A Lattice-Based Threshold Ring Signature Scheme

Pierre-Louis Cayrel[1], Richard Lindner[2], Markus Rückert[2], and Rosemberg Silva[3]

[1] CASED – Center for Advanced Security Research Darmstadt,
Mornewegstrasse, 32
64293 Darmstadt
Germany
pierre-louis.cayrel@cased.de
[2] Technische Universität Darmstadt
Fachbereich Informatik
Kryptographie und Computeralgebra,
Hochschulstraße 10
64289 Darmstadt
Germany
{rlindner,rueckert}@cdc.informatik.tu-darmstadt.de
[3] State University of Campinas (UNICAMP)
Institute of Computing
P.O. Box 6176
13084-971 Campinas
Brazil
rasilva@ic.unicamp.br

Abstract. In this article, we propose a new lattice-based threshold ring signature scheme, modifying Aguilar's code-based solution to use the short integer solution (SIS) problem as security assumption, instead of the syndrome decoding (SD) problem. By applying the CLRS identification scheme, we are also able to have a performance gain as result of the reduction in the soundness error to 1/2 per round. Such gain is also maintained through the application of the Fiat-Shamir heuristics to derive signatures from our identification scheme. From security perspective we also have improvements, because our scheme exhibits a worst-case to average-case reduction typical of lattice-based cryptosystems. This gives us confidence that a random choice of parameters results in a system that is hard to break, in average.

Keywords: Identification scheme, lattice-based cryptography, SIS problem, threshold ring signature, zero-knowledge

[3] Supported by The State of São Paulo Research Foundation under grant 2008/07949-8.

1 Introduction

The concept of allowing a member of a group to anonymously sign documents on behalf of the entire group was created by Chaum and van Heyst [13]. In the original scheme, however, there is an entity called group manager that can reveal the identity of the actual signer. A variation of this concept, proposed by Rivest, Shamir and Tauman [27], and called Ring Signature, prevents anonymity from being revoked. It was further extended by Bresson, Stern and Szydlo into a Threshold Ring Signature Scheme, which consists of a protocol that enables a group composed of t people belonging to a larger group of size N to jointly and anonymously sign a given document [9]. The minimum size t of the subgroup required to generate a valid signature is a parameter enforced by the protocol. Aguilar, Cayrel, Gaborit and Laguillaumie [2] made a construction of a TRSS scheme, achieving signature sizes and time complexities that are linear in N and independent of t. Besides, it is existentially unforgeable under chosen message attack in the random oracle model. Let us call their construction TRSS-C (short for Threshold Ring Signature Scheme using Codes).

It is based on error-correcting codes, and is the best known threshold ring signature scheme, from time complexity perspective. Differently from its number-theoretic predecessors, which exhibited a complexity of $\mathcal{O}(t.N)$ (where N is the size the group of users, and t is the size of the sub-group willing to sign a message), TRSS-C has a complexity given by $\mathcal{O}(N)$, clearly independent of the number of users that want to jointly sign a message. However, as seen in [19], signature schemes derived from identification schemes with high soundness error tend to be inefficient in terms of signature size. The same happens to TRSS-C.

1.1 Our Contribution

Our work consists of a lattice-based threshold ring signature scheme, combining Aguilar's [2] and Cayrel's [11] results, and is based on an identification scheme that has lower soundness error. This enables a performance gain due to the smaller number of rounds of execution, as well as an achievement of shorter signatures. The security of our scheme is based on the hardness of the lattice SIS problem. Provided that a suitable set of parameters is used, a reduction from worst-case in Gap-SVP to average-case in SIS is preserved. Such reduction, typical of lattice-based cryptosystems, gives confidence that the construction is safe, even for randomly chosen parameters. Aiming an easier notation, along the text our scheme will be referred to as TRSS-L (Threshold Ring Signature Scheme Based on Lattices).

1.2 Related work

Code-Based Threshold Ring Signature Schemes
The TRSS-C scheme relies on the hardness of the minimum distance (MD) problem and the existence of collision resistant hash functions as security assumptions [2]. It generalizes the identification scheme designed by Stern [29] and inherits

its same limits as far as signature sizes are regarded, when applying the Fiat-Shamir heuristics: a high number of rounds in order to read a specified security level.

The group of signers is composed of t entities out of a group of N. One of the signers is chosen as leader, and executes $t-1$ simultaneous Stern's protocols with the other signers. Such leader applies the Fiat-Shamir heuristic over the generalized Stern's scheme in order to generate signatures. He also generates master commitments, hiding the identity of the signers by means of a product of permutations.

Dallot and Verganaud [15] have also proposed a code-based threshold ring signature scheme. It is not derived from an identification, differently from TRSS-C. Rather, it bears similarly with the CFS signature scheme [14] in the sense of requiring a number of decoding operations that grows with the factorial of the number of errors that its underlying Goppa code can correct. Therefore, though the signatures are short, a considerable computational effort is necessary to generate them. Plus, as opposed to our construction, Dallot's uses trapdoors.

Lattice-Based Signature Schemes
To the best of our knowledge, our threshold ring signature scheme is the first lattice-based. Recently, Brakerski and Kalai [8] presented a generic framework for constructing signature schemes, including ring and identity types, in the standard model. They presented an example based on SIS. Their work does not include threshold constructions, though.

1.3 Organization of the document

This paper is divided as follows. In Section 2, we give general definitions regarding lattices, identification and ring signature schemes. Then, we describe our lattice-based Threshold Ring Signature Scheme in Section 3. Subsequently, we provide demonstrations of security of our scheme in Section 4. Afterwards, a discussion of performance aspects of the scheme follows in Section 5. Lastly, an appreciation of the scheme and future lines of work are given in Section 6.

This section presented an overview of lattice-based signatures systems and how our proposal relates to them. The next one lists the definitions of some concepts that we use along the text in order to detail the design of our signature scheme. It dedicates special attention to the aspects related to performance and security.

2 Preliminaries

In this part of the article, we give the definition of the hard lattice problem connected with the security of our signature scheme. Furthermore, we detail the code-based construction from which our design derives.

The advent of quantum computers poses a serious threat to Cryptography, due to an algorithm devised by Shor [28] which is able to calculate in polynomial-time prime factorization and discrete logarithms. Post-Quantum Cryptography is a denomination given to the sub-areas that are known to be still resilient to quantum computers. Systems built upon lattice hard problems are included on them.

2.1 Lattices

Besides resilience to known quantum attacks, strong security proofs are an important feature of lattice-based constructions. Here, we show the basic definitions applied in the design of our threshold ring signature scheme.

Definition 1. *A lattice is a discrete subgroup of $\mathbb{R}^m$ with dimension $n \leq m$. In general, for cryptographic applications, it is restricted to $\mathbb{Z}^m$. It can be represented by a basis comprising n linear independent vectors of $\mathbb{R}^m$.*

We define below the hard problems in the lattice domain that serve as security assumptions in the schemes described in this article. The definitions make use of the max-norm or ℓ_∞.

Definition 2. ***(Shortest Vector Problem - SVP)*** *Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find a non-zero lattice vector $\mathbf{Bx}$ such that $\|\mathbf{Bx}\| \leq \|\mathbf{By}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n \backslash \{0\}$.*

Definition 3. ***(Closest Vector Problem - CVP)*** *Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, find $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\|$ is minimum.*

These two problems also admit approximate formulation, as stated below for a factor γ.

Definition 4. ***(Approximate SVP$_\gamma$)*** *Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, find a non-zero lattice vector $\mathbf{Bx}$ such that $\|\mathbf{Bx}\| \leq \gamma \cdot \|\mathbf{By}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n \backslash \{0\}$.*

Definition 5. ***(Approximate CVP$_\gamma$)*** *Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, find $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma \cdot \|\mathbf{By} - \mathbf{t}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n$.*

In addition to the exact and approximate formulations, one can also state these problems as promises, as outlined below.

Definition 6. ***(GapSVP$_\gamma$)*** *It is a promise problem for which the YES and NO instances are defined as:*

- *YES: pairs $(\mathbf{B}, r)$ where $\mathbf{B} \in \mathbb{Z}^{m \times n}$ is a lattice basis and $r \in \mathbb{Q}$ is a rational number such that $\|\mathbf{Bz}\| \leq r$ for some $\mathbf{z} \in \mathbb{Z}^n \backslash \{0\}$.*
- *NO: pairs $(\mathbf{B}, r)$ where $\mathbf{B} \in \mathbb{Z}^{m \times n}$ is a lattice basis and $r \in \mathbb{Q}$ is a rational number such that $\|\mathbf{Bz}\| > \gamma \cdot r$ for all $\mathbf{z} \in \mathbb{Z}^n \backslash \{0\}$.*

Definition 7. ***(GapCVP$_\gamma$)*** *It is a promise problem for which the YES and NO instances are defined as:*

- *YES: triplets* $(\mathbf{B}, \mathbf{t}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *is a lattice basis,* $\mathbf{t} \in \mathbb{Z}^m$ *is a vector, and* $r \in \mathbb{Q}$ *is a rational number such that* $\|\mathbf{Bz} - \mathbf{t}\| \leq r$ *for some* $\mathbf{z} \in \mathbb{Z}^n$.
- *NO: triplets* $(\mathbf{B}, \mathbf{t}, r)$ *where* $\mathbf{B} \in \mathbb{Z}^{m \times n}$ *is a lattice basis,* $\mathbf{t} \in \mathbb{Z}^m$ *is a vector, and* $r \in \mathbb{Q}$ *is a rational number such that* $\|\mathbf{Bz} - \mathbf{t}\| > \gamma \cdot r$ *for all* $\mathbf{z} \in \mathbb{Z}^n$.

A thorough discussion on the hardness of these problems can be found in [23].

Definition 8. ***(Short Integer Solution - SIS)*** *Given* $\mathbf{A} \in \mathbb{Z}^{n \times m}$ *and a prime number* q*, find a vector* $\mathbf{v}$ *in the lattice* $\Lambda_q^{\perp} = \{\mathbf{x} \in \mathbb{Z}^m \ : \ \mathbf{Ax} = \mathbf{0} \bmod q\}$ *with length limited by* $\|\mathbf{v}\| \leq L$.

From the perspective of cryptography, one of the most interesting results involving lattices consists in showing that breaking a randomly chosen instance in some schemes is at least as hard as finding solutions for worst-case instances of hard lattice problems. In [3] and [4], for example, Ajtai uses computationally intractable approximations of lattice problems as building blocks of cryptosystems.

As far as saving space to represent lattice basis is regarded, Micciancio showed through cyclic lattices that it is possible to reach storage that grows linearly with the lattice dimension [22]. His one-way compression functions also achieved the collision resistance property with the use of ideal lattices, as seen in [20]. Such work also specified the conditions that should be satisfied in order to assure the existence of average-case/worst-case connection.

Lattice applications to identification purposes have also provided good results. For instance, in Lyubashevsky's identification scheme, provably secure against active attacks [18], the hardness assumption is the difficulty in approximating the shortest vector in all lattices to within a factor of $\tilde{O}(n^2)$, where n is a security parameter corresponding to the lattice rank over which the hard problem is defined. The parameters seen there, however, are somewhat big to be considered practical.

By using weaker security assumptions, on the other hand, one can achieve parameters that are small enough to be used in practice, as seen in the identification scheme proposed by Kawachi et al. in [16]. In this later work, the authors suggest to use approximate Gap-SVP or SVP within $\tilde{O}(n)$ factors. Similar approach to improve efficiency was used in CLRS [11], which is one of the pillars of our signature scheme.

2.2 Ideal Lattices

In spite of the good security properties that can be achieved through lattice constructions, one issue has historically been presented as obstacle for their adoption: the huge key sizes. Through ideal lattices, this subject was sucessfully addressed in [20] and [18].

Definition 9. ***(Ideal Lattice)*** *Given a lattice L, such that $L \subseteq \mathbb{Z}^n$, a polynomial $f(X) = f_0 + \ldots + f_{n-1}X^{n-1} + X^n$ and a mapping $\phi_f(v_0, \ldots, v_{n-1}) \longmapsto v_0 + v_1X + \ldots + v_{n-1}X^{n-1} + f(X)\mathbb{Z}[X]$. L is considered an ideal lattice, if $\phi_f(L)$ is an ideal in $R_f = \mathbb{Z}[X]/\langle f(X)\rangle$. Likewise, if I is an ideal in R_f, then its image L under $\phi_f^{-1}(I)$ is an ideal sublattice of $\mathbb{Z}^n$.*

Not only does this kind of lattice allow compact basis representation, but also enables efficient use of FFT to carry out operations over its elements. The signature scheme that we propose in this article can profit from these features, when implemented over this kind of lattice.

2.3 Threshold Ring Signatures

We depict here a threshold ring signature scheme, listing its basic operations and main features.

Definition 10. ***(Threshold Ring Signature)*** *Given an input security parameter λ, an integer n representing the number of users, and an integer t representing the minimum number of users required to jointly generate a valid signature, threshold ring signature scheme is a set of four algorithms described as below*

- *Setup: generates the public parameters corresponding to the security parameter.*
- *Key Generation: creates pairs of keys (s, p) (one for each user that composes the ring), secret and public respectively, related by a hard problem.*
- *Signature Generation: on input a message m, a set of public keys $\{p_1, \ldots, p_n\}$ and a sub-set of t secret keys, it issues a ring signature σ.*
- *Signature Verification: on input a message m, its ring signature σ and a set of public keys $\{p_1, \ldots, p_n\}$, it outputs 1 in case the signature is valid, and 0 otherwise.*

Definition 11. ***(Existentially Unforgeable)*** *A threshold ring signature with parameters (λ, n, t) is considered ϵ-existentially unforgeable, if no probabilistic polynomial time adversary $\mathcal{A}$ can generate a valid signature for any message m with probability higher than ϵ, under the conditions below :*

- *$\mathcal{A}$ knows all n public keys;*
- *$\mathcal{A}$ knows up to $t-1$ private keys;*
- *$\mathcal{A}$ has access to pairs message-signature (m', σ) with $m \neq m'$.*

Definition 12. ***(Unconditionally Source-Hiding)*** *A threshold ring signature with parameters (λ, n, t) is considered to have the anonymity property of unconditionally source-hiding if, for any message m, it is possible to generate the same signature with two different sub-sets of signers having cardinality t.*

The TRSS-C satisfies these two properties, as proved in [2]. So does our scheme, which is built with a very similar structure.

2.4 CLRS Identification Scheme

Our TRSS-L derives its organization from TRSS-C. Both are built on top of identification schemes via standard Fiat-Shamir transformations. We describe here the one used by our scheme. It is called CLRS, and was delineated by Cayrel et al. in [12]. It is lattice-based and aims to deal the soundness error matter that was seen to impact the TRSS-C performance.

As previously mentioned, the TRSS-C employs the code-based predecessor proposed by Stern [29] as one of its pillars. Its security is based on the hardness of the syndrome decoding problem. An improvement over this scheme, exploring dual constructions, was conceived by Véron [30], achieving better communication costs and better efficiency. As the basic Stern's structure, however, its soundness error is still 2/3.

By modifying the way the commitments are calculated, incorporating a value chosen at random by the verifier, Cayrel and Véron [12] were able to bound the cheating probability within a given round to 1/2, achieving thus better communication costs. The approach followed is similar to that shown in Figure 2, which corresponds to the CLRS design, that uses the SIS problem as security basis. Both schemes have a soundness error of 1/2.

The CLRS employs a 5-pass structure, and corresponds to a zero-knowledge interactive proof that an entity, designated by prover $\mathcal{P}$, knows a solution to a hard instance of the inhomogeneous SIS problem. The exact proof for the properties of completeness, soundness and zero-knowledge can be found on [12]. The arguments used in its construction and those used for GCLRS in Subsection 4.1 are alike, regarding the completeness and zero-knowledge properties. The soundness property can be proved by absurd, using the fact that a cheating prover able to correctly answer strictly more than 1/2 of the possible questions posed by the verifier (in the form of $\alpha \times b$, with $\alpha \in \mathbb{Z}_q^*$ and $b \in \mathbb{F}_2$) will have to answer to, given a fixed pair of commitments c_0 and c_1 occurring in two different rounds, both possible values of b. Provided that the commitment function used is collision resistant, this would imply that the cheating prover is able to solve the SIS, that is known to be hard.

The security (in bits) associated with a given instance of CLRS is, first of all, determined by the parameters that specify the underlying SIS problem. The second aspect to be taken into account is the overall soundness error, which is a function of the number of rounds of execution of the identification scheme. In Table 2 we list the parameters used in an instantiation of our scheme.

2.5 Permutations

The use of permutations, as described below, is of the main tools used in the proposal of Cayrel and Véron [12] to lower the soundness error in a 5-pass construction. It allows the prover to send permutations of q-ary vectors build from private information, without revealing the exact values of the individual coordinates, because they are permuted as well. A similar approach was followed in the CLRS scheme to keep private information concealed when prover and verifier

KeyGen:
$\mathbf{x} \xleftarrow{\$} \{0,1\}^m$, s.t. $\mathrm{wt}(\mathbf{x}) = m/2$
$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$
$\mathbf{y} \longleftarrow \mathbf{A}\mathbf{x} \bmod q$
$\textsc{Com} \xleftarrow{\$} \mathcal{F}$, suitable family of commitment functions
Output $(\mathrm{sk}, \mathrm{pk}) = (\mathbf{x}, (\mathbf{y}, \mathbf{A}, \textsc{Com}))$

Fig. 1. Key generation algorithm, parameters n, m, q are public.

Prover $\mathcal{P}$ $(\mathrm{sk}, \mathrm{pk}) = (\mathbf{x}, (\mathbf{y}, \mathbf{A}, \textsc{Com}))$		**Verifier** $\mathcal{V}$ $\mathrm{pk} = \mathbf{y}, \mathbf{A}, \textsc{Com}$
$\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \sigma \xleftarrow{\$} S_m$		
$\mathbf{r}_0 \xleftarrow{\$} \{0,1\}^n, \mathbf{r}_1 \xleftarrow{\$} \{0,1\}^n$		
$c_0 \longleftarrow \textsc{Com}(\sigma \,\|\| \mathbf{A}\mathbf{u}, \mathbf{r}_0)$		
$c_1 \longleftarrow \textsc{Com}(\mathbf{P}_\sigma \mathbf{u} \,\|\| \mathbf{P}_\sigma \mathbf{x}, \mathbf{r}_1)$	$\xrightarrow{c_0, c_1}$	
	$\xleftarrow{\alpha}$	$\alpha \xleftarrow{\$} \mathbb{Z}_q^*$
$\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha\mathbf{x})$	$\xrightarrow{\beta}$	
	$\xleftarrow{b}$	$b \xleftarrow{\$} \{0,1\}$
If $b = 0$:	$\xrightarrow{\sigma, \mathbf{r}_0}$	$c_0 \stackrel{?}{=} \textsc{Com}(\sigma \,\|\| \mathbf{A}\mathbf{P}_\sigma^{-1}\beta - \alpha\mathbf{y}, \mathbf{r}_0)$
		$\sigma \stackrel{?}{\in} S_m$
Else:	$\xrightarrow{\mathbf{P}_\sigma \mathbf{x}, \mathbf{r}_1}$	$c_1 \stackrel{?}{=} \textsc{Com}(\beta - \alpha\mathbf{P}_\sigma\mathbf{x} \,\|\| \mathbf{P}_\sigma\mathbf{x}, \mathbf{r}_1)$,
		$\mathbf{P}_\sigma \mathbf{x} \stackrel{?}{\in} \{0,1\}^m$

Fig. 2. CLRS Identification protocol

exchange messages over the communication channel, which can be monitored by adversaries.

Definition 13. ***(Constant n-block permutation)*** *It is a permutation Σ that acts on N blocks of size n, each of which is considered as a unit.*

Definition 14. ***(n-block permutation)*** *Given a vector $v = (V_1, \ldots, V_N)$ of size nN, a family of N permutations $\sigma_i \in S_n$ and a constant n-block permutation Σ, an n-block permutation is defined as the product permutation $\Pi = \Sigma \circ \sigma$ that acts on N blocks of size n as*

$$\Pi(v) = \Sigma(\sigma_1(V_1), \ldots, \sigma_N(V_N)).$$

We have seen in this segment some important concepts from the lattice theory that are necessary to understand the security and performance aspects of

our threshold ring signature scheme, and how it compares to its code-based counterpart. In the sequence, we detail our design, by listing and explaining the algorithms that constitute it.

3 Our Lattice-Based Threshold Ring Signature Scheme

We have described and defined the lattice problems and concepts that work as basis for our scheme in the previous section. Now, we detail the algorithms that comprise this scheme.

Taking SIS as security assumption, we modify TRSS-C [2] and obtain a construction that is more efficient than other similar lattice-based solutions, to the best of our knowledge. In order to do so, instead of using Stern's identification scheme as basis, we employ the CLRS scheme [11], which has a lower soundness error (1/2, instead of 2/3) and enables the resulting construct to reach a security goal in fewer rounds of execution.

Some lattice-based identification scheme (see [25], [18] and [17]) have time complexity and public key sizes efficiently given by $\mathcal{O}(n)$. However, they share an inefficiency: for each bit of challenge sent by the verifier, a response with size $\mathcal{O}(n)$ has to be provided by the prover. This implies in huge signature sizes when directly applying the Fiat-Shamir heuristic. The same drawback can be found in TRSS-C. This means that the number of rounds executed by such scheme is given at least by the number of bits of the hash function value (applied to commitments concatenated to the message). Our scheme addresses the first factor by splitting the challenge in two pieces: the messages $\alpha \in \mathbb{Z}_q^*$ and $b \in \mathbb{F}_2$ represented in Figure 2. This bears similarity with the identification scheme described in [19], where the challenge-like bits are assigned to an element of a polynomial ring. Dividing the hash bits over structures that are several bits wide (given by the number of bits to represent α and b, in our case) has as positive effect a fewer number of rounds to generate a signature.

The other factor that impacts the number of rounds of execution is the soundness level required. The higher of the two such values will have to be executed in order to achieve both security goals.

3.1 Adaptations made to the CLRS scheme

In the code-based threshold ring signature scheme proposed by Aguilar et al. [2], they replaced the syndrome decoding problem in the underlying Stern's identification scheme by the minimum distance problem in order to preserve anonymity. Instead of having $\mathbf{H}\mathbf{x}^T = y$, with check matrix $\mathbf{H}$ and syndrome $\mathbf{y}$ public, and word $\mathbf{x}$ private with a known Hamming weight p, they used $\mathbf{H}\mathbf{x}^T = \mathbf{0}$, what means that the secret keys now correspond to codewords $\mathbf{x}$ with Hamming weight specified by an input parameter. Plus, when the leader is computing the master commitments he can easily satisfy this equation by picking $\mathbf{x} = \mathbf{0}$ for the users that are not signing the message.

For the same reasons, we make an adaptation of the original CLRS construction, so that it can be used in our threshold ring signature scheme. Initially, each user had a key-pair represented by a secret key $\mathbf{x} \in \mathbb{F}_2^m$ and a private key $\mathbf{y} \in \mathbb{Z}^n$ related by the ISIS (Inhomogeneous SIS) problem $\mathbf{Ax} = \mathbf{y} \bmod q$, with $\mathbf{A} \in \mathbb{Z}^{n \times m}$. The secret key can be chosen at random, from a set of binary words of known Hamming weight $m/2$. This can be rewritten as $[\mathbf{A}; -\mathbf{y}][\mathbf{x}; 1]^T = \mathbf{0} \bmod q$. Making $\mathbf{A'} = [\mathbf{A}; -\mathbf{y}]$ and $\mathbf{x'} = [\mathbf{x}; 1]$, we have $\mathbf{A'x'} = \mathbf{0} \bmod q$. This is analogous to the code-based construction. It works as if every user had the same public key value: the null vector.

In Algorithm 1, the individual matrices $\mathbf{A}_i$ are calculated as described in the paragraph above, so that $\mathbf{A}_i\mathbf{x}_i = 0 \bmod q$. In Section 4, where the security proofs are given, we show that in order to break our system, one must obtain $\mathbf{x}_i$ given $\mathbf{A}_i$, which on its turn implies in being able to solve the SIS problem in the worst case. Given that this latter problem is known to be hard, our system is consequently difficult to break.

The memory size involved in storing the matrices $\mathbf{A}_i$ can be highly optimized by using ideal lattices. As discussed in Section 2, the space required by this kind of lattice grows linearly with the dimension, up to a logarithmic factor.

3.2 Applying Fiat-Shamir heuristic

From the generalized identification scheme described in Algorithm 1, we obtain a signature scheme by putting a random oracle in the place of the verifier. The source of the random values to be used with α and b is the hash value of the message to be signed concatenated with the commitments of the current round, in order to make difficult to obtain successful forgery .

Using the honest-verifier zero-knowledge nature of our underlying identification scheme and the security results stated by Pointcheval and Stern at [26] and Abdalla et al. [1] regarding the Fiat-Shamir heuristic, we can establish the security of our signature scheme in the random oracle model. In order to do so, we are making the assumption that the security results associated with signature schemes obtained from canonical identification schemes (three passes) via Fiat-Shamir are also valid for our scheme, even though its underlying identification scheme is not canonical (five passes). Their similarity resides in a commitment-challenge-answer structure.

3.3 Description of our threshold ring signature scheme

Our TRSS-L is composed of four algorithms: Setup, Key Generation, Signing, Verification. Though its structure is similar to that of the code-based scheme described in [2], the underlying identification scheme and hardness assumptions are considerably different, as emphasized in the discussions regarding security and performance, developed in Sections 4 and 5, respectively.

The **Setup** algorithm, on input a security parameter k, issues the parameters n, m, q that are used by the other three algorithms, and are necessary for the definition of the lattices and their operations.

The **Key Generation** algorithm, on input parameters k, n, m, q, N, generates the N pairs of public and private keys $(\mathbf{x}_i, \mathbf{A}_i)$, with $i \in \{0, \ldots, N-1\}$. All the private keys are binary vectors with Hamming weight $m/2+1$ and constitute solutions for the SIS problem $\mathbf{A}_i\mathbf{x}_i = 0 \bmod q$. The public keys are the matrices $\mathbf{A}_i \in \mathbb{Z}_q^{n\times(m+1)}$.

The **Signing** algorithm takes as input a message to be signed, the set of N public keys, t private keys (corresponding to the users willing to sign the message), and a hash function that computes the digest of the message concatenated with the commitments in a given round. This algorithm corresponds to the application of the Fiat-Shamir heuristics to the GCLRS scheme detailed by Algorithm 1. A group of t users, one of which is the leader L, interact in order to generate a signature. The generalized scheme works as follows: each pair $(signer_i, leader)$ executes the CLRS identification scheme, where $signer_i$ plays as prover and leader L acts as verifier, sharing the same challenges α and b. On its turn, the pair (leader, Verifier) runs an identification scheme as well, where the commitments and answers are compositions involving the values received by the leader from the other signers. As for the non-signing users, the leader generates surrogate private keys comprised of null vectors (which are trivial solutions of the SIS problem). The leader applies block permutations over theses individual values in order to achieve the goal of anonymity. The signature consists of the transcript of the interaction between the leader and the verifier.

The **Verification** algorithm takes as input the public keys of the N users and the signature. Such signature constitutes a communication transcript of a sequence of rounds of the GCLRS scheme. The verification consists in check, depending on the value of the challenges, that the corresponding commitment is correct for every round. The signature is accepted if the check was successful in every round, and rejected otherwise.

The security aspects of the construction corresponding to the algorithms that comprise our scheme will be discussed next. We also give demonstrations that our design is safe, and relate it to the CLRS signature scheme upon which it relies.

4 Security

The previous section described the algorithms that comprise our system. In the sequence, we show them to be secure, with worst-case to average-case reductions that are typical in lattice-based systems.

4.1 Honest-Verifier Zero-Knowledge Proof of Knowledge

We now prove that the Algorithm 1 (GCLRS, short for Generalized CLRS) constitutes a zero-knowledge proof of knowledge of that a group of t-out-of-N users knows t different pairs (secret key, public key). The first element of the pair is a binary vector $\mathbf{x}_i$ of length $m+1$ and Hamming weight $m/2+1$ and the second is a matrix $\mathbf{A}_i \in \mathbb{Z}^{n\times(m+1)}$, such that $\mathbf{A}_i\mathbf{x_i} = 0 \bmod q$, with $i \in \{0, \ldots, N-1\}$.

Algorithm 1 Generalized CLRS Identification Scheme (GCLRS)

procedure IDENTIFICATION SCHEME
▷ $U' = \{\text{users}\}$ and $S' = \{\text{signers}\}$, with $S' \subset U'$, $|S'| = t$ and $|U'| = N$
▷ Prover (pass 1): computes commitments
▷ **Commitment**: performed by signers S', which include the leader L
for Each signer $i \in S'$ **do** ▷ Compute commitments
$\sigma_i \xleftarrow{\$} S_{m+1}$, $\mathbf{u}_i \xleftarrow{\$} \mathbb{Z}_q^{m+1}$, $\mathbf{r}_{0,i} \xleftarrow{\$} \{0,1\}^n$ and $\mathbf{r}_{1,i} \xleftarrow{\$} \{0,1\}^n$
$c_{0,i} \longleftarrow \text{COM}(\sigma_i \,||\, \mathbf{A}_i\mathbf{u}_i, \mathbf{r}_{0,i})$ and $c_{1,i} \longleftarrow \text{COM}(\sigma_i(\mathbf{u}_i) \,||\, \sigma_i(\mathbf{x}_i), \mathbf{r}_{1,i})$
Send $c_{0,i}$ and $c_{1,i}$ to L
end for
For the non-signers $j \in U' \setminus S'$, L performs the same, but with $\mathbf{x}_j \leftarrow 0$
L chooses a random constant n-block permutation on N blocks Σ .
L computes the *master commitments* $C_0 = \text{COM}(\Sigma \,||\, c_{0,1} \,||\, \ldots \,||\, c_{0,N}, \mathbf{r}_0)$ and $C_1 = \text{COM}(\Sigma(c_{1,1}, \ldots, c_{0,N}), \mathbf{r}_1)$ and sends them to V
▷ Verifier (pass 2): imposes a value to be used to verify previous commitments
V sends $\alpha \xleftarrow{\$} \mathbb{Z}_q^*$ to L, which passes it to S'.
▷ Prover (pass 3):
for Each signer $i \in S'$ **do**
$\boldsymbol{\beta}_i \leftarrow \sigma_i(\mathbf{u}_i + \alpha\mathbf{x}_i)$
end for
For the non-signers $j \in U' \setminus S'$, L performs the same, but with $\mathbf{x}_j \leftarrow 0$
L sends $\boldsymbol{\beta} = \Sigma(\boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_{N-1})$ to V.
▷ **Challenge**:
▷ Verifier (pass 4): makes a challenge to leader L
V sends $b \xleftarrow{\$} \{0,1\}$ to L, which propagates it to S'.
▷ **Answer**:
▷ Prover (pass 5): reveals private information for the current round
for Each signer $i \in S'$ **do**
Reveal to L either σ_i or $\sigma_i(\mathbf{x}_i)$, when $b = 0$ or $b = 1$, respectively.
end for ▷ For non-signing users, L has chosen default values at the commitment phase.
if b is 0 **then**
Set $\sigma = (\sigma_0, \ldots, \sigma_{N-1})$
L reveals $\Pi = \Sigma \circ \sigma$ and $\Pi(\mathbf{r}_{0,0}, \ldots, \mathbf{r}_{0,N-1})$ to V
else
Set $\Pi(\mathbf{x}) = \Sigma(\sigma_1(\mathbf{x}_1), \ldots, \sigma_{N-1}(\mathbf{x}_{N-1}))$
L reveals $\Pi(\mathbf{x})$ and $\Pi(\mathbf{r}_{1,0}, \ldots, \mathbf{r}_{1,N-1})$ to V
end if
▷ **Verification**: correctness of *master commitments*, permutations and Hamming weight.
if b is 0 **then** ▷ A is matrix whose diagonal corresponds to the public keys A_i
V checks that $C_0 \stackrel{?}{=} \text{COM}(\Sigma \,||\, \mathbf{A}\Pi^{-1}(\boldsymbol{\beta}) \,||\, \mathbf{r}_0)$ and that Π is well formed.
else
V checks that $C_1 \stackrel{?}{=} \text{COM}(\boldsymbol{\beta} - \alpha\Pi(\mathbf{x}) \,||\, \Pi^{-1}(\boldsymbol{\beta}) \,||\, \mathbf{r}_1)$ and that $\Pi(\mathbf{x})$ has Hamming weight $t(m/2 + 1)$.
end if
end procedure

This algorithm can be seen as a composition of t simultaneous executions of the CLRS identification schemes described in Figure 2, which has already been demonstrated to be secure by Cayrel et al. in [11] in the active attack model. We will use this fact and discuss only the security of the composition described in Algorithm 1.

By interacting as verifier with each of the $t-1$ other signers and following the GCLRS protocol, the leader learns nothing about their secret keys, except that they are valid. When playing the role of prover, the leader L, in his interaction with the verifier V, does not leak any private information, either. All that V learns is that t of the users belonging to U have participated to generate a binary vector $\mathbf{X}$ of dimension $N(m+1)$ and Hamming weight $t(m/2+1)$ such that $\mathbf{AX} = 0 \bmod q$, where $\mathbf{A}$ is defined as below:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 & 0 & \cdots & 0 \\ 0 & \mathbf{A}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{A}_{N-1} \end{bmatrix}$$

Lemma 1. *Under the assumption of the hardness of the SIS problem, finding a vector* $\mathbf{v}$ *with length* $N(m+1)$ *and Hamming weight* $t(m/2+1)$ *satisfying* $(\mathbf{Av} = 0 \bmod q)$*, with* $\mathbf{A}$ *defined as above, such that the* N *blocks of length* $m+1$ *that comprise* $\mathbf{v}$ *have either 0 or* $m/2+1$ *as Hamming weight, is also hard.*

Proof: By construction of $\mathbf{A}$ and $\mathbf{v}$, finding a solution of $\mathbf{Av} = 0 \bmod q$ is at least as hard as finding a local solution $\mathbf{v}_i$ to $\mathbf{A}_i\mathbf{v}_i = 0 \bmod q$ with Hamming weight$(\mathbf{v_i}) = m/2+1$, and this latter problem problem is hard under the SIS hardness assumption. □

Theorem 1. *The GCLRS scheme is an honest verifier zero-knowledge proof of knowledge, with soundness error no greater than 1/2, that a group of* t *signers knows a vector* v *of length* $N(m+1)$ *and Hamming weight* $t(m/2+1)$*, such that each of the* N *blocks of size* m *either weights* $m/2+1$ *or zero. The scheme is secure in the random oracle model under the SIS problem hardness assumption.*

Proof:
Completeness: An honest set of signers is always able to reveal to the leader the information necessary to compute the individual commitments $c_{0,i}$ or $c_{1,i}$, by revealing σ_i or $\sigma_i(\mathbf{x}_i)$ respectively, depending on the challenge sent by the verifier V. For each component $i \in \{0, \ldots, N-1\}$ of the group, we have either weight$(\mathbf{x}_i) = m/2+1$, when the user is signing the message, or weight$(\mathbf{x}_i) = 0$ otherwise. The length of each of those vectors is $m+1$. The leader L, on his turn, is always able to disclose either Π or $\Pi\mathbf{x}$ under the same challenge values. The vector $\mathbf{x}$ is comprised of N components $\mathbf{x}'_i$ that are permutations of $\mathbf{x}_i$, and hence have the same weight. Therefore, $\mathbf{x}$ has overal length $N(m+1)$ and weight $t(m/2+1)$.

Soundness: The soundness error is bounded away from 1, and it cannot be higher than 1/2. The GCLRS scheme is composed of $t-1$ CLRS instances involving $t-1$ distinct pairs (prover, verifier). If GCLRS has a soundness error strictly above 1/2, then a cheating prover can devise a strategy to beat the system with a success probability also above 1/2. Given that CLRS can be reduced to GCLRS (it suffices to make all singing instances equal, and follow the procedure described in Subsection 3.1), we can use the cheating strategy to beat the CLRS scheme also with probability above 1/2. However, this is absurd under the assumption of SIS hardness and the commitment function collision resistance, as seen in [11].

Zero-Knowledge (ZK): Let us build a simulator as described below:

1 $Coin \xleftarrow{\$} \{0,1\}$
2 Prepare to answer a challenge that is equal to $Coin$ as follows:
   - For $Coin = 0$, pick $\mathbf{x}_i$ satisfying $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i$, but with high weight, for the t elements of the signing set. According to the way that the parameters were chosen, such solution exists with high probability and is not hard to find. Regarding the other $N-t$ components, just set $\mathbf{x}_i = 0$.
   - For $Coin = 1$, pick $\mathbf{x}_i$ with weight exactly $m/2+1$ for the t elements, but without satisfying $\mathbf{y}_i = \mathbf{A}_i\mathbf{x}_i$. The remaining components will be set as null vector.
3 $b \xleftarrow{\$} \{0,1\}$
4 If $Coin$ and b have the same value, register the current round as part of the signature. Otherwise, go back to step 1.
5 Repeat loop until the signature is complete.

The signature generated as above does not involve the actual values of the individual private keys. Besides, the uniformly random choices that are made and registered as signature follow the same distribution of a real one. Hence, looking at the real signature we learn nothing more than what we could have learnt from a simulated one. Therefore, with the simulator constructed as above, we conclude that the zero-knowledge property is observed.

□

Theorem 1 implies that the TRSS scheme is existentially unforgeable under chosen message attack in the random oracle model, assuming the hardness of the SIS problem and the existence of a collision resistant commitment function. Given the zero-knowledge property of the scheme, no information is learnt about the private keys, given access to previous signatures of different messages. Besides, even if an adversary is given $t-1$ private keys, he will not be able to generate a valid signature, unless he is able solve SIS in order to obtain an extra private key, different from those that he already possesses.

Theorem 2. *Our lattice-based threshold ring signature scheme is unconditionally source hiding.*

Proof: Our algorithm is structurally similar to TRSS-C [2]. In both, the entity playing the role of leader creates a secret vector which blockwise corresponds

to either permutations of individual private keys or null vectors. Besides, all the individual private keys are binary vectors with exactly the same Hamming weight, and the commitments correspond to one-time pad of the secrets. Hence, the distribution of the commitments associated with a given signer are indistiguishable from a random one, and also from the distribution related to a different user. Therefore, any subset of t users can produce a given signature with equal probability. □

After having discussed security aspects of our threshold ring signature scheme and related it with the hardness of average instances of the SIS problem (to which are proven to exist reductions from worst-case instances of the GapSVP problem), we next show that the design decisions taken allow gains in efficiency as well.

5 Performance

The previous section gave evidences and proofs that our system is safe. We now show that the our design choices result in a construction that is also efficient.

Our scheme can outperform TRSS-C both in terms of signature size and speed of signature generation. These two variables are linked and their reduction represents the combined effect of three different factors discussed below: smaller soundness error, wider challenge values, and use of FFT for performing multiplications.

Let us suppose that TRSS-C has a round communication payload of PL_1, whereas the corresponding value for our scheme is PL_2. The soundness error for the two schemes are $SE_1 = 2/3$ and $SE_2 = 1/2$, respectively. In order to reach a given security level L (representing the probability of successful impersonation or forgery, as specified in ISO/IEC 9798-5, for instance), the two schemes have to be repeated several times, as follows $N_1 = \left\lceil \log_{2/3} L \right\rceil$ and $N_2 = \left\lceil \log_{1/2} L \right\rceil$. Therefore, considering the fist factor (soundness error), the ratio between the two total payloads for reaching the security goal is given by

$$\frac{TPL_1}{TPL_2} = \frac{N_1 \times PL_1}{N_2 \times PL_2} = \log_{\frac{3}{2}} 2 \times \frac{PL_1}{PL_2}$$

As for the second factor represented by wider challenge values, we can have the combined effect of $\alpha \in \mathbb{Z}_q$ and $b \in \mathbb{F}_2$ to play the role of challenges. Provided that the overall soundness requirement is also satisfied (by having a minimum number of rounds executed), this avoids the necessity of executing one round per hash bit. Table 1 shows a numeric comparison between the two schemes. In order to construct this table, the following choices were made. We considered a security level close to 100 bits as constraint. For the hash function, we use the parameters from Table 2, page 90 of [5], which lists the state-of-art values. According to it, a hash length with length 224 bits will provide a level of security of 111, which is close to the value we chose. Regarding the choice of parameters for TRSS-C, we used the results listed in Section 7 of [6], and picked the code length as 2480, with which one can reach a security level of 107 bits.

The third point to consider is the application of ideal lattices in our scheme. This can speed up the most costly operations associated with multiplications between matrices and vectors, and have them executed in time $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$.

Scheme	Signature Size (Mbytes)	Number of Rounds	Hash Length (bits)
TRSS-C	47	190	224
TRSS-L	45	111	224

Table 1. Comparing TRSS Schemes for N=100, and security=111 bits

Bit-security	n	m	q	Commitment Length (bits)
111	64	2048	257	224

Table 2. Concrete Parameters

5.1 Parameters

Similarly as shown in [16], in order to guarantee with overwhelming probability that there are other solutions to $\mathbf{Ax} = \mathbf{0} \bmod q$, besides the private key possessed by each user (which is pivotal in the demonstration of security against concurrent attack), one can make q and m satisfy the relation below

$$q^n \ll card\{\mathbf{x} \in \mathbb{Z}_2^{m+1} : weight(\mathbf{x}) = m/2 + 1\}. \quad (1)$$

Besides, q has its value bounded from the following theorem by Micciancio, proved in [24].

Theorem 3. *For any polynomially bounded functions $\beta(n), m(n), q(n) = n^{O(1)}$, with $q(n) \geq 4\sqrt{m(n)}n^{1.5}\beta(n)$ and $\gamma(n) = 14\pi\sqrt{n}\beta(n)$, there is a probabilistic polynomial time reduction from solving $GapCVP_\gamma$ in the worst-case to solving $SIS_{q,m,\gamma}$ on the average with non-negligible probability. In particular, for any $m = \Theta(n \log n)$, there exists $q(n) = O(n^{2.5} \log n)$ and $\gamma = O(n\sqrt{\log n})$, such that solving $SIS_{q,m}$ on the average is at least as hard as solving $GapSVP_\gamma$ in the worst case.*

The parameters that we chose to use with our TRSS, shown in Table 2 are derived from those applied by the SWIFFT lattice-based hash proposed in [21]. The comparison exhibited in Table 1 is based in such choice. The soundness requirement alone makes TRSS-C run 190 rounds. Our scheme, on the other hand, which has lower soundness error, reaches the same goal with 111 rounds.

This section discussed about the efficiency gains that resulted from our design choices, such as the underlying identification scheme with smaller soundness error and the possibility of using ideal lattices. It is important to notice that such choices do not compromise security. In the next section we make an overall appreciation of our construction and present further lines of research associated with it.

6 Conclusions and Further Work

In this work we have shown standard applications of the Fiat-Shamir heuristics to lattice-based identification schemes in order to derive signature schemes with proven security. By means of such construction, we were able to adapt a threshold ring signature scheme from a code-based paradigm, obtaining a construction that is more efficient and has stronger security evidences. Instead of using the syndrome decoding hardness assumption as security basis, we used the SIS problem, with a suitable set of parameters. Such approach enables the application of reductions from worst-case GapSVP to average-case SIS, giving stronger security evidences for the resulting scheme.

In addition, we replaced the Stern's identification construction by the CLRS [11]. Such substitution has two positive effects on the efficiency of the threshold ring signature scheme. It reduces the soundness error from 2/3 to 1/2, allowing a specified security level to be reached with a fewer number of interactions. The reduced number of rounds implies in shorter signatures as well.

Our construction can also use ideal lattices. This results in more efficient multiplications of vectors by matrices by means of FFT. Such operations take time $\tilde{\mathcal{O}}(n)$.

As shown in [19], when compared to zero-knowledge constructions, such as Kawachi's [16] or CLRS [11], Lyubashevky's identification scheme provides better results in terms of size, if used in conjunction with Fiat-Shamir heuristics to derive a signature scheme. This is due to its extremely low soundness error. Therefore, a threshold ring signature scheme can make use of this fact to achieve shorter signatures than those shown in the present article. However, some structural changes are necessary in order to obtain the anonymity property possessed by our scheme. It requires a more involved approach than the direct application of TRSS-C construction [2].

References

1. Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology-EUROCRYPT 2002*, pages 418–433. Springer, 2002.
2. Carlos Aguilar Melchor, Pierre-Louis Cayrel, and Philippe Gaborit. A new efficient threshold ring signature scheme based on coding theory. In Buchmann and Ding [10], pages 1–16.

3. Miklós Ajtai. Generating hard instances of lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(7), 1996.
4. Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(65), 1996.
5. Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. *Post Quantum Cryptography*. Springer Publishing Company, Incorporated, 2008.
6. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. In Buchmann and Ding [10], pages 31–46.
7. Colin Boyd, editor. *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*. Springer, 2001.
8. Zvika Brakerski and Yael Tauman Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive, Report 2010/086, 2010. `http://eprint.iacr.org/`.
9. Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold ring signatures and applications to ad-hoc groups. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 465–480. Springer, 2002.
10. Johannes Buchmann and Jintai Ding, editors. *Post-Quantum Cryptography, Second International Workshop, PQCrypto 2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings*, volume 5299 of *Lecture Notes in Computer Science*. Springer, 2008.
11. Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. Improved zero-knowledge identification with lattices.
12. Pierre-Louis Cayrel and Pascal Véron. Improved code-based identification scheme. *CoRR*, abs/1001.3017, 2010. `http://arxiv.org/abs/1001.3017v1`.
13. David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.
14. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In Boyd [7], pages 157–174.
15. Léonard Dallot and Damien Vergnaud. Provably secure code-based threshold ring signatures. In Matthew G. Parker, editor, *IMA Int. Conf.*, volume 5921 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2009.
16. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security*, pages 372–389, Berlin, Heidelberg, 2008. Springer-Verlag.
17. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.
18. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2008.
19. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

20. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.
21. Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. Swifft: A modest proposal for fft hashing. In Kaisa Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pages 54–72. Springer, 2008.
22. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. In *Computational Complexity*. Springer, 2007.
23. Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.
24. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
25. Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003.
26. David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *EUROCRYPT*, pages 387–398, 1996.
27. Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Boyd [7], pages 552–565.
28. P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.
29. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, 1993.
30. Pascal Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.