SCIS 2013 The 30th Symposium on
Cryptography and Information Security
kyoto, Japan, Jan. 22 - 25, 2013
The Institute of Electronics,
Information and Communication Engineers

Homomorphic Signatures for Polynomial Functions with Shorter Signatures

Ryo Hiromasa * Yoshifumi Manabe † Tatsuaki Okamoto †

Abstract— We present homomorphic signatures for polynomial functions with shorter signatures than the ones of Boneh and Freeman. In the signing algorithm, instead of using the preimage sampling algorithm of Gentry, Peikert, and Vaikuntanathan, we use the algorithm of Micciancio and Peikert, which is more efficient (the algorithm can sample smaller preimages) than the one of Gentry et al. Consequently, the length of the signatures in the proposed scheme is $\tilde{\mathrm{O}}(n^3)$, while the scheme of Boneh et al. generates signatures of length $\tilde{\mathrm{O}}(n^{4.5})$.

Keywords: fully homomorphic encrytion, polynomially homomorphic signature.

1 Introduction

The polynomially homomorphic signature scheme proposed by Boneh and Freeman [BF11] is a kind of digital signature scheme. In this scheme, we can evaluate polynomials on *signed data*, while fully homomorphic encryption allows us to compute arbitrary operations on *encrypted data*. More precisely, we consider the following case. Alice has the set of numerical data $\{m_1, ..., m_k\}$, e.g., grades of k students in a course. She signs each triple ("*grades*", m_i, i) and obtains the set of k signatures $\{\sigma_1, ..., \sigma_k\}$. The signature set is stored on a remote server. In addition, Alice asks the remote server to compute a function f, e.g., mean, standard deviation, and other data mining algorithms. The server evaluates f on $\{\sigma_1, ..., \sigma_k\}$, and therefore the server obtains the signature σ. The signature σ authenticates the triple ("*grades*", $m, \langle f \rangle$), where $\langle f \rangle$ is an encoding of the function f and $m = f(m_1, ..., m_k)$. Then, the server publishes (m, σ). By verifying that σ is the signature on the triple, anyone can check that the server correctly computes f on the data set $\{m_1, ..., m_k\}$.

The scheme of [BF11] builds on the lattice-based hash and sign signature scheme, which was described in [GPV08]. Such a scheme, to sign messages, use the algorithm that samples a preimage of a special trapdoor oneway function, called Preimage Sampleable Functions (PSFs). If we have the trapdoor, PSFs allow us to sample preimages along some distribution. In [GPV08], Gentry et al. first proposed an algorithm for PSFs, which is also used in the scheme of [BF11]. Trapdoor generators for the algorithm were proposed in [Ajt99, AP11]. As a recent work, Micciancio and Peikert presented another trapdoor generator and sampling algorithm for PSFs in [MP12].

1.1 Our Results

We construct homomorphic signatures for polynomial functions with shorter signatures than the ones of [BF11]. Concretely, we replace the preimage sampling algorithm of [GPV08], used for signing, with the algorithm of [MP12]. As a result, the length of fresh signatures is reduced from $\tilde{\mathrm{O}}(n^{4.5})$ to $\tilde{\mathrm{O}}(n^3)$.

In this paper, we first introduce basic matters used in the paper. Next, we describe the homomorphic signatures for polynomial functions of [BF11]. Finally, we present an improvement to the signature scheme.

2 Preliminaries

2.1 Notation

We denote the set of integers by $\mathbb{Z}$ and the set of real numbers by $\mathbb{R}$. $\mathbb{Z}[x]$ denotes the set of polynomials whose coefficients are in $\mathbb{Z}$. We let $\mathbb{F}_p$ be the finite field that has p elements. $\langle f(x) \rangle$ is used to denote the group generated by $f(x)$. When we let $\mathbb{G}$ be some group and $\mathcal{P}$ be some probability distribution, we use $a \xleftarrow{U} \mathbb{G}$ to denote that a is chosen from the group $\mathbb{G}$ uniformly at random and use $b \xleftarrow{R} \mathcal{P}$ to denote that b is chosen along the probability distribution $\mathcal{P}$.

We assume that vectors are in column form and are written by using bold lower-case letters, e.g., $\mathbf{x}$. The ith element of a vector is denoted by x_i. We let the length of vectors be the l_2 (Eclidean) norm of the vectors and denote it by $\| \mathbf{x} \|$. The inner product between two vectors is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle$. Matrices are written by using bold capital letters, e.g., $\mathbf{X}$, and the ith column vector of a matrix is denoted by $\mathbf{x}_i$. We write $\mathrm{span}(\mathbf{X})$ to denote the linear space spanned by $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, i.e., $\mathrm{span}(\mathbf{X}) = \{\mathbf{Xr} : \mathbf{r} \in \mathbb{R}^n\}$. When we refer to the $n \times n$ identity matrix, we denote it by $\mathbf{I}_n$. We let $\| \mathbf{X} \|$ be the maximum length of its vectors. We denote the *Gram-Schmidt orthogonalization* of the vectors

* Kyoto University.
† NTT.

$\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ by $\tilde{\mathbf{X}} = (\tilde{\mathbf{x_1}}, ..., \tilde{\mathbf{x_n}})$, where for all i, $\tilde{\mathbf{x}_i}$ is the vector orthogonal to $\{\tilde{\mathbf{x_1}}, ..., \mathbf{x}_{\tilde{i-1}}, \mathbf{x}_{\tilde{i+1}}, ..., \tilde{\mathbf{x_n}}\}$. We let $\| \tilde{\mathbf{X}} \|$ be the norm of the matrix after the Gram-Schmidt orthogonalization.

2.2 Linear Algebra

For any square real matrix $\mathbf{X}$, we let $\mathbf{X}^+$ be the *Moore-Penrose pseudoinverse* matrix, which is the unique matrix such that $(\mathbf{X}\mathbf{X}^+)\mathbf{X} = \mathbf{X}$ and $\mathbf{X}^+(\mathbf{X}\mathbf{X}^+) = \mathbf{X}^+$. If the matrix $\mathbf{X}$ is invertible, then $\mathbf{X}^+ = \mathbf{X}^{-1}$.

We say that a symmetric matrix $\mathbf{S}$ is *positive definite* (or *positive semi-definite*), written $\mathbf{S} > \mathbf{0}$ ($\mathbf{S} \geq \mathbf{0}$), if for all nonzero $\mathbf{x} \in \mathbb{R}^n$ $\mathbf{x}^t\mathbf{S}\mathbf{x} > 0$ ($\mathbf{x}^t\mathbf{S}\mathbf{x} \geq 0$). A partial order on symmetric matrices is defined by the positive (semi-)definiteness: we denote $\mathbf{S}_1 > \mathbf{S}_2$ if $(\mathbf{S}_1 - \mathbf{S}_2) > \mathbf{0}$, and similarly for $\mathbf{S}_1 \geq \mathbf{S}_2$.

For any matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, the symmetric matrix $\mathbf{S} = \mathbf{B}\mathbf{B}^t$ is positive semi-definite, because for any nonzero real vector $\mathbf{x} \in \mathbb{R}^n$ we have $\mathbf{x}^t\mathbf{S}\mathbf{x} = \| \mathbf{B}^t\mathbf{x} \|^2 \geq 0$, where the inequality is always strict if and only if $\mathbf{B}$ is invertible. We say that the matrix $\mathbf{B}$ is a *square root* of $\mathbf{S} > \mathbf{0}$, written $\mathbf{B} = \sqrt{\mathbf{S}}$ if $\mathbf{B}\mathbf{B}^t = \mathbf{S}$. Every $\mathbf{S} \geq \mathbf{0}$ has a square root, which we can compute efficiently via the Cholesky decomposition.

For any matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, the *singular value decomposition* of $\mathbf{B}$ is the fraction as $\mathbf{B} = \mathbf{Q}\mathbf{D}\mathbf{P}^t$, where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{P} \in \mathbb{R}^{m\times m}$ are orthogonal matrices, and $\mathbf{D} \in \mathbb{R}^{n\times m}$ is a diagonal matrix that has nonnegative entries $s_i(\mathbf{B})$, called the *singular value* of $\mathbf{B}$, on the diagonal in non-increasing order.

2.3 Lattices

$\mathbf{B} = \{\mathbf{b}_1, ..., \mathbf{b}_n\} \subset \mathbb{R}^n$ is the set of n linearly independent vectors. An n dimensional *lattice* Λ generated by the basis $\mathbf{B}$ are defined as the linear combination of its vectors, i.e., $\Lambda = \mathcal{L}(\mathbf{B}) = \{\sum_i z_i\mathbf{b}_i : z_i \in \mathbb{Z}\}$. Using matrix notation, we can also define the lattice as $\Lambda = \{\mathbf{B}\mathbf{z} : \mathbf{z} \in \mathbb{Z}^n\}$. The determinant of the lattice $\Lambda = \mathcal{L}(\mathbf{B})$ is the absolute value of the determinant of its basis matrix $\det(\Lambda) = |\det(\mathbf{B})|$.

We often use a particular family of so-called *q-ary lattices* in many cryptographic applications. These are lattices Λ such that $q\mathbb{Z}^n \subseteq \Lambda \subseteq \mathbb{Z}^n$. Any integer lattice Λ can be a q-ary lattice whenever q is the integer multiple of the determinant $\det(\Lambda)$. For some positive integer q, m, n, we let $\mathbf{A} \in \mathbb{Z}^{n\times m}$ be an arbitrary matrix and define two m dimensional q-ary lattices as:

$$\begin{aligned} \Lambda^\perp(\mathbf{A}) &:= \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{0} \mod q\}, \\ \Lambda(\mathbf{A}) &:= \{\mathbf{y} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}^n, \mathbf{y} = \mathbf{A}^T\mathbf{s} \mod q\}. \end{aligned}$$

From the definition, it is easy to see that these lattices are dual to each other, i.e., $\Lambda^\perp(\mathbf{A}) = q \cdot \Lambda(\mathbf{A})^*$ and $\Lambda(\mathbf{A}) = q \cdot \Lambda^\perp(\mathbf{A})^*$. For any $\mathbf{u}$ that has a solution to $\mathbf{A}\mathbf{x} = \mathbf{u} \mod q$, we define the shifted lattice as

$$\Lambda^\perp_{\mathbf{u}}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{u} \mod q\}.$$

The coset and the quotient group $\mathbb{Z}^m/\Lambda^\perp(\mathbf{A})$ are in bijective correspondence via the mapping $(\mathbf{x} + \Lambda^\perp(\mathbf{A})) \mapsto \mathbf{A}\mathbf{x} \mod q$. In other words, by computing $\mathbf{A}\mathbf{x} \mod q$, we can reduce $\mathbf{x}$ modulo the q-ary lattice $\Lambda^\perp(\mathbf{A})$.

Micciancio and Goldwasser describe that we can convert a full-rank set in a lattice into a basis of the lattice that has an equally low Gram-Schmidt norm.

Lemma 2.1. *([MG02, Lemma 7.1]) Let λ be an n dimensional lattice. There is a deterministic, polynomial-time algorithm that takes as input an arbitrary basis of λ and a full-rank set $\boldsymbol{S} = \{s_1, \ldots, s_n\}$ in λ, and outputs a basis $\boldsymbol{T}$ of λ such that*

$$\| \tilde{\boldsymbol{T}} \| \leq \| \tilde{\boldsymbol{S}} \|, \| \boldsymbol{T} \| \leq \frac{\sqrt{m}}{2} \| \boldsymbol{S} \| .$$

2.4 Polynomial Rings and Ideals

In the proposed scheme, we consider the polynomial ring $R = \mathbb{Z}[x]/\langle F(x)\rangle$ for some monic, irreducible polynomial $F(x)$. Let $F(x)$ be a degree n integer polynomial, then each element over R is a degree $n-1$ polynomial and corresponds to a vector of degree n. We can identify R with the integer lattice $\mathbb{Z}^n$ via the correspondence. Addition over the ring is done by adding two vectors component-wise. Multiplication is polynomial multiplication modulo $F(x)$. We let γ_F be the parameter bounding how much the multiplication increases the length of the product. The parameter is defined as:

$$\gamma_F := \sup_{u,v\in R} \frac{\| u \cdot v \|}{\| u \| \cdot \| v \|}.$$

We state the fact that says $\gamma_F = \sqrt{n}$ when setting $f(x)$ to be the cyclotomic polynomial $x^n + 1$ where n is a power of 2.

Lemma 2.2. *(implied by [Gen09a, Lemma 7.4.3]) Let n be a power of two, let $f(x) = x^n + 1$, and let $R = \mathbb{Z}[x]/\langle f(x)\rangle$. For any $s, t \in R$, we have*

$$\| s \cdot t \| < \sqrt{n} \cdot \| s \| \cdot \| t \| .$$

An ideal of $R = \mathbb{Z}[x]/\langle F(x)\rangle$ is the additive subgroup $\mathcal{I} \subset R$, which is closed under multiplication by elements of R. By the correspondence between R and $\mathbb{Z}^n$, we identify the ideal $\mathcal{I}$ with an ideal lattice that is a sublattice of $\mathbb{Z}^n$. An Ideal $\mathcal{I} \subset R$ is *prime* if for $x, y \in R$, $x \cdot y \in \mathcal{I}$ implies that either $x \in \mathcal{I}$ or $y \in \mathcal{I}$. If $\mathcal{I}$ is prime ideal, then $R/\mathcal{I} \cong \mathbb{F}_{p^e}$, where the prime integer p is called the *characteristic* of $\mathcal{I}$ and e is called the *degree* of $\mathcal{I}$. An ideal is *principal* if it can be written as $\alpha \cdot R$ for some $\alpha \in R$. The norm of an ideal $\mathcal{I}$ is the size of $R/\mathcal{I}$. If $\mathcal{I}$ is a prime ideal, then we can write $\mathcal{I} = p \cdot R + h(x) \cdot R$ for some prime p and some polynomial $h(x) \in R$ whose reduction modulo p is an irreducible factor of $f(x) \mod p$. In particular, if $\mathcal{I}$ is a degree one prime ideal, $h(x) = x - a$ for some integer a. Then by mapping $z(x) \in R$ to $z(a) \mod p$ we can easily compute the quotient map R to $\mathbb{F}_p$.

In the polynomially homomorphic signature scheme of [BF11], a principal degree-one prime ideal is generated by the algorithm of Smart and Vercautren [SV10].

Lemma 2.3. *(Section 3 in [SV10]) There is an algorithm* PrincGen *that takes as input a monic irreducible polynomial $F(x) \in \mathbb{Z}[x]$ of degree n and a parameter δ, and outputs a principal degree-one prime ideal $\mathcal{I} = (p, x-a) \in \mathbb{Z}[x]/\langle f(x)\rangle$ with its generator $g_{\mathcal{I}}$ such that $\| g_{\mathcal{I}} \|_2 < \delta\sqrt{n}$.*

2.5 Gaussians

For any real $s > 0$ and any vector $\mathbf{c} \in \mathbb{R}^n$, we define a Gaussian function on $\mathbb{R}^n$ with a center $\mathbf{c}$ and a parameter s as

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{s,\mathbf{c}}(\mathbf{x}) := \exp(\frac{-\pi \cdot \| \mathbf{x} - \mathbf{c} \|^2}{s^2}).$$

When the subscripts s and $\mathbf{c}$ are taken to be 1 and $\mathbf{0}$, respectively, we may omit them. We let $\rho_{s,\mathbf{c}}(\Lambda) := \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$ be the discrete integral of $\rho_{s,\mathbf{c}}(\mathbf{x})$ over the lattice Λ.

Using the covariance matrix, we can also define the Gaussian function on $\mathbb{R}^n$, which is

$$\rho(\mathbf{x}) := \exp(-\pi \cdot \| \mathbf{x} \|^2) = \exp(-\pi \cdot \langle \mathbf{x}, \mathbf{x} \rangle).$$

For a matrix $\mathbf{B}$, let $\mathbf{S} = \mathbf{B}\mathbf{B}^t \geq \mathbf{0}$. Applying a linear transformation given by $\mathbf{B}$, we obtain the (degenerate) Gaussian function.

$$\rho_{\mathbf{B}}(\mathbf{x}) := \begin{cases} \rho(\mathbf{B}^+\mathbf{x}) = \exp(-\pi \mathbf{x}^t \mathbf{S}^+ \mathbf{x}) & \text{if } \mathbf{x} \in \text{span}(\mathbf{B}), \\ 0 & \text{otherwise.} \end{cases}$$

Since $\rho_{\mathbf{B}}$ is only determined by $\mathbf{S}$, we denote it by $\rho_{\sqrt{\mathbf{S}}}$.

Normalizing ρ_s (or $\rho_{\sqrt{\mathbf{S}}}$) by its total measure, we obtain the probability distribution function of the Gaussian distribution. For any vector $\mathbf{c} \in \mathbb{R}^n$, real $s > 0$, and n dimensional lattice Λ, the *discrete Gaussian distribution over* Λ is defined as

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

We may also omit subscripts as well as the Gaussian function described above.

In [MR07], Micciancio and Regev propose the lattice parameter related to Gaussian measures on lattices, called the *smoothing parameter*.

Definition 2.1. *For an n dimensional lattice Λ, and a real $\epsilon > 0$, the smoothing parameter $\eta_\epsilon(\Lambda)$ is defined to be the smallest s satisfying $\rho_{1/s}(\Lambda^* \setminus \{\boldsymbol{0}\}) \leq \epsilon$.*

In [Pei08], Peikert shows the upper bound on the smoothing parameter.

Lemma 2.4. *([Pei08, Lemma 3.5]) For any n dimensional lattice Λ, and real $\epsilon > 0$, it follows that*

$$\eta_\epsilon(\Lambda) \leq \frac{\sqrt{n} \cdot \sqrt{\log(2n(1+1/\epsilon)/\pi)}}{\lambda_1^{(2)}(\Lambda^*)},$$

where $\lambda_1^{(p)}(\Lambda^*)$ is the l_2 norm of the shortest vector in Λ^*. In particular, for any $\omega(\sqrt{\log n})$ function, there is a negligible function $\epsilon(n)$ in which

$$\eta_\epsilon(\Lambda) \leq \frac{\sqrt{n} \cdot \omega(\sqrt{\log n})}{\lambda_1^{(2)}(\Lambda^*)}.$$

Moreover, another upper bound on the smoothing parameter, related to the Gram-Schmidt norm, is shown in [GPV08].

Lemma 2.5. *([GPV08, Lemma 3.1]) For any n dimensional lattice Λ, a basis $\boldsymbol{B}$ of Λ, a real $\epsilon > 0$, it follows that*

$$\eta_\epsilon(\Lambda) \leq \| \tilde{\boldsymbol{B}} \| \cdot \sqrt{\log(2n(1+1/\epsilon))/\pi}.$$

Then for any $\omega(\sqrt{\log n})$, there is a negligible function $\epsilon(n)$ satisfying $\eta_\epsilon(\Lambda) \leq \| \tilde{\boldsymbol{B}} \| \cdot \omega(\sqrt{\log n})$.

In [GPV08], Gentry, Peikert, and Vaikuntanathan propose algorithms that sample from a lattice along the discrete Gaussian distribution.

Lemma 2.6. *([GPV08, Theorem 4.1]) There is a probabilistic polynomial-time algorithm that takes as input a basis $\boldsymbol{B}$ of n dimensional lattice $\Lambda = \mathcal{L}(\boldsymbol{B})$, a parameter $s \geq \| \tilde{\boldsymbol{B}} \| \cdot \omega(\sqrt{\log n})$, and a center $\boldsymbol{c} \in \mathbb{R}^n$, and outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\Lambda,s,\boldsymbol{c}}$.*

Lemma 2.7. *([GPV08, Theorem 5.9]) There is a probabilistic polynomial-time algorithm* SamplePre *that takes as input a basis $\boldsymbol{B}$ of an n dimensional lattice $\Lambda = \mathcal{L}(\boldsymbol{B})$, a parameter $s \geq \| \tilde{\boldsymbol{B}} \| \cdot \omega(\sqrt{\log n})$, and a vector $\boldsymbol{u} \in \mathbb{R}^n$, and outputs a sample from a distribution that is statistically close to $\mathcal{D}_{\Lambda+\boldsymbol{u},s}$.*

Micciancio and Regev show the fact that we have an upper bound on the norm of the element sampled from a discrete Gaussian over the lattice.

Lemma 2.8. *([MR07, Lemma 4.4]) For any n dimensional lattice Λ, real number $s > \omega(\sqrt{\log n})$, and center $\boldsymbol{c} \in \mathbb{R}^n$,it holds that*

$$Pr[\| x \| > s\sqrt{n} : x \xleftarrow{R} \mathcal{D}_{\Lambda,s,\mathbf{c}}] \leq 2^{-n+1}.$$

We say that the random variable X over $\mathbb{R}$ is δ-*sub-Gaussian* if for all $t \in \mathbb{R}$, the moment generating function of X satisfies

$$\mathbb{E}[\exp(2\pi t X)] \leq \exp(\delta) \cdot \exp(\pi s^2 t^2).$$

The term $\exp(\pi s^2 t^2)$ is correctly the moment generating function of the Gaussian distribution $\mathcal{D}_s$. According to [MP12], we need the factor $\exp(\delta)$ when working with *discrete* Gaussian distributions. We say that the random variable is *sub-Gaussian* if the factor $\delta = 0$. The class of sub-Gaussian random variables is quite a wide class that contains the standard normal and all bounded random variables. We have the following fact about the singular value of the sub-Gaussian random variables.

Lemma 2.9. *(derived from [Ver12, Corollary 5.35]) Let $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ be a δ-sub-Gaussian random matrix with parameter s. There is a universal constant $C > 0$ such that for any $t > 0$, it follows that $s_1(\boldsymbol{X}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$ except with probability at most $2\exp(\delta)\exp(-\pi t^2)$.*

2.6 The Small Integer Solution Problem

The *Small Integer Solution* (SIS) problem is the problem that finds a short nonzero vector in a certain class of lattices. In [BF11], Boneh and Freeman define a general version of the SIS problem as follows.

Definition 2.2. *Let $\mathcal{L}_n$ be the distribution over an n dimensional lattice, and let $\mathcal{L} = \{\mathcal{L}_n\}_{n\in\mathbb{N}}$ be a distribution ensemble. The $\mathcal{L} - \mathsf{SIS}_{n,\beta}$ problem is as follows: given a lattice $\Lambda \xleftarrow{R} \mathcal{L}_n$, find a nonzero vector $\boldsymbol{v} \in \Lambda$ satisfying $\| \boldsymbol{v} \| \leq \beta$. If a probabilistic, polynomial-time algorithm $\mathcal{B}$ takes as input a lattice Λ, the* advantage *of $\mathcal{B}$, denoted by $\mathcal{L} - \mathsf{SIS} - Adv[\mathcal{B}, (n, \beta)]$, is defined as the probability that $\mathcal{B}$ outputs a solution to an $\mathcal{L} - \mathsf{SIS}_{n,\beta}$ problem. $\mathcal{L} - \mathsf{SIS}_{n,\beta}$ is* infeasible *if for all $\mathcal{B}$, $\mathcal{L} - \mathsf{SIS} - Adv[\mathcal{B}, (n, \beta)] = negl(n)$.*

3 Homomorphic Signatures for Polynomial Functions

In this section, we describe the polynomially homomorphic signature scheme of [BF11]. Boneh and Freeman construct the scheme by using ideal lattices in a way that is a signature analogue of the fully homomorphic encryption scheme proposed by [Gen09b].

A polynomially homomorphic signature scheme consists of four probabilistic, polynomial-time algorithms $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Evaluate})$.

- $\mathsf{Setup}(1^n, k)$:On input a security parameter n and a data set size k, generate a public key and a secret key.
  1. Choose a monic irreducible polynomial $F(x)$ of degree n from $\mathbb{Z}[x]$. Let $R := \mathbb{Z}[x]/\langle F(x)\rangle$ be the polynomial ring corresponding to the lattice $\mathbb{Z}^n$. The ring R is the signature space.
  2. Run the $\mathsf{PrincGen}$ algorithm of Lemma 2.3 twice on input a polynomial $F(x)$ and its degree n to obtain distinct principal degree-one prime ideals $\mathcal{I} = (p, x - a)$ and $\mathcal{J} = (q, x - b)$ of R with their generators $g_{\mathcal{I}}$ and $g_{\mathcal{J}}$, respectively. The prime p defines the message space $\mathbb{F}_p$.
  3. Applying the algorithm of Lemma 2.1 to the set $\{g_{\mathcal{I}}g_{\mathcal{J}}, g_{\mathcal{I}}g_{\mathcal{J}}x, \ldots, g_{\mathcal{I}}g_{\mathcal{J}}x^{n-1}\}$, generate a basis $\mathbf{T}$ of $\mathcal{I}\cdot\mathcal{J}$.
  4. Let $v := \gamma_F^2 \cdot n^3$. Choose positive integers $y = \mathrm{poly}(n)$ and $d = O(1)$. The parameters y and d define the set of admissible function $\mathcal{F} \subset \mathbb{F}_p[x_1, ..., x_k]$ with coefficients in $\{-y, ..., y\}$, degree at most d, and constant term zero. Let $\{Y_j\}_{j=1}^{l}$ (where $l = \binom{k+d}{d} - 1$) be the set of non-constant monomials $x_1^{e_1} \cdots x_k^{e_k}$ of degree $\sum e_k \leq d$, ordered lexicographically. Let $\vec{m} = (m_1, \ldots, m_k)$, then any polynomial function $f \in \mathcal{F}$ is encoded as $\langle f\rangle = (c_1, ..., c_l) \in \mathbb{Z}^l$, and determined by $f(\vec{m}) = \sum_{j=1}^{l} c_j Y_j(\vec{m})$.
  5. Let $H : \{0,1\}^* \to (\mathbb{F}_q)^k$ be a hash function.
  6. Output $\mathsf{pk} = (F(x), p, q, a, b, v, y, d, H)$ and $\mathsf{sk} = \mathbf{T}$.
- $\mathsf{Sign}(\mathsf{sk}, \tau, m, i)$:On input a secret key sk, a tag τ, a message m, and an index i of m in the data set, output a signature on m.
  1. Compute $(\alpha_1, \ldots, \alpha_k) \leftarrow H(\tau)$.
  2. Compute $h = h(x) \in R$ satisfying $h(a) = m \mod p$ and $h(b) = \alpha_i \mod q$.
  3. Output a signature
  $$\sigma \leftarrow \mathsf{SamplePre}(\mathcal{I}\cdot\mathcal{J}, \mathbf{T}, h, v) \in (\mathcal{I}\cdot\mathcal{J}) + h.$$
- $\mathsf{Verify}(\mathsf{pk}, \tau, m, \sigma, f)$:On input a public key pk, a tag τ, a message m, a signature σ, and a function f, verify that σ is a signature on m computed by applying f. If the received signature holds all of the following conditions, then output 1; otherwise output 0.
  1. $\| \sigma \| \leq l \cdot y \cdot \gamma_F^{d-1} \cdot (v\sqrt{n} \cdot \log n)^d$.
  2. $\sigma(a) \mod p = m$.
  3. $\sigma(b) \mod q = \omega_\tau(\langle f\rangle)$. To evaluate the hash function $\omega_\tau(\langle f\rangle)$, do the following:
     (a) Compute $(\alpha_1, \ldots, \alpha_k) \leftarrow H(\tau)$.
     (b) Evaluate $\omega_\tau(\langle f\rangle) = \sum_{j=1}^{l} c_j Y_j(\alpha_1, ..., \alpha_k)$.
- $\mathsf{Evaluate}(\mathsf{pk}, \tau, f, (\sigma_1, ..., \sigma_k))$:Take as input a public key pk, a tag τ, a function f, and a tuple of signatures $(\sigma_1, ..., \sigma_k)$, and evaluate f on $(\sigma_1, ..., \sigma_k)$.
  1. Lift $f \in \mathbb{F}_p[x_1, ..., x_k]$ to $\mathbb{Z}[x_1, ..., x_k]$ by redefining $\hat{f} := \sum_{j=1}^{l} c_j Y_j(x_1, ..., x_k)$.
  2. Output $\hat{f}(\sigma_1, ..., \sigma_k)$.

Boneh and Freeman show that the above scheme is correct in the following Lemma.

Lemma 3.1. *([BF11, Lemma 6.1]) The polynomially homomorphic signature scheme is correct with overwhelming probability.*

4 An Improvement of the Polynomially Homomorphic Signature

In this section, we show how to reduce the length of signatures in the polynomially homomorphic signature scheme of [BF11]. In the first place, the scheme generates signatures by computing a preimage of a special trapdoor one-way function, called *preimage sampleable functions*. If we have a trapdoor for the functions, we can sample a preimage according to a Gaussian-like distribution. For example, we consider the function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \mod q$ for some matrix $\mathbf{A}$. When we fix a vector $\mathbf{u}$, if we know a trapdoor for $f_{\mathbf{A}}$, we can sample a preimage $\mathbf{x}$ such that $\mathbf{u} = \mathbf{A}\mathbf{x} \mod q$ along the Gaussian distribution. Sampling algorithms for such functions were proposed in [GPV08, MP12].

4.1 The Preimage Sampling Algorithm of [MP12]

In the algorithm of [MP12], Micciancio and Peikert use the matrix $\mathbf{G}$, called a *primitive matrix*, that we can efficiently sample a preimage for the function $f_{\mathbf{G}}(\mathbf{x}) = \mathbf{Gx} \mod q$. We say that $\mathbf{G}$ is primitive if its rows generate all of $\mathbb{Z}_q^m$, i.e., $\mathbf{G} \cdot \mathbb{Z}^w = \mathbb{Z}_q^m$. Usually, $\mathbf{G}$ is given as some fixed and public matrix. Concretely, when we let $\mathbf{g}^t = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}^l$ where $l = \lceil \log_2 q \rceil$, $\mathbf{G}$ is defined as follows:

$$\mathbf{G} := \begin{bmatrix} \mathbf{g}^t & & & \\ & \mathbf{g}^t & & \\ & & \ddots & \\ & & & \mathbf{g}^t \end{bmatrix} \in \mathbb{Z}^{n \times nl}.$$

In [MP12], properties of $\mathbf{G}$ are summarized in the following Lemma.

Lemma 4.1. *([MP12, Theorem 4,1]) For any integer $q \geq 2$, $n \geq 1$, and $l = \lceil \log_2 q \rceil$, there is a primitive matrix $\boldsymbol{G} \in \mathbb{Z}^{n \times nl}$ such that*

- *The lattice $\Lambda^{\perp}(\boldsymbol{G})$ has a known basis $\boldsymbol{S} \in \mathbb{Z}^{nl \times nl}$ with $\| \tilde{\boldsymbol{S}} \| \leq \sqrt{5}$ and $\| \boldsymbol{S} \| \leq \max\{\sqrt{5}, \sqrt{l}\}$.*
- *Preimage sampling for $f_{\boldsymbol{G}}(\boldsymbol{x}) = \boldsymbol{Gx} \mod q$ with Gaussian parameter $s \geq \| \tilde{\boldsymbol{S}} \| \cdot \omega(\sqrt{\log n})$ can be performed in quasilinear $O(n \cdot \log^c n)$ time.*

Since $\mathbf{G}$ is fixed, the basis $\mathbf{S}$ of $\Lambda^{\perp}(\mathbf{G})$ is also fixed. Micciancio and Peikert define $\mathbf{S}$ as

$$\mathbf{S} := \begin{bmatrix} \mathbf{S}_l & & & & \\ & \mathbf{S}_l & & & \\ & & \ddots & & \\ & & & \mathbf{S}_l & \\ & & & & \mathbf{S}_l \end{bmatrix} \in \mathbb{Z}^{nl \times nl},$$

where we let $(q_0, \ldots, q_{l-1})$ be the binary decomposition of q and

$$\mathbf{S}_l := \begin{bmatrix} 2 & & & & q_0 \\ -1 & 2 & & & q_1 \\ & -1 & & & q_2 \\ & & \ddots & & \vdots \\ & & & 2 & q_{l-2} \\ & & & -1 & q_{l-1} \end{bmatrix} \in \mathbb{Z}^{l \times l}.$$

The preimage sampleable function of [MP12] allows us to sample shorter preimages than [GPV08, AP11]. In [MP12], Micciancio and Peikert use a new trapdoor notion called a ***G*-trapdoor** to improve the quality of samples (the length of the preimage). By the norm of the ideal lattices, we cannot use this notion without any changes, so we slightly modify and inject the notion into the proposed signing algorithm.

Definition 4.1. *Let $\boldsymbol{A} \in \mathbb{Z}_q^{n \times n}$ and $\boldsymbol{G} \in \mathbb{Z}_q^{n \times nl}$ be matrices with $l = \lceil \log_2 q \rceil$. A **$\boldsymbol{G}$-trapdoor for $\boldsymbol{A}$** is a matrix $\boldsymbol{R} \in \mathbb{Z}^{n \times (nl-n)}$ such that $\boldsymbol{A}[\boldsymbol{R} \parallel \boldsymbol{I}_n] = \boldsymbol{HG}$ for some invertible matrix $\boldsymbol{H} \in \mathbb{Z}^{n \times n}$.*

The matrix $\mathbf{H}$ is determined and can be efficiently computed from the matrices $\mathbf{A}, \mathbf{R}$ and,$\mathbf{G}$.

Algorithm 1 $\mathsf{TrapGen}'(\mathbf{T}, m)$: On input a basis $\mathbf{T} \in \mathbb{Z}^{n \times n}$ and a integer m, compute a matrix $\mathbf{A}$, $\mathbf{G}$-trapdoor $\mathbf{R}$, and a tag $\mathbf{H}$.

1: Compute a vector $\mathbf{a} \in \mathbb{Z}^n$ such that $\mathbf{Ta} \equiv \mathbf{0} \mod pq$. Since $\det(\mathbf{T}) \equiv 0 \mod pq$, $\mathbf{T}$ has pq as an eigenvalue. For some vector $\mathbf{a}$, $\mathbf{Ta} = pq \cdot \mathbf{a} \equiv 0 \mod pq$, so we can always compute the vector $\mathbf{a}$ via linear algebra.
2: Let a be the element in R corresponding to $\mathbf{a}$. Define a matrix as $\mathbf{A} := \{a, ax, \ldots, ax^{n-1}\} \in \mathbb{Z}^{n \times n}$.
3: Let $l := \lceil \log pq \rceil$ and choose a $\mathbf{G}$-trapdoor $\mathbf{R} \xleftarrow{R} \mathcal{D}_{\mathbb{Z},1}^{n \times (nl-n)}$.
4: For the primitive matrix $\mathbf{G} \in \mathbb{Z}^{n \times nl}$, compute a matrix $\mathbf{H} \in \mathbb{Z}^{n \times n}$ satisfying $\mathbf{A}[\mathbf{R} \parallel \mathbf{I}_n] = \mathbf{HG}$.

Algorithm 2 $\mathsf{SamplePre}'(\mathbf{A}, \mathbf{R}, \mathbf{H}, \mathbf{u}, v)$ of [MP12]: On input a matrix $\mathbf{A}$, a $\mathbf{G}$-trapdoor $\mathbf{R}$, a tag $\mathbf{H}$, a vector $\mathbf{u}$, and a Gaussian parameter v, sample from the discrete Gaussian distribution $\mathcal{D}_{\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}), v \cdot \log n}$.

1: Choose $s \geq \sqrt{5}$ and let $\Sigma_{\mathbf{G}} := s^2 \cdot \mathbf{I}_{nl}$.
2: Let $\Sigma_{\mathbf{p}} := v^2 \mathbf{I}_n - [\mathbf{R} \parallel \mathbf{I}_n] \Sigma_{\mathbf{G}} \begin{bmatrix} \mathbf{R}^t \\ \mathbf{I}_n \end{bmatrix}$.
3: Sample $\mathbf{p} \xleftarrow{R} \mathcal{D}_{\mathbb{Z}^n, \sqrt{\Sigma_{\mathbf{p}}} \cdot \log n}$.
4: Compute $\mathbf{v} := \mathbf{H}^{-1}(\mathbf{u} - \mathbf{Ap})$.
5: Sample $\mathbf{z} \xleftarrow{R} \mathcal{D}_{\Lambda_{\mathbf{v}}^{\perp}(\mathbf{G}), \sqrt{\Sigma_{\mathbf{G}}} \cdot \log n}$.
6: Compute $\mathbf{x} := \mathbf{p} + \begin{bmatrix} \mathbf{R} \parallel \mathbf{I}_n \end{bmatrix} \mathbf{z}$.

4.2 Constructions

In Algorithms 1 and 2, we describe the algorithm that generates a $\mathbf{G}$-trapdoor for the ideal lattice $\mathcal{I} \cdot \mathcal{J}$ and the Gaussian sampler of [MP12] for generating signatures, respectively. By replacing $\mathsf{SamplePre}$ with $\mathsf{SamplePre}'$, we obtain the signing algorithm that generates smaller signatures. For this replacement, we slightly modify the Setup algorithm of [BF11] as follows.

$\mathsf{Setup}'(1^n, k)$:On input a security parameter n and a data set size k, generate a public key and a secret key.

1. Choose a monic irreducible polynomial $F(x) \in \mathbb{Z}[x]$ of degree n. Let $R := \mathbb{Z}[x]/\langle F(x) \rangle$ be the polynomial ring corresponding to the lattice $\mathbb{Z}^n$.
2. Run the $\mathsf{PrincGen}$ algorithm of Lemma 2.3 twice on input a polynomial $F(x)$ and its degree n to obtain distinct principal degree-one prime ideals $\mathcal{I} = (p, x - a)$ and $\mathcal{J} = (q, x - b)$ of R with their generators $g_{\mathcal{I}}$ and $g_{\mathcal{J}}$, respectively.
3. Define a matrix
$$\mathbf{T} := \{g_{\mathcal{I}} g_{\mathcal{J}}, g_{\mathcal{I}} g_{\mathcal{J}} x, \ldots, g_{\mathcal{I}} g_{\mathcal{J}} x^{n-1}\}.$$
4. $(\mathbf{A}, \mathbf{R}, \mathbf{H}) \leftarrow \mathsf{TrapGen}'(\mathbf{T})$.

5. Let $v := \sqrt{n} \cdot \sqrt{s_1(\mathbf{R})^2 + 1} \cdot s_1(\mathbf{S})$. Choose positive integers $y = \mathrm{poly}(n)$ and $d = O(1)$.
6. Let $H : \{0,1\}^* \to (\mathbb{F}_q)^k$ be a hash function.
7. Output $\mathsf{pk} = (F(x), p, q, a, b, v, y, d, H)$ and $\mathsf{sk} = (\mathbf{A}, \mathbf{R}, \mathbf{H})$.

We also modify the Sign algorithm as follows.

$\mathsf{Sign}'(\mathsf{sk}, \tau, m, i)$:On input a secret key sk, a tag τ, a message m, and an index i of m in the data set, output a signature on m.

1. Compute $(\alpha_1, \ldots, \alpha_k) \leftarrow H(\tau)$.
2. Compute $h = h(x) \in R$ satisfying $h(a) = m \mod p$ and $h(b) = \alpha_i \mod q$. Let $\mathbf{h}$ be the vector identified with the ring element h.
3. $\mathbf{u} := \mathbf{Ah} \mod q$.
4. Output the signature $x \in h + (\mathcal{I} \cdot \mathcal{J})$ corresponding to the vector

$$\mathbf{x} \leftarrow \mathsf{SamplePre}'(\mathbf{A}, \mathbf{R}, \mathbf{H}, \mathbf{u}, v).$$

The system parameters defined by pk are the same as described in Section 3.

Lemma 4.2. *The improved polynomially homomorphic signature scheme is correct.*

Proof. Since the fresh signature $\mathbf{x}$ generated by Sign' holds that $\mathbf{Ax} = \mathbf{u}$, we have $\mathbf{x} = \mathbf{h} + \Lambda^{\perp}(\mathbf{A})$. Via the coefficient embedding, the ring element x corresponding to the vecotr $\mathbf{x}$ is equal to a ring element in $h + \mathcal{I} \cdot \mathcal{J}$. Hence, for fresh signatures, we have

$$x(a) \mod p = m_i \text{ and } x(b) \mod q = \alpha_i,$$

so the verification conditions are satisfied. Evaluated signatures also hold the verification conditions as well as in Lemma 3.1. □

When the Gaussian parameter v is greater than the smoothing parameter of the lattice Λ, the Gaussian distribution $\mathcal{D}_{\Lambda,v,\mathbf{c}}$ behaves like the continuous Gaussian distribution $\mathcal{D}_{v,\mathbf{c}}$ in many respects. In [MP12], the knowledge related to the $\mathbf{G}$-trapdoor gives a short basis of $\Lambda^{\perp}(\mathbf{A})$, which we denote by $\mathbf{B_A}$. The Gaussian parameter $v \cdot \log n = \sqrt{s_1(\mathbf{R})^2 + 1} \cdot \sqrt{s_1(\Sigma_{\mathbf{G}}) + 2} \cdot \log n$, chosen in the preimage sampling algorithm of [MP12], is greater than $\| \tilde{\mathbf{B}}_{\mathbf{A}} \| \cdot \log n \geq \eta_{\epsilon}(\Lambda^{\perp}(\mathbf{A}))$. Therefore, we have $v \cdot \log n \geq \eta_{\epsilon}(\Lambda^{\perp}(\mathbf{A}))$. In the proposed scheme, however, it is difficult to obtain such a short basis. Thus we show the alternative upper bound on the smoothing parameter.

Lemma 4.3. *For any $\boldsymbol{G}$-trapdoor $\boldsymbol{R} \in \mathbb{Z}^{n \times (nl-n)}$, any basis $\boldsymbol{S} \in \mathbb{Z}^{nl \times nl}$ of $\Lambda^{\perp}(\boldsymbol{G})$, and negligible $\epsilon > 0$, we have*

$$\eta_{\epsilon}(\Lambda^{\perp}(\boldsymbol{A})) \leq \sqrt{n} \cdot s_1([\boldsymbol{R} \parallel \boldsymbol{I}_n]) \cdot s_1(\boldsymbol{S}) \cdot \omega(\sqrt{\log n}).$$

Proof. By the structure of $[\mathbf{R} \parallel \mathbf{I}_n] \cdot \mathbf{S}$, for any $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})^*$, it follows that $\mathbf{v}^t \cdot [\mathbf{R} \parallel \mathbf{I}_n] \cdot \mathbf{S} \in \mathbb{Z}^{nl} \setminus \{\mathbf{0}\}$. Now we let $\mathbf{S_A} = [\mathbf{R} \parallel \mathbf{I}_n] \cdot \mathbf{S}$ and denote the ith column of $\mathbf{S_A}$ by $\mathbf{S}_{\mathbf{A},i}$. Then since it follows that for some vector $\mathbf{e} \in \mathbb{Z}^{nl}$,

$$\langle \mathbf{v}, \mathbf{S_A} \cdot \mathbf{e} \rangle = \mathbf{e}_1 \cdot \langle \mathbf{v}, \mathbf{S}_{\mathbf{A},1} \rangle + \mathbf{e}_2 \cdot \langle \mathbf{v}, \mathbf{S}_{\mathbf{A},2} \rangle + \cdots + \mathbf{e}_{nl} \cdot \langle \mathbf{v}, \mathbf{S}_{\mathbf{A},nl} \rangle,$$

there are unit vectors such that $\langle \mathbf{v}, \mathbf{S_A} \cdot \mathbf{e} \rangle \geq 1$. For such a unit vector $\mathbf{e}$, we have

$$\begin{aligned} 1 &\leq \langle \mathbf{v}, \mathbf{S_A} \cdot \mathbf{e} \rangle \\ &\leq \| \mathbf{v} \|_2 \cdot \| \mathbf{S_A} \cdot \mathbf{e} \|_2 \\ &\leq \| \mathbf{v} \|_2 \cdot s_1(\mathbf{S_A}) \cdot \| \mathbf{e} \|_2 \\ &\leq \| \mathbf{v} \|_2 \cdot s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\mathbf{S}) \qquad (1) \end{aligned}$$

Let $\lambda_1^{(2)}(\Lambda^{\perp}(\mathbf{A})^*)$ be the l_2 norm of the shortest vector in $\Lambda^{\perp}(\mathbf{A})^*$. Since we have the inequation (1) for any vector $\mathbf{v} \in \Lambda^{\perp}(\mathbf{A})^*$, it follows that

$$\begin{aligned} (1) &\Leftrightarrow 1 \leq \lambda_1^{(2)}(\Lambda^{\perp}(\mathbf{A})^*) \cdot s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\mathbf{S}) \\ &\Leftrightarrow \frac{1}{\lambda_1^{(2)}(\Lambda^{\perp}(\mathbf{A})^*)} \leq s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\mathbf{S}). \end{aligned}$$

Therefore, Lemma 2.4 gives the upper bound on the smoothing parameter as

$$\begin{aligned} \eta_{\epsilon}(\Lambda^{\perp}(\mathbf{A})) &\leq \frac{\sqrt{n}}{\lambda_1^{(2)}(\Lambda^{\perp}(\mathbf{A})^*)} \cdot \omega(\sqrt{\log n}) \\ &\leq \sqrt{n} \cdot s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\mathbf{S}) \cdot \omega(\sqrt{\log n}). \end{aligned}$$

□

4.3 The Length of Fresh Signatures

In the scheme of [BF11], when we set $F(x) = x^n + 1$, the expansion factor γ_F is $\sqrt{n}$ by Lemma 2.2, so the length of fresh signatures is $v\sqrt{n} \cdot \log n = \tilde{\mathrm{O}}(n^{4.5})$ by Lemma 2.8. In the proposed scheme, the length of fresh signatures is $\tilde{\mathrm{O}}(n^3)$.

The proposed scheme uses the algorithm $\mathsf{SamplePre}'$ of [MP12] to generate signatures. $\mathsf{SamplePre}'$ outputs $\mathbf{x} = \mathbf{p} + \begin{bmatrix} \mathbf{R} \parallel \mathbf{I}_n \end{bmatrix} \mathbf{z}$, so by the linearity of the variance, $\mathbf{x}$ is output from the Gaussian distribution with parameter $v \cdot \log n$, and hence $\mathbf{x}$ has a length of at most $v\sqrt{n} \cdot \log n$ by Lemma 2.8. In the Setup' algorithm, the Gaussian parameter v is set as $v = \sqrt{n} \cdot \sqrt{s_1(\mathbf{R})^2 + 1} \cdot s_1(\mathbf{S})$, which satisfies $v \cdot \log n \geq \eta_{\epsilon}(\mathcal{I} \cdot \mathcal{J})$ by Lemma 4.3. A matrix is positive definite if and only if eigenvalues of the matrix are positive, and the eigenvalues are the square of singular values of the matrix, so $\Sigma_{\mathbf{p}}$ is positive definite as long as $v \geq s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}}) \geq s_1([\mathbf{R} \parallel \mathbf{I}_n] \cdot \sqrt{\Sigma_{\mathbf{G}}})$. In Lemma 2.3, PrincGen chooses the ideal of norm $\| G(x) \|_2^n \cdot \| F(x) \|_2^n$ where $G(x)$ is the polynomial whose coefficients are at most about $2^{\sqrt{n}}$ and $F(x)$ is the cyclotomic polynomial $x^n + 1$ chosen in the Setup'. Since $pq \approx (\| G(x) \|_2^n \cdot \| F(x) \|_2^n)^2 = (n \cdot 2^{2\sqrt{n}})^n \cdot 2^n$, $\sqrt{l} = \sqrt{\lceil \log pq \rceil}$ is $\mathrm{O}(n^{3/4})$. By the structure of $\mathbf{S}$, $s_1(\mathbf{S}) = s_1(\mathbf{S}_l) = \mathrm{O}(\sqrt{l}) = \mathrm{O}(n^{3/4})$, so the Gaussian parameter v satisfies the condition $v \geq s_1([\mathbf{R} \parallel \mathbf{I}_n]) \cdot s_1(\sqrt{\Sigma_{\mathbf{G}}})$.

By Lemma 2.9, $s_1(\mathbf{R}) = \mathrm{O}(\sqrt{nl}) = \mathrm{O}(n^{5/4})$, since the distribution $\mathcal{D}_{\mathbb{Z},1}^{n\times(nl-n)}$ used to choose $\mathbf{R}$ is a sub-Gaussian distribution. As described above, $s_1(\mathbf{S}) = \mathrm{O}(n^{3/4})$, so $v = \sqrt{n} \cdot \sqrt{s_1(\mathbf{R})^2 + 1} \cdot s_1(\mathbf{S}) = \mathrm{O}(n^{2.5})$. Eventually, the length of fresh signatures is $\tilde{\mathrm{O}}(n^3)$ by Lemma 2.8.

4.4 Unforgeability

Boneh and Freeman define the unforgeability of homomorphic signature schemes in [BF11].

Definition 4.2. *A homomorphic signature scheme* $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Evaluate})$ *is unforgeable if for all* k *and all polynomial-time adversaries* $\mathcal{A}$*, the advantage of the adversary* $\mathcal{A}$ *in the following game is negligible in the security parameter* n*.*

> ***Setup:*** *To obtain* $(\mathsf{pk}, \mathsf{sk})$*, the challenger runs* $\mathsf{Setup}(1^n, k)$ *and gives* pk *to the adversary* $\mathcal{A}$*. The public key defines a message space* $\mathcal{M}$*, a signature space* Σ*, and the set of admissible functions* $f : \mathcal{M}^k \to \mathcal{M}$*.*
>
> ***Queries:*** *The adversary can query to the Signing oracle, adaptively.* $\mathcal{A}$ *specifies a sequence of data sets* $\boldsymbol{m}_i \in \mathcal{M}^k$*. For* $i = 1, ..., k$*, the challenger chooses* $\tau_i \xleftarrow{U} \{0,1\}^n$ *uniformly at random, and gives the tag* τ_i *and the signatures* $\sigma_{i,j} \leftarrow \mathsf{Sign}(\mathsf{sk}, \tau_i, m_{i,j}, j)$ *for* $j = 1, ..., k$ *to* $\mathcal{A}$*.*
>
> ***Output:*** $\mathcal{A}$ *outputs a tag* τ^**, a message* $m^* \in \mathcal{M}$*, a function* $f \in \mathcal{F}$*, and a signature* $\sigma^* \in \Sigma$*.*

The adversary wins the game if

$$\mathsf{Verify}(\mathsf{pk}, \tau^*, m^*, \sigma^*, f) = 1$$

and

1. *for all* i*,* $\tau^* \neq \tau_i$*, or*
2. *for some* i*,* $\tau^* = \tau_i$*, but* $m^* \neq f(\vec{m}_i)$*.*

The first condition is called a type 1 forgery*, and the second condition is called a* type 2 forgery*. The advantage of* $\mathcal{A}$ *is defined as the probability that* $\mathcal{A}$ *wins the above game.*

The following theorem states the security of the improved polynomially homomorphic signature scheme.

Theorem 4.1. *For some constant* n*, let* F_n *be the polynomial chosen in Step 1 of the* Setup' *algorithm, and let* $\mathcal{L}_n$ *be the probability distribution of the ideal* $\mathcal{J}$ *output by the* $\mathsf{PrincGen}$ *algorithm when given a polynomial* $F_n(x)$ *and a parameter* $\delta = n$*. Let* $\mathcal{L}_F$ *be the set* $\{\mathcal{L}_n\}_{n\in\mathbb{N}}$*, and let* $d = p/2$*. If* $\mathcal{L}_F - \mathsf{SIS}_\beta$ *is infeasible for*

$$\beta = 2 \cdot \binom{k+d}{d} \cdot y \cdot \gamma_F^{d-1} \cdot (v\sqrt{n})^d,$$

then the improved polynomially homomorphic signature scheme is unforgeable in the random oracle model.

Proof. Here, we only describe the construction of the algorithm that breaks the unforgeability of the improved polynomially homomorphic signature scheme, since the rest of the proof is almost identical to the one of [BF11]. Let $\mathcal{A}$ be the polynomial-time adversary that wins the security game of Definition 4.2. When given the challenge lattice (Ideal) $\mathcal{J}$, we construct the algorithm $\mathcal{B}$ that solves the $\mathcal{L}_F - \mathsf{SIS}_\beta$ problem. The algorithm $\mathcal{B}$ simulates the Setup' algorithm, the signing oracle, and the hash oracle.

> **Setup' algorithm:** Using the $\mathsf{PrincGen}$ algorithm, generate an ideal $\mathcal{I}$ with its generator $g_\mathcal{I}$. For the ideal $\mathcal{I}$, using the algorithm $\mathsf{TrapGen}'$, obtain $\mathbf{A}_\mathcal{I}, \mathbf{R}_\mathcal{I}$, and $\mathbf{H}_\mathcal{I}$. Choose all other parameters as in the real Setup' algorithm from the ideal $\mathcal{I}$ and the challenge ideal $\mathcal{J}$.
>
> **Hash oracle** H**:** On input τ, if τ has already been queried to H, then return $H(\tau)$. Otherwise, for $i = 1, \ldots, k$, choose $\sigma_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,v}$, and define $H(\tau) := (\sigma_1(b) \mod q, \ldots, \sigma_k(b) \mod q)$.
>
> **Signing oracle:** When the adversary $\mathcal{A}$ queries the data set $(m_1, \ldots, m_k) \in (\mathbb{F}_p)^k$, do the following.
>
> 1. Choose $\tau \xleftarrow{U} \{0,1\}^n$. If τ has already been queried, then abort.
> 2. For $i = 1, \ldots, k$, choose $h_i \in R$ satisfying $h_i(a) \mod p = m_i$.
> 3. Compute $\mathbf{u}_i := \mathbf{A}_\mathcal{I}\mathbf{h}_i$ for $\mathbf{h}_i \in \mathbb{Z}^n$ corresponding to h_i.
> 4. Choose $\sigma_i \leftarrow \mathsf{SamplePre}'(\mathbf{A}_\mathcal{I}, \mathbf{R}_\mathcal{I}, \mathbf{H}_\mathcal{I}, \mathbf{u}_i, v)$.
> 5. $H(\tau) := (\sigma_1(b) \mod q, \ldots, \sigma_k(b) \mod q)$.
> 6. Give τ and $(\sigma_1, \ldots, \sigma_k)$.

Eventually, the adversary $\mathcal{A}$ outputs a tag τ^*, a message m^*, a function f encoded as $\langle f \rangle = (c_1, \ldots, c_l) \in \mathbb{Z}^l$, and a signature σ^*. Without loss of generality, we may assume that the tag τ^* has already been queried to the hash oracle, and let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_k)$ be the set of values chosen when computing $H(\tau^*)$. We let $\sigma_f := \sum_i c_i Y_i(\vec{\sigma})$, and the algorithm $\mathcal{B}$ outputs $\sigma^* - \sigma_f$. □

5 Conclusion

We constructed a polynomially homomorphic signature scheme with shorter signatures than the ones of [BF11], replacing the preimage sampling algorithm of [GPV08] with the algorithm of [MP12]. In [BF11], when the cyclotomic polynomial to define polynomial rings is $x^n + 1$, the length of fresh signatures is $\tilde{\mathrm{O}}(n^{4.5})$. In the proposed scheme, fresh signatures have length $\tilde{\mathrm{O}}(n^3)$. The secret key of the proposed scheme is of bit length $\mathrm{O}(n^{3.5})$, while the secret key used in [BF11] is of bit length $\tilde{\mathrm{O}}(n)$. Therefore, it is desirable to reduce the length of the secret key.

References

[Ajt99] Miklós Ajtai. Generating hard instances of the short basis problem. *ICALP*, pages 1–9, 1999.

[AP11] Joel Alwen and Chris Peikert. Generating shorter bases for hard random lattices. *Theory of Computing Systems*, 48(3):535–553, 2011.

[BF11] Dan Boneh and David Mandell Freeman. Homomorphic signature for polynomial functions. *Advances in Cryptology - EUROCRYPT 2011, LNCS*, 6632:149–168, 2011.

[CHKP12] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, October 2012.

[Gen09a] Craig Gentry. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. PhD thesis, Stanford University, http://crypto.stanford.edu/craig, 2009.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. *STOC*, pages 169–178, 2009.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. How to use a short basis: Trapdoors for hard lattices and new cryptographic constructions. *STOC*, pages 197–206, 2008.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. *Advances in Cryptology - EUROCRYPT 2012, LNCS*, 7237:738–755, 2012.

[MG02] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. *Advances in Cryptology - EUROCRYPT 2012, LNCS*, 7237:700–718, 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reduction based on Gaussian measures. *SIAM J.Comput*, 37(1):267–302, 2007.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Berstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-quantum Cryptography*, chapter 5, pages 147–187. Springer, 2009. Available at http://www.cs.tau.ac.il/ odedr/papers/pqc.pdf.

[Pei08] Chris Peikert. Limits on the hardness of the lattice problems in l_∞ norms. *Computational Complexity*, 17(2):300–351, 2008.

[Pei10] Chris Peikert. An efficient and parellel Gaussian sampler for lattices. *Advances in Cryptology - CRYPTO 2010, LNCS*, 6223:80–97, 2010.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. *Advances in Cryptology - ASIACRYPT 2009, LNCS*, 5912:617–635, 2009.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. *Public Key Cryptography - PKC 2010, LNCS*, 6056:420–443, 2010.

[Ver12] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and Gitta Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012. Available at http://www-personal.umich.edu/ romanv/papers/non-asymptotic-rmt-plain.pdf.