VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/v10127-012-0038-4
Tatra Mt. Math. Publ. **53** (2012), 33–63

IMPROVED ZERO-KNOWLEDGE IDENTIFICATION WITH LATTICES

Pierre-Louis Cayrel — Richard Lindner — Markus Rückert –
– Rosemberg Silva

ABSTRACT. Zero-knowledge identification schemes solve the problem of authenticating one party to another via an insecure channel without disclosing any additional information that might be used by an impersonator. In this paper we propose a scheme whose security relies on the existence of a commitment scheme and on the hardness of worst-case lattice problems. We adapt a code--based identification scheme devised by Cayrel, Véron and El Yousfi, which constitutes an improvement of Stern's construction. Our solution sports analogous improvements over the lattice adaption of Stern's scheme which Kawachi et al. presented at ASIACRYPT '08. Specifically, due to a smaller cheating probability close to 1/2 and a similar communication cost, any desired level of security will be achieved in fewer rounds. Compared to Lyubashevsky's scheme presented at ASIACRYPT '09, our proposal, like Kawachi's, offers a much milder security assumption: namely, the hardness of SIS for trinary solutions. The same assumption was used for the SWIFFT hash function, which is secure for much smaller parameters than those proposed by Lyubashevsky.

1. Introduction

One of the main objectives in cryptography is to provide means of access control, and identification (ID) schemes are typically applied in order to reach this goal. These schemes describe interactive protocols between a designated prover and verifier with the purpose of demonstrating that the prover knows a secret that is associated with his identity. In zero-knowledge schemes, no information about this secret is revealed, except the fact that the prover knows it. Besides, using hard lattice problems as security basis allows for very mild assumptions

2010 Mathematics Subject Classification: 94A60, 03G10.
Keywords: lattice-based cryptography, identification scheme, hash function, SIS problem, zero-knowledge.
This research was supported by CASED `www.cased.de` and FAPESP `http://www.fapesp.br.`

in the sense that they are worst-case instead of average-case and provide resistance against quantum adversaries.

There is an efficient generic construction due to Fiat and Shamir that transforms any ID scheme into a signature scheme, in the random oracle model [14]. Therefore, having an efficient ID solution from lattices gives rise to a similarly efficient signature construction, keeping the same hardness assumption. One of the main hardness assumption for ID schemes based on lattices is the short integer solution (SIS) problem. One is given an average case instance $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $m = \Omega\big(n \log(n)\big)$, and a norm bound b. Then, the task is to find a non-zero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{v} \equiv \mathbf{0} \pmod{q}$ and $\|\mathbf{v}\|_\infty \leq b$. This is hard to accomplish as long as there is at least one single n-dimensional lattice, where solving the approximate shortest vector problem is hard for approximation factors $\gamma \geq b \cdot \tilde{O}(1)$. Hence, it is desirable to build an ID scheme based on SIS with the least possible norm bound b, which is $b = 1$.

The most relevant ID schemes based on number theoretic problems, e.g., [14] and [12], do not resist quantum attacks that use Shor's algorithm [33]. One of the first schemes to resist such kind of attack was proposed by Stern [34]. It relies on the syndrome decoding problem and uses of a 3-pass zero-knowledge proof of knowledge (ZK-PoK) with a soundness error of 2/3 and perfect completeness. Recently, Kawachi, Tanaka and Xagawa [19] were able to change the security assumption of Stern's scheme to SIS with norm bound 1. With their work, Kawachi et al. provide a more efficient alternative to Lyubashevsky's ID scheme [21], [24], which uses a stronger assumption, SIS with norm bound $O\big(n^2 \log(n)\big)$. In contrast to typical zero-knowledge schemes, Lyubashevsky's construction is based on a witness-indistinguishable (not zero-knowledge) proof of knowledge. Furthermore, it has no soundness error. However, it a completeness error of $1 - 1/e$, which leads to increased communication costs and the undesirable scenario of having an honest prover being rejected by the verifier.

In code-based cryptography, there is also the scheme proposed by Cayrel, Véron and El Yousfi [11] that improves Stern's scheme by reducing the soundness error to $q/\big(2(q-1)\big) \approx 1/2$. This improvement leads to lower the communication cost, when comparing both schemes for a given security level. Currently, in terms of efficiency, there is no practical lattice-based construction that is comparable to that put forward by Cayrel, Véron and El Yousfi.

We propose such a scheme with a soundness error of $(q+1)/2q \approx 1/2$ and perfect completeness[1]. It is based on the same efficient version of the SIS problem that is used by Kawachi et al. or by the SWIFFT compression function [25]. Both the small soundness error and the mild assumption make our scheme more efficient than previous lattice-based ones. Moreover, by transferring code-based

[1]We conjecture that Cayrel, Véron and El Yousfi's scheme has the same soundness error by the arguments given in Section 3.2.

constructions to lattices, we can exploit efficiency improvements using ideal lattices without losing provable security. As a result, our scheme has smaller public keys and more efficient operations than those associated with the current code-based ID schemes.

For a comparison with the most recent lattice-based ID schemes, see Table 1, which assumes that the parameters listed in Table 2 are used, and that a soundness error of 2^{-16} (one of the values recommended in the norm ISO/IEC 9798) is specified. We computed that Lyubashevky's scheme takes 11 rounds to reach a completeness error below 1 %, when it is using the most efficient parameters listed in [22]. This paper is a longer version of [9], where we first proposed

Table 1. Comparison of lattice-based identification schemes.

Scheme	Secret key [Kbyte]	Public key [Kbyte]	Rounds	Payload [Kbyte]	Domain
Lyubashevsky [24]	0,25	2,00	11	110,00	Lattices
Kawachi et al. [19]	0,25	0,06	27	58,67	Lattices
Section 3	0,25	0,06	17	37,50	Lattices
Stern [34], Gaborit [15]	0,50	0,06	27	20,03	Code
Véron [35]	0,44	0,50	27	18,95	Code
Cayrel et al. [11]	0,20	0,10	16	5,64	Code

our lattice-based identification scheme. In the present work, we also instantiate a threshold ring signature scheme as an example of the application of Fiat-Shamir heuristic to the underlying identification scheme. Our signature scheme was first described in [10].

The content of this paper is organized as follows. We present the concepts that are used in the construction of the identification scheme in Section 2, as well as the original schemes by Stern, Cayrel, Véron and El Yousfi, whose key aspects were combined in the current work. Later, we give a detailed description of the algorithms that comprise the new scheme, and discuss the decisions that were made from a performance and security point of view in Section 3. Then, we analyze potential attacks and show how they affect the choice of parameters in Section 4. We also present a signature scheme named TRSS, presented in [10], obtained through the application of Fiat-Shamir transform to our identification scheme in Section 5. In Section 6 we present our conclusions and indicate further lines of investigation.

2. Preliminaries

Notation: We write vectors and matrices in boldface, while one-dimensional variables such as integers and reals will be regular. All vectors are columnvectors unless otherwise stated. We use $||$ to signify that multiple inputs of a function are concatenated. For example, let $h\colon \{0,1\}^* \to \{0,1\}^m$ be a hash function, and $\mathbf{a}, \mathbf{b}$ be vectors, then we write $h(\mathbf{a}||\mathbf{b})$ to denote the evaluation of h on some implicit binary encoding of $\mathbf{a}$ concatenated with an implicit encoding of $\mathbf{b}$. For the scope of this work, the actual encoding used is assumed to be efficient, and generally not discussed since it has no relevance for the results.

Security Model: We apply in the current work a string commitment scheme in the trusted setup model, according to which a trusted party honestly sets up the system parameters for the sender and the receiver.

For security model, we use impersonation under concurrent attacks. This implies that we allow the adversary to play the role of a cheating verifier prior to impersonation, possibly interacting with many different prover clones concurrently. Such clones share the same secret key, but have independent coins and keep their own state. As stated in [5], security against this kind of attack implies security against impersonation under active attack.

In the security proofs along this text we use the concept of zero-knowledge interactive proof of knowledge system. In such context, an entity called prover P has as goal to convince a probabilistic polynomial-time (PPT) verifier V that a given string x belongs to a language L, without revealing any other information.

This kind of proof satisfies three properties:

- Completeness: any true theorem can be proven.
  That is, $\forall x \in L$ $\mathrm{Prob}\big[(P,V)\,[x] = \mathrm{YES}\big] \geq 1 - \mathrm{negligible}(k)$. Where, (P,V) denotes the protocol describing the interaction between prover and verifier, and negligible(k) is a negligible function on some security parameter κ.
- Soundness: no false theorem can be proven.
  That is, $\forall x \notin L\ \forall P'$ $\mathrm{Prob}\big[(P',V)\,[x] = \mathrm{YES}\big] \leq 1/2$, where P' denotes any entity playing the role of prover.
- Zero-Knowledge: anything one could learn by listening to P, one could also have simulated by oneself.
  That is, $\forall V'_{PPT}$ $\exists S_{PPT}$ $\forall x \in L$ $\mathrm{VIEW}_{P,V'}(x)$ close to $S(x)$. Where, VIEW represents the distribution of the transcript of the communication between prover and verifier, and $S(x)$ represents the distribution of the simulation of such interaction. Depending on the proximity of $\mathrm{VIEW}_{P,V'}(x)$ and $S(x)$, as defined in [17], one can have:
  - Perfect Zero-knowledge: if the distributions produced by the simulator and the proof protocol are exactly the same.

- Statistical Zero-knowledge: if the statistical difference between the distributions produced by the simulator and the proof protocol is a negligible function.
- Computational Zero-knowledge: if the distributions produced by the simulator and the proof protocol are indistinguishable to any efficient algorithm.

String Commitment Scheme. A string commitment scheme is a protocol between two parties: a sender and a receiver. Both parties agree on a deterministic commitment function COM from a suitable family. This can be be realized, for instance, with a trusted third party. The scheme runs in two phases named committing and revealing.

In the commitment phase, the sender commits to a string s by choosing a string ρ uniformly at random and computing $c \leftarrow \text{COM}(s, \rho)$ which he sends to the receiver. In the corresponding revealing phase, the sender reveals both the string s and his chosen randomness ρ to the receiver. Then the receiver checks if the equality $c = \text{COM}(s; \rho)$ holds.

For our main protocol, we will use a commitment scheme which is secure in the sense that it is both hiding and binding. Informally, we say the scheme is statistically hiding, if a computationally unbounded attacking receiver has no noticeable advantage when correctly assigning two commitments c, c' to their respective strings s, s'. We say the scheme is computationally binding, if an attacking sender running in polynomial-time cannot change the commitment c to another value which passes the check in the revealing phase. Refer to, e.g., [18] for a formal definition.

Fiat-Shamir Transform. This heuristic converts an identification into a signature scheme by eliminating the verifier from the protocol. The challenges are computed in such a way that the use of pre-determined values that could be used in forgery is ruled-out. This is accomplished by making the challenge be the outcome of a cryptographic hash function that takes as input the commitments computed by the prover (who now plays the role of signer) and the message itself. The random aspect of the challenge is assured by the expected behavior of the hash function. This implies that the security model used is the random oracle model.

Threshold Ring Signature Scheme (TRSS). 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.

Lattices. Lattices are regular pointsets in a finite real vector space. They are formally defined as discrete additive subgroups of $\mathbb{R}^m$. They are typically represented by a basis $\mathbf{B}$ comprised of $n \leq m$ linear independent vectors in $\mathbb{R}^m$. In this case the lattice is the set of all combinations of vectors in $\mathbf{B}$ with integral coefficients, i.e., $L = \mathbf{B}\mathbb{Z}^n$. In cryptography, we usually consider exclusively integral lattices, i.e., subgroups of $\mathbb{Z}^m$.

There are some lattice-based computational problems whose hardness can be used as security assumption when building cryptographic applications. We will give definitions of all the problems relevant for this article now. We will use an unspecified norm in these definition, but for the scope of our article this will always be the max-norm.

Definition 2.1 (SVP)**.** Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, the shortest vector problem (SVP) consists in finding a non-zero lattice vector $\mathbf{Bx}$ such that $\|\mathbf{Bx}\| \leq \|\mathbf{By}\|$ for any other $\mathbf{y} \in \mathbb{Z}^n \setminus \{0\}$.

Definition 2.2 (CVP)**.** Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{t} \in \mathbb{Z}^m$, the closest vector problem (CVP) is finding $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{Bx} - \mathbf{t}\|$ is minimal.

SVP and CVP admit formulations as approximation, as well as promise (GapSVP, GapCVP) problems. For these versions the hardness can be proved under suitable approximation factors such as constants as seen, for example, in [27].

Definition 2.3 (SIS)**.** Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a positive real number b, the short integer solution (SIS) problem consists in finding a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ that satisfies the equation $\mathbf{Ax} = \mathbf{0} \pmod{q}$ and that has length $\|\mathbf{x}\| \leq b$.

There are lattice-based cryptographic hash function families for which it can be shown that breaking a randomly chosen instance is at least as hard as finding solutions for worst-case instances of lattice problems. In [3] and [4], Ajtai first showed how to use computationally intractable worst-case lattice problems as building blocks for cryptosystems. The parameter sizes involved, however, are not small enough to enable practical implementations.

Using cyclic lattices, Micciancio showed that it is possible to represent a basis, and thus public keys, with space that grows quasilinearly in the lattice

dimension [26]. Together with Lyubashevsky, he improved this initial result, achieving compression functions that are both efficient and provably secure assuming the hardness of worst-case lattice problems for a special type of lattices, namely ideal lattices [23]. We will talk in more detail about ideal lattices later on.

A variety of hard problems associated with lattices has been used as security basis in a number of cryptographic schemes. For example, Lyubashevsky's identification scheme is secure under active attacks, assuming the hardness of approximating SVP in all lattices of dimension n to within a factor of $\tilde{O}(n^2)$. By weakening the security assumption, on the other hand, one can achieve parameters small enough to make a practical implementation feasible, as seen in the identification scheme proposed by Kawachi et al. in [19]. In this later work, the authors suggest to use approximate GapSVP or SVP within factors $\tilde{O}(n)$.

Ideal lattices

Lattices are additive groups. However, there is a particular class of lattices that are also closed under (properly defined) ring multiplications. They correspond to the ideals of some polynomial quotient ring and are defined below. In the definition, we implicitly identify polynomials with their vector of coefficients.

Definition 2.4 (Ideal lattices)**.** Let f be some monic polynomial of degree n. Then, L is an *ideal lattice* if it corresponds to an ideal I in the ring $\mathbb{Z}[x]/\langle f \rangle$.

The concept of ideal lattices is very general. So, often lattice classes resulting from specific choices of f have their own names. For example, $f(x) = x^n - 1$ corresponds to cyclic lattices, and $f(x) = x^n + 1$ to anticyclic lattices. We also have the class of cyclotomic lattices resulting from all cyclotomic polynomials f. The later class is the only one relevant for practical applications at the moment.

Whereas, for general lattices of full rank n and entries of bitsize q, one needs $n^2 \log(q)$ bits to represent a basis, for ideal lattices only $n \log(q)$ bits suffice. This property addresses one of the major drawbacks usually associated with lattice--based cryptosystems: the large key sizes. Another good characteristic of the subclass of cyclotomic lattices is that associated matrix/vector multiplications can be performed in time $O(n \log(n))$ using discrete FFTs.

Lyubashevsky and Micciancio found that it is possible to restrict both SIS and SVP to the class of ideal lattices and keep the worst-case to average-case connection (for a fixed polynomial f that is irreducible over the integers) discovered by Ajtai. The corresponding problems are denoted with the prefix "Ideal-". As is customary, we again identify polynomials with their vectors of coefficients.

Definition 2.5 (Ideal-SIS)**.** Let f be some monic polynomial of degree n, and R_f be the ring $\mathbb{Z}[x]/\langle f \rangle$. Given m elements $a_1, \ldots, a_m \in R_f/qR_f$, the *Ideal-SIS*

problem consists in finding $x_1, \ldots, x_m \in R_f$ such that $\sum_{i=1}^{m} a_i x_i = 0 \pmod q$ and $\|(x_1, \ldots, x_m)\| \leq b$.

Switching between the ideal and general lattice setting for schemes based on SIS happens by replacing the randomly chosen matrix $\mathbf{A}$ for the general SIS setting with

$$\begin{aligned}
\mathbf{A}_1 &= [a_1, a_1 x, \ldots, a_1 x^{n-1}], \\
\mathbf{A}_2 &= [a_2, a_2 x, \ldots, a_2 x^{n-1}], \\
&\vdots \\
\mathbf{A}_m &= [a_m, a_m x, \ldots, a_m x^{n-1}], \\
\mathbf{A}' &= [\mathbf{A}_1 | \mathbf{A}_2 | \cdots | \mathbf{A}_m],
\end{aligned}$$

where $a_1, \ldots, a_m \in R_f / q R_f$ is chosen uniformly at random.

Identification scheme

An identification scheme is a collection of algorithms (Setup, Key Generation, Prover, Verifier) meant to provide a proof of identity for a given part. The Setup algorithm takes as input a security parameter and generates structures (such as lattice or code basis) to be used by the other algorithms. The Key Generation algorithm takes as input the parameters generated by the Setup algorithm and derives key pairs (private, public) to be associated with a set of users. The Prover and Verifier algorithms correspond to a protocol that is executed by entities P and V, respectively, such that the first convinces the latter about its identity authenticity, by proving to have knowledge of a solution to a hard problem, which establishes the relation between the components of P's key pair (private, public).

Schnorr's identification scheme

This is a number-theoretic identification scheme which relies on the hardness of the discrete logarithm problem. The protocol realizing this scheme corresponds to a zero-knowledge proof of knowledge that the entity *prover* knows the solution to x to the equation $X = \alpha^x \bmod p$, where $x \in \{1, \ldots, q-1\}$, with p and q primes such that q divides $p - 1$. It makes use of the keys and parameters generated as below.

The Key Generation algorithms is defined as follows:

- Input: security parameter k.
- Choose parameter t as a function of k.
- Choose uniformly at random the primes q and p, such that q divides $p-1$.

- Choose $\alpha \in \mathbb{Z}_p$ with order q.
- Choose the private key x uniformly at random from $\{1, \ldots, q-1\}$.
- Make the public key $X = \alpha^x \bmod p$.

The Identification Scheme protocol is comprised of the following steps:

- The *prover* chooses y uniformly at random from $\mathbb{Z}_q$, computes $Y = \alpha^y$ and sends it to the *verifier*.
- The *verifier* chooses c uniformly at random from $\mathbb{Z}_{2^t}$ and sends it to the *prover*.
- The *prover* responds with $y + cx \bmod q$.
- The *verifier* accepts the *prover* if $\alpha^z \equiv YX^c$, and Y belongs to the group generated by α and $z \in \mathbb{Z}_q$. Otherwise, the *prover* is rejected.

The security of the identification scheme above is based on the hardness of computing discrete logarithms [32]. The protocol has a soundness error of 2^{-t}.

Stern's identification scheme

The first practical code-based identification scheme was proposed by S t e r n [34]. Its basic algorithm uses a hash function h, a pair of keys $(\mathbf{i}, \mathbf{s})$ related by $\mathbf{i} = \mathbf{H}^T\mathbf{s}$, where $\mathbf{H}$ is a public parity check matrix of a given code, $\mathbf{s}$ is a private binary vector of Hamming weight p, and $\mathbf{i}$ is its public syndrome. In a given round, $\mathbf{y}$ is chosen uniformly at random from the same space as $\mathbf{s}$, a permutation σ of the integers $\big\{1, \ldots, \dim(\mathbf{y})\big\}$ is similarly chosen, and the commitments are calculated by the prover as follows:

$$\begin{aligned} c_1 &= h\big(\sigma \| \mathbf{H}^T\mathbf{y}\big), \\ c_2 &= h\big(\sigma(\mathbf{y})\big), \\ c_3 &= h\big(\sigma(\mathbf{y} \oplus \mathbf{s})\big). \end{aligned}$$

Upon receipt of a challenge b chosen uniformly at random from $\{0, 1, 2\}$, the prover reveals the information that enables the verifier to check the correctness of the commitments as below:

$$\begin{aligned} b = 0\colon &\text{ Reveal } \mathbf{y} \text{ and } \sigma. \text{ Check } c_1 \text{ and } c_2. \\ b = 1\colon &\text{ Reveal } \mathbf{y} \oplus \mathbf{s} \text{ and } \sigma. \text{ Check } c_1 \text{ and } c_3. \\ b = 2\colon &\text{ Reveal } \sigma(\mathbf{y}) \text{ and } \sigma(\mathbf{s}). \text{ Check } c_2, c_3, \\ &\quad \text{and } \mathrm{wt}\big(\sigma(\mathbf{s})\big) = p. \end{aligned}$$

This scheme has a soundness error of $2/3$. In order to reach a confidence level L on the authenticity of the prover, it has to be repeated a number r of times, so that $1 - (2/3)^r \geq L$.

In the same work Stern also proposed a few variants of the basic scheme focusing on specific goals, such as: minimize computing load, minimize number of rounds, apply identity-based construction, and employ an analogy of modular knapsacks. For the minimization of number of rounds, he suggested the following solution:

(1) The private key $\mathbf{s}$ is replaced by the generators $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ of a simplex code.

(2) Only two commitments $c_1 = h\big(\sigma \| \mathbf{H}^T \mathbf{y}\big)$ and $c_2 = h\big(\sigma(\mathbf{y}) \| \sigma(\mathbf{s}_1) \| \ldots \| \sigma(\mathbf{s}_n)\big)$ are used.

(3) The prover computes $z = \sigma\big(\mathbf{y} \oplus \bigoplus_{j=1}^{m} b_j \mathbf{s}_j\big)$ using a binary vector $\{b_1, \ldots, b_m\}$ received from the verifier.

(4) Upon challenge 0, the prover reveals σ, and the verifier checks c_1.

(5) Upon challenge 1, the prover discloses $\big\{\sigma(\mathbf{s}_1), \ldots, \sigma(\mathbf{s}_m)\big\}$, and the verifier checks that c_2 is correct and that the code generated by $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ is simplex with the required weight.

This solution replaces the 3-pass approach by a 5-pass one, but it is not effective as far as communication costs are regarded. A more efficient solution is shown in the following paragraph. It also corresponds to the underlying approach for our lattice-based solution.

Cayrel and Véron's identification scheme

The identification scheme proposed by Stern [34] was based on the hardness of the syndrome decoding problem. An improvement over this scheme, using the dual construction, was proposed by Véron [35], achieving lower communication costs and better efficiency. Like the basic Stern construct, however, a dishonest prover can have success with probability up to 2/3 in any given round.

By modifying the way the commitments are calculated, incorporating a value chosen at random by the verifier, Cayrel and Véron [11] were able to bound the cheating probability within a given round close to 1/2, with similar communication costs. The approach followed will be outlined later for the case of our scheme in Algorithm 2, where the syndrome decoding problem is replaced by the shortest vector problem as hardness assumption. It involves a 5-pass solution, similar to Stern's construction. It avoids the heavy payload associated with transmitting the whole basis of a simplex code (or of a lattice), though.

Another scheme suggested by Gaborit requires smaller storage for public data [15]. Given that the schemes we have seen are dealing with codes, this usually implies that a generator matrix or a parity check matrix is needed to fully characterize them. The idea applied by Gaborit was to use double-circulant matrices for a compact representation.

In our work, we point out that a combination of these two approaches can be used in the lattice context, namely ideal lattices (which allow a very compact representation, as efficient as double-circulant matrices) for an identification scheme structure with soundness error of 1/2. With this, we manage to have the lowest communication costs and lowest public data storage needs.

3. Our identification scheme

Taking C a y r e l and V é r o n' s scheme [11] as basis and changing the main security assumption from the syndrome decoding problem (code-based) to the short integer solution problem (lattice-based), we obtain a new identification scheme. The transformation is non-trivial since low-weight codewords that are required in one setting are not necessarily short vectors as required in the other and vice versa.

We begin by describing the new identification scheme and then give arguments regarding all major properties such as completeness, soundness, and zero--knowledge as well as performance.

3.1. Description

The scheme consists of two main parts: a key generation algorithm (Figure 1) and an interactive identification protocol (Figure 2).

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 $(\mathrm{sk}, \mathrm{pk}) = \big(\mathbf{x}, (\mathbf{y}, \mathbf{A}, \textsc{Com})\big)$

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

The key generation algorithm receives as input a set of parameters (n, m, q), e.g., $(64, 2048, 257)$ (see Section 4.1 for a discussion on why this is a sensible choice). It chooses a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ uniformly at random and selects as private key a binary vector $\mathbf{x} \in \{0,1\}^m$ of Hamming weight $m/2$. The public key consists of an n-dimensional vector $\mathbf{y} = \mathbf{A}\mathbf{x} \bmod q$, the random matrix $\mathbf{A}$, and a commitment function Com. To instantiate the algorithm, we need to

select a family of statistically hiding and computationally binding commitment functions $\mathcal{F}$.

For the time being we recommend the commitment functions used by Kawachi et al. since they merely require a lattice-based collision resistant, regular hash function, in our case SWIFFT, which allows us to have a single security assumption. The commitment functions Com that we use are deterministic algorithms, which get as second input a nonce r that is assumed to be chosen uniformly at random from a set big enough to guarantee the hiding property of the commitment.

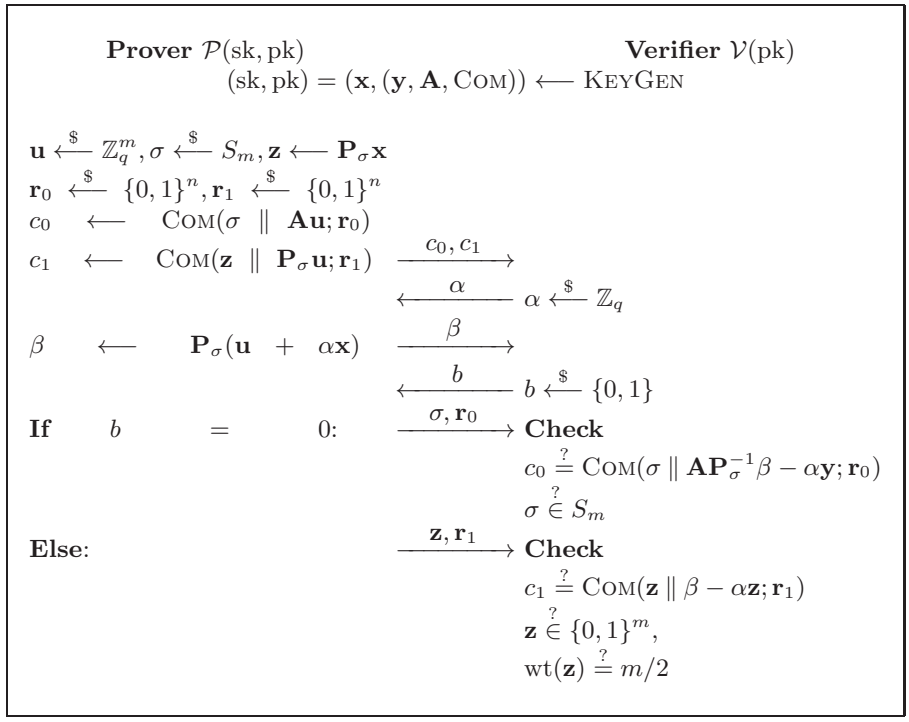

FIGURE 2. Identification protocol.

The identification protocol in Figure 2 describes the interaction between prover and verifier in order to convince the second party about the identity of the first. All computation in the protocol is performed modulo q, and we use the following notations. The set of all permutations on m elements is S_m. Any permutation $\sigma \in S_m$ is a linear operation and the associated $m \times m$ binary matrix is $\mathbf{P}_\sigma$.

The protocol is an adaption of the code-based identification scheme [11] which represents a major improvement to V é r o n' s [35] and S t e r n' s [34] schemes. In the same way our protocol represents an improvement over the lattice adaptions of Stern's scheme by K a w a c h i et al. [19]. Like Kawachi's, our adaptation to the lattice setting is non-trivial, since we need to ensure that a binary secret key is used (regardless of the Hamming weight). This needs to be guaranteed throughout the protocol which entails some change in the β that is used. Similarly to the coding-based scheme, a cheating prover, not knowing the secret key, can lead a verifier to believe that he actually knows that secret value with a probability up to $1/2$ in an individual round of execution. Therefore, in order to diminish the success rate of such an impersonation, the protocol has to be repeated a number of times, which is a function of the degree of confidence requested by the application that is using the scheme. This will be discussed further in Section 3.2, where we argue the soundness.

In the commitment phase, the prover commits to two values c_0, c_1, where c_0 is comprised of the random choices he made and c_1 contains information about his secret key. An adversary that can also correctly compute them with overwhelming probability either is able to break the commitment or to solve the hard problem that makes it possible to obtain a private key from its public counterpart. Those commitments are sent to the verifier, who responds in the second phase with value α taken uniformly at random from $\mathbb{Z}_q$. Upon receipt of this value, the prover is supposed to multiply it by the private key, add to a permuted masking value u (uniformly chosen at random from $\mathbb{Z}_q^m$) and make a permutation over the sum. Since $\mathbf{u}$ was random, β can be seen as a random variable with uniform distribution over $\mathbb{Z}_q^m$, leaking no information about the private key $\mathbf{x}$.

Upon receipt of this value, the verifier makes a challenge to the prover, picking a value uniformly at random from the set $\{0, 1\}$. The prover responds to it by revealing some piece of information that allows the verifier to compute and check the commitments. An honest prover will always be able to respond either challenges. Besides checking the correctness of the commitments, the verifier must also check that the values disclosed by the prover are well-formed, although in practice this would be solved by defining a suitable encoding for the data.

We will see in Section 3.3 how an impersonator can always cheat with a success probability of $1/2$, and that no better strategy is possible under our hardness assumptions. So in order to reach a prescribed level of security the interaction proposed here must be repeated an appropriate number of times.

Construction with ideal lattices. The present construction makes no assumptions about the structure of the SIS matrix $\mathbf{A}$. Therefore, the space necessary for storing this matrix is $\tilde{O}(n^2)$, which is too big for practical purposes.

Using ideal lattices, one can reduce such space requirements to $\tilde{O}(n)$ and simultaneously increase computation speed of matrix vector products in the form $\mathbf{Ax} \bmod q$ to $\tilde{O}(n)$ operations. This has been proposed and performed many times, perhaps most elegantly in the case of the SWIFFT compression function [25].

3.2. Security

In this section we show that the protocol in Figure 2 corresponds to a zero-knowledge interactive proof of knowledge of the predicate defined below. Let $I = \{\mathbf{A}, \mathbf{y}, m, q\}$ be public data shared by the parties P and V. Consider the predicate $\mathcal{P}(I, \mathbf{x})$ as "$\mathbf{x}$ is a binary vector of Hamming weight $m/2$ satisfying the equation $\mathbf{Ax} = \mathbf{y} \bmod q$".

We provide below proofs for the completeness, soundness and zero-knowledge properties of the identification scheme described in Figure 2. In particular, soundness holds even against concurrent attacks, i.e., an adversary may try to impersonate a given identity after having access to polynomially many verifier instances in parallel. Each of the verifier instances has the same secret key but is run with a different random tape. The challenge is to simulate the environment of the attacker during these interactions *and* still being able to extract "useful" information from the adversary during the impersonation phase. The required assumptions are that Com is a statistically hiding and computationally binding commitment scheme, e.g., based on SIS (cf. [19]), and the hardness of the SIS problem.

3.2.1. Completeness

Given that an honest prover has knowledge of the private key $\mathbf{x}$, the blending mask $\mathbf{u}$ and the permutations $\mathbf{P}_{\boldsymbol{\sigma}}$, he will always be able to derive the commitments c_0 and c_1, and reveal to the verifier the information necessary to verify that they are correct. He can also show that the private key in his possession has the appropriate Hamming weight. So, the verifier will always accept the honest prover's identity in any given round. This implies perfect completeness.

3.2.2. Zero-knowledge

We give a demonstration of the zero-knowledge property for the identification protocol shown in Figure 2. Here, we require the commitment function Com to be statistically hiding, i.e., $\text{Com}(x; r)$ is indistinguishable from the uniform distribution for an r chosen uniformly at random fro $\{0,1\}^n$. The argument x is a string of bits of arbitrary length.

Theorem 1. *Let q be prime. The described protocol is a statistically zero-knowledge proof of knowledge if the employed commitment scheme is statistically hiding.*

Proof. To prove the zero-knowledge property of our protocol, we construct a simulator S that outputs a protocol view $V = \big(c_0, c_1, \alpha, \beta, b, (\sigma, r_0), (\mathbf{z}, r_1)\big)$ without knowing the secret $\mathbf{x}$, such that V is indistinguishable from an the interaction of an honest prover with an honest verifier. It has access to a cheating verifier V^*, which contributes α and b. Therefore, S generates r_0, r_1 according to protocol and it gets $(\mathbf{A}, \mathbf{y}, \textsc{Com})$ as input. The simulator has to guess b before talking to V^*. For the moment, let us assume the guess is correct.

If $b = 0$, the simulator selects $\mathbf{u}$ and σ as per protocol and solves the equation $\mathbf{Ax} \equiv \mathbf{y} \pmod q$ for $\mathbf{x}$, which does not need to be short. With this pseudo secret key, the simulator computes c_0 and c_1 according to the protocol. The deviation in c_1 is not recognized because Com is statistically hiding. Then, S computes $\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha\mathbf{x})$ after obtaining α from $V^*(c_0, c_1)$. The result is uniform because $\mathbf{u}$ is chosen uniformly at random. As a result, S can reveal (σ, r_0), which passes the verification for $b = 0$.

If $b = 1$, the simulator needs to play against the second verification branch. It selects a binary $\mathbf{x}$ with Hamming weight $m/2$ and selects σ as per protocol. It computes c_0, c_1 and obtains $\alpha \longleftarrow V^*(c_0, c_1)$. Then, it computes $\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha\mathbf{x})$. As a result, S can reveal $\mathbf{P}_\sigma\mathbf{x}$ that passes verification.

In consequence, the simulator outputs a correct view with probability 1/2. Since the simulator has access to V^*, it can restart the verifier whenever the guess b was incorrect. The result is a statistically close simulation if Com is statistically hiding. □

3.2.3. Soundness

We now show that a dishonest prover is able to cheat a verifier to accept his identity with a probability limited by $(q+1)/2q \approx 1/2$. The number of possible queries sent by the verifier to a prover is given by all combinations of challenge bits $b \in \{0, 1\}$ and $\alpha \in \{0, \ldots, q-1\}$. Hence, there are $2q$ possible queries. Say, the dishonest prover wants to answer all challenges where $b = 0$, then he computes an alternate secret key $\mathbf{x}'$ with large entries such that $\mathbf{Ax}' = \mathbf{y}$. This is can be done with Gaussian elimination, for example. At the same time, when $\alpha = 0$ he can also answer in the case $b = 1$ by sending a random $\mathbf{z}$. Since $\alpha = 0$ this is not checked in the commitment.

Note that the $\alpha = 0$ query issue cannot be resolved by removing 0 from the set that α is drawn from, because the dishonest verifier can effectively shift the values of α by changing his protocol. Say he wants some fix α_0 to take the place

of 0 in the unmodified scheme, then he changes both the computations of the commitments and β to:

$$\begin{aligned} c_0 &\longleftarrow \textsc{Com}(\sigma \parallel \mathbf{Au} - \alpha_0\mathbf{y}; r_0), \\ \beta &\longleftarrow \mathbf{P}_\sigma(\mathbf{u} + (\alpha - \alpha_0)\mathbf{x}), \\ c_1 &\longleftarrow \textsc{Com}(\mathbf{z} \parallel \mathbf{P}_\sigma\mathbf{u} - \alpha_0\mathbf{z}; r_1). \end{aligned}$$

In effect, he can answer both challenges bits $b = 0, 1$ for $\alpha = \alpha_0$ now.

Thus, in total, the adversary can answer correctly for $q + 1$ out of $2q$ queries. In the proof, we show that if an adversary is able to answer more queries, it is also able to break one of the underlying assumptions, i.e., solve SIS or break the commitment.

Theorem 2. *If an honest verifier accepts a dishonest prover with probability $Pr \geq (q + 1)/2q + \epsilon(n)$, with $\epsilon(n)$ non-negligible, then there exists a polynomial time probabilistic machine M which breaks the binding property of the commitment* Com *or solves the SIS problem with non-negligible probability.*

P r o o f. On input $(n, m, q, \mathbf{A})$ (the SIS problem instance) and a challenge commitment function Com, we need to simulate the adversary's environment in two phases: a verification phase and an impersonation phase. In order to correctly prove knowledge of a valid secret key $\mathbf{x}$ during the verification phase, we choose $\mathbf{x}$ and $\mathbf{y}$ as in the key generation protocol and run the adversary $\mathcal{A}$ on public parameters (as per protocol).

Therefore, in the verification phase, we can perfectly simulate the prover. Since the protocol is statistically zero-knowledge, the adversary does not learn any information about $\mathbf{x}$ and the output distribution is the same as for all alternative secret keys $\mathbf{x}' \neq \mathbf{x}$.

After the first phase, we let $\mathcal{A}$ play the role of the cheating prover. First, we receive the commitments c_0, c_1. Then, because q is polynomial in n, we challenge the adversary with all $2q$ challenge pairs (α, b) and record successes as "1" and failures as "0" in a table with column labels "$b = 0$", "$b = 1$" and row labels "$\alpha = 0$", ..., "$\alpha = q-1$". This is done by rewinding the adversary appropriately.

For the moment, let us assume that there exist two rows, for α and α', such that both columns contain "1". Let $(\beta, \sigma, \mathbf{r}_0)$ and $(\beta', \sigma', \mathbf{r}'_0)$ be the outcomes for challenge $(\alpha, 0)$ and $(\alpha', 0)$, respectively. Furthermore, let $(\beta, \mathbf{z}, r_1)$ and $(\beta', \mathbf{z}', r'_1)$ be the outcomes for challenges $(\alpha, 1)$ respectively $(\alpha', 1)$.

Since the commitment Com is binding, we infer that $r_0 = r'_0$, $r_1 = r'_1$, and

$$\sigma \parallel \mathbf{A}P_\sigma^{-1}\beta - \alpha\mathbf{y} \quad = \quad \sigma' \parallel \mathbf{A}P_{\sigma'}^{-1}\beta' - \alpha'\mathbf{y}\,, \tag{1}$$

$$\mathbf{z} \parallel \beta - \alpha\mathbf{z} \quad = \quad \mathbf{z}' \parallel \beta' - \alpha'\mathbf{z}'\,. \tag{2}$$

Equation (1) implies $\sigma = \sigma'$. Similarly, (2) shows that the binary vectors $\mathbf{z}, \mathbf{z}'$ of weight $m/2$ are equal. Now, we turn to extracting $\mathcal{A}$'s secret key by rearranging

parts of (1) and (2), we get

$$\mathbf{A}P_{\sigma}^{-1}(\beta-\beta')(\alpha-\alpha')^{-1} \equiv \mathbf{y} \pmod{q}, \tag{3}$$

$$(\beta-\beta')(\alpha-\alpha')^{-1} \equiv \mathbf{z} \pmod{q}. \tag{4}$$

This proves that $\mathbf{x}' := P_{\sigma}^{-1}\mathbf{z}$ is a valid secret key and the reduction outputs the short lattice vector $\mathbf{v} = \mathbf{x} - \mathbf{x}'$. Notice that $\beta \neq \beta'$ because we have (1), $\alpha \neq \alpha'$, and $\sigma = \sigma'$. The extracted secret key is also different from the one of the simulator because the function $\mathbf{A}\mathbf{x} \bmod q$ compresses the set of valid secret keys and statistically hides them in the sense that the protocol is also witness indistinguishable. Hence, the adversary cannot learn the simulator's key but with probability $\leq 1/2 + n^{-\omega(1)}$.

What is left to show is that such a pair (α, α') exists. To see this, we apply a simple counting argument [30]. We know that $\mathcal{A}$ can answer correctly for $> q+1$ challenges. W.l.o.g., assume that it succeeds $\geq c$ times for $b = 0$ and $> q+1-c$ times for $b = 1$. Thus, there are $\geq c$ "1" entries in column "$b = 0$" and $> q+1-c$ "1" entries in column "$b = 1$".

Towards contradiction, assume that there is no such pair (α, α') for which $\mathcal{A}$ succeeds for the challenges $(\alpha, 0)$, $(\alpha, 1)$, $(\alpha', 0)$, and $(\alpha', 1)$. In other words, assume that the above extraction procedure breaks down. Then, there must be at least $c-1$ zeros in column "$b = 0$". In consequence, the total number of entries in the second column is $> c - 1 + q + 1 - c$. Since this is $> q$, we arrive at the desired contradiction and conclude that the knowledge extractor succeeds with non-negligible probability if $\epsilon(n)$ is non-negligible. □

Given that the scheme is a zero-knowledge proof of knowledge, it is also witness indistinguishable with respect to the secret $\mathbf{x}$. Fortunately, witness--indistinguishability is preserved under parallel composition. Thus, our scheme can be run many, i.e., $\omega\big(\log(n)\big)$, times in parallel to achieve a negligible soundness error but without increasing the number of rounds.

3.3. Security considerations

The code-based identification scheme proposed by Cayrel and Véron and that serves as starting point for this work has very good performance characteristics. Its security is based on the assumption that selecting a a random generator or parity check matrix will result in hard instances of the q-ary syndrome decoding problem, though. When adapting this scheme to use lattices, on the other hand, one achieves a construct based on the hardness of the SIS problem, and that has an worst-case/average-case reduction.

As pointed out in the description of the algorithms, ideal lattices can also be used in the scheme to improve performance and reduce the amount of public data. The precautions regarding the (a) irreducibility of the polynomial that characterizes the ring upon which the lattice is defined and (b) its expansion

factor must be observed, as recommended in [23]. This ensures that finding short vectors in such lattice is still hard to perform.

The present scheme is also secure against active attacks. Thus, an attacker is allowed to interact with a prover prior to attempting to impersonate him to a verifier. As consequence of the zero-knowledge property, however, no adversary that interacts with a real prover is able to obtain any knowledge that can be used later on to impersonate the prover.

We now prove that our scheme is secure against concurrent attacks, by showing that a public key corresponds to multiple secret keys and that the protocol is witness indistinguishable. It is a standard procedure, as seen in [13].

First, the existence of multiple secret keys associated with a given public key is assured by the parameter choice (see inequation 5). Second, given that our protocol is a zero-knowledge interactive proof, it is also witness indistinguishable [20].

4. Attacks

The most efficient way to attack this scheme, but probably the most difficult one, consists in solving the inhomogeneous short integer solution (ISIS) problem that is defined by the public key $\mathbf{y}$ and the public matrix $\mathbf{A}$, expressed as $\mathbf{Ax} = \mathbf{y} \bmod q$, where $\mathbf{x}$ is expected to be binary, with dimension m and Hamming weight $m/2$. This equation can be re-written as $\mathbf{A'x'} = 0 \bmod q$, with $\mathbf{A'} = [\mathbf{A}|\mathbf{y}]$ and $\mathbf{x'} = [\mathbf{x}| - 1]^T$, where T denotes the transpose of a matrix, and '|' just separates sub-matrices. Lattice basis calculation and reduction can then be applied in this second lattice to try to find a solution. The approximation factor, however, is $\tilde{O}(n)$, making the task hard.

4.1. Parameters

In order to guarantee with overwhelming probability that there are other solutions to $\mathbf{Ax} = \mathbf{y} \bmod q$, besides the private key possessed by the prover (which is pivotal in the demonstration of security against concurrent attacks), one can make q and m satisfy the relation below

$$q^n \ll card\{\mathbf{x} \in \mathbb{Z}_2^m : wt(\mathbf{x}) = m/2\}. \tag{5}$$

Besides, q is bounded by the following theorem, which M i c c i a n c i o and R e g e v proved in [28].

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.

Taking as reference the state-of-the-art lattice reduction algorithms studied in [16], the length of the shortest vector that can currently be found by the reduction algorithms is given by ($\delta \approx 1.011$):

$$length = \min\{q, q^{n/m}\delta^m\}. \tag{6}$$

We propose the set of parameters below, in Table 2, which are comparable to those used by the SWIFFT hash function. The best combinatorial attack for finding short lattice vectors [36] has a computational complexity above 2^{100} (generalized birthday attack, dividing in 16 groups at each turn). This means that our security level is 100 bits. In addition to that, the best lattice reduction algorithms return vectors with euclidean norm above 42, taking into account our set of parameters. Given that the private keys resulting from our parameters have euclidean norm 32, the choice made is safe. Besides, we can also see that the selected parameters satisfy both Theorem 3 and the restriction given by inequation 5.

TABLE 2. Concrete parameter.

Bit-security	n	m	q	Commitment Length (bits)
100	64	2048	257	224

5. Application of Fiat-Shamir transform

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 [9], 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 [19], [21] and [29]) 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 [22], 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.

5.1. 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 $\mathbf{H}\mathbf{x}^T = y$ (T stands for the operation of transposition), 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. A leader computes a signature on behalf of t out of N users. In order to do so, the leader is calculates the master commitments, where the equation relating public and private keys is trivially satisfied 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 $\mathbf{A}\mathbf{x} = \mathbf{y} \bmod q$, 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 $\mathbf{A}'\mathbf{x}' = \mathbf{0} \bmod q$. 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 Subsection 5.4, 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.

5.2. 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 P o i n t c h e v a l and S t e r n at [31] and A b d a l l a 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.

5.3. 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 Subsections 5.4 and 5.6, respectively.

The **Setup** algorithm, on input a security parameter κ, 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 κ, 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 $\mathbf{A}_i \in \mathbb{Z}_q^{n\times(m+1)}$. In order to generate a valid signature, t out of N users must be coordinated in this protocol. They are the active signers.

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.

Let us call H the hash value of the message being signed. In the application of the Fiat-Shamir heuristic, in the round i, we take the value of b as the ith bit of H. As α we take the first $\lceil \log \alpha \rceil$ bits of the hash value obtained from $h(C_0 \parallel C_1 \parallel H_{L+i})$, where $L = \text{length}(H)/2$.

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.

5.4. 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.

5.5. Honest-Verifier Zero-Knowledge Proof of Knowledge

We now prove that the Algorithm 1 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 $i \in \{0, \ldots, N-1\}$. 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 C a y r e l et al. in [9] in the active attack model. We will use this fact and discuss only the security of the composition described in Algorithm 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 $\mathbf{r}_{1,i} \xleftarrow{\$} \{0,1\}^n$
$c_{0,i} \longleftarrow \text{COM}(\sigma_i \parallel \mathbf{A}_i\mathbf{u}_i, \mathbf{r}_{0,i})$ and $c_{1,i} \longleftarrow \text{COM}(\sigma_i(\mathbf{u}_i) \parallel \sigma_i(\mathbf{x}_i), \mathbf{r}_{1,i})$
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* $C_0 = \text{COM}(\Sigma \parallel c_{0,1} \parallel \ldots \parallel c_{0,N}, \mathbf{r}_0)$ and $C_1 = \text{COM}\big(\Sigma(c_{1,1}, \ldots, c_{0,N}), \mathbf{r}_1\big)$ 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 $\Pi(\mathbf{x}) = \Sigma\big(\sigma_1(\mathbf{x}_1), \ldots, \sigma_{N-1}(\mathbf{x}_{N-1})\big)$
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 $C_0 \stackrel{?}{=} \text{COM}(\Sigma \parallel \mathbf{A}\Pi^{-1}(\boldsymbol{\beta}) \parallel \mathbf{r}_0)$ and that Π is well formed.
else
V checks that $C_1 \stackrel{?}{=} \text{COM}\big(\boldsymbol{\beta} - \alpha\Pi(\mathbf{x}) \parallel \Pi^{-1}(\boldsymbol{\beta}) \parallel \mathbf{r}_1\big)$ and that $\Pi(\mathbf{x})$ has Hamming weight $t(m/2 + 1)$.
end if
end procedure

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 (the universe of all N users) 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 5.1. *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 $\mathrm{wt}(\mathbf{v_i}) = m/2+1$, and this latter problem is hard under the SIS hardness assumption. □

Theorem 4. *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 $\mathrm{wt}(\mathbf{x}_i) = m/2+1$, when the user is signing the message, or $\mathrm{wt}(\mathbf{x}_i) = 0$ 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 5.1), 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 [9].

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 4 implies that the TRSS 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 5. *Our lattice-based threshold ring signature scheme is unconditionally source hiding.*

P r o o f. 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.

5.6. Performance

The previous section gave evidences and proofs that our system is safe. We now show that 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 $N_1 = \lceil \log_{2/3} L \rceil$ and $N_2 = \lceil \log_{1/2} L \rceil$.

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 4 shows a numeric comparison between the two schemes. In order to construct this table, the following choices were made. We considered

TABLE 3. Concrete parameters for TRSS.

Bit-security	n	m	q	Commitment length (bits)
100	64	2048	257	224

TABLE 4. Comparing TRSS schemes for N=100, and security=100 bits.

Scheme	Signature size (Mbytes)	Number of rounds	Hash length (bits)
TRSS-C	47	190	224
TRSS-L	45	111	224

a security level close to 100 bits as constraint. For the hash function, we use the parameters from Table 2, page 90 of [6], 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 [7], 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 $\mathcal{O}(n^2)$.

5.7. Parameters

Similarly as shown in [19], 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

$$q^n \ll card\{\mathbf{x} \in \mathbb{Z}_2^{m+1} : weight(\mathbf{x}) = m/2 + 1\}. \tag{7}$$

Besides, q has its value bounded from Theorem 3.

The parameters that we chose to use with our TRSS, shown in Table 3 are derived from those applied by the SWIFFT lattice-based hash proposed in [25]. The comparison exhibited in Table 4 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. Conclusion and further work

In this work we derived a lattice-based identification scheme from a code-based one. By shifting from one domain area to the other, we were able to provide stronger security evidences, given that the security arguments are now based on worst-case hardness instead of average-case. By using ideal lattices and suitable approximation factors, we were also able to obtain parameters that allow practical implementations for reasonable levels of security. We have also shown that it has better performance than all other lattice-based identification schemes. We have also presented a threshold ring signature scheme by generalizing a lattice-based identification scheme and applying the Fiat-Shamir transform to it.

A natural extension of the approach followed in the present work consists in adapting the structure of other cryptographic schemes and changing the hard problem upon which their security relies. By shifting between code and lattice domains and assessing which kind of gains such change provides, stronger security properties or more efficient implementations can be obtained.

Another extension consists in deriving other kinds of signature schemes from the current work. As we pointed out in Section 5, the present identification scheme has some characteristics that can result in efficient signature constructs, when its parameters are conveniently selected. In this context, it may be worthwhile to construct a "dual" ID scheme in the sense that it has a completeness error of $1/2$ and no soundness error as using the Fiat-Shamir transform on this "dual" scheme would result in very short signatures.

Acknowledgements. We are grateful to an anonymous referee for helpful comments.

REFERENCES

[1] ABDALLA, M.—AN, J. H.—BELLARE, M.—NAMPREMPRE, CH.: *From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security*, in: Advances in Cryptology—EUROCRYPT '02, 21st Internat. Conf. on the Theory and Appl. of Cryptographic Techniques (L. Knudsen, ed.), Amsterdam, 2002, Lecture Notes in Comput. Sci., Vol. 2332, Springer, Berlin, 2002, pp. 418–433.

[2] MELCHOR, C. A.—CAYREL, P.-L.—GABORIT, P.: *A new efficient threshold ring signature scheme based on coding theory*, in: Post-Quantum Cryptography, 2nd Internat. Workshop—PQCrypto '08, Cincinnati, OH, USA, 2008 (J. Buchmann et al., eds.), Lecture Notes in Comput. Sci., Vol. 5299, Springer, Berlin, 2008, pp. 1–16.

[3] AJTAI, M.: *Generating hard instances of lattice problems*, Electronic Colloquium on Computational Complexity (ECCC) **3** (1996).

[4] AJTAI, M.—DWORK, C.: *A public-key cryptosystem with worst-case/average-case equivalence*, Electronic Colloquium on Computational Complexity (ECCC) **3** (1996).

[5] BELLARE, M.—PALACIO, A.: *GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks*, in: Advances in Cryptology—CRYPTO '02, 22nd Annual Internat. Cryptology Conf., Santa Barbara, CA, USA, 2002 (M. Yung, ed.), Lecture Notes in Comput. Sci., Vol. 2442, Springer, Berlin, 2002, pp. 162–177.

[6] BERNSTEIN, D. J.—BUCHMANN, J.—DAHMEN, E.: *Post Quantum Cryptography*. Springer Publishing Company, Incorporated, 2008.

[7] BERNSTEIN, D. J.—LANGE, T.—PETERS, CH.: *Attacking and defending the McEliece cryptosystem*, in: Post-Quantum Cryptography—PQCrypto '08 (J. Buchmann and J. Ding, eds.), Lecture Notes in Comput. Sci., Vol. 5299, Springer, Berlin, 2008, pp. 31–46.

[8] BUCHMANN, J.—DING, J., EDS.: *Post-Quantum Cryptography,* in: 2nd Internat. Workshop—PQCrypto '08, Cincinnati, OH, USA, 2008, Lecture Notes in Comput. Sci., Vol. 5299, Springer, Berlin, 2008.

[9] CAYREL, P.-L.—LINDNER, R.—RÜCKERT, M.—SILVA, R.: *Improved zero-knowledge identification with lattices*, in: Provable Security, 4th Internat. Conf.—ProvSec '10, Malacca, Malaysia, 2010 (S. H. Heng et al., eds.), Lecture Notes in Comput. Sci., Vol. 6402, Springer, Berlin, 2010, pp. 1–17.

[10] CAYREL, P.-L.—LINDNER, R.—RÜCKERT, M.—SILVA, R.: *A lattice-based threshold ring signature scheme*, in: Progress in Cryptology—LATINCRYPT '10, 1st Internat. Conf. on Cryptology and Information Security (M. Abdalla et al., eds.) Puebla, Mexico, 2010, Lecture Notes in Comput. Sci., Vol. 6212, Springer, Berlin, 2010, pp. 255–272.

[11] CAYREL, P.-L.—VÉRON, P.—SILVA, R.: *Improved code-based identification scheme*, in: Provable Security, 4th Internat. Conf.—ProvSec '10, Malacca, Malaysia, 2010 (S.-H. Heng, et al., eds.), Lecture Notes in Comput. Sci., Vol. 6402, Springer, Berlin, 2010, pp. 1–17.

[12] FEIGE, U.—FIAT, A.—SHAMIR, A.: *Zero Knowledge Proofs of Identity*, in: Proc. of the 19th Annual ACM Symposium on Theory of Computing—STOC '87, (A. V. Aho), New York, USA, ACM, New York, 1987, pp. 210–217.

[13] FEIGE, U.—FIAT, A.—SHAMIR, A.: *Witness indistinguishable and witness hiding protocols*, in: Proc. of the 22nd Annual ACM Symposium on Theory of Computing–STOC '90, ACM, New York, 1990, pp. 416–426.

[14] FIAT, A.—SHAMIR, A.: *How to prove yourself: practical solutions to identification and signature problems*, in: Advances in Cryptology—CRYPTO '86 (A. M. Odlyzko, ed.), Santa Barbara, Calif., 1986, Lecture Notes in Comput. Sci., Vol. 263, Springer, Berlin, 1986, pp. 186–194.

[15] GABORIT, P.—GIRAULT, M.: *Lightweight code-based identification and signature*, in: IEEE Transactions on Information Theory—ISIT '07, Nice, France, 2007, IEEE, pp. 186–194.

[16] GAMA, N.—NGUYEN, P. Q.: *Predicting lattice reduction*, in: Advances in Cryptology––EUROCRYPT '08, 27th Annual Internat. Conf. on the Theory and Appl. of Cryptographic Techniques (N. Smart, ed.), Istanbul, Turkey, 2008, Lecture Notes in Comput. Sci., Vol. 4965, Springer, Berlin, pp. 31–51.

[17] GOLDWASSER, S.—MICALI, S.—RACKOFF, C.: *The knowledge complexity of interactive proof-systems*, in: Proc. of the 17th Annual ACM Symposium on Theory of Computing, ACM, New York, 1985, pp. 291–304.

[18] HALEVI, S.—MICALI, S.: *Practical and provably-secure commitment schemes from collision-free hashing*, in: Advances in Cryptology—CRYPTO '96 (N. Koblitz, ed.), Santa Barbara, California, 1996 Lecture Notes in Comput. Sci., Vol. 1109, Springer, Berlin, pp. 201–215.

[19] KAWACHI, A.—TANAKA, K.—XAGAWA, K.: *Concurrently Secure identification Schemes based on the worst-case hardness of lattice problems*, in: Advances in Cryptology—ASIACRYPT '08, 14th Internat. Conf. on the Theory and Appl. of Cryptology and Information Security (J. Pieprzyk, ed.), Melbourne, Australia, 2008. Lecture Notes in Comput. Sci., Vol. 5350, Springer, Berlin, 2008, pp. 372–389.

[20] KILIAN, J.—PETRANK, E.: *Concurrent and resettable zero-knowledge in poly-loalgorithm rounds*, in: Proc. of the 33rd Annual ACM Symposium on Theory of Computing (J. S. Vitter et al., eds.), Hersonissos, Greece, 2001, ACM, New York, NY, USA, pp. 560–569.

[21] LYUBASHEVSKY, V.: *Lattice-based identification schemes secure under active attacks*, in: Public key cryptography—PKC '08, 11th Internat. Workshop on Practice and Theory in Public-Key Cryptography (R. Cramer, ed.), Barcelona, Spain, 2008, Lecture Notes in Comput. Sci., Vol. 4939, Springer, Berlin, 2008, pp. 162–179.

[22] LYUBASHEVSKY, V.: *Fiat-Shamir with aborts: applications to lattice and factoring-based signatures*, in: Advances in Cryptology—ASIACRYPT '09, 15th Internat. Conf. on the Theory and Application of Cryptology and Information Security (M. Matsui, ed.), Tokyo, Japan, 2009, Lecture Notes in Comput. Sci., Vol. 5912, Springer, Berlin, 2009, pp. 598–616.

[23] LYUBASHEVSKY, V.—MICCIANCIO, D.: *Generalized compact knapsacks are collision resistant*, in: Automata, Languages and Programming, 33rd Internat. Colloquium––ICALP '06 (M. Bugliesi et al. ed.), Venice, Italy, 2006, Lecture Notes in Comput. Sci., Vol. 4052, Springer, Berlin, 2006, pp. 144–155.

[24] LYUBASHEVSKY, V.—MICCIANCIO, D.: *Asymptotically efficient lattice-based digital signatures*, in: Theory of Cryptography Conference—TCC '08 (R. Canetti, ed.), New York, USA, 2008, Lecture Notes in Comput. Sci., Vol. 4948, Springer, Berlin, 2008, pp. 37–54.

[25] LYUBASHEVSKY, V.—MICCIANCIO, D.—PEIKERT, CH.—ROSEN, A.: *SWIFFT: A modest proposal for FFT hashing*, in: Fast Software Encryption, 15th Internat. Workshop—FSE '08 (K. Nyberg, ed.), Lausanne, Switzerland, 2008, Lecture Notes in Comput. Sci., Vol. 5086, Springer, Berlin, 2008, pp. 54–72.

[26] MICCIANCIO, D.: *Generalized compact knapsacks, cyclic lattices, and efficient one-way functions*, Comput. Complexity **16** (2007), 365–411.

[27] MICCIANCIO, D.—GOLDWASSER, SH.: *Complexity of Lattice Problems: A Cryptographic Perspective*, in: Kluwer Academic Publishers, The Kluwer International Series in Engineering and Computer Science, Vol. 671, Kluwer Academic Publishers, Boston, 2002.

[28] MICCIANCIO, D.—REGEV, O.: *Worst-case to average-case reductions based on Gaussian measures*, SIAM J. Comput. **37** (2007), 267–302.

[29] Micciancio, D.—Vadhan, S. P. *Statistical zero-knowledge proofs with efficient povers: lattice problems and more*, in: Proc. of the 23rd Internat. Conf. on Cryptology—CRYPTO '03 (D. Boneh, ed.), Santa Barbara, 2003, Lecture Notes in Comput. Sci., Vol. 2729, Springer, Berlin, 2003, pp. 282–298.

[30] OHTA, K.—OKAMOTO, T.: *On concrete security treatment of signatures derived from identification*, in: Advances in Cryptology—CRYPTO '98, 18th Annual Internat. Cryptology Conf. (H. Krawczyk, ed.), Santa Barbara, CA, USA, 1998, Lecture Notes in Comput. Sci., Vol. 1462, Springer, Berlin, pp. 354–369.

[31] POINTCHEVAL, D.—STERN, J.: *Security proofs for signature schemes*, in: Proc. of the 15th Annual Internat. Conf. on Theory and Appl. of Cryptographic Techniques––EUROCRYPT '96 (U. Maurer, ed.), Zaragoza, Spain, 1996, Lecture Notes in Comput. Sci., Vol. 1070, Springer, Berlin, pp. 387–398.

[32] SCHNORR, C. P.: *Efficient identification and signatures for smart cards*, in: Advances in Cryptology—CRYPTO '89, Santa Barbara, CA, 1989, Lecture Notes in Comput. Sci., Vol. 435, Springer, Berlin, 1990, pp. 239–252.

[33] SHOR, P. W.: *Polynominal time algorithms for discrete logarithms and factoring on a quantum computer*, in: Algorithmic Number Theory, 1st Internat. Symposium–ANTS-I (L. M. Adleman and M.-D. A. Huang, eds.), Ithaca, NY, USA, 1994, Lecture Notes in Comput. Sci., Vol. 877, Springer, Berlin, 1994, p. 289.

[34] STERN, J.: *A new identification scheme based on syndrome decoding*, in: Advances in Cryptology—CRYPTO '93, 13th Annual Internat. Cryptology Conf. (D. R. Stinson, ed.), Santa Barbara, CA, USA, 1993, Lecture Notes in Comput. Sci., Vol. 773, Springer, Berlin, 1994, pp. 13–21.

[35] VÉRON, P.: *Improved identification schemes based on error-correcting codes*, Appl. Algebra Engrg. Comm. Comput. **8** (1996), 57–69.

[36] WAGNER, D.: *A generalized birthday problem*, in: Advances in Cryptology–CRYPTO '02, 22nd Annual Internat. Cryptology Conf. (M. Yung, ed.), Santa Barbara, CA, USA, 2002, Lecture Notes in Comput. Sci., Vol. 2442, Springer, Berlin, 2002, pp. 288–303.

[37] YUNG, M., ED.: *Advances in Cryptology*, in: Proc of the 22nd Annual Internat. Cryptology Conference, Santa Barbara, California, USA, 2002, Lecture Notes in Comput. Sci., Vol. 2442, Springer, Berlin, 2002.

Received October 18, 2012

Pierre-Louis Cayrel
Laboratoire Hubert Curien UMR CNRS 5516
de Saint-Etienne
Bâtiment F 18 rue du professeur Benoît Lauras
42000 Saint-Etienne
FRANCE
E-mail: pierre.louis.cayrel@univ-st-etienne.fr

Richard Lindner
Markus Rückert
Technische Universität Darmstadt
Department of Computer Science
Hochschulstraße 10
64289 Darmstadt
GERMANY
E-mail: rlindner@cdc.informatik.tu-darmstadt.de
rueckert@cdc.informatik.tu-darmstadt.de

Rosemberg Silva
Institute of Computing
University of Campinas
Av. Albert Einstein 1251, Cidade Universitria
13083-852 Campinas/SP
BRAZIL
E-mail: rasilva@ic.unicamp.br