf{h} : \mathbb{Z}_+ \to \{0, 1\}^k$ be a collision-resistant hash function. If one can find $x_1, x_2 \in \mathbb{Z}_+$ with $x_1 \neq x_2$ such that $\sum_{j=1}^{k} \mathbf{h}(x_1)_j \cdot \boldsymbol{\alpha}_j = \sum_{j=1}^{k} \mathbf{h}(x_2)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q)$. According to the signature algorithm, if the signature of x_1 is $\boldsymbol{\sigma}$, then the signature of x_2 will also be $\boldsymbol{\sigma}$ thereby resulting in a forgery. However, such a situation does not exist, and we have the following lemma to prove this.

Lemma 10. *Let* $\mathbf{h} : \mathbb{Z}_+ \to \{0,1\}^k$ *be a collision-resistant hash function. If for any* $x_1, x_2 \in \mathbb{Z}_+$*, it holds that* $\sum_{j=1}^{k} \mathbf{h}(x_1)_j \cdot \boldsymbol{\alpha}_j = \sum_{j=1}^{k} \mathbf{h}(x_2)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q)$*, then with overwhelming probability,* $x_1 = x_2$*.*

Proof. According to the definition of the signature algorithm, $\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k$ are linearly independent. Suppose there exist $x_1, x_2 \in \mathbb{Z}_+$ such that

$$\sum_{j=1}^{k} \mathbf{h}(x_1)_j \cdot \boldsymbol{\alpha}_j = \sum_{j=1}^{k} \mathbf{h}(x_2)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q).$$

Then it must be the case that $\mathbf{h}(x_1)_j = \mathbf{h}(x_2)_j$ for $j = 1, .., k$, that is, $\mathbf{h}(x_1) = \mathbf{h}(x_2)$. Since the hash function $\mathbf{h}$ is collision-resistant, with overwhelming probability, $x_1 = x_2$. □

According to Lemma 10, if a polynomial-time adversary $\mathcal{A}$ finds a pair of distinct messages (x_1, x_2) such that $\sum_{j=1}^{k} \mathbf{h}(x_1)_j \cdot \boldsymbol{\alpha}_j = \sum_{j=1}^{k} \mathbf{h}(x_2)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q)$, then it must be the case that $\mathbf{h}(x_1) = \mathbf{h}(x_2)$. That is, the adversary $\mathcal{A}$ has found a pair of collision points of the hash function. Obviously, the probability of such a situation is negligible.

We now provide the security proof of the proposed signature scheme.

Theorem 7. *Suppose* $q = \text{poly}(n)$*,* $k = \text{poly}(n)$ *and* $q \geq (kn)^2$*,* $h = \lfloor \frac{n}{6 \log q} \rfloor$*,* $V = k\sqrt{2n \log q} \cdot \log n$*. If the* $\boldsymbol{SIS}_{q,\beta}$ *problem is computationally hard for* $\beta = 2V \cdot \sqrt{kn}$*, then the above semigroup-homomorphic signature* $\mathcal{HS}$ *is existentially unforgeable under the adaptive chosen-message attack with a fixed message range (EUF-CMA-FMR).*

More specifically, Suppose the adversary $\mathcal{A}$ *is allowed to make at most* q_s *chosen-message queries on message subset* $\mathbb{Z}_+ (\subseteq \mathcal{M} = (\mathbb{Z}_+)^*)$ *and can break the above semigroup-homomorphic signature scheme* $\mathcal{SH}$ *with advantage* ϵ *within time* t*. Then, the simulator* $\mathcal{C}$ *can use the adversary's ability to construct an algorithm* $\mathcal{B}$ *that outputs a solution to the* $\boldsymbol{SIS}_{q,2V\cdot\sqrt{kn}}$ *problem with advantage* $\epsilon - \text{negl}(n)$ *within time* $t + \mathcal{O}(q_s T_{\mathcal{SH}} + T_{\text{SampleDom}})$*, where* $T_{\mathcal{SH}}$ *represents the time required for the simulator to generate a signature, and* $T_{\text{SampleDom}}$ *represents the time for running the sampling algorithm* ***SampleDom*** *once.*

Proof. Let $q = \text{poly}(n)$, $k = \text{poly}(\log n)$ and $q \geq (kn)^2$, $h = \lfloor \frac{n}{6 \log q} \rfloor$, a security hash $\mathbf{h} : \mathbb{Z}_+ \to \{0,1\}^*$, $V = k\sqrt{2n \log q} \cdot \log n$.

The simulator $\mathcal{C}$ generates the public key through the following steps:

(1) Randomly and uniformly select a matrix $A \in \mathbb{Z}_q^{h \times n}$. We regard A as an instance of the **SIS** problem.

(2) Sample k vectors: $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k \leftarrow$ **SampleDome**$(1^n, s)$, where where $s = V/\sqrt{k} \geq \omega(\sqrt{\log n})$. Let $\boldsymbol{\alpha}_i = A \cdot \boldsymbol{\gamma}_i (\text{mod} q) \in \mathbb{Z}_q^h$ for $i = 1, ..., k$. Here, we also require that

$\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k$ are linearly independent. If they are not, repeat this step. According to [33], the expected value of the number of repetitions will not exceed 2.

(3) Select a secure collision-resistant hash function $\mathbf{h} : \mathbb{Z}_+ = \{0,1\}^* \to \{0,1\}^k$

(4) Send $pk = (A, \mathbf{h}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k)$ to the adversary $\mathcal{A}$.

After receiving the public key, the adversary $\mathcal{A}$ chooses q_s messages $x_1, ..., x_{q_s} \in \mathbb{Z}_+$ as signature queries. The simulator $\mathcal{C}$ generates the signature through the following steps:

(1) For the message x_i, compute $\mathbf{v}_i = \mathbf{h}(x_i)$ and $\mathbf{t}_i = \sum_{j=1}^{k} \mathbf{v}_{ij}\boldsymbol{\gamma}_j$.

(2) Let $\boldsymbol{\sigma}_i = \mathbf{t}_i$.

(3) Output the signatures $\boldsymbol{\sigma}_i$, for $1 \leq i \leq q_s$.

Send the signatures $\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_{q_s}$ the adversary $\mathcal{A}$.

Next, we prove that the output distribution of the simulator is indistinguishable (within negligible statistical distance) from the output distribution in the real signature scheme.

First, in the real signature scheme, the matrix A is generated by the trapdoor generation algorithm **TrapGen**. According to Theorem 2, the matrix A statistically follows a uniform distribution. Secondly, according to Lemma 4, $\boldsymbol{\alpha}_1 = A \cdot \boldsymbol{\gamma}_1, ..., \boldsymbol{\alpha}_k = A \cdot \boldsymbol{\gamma}_k$ also statistically follow a uniform distribution. Therefore, in terms of the generation of the public key, it is statistically indistinguishable from that in the real signature scheme.

Secondly, according to Theorem 3, $\boldsymbol{\sigma}_i = \mathbf{t}_i = \sum_{j=1}^{k} \mathbf{v}_{ij}\boldsymbol{\gamma}_j$ follows the distribution $\mathcal{D}_{\mathbb{Z}^n,V}$. Moreover, since

$$A \cdot \boldsymbol{\sigma}_i = A \cdot \mathbf{t}_i = A \cdot \sum_{j=1}^{k} \mathbf{v}_{ij}\boldsymbol{\gamma}_j = \sum_{j=1}^{k} \mathbf{v}_{ij} A \cdot \boldsymbol{\gamma}_j = \sum_{j=1}^{k} \mathbf{v}_{ij}\boldsymbol{\alpha}_j = \boldsymbol{\beta}_i (\mathrm{mod} q),$$

by Lemma 9, $\boldsymbol{\sigma}_i = \mathbf{t}_i$ follows the distribution $\mathcal{D}_{\Lambda,V}$, where $\Lambda_i = \Lambda_q^{\boldsymbol{\beta}_i}(A)$. Finally, according to Lemma 6, we have

$$\| \sum_{j=1}^{k} \mathbf{v}_{ij}\boldsymbol{\gamma}_j \| \leq k \cdot \max_{1 \leq j \leq k} \| \mathbf{v}_{ij} \| \leq k \cdot \frac{V}{\sqrt{k}}\sqrt{n} = V \cdot \sqrt{k \cdot n}.$$

Therefore, the signatures generated by the simulator are also statistically indistinguishable from those of the real scheme.

After receiving the signature queries, the adversary $\mathcal{A}$ finally outputs a forged $(\mathbf{x}^*, \boldsymbol{\sigma}^*)$, where $\mathbf{x}^* \notin \mathrm{span}_{(\|)}\{x_1, ..., x_{q_s}\}$.

From verification condition 2), it can be concluded that:

$$\begin{aligned} A \cdot \boldsymbol{\sigma}^* (\bmod q) &= A \cdot [\boldsymbol{\sigma}_1^*, \ldots, \boldsymbol{\sigma}_{|\mathbf{x}^*|}^*] (\bmod q) \\ &= [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{|\mathbf{x}^*|}] \\ &= [\sum_{i=1}^{k} \mathbf{v}_{1j}^* \cdot \boldsymbol{\alpha}_j, \ldots, \sum_{i=1}^{k} \mathbf{v}_{kj}^* \cdot \boldsymbol{\alpha}_j] \\ &= [\sum_{i=1}^{k} \mathbf{v}_{1j}^* \cdot A \cdot \boldsymbol{\gamma}_j, \ldots, \sum_{i=1}^{k} \mathbf{v}_{kj}^* \cdot A \cdot \boldsymbol{\gamma}_j] (\bmod q). \end{aligned}$$

Take any arbitrary i, then

$$A \cdot \boldsymbol{\sigma}_i^* = \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot A \cdot \boldsymbol{\gamma}_j = A \cdot \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j (\bmod).$$

From verification condition 1), we know that: $\parallel \boldsymbol{\sigma}_i^* \parallel \leq V \cdot \sqrt{k \cdot n}$. By Lemma 6,

$$\parallel \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j \parallel \leq k \cdot \frac{V}{\sqrt{k}} \cdot \sqrt{n} = V \cdot \sqrt{k \cdot n}$$

holds with overwhelming probability, so $\parallel \boldsymbol{\sigma}_i^* - \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j \parallel \leq 2V \cdot \sqrt{k \cdot n}$.

If $\boldsymbol{\sigma}_i^* \neq \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j$, then the simulator outputs $\boldsymbol{\sigma}_i^* - \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j$ as a solution to the $\mathbf{SIS}_{q,2V \cdot \sqrt{k \cdot n}}$. Let the process of outputting $\boldsymbol{\sigma}_i^* - \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j$ be denoted as algorithm $\mathcal{B}$.

Next, we discuss the probability that $\boldsymbol{\sigma}_i^* \neq \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j$. Since the preimage of the algorithm **SampleDom** has conditional min-entropy $\omega(\log n)$, thus

$$\Pr[\boldsymbol{\sigma}_i^* = \sum_{i=1}^{k} \mathbf{v}_{ij}^* \cdot \boldsymbol{\gamma}_j] \leq 2^{-\omega(\log n)} = \operatorname{negl}(n).$$

Therefore, the simulator can construct an algorithm $\mathcal{B}$ by using the adversary $\mathcal{A}$'s ability that, within time $t + \mathcal{O}(q_s T_{\mathcal{SH}} + T_{\text{SampleDom}})$ outputs a solution to the $\mathbf{SIS}_{q,2V \cdot \sqrt{k \cdot n}}$ problem with advantage $\epsilon - \operatorname{negl}(n)$. □

Worst-case connections. According to Theorem 1 ([23]), in the situation where $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the $\mathbf{SIS}_{q,\beta}$ problem is as hard as approximating the SIVP problem in the worst - case scenario with a factor of $\beta \cdot \tilde{O}(\sqrt{n})$. In the Setup algorithm, the requirement $q \geq (nk)^2$ serves to ensure that q has a large enough value for Theorem 1 to be applicable. Even though the exact value of the worst - case approximation factor is determined by the parameter k, it is always a polynomial function of n regardless of the specific value of k.

5 A Linearly Semigroup-Homomorphic Signature Scheme

Based on the semigroup-homomorphic signature construction in Section 4, we will construct a secure linearly semigroup homomorphic signature. First, we introduce the linear operations on the semigroup.

(1) In this section, the message space and signature space we choose are still $((\mathbb{Z}_+)^*, \|)$ and $((\mathbb{Z}^n)^*, \|)$ respectively. In addition, we require that the empty string $\varnothing \in (\mathbb{Z}_+)^*$, and it satisfies that for any $\mathbf{v} \in (\mathbb{Z}_+)^*$,

$$\mathbf{v} \parallel \varnothing = \varnothing \parallel \mathbf{v} = \mathbf{v}.$$

Next, we define the signature of $\varnothing$ as $\aleph$. It is required that $\aleph \in (\mathbb{Z}^n)^*$, and for any $\boldsymbol{\sigma} \in (\mathbb{Z}^n)^*$,

$$\aleph \parallel \boldsymbol{\sigma} = \boldsymbol{\sigma} \parallel \aleph = \boldsymbol{\sigma}.$$

(2) Let $R = \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, $c \in \mathbb{Z}_p$, $c \neq 0$, and for any $\mathbf{v} \in (\mathbb{Z}_+)^*$, we define the scalar multiplication operation $c \cdot \mathbf{v}$ as follows:

$$c \cdot \mathbf{v} = \underbrace{\mathbf{v} \parallel \cdots \parallel \mathbf{v}}_{c}.$$

If $c = 0$, then for any $\mathbf{v} \in (\mathbb{Z}_+)^*$, we define $0 \cdot \mathbf{v} = \varnothing$. Therefore, if $\langle f \rangle = (c_1, \ldots, c_k)$, where $c_i \in \mathbb{Z}_p$, then for any
$mathbf{v}_1, \ldots, \mathbf{v}_k \in (\mathbb{Z}_+)^*$, we have

$$f(\mathbf{v}_1, \ldots, \mathbf{v}_k) = c_1\mathbf{v}_1 \parallel \cdots \parallel c_k\mathbf{v}_k = \underbrace{\mathbf{v}_1 \parallel \cdots \parallel \mathbf{v}_1}_{c_1} \parallel \cdots \parallel \underbrace{\mathbf{v}_k \parallel \cdots \parallel \mathbf{v}_k}_{c_k}.$$

(3) Let $R = \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, $c \in \mathbb{Z}_p$, $c \neq 0$, and for any $\boldsymbol{\sigma} \in (\mathbb{Z}^n)^*$, we define the scalar multiplication operation $c \cdot \boldsymbol{\sigma}$ as follows:

$$c \cdot \boldsymbol{\sigma} = \underbrace{\boldsymbol{\sigma} \parallel \cdots \parallel \boldsymbol{\sigma}}_{c} = \underbrace{[\boldsymbol{\sigma}, \boldsymbol{\sigma}, \ldots, \boldsymbol{\sigma}]}_{c}.$$

Note: When a signature $\boldsymbol{\sigma}$ is written as $[\boldsymbol{\sigma}]$, it indicates that it is considered as a matrix at that moment. If $c = 0$, then for any $\boldsymbol{\sigma} \in (\mathbb{Z}^n)^*$, we define $0 \cdot \boldsymbol{\sigma} = \aleph$. Therefore, if $\langle f \rangle = (c_1, \ldots, c_k)$, where $c_i \in \mathbb{Z}_p$, then for any $\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k \in (\mathbb{Z}^n)^*$, we have

$$f(\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_k) = c_1 \cdot \boldsymbol{\sigma}_1 \parallel \cdots \parallel c_k \cdot \boldsymbol{\sigma}_k = [c_1 \cdot \boldsymbol{\sigma}_1, \cdots, c_k \cdot \boldsymbol{\sigma}_k] = [\underbrace{\boldsymbol{\sigma}_1, \cdots, \boldsymbol{\sigma}_1}_{c_1}, \cdots, \underbrace{\boldsymbol{\sigma}_k, \cdots, \boldsymbol{\sigma}_k}_{c_k}].$$

5.1 Construction

In the following, we present a formal description of our linearly semigroup-homomorphism signature scheme.

Our linearly semigroup-homomorphic signature scheme $\mathcal{LSH}$ = (**Setup**, **Sign**, **Combine**, **Verify**) works as follows:

Setup$(1^n, pp)$: The input consists of the security parameter n and the public parameters pp. The public parameters pp are defined as follows: $q = \text{poly}(n)$, $k = \text{poly}(n)$, and $q \geq (kn)^2$, $h = \left\lfloor \frac{n}{6 \log q} \right\rfloor$, $V = k\sqrt{2n \log q} \cdot \log n$.

Furthermore, the public parameters define a message space $\mathcal{M} = ((\mathbb{Z}_+)^*, \|)$ and a signature space $\Sigma = ((\mathbb{Z}^n)^*, \|)$. The algorithm generates a pair of public key and secret key as follows:

(1) Compute $(A, T_A) \longleftarrow$ **TrapGen**(q, h, n), where $A \in \mathbb{Z}_q^{h \times n}$, T_A is a basis for lattice $\Lambda_q^{\perp}(A)$.
(2) Sample k vectors: $\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k \overset{\$}{\leftarrow} \mathbb{Z}_q^h$. These vectors are linearly independent.
(3) Select a secure collision-resistant hash function $\mathbf{h} : \mathbb{Z}_+ = \{0,1\}^* \to \{0,1\}^k$
(4) Output the public key $pk = (A, , \mathbf{h}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k)$ and the secret key $sk = T_A$.

Sign$(sk, \tau, \mathbf{v})$: Input the secret key $sk = T_A$, the tag $\tau \in \{0,1\}^n$, the message $\mathbf{v} = \mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_{|\mathbf{v}|} \in \mathcal{M}$. the signing algorithm proceeds as follows:

(1) Compute $H_\tau = \text{diag}\{2\tau_1 - 1, ..., 2\tau_n - 1\}$, where $\tau = (\tau_1, ..., \tau_n)$.
(2) Compute $T_{B^\tau} \leftarrow$**NewBasis**(A, H_τ, T_A), where T_{B^τ} is a basis for $\Lambda_q^{\perp}(B^\tau)$ and $B^\tau = AH_\tau^\top$.
(3) Compute $\boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{h}(\mathbf{v}_i)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q)$, for $1 \leq i \leq |\mathbf{v}|$.
(4) Compute $\boldsymbol{\sigma}_i \longleftarrow$ **SamplePre**$(B^\tau, T_{B^\tau}, \boldsymbol{\beta}_i, V)$, for $1 \leq i \leq |\mathbf{v}|$. Let $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{|\mathbf{v}|}$
(5) Output $\boldsymbol{\sigma}$ as signature.

Combine$(pk, \tau, \{(c_i, \boldsymbol{\sigma}_i)_{i=1}^l\})$: Given a public key pk, a data-set tag $\tau \in \{0,1\}^n$, and a set of tuples $(c_i, \boldsymbol{\sigma}_i)_{i=1}^l$, where $c_i \in R = \mathbb{Z}_p$ and $\boldsymbol{\sigma}_i \leftarrow \text{Sign}(sk, \tau, \mathbf{v}_i)$, this algorithm outputs a signature $\boldsymbol{\sigma} = c_1 \cdot \boldsymbol{\sigma}_1 \parallel \cdots \parallel c_l \cdot \boldsymbol{\sigma}_l$. (This $\boldsymbol{\sigma}$ is intended to be a signature on $c_1 \cdot \mathbf{v}_1 \parallel \cdots \parallel c_l \cdot \mathbf{v}_l$.)

Verify$(pk, \tau, \mathbf{y}, \boldsymbol{\sigma})$. On input a public key $pk = (A, , \mathbf{h}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k)$, an data-set tag $\tau \in \{0,1\}^n$, a vector $\mathbf{y} = \mathbf{y}_1 \mathbf{y}_2 \cdots \mathbf{y}_{|\mathbf{y}|} \in \mathcal{M}$, and a signature $\boldsymbol{\sigma} = \boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2 \cdots \boldsymbol{\sigma}_{|\mathbf{y}|} \in \Sigma$. The verification process is as follows:

(1) compute $H_\tau = \text{diag}\{2\tau_1 - 1, ..., 2\tau_n - 1\}$ and $B^\tau = AH_\tau^\top$.
(2) compute $\boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i)_j \cdot \boldsymbol{\alpha}_j (\text{mod} q)$, for $i = 1, ..., |\mathbf{y}|$. Let $Y = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{|\mathbf{y}|}]$.
(3) Output 1 if all of the following two conditions are satisfied; otherwise, output 0.
  1) $\| \boldsymbol{\sigma} \| \leq V \cdot \sqrt{kn}$.
  2) $B^\tau \cdot \boldsymbol{\sigma} (\text{mod} q) = Y$. (Note: $B^\tau \cdot \boldsymbol{\sigma} \equiv B^\tau \cdot [\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_{|\mathbf{y}|}] \equiv [B^\tau \cdot \boldsymbol{\sigma}_1, ..., B^\tau \cdot \boldsymbol{\sigma}_{|\mathbf{y}|}] (\text{mod} q)$)

Remark 3: Since this signature scheme is essentially built upon the semigroup-homomorphic signature constructed in Section 4, in order to avoid similar situations, we require that this signature scheme can only sign messages $\mathbf{v} \in \mathbb{Z}_+$ using the private key.

5.2 Correctness

Below, we use a theorem to demonstrate that the above signature system satisfies correctness with overwhelming probability.

Theorem 8. *The above linearly semigroup-homomorphic signature scheme satisfies correctness with overwhelming probability.*

Proof. (1) If $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_1, ..., \boldsymbol{\sigma}_{|\mathbf{y}|}] \longleftarrow \mathbf{Sign}(sk, \tau, \mathbf{y})$, where $\mathbf{y} = \mathbf{y}_1\mathbf{y}_2 \cdots \mathbf{y}_{|\mathbf{y}|} \in \mathcal{M}$, then

$$\boldsymbol{\sigma}_i \longleftarrow \mathbf{SamplePre}(B^\tau, T_{B^\tau}, \boldsymbol{\beta}_i, V),$$

where $\boldsymbol{\beta}_i = \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i)_j \cdot \boldsymbol{\alpha}_j$, for $i = 1, ..., |\mathbf{y}|$.

From Lemma 2 and Theorem 2, it follows that $\| \widetilde{T}_{B^\tau} \| \leq \| \widetilde{HT}_A \| = \| \widetilde{T}_A \| \leq \mathcal{O}(\sqrt{h \log q})$ with overwhelming probability. Since $V = k\sqrt{2n \log q} \cdot \log n$, it follows that $\frac{V}{\|\widetilde{T}_{B^\tau}\|} \geq k\sqrt{\frac{2n}{h}} \cdot \log n \geq \sqrt{\log n}$, which means $V \geq \| \widetilde{T}_{B^\tau} \| \cdot \omega(\sqrt{\log n})$.

Therefore, by Lemma 6, we have $\| \boldsymbol{\sigma}_i \| \leq V \cdot \sqrt{n} \leq V \cdot \sqrt{kn}$ with overwhelming probability. Thus

$$\| \boldsymbol{\sigma} \| = \max_{1 \leq i \leq |\mathbf{y}|} \| \boldsymbol{\sigma}_i \| \leq V \cdot \sqrt{kn}.$$

Furthermore, since $\boldsymbol{\sigma}_i \longleftarrow \mathbf{SamplePre}(B^\tau, T_{B^\tau}, \boldsymbol{\beta}_i, V)$, it follows that $B^\tau \cdot \boldsymbol{\sigma}_i (\text{mod} q) = \boldsymbol{\beta}_i$. Therefore, $B^\tau \cdot \boldsymbol{\sigma} (\text{mod} q) = Y$, where $Y = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{|\mathbf{y}|}]$.

Therefore

$$\mathbf{Verify}(pk, \tau, \mathbf{y}, \boldsymbol{\sigma}) = 1.$$

(2) If $\boldsymbol{\sigma} \longleftarrow \mathbf{Combine}(pk, \tau, \{(c_i, \boldsymbol{\sigma}_i)_{i=1}^{l}\})$, where $c_i \in \mathbb{Z}_p$, $\boldsymbol{\sigma}_i \longleftarrow \mathbf{Sign}(sk, \tau, \mathbf{v}_i)$, for $1 \leq i \leq l$. According to the definition of the algorithm **Combine**,

$$\boldsymbol{\sigma} = c_1 \cdot \boldsymbol{\sigma}_1 \| \cdots \| c_l \cdot \boldsymbol{\sigma}_l = [c_1 \cdot \boldsymbol{\sigma}_1, \cdots, c_l \cdot \boldsymbol{\sigma}_l] = [\underbrace{\boldsymbol{\sigma}_1, \cdots, \boldsymbol{\sigma}_1}_{c_1}, \cdots, \underbrace{\boldsymbol{\sigma}_l, \cdots, \boldsymbol{\sigma}_l}_{c_l}].$$

It can be seen from 1) that $\parallel \boldsymbol{\sigma}_i \parallel \leq V \cdot \sqrt{kn}$ with overwhelming probability; thus

$$\parallel \boldsymbol{\sigma} \parallel \leq V \cdot \sqrt{kn}.$$

Let $Y_i = [\boldsymbol{\beta}^i_1, ..., \boldsymbol{\beta}^i_{|\mathbf{v}_i|}]$, then $B^\tau \cdot \boldsymbol{\sigma}_i(\text{mod} q) = Y_i$, for $i = 1, ..., l$. Therefore,

$$B^\tau \cdot \boldsymbol{\sigma}(\text{mod} q) = B^\tau \cdot [\underbrace{\boldsymbol{\sigma}_1, \cdots, \boldsymbol{\sigma}_1}_{c_1}, \cdots, \underbrace{\boldsymbol{\sigma}_l, \cdots, \boldsymbol{\sigma}_l}_{c_l}] = [\underbrace{Y_1, \cdots, Y_1}_{c_1}, \cdots, \underbrace{Y_l, \cdots, Y_l}_{c_l}].$$

Let

$$\mathbf{y} = c_1 \cdot \mathbf{v}_1 \parallel \cdots \parallel c_l \cdot \mathbf{v}_l = \underbrace{\mathbf{v}_1 \parallel \cdots \parallel \mathbf{v}_1}_{c_1} \parallel \cdots \parallel \underbrace{\mathbf{v}_l \parallel \cdots \parallel \mathbf{v}_l}_{c_l}.$$

Let $\boldsymbol{\beta}_i = \sum_{j=1}^k \mathbf{h}(\mathbf{y}_i)_j \cdot \boldsymbol{\alpha}_j$, for $i = 1, \ldots, |\mathbf{y}|$, then

$$[\underbrace{Y_1, \ldots, Y_1}_{c_1}] = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{c_1|\mathbf{v}|_1}], \quad \ldots, \quad [\underbrace{Y_l, \ldots, Y_l}_{c_l}] = [\beta_{|\mathbf{y}| - \sum_{i=1}^{l-1} c_i |\mathbf{v}|_i}, \ldots, \beta_{\mathbf{y}}].$$

Let, $Y = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_{|\mathbf{y}|}]$, thus

$$Y = [\underbrace{Y_1, \cdots, Y_1}_{c_1}, \cdots, \underbrace{Y_l, \cdots, Y_l}_{c_l}].$$

Therefore,

$$B^\tau \cdot \boldsymbol{\sigma}(\text{mod} q) = Y.$$

It follows that

$$\textbf{Verify}(pk, \tau, c_1\mathbf{v}_1 \parallel \cdots \parallel c_l\mathbf{v}_l, \textbf{Combine}(pk, \tau, \{(c_i, \boldsymbol{\sigma}_i)_{i=1}^l\})) = 1.$$

□

5.3 Security

Below, we prove that our signature scheme is existentially unforgeable under adaptive chosen-message attacks with a fixed message range (EUF-CMA-FMR).

Theorem 9. *Let $\mathcal{LHS}$ be the linearly semigroup-homomorphic signature described above. Suppose $q = \text{poly}(n)$, $k = \text{poly}(n)$ and $q \geq (kn)^2$, $h = \lfloor \frac{n}{6 \log q} \rfloor$, $V = k\sqrt{2n \log q} \cdot \log n$, $\beta = 2V \cdot \sqrt{kn}$. If the $\boldsymbol{SIS}_{q,\beta}$ problem is hard, then $\mathcal{LSH}$ is is existentially unforgeable under the adaptive chosen-message attack with a fixed message range (EUF-CMA-FMR).*

More specifically, Suppose the adversary $\mathcal{A}$ is allowed to make at most q_s chosen-message queries on message subset $\mathbb{Z}_+ (\subseteq \mathcal{M} = (\mathbb{Z}_+)^*)$ and can break the above linearly *semigroup-homomorphic signature scheme $\mathcal{LSH}$ with advantage ϵ within time t. Then, the simulator $\mathcal{C}$ can use the adversary's ability to construct an algorithm $\mathcal{B}$ that outputs a solution to the* $\mathrm{SIS}_{q,2V\cdot\sqrt{kn}}$ *problem with advantage $\epsilon - \mathrm{negl}(n)$ within time $t + \mathcal{O}(q_s T_{\mathcal{LSH}} + T_{\mathrm{SampleDom}})$, where $T_{\mathcal{LSH}}$ represents the time required for the simulator to generate a signature, and $T_{\mathrm{SampleDom}}$ represents the time for running the sampling algorithm **SampleDom** once.*

Proof. Assume that the adversary $\mathcal{A}$ breaks the signature scheme $\mathcal{LHS}$ with advantage ϵ within time t. We will now prove that a simulator $\mathcal{C}$ can be constructed, which utilizes the adversary $\mathcal{A}$'s ability to forge signatures to construct a algorithm $\mathcal{B}$ that can output a solution to the $\mathbf{SIS}_{q,\beta}$ problem.

Now the simulator $\mathcal{C}$ generates the public key in the following way:

(1) Let the public parameters

$$pp = \left\{ \begin{array}{l} k = \mathrm{poly}(n), \\ q = \mathrm{poly}(n) > (kn)^2, \\ h = \left\lfloor \frac{n}{6\log q} \right\rfloor, \\ V = k\sqrt{2n\log q} \cdot \log n. \end{array} \right\}.$$

(2) Randomly and uniformly select a matrix $A \in \mathbb{Z}_q^{h\times n}$. We regard A as an instance of the **SIS** problem.

(3) Sample k vectors:

$$\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_k \leftarrow \mathbf{SampleDome}(1^n, s),$$

where $s = V/\sqrt{k} \geq \omega(\sqrt{\log n})$. Let $\boldsymbol{\alpha}_i = A \cdot \boldsymbol{\gamma}_i (\mathrm{mod} q) \in \mathbb{Z}_q^h$, for $i = 1, ..., k$.

(4) Select a secure collision-resistant hash function $\mathbf{h} : \mathbb{Z}_+ = \{0,1\}^* \rightarrow \{0,1\}^k$

(5) Send $pk = (A, \mathbf{h}, \boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k)$ to the adversary $\mathcal{A}$.

After receiving the public key pk, the adversary $\mathcal{A}$ selects a sequence of subsets $V_1, ..., V_{q_s}$ as signature queries, where $V_i = span_{\|}\{\mathbf{v}_{i1}, ..., \mathbf{v}_{il}\}$, and $(\mathbf{v}_{i1}, ..., \mathbf{v}_{il}) \in (\mathbb{Z}_+)^k$, $1 \leq i \leq q_s$, where l represents the maximum number of times of the homomorphic operation.. For each i $(i = 1, ..., q_s)$, the simulator $\mathcal{C}$ chooses τ_i uniformly from $\{0,1\}^n$ and gives the tag τ_i to $\mathcal{A}$. The steps for the simulator to return the signature queries to the adversary $\mathcal{A}$ are as follows:

(1) Compute $H_\tau = \mathrm{diag}\{2\tau_1 - 1, ..., 2\tau_n - 1\}$, where $\tau = (\tau_1, ..., \tau_n)$.

(2) Compute $T_{B^{\tau_i}} \longleftarrow \mathbf{NewBasis}(A, H_{\tau_i}, T_A)$. $T_{B^{\tau_i}}$ is a basis of $\Lambda_q^\perp(B^{\tau_i})$, where $B^{\tau_i} = AH_{\tau_i}^\top$.

(3) Compute

$$\mathbf{t}_{ij} = H \cdot [\mathbf{h}(\mathbf{v}_{ij})_1 \cdot \boldsymbol{\gamma}_1 + ... + \mathbf{h}(\mathbf{v}_{ij})_k \cdot \boldsymbol{\gamma}_k],$$

where $\mathbf{h}(\mathbf{v}_{ij}) = (\mathbf{h}(\mathbf{v}_{ij})_1, ..., \mathbf{h}(\mathbf{v}_{ij})_k)$.

(4) Let $\boldsymbol{\sigma}_{ij} = \mathbf{t}_{ij}$, for $j = 1, ..., l$.

The simulator $\mathcal{C}$ output the signature $\boldsymbol{\sigma}_{ij}$, where $i = 1, ..., q_s$ and $j = 1, ..., l$, and sends it to $\mathcal{A}$.

Next, we prove that the output distribution of the simulator is indistinguishable (within negligible statistical distance) from the output distribution in the real signature scheme.

By Lemma 8, we know that the distribution of $H \cdot [\mathbf{h}(\mathbf{v}_{ij})_1 \cdot \boldsymbol{\gamma}_1 + ... + \mathbf{h}(\mathbf{v}_{ij})_k \cdot \boldsymbol{\gamma}_k]$ is the same as that of $\mathbf{h}(\mathbf{v}_{ij})_1 \cdot \boldsymbol{\gamma}_1 + ... + \mathbf{h}(\mathbf{v}_{ij})_k \cdot \boldsymbol{\gamma}_k$. Therefore, through a discussion entirely similar to that in Theorem 7, the output of the simulator and the output of the real scheme are statistically indistinguishable.

Finally, the adversary $\mathcal{A}$ generates a valid forgery (τ^*, $mathbfy^*, \boldsymbol{\sigma}^*$), such that

$$\textbf{Verify}(pk, \tau^*, \mathbf{y}^*, \boldsymbol{\sigma}^*) = 1,$$

where $\tau^* \in \{0,1\}^n$, $\mathbf{y}^* = \mathbf{y}_1^* \mathbf{y}_2^* \cdots \mathbf{y}_{|\mathbf{y}^*|} \in \mathcal{M}$, $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}_1^* \boldsymbol{\sigma}_2^* \cdots \boldsymbol{\sigma}_{|\mathbf{y}|}^* = [\boldsymbol{\sigma}_1^*, ..., \boldsymbol{\sigma}_{|\mathbf{y}|}^*] \in \Sigma$.

Whether it is a **I-type forgery** or **II-type forgery**, the following equation holds:

$$B^{\tau^*} \cdot \boldsymbol{\sigma}^* (\text{mod} q) = Y^*,$$

where $Y^* = [\boldsymbol{\beta}_1^*, ..., \boldsymbol{\beta}_{|\mathbf{y}|}^*]$, $\boldsymbol{\beta}_i^* = \sum_{j=1}^k \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\alpha}_j$, $1 \leq i \leq |\mathbf{y}^*|$. Noting that $\boldsymbol{\alpha}_j = A \cdot \boldsymbol{\gamma}_j (\text{mod} q)$, we have:

$$\begin{aligned} B^{\tau^*} \cdot [\boldsymbol{\sigma}_1^*, \ldots, \boldsymbol{\sigma}_{|\mathbf{y}|}^*] (\text{mod}\, q) &= A \cdot H_{\tau^*}^\top \cdot [\boldsymbol{\sigma}_1^*, \ldots, \boldsymbol{\sigma}_{|\mathbf{y}|}^*] \,(\text{mod}\, q) \\ &= [A \cdot H_{\tau^*}^\top \cdot \boldsymbol{\sigma}_1^*, \ldots, A \cdot H_{\tau^*}^\top \cdot \boldsymbol{\sigma}_{|\mathbf{y}|}^*] \,(\text{mod}\, q) \\ &= [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{|\mathbf{y}^*|}] \\ &= [\sum_{j=1}^k \mathbf{h}(\mathbf{y}_1^*)_j \cdot \boldsymbol{\alpha}_j, \ldots, \sum_{j=1}^k \mathbf{h}(\mathbf{y}_{|\mathbf{y}^*|}^*)_j \cdot \boldsymbol{\alpha}_j] \\ &= [\sum_{j=1}^k \mathbf{h}(\mathbf{y}_1^*)_j \cdot A \cdot \boldsymbol{\gamma}_j, \ldots, \sum_{j=1}^k \mathbf{h}(\mathbf{y}_{|\mathbf{y}^*|}^*)_j \cdot A \cdot \boldsymbol{\gamma}_j] \end{aligned}$$

Arbitrarily take an index $i \in \{1, ..., |\mathbf{y}^*|\}$, then

$$A \cdot H_{\tau^*}^\top \cdot \boldsymbol{\sigma}_i^* (\text{mod} q) = \sum_{j=1}^k \mathbf{h}(\mathbf{y}_i^*)_j \cdot A \cdot \boldsymbol{\gamma}_j.$$

Thus,

$$A \cdot (H_{\tau^*}^\top \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^k \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j)(\text{mod} q) = 0$$

From verification condition 1), we know

$$\parallel H_{\tau^*}^\top \cdot \boldsymbol{\sigma}_i^* \parallel = \parallel \boldsymbol{\sigma}_i^* \parallel \leq V\sqrt{kn}.$$

By Lemma 6, $\| \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j \| \leq k \cdot \frac{V}{\sqrt{k}} \cdot \sqrt{n} = V \cdot \sqrt{kn}$ holds with overwhelming probability, so

$$\| H_{\tau^*}^{\top} \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j \| \leq 2V \cdot \sqrt{kn}.$$

If $H_{\tau^*}^{\top} \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j \neq 0$, then the simulator outputs $H_{\tau^*}^{\top} \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j$ as a solution to the $\mathbf{SIS}_{q,\beta}$, where $\beta = 2V \cdot \sqrt{kn}$.

Let the process of outputting $H_{\tau^*}^{\top} \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j$ be denoted as algorithm $\mathcal{B}$.

Finally, we discuss the total probability. Similar to the proof process of Theorem 7, we can still obtain that the probability of $H_{\tau^*}^{\top} \cdot \boldsymbol{\sigma}_i^* - \sum_{j=1}^{k} \mathbf{h}(\mathbf{y}_i^*)_j \cdot \boldsymbol{\gamma}_j = 0$ is $\text{negl}(n)$. Therefore, the simulator can construct an algorithm $\mathcal{B}$ by using the adversary $\mathcal{A}$'s ability that, within time $t + \mathcal{O}(q_s T_{\mathcal{LSH}} + T_{\mathbf{SampleDom}})$ outputs a solution to the $\mathbf{SIS}_{q,\beta}$ problem with advantage $\epsilon - \text{negl}(n)$. □

5.4 Privacy

Since our signature scheme only uses the private key to sign messages $\mathbf{v} \in \mathbb{Z}_+$, we will prove that the above signature scheme is weak context hiding in this case.

Theorem 10. *If the adversary only queries on the message subset $\mathcal{M}' = \mathbb{Z}_+$, then the above linearly semigroup-homomorphic signature scheme is weakly context hiding.*

Proof. The challenger runs $\mathbf{Setup}(1^n, pp)$ to generate a public-private key pair (pk, sk) and sends it to the adversary. Upon receiving the key pair, the adversary outputs $(V_0, V_1, f_1, \ldots, f_s)$, where $V_0 = \text{span}_{\|}\{\mathbf{v}_{01}, \ldots, \mathbf{v}_{0k}\}$, $V_1 = \text{span}_{\|}\{\mathbf{v}_{11}, \ldots, \mathbf{v}_{1k}\}$, and $(\mathbf{v}_{i1}, ..., \mathbf{v}_{il}) \in (\mathbb{Z}_+)^k$ for $b = 0, 1$. Additionally, it holds that $f_i(\mathbf{v}_{01}, \ldots, \mathbf{v}_{0k}) = f_i(\mathbf{v}_{11}, \ldots, \mathbf{v}_{1k})$ for $i = 1, 2, \ldots, s$.

In response, the challenger randomly selects a tag $\tau \in \{0,1\}^n$ and a random bit $b \in \{0,1\}$. Let

$$\boldsymbol{\sigma}_{bi} \leftarrow \mathbf{Sign}(sk, \tau, \mathbf{v}_{bi}), i = 1, ..., k.$$

Then, $\boldsymbol{\sigma}_{bi}$ is statistically close to $\mathcal{D}_{\Lambda + \boldsymbol{\beta}_{bi}, V}$, where $\Lambda = \Lambda_q^{\perp}(B^{\tau})$, $\boldsymbol{\beta}_{bi} = \sum_{j=1}^{k} \mathbf{h}(\mathbf{v}_{bi})_j \cdot \boldsymbol{\alpha}_j$.

Let $\langle f \rangle = (c_1, \ldots, c_k)$. According to the definition of the algorithm **Combine**, we have:

$$\mathbf{Combine}(pk, \tau, \{(c_i, \boldsymbol{\sigma}_{bi})\}_{i=1}^{k}) \to \boldsymbol{\sigma}_b = c_1 \boldsymbol{\sigma}_{b1} \parallel \cdots \parallel c_k \boldsymbol{\sigma}_{bk} = [c_1 \boldsymbol{\sigma}_{b1}, \ldots, c_k \boldsymbol{\sigma}_{bk}]$$

$$= [\underbrace{\boldsymbol{\sigma}_{b1}, \ldots, \boldsymbol{\sigma}_{b1}}_{c_1}, \ldots, \underbrace{\boldsymbol{\sigma}_{bk}, \ldots, \boldsymbol{\sigma}_{bk}}_{c_k}]$$

Note that, according to the definition of the signing algorithm, each column of the above signature $\boldsymbol{\sigma}_b$ is statistically close to $\mathcal{D}_{\Lambda+\boldsymbol{\beta}_{bl},V}$, where $\boldsymbol{\beta}_{bl} = \sum_{j=1}^{k} \mathbf{h}(f(\mathbf{v}_{b1}, ..., \mathbf{v}_{bk})_l)_j \cdot \boldsymbol{\alpha}_j$, $l = 1, ..., |f(\mathbf{v}_{b1}, ..., \mathbf{v}_{bk})|$.

Therefore, the distribution of $\boldsymbol{\sigma}_b$ depends only on the parameters: Λ, $\boldsymbol{\alpha}_1, ..., \boldsymbol{\alpha}_k$, V, f, $f(\mathbf{v}_{b1}, ..., \mathbf{v}_{bk})$.

Moreover, since $f_i(\mathbf{v}_{01}, ..., \mathbf{v}_{0k}) = f_i(\mathbf{v}_{11}, ..., \mathbf{v}_{1k})$, $i = 1, 2, ..., s$, for any f_i, $\boldsymbol{\sigma}_0$ and $\boldsymbol{\sigma}_1$ they are statistically close to the same distribution.

Consequently, even an unbounded adversary cannot win the privacy game with non-negligible advantage.

□

6 Conclusions and Open Problems

In 2002, at the Cryptographers' Track of the RSA Conference, Johnson et al.[3] put forward an open problem: how to construct a secure homomorphic signature on a semigroup instead of on a group. This paper presents the first homomorphic signature on a semigroup. To prove the security of this scheme, we define a new security model, namely EUF-CMA-FMR (existential unforgeability under adaptive chosen-message attacks within a fixed message range). We prove that the scheme is existentially unforgeable under adaptive chosen-message attacks within a fixed message range (EUF-CMA-FMR), and that it is tightly secure.

Second, we define linearly semigroup-homomorphic signatures and the corresponding security model, and extend the semigroup-homomorphic signature constructed in Section 4 to a linearly semigroup-homomorphic signature. We prove the correctness, unforgeability, and privacy of this signature scheme.

Finally, there still remain quite a number of unsolved problems here. (1) Regarding the signature length, since the signature length of our signature scheme increases linearly with the increase of the message length, is it possible to construct a semigroup-homomorphic signature with a fixed signature length? (2) Regarding the message space, is it possible to construct a semigroup-homomorphic signature on other semigroups that do not have inverses? For example, can we construct a set-homomorphic scheme that only permits the union operation? This question is mentioned in [3]. (3) Regarding security, is it possible to construct a semigroup homomorphic signature that satisfies the standard EUF-CMA? (4) Regarding applications, is it possible to construct other linearly semigroup-homomorphic signatures, or homomorphic signatures on a semigroup that support two different operations? Furthermore, could we propose a fully homomorphic signature on a semigroup, similar to a fully homomorphic signature scheme?

References

[1] F. Zhao, T. Kalker, M. Médard, and K. J. Han, "Signatures for content distribution with network coding," *2007 IEEE international symposium on information theory*, pp. 556–560, 2007.

[2] L. R. Rivest., “Two signature schemes.” *http://people.csail.mit.edu/rivest/pubs.html.*

[3] R. Johnson, D. Molnar, D. Song, and D. Wagner, “Homomorphic signature schemes,” *Cryptographers track at the RSA conference*, pp. 244–262, 2002.

[4] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, “Secure Network Coding Over the Integers,” *In Public Key CryptographyPKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography*(2010), pp. 142-160.

[5] D. Boneh, D. Freeman, J. Katz, and B. Waters, “Signing a linear subspace: Signature schemes for network coding,” *Public Key Cryptography–PKC 2009: 12th International Conference on Practice and Theory in Public Key Cryptography, Irvine, CA, USA, March 18-20, 2009. Proceedings 12*, pp. 68–87, 2009.

[6] N. Attrapadung, B. Libert, and T. Peters, “Computing on authenticated data: New privacy definitions and constructions,” *Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings 18*, pp. 367–385, 2012.

[7] D. M. Freeman, “Improved security for linearly homomorphic signatures: A generic framework,” *Public Key Cryptography–PKC 2012: 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings 15*, pp. 697–714, 2012.

[8] W. Chen, H. Lei, and K. Qi, “Lattice-based linearly homomorphic signatures in the standard model,” *Theoretical Computer Science*, vol. 634, pp. 47–54, 2016.

[9] C. J. Lin, R. Xue, S. J. Yang, X. Huang, and S. Li, “Linearly homomorphic signatures from lattices,” *The Computer Journal*, vol.63, no.12, pp. 1871–1885, 2020.

[10] W. Chen and Z. Huang, “Towards tightly secure short linearly homomorphic signatures,” *Theoretical Computer Science*, vol. 1014, p. 114758, 2024.

[11] D. Boneh and D. M. Freeman, “Homomorphic signatures for polynomial functions,” *Advances in Cryptology–EUROCRYPT 2011: 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings 30*, pp. 149–168, 2011.

[12] R. Hiromasa, Y. Manabe, and T. Okamoto, “Homomorphic signatures for polynomial functions with shorter signatures.,” *In The 30th symposium on cryptography and information security*, 2013.

[13] S. Arita and S. Kozaki, "A homomorphic signature scheme for quadratic polynomials," *In 2017 IEEE International Conference on Smart Computing (2017)*, pp. 1–6.

[14] D. Catalano, D. Fiore, and B. Warinschi, "Homomorphic signatures with efficient verification for polynomial functions," *In Annual Cryptology Conference* (2014), pp. 371–389.

[15] C. Gentry, "Fully homomorphic encryption using ideal lattices," *In Proceedings of the forty-first annual ACM symposium on Theory of computing*(2009), pp. 167–178.

[16] S. Gorbunov, V. Vaikuntanathan, and D. Wichs, "Leveled fully homomorphic signatures from standard lattices," *In Proceedings of the forty-seventh annual ACM symposium on Theory of computing*(2015), pp. 469–477.

[17] X. Boyen, X. Fan, and E. Shi, "Adaptively secure fully homomorphic signatures based on lattices," *Cryptology ePrint Archive*, 2014. https://eprint.iacr.org/2014/916.pdf.

[18] F. Wang, K. Wang, B. Li, and Y.Gao, "Leveled strongly-unforgeable identity-based fully homomorphic signatures," *In: Information SecurityISC*(2015), pp. 42–60.

[19] Y. Wang and M. Wang, "A new fully homomorphic signatures from standard lattices," *In International Conference on Wireless Algorithms, Systems, and Applications*(2020), pp. 494–506.

[20] F. Luo, F. Wang, K.Wang, and K.Chen, "A more efficient leveled strongly-unforgeable fully homomorphic signature scheme," *Information Sciences*, vol.480, pp. 70–89, 2019.

[21] R. Gayand B. Ursu, "On instantiating unleveled fully-homomorphic signatures from falsifiable assumptions," *In IACR International Conference on Public-Key Cryptography*(2024), pp. 74–104.

[22] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*. vol.48, pp. 535–553, 2010.

[23] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," *In Proceedings of the fortieth annual ACM symposium on Theory of computing*(2008), pp. 197–206.

[24] H. Guo, K. Tian, F. Liu and Z. Zheng, "Linearly Homomorphic Signature with Tight Security on Lattice," *arxiv preprint arxiv:2412.01641*, 2024.

[25] Z. Zheng, "Modern cryptography volume 1: A classical introduction to informational and mathematical principle," *Springer Nature*, 2022.

[26] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model," *In Henri Gilbert, editor, EUROCRYPT*(2010), vol.6110 pp. 553-572.

[27] Z.Zheng, F.Liu, K.Tian, "Modern cryptography: A classical introduction to informational and mathematical principle (volume 2)," *Springer Nature*, 2023.

[28] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," *In: STOC 1996, ACM*(1996), pp. 99108.

[29] D. Micciancio and S. Goldwasser, "Complexity of lattice problems: a cryptographic perspective," *Springer Science & Business Media*, vol. 671, 2002.

[30] M. Ajtai, C. Peikert, and V. Vaikuntanathan, "Generating hard instances of the short basis problem," *In Automata, Languages and Programming: 26th International Colloquium, ICALP99 Prague, Czech Republic*(1999), pp. 11–9.

[31] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on gaussian measures," *SIAM Journal on Computing*, vol.37, no.1, pp. 267–302, 2007.

[32] D. Catalano, D. Fiore, and B. Warinschi, "Homomorphic signatures with efficient verification for polynomial functions," *In Annual Cryptology Conference* (2014), pp. 371–389.

[33] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical ibe," *Advances in Cryptology–CRYPTO 2010: 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings 30*, pp. 98–115, 2010.