A New Design of Online/Offline Signatures Based on Lattice

Mingmei Zheng[1], Shao-Jun Yang[1(✉)], Wei Wu[1], Jun Shao[2], and Xinyi Huang[1]

[1] Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Informatics, Fujian Normal University, Fuzhou, China
mmingzheng@outlook.com, shao-junyang@outlook.com, {weiwu,xyhuang}@fjnu.edu.cn

[2] School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou, China
chn.junshao@gmail.com

Abstract. With the rapid development of mobile internet, a large number of lightweight devices are widely used. Therefore, lightweight cryptographic primitives are urgently demanded. Among these primitives, online/offline signatures are one of the most promising one. Motivated by this situation, we propose a lattice-based online/offline signature scheme by using the hash-sign-switch paradigm, which was introduced by Shamir and Tauman in 2001. Our scheme not only has the advantages of online/offline signatures, but also can resist quantum computer attacks. The scheme we propose is built on several techniques, such as cover-free sets and programmable hash functions. Furthermore, we design a specific chameleon hash function, which plays an important role in the hash-sign-switch paradigm. Under the Inhomogeneous Small Integer Solution (ISIS) assumption, we prove that our proposed chameleon hash function is collision-resistant, which makes a direct application of this new design. In particular, our method satisfies existential unforgeability against adaptive chosen message attacks in the standard model.

Keywords: Online/offline signature · Lattice
Chameleon hash function
The Inhomogeneous Small Integer Solution (ISIS) assumption

1 Introduction

As one of fundamental cryptographic primitives, digital signatures are the essential inventions of modern cryptography. Informally, a signer Alice establishs a public key vk while keeping a secret key sk to herself. In addition, sk and pk satisfy a certain mathematical relation. The signer Alice signs a message M using sk and obtains a digital signature σ of M. Anyone, with pk, can verify the validity of the message-signature pair (M, σ). A digital signature scheme is said to be secure if it is existentially unforgettable against adaptive chosen message

C. Su and H. Kikuchi (Eds.): ISPEC 2018, LNCS 11125, pp. 198–212, 2018.
https://doi.org/10.1007/978-3-319-99807-7_12

attacks [10]. Digital signatures are useful in e-contract signing, document notarizing, authentication, and many other scenarios with the need of data integrity check and undeniability guaranty (e.g., [3,10]). In addition, digital signatures are the essential building blocks of more advanced cryptographic schemes, such as fair exchange and authenticated data redaction (e.g., [5]).

Digital signature schemes are often built on mathematical operations, like modular exponentiation, scalar multiplication and bilinear mapping, etc. However, these operations are much too heavy for smart cards, mobile devices, FRID tags and other resource-constrained devices. For those devices with more power, it would also be a critical issue when a large number of messages must be signed within a short period of time.

As a result, a lot of approaches have been proposed to improve the efficiency of digital signatures, e.g., online/offline signatures. Online/Offline signatures speed up signature production by dividing the signing process into two phases, offline and online. Most costly computations are completed in the offline phase, when the messages to be signed are unknown and the device is idle. Such pre-computation enables the online phase to quickly sign the messages with only light computation. The notion of online/offline signatures was introduced by Even, Goldreich and Micali [8] in 1989. Their design philosophy of online/offline signatures is using the one-time signatures for the online phase, which are very fast, and an ordinary signature scheme is used at the offline phase. Motivated by the design in [8], Shamir and Tauman [16] use chameleon hash functions to develop a new paradigm called hash-sign-switch, which can convert any signature scheme into a highly efficient online/offline signature scheme. From then on, there are many results (e.g., [4,14,18]) adopting hash-sign-switch paradigm to construct online/offline signature schemes.

The security of most existing online/offline signature schemes is based on traditional number-theoretic assumptions (e.g., [8,11,14]) and they are in danger of being broken with the rapid development of quantum computing technology. Therefore, it is urgent to design online/offline signature schemes that can resist quantum computer attacks. To the best of our knowledge, little attention has been paid on anti-quantum online/offline signatures (e.g., [18,19]). In the following section, we shall present a brief review of the related work.

1.1 Related Work

Even, Goldreich and Micali [8] proposed the notion of online/offline signatures in 1989. They used a general method to convert any signature scheme into an online/offline signature scheme. In their work, if the length of M is k, then the length of σ is a quadratic polynomial in k. To further improve the efficiency, Shamir and Tauman [16] proposed a hash-sign-switch paradigm. In [16], the overhead of the signature is reduced to an additive factor of k.

Many online/offline signature schemes with different properties have been proposed, such as threshold online/offline signature schemes (e.g., [4]) and identity-based online/offline signature schemes (e.g., [14]). Nevertheless, almost

all of previous online/offline signature schemes are based on traditional number-theoretic assumptions, such as DLP and IF (e.g., [4,8,12,16]). There is a risk that these assumptions would be broken with the use of quantum computing technology. Therefore, it is necessary to design anti-quantum online/offline signature schemes. However, there are few results on anti-quantum online/offline signatures (e.g., [18,19]). Driven by the design philosophy raised by Xiang [18] and Zhang [20], we present a lattice-based online/offline signature scheme. Although the idea of [18] is enlightening, the chameleon hash function needs more rigorous proof and the correctness of some details in his scheme needs further discussion.

1.2 Our Contributions

Lightweight cryptographic primitives are widely demanded as the widespread use of lightweight devices. Online/offline signatures are one of the promising solutions for this dilemma. This makes it highly non-trivial to propose a lattice-based online/offline signature scheme, which not only has the advantages of online/offline signature schemes, but also can resist quantum computer attacks.

Compared to previous work [18], our proposed chameleon hash function includes rigorous proof and specific data. Furthermore, by applying our chameleon hash function to the original scheme [20], the new scheme is more efficient than the original one in the offline phase. The security of our scheme can be reduced to the Inhomogeneous Small Integer Solution (ISIS) assumption in the standard model.

1.3 Roadmap

After some preliminaries in Sect. 2, we give a specific chameleon hash function in Sect. 3.1, which is a core technical in our scheme. We propose a lattice-based online/offline signature scheme in Sect. 3.2, and a short conclusion is given in Sect. 4.

2 Preliminaries

In this section, we mainly describe the notion of lattice-based programmable hash functions [20] and online/offline signatures [4].

2.1 Notation

We denote the real numbers and the integers by $\mathbb{R}$ and $\mathbb{Z}$, respectively. For any positive integer N, we let $[N] = \{0, 1, \cdots, N-1\}$. For positive integer n, let the standard notation O, ω classify the growth functions, and we say that $f(n) = \widetilde{O}(g(n))$ if $f(n) = O(g(n) \cdot \log^c n)$ for some fixed constant c. We use $poly(n)$ to denote the function $f(n) = O(n^c)$ for some constant c. A negligible function, denoted usually by $negl(n)$, is $f(n)$ such that $f(n) = o(n^{-c})$ for every fixed constant c. A probability is said to be overwhelming if it is $1 - negl(n)$.

The natural security parameter is κ throughout the paper, and all other quantities are implicitly functions of κ. The notation of $\leftarrow_r$ indicates randomly choosing elements from the distribution. Let $\mathbf{I}_n$ be the $n \times n$ identity matrix. Vectors are accustomed to being in column form and wrote by bold lower-case letters, e.g. $\mathbf{x}$. Matrices are used to be bold capital letters, e.g. $\mathbf{X}$. The notation of $(\mathbf{X}\|\mathbf{Y}) \in \mathbb{R}^{n\times(m+m')}$ means that the columns of $\mathbf{X} \in \mathbb{R}^{n\times m}$ are followed by the columns of $\mathbf{Y} \in \mathbb{R}^{n\times m'}$. The length of a matrix is denoted as the norm of its longest column, i.e., $\|\mathbf{X}\| = \max_i \|\mathbf{x}_i\|$. The largest singular value of matrix $\mathbf{X}$ is measured by $s_1(\mathbf{X}) = \max_{\mathbf{t}} \|\mathbf{X}\mathbf{t}\|$, where $\mathbf{t}$ is the unit vector. A hash function $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is an encoding with full-rank difference (FRD) [20] if it satisfies the following two conditions: (1) for any $\mathbf{u} \neq \mathbf{v}$, the matrix $H(\mathbf{u}) \pm H(\mathbf{v}) = H(\mathbf{u} \pm \mathbf{v}) \in \mathbb{Z}_q^{n\times n}$ is invertible; and (2) H is computable in polynomial time in $n \log q$. In particular, for any vector $\mathbf{v} = (v, 0, \cdots, 0)^\top$, we have that $H(\mathbf{v}) = v\mathbf{I}_n$.

2.2 Lattices

We now introduce the definition of lattice and its related parameters. Formally, given m linearly independent vectors $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m) \in \mathbb{R}^{m\times m}$, the m-dimensional full-rank lattice generated by $\mathbf{B}$ is defined as $\Lambda = L(\mathbf{B}) = \{\sum_{i=1}^m x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$. For any $\mathbf{x} \in \mathbb{R}^m$, the Gaussian function $\rho_{s,\mathbf{c}}$ on $\mathbb{R}^m$ is defined as $\rho_{s,\mathbf{c}}(\mathbf{x}) = \exp(-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2)$ with center $\mathbf{c} \in \mathbb{R}^m$ and $s > 0$. We have $\rho_{s,\mathbf{c}}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{s,\mathbf{c}}(\mathbf{x})$. For any $\mathbf{c} \in \mathbb{R}^m$, real $s > 0$ and $\mathbf{x} \in \Lambda$, the discrete Gaussian distribution $D_{\Lambda,s,\mathbf{c}}$ over Λ is denoted as $D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \rho_{s,\mathbf{c}}(\mathbf{x})/\rho_{s,\mathbf{c}}(\Lambda)$. For some positive $m, n, q \in \mathbb{Z}$, let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix and considering the following two lattices: $\Lambda^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = 0 \mod q\}$ and $\Lambda(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \exists\ \mathbf{s} \in \mathbb{Z}_q^n\ s.t.\ \mathbf{z} = \mathbf{A}^\top\mathbf{s} \mod q\}$. For any $\mathbf{u} \in \mathbb{Z}^n$ admitting an solution to $\mathbf{A}\mathbf{x} = \mathbf{u} \mod q$, we have the coset $\Lambda_{\mathbf{u}}^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{u} \mod q\} = \Lambda^\perp(\mathbf{A}) + \mathbf{x}$.

The following result was quoted from [20], and it will be used in Sect. 3.

Lemma 1. *For any positive integer $m \in \mathbb{Z}$, vector $\boldsymbol{y} \in \mathbb{Z}^m$ and large enough $s \geq \omega(\sqrt{\log m})$, we have that*

$$\Pr_{\boldsymbol{x}\leftarrow_r D_{\mathbb{Z}^m,s}}[\|\boldsymbol{x}\| > s\sqrt{m}] \leq 2^{-m} \text{ and } \Pr_{\boldsymbol{x}\leftarrow_r D_{\mathbb{Z}^m,s}}[\boldsymbol{x} = \boldsymbol{y}] \leq 2^{1-m}.$$

Following [6,15], we say that a random variable $\mathbf{X}$ over $\mathbb{R}$ is subgaussian with parameter $s > 0$ if the moment-generating function satisfies $E[\exp(2\pi t\mathbf{X})] \leq \exp(\pi s^2 t^2)$ for all $t \in \mathbb{R}$,. If $\mathbf{X}$ is subgaussian, then its tails are dominated by a Gaussian with parameter s, i.e., $\Pr[|\mathbf{X}| \geq t] \leq 2\exp(-\pi t^2/s^2)$ for all $t \geq 0$. In addition, we get that a random matrix $\mathbf{X}$ is subgaussian with parameter s if all its one-dimensional marginals $\mathbf{u}^\top\mathbf{X}\mathbf{v}$ for unit vectors $\mathbf{u}$, $\mathbf{v}$ are subgaussian with parameter s. Moreover, we have that for any lattices $\Lambda \subset \mathbb{R}^m$ and $s > 0$, the distribution $D_{\Lambda,s}$ is subgaussian with parameter s.

We have the following results from the non-asymptotic theory of random matrices [17], and it gives the singular value of variable $\mathbf{X}$ exactly.

Lemma 2. *Let* $\boldsymbol{X} \in \mathbb{Z}_q^{m\times n}$ *be a subgaussian random matrix with parameter s. There exists an universal constant* $C \approx 1/\sqrt{2\pi}$ *such that for any* $t \geq 0$*, we have* $s_1(\boldsymbol{X}) \leq C \cdot s \cdot (\sqrt{m} + \sqrt{n} + t)$ *except with probability at most* $2\exp(-\pi t^2)$*.*

In 1999, Ajtai [2] proposed the first trapdoor generation algorithm to output an approximately uniform trapdoor matrix A that allows to efficiently sample short vectors in $\Lambda^{\perp}(A)$. Then this trapdoor generation algorithm has been improved in [15]. We now recall the publicly trapdoor matrix $\mathbf{G}$ in [15]. Formally, for any prime $q > 2$, integer $n \geq 1$, $k = \lceil \log_2 q \rceil$, and $\mathbf{g} = (1, 2, 4, \cdots, 2^{k-1})^{\top} \in \mathbb{Z}_q^k$, we have that the public trapdoor matrix $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}^{\top} \in \mathbb{Z}_q^{n\times nk}$, where '$\otimes$' denotes the tensor product.

We show the formal definition of $\mathbf{G}$-trapdoor [15] in the following and it will be used in Sect. 3.

Definition 1. *Let* $\boldsymbol{A} \in \mathbb{Z}_q^{n\times \bar{m}}$ *and* $\boldsymbol{G} \in \mathbb{Z}_q^{n\times nk}$ *be matrices with* $n, q, \bar{m} \in \mathbb{Z}$ *and* $k = \lceil \log_2 q \rceil$*. A* $\boldsymbol{G}$*-Trapdoor for* $\boldsymbol{A}$ *is a martix* $\boldsymbol{R} \in \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$ *such that* $\boldsymbol{A}\begin{pmatrix}\boldsymbol{R}\\ \boldsymbol{I}\end{pmatrix} = \boldsymbol{SG}$ *for some invertible matrix* $\boldsymbol{S} \in \mathbb{Z}_q^{n\times n}$*. The quality of the trapdoor is measured by its largest singular value* $s_1(\boldsymbol{R})$*.*

If $\mathbf{R}$ is a trapdoor for $\mathbf{A}$, then it can be made into an equally good trapdoor for any extension $(\mathbf{A}\|\mathbf{B})$ by padding $\mathbf{R}$ with zero rows. This leaves $s_1(\mathbf{R})$ unchanged.

Then we refer to [15] for a detailed description of the sampling algorithm, which plays an important role in our scheme in Sect. 3.

Theorem 1. *For any integer* $n \geq 1$*,* $q > 0$*,* $k = \lceil \log_2 q \rceil$*, sufficiently large* $\bar{m} = O(n \log q)$ *and some invertible tag* $\boldsymbol{S} \in \mathbb{Z}_q^{n\times n}$*, there is a polynomial time algorithm* $\mathrm{TrapGen}(1^n, 1^{\bar{m}}, q, \boldsymbol{S})$ *that outputs a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{n\times \bar{m}}$ *and a* $\boldsymbol{G}$*-trapdoor* $\boldsymbol{R} \in \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$ *with quality* $s_1(\boldsymbol{R}) \leq \sqrt{\bar{m}} \cdot \omega(\sqrt{\log n})$ *such that the distribution of* $\boldsymbol{A}$ *is* $negl(\kappa)$*-far from uniform. Moreover, given any* $\boldsymbol{u} \in \mathbb{Z}_q^n$ *and real* $s > s_1(\boldsymbol{R}) \cdot \omega(\sqrt{\log n})$*, there is an efficient algorithm* $\mathrm{SampleD}(\boldsymbol{R}, \boldsymbol{A}, \boldsymbol{S}, \boldsymbol{u}, s)$ *samples from a distribution within* $negl(\kappa)$ *statistical distance of* $D_{\Lambda_{\boldsymbol{u}}^{\perp}(\boldsymbol{A}),s}$*.*

The following lemma illustrates that the matrix we construct is statistically close to the uniform, which is applied to Theorem 4.

Lemma 3. *For any postive* $n \geq 1$*,* $q > 2$*, sufficiently large* $\bar{m} = O(n \log q)$ *and real* $s \geq \omega(\sqrt{\log \bar{m}})$*, we have that the distribution of* $\boldsymbol{u} = \boldsymbol{Ae} \mod q$ *is statistically close to uniform over* $\mathbb{Z}_q^n$*, where* $\boldsymbol{e}$ *is randomly sampled from* $D_{\mathbb{Z}^{\bar{m}},s}$ *and* $\boldsymbol{A}$ *is a uniformly random matrix over* $\mathbb{Z}_q^{n\times \bar{m}}$*.*

The Inhomogeneous Small Integer Solution($ISIS_{q,\bar{m},\bar{\beta}}$) problem was first raised by Ajtai [1]. The *ISIS* problem was an inhomogeneous variant of *SIS*, which is asked to find a short nonzero integer solution $\mathbf{e} \in \mathbb{Z}^{\bar{m}}$ to the homogeneous linear system $\mathbf{Ae} = \mathbf{0} \mod q$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times \bar{m}}$. If we set $n, q \in \mathbb{Z}$ be some polynomials in the security parameter κ, $\bar{m} = O(n \log q)$, then $\bar{\beta}$ in the $ISIS_{q,\bar{m},\bar{\beta}}$ problem can be $\widetilde{O}(n^{5.5})$ according to [20]. Both hard problems on lattices are shown in detail in [9].

Definition 2. *The Inhomogeneous Small Integer Solution (ISIS) problem (in the ℓ_2 norm) is as follows: given an integer q, a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times\bar{m}}$, a syndrone $\boldsymbol{u} \in \mathbb{Z}_q^n$ and a real $\bar{\beta}$, find a integer vector $\boldsymbol{e} \in \mathbb{Z}^{\bar{m}}$ such that $\boldsymbol{A}\boldsymbol{e} = \boldsymbol{u} \mod q$ and $\|\boldsymbol{e}\|_2 \leq \bar{\beta}$.*

2.3 Lattice-Based Programmable Hash Function

We use lattice-based PHFs to construct our signature and lattice-based PHFs was proposed by [20] in 2016. Formally, let $m, \bar{m}, n, \ell, q, u, v \in \mathbb{Z}$ be some polynomials in the security parameter κ. We denote $\mathcal{I}_n$ as the set of invertible matrices in $\mathbb{Z}_q^{n\times n}$. A hash function $\mathcal{H} : \chi \to \mathbb{Z}_q^{n\times m}$ consists of two algorithms $(\mathcal{H}.Gen, \mathcal{H}.Eval)$, i.e., $K \leftarrow \mathcal{H}.Gen(1^\kappa)$ and $H_K(X) = \mathcal{H}.Eval(K, X)$ for any input $X \in \chi$. The following definition is referenced from [20].

Definition 3. (Lattice-Based Programmable Hash Function)
A hash function $\mathcal{H} : \chi \to \mathbb{Z}_q^{n\times m}$ is a $(u, v, \beta, \gamma, \delta)$-PHF if there exist a PPT trapdoor key generation algorithm $\mathcal{H}.\mathrm{TrapGen}$ and an efficiently deterministic trapdoor evaluation algorithm $\mathcal{H}.\mathrm{TrapEval}$ such that given a uniformly random $\boldsymbol{A} \in \mathbb{Z}_q^{n\times\bar{m}}$ and a public trapdoor matrix $\boldsymbol{B} \in \mathbb{Z}_q^{n\times m}$, the following properties hold:

***Syntax**: The PPT algorithm $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \boldsymbol{A}, \boldsymbol{B})$ outputs a key K' with a trapdoor td. Besides, for any input $X \in \chi$, the deterministic algorithm $(\boldsymbol{R}_X, \boldsymbol{S}_X) \leftarrow \mathcal{H}.\mathrm{TrapEval}\,(td, K', X)$ outputs $\boldsymbol{R}_X \in \mathbb{Z}_q^{\bar{m}\times m}$ and $\boldsymbol{S}_X \in \mathbb{Z}_q^{n\times n}$ such that $s_1(\boldsymbol{R}_X) \leq \beta$ and $\boldsymbol{S}_X \in \mathcal{I}_n \cup \{\boldsymbol{0}\}$.*

***Correctness**: For all possible $(K', td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa, \boldsymbol{A}, \boldsymbol{B})$, all $X \in \chi$ and its corresponding $(\boldsymbol{R}_X, \boldsymbol{S}_X) \leftarrow \mathcal{H}.\mathrm{TrapEval}(td, K', X)$, we have $H_{K'}(X) = \mathcal{H}.Eval(K', X) = \boldsymbol{A}\boldsymbol{R}_X + \boldsymbol{S}_X\boldsymbol{B}$.*

***Statistically close trapdoor keys**: For all $(K',\ td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa,\ \boldsymbol{A},\ \boldsymbol{B})$ and $K \leftarrow \mathcal{H}.Gen(1^\kappa)$, the statistical distance between $(\boldsymbol{A}, K')$ and $(\boldsymbol{A}, K)$ is at most γ.*

***Well-distributed hidden matrices**: Let all $(K',\ td) \leftarrow \mathcal{H}.\mathrm{TrapGen}(1^\kappa,\ \boldsymbol{A},\ \boldsymbol{B})$ and any inputs $X_1, \cdots, X_u, Y_1, \cdots, Y_v \in \chi$ enjoys $X_i \neq Y_j$ for any i, j. For $(\boldsymbol{R}_{X_i}, \boldsymbol{S}_{X_i}) \leftarrow \mathcal{H}.\mathrm{TrapEval}(td,\ K',\ X_i)$ and $(\boldsymbol{R}_{Y_j}, \boldsymbol{S}_{Y_j}) \leftarrow \mathcal{H}.\mathrm{TrapEval}(td, K', Y_j)$, we have that*

$$\Pr[\boldsymbol{S}_{X_1} = \cdots = \boldsymbol{S}_{X_u} = \{\boldsymbol{0}\} \wedge \boldsymbol{S}_{Y_1}, \cdots, \boldsymbol{S}_{Y_v} \in \mathcal{I}_n] \geq \delta.$$

If γ is negligible and $\delta > 0$ is noticeable, we simply say that $\mathcal{H}$ is a (u, v, β)-PHF.

A general trapdoor matrix $\mathbf{B}$ is used for utmost generality, but the publicly known trapdoor matrix $\mathbf{B} = \mathbf{G}$ in [15] is regarded for both efficiency and simplicity. In this paper, we apply two types of lattice-based programmable hash function constructions to our scheme. Then, we show their definitions and examples from [20].

Definition 4. [Type-1]
Let $\ell, n, m, q \in \mathbb{Z}$ be some polynomials in the security parameter κ. Let E be a deterministic encoding from χ to $(\mathbb{Z}_q^{n\times n})^\ell$. Then the hash function $\mathcal{H} = (\mathcal{H}.Gen, \mathcal{H}.Eval)$ with key space $\mathcal{K} \subseteq (\mathbb{Z}_q^{n\times m})^{\ell+1}$ is defined as follows:

- $\mathcal{H}.Gen(1^\kappa)$*: Randomly choose* $(\boldsymbol{A}_0, \cdots, \boldsymbol{A}_\ell) \leftarrow_r \mathcal{K}$, *return* $K = \{\boldsymbol{A}_i\}_{i\in\{0,\cdots,\ell\}}$.
- $\mathcal{H}.Eval(K, X)$*: Let* $E(X) = (\boldsymbol{C}_1, \cdots, \boldsymbol{C}_\ell)$, *return* $\boldsymbol{Z} = \boldsymbol{A}_0 + \sum_{i=1}^{\ell} \boldsymbol{C}_i \boldsymbol{A}_i$.

In the following theorem, we show several examples of Type-1 PHF [20], which were implicated proved in [3,15].

Theorem 2. *For large enough $\bar{m} = O(n \log q)$, the hash function $\mathcal{H}$ given in Definition 4 is a weak $(1, poly(v), \beta, \gamma, \delta)$-PHF with $\beta \leq \sqrt{\ell \bar{m}} \cdot \omega(\sqrt{\log n})$, $\gamma = negl(\kappa)$, and $\delta = 1$ when instantiated as follows:*

- *Let $\mathcal{K} \subseteq (\mathbb{Z}_q^{n\times m})^2$ and $\chi = \mathbb{Z}_q^n$. Given an input $X \in \chi$, the encoding $E(X)$ returns $H(X)$ where $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is an FRD encoding.*
- *Let $\mathcal{K} \subseteq (\mathbb{Z}_q^{n\times m})^{\ell+1}$ and $\chi = \{0,1\}^\ell$. Given an input $X \in (X_1, \cdots, X_\ell) \in \chi$, the encoding $E(X)$ returns $\boldsymbol{C}_i = X_i \cdot \boldsymbol{I}_n$ for all $i \in \{1, \cdots, \ell\}$.*

The two instantiations in Theorem 2 are weak $(1, v, \beta)$-PHFs for some polynomials $v \in \mathbb{Z}$ and real $\beta \in \mathbb{R}$.

Definition 5. [Type-2]
Let $n, q \in \mathbb{Z}$ be some polynomials in the security parameter κ. For any $\ell, v \in \mathbb{Z}$ and $L = 2^\ell$, let $N \leq 16v^2\ell$, $\eta \leq 4v\ell$ and $CF = \{CF_X\}_{X\in[L]}$ be an η-uniform, v-cover-free set. Let $\tau = \lceil \log_2 N \rceil$ and $k = \lceil \log_2 q \rceil$. Then the hash function $\mathcal{H} = (\mathcal{H}.Gen, \mathcal{H}.Eval)$ from $[L]$ to $\mathbb{Z}_q^{n\times nk}$ is defined as follows:

- $\mathcal{H}.Gen(1^\kappa)$*: Randomly choose* $\hat{\boldsymbol{A}}, \boldsymbol{A}_i \leftarrow_r \mathbb{Z}_q^{n\times nk}$ *for* $i \in \{0, \cdots, \tau-1\}$, *return the key* $K = (\hat{\boldsymbol{A}}, \{\boldsymbol{A}_i\}_{i\in\{0,\cdots,\tau\}})$.
- $\mathcal{H}.Eval(K, X)$*: Given* $K = (\hat{\boldsymbol{A}}, \{\boldsymbol{A}_i\}_{i\in\{0,\cdots,\tau-1\}})$ *and integer* $X \in [L]$, *the algorithm outputs* $\boldsymbol{Z} = H_K(X)$.

Please refer to [20] for details of the algorithm in Definition 5 . In the following, we show that the hash function $\mathcal{H}$ given in Definition 5 is a $(1, v, \beta)$-PHFs for some real $\beta \in \mathbb{R}$ and $v = poly(\kappa)$.

Theorem 3. *Let $CF = \{CF_X\}_{X\in[L]}$ be an η-uniform, v-cover-free set. For any $n, q \in \mathbb{Z}$, $L = 2^\ell$, $N \leq 16v^2\ell$, $\eta \leq 4v\ell$ and $\bar{m} = O(n \log q)$, the hash function $\mathcal{H}$ given in Definition 5 is a $(1, poly(v), \beta, \gamma, \delta)$-PHF with $\beta \leq \mu v\ell \bar{m}^{1.5} \cdot \omega(\sqrt{\log \bar{m}})$, $\gamma = negl(\kappa)$, and $\delta = 1/N$, where $\tau = \lceil \log_2 N \rceil$. In particular, if we set $\ell = n$ and $v = \omega(\log n)$, then $\beta = \tilde{O}(n^{2.5})$ and the key of $\mathcal{H}$ only consists of $\tau = O(\log n)$ matrices.*

The detailed proof of this theorem has been shown in [20]. Let L, N be some polynomials in the security parameter κ and let $CF = \{CF_X\}_{X\in[L]}$ be a family of subsets of $[N]$. The family CF is said to be v-cover-free [7,13,20] over $[N]$ if for any subset $S \subseteq [L]$ of size at most v, then the union $\cup_{X\in S} CF_X$ does not cover CF_Y for all $Y \notin S$. In addition, we say that CF is η-uniform if every subset CF_X in the union family $CF = \{CF_X\}_{X\in L}$ have size $\eta \in \mathbb{Z}$. $CF = \{CF_X\}_{X\in[L]}$ is regarded as an η-uniform, v-cover-free set when mentioned in this paper. A hash function $\mathcal{H} : \chi \to \mathbb{Z}_q^{n\times m}$ can be a weak (u, v, β)-PHF, where the algorithm $\mathcal{H}.TrapGen$ additionally takes a list $X_1, \cdots, X_u \in \chi$ as inputs such that the well-distributed hidden matrices property holds.

2.4 Definition and Security Model of Online/Offline Signatures

First of all, we roughly introduce the notion of online/offline signatures defined in [16], and then introduce the security model of online/offline signatures. Shamir and Tauman use the hash-sign-switch paradigm to construct a highly efficient online/offline signature scheme, which combines any chameleon hash family (C, H) and any signature scheme (G, S, V) to get an online/offline signature scheme (G', S', V').

More specifically, let $(m_1, r_1) \in \mathcal{M} \times \mathcal{S}$ be randomly chosen. $\mathcal{M}$ is the message space, and $\mathcal{S}$ is some finite space. Generating a pair (sk, vk) of private key and public key, by applying G to the input 1^κ (where G is the key generation algorithm of the original scheme), and generating a pair (tk, hk) of private key and public key, by applying C to the input 1^κ (where C is the key generation algorithm of the chameleon hash family). $H = CH_{hk}$ is a family of randomized hash functions. In the offline phase, we run the signing algorithm S with the signing key sk to sign the message $CH_{hk}(m_1, r_1)$, and denote the output $S_{sk}(CH_{hk}(m_1, r_1))$ by σ^{off}. In the online phase, there exists a polynomial time algorithm that on inputs the pair (tk, hk), (m_1, r_1) and an actual message $m_2 \in \mathcal{M}$, then outputs a value $r_2 \in \mathcal{S}$ such that $CH_{hk}(m_1, r_1) = CH_{hk}(m_2, r_2)$. Denoting the output r_2 by σ^{on}, and sending $\sigma = (\sigma^{off}, \sigma^{on})$ as a signature of m_2. The verification algorithm V' verifies that $\sigma = (\sigma^{off}, \sigma^{on})$ is indeed a signature of the message m_2 with respect to the public key (vk, hk), and uses the algorithm V to check that σ^{off} is indeed a signature of the hash value $CH_{hk}(m_2, r_2)$ with vk.

The security notion for our online/offline signature scheme is existentially unforgeable under adaptative chosen message attacks (EUF-CMA), which says that any PPT attacker, after receiving valid signatures on a polynomial number of adaptively chosen messages, cannot produce a valid signature on a new message. Formally, the game between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$ is as follows.

KeyGen. The challenger $\mathcal{C}$ runs the key generation algorithm $\mathsf{KeyGen}(1^\kappa)$ and returns (sk, tk) as its private key, (vk, hk) as its public key. $\mathcal{C}$ gives the public key to the attacker $\mathcal{A}$, and keeps the private key.

Signing. The attacker $\mathcal{A}$ is allowed to ask for the signature of any fresh message $\mathbf{m}$. In the offline phase, the challenger $\mathcal{C}$ randomly chooses the information to compute a chameleon hash function, and then it uses sk to sign the chameleon hash function, which is regarded as offline signature message. Denote the result by σ^{off}. In the online phase, the challenger $\mathcal{C}$ uses tk and σ^{off} to sign the actual message $\mathbf{m}$, and return σ^{on} as its online signature. The challenger $\mathcal{C}$ sends the signature $\sigma = (\sigma^{off}, \sigma^{on})$ to the attacker $\mathcal{A}$. The attacker can repeat the query by any polynomial times.

Forge. The attacker $\mathcal{A}$ outputs a message-signature pair $(\mathbf{m}^*, \sigma^*)$. Let Q be the messages set required by $\mathcal{A}$ in the signing phase. If $\mathbf{m}^* \notin Q$ and $\mathsf{Verify}(vk, hk, \mathbf{m}^*, \sigma^*) = 1$, the game outputs 1, else outputs 0.

If the game outputs 1, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ in the above security game is defined as $\mathsf{Adv}^{euf-cma}_{\mathcal{A},\mathcal{SIG}}(1^k) = \Pr[\mathcal{C}\,\text{outputs}\,1]$.

Definition 6. *Let κ be the security parameter. A signature scheme $\mathcal{SIG}$ is said to be existentially unforgeable under adaptative chosen message attacks (EUF-CMA) if the advantage $\mathsf{Adv}^{euf-cma}_{\mathcal{A},\mathcal{SIG}}(1^k)$ is negligible in κ for any PPT attacker $\mathcal{A}$.*

3 Our Design of Online/Offline Signatures

We now introduce a specific chameleon hash function before proposing our online/offline signature scheme.

3.1 Our Chameleon Hash Function

A chameleon hash function is a special type of hash functions, whose collision resistance depends on the user's state of knowledge. It has three properties, i.e., efficiency, collision resistance and trapdoor collisions. Every chameleon hash function is connected with a pair of public key and private key. For further details, please refer to [16]. In particular, the chameleon hash function in [18] is defined in the ideal lattice. Inspired by this, we designed a chameleon hash function on the general lattice.

Definition 7. *Let $n, w, q \in \mathbb{Z}$ with $n = wq$, $\bar{m} = O(n \log q)$, $k = \lceil \log q \rceil$ and $(\boldsymbol{m}, \boldsymbol{r}) \in \{0,1\}^w \times \mathbb{Z}_q^{\bar{m}}$. Let $\boldsymbol{A}' \in \mathbb{Z}_q^{w \times \bar{m}}$, $\boldsymbol{B}' \in \mathbb{Z}_q^{w \times w}$ be randomly chosen. Then we have a function $CH(\boldsymbol{A}', \boldsymbol{B}', \boldsymbol{m}, \boldsymbol{r}) = (\boldsymbol{B}'\boldsymbol{m} + \boldsymbol{A}'\boldsymbol{r}) \mod q$.*

Lemma 4. *Let $n, w, q \in \mathbb{Z}$ with $n = wq$, $\bar{m} = O(n \log q)$, $k = \lceil \log_2 q \rceil$, $\boldsymbol{B}' \in \mathbb{Z}_q^{w \times w}$, $\boldsymbol{G}' \in \mathbb{Z}_q^{w \times wk}$ and $(\boldsymbol{m}, \boldsymbol{r}) \in \{0,1\}^w \times \mathbb{Z}_q^{\bar{m}}$. Let $\boldsymbol{R}' \in \mathbb{Z}_q^{(\bar{m}-wk) \times wk}$ be $\boldsymbol{G}'$-trapdoor of $\boldsymbol{A}' \in \mathbb{Z}_q^{w \times \bar{m}}$ such that $\boldsymbol{A}' \begin{pmatrix} \boldsymbol{R}' \\ \boldsymbol{I}_{wk} \end{pmatrix} = \boldsymbol{S}'\boldsymbol{G}'$ for some invertible matrix $\boldsymbol{S}' \in \mathbb{Z}_q^{w \times w}$ and $s > s_1(\boldsymbol{R}') \cdot \omega(\sqrt{\log n})$. Then we have that $CH(\boldsymbol{A}', \boldsymbol{B}', \boldsymbol{m}, \boldsymbol{r})$ in Definition 7 is a chameleon hash function.*

Proof. Let ue show the function $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r})$ enjoys three properties of hash chameleon functions.

– **Efficiency.** The function $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r}) = (\mathbf{B}'\mathbf{m} + \mathbf{A}'\mathbf{r}) \mod q$ is computable in polynomial time.
– **Collision Resistance.** Let $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}_q^{\bar{m}}$ and $\mathbf{m}_1, \mathbf{m}_2 \in \{0,1\}^w$ such that $\mathbf{m}_1 \neq \mathbf{m}_2$. We note that finding a pair of collision in $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r})$ is at least as hard as solving the ISIS problem. Assuming that a collision of $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r})$ is $(\mathbf{m}_1, \mathbf{r}_1)$ and $(\mathbf{m}_2, \mathbf{r}_2)$, then we get $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}_1, \mathbf{r}_1) = CH(\mathbf{A}', \mathbf{B}', \mathbf{m}_2, \mathbf{r}_2)$. From this, we have

$$\mathbf{B}'(\mathbf{m}_1 - \mathbf{m}_2) + \mathbf{A}'(\mathbf{r}_1 - \mathbf{r}_2) = 0 \mod q. \tag{1}$$

We randomly choose $\bar{\mathbf{A}} \in \mathbb{Z}_q^{w \times (\bar{m} - wk)}$. Let $\mathbf{A}' = (\bar{\mathbf{A}} \| \mathbf{S}'\mathbf{G}' - \bar{\mathbf{A}}\mathbf{R}')$, $\mathbf{r}_{11}, \mathbf{r}_{21} \in \mathbb{Z}_q^{\bar{m}-wk}$, $\mathbf{r}_{12}, \mathbf{r}_{22} \in \mathbb{Z}_q^{wk}$, $\mathbf{r}_1 = \begin{pmatrix} \mathbf{r}_{11} \\ \mathbf{r}_{12} \end{pmatrix}$, $\mathbf{r}_2 = \begin{pmatrix} \mathbf{r}_{21} \\ \mathbf{r}_{22} \end{pmatrix}$ and $\mathbf{r}_1 - \mathbf{r}_2 = \begin{pmatrix} \mathbf{r}_{11} - \mathbf{r}_{21} \\ \mathbf{r}_{12} - \mathbf{r}_{22} \end{pmatrix}$. Applying this variables into (1), then (1) can be rewritten as

$$\mathbf{B}'(\mathbf{m}_1 - \mathbf{m}_2) + (\bar{\mathbf{A}} \| \mathbf{S}'\mathbf{G}' - \bar{\mathbf{A}}\mathbf{R}') \begin{pmatrix} \mathbf{r}_{11} - \mathbf{r}_{21} \\ \mathbf{r}_{12} - \mathbf{r}_{22} \end{pmatrix} = 0 \mod q. \tag{2}$$

By (2), we get

$$\mathbf{B}'(\mathbf{m}_1 - \mathbf{m}_2) + \bar{\mathbf{A}}[(\mathbf{r}_{11} - \mathbf{r}_{21}) + \mathbf{R}'(\mathbf{r}_{22} - \mathbf{r}_{12})] = \mathbf{S}'\mathbf{G}'(\mathbf{r}_{22} - \mathbf{r}_{12}) \mod q. \tag{3}$$

Let $\mathbf{A} = (\bar{\mathbf{A}} \| \mathbf{B}')$, $\mathbf{z}_1 = (\mathbf{r}_{11} - \mathbf{r}_{21}) + \mathbf{R}'(\mathbf{r}_{22} - \mathbf{r}_{12})$, $\mathbf{z}_2 = \mathbf{m}_1 - \mathbf{m}_2$, $\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}$ and $\mathbf{u} = \mathbf{S}'\mathbf{G}'(\mathbf{r}_{22} - \mathbf{r}_{12})$. The formula (3) can be rewritten as $\mathbf{A}\mathbf{z} = \mathbf{u}$. Since $\mathbf{m}_1 \neq \mathbf{m}_2$, we have $\|\mathbf{z}\| \neq 0$. Moreover, we get

$$\|\mathbf{z}\| = \sqrt{\mathbf{z}_1^2 + \mathbf{z}_2^2} \leq |\mathbf{z}_1| + |\mathbf{z}_2| \leq (q + q^2 kw)\sqrt{\bar{m} - wk} + \sqrt{w} \leq \widetilde{O}(n^{5.5}) = \bar{\beta}.$$

Therefore, $\mathbf{z}$ is a valid solution to the $ISIS_{q,\bar{m},\bar{\beta}}$ instance $(\mathbf{A}, \mathbf{u})$.
– **Trapdoor Collisions.** Let $\mathbf{r}_1 \in \mathbb{Z}_q^{\bar{m}}$ and $\mathbf{m}_1, \mathbf{m}_2 \in \{0,1\}^w$ such that $\mathbf{m}_1 \neq \mathbf{m}_2$. We can get $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}_1, \mathbf{r}_1) = (\mathbf{B}'\mathbf{m}_1 + \mathbf{A}'\mathbf{r}_1) \mod q$ and let $\mathbf{U} = (\mathbf{B}'\mathbf{m}_1 + \mathbf{A}'\mathbf{r}_1) - \mathbf{B}'\mathbf{m}_2$. Then compute $\mathbf{r}_2 \leftarrow \text{SampleD}(\mathbf{R}', \mathbf{A}', \mathbf{S}', \mathbf{U}, s)$. Therefore, we have $\mathbf{A}'\mathbf{r}_2 = \mathbf{U}$ by Theorem 1. From this, we have that there exists an efficient algorithm TrapCol that inputs $(\mathbf{A}', \mathbf{B}', \mathbf{R}', \mathbf{m}_1, \mathbf{r}_1, \mathbf{m}_2)$ and outputs a vector $\mathbf{r}_2 \in \mathbb{Z}_q^{\bar{m}}$ such that $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}_1, \mathbf{r}_1) = CH(\mathbf{A}', \mathbf{B}', \mathbf{m}_2, \mathbf{r}_2)$. By Theorem 1, we can easily get that $\mathbf{r}_2$ is computationally indistinguishable from uniform in $\mathbb{Z}_q^{\bar{m}}$.

Finally, we have proved that the function $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r})$ in Definition 7 is a chameleon hash function. We denote $CH(\mathbf{A}', \mathbf{B}', \mathbf{m}, \mathbf{r})$ by $CH_{hk}(\mathbf{m}, \mathbf{r})$, where $hk = (\mathbf{A}', \mathbf{B}')$ is its public key and $tk = \mathbf{R}'$ is its private key. □

3.2 Our Proposed Online/Offline Signature Scheme

Specifically, let $w, q \in \mathbb{Z}$ be some polynomials in the security parameter κ, and let $n = wq$, $\ell < n$, $\bar{m} = O(n \log q)$, $k = \lceil \log_2 q \rceil$, $m = \bar{m} + nk$, $s = \tilde{O}(n^{2.5}) \in \mathbb{R}$,

$\mathcal{M} = \{0,1\}^w$ and $\mathcal{S} = \mathbb{Z}_q^{\bar{m}}$. The construction of the offline phase involves in the weak PHF $\mathcal{H}'$ and the $(1, v, \beta)$-PHF, which are the first instantiated Type-1 PHF $\mathcal{H}'$ given in Theorem 2 and the Type-2 PHF $\mathcal{H} = (\mathcal{H}.Gen, \mathcal{H}.Eval)$ given in Definition 5 respectively. In particular, the weak PHF $\mathcal{H}'$ mapping from $\{0,1\}^\ell$ to $\mathbb{Z}_q^{n\times nk}$ has a form of $\mathcal{H}'_{K'}(\mathbf{t}) = \mathbf{A}_0 + H(\mathbf{t})\mathbf{G}$ where $K' = \mathbf{A}_0$. We are going to define our signature scheme $\mathcal{SIG} = (\mathbf{KeyGen}, \mathbf{Sign}, \mathbf{Verify})$.

KeyGen(1^κ)**.** Given a security parameter κ.

- Randomly choose $\mathbf{A}_0 \leftarrow_r \mathbb{Z}_q^{n\times nk}$,$\mathbf{u} \leftarrow_r \mathbb{Z}_q^n$, and let $\mathbf{S} \in \mathbb{Z}_q^{n\times n}$ be an invertible matrix. Then compute $(\mathbf{A},\mathbf{R}) \leftarrow \text{TrapGen}(1^n,1^{\bar{m}},\mathbf{S},q)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n\times\bar{m}}$, $\mathbf{R} \in \mathbb{Z}_q^{(\bar{m}-nk)\times nk}$ and $K \leftarrow \mathcal{H}.Gen(1^\kappa)$. Return (vk, sk)=$((\mathbf{A}, \mathbf{A}_0,K,\mathbf{u}), \mathbf{R})$.
- Randomly choose $\mathbf{B}' \leftarrow_r \mathbb{Z}_q^{w\times w}$, let $\mathbf{S}' \in \mathbb{Z}_q^{w\times w}$ be an invertible matrix. Then compute $(\mathbf{A}', \mathbf{R}') \leftarrow \text{TrapGen}(1^w, 1^{\bar{m}}, \mathbf{S}', q)$ such that $\mathbf{A}' \in \mathbb{Z}_q^{w\times\bar{m}}$, $\mathbf{R}' \in \mathbb{Z}_q^{(\bar{m}-wk)\times wk}$, and return (hk, tk)=$((\mathbf{A}', \mathbf{B}'), \mathbf{R}')$.

The private key is (sk, tk) and the public key is (vk, hk).

Sign$(sk, tk, \mathbf{m})$**.** Given a signing key (sk, tk), the signing algorithm operates as follows.

- Offline phase: Randomly choose $\mathbf{t} \leftarrow \{0,1\}^\ell$, $(\mathbf{m}_0, \mathbf{r}_0) \in \mathcal{M} \times \mathcal{S}$ and compute $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0) = (\mathbf{B}'\mathbf{m}_0 + \mathbf{A}'\mathbf{r}_0) \mod q$. Each component in $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)$ is represented in binary, and the binary digits are arranged in the order of $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)$. Let $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)_{(2)} \in \{0,1\}^n$ and $\mathbf{A}_{CH_{hk}(\mathbf{m}_0,\mathbf{r}_0)_{(2)},\mathbf{t}} = (\mathbf{A}\|\mathbf{A}_0 + H(\mathbf{t})\mathbf{G} + H_K(CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)_{(2)})) \in \mathbb{Z}_q^{n\times m}$ such that $H_K(CH_{hk}(\mathbf{m}_0,\mathbf{r}_0)_{(2)})) = \mathcal{H}.Eval(K, CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)_{(2)})) \in \mathbb{Z}_q^{n\times nk}$. Then compute $\mathbf{e} \leftarrow \text{SampleD}(\mathbf{R}, \mathbf{A}_{CH_{hk}(\mathbf{m}_0,\mathbf{r}_0)_{(2)},\mathbf{t}}, \mathbf{S}, \mathbf{u}, s)$, store $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)$ and the output of offline phase is $\sigma^{off} = (\mathbf{e}, \mathbf{t})$.
- Online phase: Given the message $\mathbf{m} \in \{0,1\}^w$, $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0)$ and σ^{off}, compute $\mathbf{r} = \text{TrapCol}\ (\mathbf{A}', \mathbf{B}', \mathbf{R}', \mathbf{m}_0, \mathbf{r}_0, \mathbf{m})$ and return $\sigma^{on} = \mathbf{r}$.

Finally, the signature of the message $\mathbf{m}$ is $\sigma = (\sigma^{off}, \sigma^{on})$.

Verify$(vk, hk, \mathbf{m}, \sigma)$**.** Given vk, hk, $\mathbf{m}$ and σ, compute $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0) = CH_{hk}(\mathbf{m}, \mathbf{r})$. Return 1 if $\|\mathbf{e}\| \le s\sqrt{m}$ and $\mathbf{A}_{CH_{hk}(\mathbf{m},\mathbf{r})_{(2)},\mathbf{t}} \cdot \mathbf{e} = \mathbf{u}$. Otherwise, return 0.

Correctness. From the third property of chameleon hash functions, we have $CH_{hk}(\mathbf{m}_0, \mathbf{r}_0) = CH_{hk}(\mathbf{m}, \mathbf{r})$. Since $\mathbf{R}$ is a $\mathbf{G}$-trapdoor of $\mathbf{A}$, it can be extended to a $\mathbf{G}$-trapdoor for $\mathbf{A}_{CH_{hk}(\mathbf{m},\mathbf{r})_{(2)},\mathbf{t}}$ by padding with zero rows with the same quality $s_1(\mathbf{R}) \le \sqrt{m} \cdot \omega(\sqrt{\log n})$. Since $s = \tilde{O}(n^{2.5}) > s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, the vector $\mathbf{e}$ produced by SampleD follows the distribution $D_{\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A}_{CH_{hk}(\mathbf{m},\mathbf{r})_{(2)},\mathbf{t}}),s}$ and has length at most $s\sqrt{m}$ with overwhelming probability by Lemma 1. In short, the signature σ is accepted by the verification algorithm.

Theorem 4. *Let $w, q, \bar{m} \in \mathbb{Z}$ be some polynomials in the security parameter κ, $n = wq$, $k = \lceil \log_2 q \rceil$, $\ell = O(\log n)$, $v = \omega(\log n)$ and $m = \bar{m} + nk$. If there exists a PPT attacker $\mathcal{A}$ against EUF-CMA security of $\mathcal{SIG}$ that makes at most*

$Q = poly(n)$ signing queries and succeeds with probability ϵ, then there exists an algorithm $\mathcal{B}$ solving the $ISIS_{q,\bar{m},\bar{\beta}}$ problem for $\bar{\beta} = \tilde{O}(n^{5.5})$ with probability at least $\epsilon' \geq \frac{\epsilon}{Q \cdot \tilde{O}(n)}$.

Proof. Assuming that there exists an attacker $\mathcal{A}$ forging the signature with probability ϵ, then we give the construction of the algorithm $\mathcal{B}$ solving $ISIS_{q,\bar{m},\bar{\beta}}$ problem with probability at least $\epsilon' \geq \frac{\epsilon}{Q \cdot \tilde{O}(n)}$. Formally, $\mathcal{B}$ randomly chooses a vector $\mathbf{t}' \leftarrow_r \{0,1\}^\ell$ and hopes that $\mathcal{A}$ will output a forgery signature with tag $\mathbf{t}^* = \mathbf{t}'$. Then, the algorithm $\mathcal{B}$ simulates the EUF-CMA game as follows:

KeyGen

- Given an $ISIS_{q,\bar{m},\bar{\beta}}$ challenge instance $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^n$, the algorithm $\mathcal{B}$ first randomly chooses $\mathbf{R}_0 \leftarrow_r (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$ and computes $\mathbf{A}_0 = \mathbf{A}\mathbf{R}_0 - H(0\|\mathbf{t}')\mathbf{G}$. This is done by running $(K', td) \leftarrow \mathcal{H}.TrapGen(1^\kappa, \mathbf{A}, \mathbf{G})$ as in Definition 5. Therefore, the algorithm $\mathcal{B}$ returns $vk = (\mathbf{A}, \mathbf{A}_0, K', \mathbf{u})$ and $sk = (\mathbf{R}_0, td)$.
- The algorithm $\mathcal{B}$ randomly chooses $\mathbf{B}' \in \mathbb{Z}_q^{w \times w}$, an invertible matrix $\mathbf{S}' \in \mathbb{Z}_q^{w \times w}$ and computes $(\mathbf{A}', \mathbf{R}') \leftarrow \mathrm{TrapGen}(1^w, 1^{\bar{m}}, \mathbf{S}', q)$. Then $\mathcal{B}$ returns $hk = (\mathbf{A}', \mathbf{B}')$ and $tk = \mathbf{R}'$.

Finally, the simulated public key is $(vk,\ hk)$, and the simulated private key is $(sk,\ tk)$. $(\mathbf{A}, \mathbf{u})$ is uniformly distributed over $\mathbb{Z}_q^{n \times \bar{m}} \times \mathbb{Z}_q^n$ by the definition of *ISIS*. Since $\mathbf{R}_0 \leftarrow_r (D_{\mathbb{Z}^{\bar{m}},\omega(\sqrt{\log n})})^{nk}$, by Lemma 3 the matrix $\mathbf{A}_0$ is statistically close to uniform over $\mathbb{Z}_q^{n \times nk}$. Moreover, the simulated key K' is statistically close to the real key. $\mathbf{A}' \in \mathbb{Z}_q^{w \times \bar{m}}$ is $negl(\kappa)$-far from uniform by Theorem 1. Thus, the distribution of the simulated verification key is statistically close to that of the real one.

Signing. The algorithm $\mathcal{B}$ accepts signing queries from attacker $\mathcal{A}$.

- Offline phase: The algorithm $\mathcal{B}$ first randomly chooses $(\mathbf{m}_0, \mathbf{t}_0) \in \mathcal{M} \times \mathcal{S}$, and computes $\mu = CH_{hk}(\mathbf{m}_0, \mathbf{t}_0)_{(2)} \in \{0,1\}^n$. Then, $\mathcal{B}$ first randomly chooses $\mathbf{t} \leftarrow_r \{0,1\}^\ell$. If $\mathbf{t}$ has been chosen in answering the signature more than v times, $\mathcal{B}$ aborts. Otherwise, compute $(\mathbf{R}_\mu, \mathbf{S}_\mu) \leftarrow \mathcal{H}.TrapEval(td, K', \mu)$ as in Definition 5. Then we have that $\mathbf{A}_{\mu,\mathbf{t}} = (\mathbf{A}\|(\mathbf{A}_0 + H(\mathbf{t})\mathbf{G}) + H_{K'}(\mu)) = (\mathbf{A}\|(\mathbf{A}(\mathbf{R}_0 + \mathbf{R}_\mu) + (H(0\|\mathbf{t}) - H(0\|\mathbf{t}') + \mathbf{S}_\mu)\mathbf{G})$. Since $\mathbf{S}_\mu = b\mathbf{I}_n = H(b\|0)$ for some $b \in \{-1, 0, 1\}$ by the proof of Theorem 3 (for details, please refer to [20]). Let $\hat{\mathbf{S}} = H(0\|\mathbf{t}) - H(0\|\mathbf{t}') + \mathbf{S}_\mu = H(b\|(\mathbf{t} - \mathbf{t}'))$ by the homomorphic property of the FRD in [20]. We split our analysis into two different cases:
  (1) If $\mathbf{t} \neq \mathbf{t}'$ or $\mathbf{t} = \mathbf{t}' \wedge b \neq 0$, we have that $\hat{\mathbf{S}}$ is invertible. Thus $\hat{\mathbf{R}} = -(\mathbf{R}_0 + \mathbf{R}_\mu)$ is a $\mathbf{G}$-trapdoor for $\mathbf{A}_{\mu,\mathbf{t}}$. Since $s_1(\mathbf{R}_0) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$ by Lemma 2 and $s_1(\mathbf{R}_\mu) \leq \tilde{O}(n^{2.5})$, we have $\hat{\mathbf{R}} \leq \tilde{O}(n^{2.5})$. Finally, compute $\mathbf{e} \leftarrow \mathrm{SampleD}(\hat{\mathbf{R}}, \mathbf{A}_{\mu,\mathbf{t}}, \hat{\mathbf{S}}, \mathbf{u}, s)$ and return the signature $\sigma^{off} = (\mathbf{e}, \mathbf{t})$.
  (2) If $\mathbf{t} = \mathbf{t}' \wedge b = 0$, we have that $\hat{\mathbf{S}} = H(b\|(\mathbf{t} - \mathbf{t}')) = 0$. Thus $\mathcal{B}$ aborts.
- Online phase: Given the message $\mathbf{m} \in \{0,1\}^w$, $CH_{hk}(\mathbf{m}_0, \mathbf{t}_0)$ and σ^{off}, the algorithm $\mathcal{B}$ computes $\mathbf{r} = \mathrm{TrapCol}(\mathbf{A}', \mathbf{B}', \mathbf{R}', \mathbf{m}_0, \mathbf{t}_0, \mathbf{m})$ and returns $\sigma^{on} = \mathbf{r}$.

Forge. After answering at most Q signature queries, the attacker $\mathcal{A}$ outputs a forgery $\sigma^* = ((\mathbf{e}^*, \mathbf{t}^*), \mathbf{r}^*)$ for some message $\mathbf{m}^* \in \{0,1\}^w$ satisfying $\|\mathbf{e}^*\| \leq s\sqrt{m}$ and $\mathbf{A}_{CH_{hk}(\mathbf{m}^*,\mathbf{t}^*)_{(2)},\mathbf{t}^*} \cdot \mathbf{e}^* = \mathbf{u}$, and we know that $\mathbf{A}_{CH_{hk}(\mathbf{m}^*,\mathbf{t}^*)_{(2)},\mathbf{t}^*} = (\mathbf{A}\|(\mathbf{A}_0 + H(0\|\mathbf{t}^*)\mathbf{G}) + H_K(CH_{hk}(\mathbf{m}^*, \mathbf{t}^*)_{(2)})) \in \mathbb{Z}_q^{n\times m}$. $\mathcal{B}$ computes that $(\mathbf{R}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}}, \mathbf{S}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}}) \leftarrow \mathcal{H}.TrapEval(td, K', CH_{hk}(\mathbf{m}^*, \mathbf{r}^*)_{(2)})$. Moreover, if $\mathbf{t}^* \neq \mathbf{t}'$ or $\mathbf{S}_{CH_{hk}}(\mathbf{m}^*, \mathbf{r}^*)_{(2)} \neq 0$, $\mathcal{B}$ aborts. Else, we have that $\mathbf{A}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)},\mathbf{t}^*} = (\mathbf{A}\|(\mathbf{A}(\mathbf{R}_0 + \mathbf{R}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}})) = \mathbf{A}(\mathbf{I}_{\bar{m}}\| - \hat{\mathbf{R}})$, where $\hat{\mathbf{R}} = \mathbf{R}_0 + \mathbf{R}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}}$. Let $\hat{\mathbf{e}}$ be $(\mathbf{I}_{\bar{m}}\| - \hat{\mathbf{R}})\mathbf{e}^*$. Since $s_1(\mathbf{R}_0) \leq \sqrt{m} \cdot \omega(\sqrt{\log n})$ by Lemma 2 and $s_1(\mathbf{R}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}}) \leq \beta = \tilde{O}(n^{2.5})$ by Theorem 3, we have $\|\hat{\mathbf{e}}\| \leq \tilde{O}(n^{2.5}) \cdot s\sqrt{m} = \tilde{O}(n^{5.5}) = \bar{\beta}$. Therefore, the algorithm $\mathcal{B}$ outputs $\hat{\mathbf{e}} = (\mathbf{I}_{\bar{m}}\| - \hat{\mathbf{R}})\mathbf{e}^*$ as the solution of $ISIS_{q,\bar{m},\bar{\beta}}$.

Since the algorithm $\mathcal{B}$ will receive at most $Q = poly(n)$ adaptive signing queries from the attacker $\mathcal{A}$. For each message, the algorithm $\mathcal{B}$ chooses a uniformly random tag $\mathbf{t}$. If some tag $\mathbf{t}$ is chosen for more than v times in the signing queries, $\mathcal{B}$ aborts. Let $\mathbf{m}_1, \cdots, \mathbf{m}_u$ be all the messages in the signing queries that $\mathcal{B}$ chooses the same tag $\mathbf{t} = \mathbf{t}'$ by accident. And corresponding to that, the algorithm $\mathcal{B}$ randomly selects $(\mathbf{a}_i, \mathbf{b}_i) \in \mathcal{M} \times \mathcal{S}$ for $i \in \{1, \cdots, u\}$. Let $(\mathbf{R}_{CH_{hk}(\mathbf{a}_i,\mathbf{b}_i)_{(2)}}, \mathbf{S}_{CH_{hk}(\mathbf{a}_i,\mathbf{b}_i)_{(2)}}) \leftarrow \mathcal{H}.TrapEval(td, K', CH_{hk}(\mathbf{a}_i, \mathbf{b}_i)_{(2)})$. If $\mathbf{S}_{CH_{hk}(\mathbf{a}_i,\mathbf{b}_i)_{(2)}}$ is not invertible, $\mathcal{B}$ aborts. Since $\ell = O(\log n)$, we have $\frac{Q}{2^\ell} \leq \frac{1}{2}$. Notice that the probability $\mathcal{B}$ uses any tag $\mathbf{t}$ in answering the signature queries over v times is less than $Q^2 \cdot (\frac{Q}{2^\ell})^v$ through a similar method in [11], which is negligible. Therefore, the possibility of using the same tag $\mathbf{t}$ in more than $u(\geq v)$ times signing queries is negligible. If $u < v$, the possibility that $\mathbf{S}_{CH_{hk}(\mathbf{a}_i,\mathbf{b}_i)_{(2)}}$ is invertible and $\mathbf{S}_{CH_{hk}(\mathbf{m}^*,\mathbf{r}^*)_{(2)}} = 0$ for all $i \in \{1, \cdots, u\}$ (using the fact that $CH_{hk}(\mathbf{m}^*, \mathbf{r}^*)_{(2)} \notin \{CH_{hk}(\mathbf{a}_i, \mathbf{b}_i)_{(2)}\}_{i\in\{1,\cdots,u\}}$) is at least $\delta = \frac{1}{16nv^2}$ - negl(κ) by Theorem 3. Then, we have $\Pr[\mathbf{t}^* = \mathbf{t}'] \geq \frac{1}{2^\ell}$. Therefore, the success probability of solving the $ISIS_{q,\bar{m},\bar{\beta}}$ instance is at least $\epsilon' = (\epsilon - Q^2 \cdot (\frac{Q}{2^\ell})^v) \cdot \delta \cdot (\frac{1}{2^\ell} - negl(\kappa)) = \frac{\epsilon}{Q \cdot \tilde{O}(n)}$. We conclude the proof. □

3.3 Comparison

In Table 1, we give a (rough) comparison with existing schemes in the standard model. Let $w, q \in \mathbb{Z}$ be some polynomials in the security parameter and let $n = wq$ be the message length. Q presents the number of signature queries made by the attacker. Real $\bar{\beta}$ denotes the parameter for (I)SIS problem. The reducation loss is the ratio ϵ/ϵ' between the success probability ϵ of the attacker and the success probability ϵ' of the reduction.

Compared with the existing lattice-based signature schemes [6,18,20], the length of public key and signature (online) of our proposed scheme is the same as theirs. Our work is driven by the idea of Xiang [18] and Zhang [20]. Due to pre-computation in the offline phase, our scheme is more efficient than the original scheme [20] in signature production. The work [18] is motivated by [6]. As shown in Table 1, signature generation in our scheme is faster than [6] in

the online phase. Compared to the work in [18], we have more rigorous proof and specific data in our proposed chameleon hash function. Furthermore, the parameters of Xiang's algorithm do not match those defined in [18].

Table 1. Comparison with existing schemes

Schemes	DM14 [6]	ZCZ16 [20]	Xiang17 [18]	Our $\mathcal{SIG}$
public key	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$
Signature (online)	1	1	1	1
Reduction loss	$(Q^2/\epsilon)^2$	$Q \cdot \tilde{O}(n)$	$(Q^2/\epsilon)^2$	$Q \cdot \tilde{O}(n)$
param $\bar{\beta}$	$\tilde{O}(n^{3.5})$	$\tilde{O}(n^{5.5})$	$\tilde{O}(n^2)$	$\tilde{O}(n^{5.5})$
Calculation (online)	$O(\log^2 n)$	$O(\log^2 n)$	$\tilde{O}(w^2)$	$\tilde{O}(w^2)$

4 Conclusion

In this paper, we present a new chameleon hash function, the security of which can be reduced to the Inhomogeneous Small Integer Solution (ISIS) assumption. The main technical of our proposed online/offline signature scheme is our chameleon hash function and the construction of PHFs in [20]. Moreover, the online signature of our scheme consists of a single lattice vector and the public key includes a logarithmic number of matrices. In addition, our scheme is proved to be existentially unforgeable against adaptive chosen message attacks (EUF-CMA) in the standard model.

Acknowledgements. The authors would like to thank anonymous reviewers for their helpful comments. This work is supported by National Natural Science Foundation of China (61472083, 61771140, 11701089, 61472364), Distinguished Young Scholars Fund of Fujian (2016J06013), Fujian Normal University Innovative Research Team (NO. IRTL1207), Fujian Province Department of Education Project (JOPX 15066), and Zhejiang Provincial Natural Science Foundation (NO. LZ18F020003).

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, 22–24 May 1996, pp. 99–108 (1996). https://doi.org/10.1145/237814.237838
2. Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_1
3. Boyen, X.: Lattice mixing and vanishing trapdoors: a framework for fully secure short signatures and more. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_29

4. Crutchfield, C., Molnar, D., Turner, D., Wagner, D.: Generic on-line/off-line threshold signatures. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 58–74. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_5
5. Deiseroth, B., Fehr, V., Fischlin, M., Maasz, M., Reimers, N.F., Stein, R.: Computing on authenticated data for adjustable predicates. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 53–68. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38980-1_4
6. Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 335–352. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_19
7. Erdös, P., Frankl, P., Füredi, Z.: Families of finite sets in which no set is covered by the union of r others. Isr. J. Math. **51**(1–2), 79–89 (1985)
8. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. J. Cryptol. **9**(1), 35–67 (1996). https://doi.org/10.1007/BF02254791
9. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, 17–20 May 2008, pp. 197–206 (2008). https://doi.org/10.1145/1374376.1374407
10. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988). https://doi.org/10.1137/0217017
11. Hofheinz, D., Kiltz, E.: Programmable hash functions and their applications. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 21–38. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_2
12. Krawczyk, H., Rabin, T.: Chameleon hashing and signatures. IACR Cryptology ePrint Archive 1998/10 (1998). http://eprint.iacr.org/1998/010
13. Kumar, R., Rajagopalan, S., Sahai, A.: Coding constructions for blacklisting problems without computational assumptions. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 609–623. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_38
14. Liu, J.K., Baek, J., Zhou, J.Y., Yang, Y.J., Wong, J.W.: Efficient online/offline identity-based signature for wireless sensor network. Int. J. Inf. Sec. **9**(4), 287–296 (2010). https://doi.org/10.1007/s10207-010-0109-y
15. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
16. Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_21
17. Vershynin, R.: Introduction to the non-asymptotic analysis of random matrices. CoRR abs/1011.3027 (2010). http://arxiv.org/abs/1011.3027
18. Xiang, X.Y.: Online/offline signature scheme based on ideal lattices (in Chinese). J. Cryptologic Res. **4**(3), 253–261 (2017)
19. Xiang, X.Y., Li, H.: Lattice-based online/offline signature scheme (in Chinese). J. Beijing Univ. Posts Telecommun. **38**(3), 117–120, 134 (2015)
20. Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: short signatures and ibes with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 303–332. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_11