Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems

Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa

Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan
{kawachi,keisuke,xagawa5}@is.titech.ac.jp

Abstract. In this paper, we show that two variants of Stern's identification scheme [IEEE Transaction on Information Theory '96] are provably secure against concurrent attack under the assumptions on the *worst-case* hardness of lattice problems. These assumptions are weaker than those for the previous lattice-based identification schemes of Micciancio and Vadhan [CRYPTO '03] and of Lyubashevsky [PKC '08]. We also construct efficient ad hoc anonymous identification schemes based on the lattice problems by modifying the variants.

Keywords: Lattice-based cryptography, identification schemes, concurrent security, ad hoc anonymous identification schemes.

1 Introduction

Many researchers have so far developed cryptographic schemes based on combinatorial problems related to knapsacks, codes, and lattices, due to the intractability of the underlying problems, the efficiency of primitive operations, and the threat of quantum computers to number-theoretic schemes.

The cryptographic schemes based on combinatorial problems usually assume the *average-case* hardness of the underlying problem because they have to deal with randomly generated cryptographic instances such as keys, plaintexts, and ciphertexts. This implies security risk in such schemes since it is generally hard to show their average-case hardness. In fact, several attacks against such schemes, e.g., [25], were found in practical settings. The cryptographic schemes based only on the average-case hardness are more likely to be at risk of these kinds of attacks.

It is therefore significant to guarantee the security under the worst-case hardness. Ajtai [1] showed that the average-case hardness of some lattice problem is equivalent to its worst-case hardness. His seminal result opened the way to cryptographic schemes based on the worst-case hardness of lattice problems. Several lattice-based schemes were proposed such as public-key encryption schemes, e.g., by Ajtai and Dwork [2], and hash functions [1,11,19].

Among varieties of lattice-based cryptographic schemes, there are very few results on the identification (ID) schemes based on the worst-case hardness of lattice problems. For example, Micciancio and Vadhan proposed ID schemes based on the worst-case hardness of lattice problems, such as the gap versions of the Shortest Vector Problem. These schemes are obtained from their statistical zero-knowledge protocol with efficient

J. Pieprzyk (Ed.): ASIACRYPT 2008, LNCS 5350, pp. 372–389, 2008.

provers [20]. Recently, Lyubashevsky also constructed lattice-based ID schemes secure against active attack [14]. Unfortunately, the approximation factors of the underlying problems in their schemes are large for practical use as noted in [14, Sec. 5] since security parameters for ID schemes should be large in order to achieve the required hardness. Therefore, it is necessary to construct the schemes based on weaker assumptions, i.e., the assumptions on lattice problems with smaller approximation factors.

1.1 Our Contributions

In this paper, we propose two variants, which we call S^{+}_{GL} and $S^{+}_{C/IL}$, of Stern's ID scheme [26]. These variants are secure against *concurrent* attack[1] under the assumptions on the *worst-case* hardness of lattice problems, while Stern's original scheme assumes the *average-case* hardness of certain decoding problem in coding theory and the existence of a collision-resistant hash function, and its security is only against *passive* attack. The underlying problems of S^{+}_{GL} and $S^{+}_{C/IL}$ are the gap version of the Shortest Vector Problem with approximation factor $\tilde{O}(n)$ ($\mathrm{GapSVP}^{2}_{\tilde{O}(n)}$) and the Shortest Vector Problem for ideal lattices with approximation factor $\tilde{O}(n)$ ($\Lambda(f)$-$\mathrm{SVP}^{\infty}_{\tilde{O}(n)}$), respectively, where $\tilde{O}(g(n)) = O(g(n) \operatorname{poly} \log g(n))$ for a function g in n, The assumptions are weaker than those for the previous lattice-based ID schemes [20,14]. We stress that such weaker assumptions will take a step for practical use of lattice-based ID schemes.

Moreover, we show that our variants yield efficient ad hoc anonymous identification schemes (AID schemes). In an AID scheme, which introduced by Dodis, Kiayias, Nicolosi, and Shoup [7], the protocol is done by two parties, a prover and verifier, but we implicitly suppose an ad hoc group. Given public keys of all members of the group to the verifier (and the prover), the goal is to convince the verifier that the prover belongs to the group, without being specified who the prover is of the group, if and only if the prover is an actual member of the group. We formally define a concurrent version of the security notion, the security against impersonation under concurrent chosen-group attack, and prove that our AID schemes satisfy this security notion. Our schemes are based on the worst-case hardness of $\mathrm{GapSVP}^{2}_{\tilde{O}(n)}$ and $\Lambda(f)$-$\mathrm{SVP}^{\infty}_{\tilde{O}(n)}$. To authors' best knowledge, this is the first non-trivial construction under the assumption of the worst-case hardness of lattice problems.

1.2 Main Ideas

In this section, we only discuss the ID scheme S^{+}_{GL} based on GapSVP. We first construct a string commitment scheme based on the lattice problem which will be used in ID schemes. Then we will describe the idea of the proof on concurrent security of the variant. Finally, we give a sketch of our construction method of an AID scheme.

Before giving the overview, we review the underlying problem GapSVP_{γ} and the fundamental problem, the Small Integer Solution Problem ($\mathrm{SIS}_{q,m,\beta}$), on which our

[1] In *active attack*, an adversary could interact with the prover prior to impersonation. In *concurrent attack*, an adversary could interact with many different prover "clones" concurrently prior to impersonation. Each clone has the same secret key, but has independent random coins and maintains its own state. After interacting with many clones, the adversary tries impersonation.

variants are directly based. The informal definitions and the relationship of two problems are given as follows:

- $\mathrm{SIS}_{q,m,\beta}$: Given a random n-by-m matrix $\mathbf{A}$ whose elements are in $\mathbb{Z}_q$, the problem is finding an m-dimensional integral non-zero vector $\boldsymbol{z}$ such that $\boldsymbol{A}\boldsymbol{z} \equiv \mathbf{0} \pmod{q}$ and $\|\boldsymbol{z}\|_2 \leq \beta$.
- GapSVP^2_γ: Given an n-dimensional lattice L and a rational number d, the problem is outputting YES if there exists a non-zero vector $\boldsymbol{v} \in L$ such that $\|\boldsymbol{v}\|_2 \leq d$, or NO if for any non-zero vector $\boldsymbol{v} \in L$ $\|\boldsymbol{v}\|_2 > \gamma d$.
- ([19]) For suitable q and m, if there exists a probabilistic polynomial-time algorithm which solves $\mathrm{SIS}_{q,m,\beta}$ on the average then there exists a probabilistic polynomial-time algorithm which solves $\mathrm{GapSVP}^2_{\tilde{O}(\beta n^{1/2})}$ in the worst case.

As in Lyubashevsky's result [14], we use the above relationship for our security reduction. Hence we mainly deals with SIS instead of GapSVP.

We simply obtain the lattice-based hash functions as in [11]: Choose a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. For any $\boldsymbol{x} \in \{0,1\}^m$, a hash value is $f_{\mathbf{A}}(\boldsymbol{x}) := \mathbf{A}\boldsymbol{x} \bmod q$. A collision $(\boldsymbol{x}, \boldsymbol{x}')$ of the hash function $f_{\mathbf{A}}$ implies a solution $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}'$ of $\mathrm{SIS}_{q,m,\sqrt{m}}$. Thus, the security of the hash functions is based on the worst-case hardness of $\mathrm{GapSVP}^2_{\tilde{O}(\sqrt{nm})}$.

String commitment schemes: We construct a string commitment scheme from lattice-based hash functions. General constructions of string commitment schemes from collision-resistant hash functions were shown by Damgård, Pedersen, and Pfizmann [4] and Halevi and Micali [12]. Stern also constructed a string commitment scheme from collision-resistant hash functions in [26, Sec. III-A]: Let h be a hash function. Given a string s and a random string ρ, a commitment is $h(\rho \circ (\rho \oplus s))$, where $\circ$ and $\oplus$ denote the concatenation and XOR operators, respectively. However, its hiding property was not shown. We construct a string commitment scheme by a more direct and simpler way than the general one and Stern's one: Given s and ρ, a commitment is $h(\rho \circ s)$, where h is a lattice-based hash function. The binding property simply follows from the collision-resistance property of h. We derive its hiding property from ϵ-regularity of h for some negligible function ϵ (see, e.g., [16, Sec. 4.1]). As mentioned in the above, we have collision-resistant lattice-based hash functions based on the worst-case hardness of GapSVP, while Stern assumed the existence of collision-resistant hash functions.

Our ID scheme and its concurrent security: In Stern's scheme and our variant, a prover has a binary vector $\boldsymbol{x}$ with fixed Hamming weight as his/her secret key. We also feed to the prover and the verifier a matrix $\mathbf{A}$ as a system parameter and a vector $\boldsymbol{y}$ as the public key corresponding to $\boldsymbol{x}$. The task of the prover is to convince the verifier that he/she knows a correct secret key $\boldsymbol{x}$ satisfying a relation $\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$ and $\boldsymbol{x}$ has a valid weight.

In Stern's protocol [26], the prover computes three commitments and sends them to the verifier. The verifier sends a random challenge to the prover. The prover reveals two of three commitments corresponding to the challenge. He constructed the knowledge extractor which computes a collision of a hash function in a string commitment scheme or a secret key corresponding to the target public key if a passive adversary responds correctly to any challenges after sending commitments.

One of standard strategies to achieve concurrent security is to prove that a public key corresponds to multiple secret keys and that the protocol is witness indistinguishable

(WI) [8] and proof-of-knowledge: The reduction algorithm generates sk and pk and runs the adversary on pk by simulating the prover with sk. Using the knowledge extractor of the protocol, the algorithm obtains another sk' corresponding to pk with probability at least $1/2$ since the protocol is WI. The algorithm then solves the underlying problem by using pk, sk, and sk'.

In our reduction, when the algorithm is given $\mathbf{A}$, it generates a secret key $\boldsymbol{x}$ and a public key $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$, and feeds $\mathbf{A}$ and $\boldsymbol{y}$ to the adversary. Note that the algorithm can simulate the prover with $\mathbf{A}$ and $\boldsymbol{x}$ that the adversary concurrently accesses. Using the knowledge extractor for the adversary in Stern's proof, the algorithm obtains a collision of a string commitment scheme or a secret key $\boldsymbol{x}'$ such that $\boldsymbol{x}' \neq \boldsymbol{x}$ and $\mathbf{A}\boldsymbol{x}' = \boldsymbol{y}$, differently from the general strategy. In the former case, the algorithm outputs the collision $(\boldsymbol{s}, \boldsymbol{s}')$ of a hash function $h_{\mathbf{A}}$ in the string commitment scheme. Thus, the solution for SIS is obtained by $\boldsymbol{z} = \boldsymbol{s} - \boldsymbol{s}'$. In the latter case, the condition $\boldsymbol{x} \neq \boldsymbol{x}'$ will be satisfied with probability at least $1/2$ by witness indistinguishability of Stern's protocol. Thus, the algorithm has the solution $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}'$ for SIS. The ℓ_2 norm of both solutions is at most $\sqrt{m} = \tilde{O}(n^{1/2})$. From the relationship between SIS and GapSVP the assumption is the worst-case hardness of $\mathrm{GapSVP}^2_{\tilde{O}(n)}$.

AID schemes: Our construction for AID schemes also has the following structure: Each of l members in the ad hoc group has a vector $\boldsymbol{x}_i$ ($i = 1, \ldots, l$). Then, the common inputs of the scheme are a system parameter $\mathbf{A}$ and a set of public keys $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$ of the members, which satisfy $\boldsymbol{y}_i = \mathbf{A}\boldsymbol{x}_i$ ($i = 1, \ldots, l$). We can show that, by Stern's protocol, the prover can anonymously convince the verifier that the prover knows $\boldsymbol{x}_i$ corresponding to one of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$, since he/she knows a new vector $\boldsymbol{x}'$ such that $[\mathbf{A}\, \boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_l]\boldsymbol{x}' = \mathbf{0}$. (This idea is due to Wu, Chen, Wang, and Wang [27], who presented an AID scheme from certain combinatorial problem.) Additionally, we force the prover to prove that the positions of $+1$ and -1 in $\boldsymbol{x}'$ are proper by modifying Stern's protocol. We succeed to give security proof for the scheme, while Wu et al. gave no formal proof on the security of their scheme.

1.3 Comparison with Other Lattice-Based Schemes

ID schemes: In [20], Micciancio and Vadhan proposed a statistical zero-knowledge and proof-of-knowledge protocol for GapSVP. Combining it with lattice-based hash functions, we obtain an ID scheme which is secure against *passive attack* based on $\mathrm{SIS}_{q,m,\tilde{O}(n)}$, which can be reduced from $\mathrm{GapSVP}^2_{\tilde{O}(n^{1.5})}$.

In the scheme, the prover and the verifier are given a matrix $\mathbf{A}$ as a common input, and the prover has a binary vector $\boldsymbol{x}$ as secret information. The task of the prover is to convince the verifier that he/she knows $\boldsymbol{x}$ satisfying the relations that $\mathbf{A}\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{x}$ is relatively short. It seems difficult to directly simulate the prover since a simulator has to prepare a dummy short vector $\boldsymbol{x}'$ satisfying $\mathbf{A}\boldsymbol{x}' = \mathbf{0}$, which is the task of SIS itself. Thus, we cannot straightforwardly prove the concurrent security for their ID scheme.

By a simple modification, we can construct a concurrently secure ID scheme ($\mathrm{MV}^+_{\mathrm{GL}}$ for short) based on the worst-case hardness of lattice problems by Micciancio and Vadhan's ID scheme as noted in [20, Sec. 5]. In particular, applying techniques of De Santis, Di Crescenzo, Persiano, and Yung [6] and of Feige and Shamir [8], a modification of

Table 1. Comparisons among ID schemes and AID schemes. A secret key sk is $\boldsymbol{x} \in \{0,1\}^m$. The factor n denotes the security parameter. We denote the Hamming weight of $\boldsymbol{x}$ by $w_H(\boldsymbol{x})$. Assume that the protocols are repeated t times in parallel for reducing errors. In the table for AID schemes, l denotes the number of the members in the group. Note that the parameters in ideal-lattice-based versions are almost same as those in general-lattice-based versions.

ID schemes ($\mathbf{A}_0, \mathbf{A}_1, \mathbf{A} \in \mathbb{Z}_q^{n\times m}$)						
	Param.	Public key	Relation	γ in GapSVP_γ^2	Comm. cost	Errors
$\mathrm{MV}_{\mathrm{GL}}^+$ [20]	–	$\mathbf{A}_0, \mathbf{A}_1$	$\mathbf{A}_0\boldsymbol{x} = \mathbf{0}$ or $\mathbf{A}_1\boldsymbol{x} = \mathbf{0}$	$\tilde{O}(n^{1.5})$	$t \cdot \tilde{O}(n)$	1-sided
L_{GL} [14]	($\mathbf{A}$)	$\mathbf{A}, \boldsymbol{y}$	$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$	$\tilde{O}(n^2)$	$t \cdot \tilde{O}(n)$	2-sided
$\mathrm{S}_{\mathrm{GL}}^+$	$\mathbf{A}$	$\boldsymbol{y}$	$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$ and $w_H(\boldsymbol{x}) = m/2$	$\tilde{O}(n)$	$t \cdot \tilde{O}(n)$	1-sided
AID schemes ($\mathbf{A}_{i,0}, \mathbf{A}_{i,1}, \mathbf{A} \in \mathbb{Z}_q^{n\times m}$)						
Base	Param.	Set of pks	Relation	γ in GapSVP_γ^2	Comm. cost	Errors
$\mathrm{MV}_{\mathrm{GL}}^+$ [20]	–	$\{\mathbf{A}_{i,0}, \mathbf{A}_{i,1}\}_{i=1,\ldots,l}$	$\mathbf{A}_{i,0}\boldsymbol{x} = \mathbf{0}$ or $\mathbf{A}_{i,1}\boldsymbol{x} = \mathbf{0}$	$\tilde{O}(n^{1.5})$	$tl \cdot \tilde{O}(n)$	1-sided
L_{GL} [14]	$\mathbf{A}$	$\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$	$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}_i$	$\tilde{O}(n^2)$	$tl \cdot \tilde{O}(n)$	2-sided
$\mathrm{S}_{\mathrm{GL}}^+$	$\mathbf{A}$	$\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$	$\mathbf{A}\boldsymbol{x} = \boldsymbol{y}_i$ and $w_H(\boldsymbol{x}) = m/2$	$\tilde{O}(n)$	$t \cdot \tilde{O}(l+n)$	1-sided

the ID scheme can be proven to have concurrent security[2] based on the same problem as that in the original scheme.

Recently, Lyubashevsky proposed new concurrently secure ID schemes based on lattice problems [14]; we call it L_{GL} for short. In his protocol, the prover proves, given $\mathbf{A}$ and $\boldsymbol{y}$, he/she has $\boldsymbol{x} \in \{0,1\}^m$ such that $\mathbf{A}\boldsymbol{x} = \boldsymbol{y}$. Using an active adversary, his knowledge extractor obtains another vector $\boldsymbol{x}'$ such that $\mathbf{A}\boldsymbol{x}' = \boldsymbol{y}$ and the length of $\boldsymbol{x}'$ is at most $O(m^{1.5}) = \tilde{O}(n^{1.5})$. Thus, in the L_{GL} scheme, the underlying problem is $\mathrm{SIS}_{q,m,\tilde{O}(n^{1.5})}$, which can be reduced from $\mathrm{GapSVP}_{\tilde{O}(n^2)}^2$.

As mentioned in the previous section, the assumption of $\mathrm{S}_{\mathrm{GL}}^+$ is the worst-case hardness of $\mathrm{GapSVP}_{\tilde{O}(n)}^2$, which is weaker than those of $\mathrm{MV}_{\mathrm{GL}}^+$ and L_{GL}. This improvement is obtained by the condition that the knowledge extractor outputs another secret key $\boldsymbol{x}'$ whose length is at most $\sqrt{m} = \tilde{O}(\sqrt{n})$. Our schemes has 1-sided error (perfect completeness and soundness error), while L_{GL} has 2-sided error (completeness and soundness errors). As a summary, see Table 1.

AID schemes: By taking OR of l statements [6], we can straightforwardly obtain $\mathrm{MV}_{\mathrm{GL}}^+$-based and L_{GL}-based AID schemes, whose security are based on the worst-case hardness of lattice problems. We feed only $pk_1, \ldots, pk_l$ as the common inputs to the prover and the verifier. In this case, the prover convinces the verifier that he/she has a secret key corresponding to one of public keys, pk_i.

However, each of these simple modifications requires a large overhead cost involving the size of the ad hoc group. Let l be the number of the members of the group and n the security parameter. The protocol is run in t times in parallel to reduce the errors. The

[2] Combining ORing technique by De Santis et al. [6] and discarding technique by Feige and Shamir [8], we derive a construction technique for ID schemes secure against active attack. Moreover, we can construct concurrently secure ID schemes by the same technique as a folklore says.

communication costs of the $\mathrm{MV}^{+}_{\mathrm{GL}}$-based and L_{GL}-based schemes are $tl \cdot \tilde{O}(n)$. The size of a set of the public keys is $l \cdot \tilde{O}(n^2)$ and $\tilde{O}(n^2) + l \cdot \tilde{O}(n)$ in the modified versions of $\mathrm{MV}^{+}_{\mathrm{GL}}$ and L_{GL}, respectively.

On AID schemes, the L_{GL}-based and our schemes require many *vectors* proportional to the size of the group, while the $\mathrm{MV}^{+}_{\mathrm{GL}}$-based scheme requires many *matrices* proportional to the size of the group (see Table 1). Additionally, the communication cost of our schemes is $t \cdot \tilde{O}(n + l)$, while those in the $\mathrm{MV}^{+}_{\mathrm{GL}}$-based and L_{GL}-based schemes are $tl \cdot \tilde{O}(n)$. This shows the advantage of our scheme on the efficiency.

1.4 Organization

The rest of this paper is organized as follows. In Section 2, we review basic notations and notions, and the cryptographic schemes we consider. In Section 3, we review lattice-based hash functions and give a commitment scheme based on the lattice-based hash functions for our ID and AID schemes. In Section 4, we construct the ID scheme by combining the framework of Stern's scheme with our string commitment scheme. We present the AID scheme in Section 5.

In this paper, due to lack of space, we only describe the schemes based on GapSVP since the construction on $\Lambda(f)$-SVP follows from a similar strategy to that on GapSVP. We discuss the constructions on $\Lambda(f)$-SVP in the full paper.

2 Preliminaries

Basic notions and notations: We denote by n the security parameter of cryptographic schemes throughout this paper, which corresponds to the rank of the underlying lattice problems. We say that a problem is hard in the worst case if there exists no probabilistic polynomial-time algorithm solves the problem in the worst case with non-negligible probability. We sometimes use $\tilde{O}(g(n))$ for any function g in n as $O(g(n) \cdot \mathrm{polylog}(g(n)))$. We assume that all random variables are independent and uniform. For a positive integer n, let $[n]$ denote a set $\{1, 2, \ldots, n\}$.

For any $p \geq 1$, the ℓ_p norm of a vector $\boldsymbol{x} = {}^t(x_1, \ldots, x_n) \in \mathbb{R}^n$, denoted by $\|\boldsymbol{x}\|_p$, is $(\sum_{i \in [n]} x_i^p)^{1/p}$. For ease of notation, we define $\|\boldsymbol{x}\| := \|\boldsymbol{x}\|_2$. The ℓ_∞ norm is defined as $\|\boldsymbol{x}\|_\infty = \lim_{p \to \infty} \|\boldsymbol{x}\|_p = \max_{i \in [n]} |x_i|$. Let $w_H(\boldsymbol{x})$ denote the Hamming weight of $\boldsymbol{x}$, i.e., the number of non-zero elements in $\boldsymbol{x}$. Let $\mathrm{B}(m, w)$ denote the set of binary vectors in $\{0,1\}^m$ whose Hamming weights are exactly equal to w, i.e., $\mathrm{B}(m, w) := \{\boldsymbol{x} \in \{0,1\}^m \mid w_H(\boldsymbol{x}) = w\}$. We denote the concatenation of two vectors or strings $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ by $\boldsymbol{v}_1 \circ \boldsymbol{v}_2$.

We omit the definitions of zero-knowledge arguments and witness-indistinguishable protocols. For formal definitions, see textbooks, e.g., by Goldreich [10].

Hash functions: We briefly review the definition of collision-resistant hash function families. Let $\mathcal{H}_n = \{h_k : M_n \to D_n \mid k \in K_n\}$ be a family of hash functions, where M_n, D_n, and K_n denote a space of messages, digests, and indices, respectively. Let $\mathcal{H} = \{\mathcal{H}_n\}_{n \in \mathbb{N}}$. Roughly speaking, if $\mathcal{H}$ is collision resistant, any polynomial-time adversary cannot, on input a random index k, output a collision of the hash function indexed by k. For a formal definition, see, e.g., the textbook by Katz and Lindell [13, Sec. 4.6.1].

String commitment schemes: We consider a string commitment scheme in the trusted setup model. The trusted setup model is often required to construct practically efficient cryptographic schemes such as non-interactive string commitment schemes. In this model, we assume that a trusted party $\mathcal{T}$ honestly sets up a system parameter for the sender and the receiver.

First $\mathcal{T}$ distributes the index k of a commitment function to the sender and the receiver. Both parties then share a common function Com_k by a given k. The scheme runs in two phase, called committing and revealing phases. In the committing phase, the sender commits his/her decision, say a string s, to a commitment string $c = \mathrm{Com}_k(s;\rho)$ with a random string ρ and sends c to the receiver. In the revealing phase, the sender gives the receiver the decision s and the random string ρ. The receiver verifies the validity of c by computing $\mathrm{Com}_k(s;\rho)$.

We require two security notions of the string commitment schemes, statistically-hiding and computationally-binding properties. Intuitively, we say that the commitment scheme is statistically hiding, if any computationally unbounded adversarial receiver cannot distinguish two commitment strings generated from two distinct strings. Also, it is computationally binding, if any polynomial-time adversarial sender cannot change the committed string after sending the commitment. See, e.g., [12] for the formal definition.

Canonical identification schemes: Let $\mathcal{SI} = (\mathsf{SetUp}, \mathsf{KG}, \mathsf{P}, \mathsf{V})$ be an identification scheme, where SetUp is the setup algorithm which on input 1^n outputs *param*, KG is the key-generation algorithm which on input *param* outputs (pk, sk), P is the prover algorithm taking input *sk*, V is the verifier algorithm taking inputs *param* and *pk*. We say $\mathcal{SI}$ is a canonical identification scheme if it is a public-coin 3-move protocol.

We are interested in concurrent attack, which is stronger than active and passive attack. We employ the definition of concurrent security in [3]. In concurrent attack, the adversary will play the role of a cheating verifier prior to impersonation and can interact many different prover clones concurrently. Each clone has the same secret key, but has independent random coins and maintains its own state. We say $\mathcal{SI}$ is secure against impersonation under concurrent attack, if any polynomial-time adversary cannot, given a random public key of a legitimate prover, impersonate the legitimate prover. For the formal definition, see [3].

Ad hoc anonymous identification schemes: An AID scheme allows a user to anonymously prove his/her membership in a group if and only if the user is an actual member of the group, where the group is formed in an ad hoc fashion without help of the group manager. We then assume that every user registers his/her public key to the public key infrastructure.

We define the algorithms in AID schemes. An AID scheme is four tuple $\mathcal{AID} = (\mathsf{SetUp}, \mathsf{Reg}, \mathsf{P}, \mathsf{V})$, where SetUp is the setup algorithm which on input 1^n outputs *param*, Reg is the key generation and registration algorithm which on input *param* outputs (pk, sk), P is the prover algorithm taking inputs *param*, a set of public keys $R = (pk_1, \ldots, pk_l)$, and one of the secret keys sk_i such that $pk_i \in R$, and V is the verifier algorithm taking inputs *param* and R. For more formal definition, see [7].

There are two goals for security of AID schemes: security against impersonation and anonymity. Dodis et al. formally defined security against impersonation under passive

attack. They mentioned the definition of security against impersonation under concurrent attack. However, they did not give the formal definition (see [7, Sec. 3.2]). Thus, we define the security notion with respect to concurrent attack. In the setting of chosen-group attack, the adversary could force the prover to prove the membership in an arbitrary group if the prover is indeed a member of the group. Additionally, concurrent attack allows the cheating verifier to interact with the clones of any provers. Also, they allow the cheating prover to interact with the clones of provers, but prohibit it from interacting with the target provers. We say $\mathcal{AID}$ is secure against impersonation under concurrent chosen-group attack, if any polynomial-time adversary cannot impersonate the legitimate prover in the above settings.

The security notion, anonymity against full key exposure, captures the property that an adversary cannot distinguish two transcripts even if the adversary has the secret keys of all the members. We say $\mathcal{AID}$ is anonymous against full key exposure if any polynomial-time adversary cannot distinguish two provers with a common set of public keys even though the adversary generates all keys of the set. The formal definitions of two notions are in the full paper.

3 Main Tools

In this section, we review main tools, lattices, lattice problems, and lattice-based hash functions, and construct string commitment schemes.

Lattices and lattice problems: We first review fundamental notions of lattices, well-known lattice problems, and a related problem.

An n-dimensional lattice in $\mathbb{R}^m$ is the set $L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{\sum_{i\in[n]} \alpha_i \boldsymbol{b}_i \mid \alpha_i \in \mathbb{Z}\}$ of all integral combinations of n linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^m$. The sequence of vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ is called a *basis* of the lattice L and denoted by $\mathbf{B}$. For more details on lattices, see the textbook by Micciancio and Goldwasser [18].

We give the definitions of well-known lattice problems, the Shortest Vector Problem (SVP^p) and its approximation version (SVP^p_γ): The problem SVP^p is, given a basis $\mathbf{B}$ of a lattice L, finding the shortest non-zero vector $\boldsymbol{v}$ in L in the ℓ_p norm. The problem SVP^p_γ is, given a basis $\mathbf{B}$ of a lattice L, finding a non-zero vector $\boldsymbol{v}$ in L such that for any non-zero vector $\boldsymbol{x}$ in L, $\|\boldsymbol{v}\|_p \leq \gamma \|\boldsymbol{x}\|_p$.

We next give the definition of the gap version of SVP^p_γ, which is the underlying problem of lattice-based hash functions.

Definition 3.1 (GapSVP^p_γ **[18]).** *For a gap function γ, an instance of* GapSVP^p_γ *is a pair* $(\mathbf{B}, d)$ *where* $\mathbf{B}$ *is a basis of a lattice L and d is a rational number. In YES input there exists a vector $\boldsymbol{v} \in L\setminus\{\mathbf{0}\}$ such that $\|\boldsymbol{v}\|_p \leq d$. In NO input, for any vector $\boldsymbol{v} \in L\setminus\{\mathbf{0}\}$, $\|\boldsymbol{v}\|_p > \gamma d$.*

We also define the Small Integer Solution problem SIS (in the ℓ_p norm), which is often considered in the context of average-case/worst-case connections and a source of lattice-based hash functions as we see later.

Definition 3.2 ($\mathrm{SIS}^p_{q,m,\beta}$ **[19]).** *For a fixed integer q and a real β, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the problem is finding a non-zero integer vector $\boldsymbol{z} \in \mathbb{Z}^m$ such that $\mathbf{A}\boldsymbol{z} \equiv \mathbf{0} \pmod q$ and $\|\boldsymbol{z}\|_p \leq \beta$.*

The relation between SIS and GapSVP is reviewed in the next paragraph.

Lattice-based hash functions: We review the lattice-based hash functions. For a prime $q = q(n) = n^{O(1)}$ and an integer $m = m(n) > n \log q(n)$, we define a family of hash functions,

$$\mathcal{H}(q,m) = \{f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n \mid \mathbf{A} \in \mathbb{Z}_q^{n\times m}\},$$

where $f_{\mathbf{A}}(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x} \bmod q$.

Originally, Ajtai [1] showed that the worst-case hardness of GapSVP_γ^2 for some polynomial $\gamma(n)$ is reduced to the average-case hardness of $\mathrm{SIS}_{q,m,n}^2$ for suitable $q(n)$ and $m(n)$. It is known that $\mathcal{H}(q,m)$ is indeed collision resistant for suitably chosen q and m by Goldreich, Goldwasser, and Halevi [11]. They observed that finding a collision $(\boldsymbol{x},\boldsymbol{x}')$ for $f_{\mathbf{A}} \in \mathcal{H}(q,m)$ implies finding a short non-zero vector $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}'$ such that $\|\boldsymbol{z}\| \le \sqrt{m}$ and $\mathbf{A}\boldsymbol{z} \equiv \mathbf{0} \pmod q$, i.e., solving $\mathrm{SIS}_{q,m,\sqrt{m}}^2$. Recently, Micciancio and Regev showed that $\mathcal{H}(q,m)$ is collision resistant under the assumption that $\mathrm{GapSVP}_{\tilde{O}(n)}^2$ is hard in the worst case [19].

Theorem 3.1 ([19]). *For any polynomially bounded functions $\beta = \beta(n)$, $m = m(n)$, $q = q(n)$, with $q \ge 4\sqrt{mn^{3/2}}\beta$ and $\gamma = 14\pi\sqrt{n}\beta$, there exists a probabilistic polynomial-time reduction from solving GapSVP_γ^2 in the worst case to solving $\mathrm{SIS}_{q,m,\beta}^2$ on the average with non-negligible probability.*

There were another reductions from the gap version of the covering radius problem GapCRP_γ, the shortest independent vector problem SIVP_γ, and the guaranteed distance decoding problem GDD_γ by adjusting the parameters [19]. It is worth that we note the results following the above results: Peikert [22] showed the reductions from the same problems in any ℓ_p norms for $p \ge 2$. The recent paper [9, Sec. 9] by Gentry, Peikert, and Vaikuntanathan showed that the modulus q in SIS can be $\tilde{O}(n)$.

A string commitment scheme: General constructions of statistically-hiding and computationally-binding string commitment schemes are known from a family of collision-resistant hash functions [4,12]. Their constructions used universal hash functions for the statistically-hiding property.

Here, we give a more direct and simpler construction from the lattice-based hash functions without the universal hash functions. The input of the commitment function is an m-bit vector $\boldsymbol{x}$ obtained by concatenating a random string $\rho = (\rho_1, \ldots, \rho_{m/2})$ and a message string $s = (s_1, \ldots, s_{m/2})$, i.e., $\boldsymbol{x} = \rho \circ s$. We then define the commitment function on inputs s and ρ as

$$\mathrm{Com}_{\mathbf{A}}(s;\rho) := \mathbf{A}\boldsymbol{x} \bmod q = \mathbf{A}^t(\rho_1, \ldots, \rho_{m/2}, s_1, \ldots, s_{m/2}) \bmod q.$$

Lemma 3.1. *For $m > 10n \log q$, if $\mathrm{SIS}_{q,m,\sqrt{m}}$ is hard on the average, then $\mathrm{Com}_{\mathbf{A}}$ is a statistically-hiding and computationally-binding string commitment scheme in the trusted set up model. In particular, for any polynomially bounded functions $m = m(n)$, $q = q(n)$, $\gamma = \gamma(n)$, with $q \ge 4mn^{3/2}$, $\gamma = 14\pi\sqrt{nm}$, and $m > 10n \log q$, $\mathrm{Com}_{\mathbf{A}}$ is a statistically-hiding and computationally-binding string commitment scheme in the trusted setup model if GapSVP_γ^2 is hard in the worst case.*

Before the proof, we review a definition of statistical distances: Given two probability density functions ϕ_1 and ϕ_2 on a finite set S, we define the statistical distance between them as $\Delta(\phi_1,\phi_2) := \frac{1}{2}\sum_{x\in S} |\phi_1(x) - \phi_2(x)|$.

Proof. The computationally-binding property immediately follows from the collision-resistant property. We now show the statistically-hiding property.

Let $\mathbf{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_m]$. We then have $\mathrm{Com}_{\mathbf{A}}(s;\rho) = \sum_{i=1}^{m/2} \rho_i \boldsymbol{a}_i + \sum_{i=1}^{m/2} s_i \boldsymbol{a}_{i+m/2}$. The following claim in [24] says that a random subset sum of $\boldsymbol{a}_i$ is statistically close to the uniform distribution for almost all choices of $\boldsymbol{a}_i$.

Claim ([24]). Let G be some finite Abelian group and let l be some integer. For any l elements $g_1, \ldots, g_l \in G$, consider $\Delta(\sum_{i\in[l]} a_i g_i, u)$, where u and a_i is chosen uniformly at random from G and $\{0,1\}$, respectively. Then the expectation of this statistical distance over a uniform choice of $g_1, \ldots, g_l \in G$ is at most $\sqrt{|G|/2^l}$. In particular, the probability that this statistical distance is more than $(|G|/2^l)^{1/4}$ is at most $(|G|/2^l)^{1/4}$.

In our proof, we consider $\mathbb{Z}_q^n$ as a finite Abelian group G. Since $m > 10n\log q$, $(|G|/2^{m/2})^{1/4} \le q^{-n}$. Thus, for all but an at most q^{-n} fraction of $\mathbf{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m] \in \mathbb{Z}_q^{n\times m}$, we have that $\Delta(\boldsymbol{u}, \sum_{i\in[m/2]} \rho_i \boldsymbol{a}_i) \le q^{-n}$, where $\boldsymbol{u} \in \mathbb{Z}_q^n$ is uniform random variable. Assume that we have such $\mathbf{A}$. So, we have $\Delta(\boldsymbol{u}, \mathrm{Com}_{\mathbf{A}}(0^{m/2};\rho)) \le q^{-n}$. By the definition of $\mathrm{Com}_{\mathbf{A}}$, for any $s \in \{0,1\}^{m/2}$, we have $\Delta(\boldsymbol{u}, \mathrm{Com}_{\mathbf{A}}(s;\rho)) \le q^{-n}$. By the triangle inequality, we obtain

$$\Delta(\mathrm{Com}_{\mathbf{A}}(s_1;\rho_1), \mathrm{Com}_{\mathbf{A}}(s_2;\rho_2)) \le \Delta(\boldsymbol{u}, \mathrm{Com}_{\mathbf{A}}(s_1;\rho_2)) + \Delta(\boldsymbol{u}, \mathrm{Com}_{\mathbf{A}}(s_2;\rho_2)) \le 2q^{-n},$$

for any message s_1 and s_2. This shows that, for all but negligible fraction of choice of $\mathbf{A}$, the distributions of two commitments are statistically close. □

Using the Merkle-Damgård technique, we obtain a string commitment scheme whose commitment function is $\mathrm{Com}_{\mathbf{A}} : \{0,1\}^* \times \{0,1\}^{m/2} \to \mathbb{Z}_q^n$ rather than $\mathrm{Com}_{\mathbf{A}} : \{0,1\}^{m/2} \times \{0,1\}^{m/2} \to \mathbb{Z}_q^n$ as the following.

Assume that $m = 2r$. Let $\mathbf{A} = [\mathbf{B}\,\mathbf{C}]$, where $\mathbf{B}, \mathbf{C} \in \mathbb{Z}_q^{n\times r}$. For $\mathbf{X} \in \mathbb{Z}_q^{n\times l}$, we define $f_{\mathbf{X}} : \{0,1\}^l \to \mathbb{Z}_q^n$ as the hash function $f_{\mathbf{X}}(\boldsymbol{s}) = \mathbf{X}\boldsymbol{s} \bmod q$. Let l be $\lceil n\log q \rceil$ and let $t : \mathbb{Z}_q^n \to \{0,1\}^l$ be some one-to-one function that we can compute t and t^{-1} efficiently. Let $\mathsf{pad} : \{0,1\}^* \to \{0,1\}^*$ be a padding function for the Merkle-Damgård construction. Applying the Merkle-Damgård construction to $f_{\mathbf{C}}$, we obtain a new hash function $h_{\mathbf{C}} : \{0,1\}^* \to \mathbb{Z}_q^n$. The precise definition of $h_{\mathbf{C}}$ is as follows:

Hash function $h_{\mathbf{C}}$:

1. On input s, obtain a padded message $S \leftarrow \mathsf{pad}(s)$.
2. Chop it into $(S_0, \ldots, S_k)$, where $S_i \in \{0,1\}^{r-l}$.
3. Let $H_0 = \mathbf{0}$ (more generally, some fixed H_0 can be used).
4. For $i = 1$ to $k+1$ do $H_i \leftarrow f_{\mathbf{C}}(t(H_{i-1}) \circ S_{i-1})$.
5. Output H_{k+1}.

Our new commitment scheme is defined as follows: for $s \in \{0,1\}^*$ and $\rho \in \{0,1\}^r$,

$$\mathrm{Com}_{\mathbf{A}}(s;\rho) := h_{\mathbf{C}}(s) + f_{\mathbf{B}}(\rho) \bmod q.$$

Lemma 3.2. *If there exists a polynomial-time machine outputting a collision for* $\mathrm{Com}_{\mathbf{A}}$*, then there exists a polynomial-time machine outputting a collision for* $f_{\mathbf{A}}$*.*

Proof. Let us assume that we obtain a collision $(s,\rho),(\tilde{s},\tilde{\rho}) \in \{0,1\}^* \times \{0,1\}^r$ for $\mathrm{Com}_{\mathbf{A}}$. By the assumption, we have

$$h_{\mathbf{C}}(s) + f_{\mathbf{B}}(\rho) \equiv h_{\mathbf{C}}(\tilde{s}) + f_{\mathbf{B}}(\tilde{\rho}) \pmod{q}.$$

If $\rho = \tilde{\rho}$, we have $s \neq \tilde{s}$ and $h_{\mathbf{C}}(s) = h_{\mathbf{C}}(\tilde{s})$. Using the reduction for the Merkle-Damgård construction (see e.g., [13, Thm. 4.14]), we obtain $u \neq \tilde{u} \in \{0,1\}^r$ such that $f_{\mathbf{C}}(u) = f_{\mathbf{C}}(\tilde{u})$. Thus, we have a collision $u \circ \rho, \tilde{u} \circ \rho \in \{0,1\}^{2r}$ for $f_{\mathbf{A}}$.

Next, we assume that $\rho \neq \tilde{\rho}$. Let S and $\tilde{S}$ be padded messages of s and $\tilde{s}$, respectively. Assume that S and $\tilde{S}$ are chopped into $(S_0,\ldots,S_k)$ and $(\tilde{S}_0,\ldots,\tilde{S}_{k'})$, respectively. Let H_k and $\tilde{H}_{k'}$ be inner hash values for s and $\tilde{s}$ in the algorithm, respectively. By the definition of H_k and $\tilde{H}_{k'}$, we obtain

$$h_{\mathbf{C}}(s) = f_{\mathbf{C}}(t(H_k) \circ S_k),$$
$$h_{\mathbf{C}}(\tilde{s}) = f_{\mathbf{C}}(t(\tilde{H}_{k'}) \circ \tilde{S}_{k'}).$$

Combining the above equations with the assumption, we obtain

$$f_{\mathbf{A}}(t(H_k) \circ S_k \circ \rho) = f_{\mathbf{A}}(t(\tilde{H}_{k'}) \circ \tilde{S}_{k'} \circ \tilde{\rho}).$$

So, we have a collision $t(H_k) \circ S_k \circ \rho$ and $t(\tilde{H}_{k'}) \circ \tilde{S}_{k'} \circ \tilde{\rho} \in \{0,1\}^{2r}$ for $f_{\mathbf{A}}$. □

We use this commitment scheme in the rest of the paper. We often abuse the notation of $\mathrm{Com}_{\mathbf{A}}$. For example, $\mathrm{Com}_{\mathbf{A}}(v_1, v_2; \rho)$ denotes $\mathrm{Com}_{\mathbf{A}}(\mathrm{string}(v_1) \circ \mathrm{string}(v_2); \rho)$, where $\mathrm{string}(v)$ is a binary representation of v.

4 An Identification Scheme

Our variant $\mathrm{S}^+_{\mathrm{GL}}$ is obtained by replacing the string commitment scheme in Stern's ID scheme [26] with our lattice-based one. Stern's protocol deals with the decoding problem on binary codewords called the Syndrome Decoding Problem[3]. He also proposed that an analogous scheme in $\mathbb{Z}_q$, where q is extremely small (typically 3, 5, or 7) [26, Sec. VI]. We adjust this parameter to connect his framework to our assumptions of the lattice problems.

We now describe the protocol $\mathrm{S}^+_{\mathrm{GL}}$ below. Obviously, it has perfect completeness, and at most 2/3 soundness error. By parallelizing each step of this protocol in $t = \omega(\log n)$ times, the soundness error becomes negligibly small. To simplify the notations, we write Com instead of $\mathrm{Com}_{\mathbf{A}}$ and we do not write random strings in Com explicitly.

SetUp: The setup algorithm, on input 1^n, outputs a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

KG: The key-generation algorithm, on input $\mathbf{A}$, chooses a random vector $\boldsymbol{x} \in \mathrm{B}(m, m/2)$ and computes $\boldsymbol{y} := \mathbf{A}\boldsymbol{x} \bmod q$. It outputs $(pk, sk) = (\boldsymbol{y}, \boldsymbol{x})$.

P, V: The common inputs are $\mathbf{A}$ and $\boldsymbol{y}$. The prover's auxiliary input is $\boldsymbol{x}$. They interact as follows:

[3] The Syndrome Decoding Problem is defined as follows: Given $\mathbf{A} \in \mathbb{Z}_2^{n \times m}$, $\boldsymbol{y} \in \mathbb{Z}_2^n$, and $w \in \mathbb{N}$, the problem is finding a vector $\boldsymbol{x} \in \mathrm{B}(m, w)$ such that $\mathbf{A}\boldsymbol{x} \equiv \boldsymbol{y} \bmod 2$. We can consider this problem as a restricted version of $\mathrm{SIS}_{q,m,\beta}$.

Step P1: Choose a random permutation π over $[m]$ and a random vector $\boldsymbol{r} \in \mathbb{Z}_q^m$ and send commitments c_1, c_2, and c_3 computed as
- $c_1 = \mathrm{Com}(\pi, \mathbf{A}\boldsymbol{r})$,
- $c_2 = \mathrm{Com}(\pi(\boldsymbol{r}))$,
- $c_3 = \mathrm{Com}(\pi(\boldsymbol{x} + \boldsymbol{r}))$.

Step V1 Send a random challenge $Ch \in \{1, 2, 3\}$ to P.

Step P2
- If $Ch = 1$, reveal c_2 and c_3. So, send $\boldsymbol{s} = \pi(\boldsymbol{x})$ and $\boldsymbol{t} = \pi(\boldsymbol{r})$.
- If $Ch = 2$, reveal c_1 and c_3. Send $\phi = \pi$ and $\boldsymbol{u} = \boldsymbol{x} + \boldsymbol{r}$.
- If $Ch = 3$, reveal c_1 and c_2. Send $\psi = \pi$ and $\boldsymbol{v} = \boldsymbol{r}$.

Step V2
- If $Ch = 1$, check that $c_2 = \mathrm{Com}(\boldsymbol{t})$, $c_3 = \mathrm{Com}(\boldsymbol{s} + \boldsymbol{t})$, and $\boldsymbol{s} \in \mathrm{B}(m, m/2)$.
- If $Ch = 2$, check that $c_1 = \mathrm{Com}(\phi, \mathbf{A}\boldsymbol{u} - \boldsymbol{y})$ and $c_3 = \mathrm{Com}(\phi(\boldsymbol{u}))$.
- If $Ch = 3$, check that $c_1 = \mathrm{Com}(\psi, \mathbf{A}\boldsymbol{v})$ and $c_2 = \mathrm{Com}(\psi(\boldsymbol{v}))$.

Output $Dec = 1$ if all checks are passed, otherwise output $Dec = 0$.

4.1 Statistical Zero-Knowledge Property

The proof of the zero-knowledge property of the original protocol is in [26, Thm. 4]. Stern left completion of the proof as the problem for reader. Thus, we give the whole proof that Stern's protocol is statistically zero knowledge when Com is a statistically-hiding and computationally-binding string commitment scheme.

Theorem 4.1. *The protocol is statistically zero knowledge when* Com *is a statistically-hiding and computationally-binding string commitment scheme.*

Proof. Following the definition, we construct a simulator $\mathcal{S}$ which on input $\mathbf{A}$ and $\boldsymbol{y}$ and given oracle access to a cheating verifier $\mathcal{CV}$, outputs a simulated transcript. A real transcript between P and $\mathcal{CV}$ on input $\mathbf{A}$ and $\boldsymbol{y}$ is denoted by $\langle \mathsf{P}, \mathcal{CV} \rangle(\mathbf{A}, \boldsymbol{y})$.

First, $\mathcal{S}$ chooses a random value $\bar{c}$ from $\{1, 2, 3\}$ which is a prediction what value the cheating verifier $\mathcal{CV}$ will *not* choose. Next, it chooses a random tape of $\mathcal{CV}$, denoted by r'. We remark that, by the assumption on the commitment, the distributions of a challenge from $\mathcal{CV}$ in the real interaction and in the simulation are statistically close.

Case $\bar{c} = 1$: $\mathcal{S}$ computes $\boldsymbol{x}' \in \mathbb{Z}_q^m$ such that $\mathbf{A}\boldsymbol{x}' = \boldsymbol{y}$ by using linear algebra. Next, it chooses a random permutation π' over $[m]$, a random vector $\boldsymbol{r}' \in \mathbb{Z}_q^m$, and random strings ρ'_1, ρ'_2, and ρ'_3. So, it computes

- $c'_1 := \mathrm{Com}(\pi', \mathbf{A}\boldsymbol{r}'; \rho'_1)$,
- $c'_2 := \mathrm{Com}(\pi'(\boldsymbol{r}'); \rho'_2)$,
- $c'_3 := \mathrm{Com}(\pi'(\boldsymbol{x}' + \boldsymbol{r}'); \rho'_3)$.

It sends them to $\mathcal{CV}$. Since the commitment scheme is statistically hiding, the distribution of a challenge from $\mathcal{CV}$ is statistically close to the real distribution. Receiving a challenge Ch from $\mathcal{CV}$, the simulator $\mathcal{S}$ computes a transcript as follows:

- If $Ch = 1$, $\mathcal{S}$ outputs $\perp$ and halts.
- If $Ch = 2$, it outputs $(r'; (c'_1, c'_2, c'_3), 2, (\pi', \boldsymbol{x}' + \boldsymbol{r}', \rho'_1, \rho'_3))$.
- If $Ch = 3$, it outputs $(r'; (c'_1, c'_2, c'_3), 3, (\pi', \boldsymbol{r}', \rho'_1, \rho'_2))$.

We analyze the case $Ch = 2$. In this case, we obtain that

$$\langle \mathsf{P}, C\mathcal{V}\rangle(\mathbf{A}, \boldsymbol{y}) = (r; (c_1, c_2, c_3), 2, (\pi, \boldsymbol{x} + \boldsymbol{r}, \rho_1, \rho_3),$$
$$\mathcal{S}(\mathbf{A}, \boldsymbol{y}) = (r'; (c'_1, c'_2, c'_3), 2, (\pi', \boldsymbol{x}' + \boldsymbol{r}', \rho'_1, \rho'_3)).$$

Assume that $(\pi', \boldsymbol{r}', \rho'_1, \rho'_3) = (\pi, \boldsymbol{r} + \boldsymbol{x} - \boldsymbol{x}', \rho_1, \rho_3)$. By this equation, we have that $c'_1 = c_1$, $c'_3 = c_3$, and the responses from the simulator equal to the responses from the prover. Since the commitment is statistically hiding, we have the distributions of c_2 and c'_2 are statistically close. Thus, we conclude that the both distributions of the simulated transcript and the real transcript are statistically close.

It is straightforward to show it in the case $Ch = 3$ by using the equation $(\pi', \boldsymbol{r}') = (\pi, \boldsymbol{r})$. Thus, we omit this part from the proof.

Case $\bar{c} = 2$**:** $\mathcal{S}$ chooses a random permutation π' over $[m]$, two random vectors $\boldsymbol{r}' \in \mathbb{Z}_q^m$, $\boldsymbol{x}' \in \mathrm{B}(m, m/2)$, and random strings ρ'_1, ρ'_2, and ρ'_3. $\mathcal{S}$ computes commitments

- $c'_1 := \mathrm{Com}(\pi', \mathbf{A}\boldsymbol{r}'; \rho'_1)$,
- $c'_2 := \mathrm{Com}(\pi'(\boldsymbol{r}'); \rho'_2)$,
- $c'_3 := \mathrm{Com}(\pi'(\boldsymbol{x}' + \boldsymbol{r}'); \rho'_3)$.

It sends them to $C\mathcal{V}$. Receiving a challenge Ch, the simulator computes a transcript as follows:

- If $Ch = 1$, then $\mathcal{S}$ outputs $(r'; (c'_1, c'_2, c'_3), 1, (\pi'(\boldsymbol{x}'), \pi'(\boldsymbol{r}'), \rho'_2, \rho'_3))$.
- If $Ch = 2$, then it outputs $\perp$ and halts.
- If $Ch = 3$, then it outputs $(r'; (c'_1, c'_2, c'_3), 3, (\pi', \boldsymbol{r}', \rho'_1, \rho'_2))$.

We analyze the case $Ch = 1$. In this case, we have that

$$\langle \mathsf{P}, C\mathcal{V}\rangle(\mathbf{A}, \boldsymbol{y}) = (r; (c_1, c_2, c_3), 1, (\pi(\boldsymbol{x}), \pi(\boldsymbol{r}), \rho_2, \rho_3),$$
$$\mathcal{S}(\mathbf{A}, \boldsymbol{y}) = (r'; (c'_1, c'_2, c'_3), 1, (\pi'(\boldsymbol{x}'), \pi'(\boldsymbol{r}'), \rho'_2, \rho'_3)).$$

Let χ be a permutation over $[m]$ such that $\chi(\boldsymbol{x}') = \boldsymbol{x}$. In this case, we set $(\pi', \boldsymbol{r}', \rho'_2, \rho'_3) = (\pi \circ \chi^{-1}, \chi(\boldsymbol{r}), \rho_2, \rho_3)$. By this equation, we have that $\pi(\boldsymbol{x}) = \pi'(\boldsymbol{x}')$, $\pi(\boldsymbol{r}) = \pi'(\boldsymbol{r}')$, $c'_2 = c_2$, and $c'_3 = c_3$, that is, the responses from the simulator equal to the responses from the prover. Since the commitment scheme is statistically hiding, the distributions of the real transcript and the output of the simulator are statistically close.

We omit the proof of the case $Ch = 3$, since it is trivial.

Case $\bar{c} = 3$**:** $\mathcal{S}$ chooses a random permutation π over $[m]$, two random vectors $\boldsymbol{r} \in \mathbb{Z}_q^m$, $\boldsymbol{x}' \in \mathrm{B}(m, m/2)$, and random strings ρ_1, ρ_2, and ρ_3. $\mathcal{S}$ computes

- $c_1 := \mathrm{Com}(\pi, \mathbf{A}(\boldsymbol{x}' + \boldsymbol{r}) - \boldsymbol{y}; \rho_1)$,
- $c_2 := \mathrm{Com}(\pi(\boldsymbol{r}); \rho_2)$,
- $c_3 := \mathrm{Com}(\pi(\boldsymbol{x}' + \boldsymbol{r}); \rho_3)$.

It sends them to $C\mathcal{V}$.

- If $Ch = 1$, then $\mathcal{S}$ outputs $(r'; (c_1, c_2, c_3), 1, (\pi(\boldsymbol{x}'), \pi(\boldsymbol{r}), \rho_2, \rho_3)$.
- If $Ch = 2$, then it outputs $(r'; (c_1, c_2, c_3), 2, (\pi, \boldsymbol{x}' + \boldsymbol{r}'))$.
- If $Ch = 3$, it outputs $\perp$ and halts.

In the case $Ch = 1$, we consider the equation $(\pi', \boldsymbol{r}', \rho_2', \rho_3') = (\pi \circ \chi^{-1}, \chi(\boldsymbol{r}), \rho_2, \rho_3)$, where χ denotes a permutation over $[m]$ such that $\chi(\boldsymbol{x}') = \boldsymbol{x}$. The remaining part of proof is the same as that in the case $\bar{c} = 2$ and $Ch = 1$. In the case $Ch = 2$, we let $(\pi', \boldsymbol{r}', \rho_1', \rho_3') = (\pi, \boldsymbol{r} + \boldsymbol{x} - \boldsymbol{x}', \rho_1, \rho_3)$. The remaining part of proof is the same as that in the case $\bar{c} = 1$ and $Ch = 2$.

The probability that the simulator $\mathcal{S}$ outputs $\perp$ is at most $1/3 + \epsilon(n) \leq 1/2$ where ϵ is some negligible function. Additionally, by the above arguments, the distribution of the output of $\mathcal{S}$ conditioned on it is not $\perp$ is statistically close to the distribution of the real transcript. Therefore, we have constructed the simulator and completed the proof. □

Since the protocol is statistically zero knowledge for $t = 1$, it has a witness-indistinguishable property. Witness-indistinguishable property is closed under the parallel composition [8]. Thus, the above protocol is witness indistinguishable for $t = \omega(\log n)$ if a statistically-hiding string commitment scheme is used.

4.2 Security of the Protocol

We show the theorem of the security on our ID protocol, which concerns impersonation under concurrent attack.

Theorem 4.2. *For any $m(n) = \Theta(n \log n)$, there exist $q(n) = O(n^{2.5} \log n)$ and $\gamma(n) = O(n\sqrt{\log n})$ such that $m \geq 10n \log q$ and $q^n / |\mathrm{B}(m, m/2)|$ is negligible in n and the above ID scheme is secure against impersonation under concurrent attack if* GapSVP_γ^2 *is hard in the worst case.*

Before the proof of security, we need to mention the following trivial lemma.

Lemma 4.1. *For any fixed $\mathbf{A}$, let $Y := \{\boldsymbol{y} \in \mathbb{Z}_q^n \mid |\{\boldsymbol{x} \in \mathrm{B}(m, m/2) \mid \mathbf{A}\boldsymbol{x} = \boldsymbol{y}\}| = 1\}$, i.e., a set of vectors $\boldsymbol{y}$ such that the preimage $\boldsymbol{x}$ of $\boldsymbol{y}$ is uniquely determined for $\mathbf{A}$. If $q^n / |\mathrm{B}(m, m/2)|$ is negligible in n, then the probability that, if we obtain $(\boldsymbol{y}, \boldsymbol{x}) \leftarrow \mathsf{KG}(\mathbf{A})$, then $\boldsymbol{y} \in Y$ is negligible in n.*

We now prove Theorem 4.2. The part of the proof is similar to that in [26].

Proof (Proof of Theorem 4.2). Since there exists average-case/worst-case reduction from GapSVP_γ^2 to $\mathrm{SIS}^2_{q,m,\sqrt{m}}$ (Theorem 3.1), we only construct $\mathcal{A}$ solving $\mathrm{SIS}^2_{q,m,\sqrt{m}}$ on the average from an impersonator $\mathcal{I} = (\mathcal{CV}, \mathcal{CP})$ which succeeds impersonation under concurrent attack with non-negligible probability ϵ.

For the clarity, we write the transcript of interaction by (Cmt, Ch, Rsp, Dec). Since the protocol is parallelized, each Cmt, Ch, and Rsp is an ordered list which contains t elements. For example, $Cmt = (Cmt_1, \ldots, Cmt_t)$.

Given $\mathbf{A}$, $\mathcal{A}$ chooses a random secret key $\boldsymbol{x} \in \mathrm{B}(m, m/2)$ and computes $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$. Using the secret key, it can simulate the prover oracle perfectly. $\mathcal{A}$ runs $\mathcal{CV}$ on input $(\mathbf{A}, \boldsymbol{y})$ and obtains a state for $\mathcal{CP}$. $\mathcal{A}$ feeds the state to $\mathcal{CP}$ and acts as a legitimate verifier. Receiving commitments Cmt, $\mathcal{A}$ chooses three challenges $Ch^{(1)}$, $Ch^{(2)}$, and $Ch^{(3)}$ from $\{1, 2, 3\}^t$ uniformly at random. Rewinding with three challenges, $\mathcal{A}$ obtains three transcripts $(Cmt, Ch^{(i)}, Rsp^{(i)}, Dec^{(i)})$ for $i = 1, 2, 3$ as the results of the interactions.

By the Heavy Row Lemma [21], the probability that all $Dec^{(i)}$ are 1 is at least $(\epsilon/2)^3$. Meanwhile, we have

$$\Pr\left[\exists j \in [t] : \{Ch_j^{(1)}, Ch_j^{(2)}, Ch_j^{(3)}\} = \{1,2,3\}\right] = 1 - (7/9)^t$$

by a simple calculation. Thus the probability that $\mathcal{A}$ has three transcripts $(Cmt, Ch^{(i)}, Rsp^{(i)}, Dec^{(i)})$ for $i = 1,2,3$ such that $Dec^{(i)} = 1$ for all i, and $\{Ch_j^{(1)}, Ch_j^{(2)}, Ch_j^{(3)}\} = \{1,2,3\}$ for some $j \in [t]$ is at least $(\epsilon/2)^3 - (7/9)^t$, which is non-negligible since ϵ is non-negligible and $t = \omega(\log n)$.

We next show how $\mathcal{A}$ obtains a secret key or finds a collision of the hash functions in the string commitment scheme by using three good transcripts. Assume that $\mathcal{A}$ has three transcripts $(Cmt^{(i)}, Ch^{(i)}, Rsp^{(i)}, Dec^{(i)})$ for $i = 1,2,3$ such that $Cmt^{(1)} = Cmt^{(2)} = Cmt^{(3)}$, $Dec^{(i)} = 1$ for all i, and $\{Ch_j^{(1)}, Ch_j^{(2)}, Ch_j^{(3)}\} = \{1,2,3\}$ for some $j \in [t]$. Without loss of generality, we assume that $Ch_j^{(i)} = i$. We parse $Rsp_j^{(i)}$ as in Step V2. We have following equations (We omit j for simplification):

$$\begin{aligned}
c_1 &= \mathrm{Com}_{\mathbf{A}}(\phi, \mathbf{A}\boldsymbol{u} - \boldsymbol{y}; \rho_1^{(2)}) &&= \mathrm{Com}_{\mathbf{A}}(\psi, \mathbf{A}\boldsymbol{v}; \rho_1^{(3)}),\\
c_2 &= \mathrm{Com}_{\mathbf{A}}(\boldsymbol{t}; \rho_2^{(1)}) &&= \mathrm{Com}_{\mathbf{A}}(\psi(\boldsymbol{v}); \rho_2^{(3)}),\\
c_3 &= \mathrm{Com}_{\mathbf{A}}(\boldsymbol{s} + \boldsymbol{t}; \rho_3^{(1)}) &&= \mathrm{Com}_{\mathbf{A}}(\phi(\boldsymbol{u}); \rho_3^{(2)}),\\
\boldsymbol{s} &\in \mathrm{B}(m, m/2).
\end{aligned}$$

If there exists a distinct pair of arguments of $\mathrm{Com}_{\mathbf{A}}$, $\mathcal{A}$ obtains a collision for $\mathbf{A}$ and solves $\mathrm{SIS}_{q,m,\sqrt{m}}$.

Next, we suppose that there exist no distinct pairs of the arguments of $\mathrm{Com}_{\mathbf{A}}$. Let π denote the inverse permutation of ϕ. From the first equation, we have $\pi^{-1} = \phi = \psi$. Thus, we obtain $\boldsymbol{u} = \pi(\boldsymbol{s} + \boldsymbol{t})$ from the third equation. Combining it with the first equation, we have $\mathbf{A}\boldsymbol{v} = \mathbf{A}(\pi(\boldsymbol{s}) + \pi(\boldsymbol{t})) - \boldsymbol{y}$. Since $\boldsymbol{v} = \phi^{-1}(\boldsymbol{t}) = \pi(\boldsymbol{t})$ from the second equation, we obtain $\boldsymbol{y} = \mathbf{A} \cdot \pi(\boldsymbol{s})$. Since $\boldsymbol{s} \in \mathrm{B}(m, m/2)$, so $\pi(\boldsymbol{s})$ also is in $\mathrm{B}(m, m/2)$. Therefore, $\mathcal{A}$ sets $\boldsymbol{x}' := \pi(\boldsymbol{s})$.

We now have to show that $\boldsymbol{x}' \neq \boldsymbol{x}$ with probability at least $1/2$. By Lemma 4.1, there must be another secret key $\boldsymbol{x}'$ corresponding to $\boldsymbol{y}$ with overwhelming probability. Recall that the protocol is statistically witness indistinguishable. Hence, $\mathcal{I}$'s view is independent of $\mathcal{A}$'s choice of $\boldsymbol{x}$ with overwhelming probability. Thus we have $\boldsymbol{x}' \neq \boldsymbol{x}$ with probability at least $1/2$. In this case $\mathcal{A}$ outputs $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{x}'$ and solves $\mathrm{SIS}_{q,m,\sqrt{m}}$. □

We note that the above proof is extended into multi-user settings as in the proof of Lyubashevsky [14].

5 An Ad Hoc Anonymous Identification Scheme

We next construct our AID scheme based on GapSVP. First, we sketch a basic idea for our construction: Let $\mathbf{A}$ be a system parameter. Each user has a secret key $\boldsymbol{x}_i \in \mathrm{B}(m, w)$ and a public key $\boldsymbol{y}_i = \mathbf{A}\boldsymbol{x}_i$. In the AID scheme, a group is specified by a set of public keys $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l)$ of the members. Let $\boldsymbol{e}_{i,l}$ denote an l-dimensional vector ${}^t(0, \ldots, 0, 1, 0, \ldots, 0)$ whose i-th element is 1. The prover in the group, who has a secret key $\boldsymbol{x}_i$, wants convinces the verifier that he/she knows that $\boldsymbol{x}' := \boldsymbol{x}_i \circ -\boldsymbol{e}_{i,l}$ such that $[\mathbf{A}\, \boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_l]\boldsymbol{x}' = \mathbf{0}$

and $\boldsymbol{x}_i \in \mathrm{B}(m, m/2)$. Changing the parameters and using Stern's protocol, the prover can convinces the verifier that he/she has $\boldsymbol{x}'$ such that $[\mathbf{A}\,\boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_l]\boldsymbol{x}' = \mathbf{0}$, the numbers of $+1$ in $\boldsymbol{x}'$ is $m/2$, and the numbers of -1 in $\boldsymbol{x}'$ is 1. Additionally, we force the prover to prove that $\boldsymbol{x}'$ is in the form $\boldsymbol{x}' = \boldsymbol{x}_i \circ -\boldsymbol{e}_{i,l}$. To do so, we divide a permutation π in Step P1 into two permutations.

Let π_h be a permutation over $[m]$ and π_t be a permutation over $[l]$. For a permutation π over $[m+l]$, we denote $\pi = \pi_h \odot \pi_t$ if

$$\pi = \begin{pmatrix} 1 & 2 & \cdots & m \\ \pi_h(1) & \pi_h(2) & \cdots & \pi_h(m) \end{pmatrix} \cdot \begin{pmatrix} m+1 & m+2 & \cdots & m+l \\ m+\pi_t(1) & m+\pi_t(2) & \cdots & m+\pi_t(l) \end{pmatrix}.$$

For any π_h and π_t, we have $(\pi_h \odot \pi_t)^{-1} = \pi_h^{-1} \odot \pi_t^{-1}$. For any $\boldsymbol{x}_h \in \mathbb{Z}^m$ and $\boldsymbol{x}_t \in \mathbb{Z}^l$, if $\pi = \pi_h \odot \pi_t$ then $\pi(\boldsymbol{x}_h \circ \boldsymbol{x}_t) = \pi_h(\boldsymbol{x}_h) \circ \pi_t(\boldsymbol{x}_t)$.

We here construct an AID scheme based on GapSVP. Similarly to the ID scheme in Section 4, the protocol is repeated $t = \omega(\log n)$ times in parallel to achieve exponentially small soundness error. As in the previous section, we hide randomness in $\mathrm{Com}_{\mathbf{A}}$.

SetUp: Same as SetUp of the protocol in Section 4.
Reg: Same as KG of the protocol in Section 4.
P, V: The common inputs are $\mathbf{A}$ and $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l)$. The prover's auxiliary input is $\boldsymbol{x}_i$ for some $i \in [l]$. Let $\mathbf{A}' := [\mathbf{A}\,\boldsymbol{y}_1 \ \ldots \ \boldsymbol{y}_l]$ and $\boldsymbol{x} := \boldsymbol{x}_i \circ -\boldsymbol{e}_{i.l}$. We write Com instead of $\mathrm{Com}_{\mathbf{A}}$ for ease of notation. They interact as follows:

Step P1: Choose random permutations π_h over $[m]$ and π_t over $[l]$. Let $\pi = \pi_h \odot \pi_t$. Choose a random vector $\boldsymbol{r} \in \mathbb{Z}_q^{m+l}$. Send commitments c_1, c_2, and c_3 as
- $c_1 = \mathrm{Com}(\pi_h, \pi_t, \mathbf{A}'\boldsymbol{r})$,
- $c_2 = \mathrm{Com}(\pi(\boldsymbol{r}))$,
- $c_3 = \mathrm{Com}(\pi(\boldsymbol{x} + \boldsymbol{r}))$.

Step V1 Send a random challenge $Ch \in \{1, 2, 3\}$ to P.

Step P2
- If $Ch = 1$, reveal c_2 and c_3. Send $\boldsymbol{s} = \pi(\boldsymbol{x})$ and $\boldsymbol{t} = \pi(\boldsymbol{r})$.
- If $Ch = 2$, reveal c_1 and c_2. Send $\phi_h = \pi_h$, $\phi_t = \pi_t$, and $\boldsymbol{u} = \boldsymbol{x} + \boldsymbol{r}$.
- If $Ch = 3$, reveal c_1 and c_3. Send $\psi_h = \pi_h$, $\psi_t = \pi_t$, and $\boldsymbol{v} = \boldsymbol{r}$.

Step V2
- If $Ch = 1$, check that $c_2 = \mathrm{Com}(\boldsymbol{t})$, $c_3 = \mathrm{Com}(\boldsymbol{s} + \boldsymbol{t})$, and $\boldsymbol{s}$ is in the form $\boldsymbol{s}_h \circ -\boldsymbol{e}_{j,l}$ for some j and $\boldsymbol{s}_h \in \mathrm{B}(m, m/2)$.
- If $Ch = 2$, check that $c_1 = \mathrm{Com}(\phi_h, \phi_t, \mathbf{A}'\boldsymbol{u})$ and $c_3 = \mathrm{Com}((\phi_h \odot \phi_t)(\boldsymbol{u}))$.
- If $Ch = 3$, check that $c_1 = \mathrm{Com}(\psi_h, \psi_t, \mathbf{A}')$ and $c_2 = \mathrm{Com}((\psi_h \odot \psi_t)(\boldsymbol{v}))$.

Output $Dec = 1$ if all checks are passed, otherwise output $Dec = 0$.

The security of the above protocol is stated as follows. We omit the proof, since it is similar to the proof of Theorem 4.2.

Theorem 5.1. *Let $m = m(n)$ and $q = q(n)$ be polynomially bounded functions satisfying the conditions that $m \geq 10n \log q$ and $q^n / |\mathrm{B}(m, m/2)|$ is negligible in n. Assume that there exists an impersonator $\mathcal{I}$ that succeeds impersonation under concurrent chosen-group attack with non-negligible probability. Then there exists a probabilistic polynomial-time algorithm $\mathcal{A}$ that solves $\mathrm{SIS}^2_{q,m,\sqrt{m}}$.*

Combining Theorem 5.1 with Theorem 3.1, we obtain the following theorem.

Theorem 5.2. *For any* $m(n) = \Theta(n \log n)$*, there exist* $q(n) = O(n^{2.5} \log n)$ *and* $\gamma(n) = O(n\sqrt{\log n})$ *such that* $q^n / |\mathrm{B}(m, m/2)|$ *is negligible in n and the above scheme is secure against impersonation under concurrent chosen-group attack if* GapSVP_γ^2 *is hard in the worst case.*

The statistical anonymity of the above scheme follows from witness indistinguishability of the protocol.

Acknowledgement

The third author thanks Eiichiro Fujisaki for his inspiring question, which motivated us to combine Stern's protocol with lattice-based hash functions. We thank the anonymous referees for their helpful comments and their suggestions on editorial problems. This work is partly supported by Grant-in-Aid for JSPS Fellows 19-55201, and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientist (B) No.17700007, 2005 and for Scientific Research (B) No.18300002, 2006.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: STOC 1996, pp. 99–108 (1996)
2. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: STOC 1997, pp. 284–293 (1997)
3. Bellare, M., Palacio, A.: GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002)
4. Damgård, I.B., Pedersen, T.P., Pfizmann, B.: On the existence of statistically hiding bit commitment schemes and fail-stop signatures. Journal of Cryptology 10(3), 163–194 (1997)
5. Damgård, I.B., Pedersen, T.P., Pfizmann, B.: Statistical Secrecy and Multibit Commitments. IEEE Transactions on Information Theory 44(3), 1143–1151 (1998)
6. De Santis, A., Di Crescenzo, G., Persiano, G., Yung, M.: On monotone formula closure of SZK. In: FOCS 1994, pp. 454–465 (1994)
7. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 609–626. Springer, Heidelberg (2004)
8. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC 1990, pp. 416–426 (1990)
9. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206 (2008)
10. Goldreich, O.: Foundations of Cryptography: Volume I – Basic Tools. Cambridge University Press, Cambridge (2001)
11. Goldreich, O., Goldwasser, S., Halevi, S.: Collision-free hashing from lattice problems. ECCC 3(42) (1996)
12. Halevi, S., Micali, S.: Practical and provably-secure commitment scheme from collision-free hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 201–215. Springer, Heidelberg (1996)
13. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/CRC (2007)

14. Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 162–179. Springer, Heidelberg (2008)
15. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
16. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFT: A modest proposal for FFT hashing. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
17. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Computational Complexity 16, 365–411 (2007)
18. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: a cryptographic perspective. Kluwer Academic Publishers, Dordrecht (2002)
19. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measures. SIAM Journal on Computing 37(1), 267–302 (2007)
20. Micciancio, D., Vadhan, S.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer, Heidelberg (2003)
21. Ohta, K., Okamoto, T.: On concrete security treatment of signatures derived from identification. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 354–369. Springer, Heidelberg (1998)
22. Peikert, C.: Limits on the hardness of lattice problems in l_p norms. Computational Complexity 17(2), 300–351 (2008)
23. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (2006)
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)
25. Shamir, A.: A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. IEEE Transactions on Information Theory 30(5), 699–704 (1984)
26. Stern, J.: A new paradigm for public key identification. IEEE Transactions on Information Theory 42(6), 749–765 (1996)
27. Wu, Q., Chen, X., Wang, C., Wang, Y.: Shared-key signature and its application to anonymous authentication in ad hoc group. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 330–341. Springer, Heidelberg (2004)