Improved Zero-knowledge Identification with Lattices

Extended Abstract — July 1, 2010

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. 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 and Véron, 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 2008. 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 2009, 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.

Keywords. Lattice-based cryptography, identification scheme, hash function, SIS problem, zero-knowledge.

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 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 [7]. Therefore, having an efficient ID solution

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

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 $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $m = \Omega(n \log(n))$, 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., [7] and [5], do not resist quantum attacks that use Shor's algorithm [22]. One of the first schemes to resist such kind of attack was proposed by Stern [23]. 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 [11] 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 [13,16], which uses a stronger assumption, SIS with norm bound $O(n^2 \log(n))$. 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 and Véron [4] that improves the Stern's scheme by reducing the soundness error to $q/(2(q-1)) \approx 1/2$. 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 and Véron.

We propose such a scheme with a soundness error of $(q+1)/2q \approx 1/2$ and perfect completeness[4]. 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 [17]. Both the small soundness error and the mild assumption make our scheme more efficient than previous lattice-based ones. Moreover, by transferring code-based 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 [14].

Scheme	Secret key [Kbyte]	Public key [Kbyte]	Rounds	Total communication [Kbyte]	SIS norm bound
Lyubashevsky [16]	0,25	2,00	11	110,00	$\tilde{O}(n^2)$
Kawachi et al. [11]	0,25	0,06	27	58,67	1
Section 3	0,25	0,06	17	37,50	1

Table 1. Comparison of lattice-based identification schemes.

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 and Véron, whose key aspects were combined in the current work. Later, we give a detailed

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

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.

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 [3], 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, $\forall x \in L$ Prob $[(P,V)[x] = \text{YES}] \geq 1 - \text{negligible}(k)$. Where, (P,V) denotes the protocol describing the interaction between prover and verifier, and negligible(k) is a negligible function on some security parameter k.
- Soundness: no false theorem can be proven. That is, $\forall x \notin L \, \forall P'$ Prob $[(P',V)[x] = \text{YES}] \leq 1/2$
- Zero-Knowledge: anything one could learn by listening to P, one could also have simulated by oneself. That is, $\forall V'_{PPT} \, \exists S_{PPT} \, \forall x \in L$ $\text{VIEW}_{P,V'}(x)$ 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 $\text{VIEW}_{P,V'}(x)$ and $S(x)$, as defined in [10], 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.

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 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 $\mathbf{y} \in \mathbb{Z}^n \setminus \{0\}$.*

SVP admits formulations as approximation, as well as promise (or gap) problems. For these versions, the hardness can be proved under suitable approximation factors, such as constants, as seen for example in [19].

Definition 2 (SIS). *Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, 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 [1] and [2], 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 [18]. 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 [15]. 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 [11]. In this later work, the authors suggest to use approximate Gap-SVP 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 3 (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 4 (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 $0 < \|(x_1, \ldots, x_m)\| \leq b$.*

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

$$\mathbf{A}' = [a_1, a_1 x, \ldots, a_1 x^{n-1} | a_2, a_2 x, \ldots, a_2 x^{n-1} | \cdots | a_m, a_m x, \ldots, a_m x^{n-1}],$$

where $a_1, \ldots, a_m \in R_f / qR_f$ 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 A and B, 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 A's key pair (private, public).

Stern's Identification Scheme. The first practical code-based identification scheme was proposed by Stern [23]. 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 $\{1, \ldots, \dim(\mathbf{y})\}$ is similarly chosen, and the commitments are calculated by the prover as follows

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

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:

$b = 0$: Reveal $\mathbf{y}$ and σ. Check c_1 and c_2.
$b = 1$: Reveal $\mathbf{y} \oplus \mathbf{s}$ and σ. Check c_1 and c_3.
$b = 2$: Reveal $\sigma(\mathbf{y})$ and $\sigma(\mathbf{s})$. Check c_2, c_3, and $\mathrm{wt}(\sigma(\mathbf{s})) = p$

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 $c_1 = h(\sigma \| \mathbf{H}^T \mathbf{y})$ and $c_2 = h(\sigma(\mathbf{y}) \| \sigma(\mathbf{s}_1) \| \ldots \| \sigma(\mathbf{s}_n))$ are used.
3. The prover computes $z = \sigma(\mathbf{y} \oplus \bigoplus_{j=1}^{m} b_j \mathbf{s}_j)$ 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 $\{\sigma(\mathbf{s}_1), \ldots, \sigma(\mathbf{s}_m)\}$, 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 [23] was based on the hardness of the syndrome decoding problem. An improvement over this scheme, using the dual construction, was proposed by Véron [24], 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 [4] 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 [8]. 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 Identification Scheme

Taking Cayrel and Véron's scheme [4] 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:
  $\mathbf{x} \overset{\$}{\longleftarrow} \{0,1\}^m$, s.t. $\mathrm{wt}(\mathbf{x}) = m/2$
  $\mathbf{A} \overset{\$}{\longleftarrow} \mathbb{Z}_q^{n \times m}$
  $\mathbf{y} \longleftarrow \mathbf{A}\mathbf{x} \bmod q$
  $\textsc{Com} \overset{\$}{\longleftarrow} \mathcal{F}$, 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.

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 $\textsc{Com}$. 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 $\textsc{Com}$ 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.

$$
\begin{array}{lcl}
\textbf{Prover } \mathcal{P}(\mathrm{sk}, \mathrm{pk}) & & \textbf{Verifier } \mathcal{V}(\mathrm{pk}) \\
\multicolumn{3}{c}{(\mathrm{sk}, \mathrm{pk}) = (\mathbf{x}, (\mathbf{y}, \mathbf{A}, \textsc{Com})) \longleftarrow \textsc{KeyGen}} \\
 & & \\
\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^m, \sigma \xleftarrow{\$} S_m, \mathbf{z} \longleftarrow \mathbf{P}_\sigma \mathbf{x} & & \\
\mathbf{r}_0 \xleftarrow{\$} \{0,1\}^n, \mathbf{r}_1 \xleftarrow{\$} \{0,1\}^n & & \\
c_0 \longleftarrow \textsc{Com}(\sigma \parallel \mathbf{A}\mathbf{u}; \mathbf{r}_0) & & \\
c_1 \longleftarrow \textsc{Com}(\mathbf{z} \parallel \mathbf{P}_\sigma \mathbf{u}; \mathbf{r}_1) & \xrightarrow{\quad c_0, c_1 \quad} & \\
 & \xleftarrow{\quad \alpha \quad} & \alpha \xleftarrow{\$} \mathbb{Z}_q \\
\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha \mathbf{x}) & \xrightarrow{\quad \beta \quad} & \\
 & \xleftarrow{\text{Challenge } b} & b \xleftarrow{\$} \{0,1\} \\
\textbf{If } b = 0\text{:} & \xrightarrow{\quad \sigma, \mathbf{r}_0 \quad} & \textbf{Check } c_0 \overset{?}{=} \textsc{Com}(\sigma \parallel \mathbf{A}\mathbf{P}_\sigma^{-1}\beta - \alpha\mathbf{y}; \mathbf{r}_0) \\
 & & \qquad \sigma \overset{?}{\in} S_m \\
\textbf{Else}\text{:} & \xrightarrow{\quad \mathbf{z}, \mathbf{r}_1 \quad} & \textbf{Check } c_1 \overset{?}{=} \textsc{Com}(\mathbf{z} \parallel \beta - \alpha\mathbf{z}; \mathbf{r}_1) \\
 & & \qquad \mathbf{z} \overset{?}{\in} \{0,1\}^m, \mathrm{wt}(\mathbf{z}) \overset{?}{=} m/2
\end{array}
$$

Fig. 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 [4] which represents a major improvement to Véron's [24] and Stern's [23] schemes. In the same way our protocol represents an improvement over the lattice adaptions of Stern's scheme by Kawachi *et al.* [11]. 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 the 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 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 challenge. 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.

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 $\mathbf{Ax}$ to $\tilde{O}(n)$ operations. This has been proposed and performed many times, perhaps most elegantly in the case of the SWIFFT compression function [17].

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 A and B. Consider the predicate $P(I, \mathbf{x})$ 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. [11]), and the hardness of the SIS problem.

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.

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., $\textsc{Com}(x; r)$ is indistinguishable from uniform for a uniform $r \in \{0, 1\}^n$.

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 output a protocol view $V = (c_0, c_1, \alpha, \beta, b, (\sigma, r_0), (\mathbf{z}, r_1))$ 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_1, r_2 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 $\textsc{Com}$ is statistically hiding. Then, S computes $\beta \longleftarrow \mathbf{P}_\sigma(\mathbf{u} + \alpha\mathbf{x})$ after obtaining α from $V^*(c_1, c_2)$. 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_1, c_2 and obtains $\alpha \longleftarrow V^*(c_1, c_2)$. 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 $\textsc{Com}$ is statistically hiding. □

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:

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

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* $\textsc{Com}$ *or solves the SIS problem with non-negligible probability.*

Proof. On input $(n, m, q, \mathbf{A})$ (the SIS problem instance) and a challenge commitment function $\textsc{Com}$, 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

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

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

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 $\mathbf{Ax} \bmod q$ 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 [21]. 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., $\omega(\log(n))$, 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 [15]. 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 [6].

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 [12].

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 $\mathbf{A}'\mathbf{x}' = 0 \bmod q$, with $\mathbf{A}' = [\mathbf{A}|\mathbf{y}]$ and $\mathbf{x}' = [\mathbf{x}| - 1]^T$. 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

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

Besides, q is bounded by the following theorem, which Micciancio and Regev proved in [20].

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 [9], 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 [25] 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 equation 5.

Bit-security	n	m	q	Commitment Length (bits)
100	64	2048	257	256

Table 2. Concrete Parameter

5 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.

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 a signature scheme from the current work. As we pointed out in Section 5.1, 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.

5.1 Signature via Fiat-Shamir Heuristics

If the verifier is replaced by a random oracle, one can derive signature schemes from identification counterparts. As pointed out by Lyubashevsky when comparing his lattice-based identification scheme [14] with Kawachi's solution [11], the latter does not result in an efficient signature scheme due to the fact that every bit of the challenge (thus, each bit of a message digest when we consider a signature application) results in a reasonable amount of data sent by the prover. For a 240-bit message digest, for example, Kawachi's scheme would result in a signature of over two million bits, when applying Fiat-Shamir heuristics.

Our identification scheme, however, has some characteristics of Lyubashevsky's, in the sense that we can relate a message digest with the variable α that the verifier sends to the prover in "pass 2" of Algorithm 2, instead of doing it with the challenge bits. Thus, we can make the field from which such a variable is defined to have a width that better suits the signature scheme needs, circumventing the drawback pointed out above. At the same time, we need to ensure that the total number of rounds we run the scheme is bigger than the desired bit-security level of the resulting signature. This is because an attacker who can correctly guess the challenge bits for each round can generate a signature.

Acknowledgments

We are grateful to Daniele Micciancio for his advice on the soundness proof and signature discussion, and we thank the anonymous referees for their useful suggestions. This work was supported by CASED (`www.cased.de`).

References

1. Miklós Ajtai. Generating hard instances of lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(7), 1996.
2. Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(65), 1996.
3. M. Bellare and A. Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. *Advances in Cryptology-Crypto 2002*, pages 149–162, 2002.
4. Pierre-Louis Cayrel and Pascal Véron. Improved code-based identification scheme, 2010. `http://arxiv.org/abs/1001.3017v1`.

5. Uriel Feige, Amos Fiat, and Adi Shamir. Zero knowledge proofs of identity. In *STOC*, pages 210–217. ACM, 1987.
6. Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426. ACM, 1990.
7. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.
8. P. Gaborit and M. Girault. Lightweight code-based identification and signature. *IEEE Transactions on Information Theory (ISIT)*, pages 186–194, 2007.
9. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.
10. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, page 304. ACM, 1985.
11. 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.
12. Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 560–569, New York, NY, USA, 2001. ACM.
13. 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.
14. 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.
15. 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.
16. Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. In *Theory of Cryptography Conference (TCC 2008)*, pages 37–54. Springer Berlin / Heidelberg, LNCS 4948, March 2008.
17. 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.
18. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. In *Computational Complexity*. Springer, 2007.
19. 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.
20. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.
21. Kazuo Ohta and Tatsuaki Okamoto. On concrete security treatment of signatures derived from identification. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 354–369. Springer, 1998.
22. Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994.
23. 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.
24. Pascal Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.
25. David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–303. Springer, 2002.