ISSN 1342-2812

Research Reports on Mathematical and Computing Sciences

Concurrently Secure Identification Schemes
and Ad Hoc Anonymous Identification Schemes
Based on the Worst-Case Hardness of Lattice Problems

Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa

November 2007, C–249

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES C: Computer Science

Concurrently Secure Identification Schemes and Ad Hoc Anonymous 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
W8-55, 2-12-1 Ookayama Meguro-ku, Tokyo 152-8552, Japan
{kawachi, keisuke, xagawa5}@is.titech.ac.jp

November 21, 2007

Abstract

We present a direct construction for an identification scheme provably secure against concurrent attacks under the assumptions on the *worst-case* hardness of hard lattice problems such as the gap version of the Shortest Vector Problem. We also construct an ad hoc anonymous identification scheme based on the lattice problems by modifying this direct construction.

Keywords: lattice-based cryptography, identification schemes, concurrent security.

1 Introduction

Background: Many researchers have so far developed cryptographic schemes based on combinatorial problems related to knapsacks [22, 35], codes [37, 39, 32], and lattices [2, 3, 14, 16], due to the intractability of the underlying problems and the efficiency of primitive operations. The combinatorial problems have attracted more attention from cryptographic point of view since Shor's quantum algorithms [38] revealed the threat of quantum computers to number-theoretic schemes.

The cryptographic schemes employ combinatorial problems computationally intractable in the *worst case* such as NP-hard problems in order to guarantee their security. They usually assume the *average-case* hardness of the underlying problem because they have to deal with randomly generated cryptographic instances like keys, plaintexts, and ciphertexts. This implies a security risk in such schemes since it is generally hard to show the average-case hardness. In fact, several attacks against such schemes were found in the practical settings [36, 19, 7, 28]. The cryptographic schemes only with the security based 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 average-case hardness. Ajtai [2] showed that the average-case hardness of some lattice problem is equivalent to its worst-case hardness. His seminal result opened the way to the cryptographic schemes based on the worst-case hardness of lattice problems. The public-key encryption schemes were proposed by Ajtai and Dwork [3], Regev [33, 34] and Peikert and Waters [31], and hash functions by Ajtai [2] and its improvements by [13, 6, 23, 26]. The public-key schemes based only on the average-case hardness of lattice problems were also proposed by Goldreich, Goldwasser, and Halevi [14] and Hoffstein, Jeffrey, and Pipher [17].

Among many varieties of lattice-based public-key schemes, there are very few results on the identification (ID) schemes based on the worst-case hardness of lattice problems. As a major result, 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 provers [27].

In general, it is much harder to prove strong security against active attacks directly from the worst-case hardness of lattice problems than standard number-theoretic ones. For example, one of the long-standing open problems was the construction for a lattice-based public-key encryption scheme secure against the chosen ciphertext attack based on the worst-case hardness, which was most recently resolved by Peikert and Waters [31]. Except for their result, there is no direct construction for lattice-based cryptographic schemes secure against active attacks under the worst-case hardness assumption.

Our results: In this paper, we propose a *direct* construction for lattice-based ID scheme *concurrently* secure under the assumption on the *worst-case* hardness of the gap version of the Shortest Vector Problem (GapSVP) with approximation factor $O(n\sqrt{\log n})$ ($\mathrm{GapSVP}^2_{O(n\log^{1/2} n)}$) and the Shortest Vector Problem (SVP) for ideal lattices with approximation factor $O(n\log^3 n)$ ($\Lambda(f)$-$\mathrm{SVP}^\infty_{O(n\log^3 n)}$).

The gap version of the Shortest Vector Problem GapSVP has already been used in the known lattice-based cryptographic constructions [27, 26]. The SVP for ideal lattices $\Lambda(f)$-SVP was recently proposed by Lyubashevsky and Micciancio [20] to improve the efficiency of lattice-based schemes. They actually constructed compact hash functions using this problem. The efficiency of our ID scheme can be also improved by using SVP for ideal lattices.

Our ID scheme basically follows the same framework as Stern's [39]. He presented a statistical zero-knowledge argument secure under the assumption of the average-case hardness of Syndrome Decoding Problem (or a generalized version called Modular Knapsack Problem) and the existence of a collision resistant hash function. Changing the assumption from the average-case hardness of such combinatorial problems to the worst-case hardness, we successfully remove the assumption on the collision resistant hash functions since we can build all the primitives we need from the lattice problems.

Actually, we can construct a concurrently secure ID scheme based on the worst-case hardness of lattice problems from Micciancio and Vadhan's ID scheme (the MV scheme, for short) [27]. Applying a general technique by Feige and Shamir [12], a modification of the MV scheme can be proven to have concurrent security [1]. On the contrary, our construction does not depend on the general modification technique, and the security can be proven directly from the worst-case assumption.

Moreover, our direct construction yields efficient schemes for ad hoc anonymous identification (AID) based on the worst-case hardness of $\mathrm{GapSVP}^2_{O(n^2\log^{1/2} n)}$ and $\Lambda(f)$-$\mathrm{SVP}^\infty_{O(n^2\log^4 n)}$, which are secure against the concurrent chosen-group attack. The AID scheme was originally proposed and formulated by Dodis, Kiayias, Nicolosi, and Shoup [11]. The protocol is done by two parties, a prover and verifier, but we implicitly suppose a group that is made ad hoc. 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. Dodis et al. presented a general construction of AID scheme from any accumulator with one-way domain and showed that constant-size signer-ambiguous group and ring signatures can be obtained from AID schemes by using the Fiat-Shamir transformation.

Again by a simple modification of the MV scheme, we can obtain an AID scheme based on the worst-case hardness of lattice problems. Unfortunately, the simple modification 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 let n be the security parameter. In the simple modification, the size of the public keys is $l\cdot\tilde{O}(n^2)$ and the communication cost for single execution is $l\cdot\tilde{O}(n)$ in the modified version of the GapSVP-based MV scheme, where $\tilde{O}(f(n)) = O(f(n)\,\mathrm{poly}\log f(n))$ for a function f in n.

In contrast to the simple modification, the size of the public keys is $\tilde{O}(n^2)+l\cdot\tilde{O}(n)$ and the communication cost for a single execution $\tilde{O}(n+l)$ in our GapSVP-based AID scheme, which improves the efficiency of the simple modification.

[1] Feige and Shamir [12] actually showed a general construction technique for ID schemes secure against active attacks. But, by Bellare and Pracio's observation [4], we can construct concurrently secure ID schemes by the same technique.

We also modify the above concurrently secure ID scheme into our AID scheme based on a similar strategy to Wu, Chen, Wang, and Wang's [40], which gave an AID scheme secure under the average-case hardness of the Weak Dependence Problem. We formally define a concurrent version of the security notion, the security against impersonation under concurrent chosen-group attacks, and prove that our AID scheme has this security notion.

As suggested in [11], we can also obtain ring signature schemes secure in the random oracle model under the our assumptions by applying the Fiat-Shamir transform to our AID schemes.

Overview of our constructions: We first construct key-generation algorithms and string commitment schemes based on the lattice problems. Then, plugging these algorithms and schemes into the standard structure, we obtain the ID schemes based on the worst-case hardness of the underlying problems.

Our construction has a standard structure of 3-move public-coin protocols, and can be considered as a modification of Stern's zero-knowledge argument protocol [39]. As already mentioned, the assumption of his protocol is the average-case hardness of certain combinatorial problems. Also, his protocol has no explicit string commitment protocol constructed from the assumption. To obtain the security proof of our scheme only from the worst-case assumption of the lattice problems, we prepare two components; one is a string commitment scheme and the other is a key-generation algorithm.

These two components consist of the same ingredient, collision-resistant hash functions based on GapSVP and SVP for ideal lattices. These functions were introduced by Micciancio and Regev [26] and Lyubashevsky and Micciancio [20], respectively.

We construct string commitment schemes from these functions. General constructions of string commitment schemes from collision resistant hash functions were shown by Damgård, Pedersen, and Pfizmann [9, 10] and Halevi and Micali [15]. Our constructions for string commitment schemes are more direct and simpler than the general one.

A key-generation algorithm for an ID scheme generates public and secret keys of a prover. We implement this algorithm from the collision-resistant hash functions of [26] and [20] by appropriately adjusting their parameters for our security proofs.

Plugging these algorithms and schemes into Stern's structure, we obtain ID schemes based on the worst-case hardness of the underlying problems. The important point for our concurrent security is the format of the common inputs which consist of a system parameter and a public key and the relation between public and secret keys the prover wants to prove. We now compare the MV scheme with ours to briefly describe why our scheme has concurrent security.

In the MV scheme, a prover and verifier are given a matrix $\mathbf{A}$ as a common input, and a prover has a binary vector $\mathbf{x}$ as secret information. The task of the prover is to convince the verifier that he/she knows $\mathbf{x}$ satisfying the relation that $\mathbf{Ax} = \mathbf{0}$ and $\mathbf{x}$ is relatively short. In our scheme, a prover has a binary vector $\mathbf{x}$ with a fixed Hamming weight as his/her secret key. We also feed to the prover and verifier a matrix $\mathbf{A}$ as a system parameter and a vector $\mathbf{y}$ as the corresponding public key $\mathbf{y}$ to $\mathbf{x}$. The task of the prover is to convince the verifier that he/she knows a correct secret key $\mathbf{x}$ satisfying a relation $\mathbf{Ax} = \mathbf{y}$ in our case.

To show concurrent security of an ID scheme, we usually give a reduction from the concurrent attack for the scheme to the underlying problem. That is, by using an adversary capable of the concurrent attack for the scheme, we construct an efficient algorithm for the underlying problem. It should be noted that the algorithm needs to simulate the behaviour of the prover since the algorithm must answer the queries that the adversary makes in its concurrent attack in order to run the adversary correctly.

In both cases of ours and the MV scheme, the underlying problem is reduced to Small Integer Solution Problem (SIS). Given a matrix $\mathbf{A}$, the task of SIS is to find a relatively short vector $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{0}$.

In our reduction, when we are given $\mathbf{A}$, by generating just a dummy secret key $\mathbf{x}'$, we can simulate the prover with $\mathbf{A}$ and $\mathbf{x}'$ the adversary concurrently accesses. On the other hand, in the case of the MV scheme, it looks difficult to directly simulate the prover since we have to prepare a dummy short vector $\mathbf{x}'$ satisfying $\mathbf{Ax}' = \mathbf{0}$ for the simulation, but that is the task of SIS itself.

Our construction for AID schemes also has a similar structure. Each of l members in the ad hoc group

has a vector $\mathbf{x}_i$ ($i = 1, \ldots, l$). Then, the common inputs of the scheme are a system parameter $\mathbf{A}$ and a set of public keys $\mathbf{y}_1, \ldots, \mathbf{y}_l$ of the members, which satisfy $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$ ($i = 1, \ldots, , l$). We can show that the prover can anonymously convince the verifier that the prover knows $\mathbf{x}_i$ corresponding to one of $\mathbf{y}_1, \ldots, \mathbf{y}_l$ based on a similar argument to the proof of our concurrently secure ID scheme.

As mentioned above, we can construct a lattice-based AID scheme in a straightforward manner from the MV scheme. We just feed $\mathbf{A}_1, ..., \mathbf{A}_l$ as the common inputs to the prover and verifier. In this case, the prover convinces the verifier that he/she has a short vector $\mathbf{x}_i$ satisfying $\mathbf{A}_i\mathbf{x}_i = \mathbf{0}$ for some i.

While our scheme requires many *vectors* proportional to the size of the group, this straightforward scheme requires many *matrices* proportional to the size of the group, which shows the advantage of our scheme on the efficiency. Moreover, in contrast to our AID scheme, it seems difficult to prove that this straightforward scheme is secure against impersonation under concurrently chosen-group attacks.

Organization: The rest of this paper is organized as follows. We introduce basic notations and notions, and review the cryptographic schemes we consider in this paper in Section 2. In Section 3, we give a key-generation algorithm and a commitment scheme based on the Micciancio-Regev 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 key-generation algorithm and string commitment scheme. We present the AID scheme in Section 5.

In this extended abstract, due to lack of space, we only describe the schemes based on GapSVP since the construction from SVP for ideal lattices follows a similar strategy to that from GapSVP. We argue the constructions from SVP for ideal lattices in Appendix A.

2 Preliminaries

Basic notions and notations: We define a negligible amount in n as an amount that is asymptotically smaller than n^{-c} for any constant $c > 0$. More formally, $f(n)$ is a negligible function in n if $\lim_{n\to\infty} n^c f(n) = 0$ for any $c > 0$. Similarly, a non-negligible amount is one which is at least n^{-c} for some $c > 0$. 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 a non-negligible probability. We sometimes use $\tilde{O}(f(n))$ for any function f in n as $O(f(n)\cdot\mathrm{polylog}(f(n)))$. We assume that all random variables are independent and uniform if not specified. We denote by n the security parameter of cryptographic schemes throughout this paper, which corresponds to the rank of the underlying lattice problems.

For any $p \geq 1$, the l_p norm of a vector $\mathbf{x} = {}^t(x_1, \ldots, x_n) \in \mathbb{R}^n$, denoted by $\|\mathbf{x}\|_p$, is $(\sum_{i=1}^n x_i^p)^{1/p}$. For ease of notation, we define $\|\mathbf{x}\| := \|\mathbf{x}\|_2$. The infty norm is defined as $\|\mathbf{x}\|_\infty = \lim_{p\to\infty} \|\mathbf{x}\|_p = \max_i |x_i|$.

Let $w_H(\mathbf{x})$ denote the Hamming weight of $\mathbf{x}$, i.e., the number of nonzero elements in $\mathbf{x}$. 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., $\mathrm{B}(m, w) := \{\mathbf{x} \in \{0,1\}^m \mid w_H(\mathbf{x}) = w\}$.

Given two probability density functions ϕ_1 and ϕ_2 on a finite set S, we define the statistical distance between them as $\Delta(\phi_1, \phi_2) := \frac{1}{2}\sum_{x\in S} |\phi_1(x) - \phi_2(x)|$. We also use the same notation for two arbitrary functions. Note that the acceptance probability of any algorithm on inputs from X differs from its acceptance probability on inputs from Y by at most $\Delta(X, Y)$.

If $A(\cdot, \cdot, \ldots)$ is a randomized algorithm, then $y \leftarrow A(x_1, x_2, \ldots; r)$ means that y is assigned the unique output of the algorithm on inputs $x_1, x_2, \ldots$ and coins r. We often use the notation $y \leftarrow A(x_1, x_2, \ldots)$ as shorthand for first picking r at random and then setting $y \leftarrow A(x_1, x_2, \ldots, ; r)$. If S is a finite set then $s \leftarrow_R S$ indicates that s is chosen uniformly at random from S.

Provers and verifiers: An interactive algorithm A is a stateful algorithm that on input an incoming message M_{in} and state information St outputs an outgoing message M_{out} and updated state St' (We will write it as $(M_{out}, St') \leftarrow \mathsf{A}(M_{in}, St)$).

We say that A accepts if $St = 1$ and rejects if $St = 0$. An interaction between a prover P and a verifier V ends when V either accepts or rejects. We will write

$$(Tr, Dec) \leftarrow \mathbf{Run}[\mathsf{P}(p_1, \ldots)^{OP_1,\cdots} \leftrightarrow \mathsf{V}(v_1, \ldots)^{OV_1,\cdots}]$$

to indicate that we let P interact with V, having provided both P and V with fresh random coins, to get a transcript *Tr* and a boolean decision *Dec*.

Hash functions: We here define collision-resistant hash functions family.

Definition 2.1. Let $\mathcal{H}_n = \{h_k : M_n \to D_n \mid k \in K_n\}$ be a family of hash functions. Let $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$. Let $\mathcal{A}$ be an adversary. We define the following experiments of $\mathcal{A}$ for the collision-resistant property of a hash function.

$\mathbf{Exp}^{\mathrm{col}}_{\mathcal{H},\mathcal{A}}(n)$**:** $k \leftarrow_R K_n$; $(x, x') \leftarrow \mathcal{A}(1^n, k)$;
If $x', x \in M_n$, $x \neq x'$, and $h_k(x) = h_k(x')$ then return 1. Otherwise, return 0.

Let the advantage of $\mathcal{A}$ be $\mathbf{Adv}^{\mathrm{col}}_{\mathcal{H},\mathcal{A}}(n) = \Pr\left[\mathbf{Exp}^{\mathrm{col}}_{\mathcal{H},\mathcal{A}}(n) = 1\right]$. We say that $\mathcal{H}$ is collision resistant if, for any probabilistic polynomial-time adversary $\mathcal{A}$, $\mathbf{Adv}^{\mathrm{col}}_{\mathcal{H},\mathcal{A}}(n)$ is negligible in n.

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 a sender and receiver. In our case, the party $\mathcal{T}$ distributes a description of a commitment function randomly chosen from a family of commitment functions as the system parameter.

Let $C_n = \{\mathrm{Com}_k : M_n \times R_n \to C_n \mid k \in K_n\}$ be a family of commitment functions and let $C = \{C_n\}_{n\in\mathbb{N}}$. First, $\mathcal{T}$ on input 1^n distributes the system parameter $k \in K_n$ to a sender and receiver. Both parties then share a common function by a given k. After sharing the function Com_k, the scheme executes two phases, called committing and revealing phases. In the committing phase, the sender commits his/her decision, say, a string $s \in M_n$ to a commitment string $c = \mathrm{Com}_k(s;\rho)$ with a random string $\rho \in R_n$. He/She then sends the commitment string c to the receiver. In the revealing phase, the receiver verifies the sender's decision s in the committing phase. To do so, the sender gives the receiver the decision s and the random string ρ. The receiver can then easily verify the validity of c by computing $\mathrm{Com}_k(s;\rho)$. The security notion of the string commitment schemes we require can be formalized as follows:

Definition 2.2. We say a string commitment scheme C is statistically hiding and computationally binding if it has the following properties:

Statistical-hiding property:
For any two strings $s, s' \in M_n$, the statistical distance between $\mathrm{Com}_k(s;\rho)$ and $\mathrm{Com}_k(s';\rho')$ is negligible in n for random strings ρ and ρ'.

Computational-binding property:
Let $\mathcal{A}$ be an adversary. We consider the following experiment of $\mathcal{A}$:

$\mathbf{Exp}^{\mathrm{bd}}_{C,\mathcal{A}}(n)$**:** $k \leftarrow_R K_n$; $((s,\rho),(s',\rho')) \leftarrow \mathcal{A}(1^n, k)$;
If $s, s' \in M_n$, $s \neq s'$, and $\mathrm{Com}_k(s;\rho) = \mathrm{Com}_k(s';\rho)$ then return 1.
Otherwise, return 0.

We define the advantage of $\mathcal{A}$ as $\mathbf{Adv}^{\mathrm{bd}}_{C,\mathcal{A}}(n) = \Pr\left[\mathbf{Exp}^{\mathrm{bd}}_{C,\mathcal{A}}(n) = 1\right]$. Then, $\mathbf{Adv}^{\mathrm{bd}}_{C,\mathcal{A}}(n)$ is negligible in n for any probabilistic polynomial-time adversary $\mathcal{A}$.

Intuitively, if C is statistically hiding, any computationally unbounded adversarial receiver cannot distinguish two commitment strings generated from two distinct strings. Also, it is computationally hiding, any polynomial-time adversarial sender cannot change the committed string after sending the commitment.

Canonical identification schemes: We adopt the definition of identification schemes given in [1]. 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.

Security against impersonation under concurrent attacks: We are interested in concurrent attacks, which are stronger than active and passive attacks. So, we review the definition of concurrent security in [4].

In concurrent attacks, the adversary would play the role of a cheating verifier prior to impersonation, but could interact many different prover "clones" concurrently. Each clone has the same secret key, but has independent random coins and maintain its own state.

Let an impersonator $\mathcal{I} = (\mathsf{CV}, \mathsf{CP})$ be a pair of probabilistic polynomial-time algorithms, the cheating verifier and cheating prover. CV would interacts with each of clones, which is identified by a session ID *s*.

We describe the formal definition as follows. Consider the experiment $\mathbf{Exp}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}(n)$ between the challenger and the impersonator $\mathcal{I} = (\mathsf{CV}, \mathsf{CP})$.

Experiment $\mathbf{Exp}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}(n)$: (See also Table 1 in Appendix B.)

Setup Phase: The challenger obtains $param \leftarrow \mathsf{SetUp}(1^n)$. Next, it obtains $(pk, sk) \leftarrow \mathsf{KG}(param)$ and sets $PS := \emptyset$, where PS denotes the set of prover's sessions. The impersonator CV is given the system parameter *param*.

Learning Phase: The impersonator CV can query to the prover oracle $\textsc{Prov}$.

- The oracle $\textsc{Prov}$ receives inputs s, M_{in}. If $s \notin PS$ then it adds s to PS, pick a random coin ρ, and sets a state of the prover $St_{\mathsf{P}}[s] := (param, sk, \rho)$. Next, it obtains $(M_{out}, St_{\mathsf{P}}[s]) \leftarrow \mathsf{P}(M_{in}, St_{\mathsf{P}}[s])$. It returns M_{out}.

Challenge Phase: CV outputs St_{CP}. The challenger gives St_{CP} to CP. Finally, the challenger obtains $(Tr, Dec) \leftarrow \mathbf{Run}[\mathsf{CP}(St_{\mathsf{CP}}) \leftrightarrow \mathsf{V}(param, pk)]$ and returns *Dec*.

Definition 2.3. Let $\mathcal{SI} = (\mathsf{SetUp}, \mathsf{KG}, \mathsf{P}, \mathsf{V})$ be an ID scheme, $\mathcal{I} = (\mathsf{CV}, \mathsf{CP})$ an impersonator, and n a security parameter. We define the advantage of $\mathcal{I}$ as $\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}(n) = \Pr\left[\mathbf{Exp}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}(n) = 1\right]$. We say that $\mathcal{SI}$ is secure against impersonation under concurrent attacks if $\mathbf{Adv}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}(\cdot)$ is negligible for every polynomial-time $\mathcal{I}$.

Ad hoc anonymous identification schemes: An AID scheme [11] allows an 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 a group manager. We then assume that every user registers his/her public key to the public key infrastructure.

An ad hoc anonymous identification (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* output (pk, sk), P is the prover algorithm taking inputs *param*, a set of public keys $R = (pk_1, \ldots, pk_l)$, and one of secret key sk_i such that $pk_i \in R$, V is the verifier algorithm taking inputs *param* and *R*. We omit the group public key construction and group secret key construction algorithms in the definition of [11] to simplify notations.

There are two goals for security of AID schemes: security against impersonation and anonymity.

Security against impersonation under concurrent chosen-group attacks: In the setting of chosen-group attacks, an adversary could force the prover to prove the membership in an arbitrary group if the prover is indeed a member of the group. "Concurrent" attacks allow the cheating verifier and prover to

interact with the clones of provers except for target provers, whereas only the cheating verifier can interact with the clones of a prover in Definition 2.3.

We describe the formal definition of the security as follows. Consider the following experiment $\mathbf{Exp}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}(n)$ between a challenger and the impersonator $\mathcal{I} = (\mathsf{CV}, \mathsf{CP})$.

Experiment $\mathbf{Exp}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}(n)$: (See also Table 2 in Appendix B.)

Setup Phase: The challenger obtains $param \leftarrow \mathsf{SetUp}(1^n)$ and initializes $HU, CU, AU, PS := \emptyset$, where HU, CU, and TU denote the sets of honest users, corrupted users, and target users, respectively, and PS denotes the set of prover's session. The impersonator CV is given the system parameter $param$.

Learning Phase: The impersonator CV can query to the three oracles Init, Corr, and Prov.

- The oracle Init receives input i. If $i \in HU \cup CU \cup TU$ then returns $\perp$. Otherwise, it obtains $(pk_i, sk_i) \leftarrow \mathsf{Reg}(param; \rho_i)$, adds i to HU, and provides $\mathcal{I}$ with pk_i.
- The oracle Corr receives input i. If $i \notin HU \setminus TU$ then returns $\perp$. Otherwise, it adds i to CU, deletes i in HU, and returns ρ_i to $\mathcal{I}$.
- The oracle Prov receives inputs R, i, s, and M_{in}. If $pk_i \notin R$ or $i \notin HU \setminus TU$ then returns $\perp$. (Note that the public keys in R need not to be registered.) If $(R, i, s) \notin PS$ then it adds (R, i, s) to PS, pick a random coin ρ, and sets a state of the prover $St_{\mathsf{P}}[(R, i, s)] := (param, R, sk_i, \rho)$. Next, it obtains $(M_{out}, St_{\mathsf{P}}[(R, i, s)]) \leftarrow \mathsf{P}(M_{in}, St_{\mathsf{P}}[(R, i, s)])$. It returns M_{out}.

Challenge Phase: CV outputs a set of public keys $R_t = (pk_{i_1}, \ldots, pk_{i_l})$ and St_{CP}. If $R \not\subseteq HU$ then the challenger outputs 0 and halts. Otherwise, the challenger sets $TU := R_t$ and gives St_{CP} to CP. CP can query to the oracles Init, Corr, and Prov as in the learning phase. Finally, the challenger obtains $(Tr, Dec) \leftarrow \mathbf{Run}[\mathsf{CP}(St_{\mathsf{CP}})^{\text{Init,Corr,Prov}} \leftrightarrow \mathsf{V}(param, R_t)]$ and outputs Dec.

Definition 2.4. Let $\mathcal{AID} = (\mathsf{SetUp}, \mathsf{Reg}, \mathsf{P}, \mathsf{V})$ be an AID scheme and $\mathcal{I} = (\mathsf{CV}, \mathsf{CP})$ an impersonator. Let n be a security parameter. The advantage of $\mathcal{I}$ in attacking $\mathcal{AID}$ is defined by

$$\mathbf{Adv}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}(n) := \Pr\left[\mathbf{Exp}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}(n) = 1\right].$$

We say that $\mathcal{AID}$ is secure against impersonation under concurrent chosen-group attacks if $\mathbf{Adv}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}(\cdot)$ is negligible for every polynomial-time $\mathcal{I}$.

We note that our definition is the concurrent version of the soundness definition in [11].

Anonymity against full key exposure: This security notion captures the property that an adversary cannot distinguish two transcripts even if the adversary has secret keys of all the members. Anonymity against full key exposure for an AID scheme $\mathcal{AID}$ is defined by using the following experiment $\mathbf{Exp}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}(n)$ between a challenger and adversary $\mathcal{A}$:

Experiment $\mathbf{Exp}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}(n)$: (See also Table 3 in Appendix B.)

Setup Phase: The challenger runs the algorithm SetUp with input 1^n and obtains $param$. The adversary $\mathcal{A}$ is given the system parameter $param$.

Challenge Phase: $\mathcal{A}$ requests a challenge by sending to the challenger the values $((pk_{i_0}, sk_{i_0}), (pk_{i_1}, sk_{i_1}), R)$. Here the set of public keys R contains pk_{i_0} and pk_{i_1}, where (pk_{i_0}, sk_{i_0}) and (pk_{i_1}, sk_{i_1}) are valid key pairs. The challenger chooses a random bit $b \in \{0, 1\}$ and runs the protocol as a prover who has sk_{i_b}. $\mathbf{Run}[\mathsf{P}(param, R, sk_{i_b}) \leftrightarrow \mathcal{A}]$.

Output Phase: $\mathcal{A}$ finally outputs its guess b^* for b. If $b = b^*$ the challenger returns 1. Otherwise returns 0.

Definition 2.5. Let $\mathcal{AID} = (\mathsf{SetUp}, \mathsf{Reg}, \mathsf{P}, \mathsf{V})$ be an AID scheme, $\mathcal{A}$ an adversary, and n a security parameter. The advantage of $\mathcal{A}$ in attacking $\mathcal{AID}$ is defined by

$$\mathbf{Adv}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}(n) := \left| \Pr\left[\mathbf{Exp}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right|.$$

We say that $\mathcal{AID}$ has anonymity with full key exposure if $\mathbf{Adv}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}(\cdot)$ is negligible for every polynomial-time $\mathcal{A}$.

3 Main Tools

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 $L(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i=1}^n \alpha_i \mathbf{b}_i \mid \alpha_i \in \mathbb{Z}\}$ of all integral combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$. The sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a *basis* of the lattice L. We also denote $\mathbf{B}$ as the sequence of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ For more details on lattices, see the textbook by Micciancio and Goldwasser [25]. We give the definitions of well-known lattice problems, the Shortest Vector Problem (SVP^p) and its approximation version.

Definition 3.1 (SVP^p)**.** Given a basis $\mathbf{B}$ of a lattice L, the problem is finding a non-zero vector $\mathbf{v} \in L$ such that for any non-zero vector $\mathbf{x} \in L$, $\|\mathbf{v}\|_p \leq \|\mathbf{x}\|_p$.

Definition 3.2 (SVP^p_γ)**.** Given a basis $\mathbf{B}$ of a lattice L, the problem is finding a non-zero vector $\mathbf{v} \in L$ such that for any non-zero vector $\mathbf{x} \in L$, $\|\mathbf{v}\|_p \leq \gamma \|\mathbf{x}\|_p$.

A few lattice-based cryptographic schemes are based on the worst-case hardness of SVP^p_γ for some γ, e.g., [2, 33, 34].

We next give the definition of the gap version of SVP^p_γ, which is the underlying problem of Micciancio-Regev hash functions [26].

Definition 3.3 (GapSVP^p_γ)**.** For a gap function γ, an instance of GapSVP_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 non-zero vector $\mathbf{v} \in L$ such that $\|\mathbf{v}\|_p \leq d$. In NO input, for any non-zero vector $\mathbf{v} \in L$, $\|\mathbf{v}\|_p > \gamma d$.

We also define the Small Integer Solution problem SIS (in the l_2 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 3.4 ($\mathrm{SIS}^p_{q,m,\beta}$)**.** For a fixed integer q and real β, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the problem is finding a non-zero integer vector $\mathbf{z} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\mathbf{Az} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\|_p \leq \beta$.

Hash functions based on lattice problems: Next, we review a family of collision-resistant hash functions given by Micciancio and Regev [26].

Let n be a security parameter (or a rank of an underlying lattice problem). 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, $\mathcal{H}(q, m) = \{f_{\mathbf{A}} : \{0,1\}^{m(n)} \to \mathbb{Z}^n_{q(n)} \mid \mathbf{A} \in \mathbb{Z}^{n \times m(n)}_{q(n)}\}$, where $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{Ax} \bmod q(n)$.

Originally, Ajtai [2] showed, for suitably chosen $q(n)$ and $m(n)$, the problem, which is, given $\mathcal{H}_{q,m}$, finding a short non-zero vector $\mathbf{v}$ in a lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^n \mid \mathbf{Ax} \equiv \mathbf{0} \pmod q\}$ such that $\|\mathbf{v}\| \leq n$, i.e., solving $\mathrm{SIS}^2_{q,m,n}$, is hard on average under the assumption that SVP_γ is hard in the worst case within some polynomial approximation factor γ. It is known that $\mathcal{H}(q, m)$ is indeed collision resistant for suitably chosen q and m by Goldreich, Goldwasser, and Halevi [13]. They observed that finding a collision $(\mathbf{x}, \mathbf{x}')$ for $f_{\mathbf{A}} \in \mathcal{H}(q, m)$ implies finding a short non-zero vector $\mathbf{z} = \mathbf{x} - \mathbf{x}'$ such that $\|\mathbf{z}\| \leq \sqrt{m}$ and $\mathbf{Az} = \mathbf{0} \bmod q$, i.e, solving $\mathrm{SIS}^2_{q,m,\sqrt{m}}$. Cai and Nerurkar [6] and Micciancio [23] improved an approximation factor of the underlying lattice problems. Recently, Micciancio and Regev showed that $\mathcal{H}(q, m)$ is collision resistant under the assumption that the gap version of $\mathrm{SVP}^2_{\tilde{O}(n)}$ is hard in the worst case [26].

Theorem 3.5 ([26])**.** *For any polynomially bounded functions $\beta = \beta(n)$, $m = m(n)$, $q = q(n)$, with $q \geq 4\sqrt{m}n^{3/2}\beta$ and $\gamma = 14\pi\sqrt{n}\beta$, there exists a probabilistic polynomial-time reduction from solving* GapSVP^2_γ *in the worst case to solving* $\mathrm{SIS}^2_{q,m,\beta}$ *on the average with non-negligible probability.*

Setup and key-generation algorithms: Next, we restrict the domain of hash functions to work the hash function for the key-generation algorithm in Stern's ID scheme. The restricted version is given as follows:

$$\mathcal{H}'(q,m,w) = \left\{h_{\mathbf{A}} : \mathrm{B}(m,w) \to \mathbb{Z}^n_{q(n)} \mid \mathbf{A} \in \mathbb{Z}^{n\times m(n)}_{q(n)}\right\},$$

where $h_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q(n)$. Observe that finding a collision $(\mathbf{x}, \mathbf{x}')$ for $h_{\mathbf{A}} \in \mathcal{H}'(q,m,w)$ implies finding a short vector $\mathbf{z} = \mathbf{x} - \mathbf{x}'$ such that $\|\mathbf{z}\| \leq \sqrt{2w}$ and $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$, i.e, solving the instance $(q, \mathbf{A}, \sqrt{2w})$ of $\mathrm{SIS}^2_{q,m,\sqrt{2w}}$.

The setup algorithm on input 1^n outputs a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. The key-generation algorithm on input $\mathbf{A}$ chooses a random vector $\mathbf{x} \in \mathrm{B}(m,w)$, computes $\mathbf{y} = \mathbf{A}\mathbf{x}$, and outputs $(pk, sk) = (\mathbf{y}, \mathbf{x})$.

String commitment scheme: In this section, we describe a statistically hiding and computationally binding string commitment scheme based on the Micciancio-Regev 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, $\mathcal{H}(q,m) = \{f_{\mathbf{A}} : \{0,1\}^{m(n)} \to \mathbb{Z}^n_{q(n)} \mid \mathbf{A} \in \mathbb{Z}^{n\times m(n)}_{q(n)}\}$, where $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q(n)$.

General constructions of a statistically hiding and computationally binding string commitment scheme are known from a family of collision-resistant hash functions [9, 10, 15]. Their constructions used universal hash functions for the statistical-hiding property. Meanwhile, we can give more direct and simpler construction from the Micciancio-Regev hash functions without the universal hash functions.

We now describe our string commitment scheme. The input of the commitment function is an m-bit vector $\mathbf{x}$ 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., $\mathbf{x} = {}^t(\rho_1, \ldots, \rho_{m/2}, s_1, \ldots, s_{m/2})$. We then define the commitment function on inputs s and ρ as

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

Lemma 3.6. *For any polynomially bounded functions $m = m(n)$, $q = q(n)$, $\gamma = \gamma(n)$, with $q \geq 4mn^{3/2}$, $\gamma = 14\pi\sqrt{nm}$, and $m > 10n\log q$, if* GapSVP^2_γ *is hard in the worst case then* $\mathrm{Com}_{\mathbf{A}}$ *is a statistically hiding and computationally binding string commitment scheme in the trusted setup model.*

In particular, for any $m(n) = \Theta(n\log n)$, there exists $q(n) = O(n^{2.5}\log n)$, and $\gamma(n) = O(n\sqrt{\log n})$, such that $m(n) > 10n\log q$ and if GapSVP^2_γ *is hard in the worst case then* $\mathrm{Com}_{\mathbf{A}}$ *is a statistically hiding and computationally binding string commitment scheme in the trusted setup model.*

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

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

Claim 3.7 (Claim 5. 3 in [34])**.** *Let G be a finite Abelian group and let $l \geq c\log|G|$. If $c \geq 5$, $\Pr_{g_1,\ldots,g_l\in G}\left[\Delta\left((g_1,\ldots,g_l), \sum_{i=1}^l r_ig_i\right), ((g_1,\ldots,g_l),u)\right) > \frac{2}{|G|}\right] \leq \frac{1}{|G|}$ for random variables $g_1,\ldots,g_l \in G, r_1,\ldots,r_l \in \{0,1\}$ and $u \in G$.*

In our proof, we consider $\mathbb{Z}_q^n$ as the finite group G. $\Delta(((\mathbf{a}_1,\ldots,\mathbf{a}_{m/2}), \sum_{i=1}^{m/2}\rho_i\mathbf{a}_i), ((\mathbf{a}_1,\ldots,\mathbf{a}_{m/2}),\mathbf{u}))$ is then negligible with probability exponentially close to 1, where $\mathbf{u} \in \mathbb{Z}_q^n$ is a uniform random variable. Thus, $\Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(0^{m/2};\rho)), (\mathbf{A},\mathbf{u}))$ is negligible. Since $\Delta((\mathbf{A}, \mathbf{u} + \sum_{i=1}^{m/2} s_i\mathbf{a}_{i+m/2}), (\mathbf{A}, \sum_{i=1}^{m/2}\rho_i\mathbf{a}_i + \sum_{i=1}^{m/2} s_i\mathbf{a}_{i+m/2}))$ is also negligible, $\Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s;\rho)), (\mathbf{A},\mathbf{u}))$ is negligible for any message s. By the triangle inequality, we have

$$\Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_1;\rho_1)), (\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_2;\rho_2))) \leq \Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_1;\rho_1)), (\mathbf{A},\mathbf{u})) + \Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_2;\rho_2)), (\mathbf{A},\mathbf{u}))$$

for any messages s_1 and s_2 and uniform random strings ρ_1 and ρ_2. It follows that $\Delta((\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_1;\rho_1)), (\mathbf{A}, \mathrm{Com}_{\mathbf{A}}(s_2;\rho_2)))$ is negligible in n, which completes the proof. □ □

Using the Merkle-Damgård technique [21, 8], we obtain the 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 $\mathrm{Com}_{\mathbf{A}} : \{0,1\}^{m/2} \times \{0,1\}^{m/2} \to \mathbb{Z}_q^n$. We use this commitment scheme in the rest of the paper.

4 Identification Scheme

Plugging the setup and key-generation algorithms and the string commitment scheme into Stern's ID scheme [39], we obtain a concrete identification scheme. Our key-generation algorithm on input $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ outputs a vector $\mathbf{x} \in \mathrm{B}(m, w)$ as the secret key and a vector $\mathbf{y} = \mathbf{A}\mathbf{x}$ as the public key.

Our protocol is obtained by modifying Stern's ID scheme [39], which presents a zero-knowledge argument protocol based on the decoding problem on binary codewords called Syndrome Decoding Problem. Since his protocol deals with binary codewords, it works on the binary field $\mathbb{Z}_2$. Stern also proposed that an analogous scheme in $\mathbb{Z}_q$, where q is extremely small number (typically 3, 5, or 7) [39]. We adjust this parameter to connect his framework to our assumptions of the lattice problems.

The following is our ID scheme based on GapSVP. Note that this protocol is in parallel repeated n times to achieve an exponentially small soundness error. (The soundness error is at most $2/3$ for the single repetition.)

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

KG**:** The key-generation algorithm, on input $\mathbf{A}$, chooses a random vector $\mathbf{x} \in \mathrm{B}(m, w)$, computes $\mathbf{y} := \mathbf{A}\mathbf{x} \bmod q$. Outputs $(pk, sk) = (\mathbf{y}, \mathbf{x})$.

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

Step P1: For every $i \in \{1, \ldots, n\}$, choose a random permutation π_i over $\{1, \ldots, m\}$, a random vector $\mathbf{r}_i \in \mathbb{Z}_q^m$, and random strings $\rho_{i,1}, \rho_{i,2}$, and $\rho_{i,3}$. Compute $c_{i,1} = \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}\mathbf{r}_i; \rho_{i,1})$, $c_{i,2} = \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{r}_i); \rho_{i,2})$ and $c_{i,3} = \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{x}+\mathbf{r}_i); \rho_{i,3})$. Send $Cmt := ((c_{1,1}, c_{1,2}, c_{1,3}), \ldots, (c_{n,1}, c_{n,2}, c_{n,3}))$ to V.

Step V1 V sends random challenges $Ch := (Ch_1, \ldots, Ch_n) \in \{1,2,3\}^n$ to P.

Step P2 Parse Ch as $(Ch_1, \ldots, Ch_n)$.

1. If $Ch_i = 1$, P reveals $c_{i,2}$ and $c_{i,3}$. Set $Rsp_i = (\pi_i(\mathbf{x}), \pi_i(\mathbf{r}_i), \rho_{i,2}, \rho_{i,3})$.
2. If $Ch_i = 2$, P reveals $c_{i,1}$ and $c_{i,3}$. Set $Rsp_i = (\pi_i, \mathbf{x} + \mathbf{r}_i, \rho_{i,1}, \rho_{i,3})$.
3. If $Ch_i = 3$, P reveals $c_{i,1}$ and $c_{i,2}$. Set $Rsp_i = (\pi_i, \mathbf{r}_i, \rho_{i,1}, \rho_{i,2})$.

Set $Rsp := (Rsp_1, \ldots, Rsp_n)$ and send Rsp to V.

Step V2 Parse Rsp as $(Rsp_1, \ldots, Rsp_n)$.

1. If $Ch_i = 1$, parse Rsp_i as $(\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, \rho_{i,2}, \rho_{i,3})$. Check whether the weight of $\mathbf{x}$ and the commitments $c_{i,2}$ and $c_{i,3}$ are correct, that is, $\mathbf{z}_{i,1} \in \mathrm{B}(m, w)$, $c_{i,2} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_{i,2}; \rho_{i,2})$, and $c_{i,3} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_{i,1} + \mathbf{z}_{i,2}; \rho_{i,3})$. If they are correct, set $Dec_i = 1$ and otherwise set $Dec_i = 0$.
2. If $Ch_i = 2$, parse Rsp_i as $(\pi_i, \mathbf{z}_i, \rho_{i,1}, \rho_{i,3})$. Check whether the commitments $c_{i,1}$ and $c_{i,3}$ are correct, that is, $c_{i,1} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}\mathbf{z}_i - \mathbf{y}; \rho_{i,1})$ and $c_{i,3} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{z}_i); \rho_{i,3})$. If they are correct, set $Dec_i = 1$ and otherwise set $Dec_i = 0$.
3. If $Ch_i = 3$, parse Rsp_i as $(\pi_i, \mathbf{z}_i, \rho_{i,1}, \rho_{i,2})$. Check whether the commitments $c_{i,1}$ and $c_{i,2}$ are correct, that is, $c_{i,1} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}\mathbf{z}_i; \rho_{i,1})$ and $c_{i,2} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{z}_i); \rho_{i,2})$. If they are correct, set $Dec_i = 1$ and otherwise set $Dec_i = 0$.

If $Dec_i = 1$ for all i, set $Dec = 1$. Otherwise set $Dec = 0$. Output Dec.

We next give the security proof of our ID protocol, which concerns impersonation under concurrent attacks. Before the proof of security, we need to note the following trivial lemma.

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

We now show the security of the above ID scheme as follows.

Theorem 4.2. *For any polynomially bounded functions* $m = m(n)$*,* $q = q(n)$*, with* $q \geq 4mn^{3/2}$*,* $\gamma = 14\pi\sqrt{nm}$*,* $m \geq 10n \log q$*, and* $q^n / |\mathrm{B}(m,w)|$ *is negligible in* n*, if* GapSVP_γ^2 *is hard in the worst case then the above ID scheme is secure against impersonation under concurrent attacks.*

In particular, for any $m(n) = \Theta(n \log n)$*, there exists* $q(n) = O(n^{2.5} \log n)$*,* $\gamma(n) = O(n\sqrt{\log n})$*, and* $w(n) = \Theta(m(n))$ *such that* $q^n / |\mathrm{B}(m,w)|$ *is negligible in* n *and if* GapSVP_γ^2 *is hard in the worst case then the ID scheme is secure against impersonation under concurrent attacks.*

Proof. We show that if there exists an impersonator $\mathcal{I}$ which succeeds impersonation under concurrent attacks with non-negligible probability ϵ, there exists $\mathcal{A}$ that solves $\mathrm{SIS}^2_{q,m,\sqrt{m}}$ on average. Then there exists any instance of GapSVP_γ^2 by Theorem 3.5.

We first overview the strategy of $\mathcal{A}$. The algorithm $\mathcal{A}$ can control the impersonator $\mathcal{I}$ by feeding a random tape and a challenge. Given $\mathbf{A}$, $\mathcal{A}$ chooses a random secret key $\mathbf{x} \in \mathrm{B}(m,w)$ and compute $\mathbf{y} := \mathbf{Ax}$. $\mathcal{A}$ executes $\mathcal{I}$ on inputs $(\mathbf{A}, \mathbf{y})$. We note that $\mathcal{A}$ can simulate the oracles Conv and Prov, since $\mathcal{A}$ has the secret key $\mathbf{x}$. $\mathcal{A}$ executes $\mathcal{I}$ three times with random challenges and a fixed random tape. Then, $\mathcal{A}$ obtains three transcripts $(Cmt^{(i)}, Ch^{(i)}, Rsp^{(i)}, Dec^{(i)})$ for $i = 1,2,3$ as the results of the interactions between $\mathcal{I}$ and $\mathcal{A}$. Note that $Cmt^{(1)} = Cmt^{(2)} = Cmt^{(3)}$ since $\mathcal{A}$ fixes the random tape to work $\mathcal{I}$. By the assumption, $\mathcal{A}$ obtain good transcript such that with non-negligible probability $Dec^{(i)} = (Dec_1^{(i)}, \ldots, Dec_n^{(i)})$ are all 1 for every i. Then, $\mathcal{A}$ can find $\mathbf{x}'$ from $(\mathbf{A}, \mathbf{y})$ or find $(s,\rho) \neq (s',\rho')$ such that $\mathrm{Com}_{\mathbf{A}}(s;\rho) = \mathrm{Com}_{\mathbf{A}}(s';\rho')$ by using the fact that $Cmt^{(1)} = Cmt^{(2)} = Cmt^{(3)}$. In the former case, we will show that $\mathbf{x}' \neq \mathbf{x}$ with probability at least $1/2$. $\mathcal{A}$ outputs $\mathbf{z} = \mathbf{x}' - \mathbf{x}$. Since $\mathbf{z} \in \{-1,0,+1\}^m$, the norm $\|\mathbf{z}\| \leq \sqrt{m}$. In the latter case, $\mathcal{A}$ computes $\mathbf{z} \neq \mathbf{z}' \in \{0,1\}^m$ from (s,ρ) and (s',ρ') such that $\mathrm{Com}_{\mathbf{A}}(s;\rho) = \mathbf{Az}$ and $\mathrm{Com}_{\mathbf{A}}(s';\rho') = \mathbf{Az}'$. Thus, $\mathcal{A}$ outputs $\mathbf{z}'' = \mathbf{z}' - \mathbf{z}$, where $\|\mathbf{z}\| \leq \sqrt{m}$.

$\mathcal{A}$ then executes the following procedure.

1. Choose a random tape r of $\mathcal{I}$.
2. Choose challenges $Ch^{(1)}, Ch^{(2)}, Ch^{(3)}$ randomly.
3. For each $i = 1,2,3$, execute the experiment with random challenges $Ch^{(i)}$ and a fixed random tape r, and then $\mathcal{I}$ outputs three tuples of transcripts $(Cmt^{(i)}, Ch^{(i)}, Rsp^{(i)}, Dec^{(i)})$.

We have that the probability that all $Dec^{(i)}$ are 1 is at least $(\epsilon/2)^3$ by the Heavy Row Lemma [29]. Also, we have $\Pr[\exists j : Ch_j^{(1)} \neq Ch_j^{(2)}, Ch_j^{(2)} \neq Ch_j^{(3)}, Ch_j^{(3)} \neq Ch_j^{(1)}] = 1 - (7/9)^n$ by a simple calculation. $\mathcal{A}$ therefore obtains good three transcripts with non-negligible probability $(\epsilon/2)^3 - (7/9)^n$.

We next show how $\mathcal{A}$ obtain a secret key or violate the binding property of the string commitment scheme by using three good transcripts. Assume that $\mathcal{A}_2$ 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 $j \in \{1,\ldots,n\}$. Without loss of generality, we assume that $Ch_j^{(i)} = i$. We parse $Rsp_j^{(i)}$ as in Step V2. From the assumption, we have four equations as follows (We omit j for simplification):

$$\begin{array}{rll} c_1 & = \mathrm{Com}_{\mathbf{A}}(\pi^{(2)}, \mathbf{Az}^{(2)} - \mathbf{y}; \rho_1^{(2)}) & = \mathrm{Com}_{\mathbf{A}}(\pi^{(3)}, \mathbf{Az}^{(3)}; \rho_1^{(3)}), \\ c_2 & = \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_2^{(1)}; \rho_2^{(1)}) & = \mathrm{Com}_{\mathbf{A}}(\pi^{(3)}(\mathbf{z}^{(3)}); \rho_2^{(3)}), \\ c_3 & = \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_1^{(1)} + \mathbf{z}_2^{(1)}; \rho_3^{(1)}) & = \mathrm{Com}_{\mathbf{A}}(\pi^{(2)}(\mathbf{z}^{(2)}); \rho_3^{(2)}), \\ \mathbf{z}_1^{(1)} & \in \mathrm{B}(m,w) & \end{array}$$

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

Next, we suppose that there exist no distinct pair of arguments of $\mathrm{Com}_{\mathbf{A}}$. Let π denote the inverse permutation of $\pi^{(2)}$. From the first equation, we have $\pi^{-1} = \pi^{(2)} = \pi^{(3)}$. Thus, we obtain $\mathbf{z}^{(2)} = \pi(\mathbf{z}_1^{(1)} + \mathbf{z}_2^{(1)})$

from the third equation. Combining it with the first equation, we have $\mathbf{A}\mathbf{z}^{(3)} = \mathbf{A}(\pi(\mathbf{z}_1^{(1)}) + \pi(\mathbf{z}_2^{(1)})) - \mathbf{y}$. We obtain $\mathbf{y} = \mathbf{A}\pi(\mathbf{z}_1^{(1)})$ since $\pi^{-1}(\mathbf{z}_2^{(1)}) = \mathbf{z}^{(3)}$ in the second equation. We already have $\mathbf{z}_1^{(3)} \in \mathrm{B}(m, w)$. Then, $\mathcal{A}$ set $\mathbf{x}' := \pi(\mathbf{z}_1^{(3)})$.

We now have to show that $\mathbf{x}' \neq \mathbf{x}$ with probability at least $1/2$. By Lemma 4.1, there must be another secret key $\mathbf{x}'$ with overwhelming probability. Note that the protocol is indeed a statistical witness-indistinguishable protocol and $\mathcal{I}$'s view is independent of $\mathcal{A}$'s choice of $\mathbf{x}$ with overwhelming probability, since Com is a statistically hiding string commitment scheme. Thus we have $\mathbf{x}' \neq \mathbf{x}$ with probability at least $1/2$. In this case $\mathcal{A}$ solves $\mathrm{SIS}_{q,m,\sqrt{m}}$ as in the overview. □ □

Note that, combining the witness-indistinguishability property of Stern's scheme with Lemma 4.1, we indeed show that our scheme has the witness-hiding property.

In the case of the parameters specified in Theorem 4.2, the size of the system parameter $\mathbf{A}$ is $\tilde{O}(n^2)$ and the size of the public key $\mathbf{y}$ is $\tilde{O}(n)$. The cost of communication in the single repetition is $\tilde{O}(n)$ and the total cost of communication is $\tilde{O}(n^2)$ by the parallel repetition.

5 Ad Hoc Anonymous Identification Scheme

Our construction for lattice-based AID schemes is inspired by the results in Wu, Chen, Wang, and Wang [40], which proposed an AID scheme based on the Weak Dependence Problem[2]. The idea of [40] is as follows: Let $(a_1, \ldots, a_m)$ be a system parameter. Each user chooses a secret key $\mathbf{x}_i \in \{-1, 0, +1\}$ and computes a public key $y_i = \sum_{j \in \mathbf{x}_i} a_j$. In their AID scheme, a group is specified by the set of public keys $(y_1, \ldots, y_l)$ and he/she proves that he/she has a partition $\mathbf{x}' = {}^t({}^t\mathbf{x}_i\, {}^t-\mathbf{e}_{i,l}) \in \{-1, 0, +1\}^{m+l}$ for an instance $(a_1, \ldots, a_m, y_1, \ldots, y_l)$, where $\mathbf{e}_{i,l}$ is an l-dimensional vector ${}^t(0 \ldots, 0\,1\,0 \ldots 0)$ whose i-th element is 1.

Our construction is as follows: Let $\mathbf{A}$ be a system parameter. Each user has a secret key $\mathbf{x}_i \in \mathrm{B}(m, w)$ and public key $\mathbf{y}_i := \mathbf{A}\mathbf{x}_i$. In the AID scheme, a group is specified by a set of public keys $(\mathbf{y}_1, \ldots, \mathbf{y}_l)$ of the members. A user in the group, who has a secret key $\mathbf{x}_i$, convinces the verifier that he/she know that $\mathbf{x}' := {}^t({}^t\mathbf{x}_i {}^t\mathbf{e}_{i,l})$ such that $[\mathbf{A}\,\mathbf{y}_1 \ldots \mathbf{y}_l]\mathbf{x}' = \mathbf{0}$, the number of 1 in $\mathbf{x}'$ is w, and the number of -1 in $\mathbf{x}'$ is 1.

We here construct an AID scheme based on GapSVP. We define $\mathrm{B}'(m, w)$ as $\{\mathbf{x} \in \{-1, 0, +1\}^m \mid w_{+1}(\mathbf{x}) = w \text{ and } w_{-1}(\mathbf{x}) = 1\}$, where $w_{+1}(\mathbf{x})$ denotes the number of $+1$ in $\mathbf{x}$ and $w_{-1}(\mathbf{x})$ denotes the number of -1 in $\mathbf{x}$.

Similarly to the ID scheme in Section 4, the protocol is repeated n times in parallel to achieve an exponentially small soundness error. (The soundness error is at most $2/3$ for the single repetition again.)

SetUp: On input 1^n, output a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.

Reg: On input *param*, choose a random vector $\mathbf{x}$ from $\mathrm{B}(m, w)$ and compute $\mathbf{y} := \mathbf{A}\mathbf{x}$. Output $(pk, sk) := (\mathbf{y}, \mathbf{x})$.

P, V: The common inputs are $\mathbf{A}$ and $(\mathbf{y}_1, \ldots, \mathbf{y}_l)$. The prover's auxiliary input is $sk_i = \mathbf{x}$. Let $\mathbf{A}' := [\mathbf{A}\,\mathbf{y}_1 \ldots \mathbf{y}_l] \in \mathbb{Z}_q^{n \times (m+l)}$ and $\mathbf{x}' := {}^t({}^t\mathbf{x}\, {}^t-\mathbf{e}_{i,l})$. They interact according to the following protocol:

Step P1 For every $i \in \{1, \ldots, n\}$, choose a random permutation π_i over $\{1, \ldots, m+l\}$, a random vector $\mathbf{r}_i \in \mathbb{Z}_q^{m+l}$, and random strings $\rho_{i,1}, \rho_{i,2}$, and $\rho_{i,3}$. Compute $c_{i,1} := \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}'\mathbf{r}_i; \rho_{i,1})$, $c_{i,2} := \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{r}_i); \rho_{i,2})$ and $c_{i,3} := \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{x}' + \mathbf{r}_i); \rho_{i,3})$. Send $Cmt := ((c_{1,1}, c_{1,2}, c_{1,3}), \ldots, (c_{n,1}, c_{n,2}, c_{n,3}))$ to V.

Step V1 V sends random challenges $Ch := (Ch_1, \ldots, Ch_n) \in \{1, 2, 3\}^n$ to P.

Step P2 Parse Ch as $(Ch_1, \ldots, Ch_n)$.

1. If $Ch_i = 1$, P reveals $c_{i,2}$ and $c_{i,3}$. Set $Rsp_i := (\pi_i(\mathbf{x}'), \pi_i(\mathbf{r}_i), \rho_{i,2}, \rho_{i,3})$.

[2] The Weak Dependence Problem is, given $(a_1, \ldots, a_k)$ where a_i is an l-bit natural number, to find a partition $\mathbf{x} \in \{-1, 0, +1\}^k \backslash \{\mathbf{0}\}$ such that $\sum_{i=1}^k x_i a_i = 0$, i.e., to find two non-empty subsets S_1 and S_2 in $\{1, \ldots, k\}$ such that $S_1 \cap S_2 = \emptyset$ and $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i$.

2. If $Ch_i = 2$, P reveals $c_{i,1}$ and $c_{i,3}$. Set $Rsp_i := (\pi_i, \mathbf{x}' + \mathbf{r}_i, \rho_{i,1}, \rho_{i,3})$.
3. If $Ch_i = 3$, P reveals $c_{i,1}$ and $c_{i,2}$. Set $Rsp_i := (\pi_i, \mathbf{r}_i, \rho_{i,2}, \rho_{i,3})$.

Send $Rsp := (Rsp_1, \ldots, Rsp_n)$ to V.

Step V2 Parse Rsp as $(Rsp_1, \ldots, Rsp_n)$.

1. If $Ch_i = 1$, parse Rsp_i as $(\mathbf{z}_{i,1}, \mathbf{z}_{i,2}, \rho_{i,2}, \rho_{i,3})$. Check whether the weight of $\mathbf{x}'$ and the commitments $c_{i,2}$ and $c_{i,3}$ are correct, that is, $\mathbf{z}_{i,1} \in \mathrm{B}'(m+l, w)$, $c_{i,2} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_{i,2}; \rho_{i,2})$, and $c_{i,3} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\mathbf{z}_{i,1} + \mathbf{z}_{i,2}; \rho_{i,3})$. If they are correct set $Dec_i := 1$ and otherwise $Dec_i := 0$.
2. If $Ch_i = 2$, parse Rsp_i as $(\pi_i, \mathbf{z}_i, \rho_{i,1}, \rho_{i,3})$. Check whether the commitments $c_{i,1}$ and $c_{i,3}$ are correct, that is, $c_{i,1} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}'\mathbf{z}_i; \rho_{i,1})$ and $c_{i,3} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{z}_i); \rho_{i,3})$. If they are correct set $Dec_i := 1$ and otherwise $Dec_i := 0$.
3. If $Ch_i = 3$, parse Rsp_i as $(\pi_i, \mathbf{z}_i, \rho_{i,1}, \rho_{i,2})$. Check whether the commitments $c_{i,1}$ and $c_{i,2}$ are correct, that is, $c_{i,1} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i, \mathbf{A}'\mathbf{z}_i; \rho_{i,1})$ and $c_{i,2} \stackrel{?}{=} \mathrm{Com}_{\mathbf{A}}(\pi_i(\mathbf{z}_i); \rho_{i,2})$. If they are correct set $Dec_i := 1$ and otherwise $Dec_i := 0$.

If $Dec_i = 1$ for all $i \in \{1, \ldots, n\}$, then set $Dec := 1$. Otherwise $Dec := 0$. Output Dec.

Theorem 5.1. *Let $\beta := \max\{(w+1)^{3/2}, \sqrt{m}\}$. Assume that there exists an impersonator $\mathcal{I}$ that succeeds impersonation under concurrent chosen-group attacks with non-negligible probability. Then there exists a probabilistic polynomial-time algorithm $\mathcal{A}$ that solves $\mathrm{SIS}^2_{q,m,\beta}$.*

Proof. The algorithm $\mathcal{A}$, given input $\mathbf{A}$, feeds $\mathbf{A}$ to the impersonator $\mathcal{I}$. In the experiment, the impersonator $\mathcal{I}$ will call Init, Corr, Conv, and Prov. If $\mathcal{I}$ calls Init with input i, then $\mathcal{A}$ chooses $\mathbf{s}_i$ at random, computes $\mathbf{y}_i := \mathbf{A}\mathbf{s}_i$, and returns $\mathbf{y}_i$ to $\mathcal{I}$. If $\mathcal{I}$ calls Corr with input i, Conv with inputs i, R, or Prov with inputs i, R, s, M_i, then $\mathcal{A}$ can simulate the oracle Corr, since $\mathcal{A}$ has a secret key $\mathbf{s}_i$ with respect to a public key $\mathbf{y}_i$.

At the end of the experiment, $\mathcal{I}$ will impersonate as a group which is specified by the set of public keys $R = (\mathbf{y}_1, \ldots, \mathbf{y}_l)$. Rewinding $\mathcal{I}$ three times, $\mathcal{A}$ obtain a collision (s, ρ) and (s', ρ') for the commitment scheme $\mathrm{Com}_{\mathbf{A}}$ or a vector $\mathbf{x} = {}^t({}^t\mathbf{x}_1\, {}^t\mathbf{x}_2)$ such that $[\mathbf{A}\, \mathbf{y}_1 \ldots \mathbf{y}_l]\mathbf{x} = \mathbf{0}$, where $\mathbf{x}_1 \in \{-1, 0, 1\}^m$ and $\mathbf{x}_2 \in \{-1, 0, 1\}^l$ and $\mathbf{x} \in \mathrm{B}'(m+l, w)$ as in the proof of Theorem 4.2.

In the former case, $\mathcal{A}$ computes $\mathbf{z} \neq \mathbf{z}' \in \{0,1\}^m$ such that $\mathrm{Com}_{\mathbf{A}}(s; \rho) = \mathbf{A}\mathbf{z}$ and $\mathrm{Com}_{\mathbf{A}}(s'; \rho') = \mathbf{A}\mathbf{z}'$. Hence, $\mathcal{A}$ can output $\mathbf{z}'' = \mathbf{z}' - \mathbf{z}$ such that $\|\mathbf{z}''\| \leq \sqrt{m}$.

In the latter case, we have $\mathbf{A}\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{y}_i = \mathbf{0}$, that is, $\mathbf{A}\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{A}\mathbf{s}_i = \mathbf{0}$. Hence, we obtain that $\mathbf{A}(\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{s}_i) = \mathbf{0}$. Recall that the numbers of $+1$ in $\mathbf{x}$ is w and that of -1 in $\mathbf{x}$ is 1. Hence, $\left\|\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{s}_i\right\| \leq \|\mathbf{x}_1\| + \sum_{i=1}^{l} \left|x_{2,i}\right| \|\mathbf{s}_i\| \leq \sqrt{w+1} + (w+1)\sqrt{w} \leq (w+1)^{3/2}$. By the same argument as in the proof of Theorem 4.2, we have that $\mathbf{x}_1 + \sum_i x_{2,i}\mathbf{s}_i \neq \mathbf{0}$ with probability at least $1/2$. Thus, $\mathcal{A}$ outputs $\mathbf{z} := \mathbf{x}_1 + \sum_i x_{2,i}\mathbf{s}_i$ and solves $\mathrm{SIS}^2_{q,m,(w+1)^{3/2}}$ with non-negligible probability. □ □

Combining Theorem 5.1 with Theorem 3.5, we obtain the following theorem.

Theorem 5.2. *For any polynomially bounded functions $m = m(n)$, $q = q(n)$, with $q \geq 4\sqrt{m}n^{3/2}\max\{(w+1)^{3/2}, \sqrt{m}\}$, $\gamma = 14\pi\sqrt{n}\max\{(w+1)^{3/2}, \sqrt{m}\}$, $m \geq 10n\log q$ and $\binom{m}{w}/q^n = 2^{\omega(\log n)}$, if GapSVP^2_γ is hard in the worst case then the above scheme is secure against impersonation under concurrent chosen-group attacks.*

In particular, for any $m(n) = \Theta(n \log n)$, there exists $q(n) = O(n^{3.5}\log n)$, $\gamma(n) = O(n^2\sqrt{\log n})$, and $w(n) = \Theta(m(n))$ such that $q^n / |\mathrm{B}(m, w)|$ is negligible in n and if GapSVP^2_γ is hard in the worst case then the above scheme is secure against impersonation under concurrent chosen-group attacks.

Since the statistical anonymity of the above scheme is directly implied by the witness-indistinguishability of Stern's scheme, we omit the proof.

In the case of the parameters specified in Theorem 5.2, the cost of communication in the single repetition is $\tilde{O}(n + l)$ and the total cost of communication is $n \cdot \tilde{O}(n + l)$.

References

[1] Abdalla, M., An, J. H., Bellare, M., and Namprempre, C. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology – EUROCRYPT 2002*, L. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 418–433.

[2] Ajtai, M. Generating hard instances of lattice problems (extended abstract). In *Proceedings on 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, ACM, pp. 99–108. See also ECCC TR96-007.

[3] Ajtai, M., and Dwork, C. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings on 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, ACM, pp. 284–293. See also ECCC TR96-065.

[4] Bellare, M., and Palacio, A. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Advances in Cryptology – CRYPTO 2002*, M. Yung, Ed., vol. 2442 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 162–177.

[5] Brassard, G., Ed. *Advances in Cryptology – CRYPTO '89*, vol. 435 of *Lecture Notes in Computer Science*, Springer-Verlag.

[6] Cai, J.-Y., and Nerurkar, A. An improved worst-case to average-case connection for lattice problems. In *38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, IEEE Computer Society, pp. 4 68–477.

[7] Coster, M. J., Joux, A., LaMacchia, B. A., Odlyzko, A. M., Schnorr, C.-P., and Stern, J. Improved low-density subset sum algorithms. *Computational Complexity 2* (1992), 111–128.

[8] Damgård, I. A design principle for hash functions. In Brassard [5], pp. 416–427.

[9] Damgård, I. B., Pedersen, T. P., and Pfizmann, B. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *Journal of Cryptology 10*, 3 (1997), 163–194. Preliminary version in *CRYPTO '93*, 1993.

[10] Damgård, I. B., Pedersen, T. P., and Pfizmann, B. Statistical secrecy and multibit commitments. *IEEE Transactions on Information Theory 44*, 3 (May 1998), 1143–1151.

[11] Dodis, Y., Kiayias, A., Nicolosi, A., and Shoup, V. Anonymous identification in *ad hoc* groups. In *Advances in Cryptology – EUROCRYPT 2004*, C. Cachin and J. Camenisch, Eds., vol. 3027 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 609–626.

[12] Feige, U., and Shamir, A. Witness indistinguishable and witness hiding protocols. In *Proceedings on 22nd Annual ACM Symposium on Theory of Computing (STOC '90)*, ACM, pp. 416–426.

[13] Goldreich, O., Goldwasser, S., and Halevi, S. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC) 3*, 42 (1996).

[14] Goldreich, O., Goldwasser, S., and Halevi, S. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology – CRYPTO '97*, B. S. Kaliski, Jr., Ed., vol. 1294 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 112–131.

[15] Halevi, S., and Micali, S. Practical and provably-secure commitment scheme from collision-free hashing. In *Advances in Cryptology – CRYPTO '96*, N. Koblitz, Ed., vol. 1109 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 201–215.

[16] Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J. H., and Whyte, W. NTRUSign: Digital signature using the NTRU lattice. In *Topics in Cryptology – CT-RSA 2003*, M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 122–140.

[17] Hoffstein, J., Pipher, J., and Silverman, J. H. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory, Third International Symposium, ANTS-III*, J. Buhler, Ed., vol. 1423 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 267–288.

[18] Impagliazzo, R., and Zuckerman, D. How to recycle random bits. In *30th Annual Symposium on Foundations of Computer Science (FOCS '89)*, IEEE Computer Society, pp. 248–253.

[19] Lagarias, J. C., and Odlyzko, A. M. Solving low-density subset sum problems. *Journal of the ACM 32*, 1 (1985), 229–246.

[20] Lyubashevsky, V., and Micciancio, D. Generalized compact knapsacks are collision resistant. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Part II*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds., vol. 4052 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 144–155.

[21] Merkle, R. C. One way hash functions and DES. In Brassard [5], pp. 428–446.

[22] Merkle, R. C., and Hellman, M. E. Hiding information and signatures in trap door knapsacks. *IEEE Transactions on Information Theory 24*, 5 (September 1978), 525–530.

[23] Micciancio, D. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing 34*, 1 (2004), 118–169. Preliminary version in *STOC 2002*, 2002.

[24] Micciancio, D. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. *Computational Complexity* (2006). To be appeared.

[25] Micciancio, D., and Goldwasser, S. *Complexity of Lattice Problems: a cryptographic perspective*, vol. 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[26] Micciancio, D., and Regev, O. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*, IEEE Computer Society, pp. 372–381.

[27] Micciancio, D., and Vadhan, S. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Advances in Cryptology – CRYPTO 2003*, D. Boneh, Ed., vol. 2729 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 282–298.

[28] Nguyen, P. Q., and Regev, O. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. In *Advances in Cryptology – EUROCRYPT 2006*, S. Vaudenay, Ed., vol. 4004 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 271–288.

[29] Ohta, K., and Okamoto, T. On concrete security treatment of signatures derived from identification. In *Advances in Cryptology – CRYPTO '98*, H. Krawczyk, Ed., vol. 1462 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 354–369.

[30] Peikert, C., and Rosen, A. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography, 3rd Theory of Cryptography Conference, TCC 2006*, S. Halevi and T. Rabin, Eds., vol. 3876 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 145–166.

[31] Peikert, C., and Waters, B. Lossy trapdoor functions and their applications. *Electronic Colloquium on Computational Complexity (ECCC) 14*, 080 (2007).

[32] Pointcheval, D., and Poupard, G. A new NP-complete problem and public-key idenitification. *Designs, Codes and Cryptography 28*, 1 (January 2003), 5–31.

[33] Regev, O. New lattice-based cryptographic constructions. *Journal of the ACM 51*, 6 (2004), 899–942. Preliminary version in *STOC 2003*, 2003.

[34] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings on the 37th Annual ACM Symposium on Theory of Computing (STOC 2005)*, H. N. Gabow and R. Fagin, Eds., ACM, pp. 84–93.

[35] Shamir, A. A zero-knowledge proof for knapsacks. Presented at a workshop on Probabilistic Algorithms, Marseille, March 1976.

[36] Shamir, A. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory 30*, 5 (1984), 699–704.

[37] Shamir, A. An efficient identification scheme based on permuted kernels (extended abstract). In Brassard [5], pp. 606–609.

[38] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5 (1997), 1484–1509.

[39] Stern, J. A new paradigm for public key identification. *IEEE Transactions on Information Theory 42*, 6 (November 1996), 749–765. Preliminary version in *CRYPTO '93*, 1993.

[40] Wu, Q., Chen, X., Wang, C., and Wang, Y. Shared-key signature and its application to anonymous authentication in ad hoc group. In *Information Security, 7th International Conference, ISC 2004*, K. Zhang and Y. Zheng, Eds., vol. 3225 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 330–341.

A Constructions from the Lyubashevsky-Micciancio Hash Functions

Several families of lattice-based hash functions are known to have small description sizes such as [24, 30, 20]. In this section, we construct the ID scheme and the AID schemes based on the compact hash functions of Micciancio and Lyubashevsky [20]. We basically use the notations of [20].

Let $f \in \mathbb{Z}[x]$ be a monic and irreducible polynomial of degree n. Consider the quotient ring $\mathbb{Z}[x]/\langle f \rangle$. We use the standard set of representatives $\{(g \bmod f) : g \in \mathbb{Z}[x]\}$. In this section we identify a polynomial $\mathbf{a}(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathbb{Z}[x]/\langle f \rangle$ with an n-dimensional integer vector $\mathbf{a} = {}^t(a_0, \ldots, a_{n-1})$. We define a norm with respect to f as follows: For $g \in \mathbb{Z}[x]$, $\|(g + \langle f \rangle)\|_f = \|g \bmod f\|_\infty$. We write $\|g\|_f$ instead of $\|g + \langle f \rangle\|_\infty$.

We note that any ideal $I \subseteq \mathbb{Z}[x]/\langle f \rangle$ defines the corresponding n-dimensional integer lattice $L(I) \subseteq \mathbb{Z}^n$. Notice that a class of the lattices representable in this way is contained in a general class of all integer lattices $L(\mathbf{B}) \subseteq \mathbb{Z}^n$. If a given lattice in SVP^p is restricted in a class Λ of lattices, we denote by Λ-SVP^p the problem over such restricted lattices in Λ. We also denote by $\Lambda(f)$ the set of lattices that are isomorphic to ideals of $\mathbb{Z}[x]/\langle f \rangle$. See [20] for the details. We here deal with $\Lambda(f)$-SVP$^\infty_\gamma$, i. e. , SVP with approximation factor γ for l_∞ norm whose input lattices are restricted in $\Lambda(f)$.

A.1 The Lyubashevsky-Micciancio Hash Functions

Lyubashevsky and Micciancio constructed a family of collision-resistant hash functions based on the worst-case hardness of $\Lambda(f)$-SVP for suitable f.

We review what f is suitable for the construction of Lyubashevsky and Micciancio. The property of f is defined as that the ring norm $\|g\|_f$ is not much bigger than $\|g\|_\infty$ for any polynomial g. Formally, Lybashevsky and Micciancio capture this property as the *expansion factor* of f:

$$\mathrm{EF}(f, k) = \max_{g \in \mathbb{Z}[x], \deg(g) \le k(\deg(f)-1)} \|g\|_f \,/\, \|g\|_\infty .$$

For example, a simple calculation shows that $\mathrm{EF}(x^n \pm 1, k) \le k$ and $\mathrm{EF}(x^{n-1} + x^{n-2} + \cdots + 1, k) \le 2k$. We say a polynomial f is suitable if f is a monic and irreducible in $\mathbb{Z}[x]$ and there is a constant c such that $\mathrm{EF}(f, k) \le ck$ for any natural number k. The security of the Lyubashevsky-Micciancio hash functions is based on the worst-case hardness of $\Lambda(f)$-SVP for a suitable polynomial f. See [20] for more details. They adopt a family of polynomials such as $x^n + 1$ and $x^{n-1} + x^{n-2} + \cdots + 1$ for n such that the polynomials are irreducible in $\mathbb{Z}[x]$. Let $D(m, d) = \{\mathbf{x} \in \mathbb{Z}^m \mid \|\mathbf{x}\|_\infty \le d\}$. We now describe a family of hash functions given in [20].

$$\mathcal{H}_{\mathcal{I}}(f, q, m, d) = \left\{ h_A : D(m, d) \to \mathbb{Z}_{q(n)}[x]/\langle f \rangle \mid A = (\mathbf{a}_1, \ldots, \mathbf{a}_{m(n)/n}) \in (\mathbb{Z}_{q(n)}/\langle f \rangle)^{m(n)/n} \right\},$$

where $h_A(\mathbf{x}) = \sum_{i=1}^{m/n} \mathbf{a}_i \otimes \mathbf{x}_i$ ($\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_{m/n}) \in \mathbb{Z}^m$ and $\otimes$ means a product operator over the ring $\mathbb{Z}_q[x]/\langle f \rangle$).

They showed the following theorem.

Theorem A.1 ([20])**.** *Let $E = \mathrm{EF}(f, 3)$. Let $m > n \log q / \log 2d$ and $q > 2Edmn^{3/2} \log n$. Then for $\gamma = 8E^2 dmn \log^2 n$, if $\Lambda(f)$-SVP$^\infty_\gamma$ is hard in the worst case then $\mathcal{H}_{\mathcal{I}}(f, q, m, d)$ is collision resistant.*

Next, let $\mathcal{H}'_{\mathcal{I}}(f, q, m)$ be a restricted version of the above hash functions:

$$\mathcal{H}'_{\mathcal{I}}(f, q, m) = \left\{ h_A : \{0,1\}^{m(n)} \to \mathbb{Z}_{q(n)}[x]/\langle f \rangle \mid A = (\mathbf{a}_1, \ldots, \mathbf{a}_{m(n)/n}) \in (\mathbb{Z}_{q(n)}/\langle f \rangle)^{m(n)/n} \right\}.$$

Corollary A.2. *For any $m(n) = \Theta(n \log n)$, there exists $q(n) = \Theta(m\sqrt{n} \log n)$ and $\gamma = \Theta(m \log^2 n)$, such that, for a suitable polynomial f, if $\Lambda(f)$-SVP$^\infty_\gamma$ is hard in the worst case then $\mathcal{H}'_{\mathcal{I}}(f, q, m)$ is collision resistant.*

Finally, we introduce a restricted version of $\mathcal{H}_{\mathcal{I}}(f,q,m)$ for use in our identification scheme:

$$\mathcal{H}'_{\mathcal{I}}(f,q,m,w) = \left\{ h_A : \mathrm{B}(m,w) \to \mathbb{Z}_{q(n)}[x]/\langle f\rangle \mid A = (\mathbf{a}_1,\ldots,\mathbf{a}_{m(n)/n}) \in (\mathbb{Z}_{q(n)}[x]/\langle f\rangle)^{m(n)/n} \right\}.$$

We can consider $A = (\mathbf{a}_1,\ldots,\mathbf{a}_{m/n})$ as $\mathbf{A}' = [\mathrm{Rot}_f(\mathbf{a}_1),\ldots,\mathrm{Rot}_f(\mathbf{a}_{m/n})]$ and $\mathbf{y} = \sum_{i=1}^{m/n} \mathbf{a}_i \otimes \mathbf{x}_i$ as $\mathbf{y} = \mathbf{A}'\mathbf{x}$, where $\mathrm{Rot}_f(\mathbf{a}) = [\mathbf{a}, \mathbf{e}_1 \otimes \mathbf{a},\ldots,\mathbf{e}_{n-1}\otimes\mathbf{a}]$ and $\mathbf{e}_i$ denotes a vector whose i-th coordinate is 1 and other coordinates are 0. Thus, we can plug it into the extended version of Stern's scheme as the setup and key-generation algorithms.

The setup algorithm on input 1^n outputs $A = (\mathbf{a}_1,\ldots,\mathbf{a}_{m/n})$ from $\mathbb{Z}_q/\langle f\rangle$ independently and uniformly at random. The key-generation algorithm on input A, chooses a random vector $\mathbf{x} \in \mathrm{B}(m,w)$ uniformly at random, computes a vector $\mathbf{y} = \mathbf{A}'\mathbf{x}$, and outputs $(pk, sk) = (\mathbf{y}, \mathbf{x})$.

A.2 String Commitment Scheme

Using $\mathcal{H}'_{\mathcal{I}}(f,q,m)$, we also obtain a simple string commitment scheme if a suitable $f \in \mathbb{Z}[x]$ is irreducible polynomial in $\mathbb{Z}_q[x]$.

We first explain why we need the irreducibility of f in $\mathbb{Z}_q[x]$. We need to estimate a lowerbound of m for the statistical-hiding property as in the proof of Lemma 3.6. Suppose that f is reducible in $\mathbb{Z}_q[x]$. Even in this case, we can obtain a lowerbound $\Omega(n^2 \log q)$ of m using Theorem 4. 2 in [24]. However, this lowerbound loses the advantage (i. e. , compactness) of the Lyubashevsky-Micciancio construction since that makes the size of the hash function much larger. On the other hand, if we assume that f is irreducible in $\mathbb{Z}_q[x]$, then we obtain a much better lowerbound $\Omega(n \log q)$, which preserves the advantage of their construction.

Here, we say a polynomial f of degree n is strongly suitable for q if f is an suitable polynomial and irreducible in $\mathbb{Z}_q[x]$. For example, consider $f(x) = (x^n - 1)/(x-1) = x^{n-1} + x^{n-2} + \cdots + 1$. The polynomial $f(x)$ is irreducible polynomial in $\mathbb{Z}[x]$ if n is prime and used in [20, 30]. We note that if $q \bmod n$ is a primitive root of $\mathbb{Z}_n^*$ then $f(x)$ is irreducible in $\mathbb{Z}_q[x]$. Thus, we apply the following lemma to $\mathcal{H}'_{\mathcal{I}}(f,q,m)$ and obtain the statistical-hiding property of a string commitment scheme.

Lemma A.3. *Let q be a prime $q = q(n) = n^{O(1)}$ and m an integer such that $m = m(n) > 2n\log q$. Let $f \in \mathbb{Z}_q[x]$ of degree n be a strongly suitable polynomial for q. The statistical distance between $(\mathbf{a}_1,\ldots,\mathbf{a}_{m/n}, \sum_{i=1}^{m/n} \mathbf{a}_i \otimes \mathbf{x}_i)$ and the uniform distribution over the set $(\mathbb{Z}_q[x]/\langle f\rangle)^{(m/n)+1}$ is negligible in n.*

Proof. We bound the collision probability of two random variables $(\mathbf{a}_1,\ldots,\mathbf{a}_{m/n}, \sum_i \mathbf{a}_i \otimes \mathbf{x}_i)$ and $(\mathbf{a}'_1,\ldots,\mathbf{a}'_{m/n}, \sum_i \mathbf{a}'_i \otimes \mathbf{x}'_i)$, where the elements $\mathbf{a}_i, \mathbf{a}'_i \in \mathbb{Z}_q[x]/\langle f\rangle$ and $\mathbf{x}_i, \mathbf{x}'_i \in \{0,1\}^n$ are all chosen independently and uniformly at random from their respective sets. The collision probability is

$$\begin{aligned}\Pr[\text{Collision}] &= \Pr[\mathbf{a}_i = \mathbf{a}'_i \text{ for all } i] \cdot \Pr\left[\sum_i \mathbf{a}_i \otimes \mathbf{x}_i = \sum_i \mathbf{a}'_i \otimes \mathbf{x}'_i \mid \mathbf{a}_i = \mathbf{a}'_i \text{ for all } i\right] \\ &= \frac{1}{q^m} \Pr\left[\sum_i \mathbf{a}_i \otimes (\mathbf{x}_i - \mathbf{x}'_i) = \mathbf{0}\right].\end{aligned}$$

By Lemma A.4 below, the probability over the random choice of $\mathbf{a}_i$ that $\sum_i \mathbf{a}_i \otimes (\mathbf{x}_i - \mathbf{x}'_i) = \mathbf{0}$ equals to $1/\left|\langle \mathbf{x}_1 - \mathbf{x}'_1,\ldots,\mathbf{x}_{m/n} - \mathbf{x}'_{m/n}\rangle\right|$. We note that $\mathbb{Z}_q[x]/\langle f\rangle$ is a field and ideals in it are only $\langle \mathbf{0}\rangle$ and $\mathbb{Z}_q[x]/\langle f\rangle$. Thus, we have

$$\begin{aligned}\Pr[\text{Collision}] &= \frac{1}{q^m}\left(\frac{1}{q^n}\Pr[\langle \mathbf{x}_1 - \mathbf{x}'_1,\ldots,\mathbf{x}_{m/n} - \mathbf{x}'_{m/n}\rangle = \mathbb{Z}_q[x]/\langle f\rangle] + \Pr\left[\langle \mathbf{x}_1 - \mathbf{x}'_1,\ldots,\mathbf{x}_{m/n} - \mathbf{x}'_{m/n}\rangle = \langle \mathbf{0}\rangle\right]\right) \\ &= \frac{1}{q^m}\left(\frac{1}{q^n}\left(1 - \frac{1}{2^m}\right) + \frac{1}{2^m}\right) \\ &= \frac{1}{q^{m+n}}\left(1 + \frac{q^n - 1}{2^m}\right).\end{aligned}$$

By Lemma A.5 below, we have the statistical distance is at most $\sqrt{(q^n-1)/(4\cdot 2^m)}$. Hence, the assumption that $m > 2n\log q$ implies the above upperbound is negligible in n. □ □

Lemma A.4 (Lemma 4. 4 in [24])**.** *Let R be a finite ring, and $z_1,\ldots,z_m \in R$ a sequence of arbitrary ring elements. If $a_1,\ldots,a_m \in R$ are independently and uniformly distributed ring elements, then $\sum a_i\cdot z_i$ is uniformly distributed over the ideal $\langle z_1,\ldots,z_m\rangle$ generated by $z_1,\ldots,z_m$. In particular, for any $z_1,\ldots,z_m \in R$ and randomly chosen $a_1,\ldots,a_m \in R$, the probability that $\sum a_i\cdot z_i = 0$ is exactly $1/|\langle z_1,\ldots,z_m\rangle|$.*

Lemma A.5 ([18])**.** *Let V and V' be independent and identically distributed random variables taking values in a finite set S. If V and V' have collision probability $\Pr[V=V'] \le (1+4\epsilon^2)/|S|$, then the statistical distance between V and the uniform distribution over S is at most ϵ.*

For $A=(\mathbf{a}_1,\ldots,\mathbf{a}_{m/n})$, we define the commitment function on input $(s,r)=(\mathbf{s}_1,\ldots,\mathbf{s}_{m/2n},\mathbf{r}_1,\ldots,\mathbf{r}_{m/2n})$ as

$$\mathrm{Com}_A(s;r) := \sum_{i=1}^{m/2n} \mathbf{r}_i\otimes\mathbf{a}_i + \sum_{i=1}^{m/2n} \mathbf{s}_i\otimes\mathbf{a}_{i+m/2n}.$$

Now, we obtain the following lemma as in Lemma 3.6.

Lemma A.6. *For any $m(n)=\Theta(n\log n)$, there exists $q(n)=\Theta(m\sqrt{n}\log n)$ and $\gamma=\Theta(m\log^2 n)$, such that, $m(n) > 2n\log q$ and for a strongly suitable polynomial f for $q(n)$, if $\Lambda(f)$-SVP$_\gamma^\infty$ is hard in the worst case then Com_A is a statistically hiding and computationally binding string commitment scheme in the trusted setup model.*

Using the Merkle-Damgård technique [21, 8], we obtain the string commitment scheme whose commitment function is $\mathrm{Com}_A : \{0,1\}^*\times\{0,1\}^{m/2}\to\mathbb{Z}_q^n$ rather than $\mathrm{Com}_A : \{0,1\}^{m/2}\times\{0,1\}^{m/2}\to\mathbb{Z}_q^n$.

A.3 Identification scheme and Ad Hoc Identification Scheme

We obtain the ID scheme and AID scheme by combining the above setup and key-generation algorithms and the string commitment scheme with the extended version of Stern's scheme as in Section 4 and 5. One can prove the securities of the schemes in the same manner to the proof of Theorems 4.2 and 5.2.

Theorem A.7. *Let f be a polynomial and $E := \mathrm{EF}(f,3)$. Let $m=m(n)$, $q=q(n)$, and $w=w(n)$ be polynomially bounded functions such that $m > 2n\log q$, $q > 2Emn^{3/2}\log n$, and $q^n/|\mathrm{B}(m,w)|$ is negligible in n. Assume that f is a strongly suitable polynomial for q. Then for $\gamma = 8E^2mn\log^2 n$, if $\Lambda(f)$-SVP$_\gamma^\infty$ is hard in the worst case then the ID scheme which uses the above setup and key-generation algorithms and the above string commitment scheme is secure against impersonation under concurrent attacks.*

Sketch of proof: We show that if there exists an impersonator $\mathcal{I}$ which succeeds impersonation under concurrent attacks with non-negligible probability ϵ, there exists $\mathcal{A}$ that finds a collision $(\mathbf{z}_1,\mathbf{z}_2)$ for $\mathcal{H}'_{\mathcal{I}}(f,q,m)$.

Given $A=(\mathbf{a}_1,\ldots,\mathbf{a}_{m/n})$, $\mathcal{A}$ chooses a random secret key $\mathbf{x}\in\mathrm{B}(m,w)$ and compute $\mathbf{y}:=\mathbf{A}\mathbf{x}$, where $\mathbf{A}:=[\mathrm{Rot}_f(\mathbf{a}_1)\ \ldots\ \mathrm{Rot}_f(\mathbf{a}_{m/n})]$. $\mathcal{A}$ executes $\mathcal{I}$ on inputs $(\mathbf{A},\mathbf{y})$. We note that $\mathcal{A}$ can simulate the oracles Conv and Prov, since $\mathcal{A}$ has the secret key $\mathbf{x}$. $\mathcal{A}$ executes $\mathcal{I}$ three times with random challenges and a fixed random tape. Then, $\mathcal{A}$ obtains three transcripts $(Cmt^{(i)},Ch^{(i)},Rsp^{(i)},Dec^{(i)})$ for $i=1,2,3$ as the results of the interactions between $\mathcal{I}$ and $\mathcal{A}$. Note that $Cmt^{(1)}=Cmt^{(2)}=Cmt^{(3)}$ since $\mathcal{A}$ fixes the random tape to work $\mathcal{I}$. By the assumption, $\mathcal{A}$ obtain good transcript such that with non-negligible probability $Dec^{(i)}=(Dec_1^{(i)},\ldots,Dec_n^{(i)})$ are all 1 for every i. Then, $\mathcal{A}$ can find $\mathbf{x}'$ from $(\mathbf{A},\mathbf{y})$ or find $(s,\rho)\neq(s',\rho')$ such that $\mathrm{Com}_A(s;\rho)=\mathrm{Com}_A(s';\rho')$ by using the fact that $Cmt^{(1)}=Cmt^{(2)}=Cmt^{(3)}$. In the former case, we can show that $\mathbf{x}'\neq\mathbf{x}$ with probability at least $1/2$ as in the proof of Theorem 4.2. $\mathcal{A}$ outputs $(\mathbf{x},\mathbf{x}')$. Since $\mathbf{x},\mathbf{x}'\in\mathrm{B}(m,w)\subseteq\{0,1\}^m$, $\mathcal{A}$ indeed finds a collision for $\mathcal{H}'_{\mathcal{I}}(f,q,m)$. In the latter case, $\mathcal{A}$ computes $\mathbf{z}\neq\mathbf{z}'\in\{0,1\}^m$ from (s,ρ) and (s',ρ') such that $\mathrm{Com}_A(s;\rho)=\mathbf{A}\mathbf{z}$ and $\mathrm{Com}_A(s';\rho')=\mathbf{A}\mathbf{z}'$. Thus, $\mathcal{A}$ outputs $(\mathbf{z},\mathbf{z}')$ as a collision for $\mathcal{H}'_{\mathcal{I}}(f,q,m)$. □

Theorem A.8. *Let f be a polynomial and $E := \mathrm{EF}(f, 3)$. Let $m = m(n)$, $q = q(n)$, and $w = w(n)$ be polynomially bounded functions such that $m > 2n \log q / \log 2(w+1)$, $q > 2E(w+1)mn^{3/2} \log n$, and $q^n / |\mathrm{B}(m,w)|$ is negligible in n. Assume that f is a strongly suitable polynomial for q. Then for $\gamma = 8E^2(w+1)mn \log^2 n$, if $\Lambda(f)$-$\mathrm{SVP}_\gamma^\infty$ is hard in the worst case then the AID scheme which uses the above setup and key-generation algorithms and the above string commitment scheme is secure against impersonation under concurrent attacks.*

Sketch of proof: We show that if there exists an impersonator $\mathcal{I}$ which succeeds impersonation under concurrent chosen-group attacks with non-negligible probability ϵ, there exists $\mathcal{A}$ that finds a collision $(\mathbf{z}_1, \mathbf{z}_2)$ for $\mathcal{H}_I(f, q, m, w+1)$.

The algorithm $\mathcal{A}$, given input $A = (\mathbf{a}_1, \ldots, \mathbf{a}_{m/n})$, feeds A to the impersonator $\mathcal{I}$. Let $\mathbf{A} := [\mathrm{Rot}_f(\mathbf{a}_1) \ \ldots \ \mathrm{Rot}_f(\mathbf{a}_{m/n})]$. In the experiment, the impersonator $\mathcal{I}$ will call Init, Corr, Conv, and Prov. If $\mathcal{I}$ calls Init with input i, then $\mathcal{A}$ chooses $\mathbf{s}_i \in \mathrm{B}(m,w)$ at random, computes $\mathbf{y}_i := \mathbf{A}\mathbf{s}_i$, and returns $\mathbf{y}_i$ to $\mathcal{I}$. If $\mathcal{I}$ calls Corr with input i, Conv with inputs i, R, or Prov with inputs i, R, s, M_i, then $\mathcal{A}$ can simulate the oracle Corr, since $\mathcal{A}$ has a secret key $\mathbf{s}_i$ with respect to a public key $\mathbf{y}_i$.

At the end of the experiment, $\mathcal{I}$ will impersonate as a group which is specified by the set of public keys $R = (\mathbf{y}_1, \ldots, \mathbf{y}_l)$. Rewinding $\mathcal{I}$ three times, $\mathcal{A}$ obtain a collision (s, ρ) and (s', ρ') for the commitment scheme Com_A or a vector $\mathbf{x} = {}^t({}^t\mathbf{x}_1 \, {}^t\mathbf{x}_2)$ such that $[\mathbf{A}\, \mathbf{y}_1 \ \ldots \ \mathbf{y}_l]\mathbf{x} = \mathbf{0}$, where $\mathbf{x}_1 \in \{-1, 0, 1\}^m$ and $\mathbf{x}_2 \in \{-1, 0, 1\}^l$ and $\mathbf{x} \in \mathrm{B}'(m+l, w)$ as in the proof of Theorem 4.2.

In the former case, $\mathcal{A}$ computes $\mathbf{z} \neq \mathbf{z}' \in \{0,1\}^m$ such that $\mathrm{Com}_A(s; \rho) = \mathbf{A}\mathbf{z}$ and $\mathrm{Com}_A(s'; \rho') = \mathbf{A}\mathbf{z}'$. Hence, $\mathcal{A}$ outputs $(\mathbf{z}, \mathbf{z}')$ as a collision for $\mathcal{H}_I(f, q, m, w+1)$.

In the latter case, we have $\mathbf{A}\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{y}_i = \mathbf{0}$, that is, $\mathbf{A}\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{A}\mathbf{s}_i = \mathbf{0}$. Hence, we obtain that $\mathbf{A}(\mathbf{x}_1 + \sum_{i=1}^{l} x_{2,i}\mathbf{s}_i) = \mathbf{0}$. By the same argument as in the proof of Theorem 4.2, we have that $\mathbf{x}_1 + \sum_i x_{2,i}\mathbf{s}_i \neq \mathbf{0}$ with probability at least $1/2$. Recall that the numbers of $+1$ in $\mathbf{x}$ is w and that of -1 in $\mathbf{x}$ is 1. Thus we can split the vector $\mathbf{x}_1 + \sum_i x_{2,i}\mathbf{s}_i$ into two vector $\mathbf{z}_1$ and $\mathbf{z}_2$ such that $\mathbf{x}_1 + \sum_i x_{2,i}\mathbf{s}_i = \mathbf{z}_1 + \mathbf{z}_2$, $\mathbf{A}\mathbf{z}_1 = \mathbf{A}\mathbf{z}_2$, and $\mathbf{z}_1, \mathbf{z}_2 \in D(m, w+1)$. Hence, $\mathcal{A}$ outputs $(\mathbf{z}_1, \mathbf{z}_2)$ as a collision for $\mathcal{H}_I(f, q, m, w+1)$. □

B Tables

Experiment: $\mathbf{Exp}^{\text{imp-ca}}_{\mathcal{SI},\mathcal{I}}$	
Input:	n
Run:	1. $param \leftarrow \mathsf{SetUp}(1^n)$
	2. $(pk, sk) \leftarrow \mathsf{KG}(param)$
	3. $PS \leftarrow \emptyset$
	4. $St_{\mathsf{CP}} \leftarrow \mathsf{CV}(1^n, param, pk)^{\text{Prov}}$
	5. $(Tr, Dec) \leftarrow \mathbf{Run}[\mathsf{CP}(St_{\mathsf{CP}}) \leftrightarrow \mathsf{V}(param, pk)]$
Output:	Dec

Prover oracle: Prov	
Input:	s, M_{in}
Run:	1. If $s \notin PS$ then
	1-1. $PS \leftarrow PS \cup \{s\}$
	1-2. Pick a random coin ρ for P
	1-3. $St_{\mathsf{P}}[s] \leftarrow (param, sk, \rho)$
	2. $(M_{out}, St_{\mathsf{P}}[s]) \leftarrow \mathsf{P}(M_{in}, St_{\mathsf{P}}[s])$
Output:	M_{out}

Table 1: Experiment and Oracles for Definition 2.3.

Experiment: $\mathbf{Exp}^{\text{imp-cca}}_{\mathcal{AID},\mathcal{I}}$	
Input:	n
Run:	1. $param \leftarrow \mathsf{SetUp}(1^n)$
	2. $HU, CU, TU, PS \leftarrow \emptyset$
	3. $(R_t, St_{\mathsf{CP}}) \leftarrow \mathsf{CV}(1^n, param)^{\text{Init,Corr,Prov}}$
	4. If $R_t \notin HU$ then return 0;
	5. $TU \leftarrow R_t$
	5. $(Tr, Dec) \leftarrow \mathbf{Run}[\mathsf{CP}(St_{\mathsf{CP}})^{\text{Init,Corr,Prov}} \leftrightarrow \mathsf{V}(param, R_t)]$
Output:	Dec

User initiation oracle: Init		User corruption oracle: Corr	
Input:	i	Input:	i
Run:	1. If $i \in CU \cup HU \cup TU$ then return $\perp$	Run:	1. If $i \notin HU \setminus TU$ then return $\perp$
	2. $(pk[i], sk[i]) \leftarrow \mathsf{Reg}(param; \rho[i])$		2. $CU \leftarrow CU \cup \{i\}$
	3. $HU \leftarrow HU \cup \{i\}$		3. $HU \leftarrow HU \setminus \{i\}$
Output:	$pk[i]$	Output:	$\rho[i]$

Prover oracle: Prov	
Input:	i, R, s, M_{in}
Run:	1. If $pk[i] \notin R$ then return $\perp$
	2. If $i \notin HU \setminus TU$ then return $\perp$
	3. If $(i, R, s) \notin PS$ then
	3-1. $PS \leftarrow PS \cup \{(i, R, s)\}$
	3-2. Pick a random coin ρ for P
	3-3. $St_{\mathsf{P}}[i, R, s] \leftarrow (sk_i, R, \rho)$
	4. $(M_{out}, St_{\mathsf{P}}[i, R, s]) \leftarrow \mathsf{P}(M_{in}, St_{\mathsf{P}}[i, R, s])$
Output:	M_{out}

Table 2: Experiment and oracles for Definition 2.4.

Experiment: $\mathbf{Exp}^{\text{anon-fke}}_{\mathcal{AID},\mathcal{A}}$	
Input:	n
Run:	1. $param \leftarrow \mathsf{SetUp}(1^n)$
	2. $((pk_{i_0}, sk_{i_0}), (pk_{i_1}, sk_{i_1}), R, St_{\mathcal{A}}) \leftarrow \mathcal{A}(param)$.
	3. $b \leftarrow_R \{0, 1\}$.
	4. $b^* \leftarrow \mathbf{Run}[\mathsf{P}(param, R, sk_{i_b}) \leftrightarrow \mathcal{A}(St_{\mathcal{A}})]$
	5. If $b = b^*$ then $Dec := 1$. Otherwise $Dec := 0$.
Output:	Dec

Table 3: Experiment and oracles for Definition 2.5.