Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures

Dan Boneh* and David Mandell Freeman**

Stanford University, {dabo,dfreeman}@cs.stanford.edu

Abstract. We propose a linearly homomorphic signature scheme that authenticates vector subspaces of a given ambient space. Our system has several novel properties not found in previous proposals:

- It is the first such scheme that authenticates vectors defined over *binary fields*; previous proposals could only authenticate vectors with large or growing coefficients.
- It is the first such scheme based on the problem of *finding short vectors in integer lattices*, and thus enjoys the worst-case security guarantees common to lattice-based cryptosystems.

Our scheme can be used to authenticate linear transformations of signed data, such as those arising when computing mean and Fourier transform or in networks that use network coding. Our construction gives an example of a cryptographic primitive — homomorphic signatures over $\mathbb{F}_2$ — that can be built using lattice methods, but cannot currently be built using bilinear maps or other traditional algebraic methods based on factoring or discrete log type problems.

Security of our scheme (in the random oracle model) is based on a new hard problem on lattices, called k-SIS, that reduces to standard average-case and worst-case lattice problems. Our formulation of the k-SIS problem adds to the "toolbox" of lattice-based cryptography and may be useful in constructing other lattice-based cryptosystems.

As a second application of the new k-SIS tool, we construct an ordinary signature scheme and prove it k-time unforgeable in the standard model assuming the hardness of the k-SIS problem. Our construction can be viewed as "removing the random oracle" from the signatures of Gentry, Peikert, and Vaikuntanathan at the expense of only allowing a small number of signatures.

Keywords. Lattice-based cryptography, homomorphic signatures.

1 Introduction

A *linearly homomorphic signature scheme* signs n-dimensional vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ defined over some finite field $\mathbb{F}_p$ and outputs one signature per vector. The linear homomorphic property is that given these k signatures, anyone can produce a signature on any vector $\mathbf{v}$ in the $\mathbb{F}_p$-linear span of $\mathbf{v}_1, \ldots, \mathbf{v}_k$. The signature is

* Supported by NSF.
** Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

secure if it is difficult to produce a signature on any vector $\mathbf{v} \in \mathbb{F}_p^n$ outside the linear span of $\mathbf{v}_1, \ldots, \mathbf{v}_k$. We give precise definitions in Section 2.

The original motivation for linearly homomorphic signatures comes from the *network coding* routing mechanism [12, 11, 25, 4, 13]. In a computer network that uses network coding, a message sender signs a number of "augmented" message vectors and transmits the resulting vector-signature pairs through the network to a recipient. Each router along the way receives a number of signed vectors and creates a random linear combination $\mathbf{v}$ of the vectors it receives. The router uses the homomorphic property to derive a signature on $\mathbf{v}$ and forwards $\mathbf{v}$ and its signature to the next router, which then does the same with the signed vectors it receives. The ultimate recipient obtains several random linear combinations of the original message vectors, discards all vectors that are not properly signed, and recovers the original message by solving a full-rank linear system over $\mathbb{F}_p$. Security of the signature scheme ensures that the recipient obtains the originally transmitted message vectors. In implementations there is a desire to use network coding with addition over $\mathbb{F}_2$, so that computations on messages are simple XORs and decoding amounts to solving a linear system over $\mathbb{F}_2$.

Beyond network coding, linearly homomorphic signatures enable linear computations on authenticated data. For example, consider a server that stores signed data samples $s_1, \ldots, s_n$ in $\mathbb{F}_p$. The signature on sample s_i is actually a signature on the vector $(s_i|\mathbf{e}_i) \in \mathbb{F}_p^{n+1}$, where $\mathbf{e}_i$ the ith unit vector in $\mathbb{F}_p^n$. The server stores (i, s_i) and a signature on $(s_i|\mathbf{e}_i)$. (The vector $\mathbf{e}_i$ need not be stored with the data and can be reconstructed from i when needed.) Using the homomorphic property, the server can compute a signature σ on the sum $(\sum_{i=1}^n s_i, 1, \ldots, 1)$. If σ reveals no other information about the original samples, then the server can publish the sum $\sum_{i=1}^n s_i$ and the signature σ on the sum while maintaining privacy of the original data. The "augmentation" $(1, \ldots, 1)$ proves that the published message really is the claimed sum of the original samples.[1] More generally, the server can publish an authenticated inner product of the samples $\mathbf{s} := (s_1, \ldots, s_n)$ with any known vector $\mathbf{c} \in \mathbb{F}_p^n$ without leaking additional information about the samples. This is needed, for example, to publish an authenticated Fourier coefficient from the Fourier transform of $\mathbf{s}$. It can also be used to compute an authenticated least squares fit for a set of signed data.

Previous results on linearly homomorphic signatures make use of groups in which the discrete logarithm problem is hard [18, 11, 25, 4] or the RSA assumption holds [13]. In the former case, signatures are linearly homomorphic over $\mathbb{F}_p$ for some large p, while in the latter case, signatures are homomorphic over the integers (with some bound on the size of the coefficients allowed in linear

[1] Strictly speaking, in order to prevent mix-and-match attacks between different data sets one needs to link the n samples with a random tag that uniquely identifies the data set.

combinations). In particular, no previous scheme can support linear operations over a small field such as $\mathbb{F}_2$. This appears to be an inherent limitation of discrete log-type systems, since the discrete log problem is not hard in $\mathbb{F}_2$. A similar limitation prevents an RSA-based system over $\mathbb{F}_2$.

More distantly related to our work is the notion of "redactable" signatures [24, 16, 15, 3, 22, 21, 10, 8, 7]. These schemes have the property that given a signature on a message, anyone can derive a signature on subsets of the message. Our focus here is quite different — we look at linear operations on tuples of authenticated vectors rather than a subset operation on a single message.

Our contributions.

- **Homomorphic signatures over $\mathbb{F}_2$:** We construct the first unforgeable linearly homomorphic signature scheme that authenticates vectors with coordinates in $\mathbb{F}_2$. Our construction gives an example of a cryptographic primitive that can be built using lattice methods, but cannot currently be built using bilinear maps or other traditional algebraic methods based on factoring or discrete log type problems. Our scheme can be easily modified to authenticate vectors with coefficients in other small fields, including prime fields and extension fields such as $\mathbb{F}_{2^d}$. In addition, our scheme is *private*, in the sense that a derived signature on a vector $\mathbf{v}$ leaks no information about the original signed vectors beyond what is revealed by $\mathbf{v}$.
- **A simple k-time signature without random oracles:** We describe a stateless signature scheme and prove it secure in the standard model when used to sign at most k messages, for small values of k. The public key of our scheme is significantly smaller than that of any other stateless lattice-based signature scheme that can sign multiple large messages and is secure in the standard model. Our construction can be viewed as "removing the random oracle" from the signature scheme of Gentry, Peikert, and Vaikuntanathan [14], but only for signing k messages.
- **New tools for lattice-based signatures:** Unforgeability of both of our schemes is based on a new hard problem on lattices, which we call the *k-Small Integer Solutions* (k-SIS) problem. We show that k-SIS reduces to the standard *Small Integer Solution* (SIS) problem, which is known to be as hard as standard worst-case lattice problems [20].

  Unforgeability of our k-time signature scheme depends on bounds for the length of vectors sampled from discrete Gaussian distributions. We prove both upper and lower bounds that are essentially as tight as possible. Our upper bound improves on a result of Micciancio and Regev [20, Lemma 4.4], and our lower bound is (to our knowledge) the first such bound in the literature.

Privacy of our linearly homomorphic scheme depends on a new result on discrete Gaussian distributions, namely, that the distribution of a sum of samples from a discrete Gaussian is statistically close to a discrete Gaussian distribution that depends *only on the sum* and not on the individual samples. While the analogous result for *continuous* Gaussians is well known, this is (to our knowledge) the first such result for discrete Gaussians.

Overview of the homomorphic signature scheme. Our construction builds on the signature scheme of Gentry, Peikert, and Vaikuntanathan [14], in which signatures are short vectors σ in lattices defined modulo some large integer q. The key idea in our construction is to use short vectors σ in (cosets of) lattices defined modulo $2q$, which allows us to encode different information modulo 2 and modulo q: $\sigma \bmod 2$ encodes information about the vector being signed, while $\sigma \bmod q$ encodes a solution to a hard problem, ensuring that an adversary cannot forge the signature.

The fact that σ is a short *integer* vector ensures that the two parts cannot be attacked independently. Specifically, applying the Chinese remainder theorem to two vectors σ_2 and σ_q that are correct mod 2 and mod q, respectively, does not produce a short integer vector. This property appears to be unique to lattice-based cryptography: if we attempted a similar construction in discrete log groups of order $2q$, we would easily be able to attack the order 2 and order q parts independently.

Concretely, our construction works as follows. Let q be an odd prime. To sign a vector subspace $V = \text{span}(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ of $\mathbb{F}_2^n$, we define a matrix $\mathbf{A}_V \in \mathbb{Z}_{2q}^{m \times n}$ and then sign each basis vector $\mathbf{v}_i$. A signature on $\mathbf{v}_i \in \mathbb{F}_2^n$ is a low-norm vector $\sigma_i \in \mathbb{Z}^m$ such that $\mathbf{A}_V \cdot \sigma_i = q \cdot \mathbf{v}_i \mod 2q$. A signature $\sigma \in \mathbb{Z}^m$ on a vector $\mathbf{y} \in \mathbb{F}_2^n$ is valid if σ has small norm and $\mathbf{A}_V \cdot \sigma = q \cdot \mathbf{y} \bmod 2q$.

Producing such a signature requires knowing a short basis for the integer lattice defined by the kernel of $\mathbf{A}_V$; to obtain such a basis we combine the trapdoor generation algorithm of Alwen and Peikert [2] with the basis delegation mechanism of Cash, Hofheinz, Kiltz, and Peikert [9].

The homomorphic property of our scheme is now immediate: if we are given arbitrary vector-signature pairs $(\mathbf{v}_j, \sigma_j) \in \mathbb{F}_2^n \times \mathbb{Z}^m$ for $j = 1, \ldots, \ell$, we can create a signature on $\mathbf{v} = \mathbf{v}_1 + \cdots + \mathbf{v}_\ell \in \mathbb{F}_2^n$ by computing $\sigma = \sigma_1 + \cdots + \sigma_\ell \in \mathbb{Z}^m$. Since the σ_j are all valid signatures on the $\mathbf{v}_j$, we see that $\mathbf{A}_V \cdot \sigma = q \cdot \mathbf{v} \bmod 2q$ and σ has low norm (if ℓ is sufficiently small), so σ is a valid signature on $\mathbf{v}$.

Security and the k-SIS problem. To prove unforgeability, we need to show that given signatures on basis vectors of V, it is impossible to generate a signature

on a vector outside of V. To do so we define the k-SIS problem, which, roughly speaking, is as follows:

> Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and k short vectors $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \mathbb{Z}^m$ satisfying $\mathbf{A} \cdot \mathbf{e}_i = 0 \bmod q$, find a short vector $\mathbf{e} \in \mathbb{Z}^m$ satisfying $\mathbf{A} \cdot \mathbf{e} = 0 \bmod q$, such that $\mathbf{e}$ is not in $\mathbb{Q}$-span$(\{\mathbf{e}_1, \ldots, \mathbf{e}_k\})$.

When $k = 0$ this is the standard SIS problem [20].

In Section 5 we show that an adversary that breaks the homomorphic signature scheme (defined mod $2q$) in the random oracle model can be used to solve the k-SIS problem (defined mod q). In Section 4 we show that the k-SIS problem is as hard as the SIS problem. Our reduction degrades exponentially in k, which forces us to use a constant-size k if we want our linearly homomorphic scheme to be provably secure based on worst-case lattice problems. It is an important open problem to give either a tighter reduction to SIS or a direct reduction from k-SIS to worst-case lattice problems.

For some applications of linearly homomorphic signatures it is desirable that the derived signatures be private; that is, a derived signature on a vector $\mathbf{v}$ in span$(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ should not leak information about the original vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ beyond what is revealed by $\mathbf{v}$. For our construction, to prove privacy it suffices to show that the distribution obtained by summing independent discrete Gaussians depends only on the coset of the sum and the linear combination being computed. We prove this statement in Section 4.

Overview of the k-time signature scheme. Our goal is to use the same mechanism as in the homomorphic signature scheme to construct an ordinary signature scheme. Since homomorphic signatures are not existentially unforgeable, we must find a way to remove the homomorphic property. To do this, we impose the requirement that the length of a valid signature $\sigma \in \mathbb{Z}^m$ be very close to the expected length of the vector produced by the signing algorithm. We then show that any linear combination of valid signatures will be too long to satisfy this tight bound, so the homomorphic property is of no use to an adversary.

As with the homomorphic scheme, the security of our k-time signature scheme follows from hardness of the k-SIS problem. We prove security (in the standard model) against a static attacker, who submits all of his message queries before receiving the public key. By a standard transformation using *chameleon hashes* [17], this implies the existence of a scheme secure against an adaptive attacker, also in the standard model. Our security proof requires tight bounds on the length of a vector sampled from a discrete Gaussian distribution. We use new upper and lower bounds that are essentially as tight as possible.

Lyubashevsky and Micciancio [19] give a lattice-based one-time signature scheme secure in the standard model and show how it can be converted to sign k

messages. For small values of k, our construction gives a more efficient stateless k-time signature than is produced by this conversion.

Outline of the paper. Section 2 gives a formal definition and security model for linearly homomorphic signatures. In Section 3 we review facts about lattices. Section 4 describes the new tools we use in our constructions, including the k-SIS problem and our reduction of k-SIS to SIS. In Section 5 we present our homomorphic scheme, and in Section 6 we present our k-time signature scheme. Finally, in Section 7 we describe extensions of our scheme and open questions. Because of space considerations, proofs have been omitted and can be found in the full version of this paper [6].

2 Linearly Homomorphic Signatures

We define linearly homomorphic signatures over any principal ideal domain R. These signatures authenticate tuples (a.k.a. vectors) of elements of R. This definition encompasses the homomorphic signatures over finite fields defined by Boneh et al. [4] as well as the signatures over $\mathbb{Z}$ and $\mathbb{Z}_N$ defined by Gennaro et al. [13]. While we describe the system in terms of a fixed ring R, it may be that R is determined by the Setup algorithm, as in the case where the size of R depends on the system's security parameter.

To prevent "mix-and-match" attacks, each set of vectors signed is given a unique identifier id, which serves to tie together all vectors that belong to the same file or data set. Our security model requires that this identifier be unpredictable; in our scheme it is chosen at random by the signer.

Definition 1 (adapted from [4]). Let R be a principal ideal domain. A *linearly homomorphic signature scheme over* R is a tuple of probabilistic, polynomial-time algorithms $(\mathsf{Setup}, \mathsf{Sign}, \mathsf{Combine}, \mathsf{Verify})$ with the following functionality:

- $\mathsf{Setup}(n, \mathsf{params})$. On input a security parameter n (in unary) and additional public parameters params that include the dimension N of the ambient space and the dimension k of subspaces to be signed, this algorithm outputs a public key pk and a secret key sk.
- $\mathsf{Sign}(\mathsf{sk}, \mathsf{id}, \mathbf{v})$. On input a secret key sk, an identifier $\mathsf{id} \in \{0,1\}^n$, and a vector $\mathbf{v} \in R^N$, this algorithm outputs a signature σ.
- $\mathsf{Combine}(\mathsf{pk}, \mathsf{id}, \{(\alpha_i, \sigma_i)\}_{i=1}^{\ell})$. On input a public key pk, an identifier id, and a set of tuples $\{(\alpha_i, \sigma_i)\}_{i=1}^{\ell}$ with $\alpha_i \in R$, this algorithm outputs a signature σ. (This σ is intended to be a signature on $\sum_{i=1}^{\ell} \alpha_i \mathbf{v}_i$.)
- $\mathsf{Verify}(\mathsf{pk}, \mathsf{id}, \mathbf{y}, \sigma)$. On input a public key pk, an identifier $\mathsf{id} \in \{0,1\}^n$, a vector $\mathbf{y} \in R^N$, and a signature σ, this algorithm outputs either 0 (reject) or 1 (accept).

We require that for each $(\mathsf{pk}, \mathsf{sk})$ output by $\mathsf{Setup}(n, \mathsf{params})$, we have:

1. For all id and $\mathbf{y} \in R^N$, if $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{id}, \mathbf{y})$ then $\mathsf{Verify}(\mathsf{pk}, \mathsf{id}, \mathbf{y}, \sigma) = 1$.
2. For all $\mathsf{id} \in \{0,1\}^n$ and all sets of triples $\{(\alpha_i, \sigma_i, \mathbf{v}_i)\}_{i=1}^{\ell}$, if it holds that $\mathsf{Verify}(\mathsf{pk}, \mathsf{id}, \mathbf{v}_i, \sigma_i) = 1$ for all i, then

$$\mathsf{Verify}\left(\mathsf{pk},\ \mathsf{id},\ \textstyle\sum_i \alpha_i \mathbf{v}_i,\ \mathsf{Combine}\left(\mathsf{pk},\ \mathsf{id},\ \{(\alpha_i, \sigma_i)\}_{i=1}^{\ell}\right)\right) = 1.$$

In our lattice-based linearly homomorphic signature scheme, we cannot combine arbitrarily many valid signatures and still guarantee successful verification. We capture this property by saying that the scheme is *L-limited* if correctness property (2) holds for all $\ell \leq L$ whenever the σ_i are output by the Sign algorithm.

Unforgeability. The security model for linearly homomorphic signatures allows an adversary to make adaptive signature queries on files of his choosing, with the signer randomly choosing the identifier id for each file queried. The winning condition captures the fact that there are two distinct types of forgeries: a vector-signature pair $(\mathbf{y}^*, \sigma^*)$ that verifies for some file *not* queried to the signer (a *type 1 forgery*), or a pair $(\mathbf{y}^*, \sigma^*)$ that verifies for some file that *was* queried to the signer, but for which $\mathbf{y}^*$ is not a linear combination of the vectors queried (a *type 2 forgery*).

Definition 2 (adapted from [4]). A homomorphic signature scheme $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Combine}, \mathsf{Verify})$ over R is *unforgeable* if the advantage of any probabilistic, polynomial-time adversary $\mathcal{A}$ in the following security game is negligible in the security parameter n:

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

Queries: Proceeding adaptively, $\mathcal{A}$ specifies a sequence of k-dimensional subspaces $V_i \subset R^N$, represented as a k-tuples of basis vectors $\mathbf{v}_{i1}, \ldots, \mathbf{v}_{ik}$. For each i, the challenger chooses id_i uniformly from $\{0,1\}^n$ and gives to $\mathcal{A}$ the identifier id_i and the j signatures $\sigma_{ij} \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{id}_i, \mathbf{v}_{ij})$ for $j = 1, \ldots, k$.

Output: $\mathcal{A}$ outputs $\mathsf{id}^* \in \{0,1\}^n$, a *non-zero* vector $\mathbf{y}^* \in R^N$, and a signature σ^*.

The adversary *wins* if $\mathsf{Verify}(\mathsf{pk}, \mathsf{id}^*, \mathbf{y}^*, \sigma^*) = 1$, and either (1) $\mathsf{id}^* \neq \mathsf{id}_i$ for all i (a *type 1 forgery*), or (2) $\mathsf{id}^* = \mathsf{id}_i$ for some i but $\mathbf{y}^* \notin V_i$ (a *type 2 forgery*). The *advantage* $\mathrm{HomSig\text{-}Adv}[\mathcal{A}, \mathcal{S}]$ of $\mathcal{A}$ is defined to be the probability that $\mathcal{A}$ wins the game.

Privacy. Given signatures on vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ in R^N, it is desirable that derived signatures on a vector $\mathbf{v}$ in span$(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ not leak any information about $\mathbf{v}_1, \ldots, \mathbf{v}_k$ beyond what is revealed by $\mathbf{v}$. We are not trying to hide the fact the derivation took place or the function that was used to compute $\mathbf{v}$, merely the inputs to the function.

More precisely, we define privacy for linearly homomorphic signatures using a variation of a definition from [8]. The definition captures the idea that given signatures on a number of derived vectors in one of two different vector spaces, the attacker cannot tell which space the derived signatures came from. This indistinguishability holds even if the secret key is leaked. We call signatures with this property *weakly context hiding*. The reason for "weak" is that we are not hiding the fact that derivation took place or the computed function and we assume the original signatures are not public. Ahn et al. [1] define a stronger notion of privacy, called *strong context hiding*, that requires that derived signatures be distributed as independent fresh signatures on the same message; this requirement ensures privacy even if the original signatures are exposed.

Definition 3. A homomorphic signature scheme $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Combine}, \mathsf{Verify})$ over R is *weakly context hiding* if the advantage of any probabilistic, polynomial-time adversary $\mathcal{A}$ in the following security game is negligible in the security parameter n:

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

Challenge: $\mathcal{A}$ outputs $(V_0, V_1, f_1, \ldots, f_s)$ where V_0 and V_1 are linear spaces over R^N represented as k-tuples of vectors $(\mathbf{v}_1^{(b)}, \ldots, \mathbf{v}_k^{(b)})$ for $b = 0, 1$. The functions $f_1, \ldots, f_s$ are R-linear functions[2] on $(R^N)^k$ satisfying

$$f_i\big(\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_k^{(0)}\big) = f_i\big(\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_k^{(1)}\big) \qquad \text{for all } i = 1, \ldots, s.$$

In response, the challenger generates a random bit $b \in \{0, 1\}$ and a random tag $\tau \in \{0, 1\}^n$ and signs the vector space V_B using the tag τ. Next, for $i = 1, \ldots, s$ the challenger uses $\mathsf{Combine}$ to derive signatures σ_i on $f_i(\mathbf{v}_1^{(b)}, \ldots, \mathbf{v}_k^{(b)})$ and sends $\sigma_1, \ldots, \sigma_s$ to $\mathcal{A}$. The functions $f_1, \ldots, f_s$ can be output adaptively after V_0, V_1 are output.

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

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

[2] If the scheme is L-limited, we require the f_i to have at most L nonzero coefficients.

Winning the context hiding game game means that the attacker was able to determine whether the challenge signatures were derived from signatures on V_0 or from signatures on V_1. We note that for discrete log-based linearly homomorphic signatures such as those of [4], weak context hiding follows from the uniqueness of the signature.

3 Background on Lattices

In this section we describe the lattices we will be using and their properties. Precise statements of these results can be found in the full version of this paper [6].

Notation. For any integer $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers modulo q. When q is prime, $\mathbb{Z}_q$ is a field and is sometimes denoted $\mathbb{F}_q$. We let $\mathbb{Z}_q^{n\times m}$ denote the set of $n \times m$ matrices with entries in $\mathbb{Z}_q$. We denote matrices by capital boldface letters and vectors by lowercase boldface letters. We say a function $f : \mathbb{Z} \to \mathbb{R}^+$ is *negligible* if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathrm{negl}(n)$ to denote a negligible function of n. The function $\lg x$ is the base 2 logarithm of x.

Lattices. An m-dimensional lattice Λ is a full-rank discrete subgroup of $\mathbb{R}^m$. We will be interested in *integer lattices* Λ, i.e., those whose points have coordinates in $\mathbb{Z}^m$. For any integer $q \geq 2$ and any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define

$$\Lambda_q^{\perp}(\mathbf{A}) := \big\{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\cdot\mathbf{e} = \mathbf{0} \bmod q\big\},\ \Lambda_q^{\mathbf{u}}(\mathbf{A}) := \big\{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\cdot\mathbf{e} = \mathbf{u} \bmod q\big\}.$$

Alwen and Peikert [2, Theorem 3.2] describe an algorithm $\mathsf{TrapGen}$ that outputs an (almost) uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with a "short" basis for $\Lambda_q^{\perp}(\mathbf{A})$. Cash et al. [9, Lemma 3.2] give an algorithm $\mathsf{ExtBasis}$ that "delegates" a basis: given two matrices $\mathbf{A}, \mathbf{A}'$ in $\mathbb{Z}_q^{n\times m}$ and a short basis for $\Lambda_q^{\perp}(\mathbf{A})$, the algorithm computes a short basis for $\Lambda_q^{\perp}(\mathbf{A}\|\mathbf{A}')$. The length of a basis $\mathbf{T}$ is measured by the *Gram-Schmidt norm* and denoted by $\|\widetilde{\mathbf{T}}\|$.

For any real $\epsilon > 0$, the *smoothing parameter* $\eta_\epsilon(\Lambda)$ is defined to be the smallest positive s such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \epsilon$ [20].

Gaussian distributions. Let L be a subset of $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any positive parameter $\sigma \in \mathbb{R}_{>0}$, let $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) := \exp\left(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2\right)$ be a Gaussian function on $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ. Let $D_{\sigma,\mathbf{c}}$ be the continuous Gaussian distribution over $\mathbb{R}^m$ with center $\mathbf{c}$ and parameter σ, with $D_{\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\sigma^n$. Let $\rho_{\sigma,\mathbf{c}}(L) := \sum_{\mathbf{x}\in L}\rho_{\sigma,\mathbf{c}}(\mathbf{x})$ be the discrete integral of $\rho_{\sigma,\mathbf{c}}$ over L. Finally, let $\mathcal{D}_{L,\sigma,c}$ be the discrete Gaussian distribution over L with center $\mathbf{c}$ and parameter σ. In particular, for all $\mathbf{y} \in L$, we have $\mathcal{D}_{L,\sigma,\mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{y})}{\rho_{\sigma,\mathbf{c}}(L)}$. For notational convenience, $\rho_{\sigma,\mathbf{0}}$ and $\mathcal{D}_{L,\sigma,\mathbf{0}}$ are abbreviated as ρ_σ and $\mathcal{D}_{L,\sigma}$, respectively.

Gentry, Peikert, and Vaikuntanathan [14, Theorems 4.1 and 5.9] describe algorithms SampleGaussian and SamplePre that output vectors sampled from distributions statistically close to $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}),\sigma,\mathbf{c}}$ and $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}),\sigma}$, respectively.

Hardness assumption. The security of our signature schemes is based on the problem of finding short vectors in $\Lambda_q^\perp(\mathbf{A})$ for random $\mathbf{A}$. This is known as the *Small Integer Solution* (SIS) problem, and is defined as follows.

Definition 4. An instance of the $\mathsf{SIS}_{q,m,\beta}$ problem is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. A solution to the problem is a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \leq \beta$ and $\mathbf{A}\cdot\mathbf{v} = \mathbf{0} \bmod q$ (i.e., $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$). If $\mathcal{B}$ is an algorithm that takes as input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define $\mathsf{SIS}\text{-}\mathrm{Adv}[\mathcal{B},(q,m,\beta)]$ to be the probability that $\mathcal{B}$ outputs a solution to a uniformly random $\mathsf{SIS}_{q,m,\beta}$ problem instance $\mathbf{A}$.

Micciancio and Regev [20] and Gentry et al. [14] show that the (average case) SIS problem for $\beta = \mathrm{poly}(n)$ is hard assuming worst-case hardness of certain standard approximation problems on lattices.

4 New Tools

4.1 A "One-More" SIS Problem

The security of most lattice-based signature schemes depends on the adversary's inability to find a short vector in $\Lambda_q^\perp(\mathbf{A})$ for some public matrix $\mathbf{A}$. However, for our linearly homomorphic signatures this criterion is insufficient. Roughly speaking, an adversary in our scheme will be given several short vectors $\mathbf{e}_1,\ldots,\mathbf{e}_k \in \Lambda_q^\perp(\mathbf{A})$ and must produce a short vector in $\Lambda_q^\perp(\mathbf{A})$ that is not in the span of the $\mathbf{e}_i$. This is a "one-more" variant of the standard SIS problem, analogous to the "one-more discrete logarithm" problem in group-based cryptography (see e.g., [23]). We will see in Section 4.3 that for certain choices of parameters the problem is equivalent to finding *any* short vector in $\Lambda_q^\perp(\mathbf{A})$ distinct from $\{\pm\mathbf{e}_i\}$, making the "one-more" analogy even more appropriate. We now formally define the problem.

Definition 5. For any integer $k \geq 0$, an instance of the $k\text{-}\mathsf{SIS}_{q,m,\beta,\sigma}$ problem is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a set of k vectors $\mathbf{e}_1,\ldots,\mathbf{e}_k \in \Lambda_q^\perp(\mathbf{A})$. A solution to the problem is a nonzero vector $\mathbf{v} \in \mathbb{Z}^m$ such that

1. $\|\mathbf{v}\| \leq \beta$,
2. $\mathbf{A}\cdot\mathbf{v} = \mathbf{0} \bmod q$ (i.e., $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$), and
3. $\mathbf{v} \notin \mathbb{Q}\text{-span}(\{\mathbf{e}_1,\ldots,\mathbf{e}_k\})$.

If $\mathcal{B}$ is an algorithm that takes as input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and vectors $\mathbf{e}_i \in \mathbb{Z}^m$ for $i = 1,\ldots,k$, we define $k\text{-}\mathsf{SIS}\text{-}\mathrm{Adv}[\mathcal{B},(q,m,\beta,\sigma)]$ to be the probability that $\mathcal{B}$ outputs a solution to a $k\text{-}\mathsf{SIS}_{q,m,\beta,\sigma}$ problem instance $(\mathbf{A},\mathbf{e}_1,\ldots,\mathbf{e}_k)$ over uniformly random $\mathbf{A}$ in $\mathbb{Z}_q^{n\times m}$ and $\mathbf{e}_i$ drawn from the distribution $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}),\sigma}$.

The main result of this section is to show that an adversary $\mathcal{A}$ that solves the k-SIS problem in dimension m can be used to solve the SIS problem in dimension $m - k$. More precisely, we have the following:

Theorem 6. *Let q be a prime, and let m, β, σ, and k, be polynomial functions of a security parameter n. Suppose that $m \geq 2n \lg q$, $m/k > n$, $\sigma > \omega(\sqrt{\log m})$, $t > \omega(\sqrt{\log n})$, and $q > \sigma \cdot \omega(\sqrt{\log m})$.*

Let $\beta' = \beta \cdot (k^{3/2} + 1)k!(t\sigma)^k$. Let $\mathcal{A}$ be a polynomial-time adversary for the k-$\mathsf{SIS}_{q,m,\beta,\sigma}$ problem. Then there exists a polynomial-time algorithm $\mathcal{B}$ that solves $\mathsf{SIS}_{q,m-k,\beta'}$, such that

$$\mathsf{SIS}\text{-}\mathrm{Adv}[\mathcal{B}, (q, m-k, \beta')] \geq k\text{-}\mathsf{SIS}\text{-}\mathrm{Adv}[\mathcal{A}, (q, m, \beta, \sigma)] - \mathrm{negl}(n).$$

Since the SIS problem is only assumed to be hard for parameters $\beta = \mathrm{poly}(n)$, the fact that the above reduction degrades exponentially in k means that k must be chosen to be small enough so that β' is still polynomial in n. In our application the parameter σ is $\omega(\sqrt{n})$, which means that k must be chosen to be $O(1)$. In this case, if we take $t = O(\log \sigma)$ and $\beta' = \beta \cdot O(\sigma^k \log^k \sigma)$, then Theorem 6 shows that if the $\mathsf{SIS}_{q,m-k,\beta'}$ problem is hard, then the k-$\mathsf{SIS}_{q,m,\beta,\sigma}$ problem is also hard.

The idea of the proof of Theorem 6 is as follows: given an SIS challenge $\mathbf{A}' \in \mathbb{Z}_q^{n \times (m-k)}$, we can choose k random vectors $\mathbf{e}_i$ from a Gaussian distribution over $\mathbb{Z}^m$ and append k columns to $\mathbf{A}'$ to create a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ such that the $\mathbf{e}_i$ are in $\Lambda_q^\perp(\mathbf{A})$. If the k-SIS adversary $\mathcal{A}$ outputs a short vector $\mathbf{e}^* \in \Lambda_q^\perp(\mathbf{A})$ that is $\mathbb{Q}$-linearly independent of the $\{\mathbf{e}_i\}$, then we can use Gaussian elimination over $\mathbb{Z}$ (via Cramer's rule) to compute a short vector $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$ with zeroes in the last k entries. Reading off the first $m - k$ entries of $\mathbf{v}$ gives us a short vector in $\Lambda_q^\perp(\mathbf{A}')$. Details can be found in the full version of this paper [6].

4.2 Tight Bounds on the Length of Gaussian Samples

Signatures in our schemes will be "short" vectors sampled from Gaussian distributions over cosets of a particular lattice. To quantify what "short" means, we must demonstrate an upper bound on the length of a vector sampled from a Gaussian.

Micciancio and Regev [20, Lemma 4.4] show that if σ is larger than the smoothing parameter of the n-dimensional lattice Λ, then with overwhelming probability the length of a vector sampled from $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ is at most $\sigma\sqrt{n}$. They also show that the *expected* length of such a sample is at most $\sigma\sqrt{n/2\pi} + \mathrm{negl}(n)$. Our result below "bridges the gap" between the upper bound and the expected length. Furthermore, we show an equally strong *lower* bound on the length of the Gaussian sample.

Proposition 7. *Let $\Lambda \subset \mathbb{R}^n$ be a lattice. Suppose $\sigma \geq \eta_\epsilon(\Lambda)$ for some negligible ϵ. Let $\mathbf{c} \in \mathbb{R}^m$ be any vector. Then for any constant $\alpha > 0$ we have*

$$\Pr\left[(1-\alpha)\sigma\sqrt{\tfrac{n}{2\pi}} \leq \|\mathbf{x}-\mathbf{c}\| \leq (1+\alpha)\sigma\sqrt{\tfrac{n}{2\pi}} : x \xleftarrow{\mathrm{R}} \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}\right] \;\geq\; 1-\mathrm{negl}(n).$$

4.3 Removing Linear Independence from the k-SIS Problem

We now show that for small values of k and tight length bounds, we can relax the linear independence condition in the statement of the k-SIS problem. Specifically, if we can find *any* nonzero vector $\mathbf{e}^*$ of the required length not equal to $\pm\mathbf{e}_i$ for any of the k vectors $\mathbf{e}_i$ in the problem statement, then with overwhelming probability $\mathbf{e}^*$ is not in the linear span of the $\mathbf{e}_i$.

Proposition 8. *Suppose $m \geq 2n \lg q$ and $k \cdot \omega(\sqrt{\log n}) < \min(\sigma, m^{1/4})$. Let $(\mathbf{A}, \mathbf{e}_1, \ldots, \mathbf{e}_k)$ be a k-SIS challenge with $\mathbf{A}$ chosen uniformly at random from $\mathbb{Z}_q^{n\times m}$ and $\mathbf{e}_i$ sampled from $\mathcal{D}_{\Lambda_q^\perp(\mathbf{A}),\sigma}$. Then with overwhelming probability, the only nonzero vectors of length at most $1.1 \cdot \sigma\sqrt{m/2\pi}$ in $\mathbb{Q}$-span$(\mathbf{e}_1, \ldots, \mathbf{e}_k)$ are the vectors $\pm\mathbf{e}_i$ for $i = 1, \ldots, k$.*

4.4 Linear Combinations of Discrete Gaussians

The privacy property of our linearly homomorphic scheme will follow from the fact that the distribution obtained by summing independent discrete Gaussian samples is itself a discrete Gaussian distribution that depends only on the coset of the sum and the linear combination being computed.

Theorem 9. *Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice and $\sigma \in \mathbb{R}$. For $i = 1, \ldots, k$ let $\mathbf{t}_i \in \mathbb{Z}^m$ and let X_i be mutually independent random variables sampled from $\mathcal{D}_{\Lambda+\mathbf{t}_i,\sigma}$. Let $\mathbf{c} = (c_1, \ldots, c_k) \in \mathbb{Z}^k$, and define*

$$g := \gcd(c_1, \ldots, c_k), \qquad \mathbf{t} := \textstyle\sum_{i=1}^k c_i\mathbf{t}_i.$$

Suppose that $\sigma > \|\mathbf{c}\| \cdot \eta_\epsilon(\Lambda)$ for some negligible ϵ. Then $Z = \sum_{i=1}^k c_i X_i$ is statistically close to $\mathcal{D}_{g\Lambda+\mathbf{t},\|\mathbf{c}\|\sigma}$.

In the full version of this paper [6] we generalize this theorem to $\bar{Z} = A \cdot \bar{X}$, where $\bar{X} = (X_1, \ldots, X_k)$ and A is an $s \times k$ matrix.

5 A Linearly Homomorphic Signature Scheme over $\mathbb{F}_2$

We now describe our linearly homomorphic signature scheme over $\mathbb{F}_2$. Our construction is inspired by the signature scheme of Gentry, Peikert, and Vaikuntanathan [14]. In the GPV scheme, signatures are short vectors in $\Lambda_q^{\mathbf{u}}(\mathbf{A})$, where

$\mathbf{u}$ is the hash of the message to be signed. The key idea in our construction of homomorphic signatures is to work simultaneously modulo 2 and modulo an odd prime q. Specifically, a signature on a vector $\mathbf{v} \in \mathbb{F}_2^n$ is a short vector $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{e}$ is in both $\Lambda_q^\perp(\mathbf{A})$ and $\Lambda_2^{\mathbf{v}}(\mathbf{A})$. The mod 2 part ties the signature to the message, while the mod q part ensures that the signature cannot be forged. By the Chinese remainder theorem, such a vector $\mathbf{e}$ is in the lattice $\Lambda_{2q}^{q\cdot\mathbf{v}}(\mathbf{A})$.

In order to be able to sign multiple files, the matrix $\mathbf{A}$ must be different for every file, yet still have a trapdoor that allows us to generate signatures using the SamplePre algorithm. To achieve this, we divide $\mathbf{A}$ into two parts. The left half is a public matrix generated by the TrapGen algorithm; the right half depends on the identifier of the file being signed. Given the secret basis output by TrapGen, we can use the ExtBasis algorithm to compute a short basis for $\Lambda_{2q}^\perp(\mathbf{A})$.

Our scheme is as follows:

$\mathsf{Setup}(n, \mathsf{params})$. Given a security parameter n and parameters $\mathsf{params} = (N, k, L, m, q, \sigma)$, where $N = n$ is the dimension of vectors to be signed, $k < n$ is the dimension of subspaces to be signed, $L \geq 1$ is the maximum number of linear combinations that can be authenticated, $m(n, L) > n$ is an integer, $q(n, L)$ is an odd prime, and $\sigma(n, L)$ is a real number, do the following:

1. Run $\mathsf{TrapGen}(n, m, 2q)$ to generate a matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n\times m}$ and a basis $\mathbf{T}$ of $\Lambda_{2q}^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{T}}\| \leq 30\sqrt{n \lg 2q}$.
2. Let $H : \{0,1\}^* \to \mathbb{Z}_{2q}^{n\times m}$ be a hash function, viewed as a random oracle.
3. Output the public key $\mathsf{pk} \leftarrow (\mathbf{A}, H)$, and the private key $\mathsf{sk} \leftarrow (\mathbf{A}, H, \mathbf{T})$.

$\mathsf{Sign}(\mathsf{sk}, \mathsf{id}, \mathbf{v})$. Given secret key $\mathsf{sk} = (\mathbf{A}, H, \mathbf{T})$, identifier $\mathsf{id} \in \{0,1\}^n$, and a vector $\mathbf{v} \in \mathbb{F}_2^n$, do the following:

1. Set $\mathbf{B} \leftarrow \mathbf{A}\|H(\mathsf{id}) \in \mathbb{Z}_{2q}^{n\times 2m}$.
2. Let $\mathbf{S} \leftarrow \mathsf{ExtBasis}(\mathbf{T}, \mathbf{B})$ be a basis for $\Lambda_{2q}^\perp(\mathbf{B})$ with $\|\widetilde{\mathbf{S}}\| = \|\widetilde{\mathbf{T}}\|$.
3. Output $\mathbf{e} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{S}, \sigma, q \cdot \mathbf{v})$.

$\mathsf{Combine}(\mathsf{pk}, \mathsf{id}, \{(\alpha_i, \mathbf{e}_i)\}_{i=1}^\ell)$. Given a public key pk, an identifier id, and pairs $\{(\alpha_i, \mathbf{e}_i)\}_{i=1}^\ell$ with $\alpha_i \in \mathbb{F}_2 = \{0,1\}$, output $\mathbf{e} \leftarrow \sum_{i=1}^\ell \alpha_i \mathbf{e}_i \in \mathbb{Z}^{2m}$.

$\mathsf{Verify}(\mathsf{pk}, \mathsf{id}, \mathbf{y}, \mathbf{e})$. Given a public key $\mathsf{pk} = (\mathbf{A}, H)$, an identifier id, a signature $\mathbf{e} \in \mathbb{Z}^{2m}$, and a vector $\mathbf{y} \in \mathbb{F}_2^n$, do the following:

1. Set $\mathbf{B} \leftarrow \mathbf{A}\|H(\mathsf{id}) \in \mathbb{Z}_{2q}^{n\times 2m}$.
2. If (a) $\|\mathbf{e}\| \leq L \cdot \sigma\sqrt{2m}$ and (b) $\mathbf{B} \cdot \mathbf{e} = q \cdot \mathbf{y} \bmod 2q$, output 1. Otherwise output 0.

Proposition 10. *Suppose $\sigma \geq 30\sqrt{n \lg 2q} \cdot \omega(\sqrt{\log n})$. Then the scheme described above is an L-limited linearly homomorphic signature scheme over $\mathbb{F}_2$.*

Unforgeability. We prove unforgeability of our linearly homomorphic signature scheme over $\mathbb{F}_2$ in the random oracle model. Given an adversary that breaks the signature scheme over $\mathbb{Z}_{2q}$, we construct an adversary that simulates the signature scheme and the hash function H and solves the k-SIS problem over $\mathbb{Z}_q$. By Theorem 6, this adversary can in turn be used to solve the SIS problem over $\mathbb{Z}_q$.

Our simulation begins by guessing which of the adversary's signature and hash queries will correspond to the file identifier id^* associated with the adversary's forgery and outputting a public key $\mathbf{A}$ derived from the k-SIS challenge matrix. For queries *not* associated with id^*, the simulator "swaps the roles" of the public key and hash function as follows: we use TrapGen to program the random oracle with a matrix $H(\mathsf{id})$ for which we know a short basis, and we use ExtBasis to compute a short basis for $\mathbf{A}\|H(\mathsf{id})$. We can then compute the signatures as in the real system.

For the query id^*, we construct $H(\mathsf{id}^*)$ so that the k-SIS challenge vectors are valid signatures for the vectors queried by the adversary. We construct the mod q part of $H(\mathsf{id}^*)$ using the fact that valid signatures are elements of $\Lambda_q^\perp(\mathbf{A}\|H(\mathsf{id}^*))$, and we construct the mod 2 part of $H(\mathsf{id}^*)$ using the fact that the k-SIS challenge vectors are statistically close to random mod 2.

With this setup, a forged signature is exactly a solution to the k-SIS problem mod q. We now give the theorem; the proof appears in the full paper [6].

Theorem 11. *Let $\mathcal{N}$ be the linearly homomorphic signature scheme over $\mathbb{F}_2$ described above. Suppose that $m = \lceil 6n \lg 2q \rceil$ and $\sigma = 30\sqrt{n \lg 2q} \log n$. Let $\beta = L \cdot \sigma\sqrt{2m}$. Then $\mathcal{N}$ is unforgeable in the random oracle model assuming that k-$\mathsf{SIS}_{q,2m,\beta,\sigma}$ is infeasible.*

Since the SIS problem is only assumed to be hard for $\beta' = \mathrm{poly}(n)$, by Theorem 6 our choice of σ forces k to be $O(1)$ to ensure security based on SIS.

Privacy. In our linearly homomorphic signature scheme, one derives a signature on a linear combination $\mathbf{v}$ of messages by taking a linear combination of the signatures on the original messages $\mathbf{v}_1, \ldots, \mathbf{v}_k$. Hence, the derived signature on $\mathbf{v}$ is a linear combination of short vectors in cosets of some lattice Λ. To show that this derived signature does not leak information about the original signatures, we use Theorem 9 to show that a linear combination of k signatures generated by our signing algorithm is itself a short vector sampled from a distribution that depends only on the function computed and the message $\mathbf{v}$ output by the function. In particular, the derived signature does not depend on the original vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ (up to negligible statistical distance). It follows that the derived signature does not leak any information about $\mathbf{v}_1, \ldots, \mathbf{v}_k$ beyond

what is revealed by $\mathbf{v}$. We note that the length of σ reveals information about the computed linear function, but not about the original messages.

The above argument implies that a single derived signature is private; for multiple signatures we use a similar argument with an appropriate generalization of Theorem 9. The proof of our privacy theorem can be found in the full paper [6].

Theorem 12. *Let $\mathcal{N}$ be the linearly homomorphic signature scheme over $\mathbb{F}_2$ described in Section 5. Suppose that k is constant, $m = \lceil 6n \lg 2q \rceil$ and $\sigma = 30\sqrt{n \lg 2q} \log n$. Then $\mathcal{N}$ is weakly context hiding.*

6 k-Time GPV Signatures Without Random Oracles

In this section we give a second application of the k-SIS mechanism described in Section 4, namely, a stateless variant of the the signature scheme of Gentry, Peikert, and Vaikuntanathan [14] that is k-time unforgeable in the standard model. The notion of k-time security means that a signing key can only be used to sign k messages. In particular, a forger is allowed at most k signing queries.

The main idea is to construct signatures as in our homomorphic scheme of Section 5, but remove the homomorphic property by setting the bound on the length of a valid signature to be very close to the expected length of the signature. Since we expect a small number of such vectors to form a set that is nearly orthogonal, any linear combination of signatures will produce a vector that is too long to be accepted as a valid signature.

We prove our signature scheme *weakly unforgeable*; i.e., unforgeable under a static chosen-message attack, in which the adversary must submit all signature queries before seeing the public key. A standard transformation using *chameleon hashes* [17] produces a scheme that is unforgeable under the usual notion of adaptive chosen-message attack.

We now describe our weakly unforgeable signature scheme, which is essentially a GPV signature in which hashing is replaced with the Chinese remaindering of the message (viewed as a vector in $\mathbb{F}_2^n$) with the zero vector in $\mathbb{Z}_q^n$.

$\mathsf{Setup}(n, \mathsf{params})$. Given a security parameter n that is also the bit length of messages to be signed, do the following:

1. Choose an odd prime q. Set $m \leftarrow \lceil 6n \lg 2q \rceil$. Set $\sigma \leftarrow 30\sqrt{n \lg 2q} \log n$.
2. Run $\mathsf{TrapGen}(n, m, 2q)$ to generate a matrix $\mathbf{A} \in \mathbb{Z}_{2q}^{n \times m}$ and a basis $\mathbf{T}$ of $\Lambda_{2q}^{\perp}(\mathbf{A})$ such that $\|\widetilde{\mathbf{T}}\| \leq \sigma / \log n$.
3. Output the public key $\mathsf{pk} \leftarrow (\mathbf{A}, \sigma)$ and the private key $\mathsf{sk} \leftarrow (\mathbf{A}, \sigma, \mathbf{T})$.

$\mathsf{Sign}(\mathsf{sk}, \mathbf{v})$. Given secret key $\mathsf{sk} = (\mathbf{A}, \sigma, \mathbf{T})$, and a message $\mathbf{v}$ (interpreted as a vector in $\mathbb{F}_2^n$), output $\mathbf{e} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \sigma, q \cdot \mathbf{v})$.

$\mathsf{Verify}(\mathsf{pk}, \mathbf{e}, \mathbf{v})$. Given a public key $\mathsf{pk} = (\mathbf{A}, \sigma)$, a signature $\mathbf{e} \in \mathbb{Z}^m$, and a message $\mathbf{v} \in \mathbb{F}_2^n$, output 1 if (a) $0 < \|\mathbf{e}\| \leq 1.1 \cdot \sigma\sqrt{m/2\pi}$ and (b) $\mathbf{A} \cdot \mathbf{e} = q \cdot \mathbf{v} \bmod 2q$. Otherwise output 0.

Correctness of our scheme follows from Proposition 7 (with $\alpha = 0.1$), using [14, Lemma 5.3] to bound the smoothing parameter. In our fully unforgeable scheme, the vector $\mathbf{v}$ used in signing is not the message but rather $H(M, r)$, where M is the message, r is random, and H is a chameleon hash function. The signature includes the randomness r in addition to the vector $\mathbf{e}$. For a discussion of lattice-based chameleon hash functions, see [9, §2.2].

An adversary attacking our k-time signature scheme requests k signatures $\mathbf{e}_i$ on messages of his choice, receives a public key and the signatures, and then outputs a message $\mathbf{v}^*$ and a signature $\mathbf{e}^*$. The adversary wins the game if $\mathsf{Verify}(\mathsf{pk}, \mathbf{e}^*, \mathbf{v}^*) = 1$ and $\mathbf{v}^*$ is not equal to any of the messages queried.

As was the case for our homomorphic scheme of Section 5, a valid forgery is a short vector $\mathbf{e}^*$ in $\Lambda_q^\perp(\mathbf{A})$. The key idea in our security proof is that the length bound on a valid $\mathbf{e}^*$ is so tight that by Proposition 8, the only nonzero integer vectors of comparable length in the $\mathbb{Q}$-span of the requested signatures $\mathbf{e}_i$ are the vectors $\pm\mathbf{e}_i$. (Since $-\mathbf{e}_i$ authenticates the same message as $\mathbf{e}_i$, the signature $-\mathbf{e}_i$ is not a valid forgery.) Thus $\mathbf{e}^*$ is outside the linear span of the $\mathbf{e}_i$, and we can use it to solve the k-SIS instance in which the $\mathbf{e}_i$ are the challenge vectors.

Our security theorem is as follows; the proof appears in the full paper [6].

Theorem 13. *Let $\mathcal{S}$ be the signature scheme described above. Suppose k is constant and $\beta = 1.1 \cdot \sigma\sqrt{m/2\pi}$. Then $\mathcal{S}$ is a weakly unforgeable k-time signature scheme assuming that k-$\mathsf{SIS}_{q,m,\beta,\sigma}$ is infeasible.*

7 Further Directions

Extending the linearly homomorphic system. While our linearly homomorphic scheme in Section 5 authenticates vectors with coordinates in $\mathbb{F}_2$, the same construction works for any field $\mathbb{F}_p$ where p is a small prime. We simply set $m = \lceil 6n \lg pq \rceil$ and $\sigma = 30\sqrt{n \lg pq} \log n$, and sign a vector $\mathbf{v} \in \mathbb{F}_p^n$ using the lattice $\Lambda_{pq}^{q \cdot \mathbf{v}}(\mathbf{A})$. If p is odd and we identify $\mathbb{F}_p$ with $\{-(p-1)/2, \ldots, (p-1)/2\}$, then the output of $\mathsf{Combine}$ on ℓ vectors can be up to $\ell(p-1)$ times as long as the largest input vector. An argument as in Proposition 10 shows that the resulting system is $L/(p-1)$-limited.

More interestingly, our system can also be used to authenticate vector spaces defined over non-prime fields. Suppose for concreteness that our vectors live in $(\mathbb{F}_{2^d})^n$. If we fix a basis for $\mathbb{F}_{2^d}$ over $\mathbb{F}_2$, then when computing signatures we may view the vectors as elements of $(\mathbb{F}_2)^{nd}$ and compute signatures in exactly the same manner as above. The difference comes when computing linear

combinations over $\mathbb{F}_{2^d}$: in our representation multiplying an element $x \in \mathbb{F}_{2^d}$ by an element $\alpha \in \mathbb{F}_{2^d}$ consists of multiplying the corresponding vector $\mathbf{x} \in \mathbb{F}_2^d$ by a matrix $M_\alpha \in \mathbb{F}_2^{d \times d}$. To compute this action on the signature vector $\mathbf{e} \in \mathbb{Z}^m$, we lift M_α to an integer matrix with entries in $\{0, 1\}$ and group the elements of $\mathbf{e}$ into d-tuples corresponding to the underlying elements of $\mathbb{F}_{2^d}$. Multiplying each d-tuple by M_α now has the effect of multiplying the underlying elements of $\mathbb{F}_{2^d}$ by α. We see that this action increases the length of $\mathbf{e}$ by a factor of at most d, so combining ℓ vectors gives an output that is up to ℓd times as long as the largest input vector. By the same argument as above, the system over $\mathbb{F}_{2^d}$ is L/d-limited.

Open problem. An important open problem inspired by our construction is to find a tight reduction of k-SIS to worst-case lattice problems, either by improving on the reduction to SIS given by Theorem 6 or by a direct argument. An improved reduction would support the use of the k-SIS problem in developing cryptosystems for other applications and would also allow us to implement our systems with smaller parameters.

An alternative construction. In recent work [5], we have developed a different construction of linearly homomorphic signatures over small fields. Unforgeability of the new scheme (in the random oracle model) reduces directly to the SIS problem, without going through the intermediate k-SIS reduction. As a result, the new scheme can authenticate linear combinations of polynomially many vectors, whereas the one in this paper requires the number of vectors combined to be constant. Signatures in the two schemes are of comparable length.

Acknowledgments. We thank Chris Peikert for helpful discussions, and in particular for providing the idea of using continuous Gaussians to prove Proposition 7. We thank Susan Hohenberger, Hugo Krawczyk, Daniele Micciancio, and Brent Waters for useful conversations. We also thank the anonymous reviewers for helpful suggestions.

References

1. J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters. "Computing on authenticated data." Manuscript (2010).
2. J. Alwen and C. Peikert. "Generating shorter bases for hard random lattices." In *STACS* (2009), 75–86. Full version available at `http://www.cc.gatech.edu/~cpeikert/pubs/shorter.pdf`.
3. G. Ateniese, D. H. Chou, B. de Medeiros, and G. Tsudik. "Sanitizable signatures." In *Computer Security — ESORICS '05*, Springer LNCS **3679** (2005), 159–177.
4. D. Boneh, D. Freeman, J. Katz, and B. Waters. "Signing a linear subspace: Signature schemes for network coding." In *Public-Key Cryptography — PKC '09*, Springer LNCS **5443** (2009), 68–87.

5. D. Boneh and D. M. Freeman. "Homomorphic signatures for polynomial functions." Manuscript (2010).
6. D. Boneh and D. M. Freeman. "Homomorphic signatures over binary fields and new tools for lattice-based signatures." Cryptology eprint report 2010/453 (2010). Available at `http://eprint.iacr.org/2010/453`. Full version of this paper.
7. C. Brzuska, H. Busch, Ö. Dagdelen, M. Fischlin, M. Franz, S. Katzenbeisser, M. Manulis, C. Onete, A. Peter, B. Poettering, and D. Schröder. "Redactable signatures for tree-structured data: Definitions and constructions." In *Applied Cryptography and Network Security — ACNS '10*, Springer LNCS **6123** (2010), 87–104.
8. C. Brzuska, M. Fischlin, T. Freudenreich, A. Lehmann, M. Page, J. Schelbert, D. Schröder, and F. Volk. "Security of sanitizable signatures revisited." In *Public Key Cryptography — PKC '09*, Springer LNCS **5443** (2009), 317–336.
9. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. "Bonsai trees, or, how to delegate a lattice basis." In *Advances in Crypto. — EUROCRYPT '10*, Springer LNCS **6110** (2010), 523–552.
10. E.-C. Chang, C. L. Lim, and J. Xu. "Short redactable signatures using random trees." In *Topics in Cryptology — CT-RSA '09*, Springer LNCS **5473** (2009), 133–147.
11. D. Charles, K. Jain, and K. Lauter. "Signatures for network coding." *International Journal of Information and Coding Theory* **1** (2009), 3–14.
12. C. Fragouli and E. Soljanin. "Network coding fundamentals." *Found. Trends Netw.* **2** (2007), 1–133.
13. R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin. "Secure network coding over the integers." In *Public Key Cryptography — PKC '10*, Springer LNCS **6056** (2010), 142–160.
14. C. Gentry, C. Peikert, and V. Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions." In *STOC*, ed. R. E. Ladner and C. Dwork. ACM (2008), 197–206.
15. S. Haber, Y. Hatano, Y. Honda, W. Horne, K. Miyazaki, T. Sander, S. Tezoku, and D. Yao. "Efficient signature schemes supporting redaction, pseudonymization, and data deidentification." In *ASIACCS '08* (2008), 353–362.
16. R. Johnson, D. Molnar, D. Song, and D. Wagner. "Homomorphic signature schemes." In *Topics in Cryptology — CT-RSA 2002*, Springer LNCS **2271** (2002), 244–262.
17. H. Krawczyk and T. Rabin. "Chameleon signatures." In *Network and Distributed System Security Symposium (NDSS)* (2000).
18. M. Krohn, M. Freedman, and D. Mazières. "On-the-fly verification of rateless erasure codes for efficient content distribution." In *Proc. of IEEE Symposium on Security and Privacy* (2004), 226–240.
19. V. Lyubashevsky and D. Micciancio. "Asymptotically efficient lattice-based digital signatures." In *Theory of Cryptography — TCC '08*, Springer LNCS **4948** (2008), 37–54.
20. D. Micciancio and O. Regev. "Worst-case to average-case reductions based on Gaussian measures." In *45th Annual IEEE Symposium on Foundations of Computer Science — FOCS '04* (2004), 372–381.
21. K. Miyazaki, G. Hanaoka, and H. Imai. "Digitally signed document sanitizing scheme based on bilinear maps." In *ACM Symposium on Information, Computer and Communications Security — ASIACCS '06* (2006), 343–354.
22. K. Miyazaki, M. Iwamura, T. Matsumoto, R. Sasaki, H. Yoshiura, S. Tezuka, and H. Imai. "Digitally signed document sanitizing scheme with disclosure condition control." *IEICE Transactions on Fundamentals* **E88-A** (2005), 239–246.
23. P. Paillier and D. Vergnaud. "Discrete-log-based signatures may not be equivalent to discrete log." In *Advances in Cryptology — ASIACRYPT '05*, Springer LNCS **3788** (2005), 1–20.
24. R. Steinfeld, L. Bull, and Y. Zheng. "Context extraction signatures." In *Information Security and Cryptology (ICISC)*, Springer LNCS **2288** (2001), 285–304.
25. F. Zhao, T. Kalker, M. Médard, and K. Han. "Signatures for content distribution with network coding." In *Proc. Intl. Symp. Info. Theory (ISIT)* (2007).