

Tightly-Secure Signatures From Lossy Identification Schemes

Michel Abdalla¹, Pierre-Alain Fouque¹,
Vadim Lyubashevsky¹, and Mehdi Tibouchi²

¹ École normale supérieure

{michel.abdalla,pierre-alain.fouque,vadim.lyubashevsky}@ens.fr

² NTT Information Sharing Platform Laboratories

tibouchi.mehdi@lab.ntt.co.jp

Abstract. In this paper we present three digital signature schemes with tight security reductions. Our first signature scheme is a particularly efficient version of the short exponent discrete log based scheme of Girault et al. (J. of Cryptology 2006). Our scheme has a tight reduction to the *decisional* Short Discrete Logarithm problem, while still maintaining the non-tight reduction to the *computational* version of the problem upon which the original scheme of Girault et al. is based. The second signature scheme we construct is a modification of the scheme of Lyubashevsky (Asiacrypt 2009) that is based on the worst-case hardness of the shortest vector problem in ideal lattices. And the third scheme is a very simple signature scheme that is based directly on the hardness of the Subset Sum problem. We also present a general transformation that converts, what we term *lossy* identification schemes, into signature schemes with tight security reductions. We believe that this greatly simplifies the task of constructing and proving the security of such signature schemes.

Keywords. Signature schemes, tight reductions, Fiat-Shamir.

1 Introduction

Due to the widespread use of digital signature schemes in practical applications, their construction and security analysis comprises an important area of modern cryptography. While there exist many digital signatures that are secure in the *standard* model (e.g. [16,9,25,6,5]), they are usually less efficient than those that are proved secure in the *random oracle* model, and so are not as suitable for practical applications. Signature schemes secure in the random oracle model generally fall into one of two categories. In the first category are schemes constructed using the *Full Domain Hash* (FDH) approach [4], and in the second are schemes based on the *Fiat-Shamir* technique [12]. Our current work focuses on the latter type.

Proving the security of schemes that are designed using the Fiat-Shamir heuristic (e.g. [24,48,20]) generally involves an invocation of the *forking lemma* [43]. Reductions with this feature entail getting one forgery from the adversary, then rewinding him back to a particular point, and then re-running the adversary

from that point with the hope of getting another forgery. Using these two related forgeries, the reduction can extract an answer to some underlying hard problem such as discrete log or factorization. Due to the fact that two related forgeries are required and one also needs to guess on which of the q_h of his random oracle query the adversary will forge on, a reduction using an adversary that succeeds with probability ε in forging a signature will have probability ε^2/q_h of breaking the hardness assumption. Asymptotically, this does not cause a problem, since the reduction only incurs a polynomial loss in the success probability. The reduction does not, however, provide us with useful guidance for setting concrete parameters because it is unclear whether the efficiency loss is just an artifact of the proof or whether it represents an actual weakness of the scheme. It is therefore preferable to construct protocols that have a tight proof of security by avoiding the use of the forking lemma.

1.1 Related Work and Contributions

Constructing number-theoretic signature schemes with tight security reductions has received some attention in the past. The first work in this direction is due to Bellare and Rogaway [4], who proposed an RSA-based signature scheme known as PSS whose security is tightly related to the security of the RSA function. Later, in the context of signature schemes based on the Fiat-Shamir heuristic, Micali and Reyzin [36] showed that it is sometimes possible to modify the Fiat-Shamir transform in order to achieve tighter reductions. In more recent work, Goh and Jarecki [21] and Katz and Wang [27,22] constructed digital signatures with tight security reductions based on the Computational and Decisional Diffie-Hellman problems. These latter two schemes are versions of the Schnorr signature scheme, and thus inherit most of its characteristics. In particular, the scheme based on the DDH problem has a very simple construction and a rather short signature size. There are other signature schemes, though, that possess other desirable features, but do not yet have a tight security reduction. A notable example of such a scheme is the one of Girault, Poupard, and Stern [20] which is extremely efficient when the signer is allowed to perform pre-processing before receiving the signature. One of the contributions of this paper is a construction of a scheme that possesses all the advantages of the scheme in [20] in addition to having a tight security reduction.

As far as we are aware, there has not been any previous work that specifically considered tight reductions for lattice-based signatures. Similar to number-theoretic constructions, lattice-based signatures secure in the random oracle model are built using either the Full Domain Hash [18,50,39] or the Fiat-Shamir [40,31,28,32,33] methodologies. While FDH-based lattice signatures have tight reductions, the currently most efficient lattice-based schemes (in terms of both the signature size and the running time) are those based on the Fiat-Shamir framework [32,33]. And so it is an interesting problem whether it's possible to construct an efficient Fiat-Shamir based scheme that has tight reductions. The construction of such a scheme is another contribution of this work, though it is unfortunately a little less efficient than the ones in [32,33].

The third scheme that we construct in our work is based on the hardness of the low-density subset sum problem. Due to a known reduction from subset sum to lattice problems [30,13], all signature schemes based on lattices are already based on subset sum. The aforementioned reduction, however, incurs a loss, and so the lattice-based schemes are not based on as hard a version of subset sum as we achieve in this paper by building a scheme directly on subset sum. Additionally, our scheme is surprisingly simple (to describe and to prove) and we believe that it could be of theoretical interest.

Proving schemes secure using the Fiat-Shamir heuristic is usually done by first building a 3-move identification scheme secure against passive adversaries, and then applying the Fiat-Shamir transformation, which was proven in [1] to yield provably secure signatures. The advantage of building schemes using this modular approach is that one does not have to deal with any (usually messy) issues pertaining to random oracles when building the identification scheme – all mention of random oracles is delegated to the black-box transformation. For signature schemes with tight security reductions, however, this construction method does not work. The reason is that the transformation of [1] inherently loses a factor of q_h in the success probability of the impersonator to the ID scheme in relation to the forger of the signature scheme, which results in a non-tight security reduction.

In this paper, we give a black-box transformation analogous to that of [1] that converts what we call, *lossy identification schemes* into signature schemes with tight security reductions. Roughly speaking, a lossy identification scheme is a three move commit-challenge-response identification scheme that satisfies the following four simple properties:

1. Completeness: the verification algorithm must accept a valid interaction with non-negligible probability.
2. Simulatability: there is a simulator, who does not have access to the secret key, who is able to produce valid interaction transcripts that are statistically indistinguishable from real ones.
3. Key indistinguishability: there is an algorithm that produces *lossy* keys that are computationally indistinguishable from the real keys.
4. Lossiness: when the keys are lossy, it is statistically impossible to provide a valid response to a random challenge after making a commitment.

Properties 1 and 2 are generally true of all identification schemes, whereas properties 3 and 4 are particular to the lossy case and are crucially required for obtaining a tight black-box transformation. Our transformation converts a lossy identification scheme into a signature scheme and proves that a successful forger can be converted into a successful impersonator to the identification scheme. Since the only non-statistical property in the definition above is property 3, it means that the successful impersonator breaks this property, which is where we will plant the instance of the hard problem that we are trying to solve. We demonstrate the usefulness and generality of this approach by building our signature schemes in this way.

1.2 Overview of Our Signature Schemes

Construction based on the (decisional) Short Discrete Logarithm Problem. The (computational) c -Discrete Logarithm with Short Exponent (c -DLSE) problem in a cyclic group \mathbb{G} with generator g is the well-studied problem of recovering the discrete logarithm x of a given group element g^x when x is a c -bit long integer, c being typically much smaller than the bit-size of \mathbb{G} . Pollard’s lambda algorithm [44] solves this problem in time $O(2^{c/2})$, but when \mathbb{G} is a subgroup of prime order in \mathbb{Z}_p^* and c is at least twice the security parameter ($c = 160$ for the 80-bit security level, say), the c -DLSE problem is believed to be as hard as the full-length discrete logarithm problem [51,41]. A number of cryptographic schemes are based on the hardness of the c -DLSE problem, including pseudo-random bit generators [41,14,15], key agreement protocols [17] and signature schemes including Girault-Poupard-Stern (GPS) signatures [45,20].

Like other discrete log-based schemes [48,27,7], GPS is an online/offline scheme in the sense of Even, Goldreich and Micali [10,11]: when preprocessing can be done prior to receiving the message to be signed, signature generation becomes very efficient. The main advantage of GPS signatures, however, is that this online signature generation step doesn’t even require a modular reduction, which according to the work of [47], can save as much as 60% of the signing time, which makes the scheme extremely well-suited for situations where processing time is at a premium.

Our scheme, described in Section 4, is very similar to the scheme of [20], but with some tweaks making it possible to choose smaller parameters. Moreover, while the security proof for GPS is a very loose reduction to the computational c -DLSE problem, our security proof provides a *tight* reduction, which is however to the *decisional* short discrete log problem (c -DSDL). Informally, the c -DSDL problem asks to distinguish between a pair (g, g^x) where x is c -bit long and a pair (g, h) where h is uniformly random. No better algorithm is known for solving this problem than actually computing the discrete logarithm and checking whether it is small—in fact, a search-to-decision reduction was established by Koshihara and Kurosawa [29].

Given the pair (g, g^x) , we set it as the public key, which by our assumption is computationally indistinguishable from (g, g^x) where x is random (i.e. not small). We then build an identification scheme that satisfies our simulatability requirement, and furthermore show that it is information-theoretically impossible to respond to a random challenge if x is not small. Using our transformation to signatures, this implies that if a forger can produce a valid forgery, then he can respond to a random challenge, which would mean that x is small.

In the end, we obtain a tightly-secure scheme which is quite efficient in terms of size (signatures are around 320-bits long at the 80-bit security level) and speed, especially when used with coupons (in which case signature generation only requires a single multiplication between integers of 80 and 160 bits respectively).

Construction Based on the Shortest Vector Problem in Ideal Lattices.

In Section 5, we give a construction of a signature scheme based on the hardness

of the approximate worst-case shortest vector problem in ideal lattices. Our scheme is a modification of the scheme in [32] that eliminates the need to use the forking lemma. The scheme in [32] was shown to be secure based on the hardness of the RING-SIS problem, which was previously shown to be as hard as worst-case ideal lattice problems [34,42]. In this work, we construct a similar scheme, but instead have it based on the hardness of the RING-LWE problem, which was recently shown to also be as hard as the worst-case shortest vector problem under quantum reductions [35].

The secret key in our scheme consists of two vectors $\mathbf{s}_1, \mathbf{s}_2$ with small coefficients in the ring $\mathcal{R} = \mathbb{Z}_q[\mathbf{x}]/(\mathbf{x}^n + 1)$, and the public key consists of a random element $\mathbf{a} \in \mathcal{R}$ and $\mathbf{t} = \mathbf{a}\mathbf{s}_1 + \mathbf{s}_2$. The RING-LWE reduction states that distinguishing (\mathbf{a}, \mathbf{t}) from a uniformly random pair in $\mathcal{R} \times \mathcal{R}$ is as hard as solving worst-case lattice problems. In our identification scheme, the commitment is the polynomial $\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$ where $\mathbf{y}_1, \mathbf{y}_2$ are elements in \mathcal{R} chosen with a particular distribution. The challenge is an element $\mathbf{c} \in \mathcal{R}$ with small coefficients, and the response is $(\mathbf{z}_1, \mathbf{z}_2)$ where $\mathbf{z}_1 = \mathbf{y}_1 + \mathbf{s}_1\mathbf{c}$ and $\mathbf{z}_2 = \mathbf{y}_2 + \mathbf{s}_2\mathbf{c}$. As in [32], the procedure sometimes aborts in order to make sure that the distribution of $(\mathbf{z}_1, \mathbf{z}_2)$ is independent of the secret keys. The verification procedure checks that $\mathbf{z}_1, \mathbf{z}_2$ have “small” coefficients, and that $\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{c}\mathbf{t} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$.

The crux of the security proof lies in showing that whenever (\mathbf{a}, \mathbf{t}) is truly random, it is information-theoretically impossible to produce a valid response to a random challenge. Proving this part in our security reduction requires analyzing the ideal structure of the ring \mathcal{R} using techniques similar to the ones in [37]. This analysis is somewhat loose, however, so that the resulting signature scheme is not as efficient as the one in [32]. We believe that improving the analysis (possibly using some recent techniques in [49]) and obtaining a more efficient signature scheme is an interesting research direction.

Construction Based on Subset Sum. In Section 6, we present a very simple scheme based on the hardness of the subset sum problem. The secret key consists of an $n \times k$ 0/1 matrix \mathbf{X} , and the public key consists of a random vector $\mathbf{a} \in \mathbb{Z}_M^n$, as well as a k -dimensional vector of subset sums $\mathbf{t} = \mathbf{a}^T \mathbf{X} \bmod M$ that use \mathbf{a} as weights. The main idea for constructing the lossy identification scheme is to achieve the property that if the vector \mathbf{t} is uniformly random, rather than being a vector of valid subset sums, then it should be impossible (except with a small probability) to produce a valid response to a random challenge. And so an adversary who is able to break the resulting signature scheme can be used to distinguish vectors \mathbf{t} that are valid subset sums of the elements in \mathbf{a} from those that are just uniformly random. We defer further details to Section 6.

2 Preliminaries

2.1 The Decisional Short Discrete Logarithm Problem

Let \mathbb{G} be a finite, cyclic group of prime order q whose group operation is noted multiplicatively, and g a fixed generator of \mathbb{G} . Let further c be a size parameter.

The c -decisional discrete logarithm (c -DSDL) problem may be informally described as the problem of distinguishing between tuples of the form (g, h) for a uniformly random $h \in \mathbb{G}$ and tuples of the form (g, g^x) with x uniformly random in $\{0, \dots, 2^c - 1\}$. More precisely:

Definition 1. A distinguishing algorithm \mathcal{D} is said to (t, ε) -solve the c -DSDL problem in group \mathbb{G} if \mathcal{D} runs in time at most t and satisfies:

$$\left| \Pr[x \xrightarrow{\$} \mathbb{Z}_q : \mathcal{D}(g, g^x) = 1] - \Pr[x \xrightarrow{\$} \{0, \dots, 2^c - 1\} : \mathcal{D}(g, g^x) = 1] \right| \geq \varepsilon$$

We say that \mathbb{G} is a (t, ε) - c -DSDL group if no algorithm (t, ε) -solves the c -DSDL problem in \mathbb{G} .

This problem is related to the well-known (computational) c -discrete logarithm with short exponent (c -DLSE) problem. In fact, for the groups where that problem is usually considered, namely prime order subgroups of \mathbb{Z}_p^* where p is a safe prime, a search-to-decision reduction is known for all c [29]: if the c -DLSE problem is hard, then so is the c -DSDL problem. The reduction is not tight, however, so while the signature scheme presented in the next section admits a tight reduction to the decisional problem, there is a polynomial loss in the reduction to the search problem.

2.2 The Ring-LWE Problem and Lattices

For any positive integer n and any positive real σ , the distribution $D_{\mathbb{Z}^n, \sigma}$ assigns the probability proportional to $e^{-\pi \|\mathbf{y}\|^2 / \sigma^2}$ to every $\mathbf{y} \in \mathbb{Z}^n$ and 0 everywhere else. For any odd prime p , the ring $\mathcal{R} = \mathbb{Z}_p[\mathbf{x}] / (\mathbf{x}^n + 1)$ is represented by polynomials of degree at most $n - 1$ with coefficients in the range $[-\frac{p-1}{2}, \frac{p-1}{2}]$. As an additive group, \mathcal{R} is isomorphic to \mathbb{Z}_p^n , and we use the notation $\mathbf{y} \xrightarrow{\$} D_{\mathcal{R}, \sigma}$ to mean that a vector \mathbf{y} is chosen from the distribution $D_{\mathbb{Z}^n, \sigma}$ and then mapped to a polynomial in \mathcal{R} in the natural way (i.e. position i of the vector corresponds to the coefficient of the \mathbf{x}^i term of the polynomial). The (decisional) Ring Learning With Errors Problem (RING-LWE) over the ring \mathcal{R} with standard deviation σ is to distinguish between the following two oracles: \mathcal{O}_0 outputs random elements in $\mathcal{R} \times \mathcal{R}$, while the oracle \mathcal{O}_1 has a secret $\mathbf{s} \in \mathcal{R}$ where $\mathbf{s} \xrightarrow{\$} D_{\mathcal{R}, \sigma}$, and on every query it chooses a uniformly random element $\mathbf{a} \xrightarrow{\$} \mathcal{R}$, $\mathbf{e} \xrightarrow{\$} D_{\mathcal{R}, \sigma}$, and outputs $(\mathbf{a}, \mathbf{a}\mathbf{s} + \mathbf{e})$. The RING-LWE problem is a natural generalization of the LWE problem [46] to rings and it was recently shown in [35] that if $p = \text{poly}(n)$ is a prime congruent to 1 mod $2n$, then solving the RING-LWE problem over the ring \mathcal{R} with standard deviation³ σ is as hard as finding an approximate shortest vector in all ideal lattices in the ring $\mathbb{Z}[\mathbf{x}] / (\mathbf{x}^n + 1)$. Intuitively, the smaller the ratio between p and σ is, the smaller the vector the reduction is able to find, and thus it is preferable to keep this ratio low.

³ In the actual reduction of [35], the standard deviation is itself chosen from a somewhat complicated probability distribution, but if the number of times the RING-LWE oracle is queried is bounded (in this paper it only needs to provide one output), then the standard deviation can be fixed.

2.3 The Subset Sum Problem

In the search version of the random subset sum problem, $\text{SS}(n, M)$, one is given n elements a_i generated uniformly at random in \mathbb{Z}_M (in this paper, we will only deal with *low-density* instances of the problem, where $M > 2^n$) and an element $t = \sum a_i s_i \bmod M$, where the s_i are randomly chosen from $\{0, 1\}$, and is asked to find the s_i (with high probability, there is only one possible set of s_i). The decision version of the problem, which was shown to be as hard as the search version [26,38], is to distinguish an instance (a_1, \dots, a_n, t) where $t = a_1 x_1 + \dots + a_n x_n \bmod M$ from the instance (a_1, \dots, a_n, t) where t is uniformly random in \mathbb{Z}_M . The low-density $\text{SS}(n, M)$ problem is hardest when $M \approx 2^n$, in which case the best algorithm runs in time $2^{\Omega(n)}$ (see for example [3]), but the best known algorithms for the problem when $M = n^{O(n)}$, still require time $2^{\Omega(n)}$. As M increases, however, the problem becomes easier, until it is solvable in polynomial-time when $M = 2^{\Omega(n^2)}$ [30,13].

2.4 Signature Schemes

Definition 2. A signature scheme Sig is composed of three algorithms (GenKey , Sign , Verify) such that:

- The key generation algorithm GenKey takes as input the security parameter in unary notation and outputs a pair (pk, sk) containing the public verification key and the secret signing key.
- The signing algorithm Sign takes as input a message m and the signing key sk and outputs a signature σ . This algorithm can be probabilistic so that many signatures can be computed for the same message.
- The verification algorithm Verify takes as input a message m , a signature σ and the public key pk and outputs 1 if the signature is correct and 0 otherwise.

The standard security notion for signature scheme is *strong existential unforgeability against adaptive chosen-message attacks* [23] which informally means that, after obtaining signatures on polynomially many arbitrary messages of his choice, an adversary cannot produce a new valid signature, even for a message m for which he already knows a correct signature.

Definition 3. Let $\text{Sig} = (\text{GenKey}, \text{Sign}, \text{Verify})$ be a signature scheme and let H be a random oracle. We say that Sig is $(t, q_h, q_s, \varepsilon)$ -strongly existentially unforgeable against adaptive chosen-message attacks, if there is no algorithm \mathcal{D} that runs in time at most t , while making at most q_h hash queries and at most q_s signing queries, such that

$$\Pr[(pk, sk) \leftarrow \text{GenKey}(1^k); (m, \sigma) \leftarrow \mathcal{D}^{\text{Sign}(sk, \cdot), H(\cdot)}(pk) : (m, \sigma) \notin \mathcal{S} \wedge \text{Verify}(m, \sigma, pk) = 1] \geq \varepsilon,$$

where \mathcal{S} is the set of message-signature pairs returned by the signing oracle.

3 Lossy Identification Schemes

In order to unify the security proofs of our signature schemes without sacrificing the tightness of the reduction, we introduce in this section a new class of identification schemes, called lossy identification schemes. In these schemes, the public key associated with the prover can take one of two indistinguishable forms, called *normal* and *lossy*. When the public key is normal, the scheme behaves as a standard identification scheme with similar security guarantees against impersonation attacks. However, in the lossy case, the public key may not have a corresponding secret key and no prover (even computationally unbounded ones) should be able to make the verifier accept with non-negligible probability.

As with other identification schemes used to build signature schemes via the Fiat-Shamir transform, the identification schemes that we consider in this paper consist of a canonical three-move protocol, as defined in [1]. In these protocols, the verifier’s move consists in choosing a random string from the challenge space and sending it to the prover. Moreover, its final decision is a *deterministic* function of the conversation transcript and the public key. Since our results can be seen as a generalization of the results of Abdalla *et al.* [1] to the lossy setting, we use their definitions as the basis for ours below.

Definition 4. A lossy identification scheme ID is defined by a tuple $(\text{KeyGen}, \text{LosKeyGen}, \text{Prove}, c, \text{Verify})$ such that:

- KeyGen is the normal key generation algorithm which takes as input the security parameter in unary notation and outputs a pair (pk, sk) containing the publicly available verification key and the prover’s secret key.
- LosKeyGen is the lossy key generation algorithm which takes as input the security parameter in unary notation and outputs a lossy verification key pk .
- Prove is the prover algorithm which takes as input the current conversation transcript and outputs the next message to be sent to the verifier.
- $c(k)$ is a function of the security parameter which determines the length of the challenge sent by the verifier.
- Verify is a deterministic algorithm which takes the conversation transcript as input and outputs 1 to indicate acceptance or 0 otherwise.

Following [1], we associate to ID , k , and (pk, sk) a randomized *transcript generation oracle* $\text{Tr}_{pk, sk, k}^{\text{ID}}$ which takes no inputs and returns a random transcript of an “honest” execution. However, to adapt it to specific setting of our schemes, we modify to the original definition to take into account the possibility that the prover may fail and output \perp as response during the execution of the identification protocol. Moreover, when this happens, instead of outputting (cmt, ch, \perp) , our transcript generation oracle will simply return a triplet (\perp, \perp, \perp) to simulate the scenario in which the verifier simply *forgets* failed identification attempts. Interestingly, as we show later in this section, this weaker requirement is sufficient for building secure signature schemes as failed impersonation attempts will

be kept hidden from the adversary since the tasks of generating the commitment and challenge are performed by the signer. More precisely, the transcript generation oracle $\text{Tr}_{pk,sk,k}^{\text{ID}}$ is defined as follows:

$\text{Tr}_{pk,sk,k}^{\text{ID}}()$:

- 1: $cmt \xleftarrow{\$} \text{Prove}(sk)$
- 2: $ch \xleftarrow{\$} \{0, 1\}^{c(k)}$
- 3: $rsp \xleftarrow{\$} \text{Prove}(sk, cmt, ch)$
- 4: **if** $rsp = \perp$ **then** $(cmt, ch) \leftarrow (\perp, \perp)$
- 5: **return** (cmt, ch, rsp)

Definition 5. An identification scheme is said to be lossy if it has the following properties:

1. **Completeness of normal keys.** We say that ID is ρ -complete, where ρ is a non-negligible function of k , if for every security parameter k and all honestly generated keys $(pk, sk) \xleftarrow{\$} \text{KeyGen}(1^k)$, $\text{Verify}(pk, cmt, ch, rsp) = 1$ holds with probability ρ when $(cmt, ch, rsp) \xleftarrow{\$} \text{Tr}_{pk,sk,k}^{\text{ID}}()$.
2. **Simulatability of transcripts.** Let (pk, sk) be the output of $\text{KeyGen}(1^k)$ for a security parameter k . Then, we say that ID is ε -simulatable if there exists a PPT algorithm $\tilde{\text{Tr}}_{pk,k}^{\text{ID}}$ with no access to the secret key sk which can generate transcripts $\{(cmt, ch, rsp)\}$ whose distribution is statistically indistinguishable from the transcripts output by $\text{Tr}_{pk,sk,k}^{\text{ID}}$, where ε is an upper-bound for the statistical distance. When $\varepsilon = 0$, then ID is said to simulatable.
3. **Indistinguishability of keys.** Consider the experiments $\text{Exp}_{\text{ID}, \mathcal{D}}^{\text{ind-keys-real}}(k)$ and $\text{Exp}_{\text{ID}, \mathcal{D}}^{\text{ind-keys-lossy}}(k)$ in which we generate pk via $\text{KeyGen}(1^k)$, respectively $\text{LosKeyGen}(1^k)$, and provide it as input to the distinguishing algorithm \mathcal{D} . We say that \mathcal{D} can (t, ε) -solve the key-indistinguishability problem if \mathcal{D} runs in time t and

$$|\Pr[\text{Exp}_{\text{ID}, \mathcal{D}}^{\text{ind-keys-real}}(k) = 1] - \Pr[\text{Exp}_{\text{ID}, \mathcal{D}}^{\text{ind-keys-lossy}}(k) = 1]| \geq \varepsilon.$$

Furthermore, we say that ID is (t, ε) -key-indistinguishable if no algorithm (t, ε) -solves the key-indistinguishability problem.

4. **Lossiness.** Let \mathcal{I} be an impersonator, st be its state, and k be a security parameter. Let $\text{Exp}_{\text{ID}, \mathcal{I}}^{\text{los-imp-pa}}(k)$ be the following experiment played between \mathcal{I} and a hypothetical challenger:

$\text{Exp}_{\text{ID}, \mathcal{I}}^{\text{los-imp-pa}}(k)$:

- 1: $pk \xleftarrow{\$} \text{LosKeyGen}(1^k)$
- 2: $(st, cmt) \xleftarrow{\$} \mathcal{I}^{\tilde{\text{Tr}}_{pk,k}^{\text{ID}}}(pk)$; $ch \xleftarrow{\$} \{0, 1\}^{c(k)}$; $rsp \xleftarrow{\$} \mathcal{I}(st, ch)$
- 3: **return** $\text{Verify}(pk, cmt, ch, rsp)$

Sign (sk, m): 1: $ctr \leftarrow 0$ 2: while $ctr \leq \ell$ and $rsp = \perp$ do 3: $ctr \leftarrow ctr + 1$ 4: $cmt \leftarrow \text{Prove}(sk)$ 5: $ch \leftarrow H(cmt, m)$ 6: $rsp \leftarrow \text{Prove}(sk, cmt, ch)$ 7: end while 8: if $rsp = \perp$ then $cmt \leftarrow \perp$ 9: $\sigma \leftarrow (cmt, rsp)$ 10: return σ	Verify (pk, m, σ): 1: parse σ as (cmt, rsp) 2: $ch \leftarrow H(cmt, m)$ 3: $d \leftarrow \text{Verify}(pk, cmt, ch, rsp)$ 4: return d
---	--

Fig. 1. Description of our signature scheme $\text{Sig}[\text{ID}, \ell] = (\text{GenKey}, \text{Sign}, \text{Verify})$, where $\text{ID} = (\text{KeyGen}, \text{LosKeyGen}, \text{Prove}, c, \text{Verify})$ is a lossy identification scheme, H is a random oracle, and ℓ is a bound on the number of signing attempts.

We say \mathcal{I} ε -solves the impersonation problem with respect to lossy keys if

$$\Pr[\mathbf{Exp}_{\text{ID}, \mathcal{I}}^{\text{los-imp-pa}}(k) = 1] \geq \varepsilon.$$

Furthermore, we say that ID is ε -lossy if no (computationally unrestricted) algorithm ε -solves the impersonation problem with respect to lossy keys.

As in [1], we need to use the concept of min-entropy [8] to measure the maximum likelihood that a commitment generated by the prover collides with a fixed value. The precise definition of min-entropy can be found in Definition 3.2 in [1].

Transform. The signature schemes that we consider in this paper are built from lossy identification schemes via the Fiat-Shamir transform [12], in which the challenge becomes the hash of the message together with the commitment. However, since we do not assume perfect completeness of normal keys for the underlying lossy identification scheme, the signing algorithm will differ slightly from those considered in [1] in order to decrease the probability of abort during signing. More precisely, let $\text{ID} = (\text{KeyGen}, \text{LosKeyGen}, \text{Prove}, c, \text{Verify})$ be a lossy identification scheme and let H be a random oracle. Let ℓ be a parameter defining the maximum number of signing attempts. We can construct a signature scheme $\text{Sig}[\text{ID}, \ell] = (\text{GenKey}, \text{Sign}, \text{Verify})$, where GenKey simply calls KeyGen from the ID scheme, and $\text{Sign}, \text{Verify}$ are depicted in Figure 1.

We remark that the signature length of the scheme in Figure 1 can sometimes be optimized by setting $\sigma = (ch, rsp)$. However, this is only possible when the commitment value cmt is uniquely defined by (ch, rsp) , which is the case for all the schemes considered in this paper.

Theorem 1. Let $\text{ID} = (\text{KeyGen}, \text{LosKeyGen}, \text{Prove}, c, \text{Verify})$ be a lossy identification scheme whose commitment space has min-entropy $\beta(k)$, let H be a random

oracle, and let $\text{Sig}[\text{ID}] = (\text{GenKey}, \text{Sign}, \text{Verify})$ be the signature scheme obtained via the transform in Figure 1. If ID is ε_s -simulatable, ρ -complete, (t', ε_k) -key-indistinguishable, and ε_ℓ -lossy, then $\text{Sig}[\text{ID}]$ is $(t, q_h, q_s, \varepsilon)$ -strongly existentially unforgeable against adaptive chosen-message attacks in the random oracle model for:

$$\begin{aligned}\varepsilon &= \varepsilon_k + q_s \varepsilon_s + (q_h + 1) \varepsilon_\ell + \ell(q_s + q_h + 1) q_s / 2^\beta \\ t &\approx t' - O(q_s \cdot t_{\text{Sign}})\end{aligned}$$

where t_{Sign} denotes the average signing time. Furthermore, the probability that $\text{Sig}[\text{ID}]$ outputs a valid signature is $1 - (1 - \rho)^\ell$.

Proof overview. In order to prove the security of the signature scheme based on the security properties of the underlying lossy identification scheme, the main idea is to use honest transcripts generated by the identification scheme to answer signature queries made the adversary by appropriately programming the random oracle. More precisely, let (cmt, ch, rsp) be a valid transcript (i.e., $\text{Verify}(pk, cmt, ch, rsp) = 1$). To answer a query m to the signing oracle, we need to program the random oracle to set $H(cmt, m) = ch$ so that (cmt, rsp) is a valid signature for m . Unfortunately, this programming may conflict with previous values outputted by the hash oracle. To address this problem, the first step of the proof is to show that such collisions happen with probability at most $\ell(q_s + q_h + 1)q_s/2^\beta$.

Next, we make a sequence of small changes to the security experiment to be able to bound the success probability of the forger. The first significant modification is to change the simulation of the signing oracle so that it no longer uses the secret key. This is done by replacing the transcript generation oracle $\text{Tr}_{pk, sk, k}^{\text{ID}}$ with its simulated version $\tilde{\text{Tr}}_{pk, k}^{\text{ID}}$. Since we make at most q_s calls to $\tilde{\text{Tr}}_{pk, k}^{\text{ID}}$, the difference in the success probability of the forger changes by at most $q_s \varepsilon_s$ due to the simulatability of ID .

The second important modification is to replace the key generation algorithm with its lossy version. Since the secret key is no longer needed in the simulation of the signing oracle, the difference in the success probability of the forger changes by at most ε_k due to the key-indistinguishability of ID .

Finally, we can bound the success probability of the forger in this final experiment by relating this probability with that of solving the impersonation problem with respect to lossy keys. Since we need to guess the hash query which will be used in the forgery to be able to break the underlying impersonation problem, we lose a factor $q_h + 1$ in the reduction, resulting in the term $(q_h + 1)\varepsilon_\ell$ in the theorem. For more details, please refer to the full version of this paper [2].

4 A Signature Scheme Based on the DSDL Problem

In this section we describe our short discrete log based signature scheme. While it looks similar to the prime-order version of the Girault-Poupard-Stern identification scheme [19,45,20], the proof strategy is in fact closer to the one used by

Katz and Wang for their DDH-based signature scheme [27,22]. We first present a lossy identification scheme and then use the generic transformation from the previous section to obtain the signature scheme.

The public parameters of the identification scheme are a cyclic group \mathbb{G} of prime order q (typically chosen as the subgroup of order q in \mathbb{Z}_p^* where p is prime), a generator g of \mathbb{G} , and size parameters c, k, k' . The secret key is a small (relative to q) integer x and the public key consists of a single group element $h = g^x \bmod p$. The prover's first move is to generate a small (but larger than x) random integer y and send $u = g^y$ as a commitment to the verifier. Next, the (honest) verifier picks a value e uniformly in $\{0, \dots, 2^k - 1\}$ and sends it to the prover. After receiving e from the verifier, the prover computes $z = ex + y$ (without any modular reduction), and checks whether z is in the range $\{2^{k+c}, \dots, 2^{k+k'+c} - 1\}$. If z is in the “correct” range, then the prover sends z to the verifier, who can check the verifying equation $u = g^z/h^e$ to authenticate the prover. If z is outside the correct range, the prover sends \perp to indicate failure—as in [31,32], this check is important to ensure that the distribution of the value z is independent of the secret key x . In the full version of this paper [2], we prove:

Theorem 2. *If \mathbb{G} is a (t, ε) -c-DSDL group, then the identification scheme described above is perfectly simulatable, ρ -complete, (t, ε) -key-indistinguishable, and ε_ℓ -lossy, for $\rho = 1 - 2^{-k'}$ and $\varepsilon_\ell \leq 2^{2k+k'+c+2}/q + 1/2^k$.*

In order to obtain our signature scheme based on the DSDL problem, we apply the transform provided in the previous section to the identification scheme described above. The full description of the resulting scheme is provided in Figure 2. In addition to those of the underlying identification scheme, the public parameters of the signature scheme also include the maximum number of signing attempts ℓ and a random oracle $H: \{0, 1\}^* \rightarrow \{0, \dots, 2^k - 1\}$. The key pair is as before. To sign a message m , we generate a small (but larger than x) random integer y and compute $e \leftarrow H(g^y \bmod p, m)$. Finally, we set $z = ex + y$ and check whether z is in the correct range. If it's not, we restart the signature process. In case of ℓ failures, the signing algorithm simply outputs (\perp, \perp) to indicate failure. Otherwise, the signature will consist of the pair $\sigma = (z, e)$. Since the probability that z is not in the correct range is smaller than $1/2^{k'}$, the signing algorithm will fail with probability at most $(1 - 1/2^{k'})^\ell$. Moreover, the average number of iterations is $1/(1 - 1/2^{k'})$. As a direct consequence of Theorems 1 and 2, we get:

Theorem 3. *If \mathbb{G} is a (t', ε') -c-DSDL group, then this signature scheme is $(t, q_h, q_s, \varepsilon)$ -strongly existentially unforgeable against adaptive chosen-message attacks in the random oracle model for:*

$$\varepsilon = \varepsilon' + (q_h + 1) \cdot \frac{2^{2k+k'+c+2}}{q} + \ell(q_s + q_h + 1) \cdot \frac{q_s}{2^k}$$

$$t \approx t' - O(q_s \cdot t_1)$$

(where t_1 is the cost of an exponentiation in \mathbb{G}), and it outputs a valid signature with probability $1 - 2^{-k'\ell}$.

KeyGen(): Pick $x \xleftarrow{\$} \{0, \dots, 2^c - 1\}$ as the private key, and $X \leftarrow g^x \bmod p$ as the public key.

Sign(m, x):

- 1: $ctr \leftarrow 0$
- 2: $y \xleftarrow{\$} \{0, \dots, 2^{k+k'+c} - 1\}$
- 3: $e \leftarrow H(g^y \bmod p, m)$
- 4: $z \leftarrow ex + y$
- 5: **if** $z \notin \{2^{k+c}, \dots, 2^{k+k'+c} - 1\}$ and $ctr < \ell$ **then**
- 6: $ctr \leftarrow ctr + 1$
- 7: **goto** Step 2
- 8: **if** $z \notin \{2^{k+c}, \dots, 2^{k+k'+c} - 1\}$ **then** $(z, e) \leftarrow (\perp, \perp)$
- 9: **return** $\sigma = (z, e)$

Verify($m, X, \sigma = (z, e)$): accept if and only if $z \in \{2^{k+c}, \dots, 2^{k+k'+c} - 1\}$ and $e = H(g^z \cdot X^{-e} \bmod p)$.

Fig. 2. DSDL-Based Signature Scheme.

Remarks

1. The scheme in Figure 2 uses (z, e) instead of (z, g^y) as the signature since (z, e) can be used to recover g^y , but the length of e is shorter than that of g^y .
2. This is an online/offline signature scheme: it can be used with coupons by pre-computing $(y, g^y \bmod p)$ independently of the message. In the rare case when z is not in the right interval (which can be checked without even computing a multiplication), it suffices to use another coupon.
3. The reduction is not *completely* tight: there is a small loss of $\ell \cdot q_s$. As in [22], this loss can be avoided by ensuring that the masking parameter y is always the same for a given message, either by making the scheme stateful (keeping track of the randomness on signed messages) or by generating y as a deterministic, pseudorandom function of the signed message and the private key (but the resulting scheme is no longer online/offline).

Suggested Parameters. We propose the following parameters for an instantiation of our scheme with an 80-bit security level. The group \mathbb{G} is a subgroup of order q in \mathbb{Z}_p^* , where p is a 1024-bit prime and q a prime factor of $p - 1$ of length ≥ 490 bits. Moreover, we set $(c, k, k') = (160, 80, 8)$. The size of the public key $g^x \bmod p$ is then 1024 bits and the size of the signature (z, e) is $k + k' + c + k = 328$ bits.

A full signature requires a single exponentiation of 248 bits in \mathbb{Z}_p^* with fixed base, which is about as efficient as comparable schemes (faster than the two 160-bit exponentiations in the Katz-Wang DDH scheme, for example). In our scheme, there is a $1/2^{k'} = 1/256$ chance that the signing algorithm will have to be repeated, but this has little effect on the expected running time.

Parameter	Definition
n	integer that is a power of 2
σ	standard deviation of the secret key coefficients
p	“small” prime equal to 1 mod $2n$
\mathcal{R}	ring $\mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1 \rangle$
\mathcal{C}	$\{\mathbf{g} \in \mathcal{R} : \ \mathbf{g}\ _\infty \leq \log n\}$
\mathcal{M}	$\{\mathbf{g} \in \mathcal{R} : \ \mathbf{g}\ _\infty \leq n^{3/2} \sigma \log^3 n\}$
\mathcal{G}	$\{\mathbf{g} \in \mathcal{R} : \ \mathbf{g}\ _\infty \leq (n-1)\sqrt{n}\sigma \log^3 n\}$

Fig. 3. Parameter Definitions

When used with coupons, the scheme is possibly the fastest option available, with an online cost of one single integer multiplication between a 80-bit number and a 160-bit number, and no modular reduction.

5 A Signature Scheme Based on Lattices

In this section, we present a signature scheme whose security is based on the hardness of the RING-LWE problem. Towards this goal, we first describe a lossy identification scheme based on the RING-LWE problem and then use our generic transformation in Section 3 to obtain the signature scheme.

Our identification scheme depends on some public parameters defined in Figure 3. The secret key consists of two polynomials $\mathbf{s}_1, \mathbf{s}_2$ with “small” coefficients chosen from the distribution $D_{\mathcal{R}, \sigma}$, and the public key consists of a randomly-chosen element $\mathbf{a} \in \mathcal{R}$ and of the value $\mathbf{t} = \mathbf{a}\mathbf{s}_1 + \mathbf{s}_2$. Under the RING-LWE assumption in the ring \mathcal{R} , the public key is thus indistinguishable from a uniformly random element of \mathcal{R}^2 .

In our protocol, the prover’s first move is to create two “small” polynomials $\mathbf{y}_1, \mathbf{y}_2$ (larger than $\mathbf{s}_1, \mathbf{s}_2$ by a factor $\approx n$) from the set \mathcal{M} , and then send the value $\mathbf{u} = \mathbf{a}\mathbf{y}_1 + \mathbf{y}_2$ to the verifier. Upon receipt of \mathbf{u} , the (honest) verifier chooses a value \mathbf{c} uniformly at random in the set \mathcal{C} and sends it to the prover. After receiving \mathbf{c} from the verifier, the prover sets $\mathbf{z}_1 \leftarrow \mathbf{s}_1\mathbf{c} + \mathbf{y}_1$ and $\mathbf{z}_2 \leftarrow \mathbf{s}_2\mathbf{c} + \mathbf{y}_2$ and checks whether the \mathbf{z}_i ’s are both in \mathcal{G} . If they are, the prover then sends the response $(\mathbf{z}_1, \mathbf{z}_2)$ to the verifier. If one (or both) of the \mathbf{z}_i are outside of \mathcal{G} (which happens with probability approximately $1 - 1/e^2$), then the prover simply sends (\perp, \perp) . Finally, the verifier simply checks whether the \mathbf{z}_i ’s are in \mathcal{G} and that $\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 = \mathbf{t}\mathbf{c} + \mathbf{u}$.

At this point, we would like to point out that using the recent techniques in [33], it is possible to lower the bitsize of the response $(\mathbf{z}_1, \mathbf{z}_2)$ by choosing the polynomials $\mathbf{y}_1, \mathbf{y}_2$ from a normal distribution and then doing a somewhat more involved rejection sampling when deciding whether to send $(\mathbf{z}_1, \mathbf{z}_2)$ or (\perp, \perp) to the verifier.

In the full version of this paper [2], we prove:

Theorem 4. *If $p \gg \sigma^{2/\alpha} \cdot n^{3/\alpha+\eta}$ for some $\eta > 0$, and the RING-LWE problem over \mathcal{R} with standard deviation σ is (ε, t) -hard, then the identification scheme*

KeyGen(): Pick $\mathbf{s}_1, \mathbf{s}_2 \xleftarrow{\$} D_{\mathcal{R}, \sigma}$ and set $(\mathbf{s}_1, \mathbf{s}_2)$ as the private key. Select $\mathbf{a} \xleftarrow{\$} \mathcal{R}$ and let the public key be (\mathbf{a}, \mathbf{t}) , where $\mathbf{t} \leftarrow \mathbf{a}\mathbf{s}_1 + \mathbf{s}_2$. Let H be a random oracle mapping to the range \mathcal{C} .

Sign($m, \mathbf{a}, \mathbf{s}_1, \mathbf{s}_2$):

- 1: $ctr \leftarrow 0$
- 2: $\mathbf{y}_1, \mathbf{y}_2 \xleftarrow{\$} \mathcal{M}$
- 3: $\mathbf{c} \leftarrow H(\mathbf{a}\mathbf{y}_1 + \mathbf{y}_2, m)$
- 4: $\mathbf{z}_1 \leftarrow \mathbf{s}_1\mathbf{c} + \mathbf{y}_1, \mathbf{z}_2 \leftarrow \mathbf{s}_2\mathbf{c} + \mathbf{y}_2$
- 5: **if** \mathbf{z}_1 or $\mathbf{z}_2 \notin \mathcal{G}$ and $ctr < \ell$ **then**
- 6: $ctr \leftarrow ctr + 1$
- 7: **goto** Step 2
- 8: **if** \mathbf{z}_1 or $\mathbf{z}_2 \notin \mathcal{G}$ **then** $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{c}) \leftarrow (\perp, \perp, \perp)$
- 9: **return** $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{c})$

Verify($m, \mathbf{z}_1, \mathbf{z}_2, \mathbf{c}, \mathbf{a}, \mathbf{t}$): accept if and only if $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{G}$ and $\mathbf{c} = H(\mathbf{a}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}\mathbf{c}, m)$.

Fig. 4. Lattice-Based Signature Scheme

described above is ε_s -simulatable, ρ -complete, (t, ε) -key-indistinguishable and ε_ℓ -lossy, for $\rho \geq 1/e^2 - 2/(en)$ and $\varepsilon_s, \varepsilon_\ell \leq \text{negl}(n)$.

In order to obtain our signature scheme based on lattices, we apply our generic transform to the identification scheme described above. The full description of the resulting scheme is provided in Figure 4.

6 A Signature Scheme Based on Subset Sum

In this section, we construct a lossy identification scheme based on the hardness of the random SS(n, M) problem for $M > (2kn + 1)^n \cdot 3^{2k}$, where k is a security parameter. The secret key is a random matrix $\mathbf{X} \xleftarrow{\$} \{0, 1\}^{n \times k}$, and the public key consists of a vector $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_M^n$, and a vector $\mathbf{t} = \mathbf{a}^T \mathbf{X} \bmod M$. In the first step of the protocol, the prover selects a vector $\mathbf{y} \xleftarrow{\$} \{-kn, \dots, kn\}^n$ and sends an integer commitment $u = \langle \mathbf{a}, \mathbf{y} \rangle \bmod M$ to the verifier. The verifier selects a random challenge vector $\mathbf{c} \xleftarrow{\$} \{0, 1\}^k$, and sends it to the prover, who checks that \mathbf{c} is indeed a valid challenge vector. The prover then computes a possible response $\mathbf{z} = \mathbf{X}\mathbf{c} + \mathbf{y}$ (note that there is no modular reduction here), and sends it to the verifier if it is in the range $\{-kn + k, \dots, kn - k\}^n$. If \mathbf{z} is not in this range, then the prover sends \perp . Upon receiving a \mathbf{z} , the verifier accepts the interaction if $\mathbf{z} \in \{-kn + k, \dots, kn - k\}^n$ and $\langle \mathbf{a}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{c} \rangle \bmod M = u$.

It is easy to see that in the case that the prover does not send \perp , he will be accepted by the verifier since

$$\langle \mathbf{a}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{c} \rangle \bmod M = \mathbf{a}^T \mathbf{X}\mathbf{c} + \langle \mathbf{a}, \mathbf{y} \rangle - \mathbf{a}^T \mathbf{X}\mathbf{c} \bmod M = u.$$

Then, we observe that the probability that for any element $\bar{\mathbf{z}} \in \{-kn + k, \dots, kn - k\}^n$, the probability that the response will be $\mathbf{z} = \bar{\mathbf{z}}$ is

$$Pr[\mathbf{z} = \bar{\mathbf{z}}] = Pr[\mathbf{y} = \bar{\mathbf{z}} - \mathbf{X}\mathbf{c}] = 1/|\{-kn, \dots, kn\}^n|,$$

since all the coefficients of the vector $\mathbf{X}\mathbf{c}$ have absolute value at most k . Therefore every element \mathbf{z} in the set $\{-kn + k, \dots, kn - k\}^n$ has an equal probability of being outputted and the probability that $\mathbf{z} \neq \perp$ is

$$\rho = |\{-kn + k, \dots, kn - k\}^n| / |\{-kn, \dots, kn\}^n| \approx (1 - 1/n)^n \approx 1/e.$$

And thus the simulatability property of the scheme is satisfied since one can create a valid transcript by generating (\perp, \perp, \perp) with probability $1 - \rho$, and otherwise pick a random $\mathbf{z} \in \{-kn + k, \dots, kn - k\}^n$, a random $\mathbf{c} \in \{0, 1\}^k$, and output $(\langle \mathbf{a}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{c} \rangle \bmod M, \mathbf{c}, \mathbf{z})$.

The lossy public keys are just two uniformly random vectors \mathbf{a} and \mathbf{t} , and so the indistinguishability of these keys from the real keys is directly based on the hardness of the $SS(n, M)$ problem using a standard hybrid argument.

To show lossiness, we observe that if \mathbf{t} is uniformly random in \mathbb{Z}_M^k , then it can be shown that with high probability, for any choice of $u \in \mathbb{Z}_M$, there is at most one value \mathbf{c} such that u can be written as $\langle \mathbf{a}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{c} \rangle \bmod M$. Indeed, if there exist two pairs $(\mathbf{z}, \mathbf{c}), (\mathbf{z}', \mathbf{c}')$, such that

$$\langle \mathbf{a}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{c} \rangle = \langle \mathbf{a}, \mathbf{z}' \rangle - \langle \mathbf{t}, \mathbf{c}' \rangle \bmod M,$$

then we have

$$\langle \mathbf{a}, \mathbf{z} - \mathbf{z}' \rangle - \langle \mathbf{t}, \mathbf{c} - \mathbf{c}' \rangle \bmod M = 0. \quad (1)$$

The set of valid pairs $(\mathbf{z} - \mathbf{z}', \mathbf{c} - \mathbf{c}')$ consists of $(2kn + 1)^n \cdot 3^k$ elements. If (\mathbf{a}, \mathbf{t}) is chosen completely at random, then for each of those valid pairs, the probability that Equation (1) is satisfied is $1/M$ (this assumes that either \mathbf{a} or \mathbf{t} has at least one element that is invertible modulo M , which is the case with extremely high probability), and so the probability over the randomness of \mathbf{a} and \mathbf{t} that Equation (1) is satisfied for any of the valid pairs is at most $(2kn + 1)^n \cdot 3^k / M$, which by our choice of M , is at most 3^{-k} .

To convert this lossy identification scheme to a signature scheme, one would simply perform the transformation described in Figure 1, as we did for the other schemes in this paper. And as for the lattice-based scheme in Section 5, we point out that the technique in [33] can be used to reduce the coefficients of the signature by about a factor of \sqrt{n} to make them fall in the range $\{-O(k\sqrt{n}), \dots, O(k\sqrt{n})\}^n$ by sampling the vector \mathbf{y} from a normal distribution and performing a somewhat more involved rejection sampling procedure when deciding whether or not to send the response \mathbf{z} . This would also allow us to reduce the modulus M to approximately $M = O(k\sqrt{n})^n \cdot 3^{2k}$, which makes the $SS(n, M)$ problem more difficult. Another possible optimization could include making k larger, but making the vector \mathbf{c} sparser (while still making sure that it comes from a large enough set), which would result in a shorter vector $\mathbf{X}\mathbf{c}$.

Acknowledgments. This work was supported in part by the European Research Council and by the European Commission through the ICT Program under Contract ICT-2007-216676 ECRYPT II.

We thank Mihir Bellare and Eike Kiltz for helpful comments on a preliminary version of this paper.

References

1. M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In L. R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Apr. / May 2002.
2. M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes, 2012. Full version of this paper.
3. A. Becker, J.-S. Coron, and A. Joux. Improved generic algorithms for hard knapsacks. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 364–385. Springer, May 2011.
4. M. Bellare and P. Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 399–416. Springer, May 1996.
5. X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In P. Q. Nguyen and D. Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer, May 2010.
6. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, May 2010.
7. B. Chevallier-Mames. An efficient CDH-based signature scheme with a tight security reduction. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 511–526. Springer, Aug. 2005.
8. B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *26th FOCS*, pages 429–442. IEEE Computer Society Press, Oct. 1985.
9. R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. *ACM Trans. Inf. Syst. Secur.*, 3(3):161–185, 2000.
10. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital schemes. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 263–275. Springer, Aug. 1990.
11. S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.
12. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Aug. 1987.
13. A. M. Frieze. On the lagarias-odlyzko algorithm for the subset sum problem. *SIAM J. Comput.*, 15(2):536–539, 1986.
14. R. Gennaro. An improved pseudo-random generator based on discrete log. In M. Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 469–481. Springer, Aug. 2000.
15. R. Gennaro. An improved pseudo-random generator based on the discrete logarithm problem. *Journal of Cryptology*, 18(2):91–110, Apr. 2005.

16. R. Gennaro, S. Halevi, and T. Rabin. Secure hash-and-sign signatures without the random oracle. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 123–139. Springer, May 1999.
17. R. Gennaro, H. Krawczyk, and T. Rabin. Secure Hashed Diffie-Hellman over non-DDH groups. In C. Cachin and J. Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 361–381. Springer, May 2004.
18. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.
19. M. Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number (rump session). In I. Damgård, editor, *EUROCRYPT'90*, volume 473 of *LNCS*, pages 481–486. Springer, May 1990.
20. M. Girault, G. Poupard, and J. Stern. On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19(4):463–487, Oct. 2006.
21. E.-J. Goh and S. Jarecki. A signature scheme as secure as the Diffie-Hellman problem. In E. Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 401–415. Springer, May 2003.
22. E.-J. Goh, S. Jarecki, J. Katz, and N. Wang. Efficient signature schemes with tight reductions to the Diffie-Hellman problems. *Journal of Cryptology*, 20(4):493–514, Oct. 2007.
23. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.
24. L. C. Guillou and J.-J. Quisquater. A “paradoxical” identity-based signature scheme resulting from zero-knowledge. In S. Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 216–231. Springer, Aug. 1990.
25. S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 654–670. Springer, Aug. 2009.
26. R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of Cryptology*, 9(4):199–216, 1996.
27. J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In S. Jajodia, V. Atluri, and T. Jaeger, editors, *ACM CCS 03*, pages 155–164. ACM Press, Oct. 2003.
28. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In J. Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, Dec. 2008.
29. T. Koshihara and K. Kurosawa. Short exponent Diffie-Hellman problems. In F. Bao, R. Deng, and J. Zhou, editors, *PKC 2004*, volume 2947 of *LNCS*, pages 173–186. Springer, Mar. 2004.
30. J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. In *24th FOCS*, pages 1–10, 1983.
31. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In R. Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, Mar. 2008.
32. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Dec. 2009.
33. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, *LNCS*. Springer, 2012.

34. V. Lyubashevsky and D. Micciancio. Generalized compact Knapsacks are collision resistant. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 144–155. Springer, July 2006.
35. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In H. Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, May 2010.
36. S. Micali and L. Reyzin. Improving the exact security of digital signature schemes. *Journal of Cryptology*, 15(1):1–18, 2002.
37. D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.
38. D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Aug. 2011.
39. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, LNCS. Springer, 2012. Preliminary version at <http://eprint.iacr.org/2011/501>.
40. D. Micciancio and S. P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, Aug. 2003.
41. S. Patel and G. S. Sundaram. An efficient discrete log pseudo random generator. In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 304–317. Springer, Aug. 1998.
42. C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In S. Halevi and T. Rabin, editors, *TCC 2006*, volume 3876 of *LNCS*, pages 145–166. Springer, Mar. 2006.
43. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
44. J. M. Pollard. Kangaroos, monopoly and discrete logarithms. *Journal of Cryptology*, 13(4):437–447, 2000.
45. G. Poupard and J. Stern. Security analysis of a practical “on the fly” authentication and signature generation. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 422–436. Springer, May / June 1998.
46. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
47. B. Santoso, K. Ohta, K. Sakiyama, and G. Hanaoka. Improving efficiency of an ‘on the fly’ identification scheme by perfecting zero-knowledgeness. In J. Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *LNCS*, pages 284–301. Springer, Mar. 2010.
48. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
49. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, May 2011.
50. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Dec. 2009.
51. P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman key agreement with short exponents. In U. M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 332–343. Springer, May 1996.