Short Signatures from Homomorphic Trapdoor Functions

Jacob Alperin-Sheriff[*]

January 1, 2015

Abstract

We present a lattice-based stateless signature scheme provably secure in the standard model. Our scheme has a *constant* number of matrices in the public key and a single lattice vector (plus a tag) in the signatures. The best previous lattice-based encryption schemes were the scheme of Ducas and Micciancio (CRYPTO 2014), which required a logarithmic number of matrices in the public key and that of Bohl et. al (J. of Cryptology 2014), which required a logarithmic number of lattice vectors in the signature. Our main technique involves using fully homomorphic computation to compute a degree d polynomial over the tags hidden in the matrices in the public key. In the scheme of Ducas and Micciancio, only functions *linear* over the tags in the public key matrices were used, which necessitated having d matrices in the public key.

As a matter of independent interest, we extend Wichs' (eprint 2014) recent construction of homomorphic trapdoor functions into a primitive we call puncturable homomorphic trapdoor functions (PHTDFs). This primitive abstracts out most of the properties required in many different lattice-based cryptographic constructions. We then show how to combine a PHTDF along with a function satisfying certain properties (to be evaluated homomorphically) to give an eu-scma signature scheme.

[*] School of Computer Science, College of Computing, Georgia Institute of Technology. Email: `jmas6@cc.gatech.edu`

1 Introduction

Lattice-based cryptography has made great strides since the original work of Ajtai [AD97, Ajt96]. In many areas, it holds its own in comparison to cryptography based on various quantum-insecure number-theoretic problems such as RSA and Diffie-Hellman. Variants of the very efficient commercial `NTRUEncrypt` scheme have been proved secure under worst-case lattice problems [HPS98, SS11]. Moreover, lattice-based cryptography has led to cryptographic primitives such as fully homomorphic encryption (FHE) which have not been realized at all under classical number-theoretic hardness assumptions [Gen09].

In the area of digital signatures, lattice-based cryptography is also able to hold its own–in the random oracle model. The work in this area [Lyu09, Lyu12, GLP12, DDLL13, BG14] has led to signature schemes which are very efficient. Signing is about as fast as in state-of-the-art schemes providing comparable security (in the random oracle model) based on quantum-insecure problems, and verification is much faster. On the downside, the key and signatures are quite a bit larger.

The gap in key size is even more pronounced for stateless signatures in the standard model. In state-of-the-art schemes based on the security of standard number-theoretic assumptions as RSA and decisional Bilinear Diffie-Hellman (BDH), [Wat09, HW09] both the public key and signatures contain only a constant number of "basic" elements (elements of $\mathbb{Z}_N^*$ in the RSA scheme, group elements in the BDH-based scheme). This results in the public key and signatures having size *linear* ($O(\lambda)$) in the security parameter. By contrast, even when using "compact" algebraic lattices [Mic02], the "basic" elements in lattice-based cryptography (matrices of a certain size in $\mathbb{Z}_q$ for the public key, where q can usually be polynomial in λ, vectors of a certain size in $\mathbb{Z}_q$ for the signatures) have size quasilinear ($\tilde{O}(\lambda)$) in the security parameter. Moreover, the best known schemes require either a logarithmic number of "basic" elements in the public key [DM14] or a logarithmic number of "basic" elements in signatures [BHJ$^+$14].

The efficient RSA-based scheme cited above [HW09] uses the well-studied prefix-guessing technique for achieving static security in signature schemes. In this technique, the simulator samples from the polynomially-sized set of shortest prefixes not found in the messages received from the adversary and then sets up the scheme so that a forgery on a message with that prefix lets the simulator solve RSA. A short public key is achieved by embedding trapdoors for all prefixes of the messages received by the adversary into the public key. The efficient BDH-based scheme [Wat09] is a generic transformation from an efficient fully secure identity-based encryption scheme. The techniques used in the RSA scheme does not appear to have an easy realization in the lattice setting, and since the most efficient known lattice-based fully secure identity-based encryption scheme requires a linear number of "basic" elements in the public key [ABB10], it also does not appear to be a productive avenue of investigation for improving efficiency in signatures. Instead of attempting to use these techniques, we turn to an area where lattice-based cryptography has proven to be particularly versatile: homomorphic computation.

1.1 Our Results

We present the first standard model construction of a lattice-based stateless signature scheme with short signatures and a *constant* number of matrices in the public key. The constant number of matrices does come at a price, as our scheme requires a significantly larger SIS parameter β (and correspondingly, a larger modulus q) to achieve similar levels of security to previous schemes. See Figure 1.1 for a detailed comparison to previous works.

Our starting point is the recent scheme of Ducas and Micciancio [DM14], which combines the confined guessing technique of Böhl et al. [BHJ$^+$14] with the "vanishing trapdoors" technique of Boyen [Boy10]. We provide two improvents to the scheme:

Scheme	Pub. Key $R_q^{1\times k}$ mat.	Secret Key $R_q^{k\times k}$ mat.	Signature R_q^k vec.	Reduction loss	SIS param β
[LM08] (Trees)	1	1	$\log n$	Q	$\tilde{\Omega}(n^2)$
[CHKP10]	n	n	n	Q	$\tilde{\Omega}(n^{3/2})$
[Boy10, MP12]	n	n	1	Q	$\tilde{\Omega}(n^{7/2})$, $\tilde{\Omega}(n^{5/2})$
[BHJ+14]	1	1	d	$(Q^2/\epsilon)^c$	$\tilde{\Omega}(n^{5/2})$
[DM14]	d	1	1	$(Q^2/\epsilon)^c$	$\tilde{\Omega}(n^{7/2})$
This work	1	1	1	$(Q^2/\epsilon)^c$	$\tilde{\Omega}(d^{2d} \cdot n^{11/2})$

In the above table, we have ignored constant factors to avoid clutter. The comparison is in the ring setting because as written, [LM08, DM14] have no realization over general lattices. Here, $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$ for some cyclotomic polynomial f of degree n, the modulus $q \geq \beta\sqrt{n}\omega(\sqrt{\log n})$, and $k = O(\log q)$. Q is the number of signatures queries made by the adversary and ϵ is its success probability. For those schemes using the confined guessing technique, d is a value satisfying $2Q^2/\epsilon < 2^{\lfloor c^d \rfloor}$ for an arbitrary constant $c > 1$ (which governs a trade-off between public key size and the reduction loss). The reduction loss is the ratio ϵ'/ϵ between the success probability ϵ' of the reduction and the success probability ϵ of the adversary. In order to be secure against any PPT adversary succeeding with non-negligible probability, we need $d = \omega(\log \log n)$. For our scheme, we can choose d to be as large as $O(\log(n)/\log(\log(n)))$ and still end up with a polynomial-sized SIS parameter β. However, it does have the downside that, choosing $d = \log(n)$, as in previous works using confined guessing, results in a SIS parameter of size $n^{\tilde{\Omega}(\log \log n)}$.

Figure 1: Comparison to previous standard-model lattice-based signature schemes in the ring setting

Instantiation Over General Lattices. The Ducas-Micciancio scheme only works in the ring-based setting. The reason for this is that, as written, the scheme requires that their tags (represented as matrices) commute with a certain structured matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ under multiplication. Over general lattices, the only tags which commute with $\mathbf{G}$ are scalar tags, and there are only a (small) polynomial number of these tags which are "short", while a superpolynomial number are required for the construction. As a result, they were forced to use the ring setting, where their tag space can be represented as ring elements, which do in fact commute with $\mathbf{G}$.

To resolve this non-commutativity issue, we recall a technique which appears to have first been explicitly presented by [Xag13], although it appears earlier implicitly in an earlier work in the area of fully homomorphic encryption [BV11]. This technique involves a function which we denote (somewhat abusively) as $\mathbf{G}^{-1}(\mathbf{U})$, where the output of the function is a $\{0,1\}^{m\times m}$ matrix $\mathbf{X}$ such that $\mathbf{GX} = \mathbf{U}$. As a result, in order to multiply a tag $\mathbf{H}$ "commutatively" with $\mathbf{G}$, we simply compute $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{HG}) = \mathbf{HG}$. As a further benefit, in the actual construction, the multiplication causes the size of the trapdoor to grow according to the norm of the tag, which forced them to use tags from a set with small norm. As $\mathbf{G}^{-1}(\mathbf{X})$ has small norm regardless of the size of $\mathbf{X}$, our scheme is not subject to this restriction in terms of tag choice.

Homomorphic Computation of Trapdoors. Lattice-based signature schemes using the "vanishing trapdoors" technique of Boyen ([Boy10, MP12, DM14]) involve homomorphically evaluating a function $g = g_{(\mu,T)}$ over tags $\hat{t}_i$ statistically hidden in matrices $(\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_d)$ in the public key, where the function g to be evaluated is from a function family indexed by the message μ being signed and (possibly) some additional set of tags T sampled randomly. The output of this function is a matrix $\mathbf{A}_{(\mu,T)}$, with an associated trapdoor, the size of which grows in a manner depending on the function g being evaluated. The trapdoor can be used by the signer to sample short vectors from some coset of a lattice constructed using $\mathbf{A}_{(\mu,T)}$ whenever the tag hidden in $\mathbf{A}_{(\mu,T)}$ is *invertible*; these preimages are combined with the tags T to form the signature. However, when the tag hidden is 0, we refer to $\mathbf{A}_{(\mu,T)}$ as *punctured*, because in this case we can no longer sample short vectors, and in fact, $\mathbf{A}_{(\mu,T)}$ can be used to embed a challenge for the SIS problem.

In order to make the signature scheme secure, the function family g must be such that we can choose two sets of $d+1$ tags $\hat{T}_1, \hat{T}_2$ to hide in the matrices in the public key such that the tags satisfy certain properties. The first set of tags is used in the actual signature scheme, and they must be such that for any message μ and tags T, the tag associated with matrix $\mathbf{A}_{(\mu,T)}$ is *invertible*. The second set of tags is used in the security reduction, and they must be such that we can (with non-negligible probability) produce a signature for each of the adversary's queried messages while ensuring that a forgery output by the adversary will result in a punctured $\mathbf{A}_{\mu^*,T^*}$ with non-negligible probability.

In these previous schemes, the function g being computed was *linear* over the $d+1$ tags in the public key. We use a technique from recent works works by Boneh et al. [BGG$^+$14] and Wichs et al. [GVW14] to allow us to homomorphically compute a degree d polynomial over these tags. This lets us use just two tagged matrices in the public key instead of d. Computation in these works was defined in terms of basic operations of addition, multiplication, addition-with-constant, and multiplication-by-constant. However, naive evaluation of computation over homomorphic trapdoors results in larger growth in the size of the trapdoor than is necessary. To reduce the growth of the trapdoor, we use the "right-associativity" technique for multiplication. This technique was developed by Brakerski and Vaikuntanathan [BV14] and by Alperin-Sheriff and Peikert [ASP14] in the context of bootstrapping the GSW homomorphic encryption scheme [GSW13]. Those two papers only used the technique with $\{0,1\}$ messages (tags in our context) which resulted in "quasi-additive" noise growth in the size of the (implicit) trapdoors. Here, we show that for tags of bounded size at most d, the technique allows us to homomorphically evaluate a degree d polynomial while causing the trapdoors to grow by a $d^d \operatorname{poly}(n)$ factor.

Puncturable Homomorphic Trapdoor Functions. As a side contribution, we abstract out the properties required to construct "vanishing trapdoor"-based cryptographic primitives, including signatures, identity-based encryption and attribute-based encryption, into a primitive we call puncturable homomorphic trapdoor functions (PHTDFs). These functions are an extension of the Wichs et al. definition of homomorphic trapdoor functions [GVW14]. A PHTDF consists of a tagged function space a_i with corresponding trapdoors r_i. One can homomorphically compute functions g of these a_i, r_i to get a tagged function a^* and (if one knows the trapdoors for the original a_i) trapdoor r^*. Whenever the tag associated with a^* is invertible, one can use the trapdoor r^* to invert the function $f_{pk,a^*,x}$, where $x \in \mathcal{X}$ are indices.

For security, we require that for a punctured a^*, it should be difficult to find collisions $x \neq x'$, u, u' such that $f_{pk,a^*,x}(u) = f_{pk,a^*,x'}(u')$, even given oracle access to an inverter for $f_{pk,a_i,x}$ for arbitrary a_i with invertible tags and arbitrary x.

We do not expect realizations of PHTDFs under the various classical assumptions (just as we do not expect realizations of fully homomorphic cryptography in general from those assumptions). However, we believe that building our signature scheme generically using a PHTDF makes the proof of security easier to follow. Perhaps more importantly, we believe that viewing "vanishing trapdoor"-based cryptographic primitives under this framework and focusing on the function g to be computed may lead to realizations of these primitives with smaller public keys.

Organization. The remainder of the paper is organized as follows. In Section 2 we recall some preliminary information about lattice-based cryptography. In Section 3 we define and construct puncturable homomorphic trapdoor functions (PHTDFs). In Section 4 we show how to use a PHTDF to construct a secure signature scheme given that the function g being homomorphically computed satisfies certain properties. In Section 5 we provide our main result, the explicit construction and analysis of the function g.

Acknowledgements. I would like to thank Léo Ducas and Daniele Micciancio for some helpful correspondence regarding their signature scheme in the early stages of this work. Thanks are also due to the anonymous PKC '15 reviewers and to my advisor Chris Peikert for their helpful comments. Thanks are due to

2 Preliminaries

We write $[d]$ for a positive integer d to denote the set $\{1, \ldots, d\}$. We denote vectors over $\mathbb{Z}$ with lower-case bold letters (e.g. $\mathbf{x}$), and matrices by upper-case bold letters (e.g. $\mathbf{A}$). For an integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q. We represent the elements of $\mathbb{Z}_q$ as integers in $(-q/2, q/2]$ and define $|x| \in \mathbb{Z}_q$ by taking the absolute value of the representative in this range. We say that a function is *negligible*, written $\text{negl}(n)$, if it vanishes faster than the inverse of any polynomial in n. For a matrix $\mathbf{X} \in \mathbb{R}^{n \times k}$, the *largest singular value* (also known as the *spectral norm*) of $\mathbf{X}$ is defined as $s_1(\mathbf{X}) = \max_{\|\mathbf{u}\|=1} \|\mathbf{Xu}\|$.

2.1 Signatures

We briefly recall the standard definitions of digital signature schemes. A *signature scheme* SIG is a triple $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ of PPT (probabilistic polynomial time) algorithms, together with a message space $\mathcal{M} = \mathcal{M}_\lambda$. It is correct if, for all messages $\mu \in \mathcal{M}_\lambda$, $\mathsf{Ver}(vk, \mu, \sigma) = 1$ holds true, except with negligible probability in λ over the choice of $(sk, vk) \leftarrow \mathsf{Gen}(1^\lambda)$ and $\sigma \leftarrow \mathsf{Sign}(sk, \mu)$.

We now recall the standard security definitions for digital signature schemes. Existential unforgeability under static chosen-message attack, or eu-scma, is as follows: the adversary $\mathcal{A}$ first outputs a list of message $\mu_1, \ldots, \mu_Q$ to be signed, for some $Q = \text{poly}(n)$. The challenger then generates keys $(vk, sk) \leftarrow \mathsf{Gen}$ and signatures $\sigma_i \leftarrow \mathsf{Sign}(sk, \mu_i)$ for each $i \in [Q]$, and sends vk and $\{\sigma_i\}_{i \in [Q]}$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs an attempted forged signature (μ^*, σ^*). In order to satisfy eu-scma security, the probability that $\mu^* \neq \mu_i$ for any $i \in [Q]$ and that $\mathsf{Ver}(vk, \mu^*, \sigma^*) = 1$ accepts should be negligible in the security parameter λ.

Existential unforgeability under *adaptive* chosen-message attack (eu-acma security) is defined in a similar manner. The difference is that under this notion, $\mathcal{A}$ receives the verification key vk before making any queries, and is allowed to make queries one at a time, receiving back a signature before having to make its next query.

A standard technique for achieving adaptive security from static security is chameleon hashing [KR00]. An efficient construction (which has an immediate analog over general lattices) requiring a constant number of matrices was given by Ducas and Micciancio [DM14]. As a result, we can use it to make our eu-scma signature scheme adaptively secure without increasing the asymptotic size of our public key by more than a constant factor.

2.2 Lattices and Gaussians

A (full-rank) m-dimensional *integer lattice* Λ is an additive subgroup of $\mathbb{Z}^m$ with finite index. This work is concerned with the family of integer lattices whose cryptographic importance was first demonstrated by Ajtai [Ajt96]. For integers $n \geq 1$, modulus $q \geq 2$, an m-dimensional lattice from this family is specified by an "arity check" matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$:

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{Ax} = \mathbf{0} \in \mathbb{Z}_q^n\} \subseteq \mathbb{Z}^m.$$

For $\mathbf{y}$ in the subgroup of $\mathbb{Z}_q^n$ generated by the columns of $\mathbf{A}$, we define the coset

$$\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{y} \bmod q\} = \Lambda^{\perp}(\mathbf{A}) + \bar{\mathbf{x}},$$

where $\bar{\mathbf{x}} \in \mathbb{Z}^m$ is an arbitrary solution to $\mathbf{A}\bar{\mathbf{x}} = \mathbf{y}$.

We briefly recall Gaussian distributions over lattices (for more details see [MR04, GPV08]). For $s > 0$ and dimension $m \geq 1$, the Gaussian function $\rho_s : \mathbb{R}^m \to (0, 1]$ is defined as $\rho_s(\mathbf{x}) = \exp(-\pi \|\mathbf{x}\|^2 / s^2)$. For a coset $\Lambda + \mathbf{c}$ of a lattice Λ, the *discrete Gaussian distribution* $D_{\Lambda+\mathbf{c},s}$ (centered at zero) assigns probability proportional to $\rho_s(\mathbf{x})$ to each vector in the coset, and probability zero elsewhere.

We will need several standard facts about discrete Gaussians over lattices. First, for $\epsilon > 0$ the *smoothing parameter* [MR04] $\eta_\epsilon(\Lambda)$ of an n-dimensional lattice is a positive real value. We will not need its precise definition in this work. Instead, we recall the few relevant facts that we need; for more details, see, e.g., [MR04, GPV08, MP12].

Lemma 2.1. *Let $m \geq Cn \lg q$ for some constant $C > 1$.*

1. *For any $\omega(\sqrt{\log n})$ function, we have $\eta_\epsilon(\mathbb{Z}^n) \leq \omega(\sqrt{\log n})$ for some negligible $\epsilon(n) = \mathrm{negl}(n)$.*
2. *With all but $\mathrm{negl}(n)$ probability over the uniformly random choice of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the following holds: For $\mathbf{e} \leftarrow D_{\mathbb{Z}^m,r}$ where $r = \omega(\sqrt{\log n})$, the distribution of $\mathbf{y} = \mathbf{A}\mathbf{e} \bmod q$ is within $\mathrm{negl}(n)$ statistical distance of uniform, and the conditional distribution of $\mathbf{e}$ given $\mathbf{y}$ is $D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{A}),r}$.*
3. *For any m-dimensional lattice Λ, any $\mathbf{c} \in \mathbb{Z}^m$, and any $r \geq \eta_\epsilon(\Lambda)$ where $\epsilon(n) = \mathrm{negl}(n)$, we have $\|D_{\Lambda+\mathbf{c},r}\| \leq r\sqrt{m}$ with all but $\mathrm{negl}(n)$ probability. In addition, for $\Lambda = \mathbb{Z}$ we have $|D_{\mathbb{Z},r}| \leq r \cdot \omega(\sqrt{\log n})$ except with $\mathrm{negl}(n)$ probability.*
4. *For any $r > 0$, and for $\mathbf{R} \leftarrow D_{\mathbb{Z},r}^{n \times k}$, we have $s_1(\mathbf{R}) \leq r \cdot O(\sqrt{n} + \sqrt{k})$ except with $\mathrm{negl}(n)$ probability.*
5. *Let $\Lambda \subset \mathbb{R}^m$ be a lattice and $r \geq 2\eta_\epsilon(\Lambda)$ for some $\epsilon \in (0, 1)$. Then for any $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{y} \in \Lambda + \mathbf{c}$, we have $\Pr[D_{\Lambda+\mathbf{c},r} = \mathbf{y}] \leq 2^{-n} \cdot \frac{1+\epsilon}{1-\epsilon}$.*

The SIS problem. For $\beta > 0$, the *short integer solution* problem $\mathsf{SIS}_{n,q,\beta}$ is an average-case version of the approximate shortest vector problem on $\Lambda^{\perp}(\mathbf{A})$. Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ for any $m = \mathrm{poly}(n)$, the problem is to find a nonzero vector $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leq \beta$. For $q \geq \beta\sqrt{n}\omega(\sqrt{\log n})$, it has been shown that solving this problem with non-negligible success probability over the random choice of $\mathbf{A}$ is at least as hard as probabilistically approximating the classic Shortest Independent Vectors Problem (SIVP) on n-dimensional lattices to within $\tilde{O}(\beta\sqrt{n})$ factors in the *worst case*. [Ajt96, MR04, GPV08]

2.3 Trapdoors for Lattices

In this section, we recall the efficient trapdoor construction and associated sampling algorithm of Micciancio and Peikert [MP12], which is at the heart of our signature scheme. This construction uses a universal public "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times w}$ for which there is an efficient discrete Gaussian sampling algorithm for any parameter $r \geq \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^{\perp}(\mathbf{G}))$ (for some $\epsilon(n) = \mathrm{negl}(n)$), i.e., an algorithm that, given any $\mathbf{y} \in \mathbb{Z}_q^n$ and r, outputs a sample from $D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{G}),r}$. For concreteness, as in [MP12] we take $\mathbf{G} = \mathbf{I}_n \otimes [1, 2, 4, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{n \times nk}$ for $k = \lceil \lg q \rceil$. We will somewhat abuse notation by writing $\mathbf{G}^{-1}(\mathbf{v})$ to denote computing the lexically first (using $0 < 1$) binary vector $\mathbf{x} \in \{0, 1\}^{nk}$ such that $\mathbf{G}\mathbf{x} = \mathbf{v}$; the vector will be unique if $q = 2^\ell$, but for other moduli (including the ones used in our scheme), there will potentially be more than one such vector.

Following [MP12], we say that an integer matrix $\mathbf{R} \in \mathbb{Z}^{(m)\times nk}$ is a $\mathbf{G}$-trapdoor with tag $\mathbf{H}$ for $\mathbf{A} \in \mathbb{Z}_q^{n\times(m+nk)}$ if $\mathbf{A}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{I}\end{smallmatrix}\right] = \mathbf{HG}$ for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$. If $\mathbf{H} = \mathbf{0}$, we say that $\mathbf{R}$ is a "punctured" trapdoor for $\mathbf{A}$. We require the following two lemmas regarding these trapdoors.

Lemma 2.2 ([MP12]). *There is a probabilistic polynomial time algorithm* $\mathsf{GenTrap}(\bar{\mathbf{A}}, \mathbf{H}, r)$ *that on input a matrix* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$*, a matrix* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *and parameter* $r \geq \omega(\sqrt{\log(n)})$*, outputs a matrix* $\mathbf{A} := [\bar{\mathbf{A}} \mid -\bar{\mathbf{A}}\mathbf{R} + \mathbf{HG}] \in \mathbb{Z}_q^{n\times(m+nk)}$ *which is statistically close to uniform over the choice of* $\mathbf{R}$ *as well as a matrix* $\mathbf{R} \in \mathbb{Z}_q^{m\times nk}$ *such that if* $\mathbf{H}$ *is invertible,* $\mathbf{R}$ *is a* $\mathbf{G}$*-trapdoor with tag* $\mathbf{H}$ *for* $\mathbf{A}$*, while if* $\mathbf{H} = \mathbf{0}$*, then* $\mathbf{R}$ *is a punctured trapdoor.*

Lemma 2.3 ([MP12]). *Let* $\mathbf{R}$ *be a* $\mathbf{G}$*-trapdoor for* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$*. There is an efficient randomized algorithm* $\mathsf{SampleD}(\mathbf{A}, \mathbf{u}, \mathbf{R}, r)$ *that, given* $\mathbf{R}$*, any* $\mathbf{u} \in \mathbb{Z}_q^n$*, and any* $r \geq s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n}) \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ *(for some* $\epsilon(n) = \mathrm{negl}(n)$*), samples from a distribution within* $\mathrm{negl}(n)$ *distance of* $D_{\Lambda^\perp_{\mathbf{u}}(\mathbf{A}),r}$*.*

3 Puncturable Homomorphic Trapdoor Functions

In this section we define and describe our construction of puncturable homomorphic trapdoor functions, a primitive that abstracts out the properties required for our signature scheme, as well as those for many other lattice-based cryptographic primitives.

3.1 Definition

Our definition is an extension of the Wichs et al. definition of homomorphic trapdoor functions [GVW14]. A *puncturable homomorphic trapdoor function* (PHTDF) scheme consists of six polynomial-time algorithms $(\mathsf{Gen}, \mathsf{GenTrap}, f, \mathsf{Invert}, \mathsf{Eval}^{td}_{pk}, \mathsf{Eval}^{func}_{pk})$ with the following syntax:

$pk \leftarrow \mathsf{Gen}(1^\lambda)$ generates the public key. The parameter λ also defines a tag space $\mathcal{T}$, a trapdoor space $\mathcal{R}$, a tagged function space $\mathcal{A}$,[1] an index space $\mathcal{X}$, an input space $\mathcal{U}$ and an output space $\mathcal{V}$. We also need an efficiently sampleable key distribution $D_{\mathcal{R}}$ over $\mathcal{R}$, a parameterized input distribution $D_{\mathcal{U},s}$ over $\mathcal{U}$ and a tag set distribution $D_{\mathcal{T}^\ell}$ over $\mathcal{T}^\ell$. We also need to be able to sample $\mathcal{V}$ uniformly at random.

$(a, r) \leftarrow \mathsf{GenTrap}(pk, t)$ generates a trapdoor $r \leftarrow D_{\mathcal{R}}$ for $(pk, a \in \mathcal{A})$, with t the tag *associated with* a. We need the distribution of a to be statistically close to uniform over $\mathcal{A}$. We define the auxiliary function $\mathsf{Tag}(pk, a, r)$ to output $t \in \mathcal{T}$ if t is the tag associated with a and r is a trapdoor for a.

$f_{pk,a,x} : \mathcal{U} \to \mathcal{V}$ is a deterministic function (not necessarily injective) indexed by $x \in \mathcal{X}$, and $pk, a \in \mathcal{A}$.

$\mathsf{Invert}_{r,pk,a,x,s} : \mathcal{V} \to \mathcal{U}$ is a *puncturable* probabilistic inverter indexed by $x \in \mathcal{X}$, $r \in \mathcal{R}$ and $pk, a \in \mathcal{A}$. The parameter $s \in \mathbb{R}$ relates to some property $\mathsf{Prop} : \mathcal{U} \to \mathbb{R}$ of the desired inverse when the function space is not injective. We need to be able to find an inverse u such that $\mathsf{Prop}(u) \leq s$ (with overwhelming probability) whenever the tag t associated with a is *invertible* and whenever the trapdoor r is strong enough to invert with parameter s. If these conditions are not fulfilled, it outputs $\perp$.

[1] Technically, the function space is also parameterized by the public key and the index space.

$r^* \leftarrow \mathsf{Eval}^{td}_{pk}(g, \{(a_i, r_i)\}_{i\in[\kappa]}, T), a^* \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{a_i\}_{i\in[\kappa]}, T)$ are deterministic trapdoor/function homomorphic evaluation algorithms. The algorithms take as input some function $g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}$, a set of tags $T = \{t_j\}_{j\in[\ell]} \in \mathcal{T}$, as well as values $a_i \in \mathcal{A}$, $r_i \in \mathcal{R}$. The outputs are $r^* \in \mathcal{R}$ and $a^* \in \mathcal{A}$.

Correctness of Evaluation. Let $(pk) \leftarrow \mathsf{Gen}(1^\lambda)$, $\hat{T} = \{\hat{t}_i \in \mathcal{T}\}_{i\in[\kappa]}$, $T = \{t_j \in \mathcal{T}\}_{j\in[\ell]}$, $(r_i, a_i) \leftarrow \mathsf{GenTrap}(pk, \hat{t}_i)$. Let $g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}$ and let $t^* := g(\hat{T}, T)$. We require that for

$$r^* \leftarrow \mathsf{Eval}^{td}_{pk}(g, \{(a_i, r_i)\}_{i\in[\kappa]}, T), \quad a^* \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{a_i\}_{i\in[\kappa]}, T)$$

we have that r^* is a trapdoor for (pk, a^*), and that t^* is the tag associated with a^*.

Leveled relaxation: In a *leveled* fully homomorphic scheme, each trapdoor $r_i \in \mathcal{R}$ has an associated level of noise $\beta_i \in \mathbb{R}$. The noise level β^* of the homomorphically computed key r^* is larger than the initial noise levels, and depends in concrete instantiations on the function g (and the method by which it is computed), the initial trapdoors r_i, and the tags $\hat{t}_i \in \mathcal{T}$. If the noise level $\beta^* > \beta_{max}$ for some threshold β_{max}, the trapdoor will not be strong enough to compute $\mathsf{Invert}_{r,pk,a,x,s}$, thus limiting the type of functions g that can be computed.

Definition 3.1. We call a function g *admissible* with parameter s on the set of tags $\hat{T} := \{t_i\}_{i\in[\kappa]}$, if, whenever, the initial trapdoors r_i have noise levels $\beta_i \leq \beta_{init} = \omega(\sqrt{\log n})$, r^* will have noise level $\beta^* \leq s/\omega(\sqrt{\log n})$ with overwhelming probability.

In our concrete construction below, the trapdoors will be matrices $\mathbf{R}_i$, and we measure the noise level using the spectral norm $s_1(\mathbf{R}_i)$.

Distribution Equivalence of Inversion. We require the following statistical indistinguishability for $pk \leftarrow \mathsf{Gen}(1^\lambda)$, trapdoor/function pair (r, a) with an invertible tag t:

$$(pk, r, a, x, u, v) \overset{s}{\approx} (pk, r, a, x, u', v')$$

where $x \in \mathcal{X}$ is arbitrary, $u \leftarrow D_{\mathcal{U},s}$, $v := f_{pk,a,x}(u)$, $v' \leftarrow \mathcal{V}$ and $u' \leftarrow \mathsf{Invert}_{r,pk,a,x,s}(v')$.

Security for PHTDFs. Our security definition allows us to use a PHTDF easily as a building block in our construction of a secure signature scheme. Roughly, we need it to be difficult, given two *punctured* trapdoor functions $a^{(1)}, a^{(2)}$ computed using a function $g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}$ admissible with parameter s, to find a collision $u, u', x, x' \in \mathcal{U}$, with $x \neq x' \in \mathcal{X}$ such that $f_{pk,a^{(1)},x}(u) = f_{pk,a^{(2)},x'}(u')$, even given oracle access to inversion queries for non-punctured functions a' computed using that same function g.

The security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ is parameterized by a security parameter λ, as well as a function $g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}$ and set of tags $\hat{T} := \{t_i\}_{i\in[\kappa]}$ such that g is admissible with some parameter s on the set of tags $\hat{T}$:

1. $\mathcal{C}$ runs $pk \leftarrow \mathsf{Gen}(1^\lambda)$ and then computes $(a_i, r_i) \leftarrow \mathsf{GenTrap}(pk, \hat{t}_i)$ for each $i \in [\kappa]$. $\mathcal{A}$ is given pk and $\{a_i\}$.
2. $\mathcal{A}$ may make inversion queries. To query, $\mathcal{A}$ sends some $v \in \mathcal{V}$, $x \in \mathcal{X}$ and some $T' := \{t'_j\}_{j\in[\ell]}$ such that $g(\hat{T}, T')$ is *invertible*. $\mathcal{C}$ computes $r' \leftarrow \mathsf{Eval}^{td}_{pk}(g, \{(a_i, r_i)\}, T')$ as well as $a' \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{a_i\}, T')$, samples $u \leftarrow \mathsf{Invert}_{r',pk,a',x,s}(v)$ and returns u to $\mathcal{A}$.

3. $\mathcal{A}(1^\lambda)$ outputs tag sets $T^{(1)} := \{t_j^{(1)}\}, T^{(2)} := \{t_j^{(2)}\} \in \mathcal{T}^\ell$ which satisfy $g(\hat{T}, T^{(1)}) = g(\hat{T}, T^{(2)}) = 0$, as well as $(u^{(1)}, u^{(2)} \in \mathcal{U}, x^{(1)} \neq x^{(2)} \in \mathcal{X})$, and wins if $f_{pk,a^{(1)},x^{(1)}}(u^{(1)}) = f_{pk,a^{(2)},x^{(2)}}(u^{(2)})$, where $\mathsf{Prop}(u^{(1)}), \mathsf{Prop}(u^{(2)}) \le s$ and $a^{(b)} \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{(a_i)\}, T^{(b)})$ for $b \in \{1, 2\}$.

We say the PHTDF satisfies $(\lambda, g, \{t_i\})$-collision resistance when punctured (CRP) security if every PPT adversary has a negligible probability of success in the above game.

3.2 Construction: Basic Algorithms

Our explicit lattice-based construction of PHTDFs is as follows. While our construction is written over general lattices, it may be instantiated easily in the ring setting. We have a lattice dimension parameter $n := \lambda$ and a prime modulus $q > 2$. Other parameters include $\bar{m} = O(n \log q)$, $k = \lceil \log_2 q \rceil$, $r = \omega(\sqrt{\log n})$, $m = \bar{m} + nk$. We let $\mathcal{A} = \mathbb{Z}_q^{n \times nk}$, $\mathcal{R} = \mathbb{Z}_q^{\bar{m} \times nk}$, $D_{\mathcal{R}} = D_{\mathbb{Z},r}^{\bar{m} \times nk}$, $\mathcal{X} = \{-1, 0, 1\}^{\bar{m}}$, $\mathcal{U} = \mathbb{Z}_q^m$, $D_{\mathcal{U},s} = D_{\mathbb{Z},s}^m$, $\mathcal{V} = \mathbb{Z}_q^n$.

Tags. We first consider the general lattice setting. For a degree-n polynomial $f(x)$, the ring $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ has a standard embedding into $\mathrm{M}_n(\mathbb{Z}_q)$ [ABB10]. The embedding sets the ith row of the matrix $\mathbf{H}$ corresponding to $g(x) \in R_q$ to the coefficients (in increasing order of degree) of $x^{i-1} g(x) \bmod R_q$. By choosing $f(x)$ to be irreducible over $\mathbb{Z}_q$, we ensure that R_q is a field, and set $\mathcal{T} = R_q$. In this setting, there are some operations that can only be done over the subring of scalar elements $\{t\mathbf{I}_n\} \subset \mathcal{T}$, because for technical reasons, the tags must be commutative under multiplication with respect to arbitrary matrices.

When using our construction in the ring setting, some care must be taken in constructing the embedding; see [DM14] for an instantiation of a tag space that will work for our scheme as well.

$\mathsf{Gen}(1^\lambda)$: Choose $\bar{\mathbf{A}}, \mathbf{B} \leftarrow \mathbb{Z}_q^{n \times \bar{m}}$, and output $pk = (\bar{\mathbf{A}}, \mathbf{B})$.

$\mathsf{GenTrap}(\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times \bar{m}}, \mathbf{H} \in \mathcal{T})$: Output $(\mathbf{A} = [\bar{\mathbf{A}} \mid \mathbf{C}], \mathbf{R}) \leftarrow \mathsf{GenTrap}(\bar{\mathbf{A}}, \mathbf{H}, r)$ using Lemma 2.2.

$\mathsf{Tag}_{pk=\bar{\mathbf{A}}}(\mathbf{C} \in \mathbb{Z}_q^{n \times nk}, \mathbf{R})$: Compute $\mathbf{X} = [\bar{\mathbf{A}} \mid \mathbf{C}] \left[\begin{smallmatrix} \mathbf{R} \\ \mathbf{I} \end{smallmatrix}\right]$. If $\mathbf{X} = \mathbf{HG}$ for some $\mathbf{H}$, output $\mathbf{H}$.

$f_{\mathbf{A}=[\bar{\mathbf{A}}|\mathbf{C}],\mathbf{x} \in \mathcal{X}}(\mathbf{u} \in \mathcal{U})$**:** Let $\mathbf{z} = (\mathbf{u}, \mathbf{x}) \in \mathbb{Z}_q^{m+\bar{m}}$. Output $\mathbf{v} := [\mathbf{A} \mid \mathbf{B}]\mathbf{z}$.

$\mathsf{Invert}_{\mathbf{A}=[\bar{\mathbf{A}}|\mathbf{C}],\mathbf{R} \in \mathcal{R},\mathbf{x} \in \mathcal{X},s \in \mathbb{R}}(\mathbf{v} \in \mathcal{V})$**:** Compute $\mathbf{H} = \mathsf{Tag}(\mathbf{A}, \mathbf{R})$. If $\mathbf{H} = 0$ or $\mathbf{H} = \perp$ or if the parameter $s < s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, output $\perp$. Otherwise, output $\mathbf{u} \leftarrow \mathsf{SampleD}(\mathbf{A}, \mathbf{v} - \mathbf{Bx}, \mathbf{R}, s)$ using Lemma 2.3. We define $\mathsf{Prop}(\mathbf{u}) := \|\mathbf{u}\|/\sqrt{\bar{m}}$, so that by Lemma 2.1, with overwhelming probability over the randomness of $\mathsf{SampleD}$, $\mathsf{Prop}(\mathbf{u}) \le s$.

As long as $s \ge s_1(\mathbf{R}) \cdot \omega(\sqrt{\log n})$, distributional equivalence of inversion follows immediately from Lemma 2.1

3.3 Construction: Homomorphic Computation

We define the algorithms Eval^{td}_{pk}, $\mathsf{Eval}^{func}_{pk}$ by showing how to compute basic operations. Note that in our actual construction in Section 5, instead of naively composing addition and multiplication, we extend a technique used by Brakerski and Vaikuntanathan [BV14] and Alperin-Sheriff and Peikert [ASP14] regarding *chaining homomorphic multiplications* that was originally developed in the context of bootstrapping the

GSW [GSW13] encryption scheme. These techniques carry over immediately, because the only difference between the form of a GSW ciphertext and the form of our trapdoor functions is that in the former case, the matrix $\mathbf{A}$ in the public key is an LWE instance $\mathbf{A} = \left[\substack{\bar{\mathbf{A}} \\ \mathbf{b}^t = \mathbf{s}^t \bar{\mathbf{A}} + \mathbf{e}^t} \right]$, while in our case it is truly uniform.

In the descriptions of the basic homomorphic operations on tags below, $\mathbf{C}_i \in \mathcal{A}$ is a trapdoor function, $\mathbf{C}_i := \bar{\mathbf{A}}\mathbf{R}_i + t_i\mathbf{G}$.

Homomorphic addition $(\mathbf{C}_1 \boxplus \mathbf{C}_2)$ of two functions is defined as $\mathbf{C}^* \leftarrow \mathbf{C}_1 + \mathbf{C}_2$. The operation induces $\mathbf{R}^* \leftarrow \mathbf{R}_1 + \mathbf{R}_2$ on the trapdoors. The tags t_1, t_2 may be any elements in $\mathcal{T}$.

Homomorphic multiplication $(\mathbf{C}_1 \boxdot \mathbf{C}_2)$ is defined as $\mathbf{C}^* \leftarrow \mathbf{C}_1 \cdot \mathbf{G}^{-1}(\mathbf{C}_2)$, and is *right associative*. The operation induces $\mathbf{R}^* \leftarrow \mathbf{R}_1\mathbf{G}^{-1}(\mathbf{C}_2) + t_1\mathbf{R}_2$ on the trapdoors. Here the tag t_1 must be a scalar (of the form $t_1\mathbf{I}_n$), while t_2 can be any element of $\mathcal{T}$. Unlike in the GSW scheme, we need to compute $\mathbf{G}^{-1}(\mathbf{C}_2)$ in the deterministic manner described in Section 2.3.

Multiplication by a constant $(\mathbf{C} \boxdot \mathbf{H})$ is defined as $\mathbf{C}^* \leftarrow \mathbf{C} \cdot \mathbf{G}^{-1}(\mathbf{H}\mathbf{G})$. The operation induces $\mathbf{R}^* \leftarrow \mathbf{R}\mathbf{G}^{-1}(\mathbf{H}\mathbf{G})$ on the trapdoor. The constant tag $\mathbf{H}$ may be any element in $\mathcal{T}$, but the tag t associated with $\mathbf{C}$ must be a scalar.

Addition of a constant $(\mathbf{C} \boxplus \mathbf{H})$ is defined as $\mathbf{C}^* \leftarrow \mathbf{C} + \mathbf{H}\mathbf{G}$. This operation does not affect the trapdoor, which remains $\mathbf{R}$. Both the constant tag $\mathbf{H}$ and the tag t associated with $\mathbf{C}$ may be any element in $\mathcal{T}$.

3.4 Construction: Security

We now prove the security of our PHTDF construction under the SIS assumption.

Theorem 3.2. *Let g be admissible with parameter s for $\hat{T} := \{t_i\}_{i \in [\kappa]}$. Assuming the $\mathsf{SIS}_{n,q,\beta}$ assumption holds for $\beta := O(s^2\sqrt{n \log q})$, the scheme satisfies CRP$_{n,g,\hat{T}}$ security.*

Proof. Let g be admissible with parameter s for $\hat{T} := \{t_i\}_{i \in [\kappa]}$. Now, assume there exists an adversary $\mathcal{A}$ that wins the PHTDF security game for the above scheme with non-negligible probability δ with respect to g and $\hat{T}$. We consider an alternate game where we change our PHTDF by setting $\mathbf{B} = \bar{\mathbf{A}}\mathbf{S}$ for $\mathbf{S} \leftarrow D^{\bar{m} \times \bar{m}}_{\mathbb{Z}, \omega(\sqrt{\log n})}$ instead of choosing it uniformly at random. By Lemma 2.1, this change is statistically indistinguishable and it does not affect our ability to invert, so that the adversary $\mathcal{A}$ still wins with non-negligible probability.

We now give a reduction to SIS. The reduction uses the challenge matrix $\bar{\mathbf{A}}$ as the public key. It then computes $(\mathbf{C}_i, \mathbf{R}_i) \leftarrow \mathsf{GenTrap}(\bar{\mathbf{A}}, t_i, \omega(\sqrt{\log n}))$. We can invert with parameter s on any trapdoor function and trapdoor $\mathbf{C}', \mathbf{R}'$ output by $\mathsf{Eval}^{func}_{pk}$, Eval^{td}_{pk} when computing g homomorphically on $\hat{T}$ and arbitrary other tags $T' := \{t'_j\}_{j \in [\ell]}$ as long as $g(\hat{T}, T')$ is invertible, so we can properly answer all of $\mathcal{A}$'s inversion queries.

With non-negligible probability δ, $\mathcal{A}$ eventually outputs tag sets $T^{(1)} := \{t^{(1)}_j\}_{j \in [\ell]}$, $T^{(2)} := \{t^{(2)}_j\}_{j \in [\ell]}$ such that $g(\hat{T}, T^{(1)}) = g(\hat{T}, T^{(2)}) = 0$ along with $(\mathbf{u}^{(1)} = (\mathbf{u}^{(1)}_1 \in \mathbb{Z}^{\bar{m}}_q, \mathbf{u}^{(1)}_2 \in \mathbb{Z}^{nk}_q), \mathbf{x}^{(1)}), (\mathbf{u}^{(2)} = (\mathbf{u}^{(2)}_1 \in \mathbb{Z}^{\bar{m}}_q, \mathbf{u}^{(2)}_2 \in \mathbb{Z}^{nk}_q), \mathbf{x}^{(2)})$ such that $\mathsf{Prop}(\mathbf{u}^{(1)}), \mathsf{Prop}(\mathbf{u}^{(2)}) \leq s$, $\mathbf{x} \neq \mathbf{x}'$ and $f_{\mathbf{A}^{(1)} = [\bar{\mathbf{A}}|\mathbf{C}^{(1)}], \mathbf{x}^{(1)}}(\mathbf{u}^{(1)}) = f_{\mathbf{A}^{(2)} = [\bar{\mathbf{A}}|\mathbf{C}^{(2)}], \mathbf{x}^{(2)}}(\mathbf{u}^{(2)})$, where $\mathbf{C}^{(b)} := \mathsf{Eval}^{func}_{pk}(g, \{t_i\}, \{t^{(b)}_j\})$. Since g is admissible with parameter s, we have $\mathbf{C}^{(b)} = \bar{\mathbf{A}}\mathbf{R}^{(b)}$, where $s_1(\mathbf{R}^{(b)}) \leq s/\omega(\sqrt{\log n})$. Let

$$\mathbf{v} := \mathbf{u}^{(1)}_1 - \mathbf{u}^{(2)}_1 + \mathbf{R}^{(1)}\mathbf{u}^{(1)}_2 - \mathbf{R}^{(2)}\mathbf{u}^{(2)}_2 + \mathbf{S}(\mathbf{x}^{(1)} - \mathbf{x}^{(2)})$$

Recalling the definition of f, we then have that $\bar{\mathbf{A}}\mathbf{v} = \mathbf{0}$.

Since $\mathsf{Prop}(\mathbf{u}^{(1)}), \mathsf{Prop}(\mathbf{u}^{(2)}) \leq s$, by Lemma 2.1 with overwhelming probability $\|\mathbf{R}^{(b)}\mathbf{u}_2^{(b)}\| \leq O(s^2(\sqrt{\bar{m}+nk})) \leq O(s^2\sqrt{n\log q})$. The other terms in $\mathbf{v}$ can be bounded at or below $O(s^2\sqrt{n\log q})$, so

$$\|\mathbf{v}\| \leq O(s^2(\sqrt{n\log q})) \leq \beta$$

Thus, in order to show that we have in fact solved SIS, we need only show that with non-negligible probability, $\mathbf{v} \neq \mathbf{0}$. To do so, we let $\mathbf{x}^* = \mathbf{x}^{(1)} - \mathbf{x}^{(2)}$ and choose some entry of $\mathbf{x}^*$ that is nonzero (since $\mathbf{x}^{(1)} \neq \mathbf{x}^{(2)}$, such an entry exists); without loss of generality we may say it is x_1^*. Now, let $\mathbf{s}$ be the first column of $\mathbf{S}$, let $\mathbf{b}$ be the first column of $\mathbf{B}$, and fix the last $\bar{m}-1$ columns of $\mathbf{S}$ as well as $\mathbf{u}^{(b)}$, $\mathbf{R}^{(b)}$. Then we have $\mathbf{v} = \mathbf{0}$ only if $\mathbf{s} \cdot x_1^* = \mathbf{y}$ for some fixed $\mathbf{y}$. Over the view of the adversary, $\mathbf{s}$ is distributed as a discrete Gaussian of parameter larger than $2\eta_\epsilon(\Lambda^\perp(\bar{\mathbf{A}}))$ for an $\epsilon = \operatorname{negl}(n)$ over the coset $\Lambda_{\mathbf{b}}^\perp(\bar{\mathbf{A}})$. As a result, by Lemma 2.1 we have $\mathbf{s} = \mathbf{y}/x_1^*$ with only negligible probability, so that with probability $\delta - \operatorname{negl}(n)$, $\mathbf{v} \neq \mathbf{0}$ and we have solved SIS. □

4 Signatures from PHTDFs

Here we show how to construct a statically secure (eu-scma) signature scheme using a PHTDF, in a manner that encompasses both our signature scheme and that of Ducas and Micciancio. The reduction itself depends directly on the properties of the *admissible* function g that is homomorphically computed in the Sign and Ver algorithms. We recall that one can apply the efficient generic transformation using chameleon hash functions to achieve full (eu-acma) security.

4.1 Required Properties of the Function g and Tags

In order for our construction to work, we need to be able to choose two different sets of tags $\hat{T} = \{t_i \in \mathcal{T}\}$, one set for the actual scheme and one set for the security reduction, to use when calling $(a_i, r_i) \leftarrow \mathsf{GenTrap}(pk, \hat{t}_i)$. In particular, when $g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}$ is homomorphically evaluated $Q = \operatorname{poly}(n)$ times on the fixed set of tags $\hat{T}$ and on sampled sets of tags $T := \{t_j\} \leftarrow D_{\mathcal{T}^\ell}$, we need the following properties to be fulfilled:

Actual Scheme The tag homomorphically computed by g must always be invertible.

Security Reduction Here, we need two different properties to be satisfied with non-negligible probabilities:

1. At most one of the Q sets of tags $T_i \leftarrow D_{\mathcal{T}^\ell}$ (corresponding to signature queries by the adversary) may result in $g(\hat{T}, T_i) = 0$.
2. $\mathcal{A}$ chooses tags $T^* := \{t_j^*\}$ for his forgery such that $g(\hat{T}, T^*) = 0$.

4.2 Generic Signature Scheme from PHTDFs

We now present our generic signature scheme constructed from a PHTDF. It is parameterized by a security parameter λ as well as a family of functions $\{g : \mathcal{T}^\kappa \times \mathcal{T}^\ell \to \mathcal{T}\} \in \mathcal{G}$ admissible with parameter s and sets of tags $\hat{T} := \{\hat{t}_i \in \mathcal{T}\}_{i\in[\kappa]}$, which allow us to satisfy the properties in Section 4.1 for the actual scheme, and $\hat{T}^* := \{\hat{t}_i^* \in \mathcal{T}\}_{i\in[\kappa]}$, which allow us to satisfy the properties for the security reduction. The specific choice of function will depend on a parameter $d = \omega(\log\log n)$ (so that $2^{\lfloor c^d \rfloor}$ for some constant $c > 1$ is superpolynomially large), which in our concrete construction of g corresponds to the degree of a polynomial. Below, we assume that d is fixed and so the specific function g in the family is determined.

We also have some requirements for the index space $\mathcal{X}$. In particular, we need there to exist a collision-resistant hash function $h : \{0,1\}^* \to \mathcal{X}$ such that there is some $x^* \in \mathcal{X}$ that is not in the range of h, i.e. no μ exists such that $h(\mu) = x^*$. In our concrete instantiation, we can satisfy this property by choosing a collision-resistant hash function which maps into $\{0,1\}^{\bar{m}}$ (in which case any $x^* \in \mathcal{X} = \{-1,0,1\}^{\bar{m}}$ with a negative element is not in the range of h and is therefore not a valid hash of any message).

$\mathsf{Gen}(1^\lambda)$**:** Compute $pk_{\text{PHTDF}} \leftarrow \mathsf{PHTDF}.\mathsf{Gen}(1^\lambda)$, and then let $v \leftarrow \mathcal{V}$, $\{(a_i, r_i) \leftarrow \mathsf{PHTDF}.\mathsf{GenTrap}(pk_{\text{PHTDF}}, t_i)\}_{i\in[\kappa]}$. We set $vk = (pk_{\text{PHTDF}}, \{a_i\}_{i\in[\kappa]}, v)$, $sk = (\{r_i\}_{i\in[\kappa]})$.

$\mathsf{Sign}(sk, x = h(\mu) \in \mathcal{X})$**:** Sample $T := \{t'_j\} \leftarrow D_{\mathcal{T}^\ell}$, and compute $r^* \leftarrow \mathsf{Eval}^{td}_{pk}(g, \{(a_i, r_i)\}_{i\in[\kappa]}, T)$ and $a^* \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{a_i\}_{i\in[\kappa]}, T)$. Let $u \leftarrow \mathsf{Invert}_{r^*,pk,a^*,x,s}(v)$, and output $\sigma = (u, T)$.

$\mathsf{Ver}(vk, x = h(\mu) \in \mathcal{X}, \sigma = (u, T \in \mathcal{T}^\ell)$**:** Compute $a^* \leftarrow \mathsf{Eval}^{func}_{pk}(g, \{a_i\}_{i\in[\kappa]}, T)$, and verify that $f_{pk,a^*,x}(u) = v$ and that $\mathsf{Prop}(u) \leq s$.

Correctness. We immediately have that the scheme is correct, since in the actual scheme the homomorphically computed tag will always be invertible, and a u output by $\mathsf{Invert}_{r^*,pk,a^*,x,s}(v)$ will always satisfy $f_{pk,a^*,x}(u) = v$, and will with overwhelming probability satisfy that $\mathsf{Prop}(u) \leq s$.

Security Reduction. We prove that this signature scheme satisfies eu-scma security assuming the underlying PHTDF satisfies $(\lambda, g, \hat{T})$-CRP security for g admissible with parameter s on a set of tags $\hat{T} := \{t_i\}_{i\in[\kappa]}$.

Theorem 4.1. *Let g be admissible with parameter s on a set of tags $\hat{T} := \{\hat{t}_i\}$ which allow us to satisfy the properties for the security reduction described in Section 4.1 with non-negligible probabilities $\epsilon_1 := \epsilon_1(Q, d, \delta), \epsilon_2 := \epsilon_2(Q, \delta)$ respectively against an eu-scma adversary $\mathcal{A}$ making Q signature queries and succeeding with probability δ. Then there exists an algorithm $\mathcal{S}^{\mathcal{A}}$ that breaks the $(\lambda, g, \hat{T})$-CRP security of the underlying PHTDF with probability $\epsilon^* = (\delta - (1 - \epsilon_1))\epsilon_2$, and so the above signature scheme is eu-scma secure.*

Proof. We assume we have an adversary $\mathcal{A}$ against the eu-scma security game who makes Q signature queries and succeeds in outputting a successful forgery with probability δ. The simulator $\mathcal{S}$ first receives the Q messages $(x^{(j)} = h(\mu^{(j)}) \in \mathcal{X})$ from the adversary. Next, it samples Q sets of tags $T_i := \{t_{ij}\}_{j\in[\ell]} \leftarrow D_{\mathcal{T}^\ell}$. With probability ϵ_1, at most one set of these tags will result in $g(\hat{T}, T_i) = 0$. If there is more than one such set of tags, the simulator aborts (this happens with probability at most $(1 - \epsilon_1)$). Otherwise, it invokes the $(\lambda, g, \{t_i\})$-CRP security game, receiving back pk_{PHTDF} and $A := \{a_i\}_{i\in[\kappa]}$.

Now, if there exists a T_i such that $g(\hat{T}, T_i) = 0$, we set $T^\diamond := T_i$ and $x^\diamond := x^{(i)}$. Otherwise, we choose some tag set[2] $\bar{T} \in \mathcal{T}^\ell$ ourselves such that $g(\hat{T}, \bar{T}) = 0$ and sample some $\bar{x}$ from the set of elements in $\mathcal{X}$ not in the range of the hash function h. We then set $T^\diamond := \bar{T}$, $\mu^\diamond := \bar{\mu}$. We then compute $a \leftarrow \mathsf{Eval}^{func}_{pk}(g, A, T^\diamond)$, sample $u^\diamond \leftarrow D_{\mathcal{U}}$, and set $v \leftarrow f_{pk,a,\mu^\diamond}(u^\diamond)$; the signature for message $\mu^\diamond$ is $\sigma^\diamond = (u^\diamond, T^\diamond)$. By the distribution equivalence of inversion property of the PHTDF, this is statistically indistinguishable from having chosen $v \leftarrow \mathcal{V}$.

We have thus "programmed" in a signature for the tag/hashed message pair $(T^\diamond, x^\diamond)$. For the rest of the messages, $g(\hat{T}, T_i)$ is invertible, so $\mathcal{S}$ is able to compute signatures for hashed messages $x^{(i)} = h(\mu^{(i)})$ by making inversion queries on $v, x^{(i)}, T_i$, receiving back u_i and setting $\sigma_i = (u_i, T_i)$ to be the signature. $\mathcal{S}$

[2] Note that such a tag set must exist since g and $\hat{T}$ allow us to fulfill the second required property for the security reduction.

then sends the verification key $(pk_{\text{PHTDF}}, \{a_i\}, v)$ and the signatures σ_i to the adversary $\mathcal{A}$, thus successfully simulating the public key and signatures in a manner indistinguishable from an actual attack.

Now, conditioned on having made it this far, since the second needed property for the security reduction is satisfied by g and $\hat{T}$, with probability ϵ_2 the tag T^* used by the adversary in a successful forgery $(\sigma^* = (u^*, T^*), x^* = h(\mu^*))$ with $\mu^* \notin \{\mu^{(i)}\}_{i \in [Q]}$ will be such that $g(\hat{T}, T^*) = 0$. Letting $a^b \leftarrow \mathsf{Eval}_{pk}^{func}(g, \{a_i\}, T^{(b)})$ for $b \in \{*, \diamond\}$, we have that

$$f_{pk_{\text{PHTDF}}, a^*, x^*}(u^*) = f_{pk_{\text{PHTDF}}, a^\diamond, x^\diamond}(u^\diamond) = v,$$

and that $\mathsf{Prop}(u^*), \mathsf{Prop}(u^\diamond) \leq s$. If $x^\diamond = x^{(i)}$ for some i, then by the collision-resistance of h, $x^\diamond \neq x^*$, while if $x^\diamond$ was chosen from outside the range of h, then we immediately have that $x^\diamond \neq x^* = h(\mu^*)$. As a result, we have that $x^\diamond \neq x^*$, so that we have broken the $(\lambda, g, \hat{T})$-CRP security of the underlying PHTDF.□

5 Lattice-Based Instantiation of the Function g

We now give the main result of our paper, the lattice-based instantiation of a function g to be homomorphically evaluated in our signature scheme, and the method of choosing tags for it that allow us to fulfill the properties described in Section 4.1.

Theorem 5.1. *Let g be as defined in Algorithm 1, $c > 1$ be a constant. Let $d = \omega(\log \log n)$, and let $\beta \geq O(d^{2d} n^{11/2} \log^{13/2} q)$. Assume there exists an eu-scma adversary making Q signature queries that succeeds with probability ϵ against the signature scheme of Section 4 instantiated with the lattice-based PHTDF of Section 3, where $2Q^2/\epsilon \leq 2^{\lfloor c^d \rfloor}$. Then there exists an algorithm $\mathcal{S}^{\mathcal{A}}$ that solves $\mathsf{SIS}_{n,q,\beta}$ with probability $\frac{\delta^{1+c}}{4Q^{2c}} - \text{negl}(n)$.*

The remainder of this section is devoted to proving those parts of this theorem we have not already shown in Theorem 3.2 and Theorem 4.1.

5.1 Tag Instantiations

We define the explicit distribution $T \leftarrow D_{\mathcal{T}^\ell}$ used as the "tag" part of the signatures from Section 4.2, following the instantiation of Ducas and Micciancio. Recall that our tag space is, in the general setting, an embedding of GF(q^n) (represented as $R_q := \mathbb{Z}_q[x]/\langle f(x) \rangle$ for $f(x)$ irreducible over $\mathbb{Z}_q$) into $\mathbb{Z}_q^{n \times n}$. Each tag set consists of d elements, where $d = \omega(\log \log n)$. However, these d elements each correspond to specific subsets of coefficients of a degree $e := 2^d = O(n)$ polynomial in R_q (the tag prefixes below), so that in the signatures, we can represent the entire set of tags with a single polynomial in R_q. As a result, our signatures remain short.

Concretely, for the constant $c > 1$ mentioned in the theorem statement above, we define sets of tag prefixes $\mathcal{T}_i = \{0,1\}^{c_i}$ of lengths $c_0 = 0$, $c_i = \lfloor c^i \rfloor$ for $i \in [d]$. For*full tags* $t \in \mathcal{T} = \mathcal{T}_d$ and $i \leq d$, we write $t_{\leq i} \in \mathcal{T}_i$ for its prefix of length c_i and $t_{[i]}$ for the difference $t_{\leq i}(x) - t_{\leq i-1}(x) \in R_q$.

We technically have more freedom than the previous work when it comes to identifying tag prefixes with ring elements, as we do not need our tag prefixes to satisfy geometric properties. Instead, the critical requirement is that for any two distinct tag prefixes $t, t' \in \mathcal{T}_i$, the difference $t(X) - t'(X)$ is *invertible* over R_q. However, for simplicity, we may nonetheless use the identification described therein, where a tag prefix $t = [t_0, \ldots, t_{c_i - 1}] \in \mathcal{T}_i$ is mapped to the ring element $t(X) = \sum_{j < c_i} t_j X^j \in R_q$.

As in the Ducas-Micciancio scheme, to choose how to tag the homomorphic trapdoor matrices $\mathbf{A}_i$ in the public key, we have a confined guessing stage [BHJ+14] where we let $i^* \leq d$ be the smallest index such that $2Q^2/\epsilon \leq |\mathcal{T}_{i^*}|$, where Q is the number of signature queries made by the adversary and ϵ is the adversary's probability of success in the eu-scma security game. Note that such an index exists, since $2Q^2/\epsilon \leq 2^{\lfloor c^d \rfloor} = |\mathcal{T}_d|$. This choice of i^* guarantees that for Q tags $\{t^{(j)}\}_{j \in Q}$ chosen uniformly at random from $\mathcal{T}$, their prefixes $\{t^{(j)}_{\leq i^*}\}$ of length c_{i^*} will all be distinct except with probability at most $\delta/2$. For our construction of g, this guarantees that $g(\hat{T}, t^{(j)}) = 0$ for at most one tag (the first security property from Section 4.1). We then choose a prefix $t^{\diamond}_{\leq i^*} \in \mathcal{T}_{i^*}$ uniformly at random, with the hope that the tag t^* in the forgery output by the adversary is such that $t^{\diamond}_{\leq i^*} = t^*_{\leq i^*}$ (so that for our construction, we will have that $g(\hat{T}, t^*) = 0$, satisfying the second security property). By keeping the adversary's view statistically independent of our choice of prefix, we will have that $t^{\diamond}_{\leq i^*} = t^*_{\leq i^*}$ with probability $1/|\mathcal{T}_{i^*}|$. Following the proof of security for the generic signature scheme in Section 4.2, if one of the tags $t^{(j)}$ is such that $t^{(j)}_{\leq i^*} = t^{\diamond}_{\leq i^*}$, we set $t^{\diamond} = t^{(j)}$, and if not, we choose the full tag $t^{\diamond}$ uniformly at random from the set of tags with prefix $t^{\diamond}_{\leq i^*}$. Our choice of i^* guarantees that $2^{c_{i^*}-1} < \frac{2Q^2}{\delta} \leq 2^{c_{i^*}} = |\mathcal{T}_{i^*}|$. We also have that $c_{i^*} \leq c^{i^*} = c(c^{i^*-1}) < c(c_{i^*-1} + 1)$. As a result, we have that $|\mathcal{T}_{i^*}| \leq 2^{c(c_{i^*-1}+1)} \leq (\frac{4Q^2}{\delta})^c$, so that the second security property is fulfilled with probability at least $(\frac{\delta}{4Q^2})^c$.

In the various previous "vanishing trapdoor"-based signature schemes, the public key contained $d + 1$ trapdoor functions a_i, and the function g computed homomorphically was *linear* over the $d + 1$ associated tags $\hat{T} = (\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_d)$. Specifically, letting $T = (t_1, \ldots, t_d) \leftarrow D_{\mathcal{T}^d}$, the function g was

$$g(\hat{T}, T) := t_0 + \textstyle\sum_{i \in [d]} \hat{t}_i t_i.$$

To achieve the required properties for the actual scheme, they set $\hat{t}_0 = 1$, $\hat{t}_i = 0$ for $i \in [d]$, so that the homomorphically computed tag $g(\hat{T}, \cdot) = 1$ was always invertible. In the security reduction, they let $t^*_{\leq i^*}$ be the prefix of length i^* on which they hoped the adversary will forge. Then they set $\hat{t}_0 = -t^*_{\leq i^*}$, $\hat{t}_i = 1$ for $1 \leq i \leq i^*$ and $\hat{t}_i = 0$ for $i^* < i \leq d$.

In our scheme, we have only two trapdoor functions $a_0 = \mathbf{A}_0, \tilde{a} = \tilde{\mathbf{A}}$ in the public key, and compute *degree-*$(d-1)$ *polynomials*. We set the associated tags $\hat{t}_0, \tilde{t}$ as follows:

Actual Scheme: Let $\hat{t}_0 = 1$, $\tilde{t} = 0$.

Security Reduction: Let $t^*_{\leq i^*}$ be the prefix of length i^* on which we hope the adversary will forge. Then we let $\hat{t}_0 = -t^*_{\leq i^*}$, $\tilde{t} = i^*$.

5.2 Computation of g

To help compute g, in Gen we add to the public key d the coefficients of degree $d - 1$ polynomials $p_1, \ldots, p_d \in \mathbb{Z}_q[x]$ with the following behavior:

$$p_i(x) := \begin{cases} 1 & 1 \leq x \leq i \\ 0 & i < x \leq d \end{cases} = \sum_{j \in \{0, d-1\}} c_{ij} x^j$$

Since $\mathbb{Z}_q$ is a finite field, these polynomials have a unique realization as degree $d - 1$ polynomials over $\mathbb{Z}_q$, and their coefficients can easily be computed using Lagrange interpolation; see, i.e. [Sha79]. The total number of bits required to store the coefficients is $d^2 \log q = o(n)$, so they do not increase the asymptotics of the public key (and technically the coefficients may be computed on the fly from d and q alone).

We can now give a concrete definition of the function g in terms of input tags $\hat{t}_0, \tilde{t}$ (which will be associated with the trapdoors as discussed in the previous section) and $T = (t_1, \ldots, t_d)$. We have

$$g(\hat{T} = (\hat{t}_0, \tilde{t}), T) := \hat{t}_0 + \sum_{i\in[d]} (p_i(\tilde{t}) t_i).$$

To reduce the space required and the noise growth of the trapdoors when evaluating g homorphically on input $(\hat{T} = (\hat{t}_0, \tilde{t}), T \leftarrow D_{\mathcal{T}^d})$ using the PHTDF from Section 3, we instead compute g in a slightly different manner. In the clear, we view g in an alternative (but identical) form (recall that c_{ij} is the jth coefficient of the polynomial p_i described above):

$$g(\hat{T} = (\hat{t}_0, \tilde{t}), T) := \hat{t}_0 + \sum_{j=0}^{d-1} \left(\tilde{t}^j \sum_{i\in[d]} (c_{ij} t_i) \right).$$

Homomorphically, we evaluate g using Algorithm 1. In the algorithm, $\mathbf{S}_i$ is the trapdoor for $\mathbf{B}_i$, $\hat{t}_0$ is the tag for $\mathbf{A}_0$, $\tilde{t}$ is the tag for $\tilde{\mathbf{A}}$.

Algorithm 1 $\mathsf{Eval}^{func}_{pk}(g, \mathbf{A}_0, \tilde{\mathbf{A}}, T)$ and $\mathsf{Eval}^{td}_{pk}(g, (\mathbf{A}_0, \mathbf{R}_0), (\tilde{\mathbf{A}}, \tilde{\mathbf{R}}, T))$

```
$\mathbf{B}_0 \leftarrow (\sum_{i\in d} c_{i0} t_i)\mathbf{G}$                         $\mathbf{S}_0 \leftarrow \mathbf{0}$
$\mathbf{B}_1 \leftarrow \mathbf{G}$                                                $\mathbf{S}_1 \leftarrow \mathbf{0}$
for $j \in [d-1]$ do
    $\mathbf{B}_1 \leftarrow \tilde{\mathbf{A}} \boxdot \mathbf{B}_1$                              $\mathbf{S}_1 \leftarrow \tilde{\mathbf{R}}\mathbf{G}^{-1}(\mathbf{B}_1) + \tilde{t}\mathbf{S}_1$
    $\mathbf{B}_2 \leftarrow \mathbf{B}_1 \boxdot (\sum_{i\in[d]} c_{ij} t_i)$                 $\mathbf{S}_2 \leftarrow \mathbf{S}_1\mathbf{G}^{-1}(\sum_{i\in[d]} c_{ij} t_i \mathbf{G})$
    $\mathbf{B}_0 \leftarrow \mathbf{B}_0 \boxplus \mathbf{B}_2$                               $\mathbf{S}_0 \leftarrow \mathbf{S}_0 + \mathbf{S}_2$
end for
$\mathbf{B}_0 \leftarrow \mathbf{B}_0 \boxplus \mathbf{A}_0$                                   $\mathbf{S}_0 \leftarrow \mathbf{S}_0 + \mathbf{R}_0$
return $\mathbf{B}_0$                                          return $\mathbf{S}_0$
```

Correctness of the algorithm follows by inspection.

5.3 Noise Growth Analysis

We now proceed to analyze the noise growth of the trapdoor $\mathbf{S}_0$ for $\mathbf{B}_0$ in Algorithm 1. The following theorem covers the tag settings in both the actual scheme and the security reduction.

Theorem 5.2. *Let* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$ *for* $\bar{m} = O(n\log q)$. *Let* $r = \omega(\sqrt{\log n})$. *Sample* $(\mathbf{A}_0, \mathbf{R}_0) \leftarrow \mathsf{GenTrap}(\bar{\mathbf{A}}, \hat{t}_0\mathbf{I}, r)$ *for some* $\hat{t}_0 \in \mathbb{Z}_q$, $(\tilde{\mathbf{A}}, \tilde{\mathbf{R}}) \leftarrow \mathsf{GenTrap}(\bar{\mathbf{A}}, \tilde{t}\mathbf{I}, r)$ *for some* $\tilde{t} \le d \in \mathbb{Z}_q$. *Then the noise level in the trapdoor* $\mathbf{S}_0$ *for* $\mathbf{B}_0$ *returned by Algorithm 1 is at most* $d^d r O(n^{5/2}\log^3 q)$. *In particular, the function g as evaluated by Algorithm 1 is admissible with parameter* $s = d^d O(n^{5/2}\log^3 q)$.

Proof. Let $\mathbf{S}_i$ denote the trapdoor for $\mathbf{B}_i$ in Algorithm 1. We denote by $\mathbf{B}_{i,j}$, $\mathbf{S}_{i,j}$ the value of $\mathbf{B}_i, \mathbf{S}_i$ at the end of the jth iteration. We have that at the end of the jth iteration,

$$\mathbf{S}_{1,j} = \tilde{\mathbf{R}}(\sum_{\ell=0}^{j-1} \tilde{t}^{\ell}\mathbf{G}^{-1}(\mathbf{B}_{1,j-\ell-1})),$$

so that $s_1(\mathbf{S}_{1,j}) \le j \cdot d^{j-1}nk \cdot s_1(\mathbf{R}) \le d^j r \cdot O(n^{3/2}\log^2 q)$. As a result, we have that $s_1(\mathbf{S}_{2,j}) \le d^j r \cdot O(n^{5/2}\log^3 q)$. Since the final value of $\mathbf{S}_0$ is just $\sum_{j\in[d-1]} \mathbf{S}_{2,j} + \mathbf{R}_0$, we have that at the end of the algorithm,

$$s_1(\mathbf{S}_0) \le d^d r O(n^{5/2}\log^3 q). \qquad \square$$

References

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572. 2010.

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293. 1997.

[Ajt96] M. Ajtai. Generating hard instances of lattice problems. *Quaderni di Matematica*, 13:1–32, 2004. Preliminary version in STOC 1996.

[ASP14] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *CRYPTO (1)*, pages 297–314. 2014.

[BG14] S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In *Topics in Cryptology - CT-RSA 2014 - The Cryptographer's Track at the RSA Conference 2014, San Francisco, CA, USA, February 25-28, 2014. Proceedings*, pages 28–47. 2014.

[BGG$^+$14] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *EUROCRYPT*, pages 533–556. 2014.

[BHJ$^+$14] F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, pages 1–33, 2014.

[Boy10] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography*, pages 499–517. 2010.

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. 2011.

[BV14] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 1–12. 2014.

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012. Preliminary version in Eurocrypt 2010.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56. 2013.

[DM14] L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 335–352. 2014.

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. 2009.

[GLP12] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, pages 530–547. 2012.

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. 2008.

[GSW13] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, pages 75–92. 2013.

[GVW14] S. Gorbunov, V. Vaikuntanathan, and D. Wichs. Leveled fully homomorphic signatures from standard lattices. Cryptology ePrint Archive, Report 2014/897, 2014. `http://eprint.iacr.org/`.

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. 1998.

[HW09] S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *CRYPTO*, pages 654–670. 2009.

[KR00] H. Krawczyk and T. Rabin. Chameleon signatures. In *NDSS*. 2000.

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, pages 37–54. 2008.

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616. 2009.

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755. 2012.

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. Preliminary version in FOCS 2002.

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718. 2012.

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[Sha79] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[SS11] D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *EUROCRYPT*, pages 27–47. 2011.

[Wat09] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636. 2009.

[Xag13] K. Xagawa. Improved (hierarchical) inner-product encryption from lattices. In *Public-Key Cryptography - PKC 2013 - 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26 - March 1, 2013. Proceedings*, pages 235–252. 2013.