Efficient Statistical Zero-Knowledge Authentication Protocols for Smart Cards Secure Against Active & Concurrent Attacks

Mohammad Sadeq Dousti and Rasool Jalili

Data & Network Security Lab, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

{dousti@ce, jalili@}sharif.edu

Abstract

We construct statistical zero-knowledge authentication protocols for smart cards based on general assumptions. The main protocol is only secure against active attacks, but we present a modification based on trapdoor commitments that can resist concurrent attacks as well. Both protocols are instantiated using lattice-based primitives, which are conjectured to be secure against quantum attacks. We illustrate the practicality of our main protocol on smart cards in terms of storage, computation, communication, and round complexities. Furthermore, we compare it to other lattice-based authentication protocols, which are either zero-knowledge or have a similar structure. The comparison shows that our protocol improves the best previous protocol.

Keywords. Statistical Zero Knowledge; Authentication; Smart Cards; Post-Quantum Cryptography; Lattice-based Cryptography.

1 Introduction

Authentication protocols are ubiquitous in everyday computing. They are present when checking email, making monetary transactions, connecting to a mobile/wireless network, and so on. From one point of view, the authentication protocols can be divided into two broad categories. In one category, the protocol is executed over an untrusted infrastructure, and the parties carrying the authentication need not be physically present in a specific location. Authentication over the Internet (or other public networks) is the best example of this type. In the other category, the party to be authenticated must be present in a pre-specified location, and it is assumed that the infrastructure connecting him to an honest verifier is trusted (i.e., no eavesdropping on or tampering with the data in transit is possible). Authentication via smart cards, security tokens, badges, magnet stripes, and biometrics fall into the second category, though in this paper we only focus on authentication protocols that can be carried out by a processor (such as a smart card). For this reason, we pick smart cards as the *representatives* of this category.

There are a number of features unique to smart-card authentication protocols:

- There is usually a single session between the smart card and the reader.
- The authentication protocol does not need a notion of key exchange, as the infrastructure is trusted.
- The smart card has limited resources regarding the storage, computation, and communication.

- Once inserted into the reader, the smart card cannot communicate with the outside world (except for *contactless* smart cards, which might communicate with another device in close range).

A security concern regarding the authentication protocols is that of the malicious verifiers. A malicious verifier poses herself as an honest verifier, engages in the protocol, deviates from the protocol, and tries to gain knowledge about the secret stored on the smart card. While bilateral authentication protocols may help by aborting the protocol in case one of the parties fails to authenticate herself to another, it does not prevent partial leakage of information. The leakage might be undesirable for systems which require a high level of security.

The best workaround is to use zero-knowledge authentication protocols, which guarantee that the verifier learns nothing about the secret. However, this high level of security comes at a price: Zero-knowledge authentication protocols are often too resource intensive to be used in practice. On the contrary, this paper aims to demonstrate a zero-knowledge authentication protocol for smart cards, with many attractive properties:

- The round complexity of the protocol is near optimal. More specifically, the minimum number of passes for a zero-knowledge proof with negligible soundness error is shown to be 3 [1] (and 4 if the simulation is black-box), while our protocol has only 5 passes. Most zero-knowledge authentication protocols in the literature do not even have a constant number of rounds.

- The protocol has a significantly lower communication complexity than similar protocols. In practice, the communication complexity determines the round complexity as well: For instance, ISO/IEC 7816-4 defines the *Application Protocol Data Unit* (APDU), which is the communication unit between a smart card and the reader. An APDU can carry up to 255 bytes of data. Therefore, our protocol has a significantly lower *practical round complexity* than similar protocols, even if their *theoretical round complexity* is lower. See Section 4.3 for more information.

- The protocol is provably secure based on the standard cryptographic assumptions. Furthermore, we provide an *exact security* [2] analysis, which reveals the minimum level of security achievable with any choice of parameters.

- The protocol is based on general assumptions, such as the existence of commitment schemes and trapdoor one-way permutations. Therefore, it can be instantiated based on the specific needs of each environment.

- The protocol is statistically zero knowledge, meaning that it does not leak any knowledge about the secret, even to an infinitely powerful malicious verifier.

- The protocol constructs (commitment and trapdoor one-way permutation) are instantiated based on lattice problems, which are believed to resist quantum attacks.

- The lattice-based instantiation uses very simple operations, such as multiplying a matrix by a vector (while protocols based on the RSA or discrete logarithm require the costly modular exponentiation operations). Therefore, the computational cost of the protocol is very low.

- The protocol, as well as its lattice-based instantiation, are modified to resist concurrent attacks. While smart cards are *not* usually used in the concurrent setting, it is theoretically instrumental to consider this setting as well.

We stress that the proposed authentication protocol achieves a high-level of security (i.e., the statistical zero-knowledge property) while being reasonably efficient. In particular, there exist more efficient authentication protocols for smart cards which are not zero knowledge. However, it is both theoretically and practically appealing to construct zero-knowledge authentication protocols for smart cards. From the theoretical point of view, we will compare our protocol to other *lattice-based* zero-knowledge authentication protocols for smart cards, and show that the proposed protocol is superior in terms of computation and communication complexities, while essentially achieving the same round and storage complexities (see Section 4.3). From a practical standpoint, zero-knowledge protocols are recommended for environments with tight security requirements, such as the data centers or military bases. In this paper, we provide evidence that our zero-knowledge authentication protocol can be implemented on smart cards, thereby satisfying the needs of security-critical (and perhaps other) environments.

Remark 1. Following a preliminary version of this work [3], Boorghany and Jalili [4] implemented our protocol, as well as several recent lattice-based authentication protocols on a real smart card. Their result shows the practicality and superiority of the protocol proposed in this paper, and is discussed briefly in Remark 6. $\lhd$

1.1 Why Lattices?

In this paper, we picked a particular instantiation of our general protocol based on lattices. For us, the most appealing feature of lattices is that no quantum attacks are known against lattice problems, and research offers evidence that both quantum and ordinary attacks require exponential time to break lattice-based constructions (for instance, see [5] and the references thereof). This stands in sharp contrast to factorization or discrete logarithm problems, for which polynomial-time quantum algorithms exist [6]. Therefore, on the advent of quantum computers, many authentication protocols for smart cards will be rendered insecure, while our lattice-based protocol will not be affected.

Other major attractions of lattice-based cryptography are the existence of worst-case to average-case reductions between lattice problems, asymptotic efficiency, and simple matrix operations. The first attraction is theoretically appealing, while the latter two are practically significant. For instance, consider *asymptotic* efficiency: For 80 bits of security, RSA-1024 is used, which beats most known lattice-based encryptions in efficiency. In contrast, for 128 bits of security, RSA-3072 is used, which is not as efficient as several lattice-based encryptions [7]. For higher bits of security, lattice-based encryptions are much more efficient than most number-theoretic encryptions. Furthermore, simple matrix operations allow lattice-based primitives to be easily implemented. In contrast, number-theoretic constructions require more complex operations, such as the modular exponentiation used in RSA.

For these reasons, lattice-based cryptography is preferable on constraint devices such as smart cards, when a high-level of security is desired.

1.2 Contributions

The main contribution of this paper, as descried above, is to offer a general zero-knowledge protocol for smart-card authentication, and prove its exact security. We also provide a specific lattice-based instantiation, which resists quantum attacks.
Other contributions of this paper are as follows.

- We provide a formal model and a formal definition for smart-card authentication. The details of our model and definition are taken from several references, but we compare and consolidate them into a single definition (Definition 1).
- Using trapdoor commitments, we show how our general protocol can be modified to resist attacks in a more hostile environment.
- We construct the first lattice-based trapdoor commitment, as described in Section 5.1. This construction exploits the achievements in lattice-based cryptography in the past few years, especially the efficient construction of trapdoor functions and public-key encryptions.

- We prove a series of useful lemmas in the appendix, which might be of independent interest.

1.3 Organization

The rest of this paper is organized as follows: Section 2 introduces the preliminaries needed for the rest of the paper, and surveys the related work. Section 3 presents the statistical zero-knowledge authentication protocol, and proves its exact security. Section 4 instantiates the general constructions of the protocol with lattice-based primitives, and analyzes the practical efficiency of the instantiated protocol. Section 5 discusses how trapdoor commitments can be used to modify the general protocol, so that it remains secure when the adversary can mount concurrent attacks on the protocol. It also instantiates the trapdoor commitments using lattice-based constructions. Section 6 concludes the paper, and provides future directions to improve the work.

This paper has an appendix, which is separated from the main text to improve the clarity, and so that the reader can focus on the main ideas of the paper. It defines the standard notions in cryptography, such as trapdoor one-way permutations, commitments, statistical distance, zero-knowledge protocols, and lattice-based problems. It also provides some useful lemmas which might be of independent interest.

2 Preliminaries and Related Work

In this section, we first define the main notations used throughout the paper, and then present a formal model and definition for smart-card authentication. Finally, we survey the papers in the area of lattice-based authentication.

Fairly standard definitions are omitted from this section, but are mentioned in Appendix A for self containment. The reader familiar with cryptography can safely skip this appendix, but we recommend to at least skim over Appendix A to get familiarized with the names and conventions we used for various cryptographic constructions.

2.1 Notation

We use the following general abbreviations: PPT for probabilistic polynomial time, ZK for zero knowledge, and SZK for statistical zero knowledge.

A function is called *negligible*, if it vanishes faster than the reciprocal of any positive polynomial. A function is *overwhelming*, if it is at most negligibly less than 1. The notation $e \leftarrow_R S$ corresponds to selecting e uniformly at random from the (finite) set S. For a random variable X, let $[X]$ denote the *support* of X. That is,

$$[X] = \{x \mid \Pr[X = x] > 0\} \ .$$

The function $\lg(\cdot)$ indicates the logarithm to the base two. The concatenation and XOR operators are denoted by "comma" and $\oplus$, respectively. For a string x, we use $|x|$ to indicate its length. Similarly, if S is a set, $|S|$ indicates its cardinality.

We denote by $\langle A, B\rangle$ a protocol between A and B. Moreover, $\langle A, B\rangle(x)$ denotes the same protocol when the common input of A and B is x. If either of the parties have a private input, that input is written in parenthesis next to its name. For instance, $\langle A(y), B\rangle(x)$ is the protocol where A has private input y. Finally, subscripting r to the name of a party means that we fixed the randomness of that party to r.

We typeset matrices (resp. vectors) by bold-face uppercase (resp. lowercase) Latin letters. For $p \geq 1$, the p-norm of a vector $\mathbf{v} = (v_1, \ldots, v_n)$ is denoted by $\|\mathbf{v}\|_p \overset{\text{def}}{=} \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}$. Notice that $\|\mathbf{v}\|_\infty = \max_i |v_i|$. In the special case of Euclidean (or ℓ^2) norm, we may simply use $\|\mathbf{v}\|$ instead of $\|\mathbf{v}\|_2$.

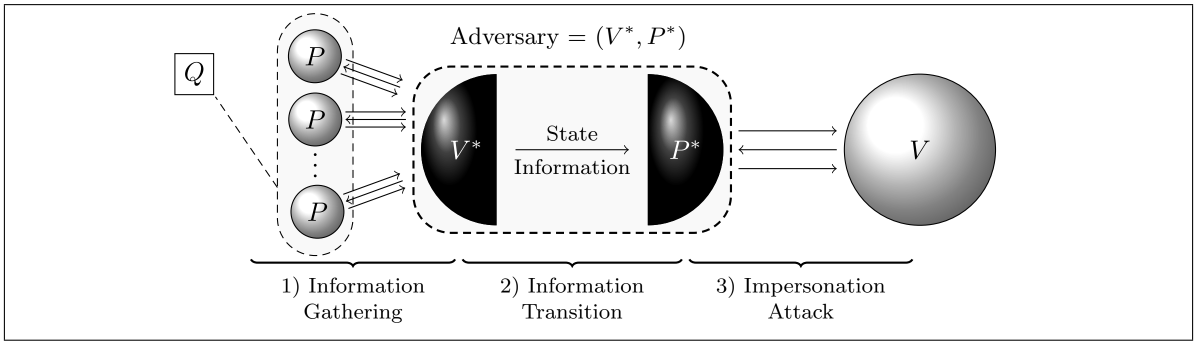

Figure 1: The smart-card authentication model.

2.2 Authentication: Model and Definition

In order to prove the security properties of cryptographic constructions, we need a *security model* and a *security definition*. The security *model* defines aspects such as the computational restrictions on the parties and the adversary, as well as how they communicate during the execution of the cryptographic construction. The model can be very general, and may be shared by several functionalities (see [8] as an example). The security *definition*, however, is specific to the functionality under investigation. It defines what it means for the functionality to be secure *within a particular model*. In many occasions, the security definition first defines a "winning condition" for the adversary, and then defines the cryptographic construction to be secure if the advantage of the adversary in winning is only negligible.

If the cryptographic construction is rather simple, the model and the definition may be unified together [9]. However, for complex constructions, there must be a separate model and a separate definition. This is especially the case for the authentication protocols, where the complexity of modeling/definition is so high that many authentication models and definitions were proposed. To name just a few, see [10–19]. See also [20–23] for a comparison of these works.

Since the focus of this paper is on efficient *zero-knowledge* authentication protocols, we have to choose a proper model which allows the authentication protocol to satisfy both efficiency and zero-knowledge properties. The aforementioned papers try to model an environment similar to the Internet, where the adversary is free to *concurrently* execute many versions of the authentication protocol. The zero-knowledge property is not necessarily preserved under the concurrent executions [1]. Moreover, it is known that only round-*inefficient* zero-knowledge protocols are concurrently secure. More precisely, only protocols with round complexity $\tilde{\Omega}(\log n)$ can be (black-box) zero knowledge [24].

Several works try to augment the standard model, and offer *constant*-round zero-knowledge protocols. This includes the *timing model* [25], the *bare public-key model* [26], and the *non–black-box zero knowledge* [27]. However, to the best of our knowledge, no *efficient* zero-knowledge authentication protocols were designed and implemented in these models. Moreover, they have no formal definition for authentication protocols.

Another approach, and the one we will take in this paper, is to model the adversary in a physically restricted way. In this approach, the adversary cannot *simultaneously* communicate with the honest prover and the honest verifier [28]. (We assume that the prover is the entity trying to authenticate himself to the verifier.) The model is known as the *smart-card authentication model*, since it was first developed with the resource restrictions of smart cards in mind. Moreover, it has no notion of *key exchange*, which is fitted to the case of smart cards, where it is *physically* guaranteed that the adversary cannot "hijack" the session after an honest party is authenticated. Finally, the model only supports *unilateral* authentication, where only the first party proves his identity to second one, but not vice versa.

The smart-card model was once the prevalent model for authentication protocols [28–45]. With the recent advent of lattice-based authentication protocols, it has gained momentum again [46–52].

Let us briefly describe the smart-card authentication model. All parties in the model are probabilistic polynomial-time (PPT) interactive Turing machines. The honest prover and the honest verifier are denoted by P and V, respectively. The adversary $\mathcal{A}$ is composed of a pair of colluding machines (V^*, P^*), where P^* and V^* play the roles of the cheating prover and the cheating verifier, respectively. The communication model is illustrated in Figure 1. As shown in the figure, the adversary attacks the protocol in *three* stages: **information gathering**, **information transition**, and **impersonation**. In the *information gathering* stage, the adversary plays the role of a cheating verifier V^*, and interacts with the honest prover P for some polynomial number of times. In this stage, V^* tries to gather from P as much information as she can. Let us denote the *state* of V^* after it halts by st. In the *information transition* stage, st is given to the cheating prover P^*. Finally, in the *impersonation* stage, $P^*(st)$ tries to misrepresent herself (as P) to the honest verifier V. It is very important to note that the smart-card authentication model does not allow P^* to communicate with P. In other words, V^* halts before the stage two (and therefore, the stage three) starts. This modeling effectively prevents attacks such as the Mafia Fraud [53] or the Chess Grandmaster Problem [54], since both attacks require the cheating prover to be "wired" to the honest prover.

The attack $\mathcal{A}$ mounts on $\langle P, V\rangle$ is categorized based on the type of interaction between V^* and P in the information gathering stage. The categories, in increasing order of strength, are as follows:

- **One-shot**: P^* attempts to impersonate to V, given only the common input. In other words, P^* does not receive any information from V^*.
- **Passive:** V^* does not actually interact with P, and merely eavesdrops on honest protocol executions.
- **Honest verifier:** V^* follows the prescribed program of V while interacting with P.
- **Active:** V^* interacts with P *sequentially.* In other words, V^* does not start interacting with a new copy of P if it is already in the middle of interaction with another copy of P. See [44–46] for example uses of this terminology.
- **Concurrent:** V^* is free to concurrently interact with a polynomial number of P's.
- **Resetting:** V^* has oracle access to each copy of P. In particular, not only can V^* run them concurrently, but also it can reset (or rewind) each copy to a previous state. This attack was first defined in [26] for zero-knowledge protocols. [55] applies the attack to authentication protocols.

As pointed out in the beginning of this section, the zero-knowledge property is not necessarily preserved under concurrent attacks. However, (auxiliary-input) zero-knowledge is preserved under active (i.e., sequential) attacks [56]. Therefore, an (auxiliary-input) zero-knowledge protocol $\langle V, P\rangle$ is as secure under *active* attacks as it is under *one-shot* attacks.[1]

Now that we described the model, let us define the syntax and semantics of authentication protocols in this model. Syntactically, an authentication protocol consists of a triple (G, P, V), where G is a PPT algorithm, and P and V are PPT interactive Turing machines. On input 1^n, the algorithm G generates a pair (x, y). Then, y is handed over to P as the private input, x is set as the common input, and the protocol $\langle V, P(y)\rangle(x)$ is executed. After the exchange of at most a polynomial (in n) number of messages, V always halts, and outputs either 1 ("accept") or 0 ("reject"). Let us denote the verifier's output by $[\![\langle V, P(y)\rangle(x)]\!]$, which might be different from a single bit in case we are dealing with a malicious verifier. Next, we define what it means for a protocol to be a secure authentication protocol.

Definition 1 (Secure Authentication Protocol). A triple (G, P, V) is called a *secure authentication protocol in the smart-card model* if the following holds:

[1]**Usage note.** In the cryptography community, the term "zero knowledge" implies "auxiliary-input zero knowledge." Consequently, we drop the "auxiliary-input" qualifier, and only speak of zero-knowledge protocols.

1. **Completeness:** For all n and for any pair $(x, y) \in [G(1^n)]$, the verifier V of the honest interaction $\langle V, P(y)\rangle(x)$, accepts with overwhelming probability (in n).

2. **Soundness:** For all $c > 0$, and for any PPT adversarial coalition $\mathcal{A} = (V^*, P^*)$, and for large enough n, the quantity:

$$\mathbf{Adv}^{\text{Attack}}_{\mathcal{A},(G,P,V)}(n) \stackrel{\text{def}}{=} \Pr\left[[\![\langle V, P^*(st)\rangle(x)]\!] = 1 \,\middle|\, (x, y) \leftarrow G(1^n),\ st \leftarrow [\![\langle V^*, Q(y)\rangle(x)]\!] \right] , \quad (1)$$

is less than n^{-c}. Here, the probability is taken over the coin tosses of G, Q, V, and $\mathcal{A} = (V^*, P^*)$. The behavior of V^* and the interactive function $Q(x, y)$ varies depending on the Attack type:

- **One-shot:** V^* simply outputs the empty string as her state.
- **Passive:** Upon each invocation, $Q(x, y)$ outputs a transcript of the honest execution $\langle V, P(y)\rangle(x)$ with fresh randomness.
- **Honest-verifier:** V^* and $Q(x, y)$ behave as V and $P(x, y)$, respectively.
- **Active:** $Q(x, y)$ keeps a flag F, indicating whether an instance of $P(x, y)$ is currently running (initially, $F = 0$). Q accepts the special message "New". Upon receiving this message, Q replies with $\perp$ if $F = 1$. Otherwise, F is set to 1, and $Q(x, y)$ will behave like $P(x, y)$ with fresh randomness. If $P(x, y)$ halts, the flag F will be set to 0 again.
- **Concurrent:** $Q(x, y)$ keeps a set ID (initially empty), and accepts the special message New(id). Upon receiving this message, Q checks whether $id \in ID$, and replies with $\perp$ if this is the case. Otherwise, Q sets $ID \leftarrow ID \cup \{id\}$, and spawns a new instance of $P(x, y)$ with fresh randomness and id as identifier—denoted $P_{id}(x, y)$. V^* can communicate with $P_{id}(x, y)$ by prefixing each message with id.
- **Resetting:** This attack is similar to the previous one, but in addition $Q(x, y)$ accepts the message Reset(id). Upon receiving this message, Q checks whether $id \in ID$, and replies with $\perp$ if this is not the case. If $id \in ID$, Q resets $P_{id}(x, y)$ to its initial state, without refreshing P_{id}'s randomness. ○

Remark 2. The term "soundness" in the definition of an authentication protocol should not be confused with the same term used in the definition of zero-knowledge proofs (or, *cryptographic proofs*, in general). Note that the soundness in Definition 1 is with respect to the *smart-card authentication model*, where the interactions involves the four parties P, V, P^*, and V^*. On the other hand, the soundness in the definition of cryptographic proofs merely involves P^* and V. Moreover, the soundness in Definition 1 is with respect to some input distribution $G(1^n)$, while the soundness in cryptographic proofs is with respect to all admissible inputs.

We remark that the meaning of the term "completeness" remains the same in both authentication protocols and cryptographic proofs. ◁

2.3 Lattice-Based ZK Proofs & Authentication: Related Work

In this section, we briefly survey zero-knowledge proofs and authentication protocols based on lattices. Appendix A.5 studies the necessary terminology to understand lattice problems.

The first lattice-based ZK authentication protocol was proposed by Micciancio and Vadhan [57], whose security was based on the hardness of GapCVP. In their protocol, the prover and the verifier share a lattice generated by long, highly non-orthogonal basis vectors, and the prover's public key is a fixed point Y outside the lattice. The prover then tries to convince the verifier that he knows a lattice point X "near" Y.

Micciancio–Vadhan's protocol is statistical zero knowledge (SZK), so even an infinitely powerful malicious verifier cannot gain any knowledge from the prover, except with negligible probability. Moreover, their protocol does not need a short-and-nearly-orthogonal basis, because the prover is not going to solve CVP. He merely knows one problem-solution pair (Y, X), generated by himself. Because the soundness error of the base protocol is $\frac{1}{2}$, it must be repeated super-logarithmically in order to obtain a protocol with negligible soundness error. This repetition cannot be performed in parallel, since otherwise the zero-knowledge property would collapse.

Lyubashevsky [46] presented a 3-pass authentication protocol based on the hardness of the SVP in *all* lattices, and a more efficient protocol based on the hardness of SVP in *ideal* lattices. Both protocols do not feature perfect completeness, and neither one is zero knowledge.

Kawachi *et al.* [47] introduced another authentication protocol based on the worst-case hardness of GapSVP. This protocol is a version of Stern's authentication protocol [58]. It assumes the availability of a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generated by a trustee (here, n, m, and q are properly chosen integers). The prover of the authentication protocol has a public key $\mathbf{y} \in \mathbb{Z}^n$, and proves that he knows a secret $\mathbf{x} \in \{0,1\}^m$, such that $\mathbf{y} = \mathbf{A}\mathbf{x} \pmod q$. The approximation factor used in their work was smaller than those of Micciancio–Vadhan [57] and Lyubashevsky [46], so the security is based on a weaker assumption. The base protocol of [47] is statistical zero knowledge. It requires super-logarithmic repetitions to make the soundness error negligible. To reduce the round complexity, the authors suggest to run the base protocol in parallel. The parallel version is no longer ZK. Nonetheless, Kawachi *et al.* prove that it is a secure authentication protocol.

In another attempt, Lyubashevsky [48] presented an authentication protocol based on the worst-case hardness of SVP and using lattice-based hash functions. He argues that, while the proposed lattice-based authentication protocols are asymptotically as efficient as number-theoretic ones, their concrete performance is not much good, due to the fact that the former treats "challenge bits" individually, while the latter treats them as a whole. He then tries to improve some of the previous lattice-based authentication protocols by exploiting the limited algebraic structure of the underlying lattice. The base protocol of [48] has a **completeness** error of $1 - e^{-1} \approx 0.63$, and therefore some parts of it are repeated (in parallel) to achieve almost perfect completeness. The protocol is not zero-knowledge, but is proved to be secure under active attacks. It has a communication cost significantly lower than Micciancio–Vadhan and Kawachi *et al.* protocols.

Xagawa and Tanaka [49] proposed two statistical zero-knowledge proofs of knowledge for NTRU encryption [59], based on a variant of Stern's authentication protocol [47, 58]. The first protocol is for the "knowledge of secret key," while the other is for "the knowledge of plaintext." The former protocol can be used directly for authentication. The base protocol has a soundness error of $\frac{2}{3}$, and should be repeated super-logarithmically.

Cayrel *et al.* [50] introduced another authentication protocol, based on the code-based authentication protocol of Cayrel and Véron [51], which in turn is based on Stern's protocol [58]. The assumption used here is the hardness of the SIS problem (as in [47]), which is milder than the assumption of Lyubashevsky [48]. However, since the soundness error of this protocol is smaller than both [47] and [48], it achieves the same level of security in fewer rounds.

Silva *et al.* [52] followed [50], and built a similar authentication protocol based on the hardness of the SIS problem. The authentication protocol consists of the repetition of a 5-pass base zero-knowledge protocol with soundness error close to $\frac{1}{2}$.

Finally, Silva *et al.* [60] presented two zero-knowledge authentication protocols based on the hardness of LWE. The first protocol has a soundness error of $\frac{2}{3}$, while this error is $\frac{1}{2}$ for the second protocol. Therefore, neither protocol can achieve zero-knowledge property and negligible soundness error with sub-logarithmic repetitions.

3 A Statistical Zero-Knowledge Authentication Protocol Secure Against Active Attacks

In this section, we exhibit a five-pass statistical zero-knowledge (SZK) authentication protocol. The number of passes is *almost optimal*: Goldreich and Krawczyk [61] proved that three-pass black-box zero-knowledge proofs (with negligible soundness error) exist only for **BPP** languages. Itoh and Sakurai [62] generalized this result to the case of *proofs of knowledge*. Katz [63] demonstrated further restrictions on the class of languages having four-pass, black-box zero-knowledge proofs.

Our protocol is inspired by the "proof of computational power" of Okamoto *et al.* [64], but it is far more efficient. One reason is that their protocol uses bit commitments, which are much slower than ordinary commitments, and have a high communication complexity. A close inspection of the proof in [64] shows that the **bit** commitments cannot be simply replaced with **general** commitments, without modifying the protocol. This is because the extractor in the proof of [64] uses the following property: Given the commitment $c \leftarrow \textsc{Com}(x\,;r)$ and the randomness r, one can efficiently compute x. All *bit* commitment schemes satisfy this property, since x is a single bit, and it can be efficiently verified whether $c = \textsc{Com}(0\,;r)$ or $c = \textsc{Com}(1\,;r)$. However, *general* commitment schemes do not necessarily satisfy this property, and therefore the proof of [64] does not carry over for the case of **general** commitment schemes.

Another important difference between our protocol and that of Okamoto *et al.* [64] is that we use trapdoor permutations, while they use one-way functions with *few* preimages[2]. Utilizing trapdoor permutations has the following *practical* advantages: (1) Inverting a (general) one-way function with *few* preimages requires super-polynomial time provers, while inverting a trapdoor permutation can be achieved by a polynomial-time prover who uses the trapdoor. (2) The prover of the protocol in [64] must commit to all possible preimages and send the commitments to the verifier, which is wasteful of bandwidth. In contrast, we use a trapdoor permutation, which has a single preimage for each image (due to its one-to-one nature).

Finally, we prove that our protocol is a secure authentication against active attacks. Later, in Section 5, we further extend the protocol to remain secure against concurrent attacks.

3.1 Protocol Description

Our protocol is listed in Protocol 1. Any trapdoor permutation (TDP) and any non-interactive statistically-hiding commitment can be used to instantiate the protocol. Please note that the corresponding definitions and notation are provided in Appendices A.1 and A.3, respectively.

The description of the commitment scheme $\mathsf{desc}(\textsc{Com}_n)$ is included as the "public parameter." This means that the prover P and the verifier V both have access to it, and know that it is selected honestly. There are many approaches to this end, several of which are as follows:

1. P and V have already agreed upon $\mathsf{desc}(\textsc{Com}_n)$ through an out-of-band mechanism. The most common way is to consider V as a server, and P as a client: The server chooses the public parameter as well as the credentials of each client, and delivers them to each client via an out-of-band mechanism (such as a token or a smart card).

2. The public parameter is selected via the so-called *common reference string* (CRS) [65].

3. A *trusted third party* (TTP) selects the public parameter. For instance, in the *public key infrastructure* (PKI) model, the TTP is a *certificate authority* (CA). Each CA can embed the public parameter in the public key of its clients, or more efficiently, in its own public key.

[2] That is, the size of preimages for any element in the range of the function is polynomial in the security parameter. See [64] for more information.

Public parameter: Description of a non-interactive statistically-hiding commitment scheme, denoted $\textsc{Com}_n$, chosen according to $\textsc{GenC}(1^n)$.

Prover's public key: Description of a TDP, denoted π_n, chosen according to $\textsc{GenP}(1^n)$.

Prover's private key: The trapdoor t_n associated with π_n.

Protocol Description

1. P picks an n-bit random string u_n, chooses $\rho_n \leftarrow \textsc{Rnd}_{\ell(n)}$, and sends V a commitment to u_n by computing $c_n \leftarrow \textsc{Com}_n(u_n\,;\rho_n)$.
2. V picks a random element x_n in $\mathsf{dom}(\pi_n) \subseteq \{0,1\}^n$ using the domain sampling algorithm: $x_n \leftarrow \textsc{Samp}(\mathsf{desc}(\pi_n))$.

   She then evaluates π_n on x_n by $y_n \leftarrow \textsc{Eval}(\mathsf{desc}(\pi_n), x_n)$, and sends y_n to P.
3. If $y_n \notin \{0,1\}^n$, P outputs a special symbol $\perp$ and aborts.

   P inverts y_n using the trapdoor: $w_n \leftarrow \textsc{InvP}(\mathsf{desc}(\pi_n), t_n, y_n)$.

   P sends V the value u_n of step 1 if $w_n = \perp$, and the value $\sigma_n \leftarrow u_n \oplus w_n$ otherwise.
4. V sends P a value z_n equal to x_n.
5. If $z_n = w_n \neq \perp$, then P will send V the value ρ_n. Otherwise, P outputs a special symbol $\perp$ and aborts.

Verification Step: V computes $v_n \leftarrow \sigma_n \oplus x_n$, and accepts *iff* $c_n = \textsc{Com}(v_n\,;\rho_n)$.

Protocol 1: A statistical zero-knowledge authentication protocol based on any TDP and any non-interactive statistically-hiding commitment scheme. Notice that if P and V act honestly, then $z_n = w_n = x_n$ and $v_n = u_n$; otherwise, they might be different.

The prover P has the description of a TDP in his *public key*, and the associated trapdoor in its *private key*. The notions of public and private keys are not to be confused with the public key *encryption* schemes. Moreover, although the most common way to securely distribute public keys is via PKI, they can be securely distributed via out-of-band mechanisms in small- or medium-sized environments. Let us denote the public parameter and the prover's public key collectively by i_n.

The prover P of Protocol 1 can always prove his ability to invert the TDP, using the associated trapdoor. Therefore, the protocol has *perfect completeness*. On the other hand, we will prove that no adversary can impersonate the prover, except with negligible probability (assuming the security of the TDP and the commitment). Therefore, the protocol has *negligible soundness error*.
Let us now describe each step of Protocol 1 in detail:

- In step 1, the prover commits to some random value u_n, which is later used in step 3 as a one-time pad key. This step is not necessary to prove the zero-knowledge property (Section 3.2). However, without this step, the proof of the security of authentication does not go through (Section 3.3).
- In step 2, the verifier sends the prover a challenge y_n in the range of π_n, whose pre-image x_n is known to him.
- In step 3, the prover makes the syntactic check $y_n \in \{0,1\}^n$, and aborts otherwise.

  If the check succeeds, he computes w_n, the inverse of y_n under π_n. The inversion algorithm may or may not succeed. In the latter case, it returns a special symbol $\perp$. This is the case if (an adversarially-chosen) y_n is not in the range of π_n. Since deciding whether an element belongs to the range of a function is not necessarily efficient, the prover cannot simply abort the protocol;

otherwise, some knowledge might leak to the malicious verifier. Let us illustrate this point with an example.

Assume that $\pi_n \colon QR_m \to QR_m$ is a the Rabin's TDP (see the end of Appendix A.1 for the notation and definition of Rabin's TDP). It is well known that deciding whether a given number is a quadratic residue is a hard problem [66]. Therefore, there is no efficient algorithm to decide whether y_n belongs to $\mathsf{range}(\pi_n)$. Now, consider a prover that aborts the protocol if he receives a quadratic non-residue, and continues otherwise. Such prover will leak knowledge about whether y_n belongs to QR_m, and therefore the protocol will not be zero knowledge.

To foil this attack, the prover will simply continue the protocol if the inversion of y_n under π_n fails: If $w_n = \perp$, the prover sends u_n to V; otherwise, he sends $u_n \oplus w_n$ to V.

Remark 3. If π_n is such that it is efficiently decidable whether a given value belongs to $\mathsf{range}(\pi_n)$, we can modify the protocol so that P immediately rejects if $y_n \notin \mathsf{range}(\pi_n)$. This change will result in a more efficient protocol, and simplifies proofs of security. $\lhd$

- In step 4, the verifier sends the value z_n, supposed to be equal to the value $x_n = \pi_n^{-1}(y_n)$ he picked at step 2 (however, a cheating verifier may opt to send a value $z_n \neq x_n$). Note that if the value y_n sent at step 2 was not in the range of π_n, the (cheating) verifier would not be able to find a proper z_n. In this case, for whatever value she sends at this step, the prover will abort the protocol in the next step.

- In step 5, the prover first checks whether the value received from the verifier is valid, and if so, decommits c_n. Otherwise, the prover will output $\perp$ and abort the protocol.

In the verification step, the verifier checks whether the prover has acted honestly. This is done by finding the randomness in the one-time pad, and verifying whether c_n is properly opened.

3.2 Zero-Knowledge Property

Let $(\mathsf{desc}(\pi_n), t_n) \in [\textsc{GenP}(1^n)]$, and $\mathsf{desc}(\textsc{Com}_n) \in [\textsc{GenC}(1^n)]$. Define

$$R_n \stackrel{\text{def}}{=} \{(i_n, t_n) \mid i_n = (\mathsf{desc}(\pi_n), \mathsf{desc}(\textsc{Com}_n))\} \ ,$$

and let $R \stackrel{\text{def}}{=} \bigcup_{n\in\mathbb{N}} R_n$. In this section, we prove the following theorem through a series of lemmas (see Appendix A.4 for related definitions):

Theorem 1. *Protocol 1 is statistical zero-knowledge (SZK) for P on R. Moreover, the zero-knowledge simulator rewinds V^* at most once, and the statistical distance between the simulated and real views is at most the hiding gap of the commitment scheme (as defined by Equation 17 in Appendix A.3).*

Notice that since the definition of zero knowledge (Definition 5) quantifies over all inputs in R, this property must hold regardless of the distribution used to choose the input. More specifically, Theorem 1 holds regardless of the randomness used by $\textsc{GenC}(1^n)$ and $\textsc{GenP}(1^n)$ to generate $\textsc{Com}_n$ and π_n, respectively. Put differently, the theorem holds for *any* statistically-hiding commitment and *any* TDP.

Proof. Let S be SZK simulator described by Algorithm 1. Notice that we used "primes" to connect the variables in the simulation to those in the real execution. For instance, the variable u'_n corresponds to u_n. Moreover, note that the simulator runs in probabilistic polynomial time, and it rewinds the verifier at most once: If the simulation does not halt after step 5, S will rewind the verifier exactly once. Otherwise, no rewinding takes place.

To prove that the output of S is statistically close to the view of V^* in the real execution, we will proceed in stages. That is, we prove that the verifier's real and simulated views are statistically close

Input (i_n)**:** Public parameter $\mathsf{desc}(\textsc{Com}_n)$, and prover's public key $\mathsf{desc}(\pi_n)$.

1. Commit to a random n-bit value u'_n by choosing $\rho'_n \leftarrow \textsc{Rnd}_{\ell(n)}$, and computing $c'_n \leftarrow \textsc{Com}_n(u'_n\,;\rho'_n)$.
2. Run $V^*_{r'}$ as a black box, to get the challenge: $y'_n \leftarrow V^*_{r'}(i_n, c'_n)$.
3. If $y'_n \notin \{0,1\}^n$, `OUTPUT` $(i_n, r', c'_n, \perp)$ and halt.

   Let σ'_n be a random n-bit value.
4. Let $z'_n \leftarrow V^*_{r'}(i_n, c'_n, \sigma'_n)$.
5. If $y'_n \neq \textsc{Eval}(\mathsf{desc}(\pi_n), z'_n)$, then `OUTPUT` $(i_n, r', c'_n, \sigma'_n, \perp)$ and halt.

   ▶ Otherwise, let $w' \leftarrow z'_n$, which in turn equals $\pi_n^{-1}(y'_n)$. **Rewind** $V^*_{r'}$ as follows:

   5.1 Let $\sigma''_n \leftarrow u'_n \oplus w'_n$ and $z''_n \leftarrow V^*_{r'}(i_n, c'_n, \sigma''_n)$.

   5.2 If $z''_n \neq w'_n$, then `OUTPUT` $(i_n, r', c'_n, \sigma''_n, \perp)$ and halt.

   5.3 `OUTPUT` $(i_n, r', c'_n, \sigma''_n, \rho'_n)$.

Algorithm 1: The algorithm for the SZK simulator S of Protocol 1.

upon receiving each message. Let P_1, P_2, and P_3 be the random variables describing the verifier's view upon receiving the first, second, and third prover's message, respectively. Similarly let S_1, S_2, and S_3 be the random variables describing the verifier's view upon receiving the first, second, and third simulated message, respectively. Below, we will prove that $\Delta(P_i\,;S_i)$ is exponentially small for $i \in \{1,2,3\}$.

To simplify the proof, we will use "helper" random variables as well. These random variables are implicitly defined by the protocol and the simulation. For instance, picking a random n-bit string u_n corresponds to sampling from the uniform distribution U_n on n-bit strings. Continuing in this manner, we denote by X the random variable corresponding to the variable x. As an example, consider the random variable Y'_n, which corresponds to y'_n defined in step 2 of the simulation. We stress that U_n, U'_n, U''_n, and U'''_n are independent uniform distributions on n-bit strings, and R and R' are random variables whose support is infinitely long bit strings, where each bit is chosen uniformly and independently. Moreover, notice that i_n is a fixed string, and not a random variable.

Stage 1. The prover computes $C_n = \textsc{Com}_n(U_n)$, and the simulator computes $C'_n = \textsc{Com}_n(U'_n)$. Since $\Delta(U_n\,;U'_n) = 0$, An application of Fact 2 of Appendix A.2 shows that $\Delta(C_n\,;C'_n) = 0$. Furthermore, because R and C_n are independent, and R' and C'_n are independent, we apply Fact 3 of Appendix A.2 to show that:

$$\Delta(P_1\,;S_1) \stackrel{\text{def}}{=} \Delta\big((i_n, R, C_n)\,;(i_n, R', C'_n)\big) = \Delta(R\,;R') + \Delta(C_n\,;C'_n) = 0\ .$$

Stage 2. Let $V_2 = (i_n, \hat{R}, \hat{C}_n, \hat{\Sigma}_n)$ represent the verifier's current view, which might be either the real view $P_2 = (i_n, R, C_n, \Sigma_n)$ or the simulated view $S_2 = (i_n, R', C'_n, \Sigma'_n)$. Let f be the function that the verifier applies to its view to compute the challenge; i.e., $\hat{Y}_n \leftarrow f(i_n, \hat{R}, \hat{C}_n)$. If $\hat{Y}_n \notin \{0,1\}^n$, then $\hat{\Sigma}_n = \perp$. Since $R \sim R'$ and $C_n \sim C'_n$, Corollary 1 of Appendix A.2 shows that $Y_n \sim Y'_n$. Therefore, the probability that $\Sigma_n = \perp$ equals the probability that $\Sigma'_n = \perp$. This shows that if $\hat{Y}_n \notin \{0,1\}^n$, the random variables P_2 and S_2 are identically distributed.

In the rest, we implicitly assume that $\hat{Y}_n \in \{0,1\}^n$. Define $\hat{W}_n \leftarrow \pi_n^{-1}(\hat{Y}_n)$. Let E be the event that $W_n = \perp$ in the real execution. The random variable V_2 takes either of the following forms:

- $S_2 = (i_n, R', C'_n, U''_n)$
- $P_2|E = (i_n, R, C_n, U_n)$
- $P_2|\overline{E} = (i_n, R, C_n, U_n \oplus W_n)$

Define the permutation g on V_2 as follows: g is the identity permutation if E. Otherwise, it permutes V_2 as follows: $(i_n, \hat{R}, \hat{C}_n, \hat{\Sigma}_n) \xrightarrow{g} (i_n, \hat{R}, \hat{C}_n, \hat{\Sigma}_n \oplus \hat{W}_n)$. Notice that g satisfies the following properties:

- It maps S_2 to a random variable with *identical* distribution. More precisely, since U''_n is independent from W_n, we have $U''_n \oplus W_n \sim U'''_n$. Hence,

$$S_2 = (i_n, R', C'_n, U''_n) \xrightarrow{g} (i_n, R', C'_n, U'''_n) \sim S_2 \ .$$

- It maps $P_2|E$ to itself (because g is the identity permutation if E).
- It maps $P_2|\overline{E}$ to a random variable identically distributed with $P_2|E$.

According to Lemma 4 of Appendix A.2, $\Delta(P_2\,;S_2) = \Delta(g(P_2)\,;g(S_2)) = \Delta(P_2|E\,;S_2)$. Furthermore, since R is independent from (C_n, U_n) and R' is independent from (C'_n, U'_n), we can apply Fact 3 of Appendix A.2:

$$\begin{aligned}\Delta(P_2|E\,;S_2) &= \Delta\Big((R, C_n, U_n)\,;(R', C'_n, U''_n)\Big) \\ &= \Delta(R\,;R') + \Delta\Big((C_n, U_n)\,;(C'_n, U''_n)\Big) \ ,\end{aligned}$$

where $\Delta(R\,;R') = 0$ because $R \sim R'$. Consequently, the following relations hold:

$$\Delta(P_2\,;S_2) = \Delta\Big((C_n, U_n)\,;(C'_n, U''_n)\Big) = \frac{1}{2}\sum_{c,u}\Big|\Pr[U_n = u, C_n = c] - \Pr[U''_n = u, C'_n = c]\Big| \quad (2)$$

$$= \frac{1}{2}\sum_{c,u}\Big|\Pr[C_n = c \mid U_n = u]\Pr[U_n = u] - \Pr[C'_n = c]\Pr[U''_n = u]\Big| \quad (3)$$

$$= 2^{-n-1}\sum_{c,u}\Big|\Pr[\textsc{Com}_n(U_n) = c \mid U_n = u] - \Pr[\textsc{Com}_n(U'_n) = c]\Big| \quad (4)$$

$$= 2^{-n-1}\sum_{c,u}\Big|\Pr[\textsc{Com}_n(u) = c] - \sum_{u'}\big(\Pr[\textsc{Com}_n(U'_n) = c \mid U'_n = u']\Pr[U'_n = u']\big)\Big| \quad (5)$$

$$= 2^{-n-1}\sum_{c,u}\Big|\Pr[\textsc{Com}_n(u) = c] - 2^{-n}\sum_{u'}\Pr[\textsc{Com}_n(u') = c]\Big| \quad (6)$$

$$= 2^{-n-1}\sum_{c,u}\Big|2^{-n}\sum_{u'}\big(\Pr[\textsc{Com}_n(u) = c] - \Pr[\textsc{Com}_n(u') = c]\big)\Big| \quad (7)$$

$$\leq 2^{-2n-1}\sum_{u,u'}\sum_{c}\Big|\Pr[\textsc{Com}_n(u) = c] - \Pr[\textsc{Com}_n(u') = c]\Big| \quad (8)$$

$$= 2^{-2n}\sum_{u,u'}\Delta(\textsc{Com}_n(u)\,;\textsc{Com}_n(u')) \leq 2^{-2n}\sum_{u,u'}(2^{-\delta n}) = 2^{-\delta n} \ . \quad (9)$$

Let us briefly discuss the above (in)equalities. Equation 2 follows from the definition of the statistical distance. In Equation 3, we used the definition of conditional probability, and the independence of U''_n and C'_n. Equation 4 uses the definitions $C_n = \textsc{Com}_n(U_n)$ and $C'_n = \textsc{Com}_n(U'_n)$, as well as the fact that $\Pr[U_n = u] = \Pr[U'_n = u] = 2^{-n}$. In Equation 5, two identities are used: First, $\Pr[\textsc{Com}_n(U_n) = c \mid U_n = u]$ equals $\Pr[\textsc{Com}_n(u) = c]$. Second, we used the law of total probability to condition $\Pr[\textsc{Com}_n(U'_n) = c]$ on different values that U'_n may take. Equation 6 exploits the facts that $\Pr[\textsc{Com}_n(U'_n) = c \mid U'_n = u']$ equals $\Pr[\textsc{Com}_n(u') = c]$, and $\Pr[U'_n = u'] = 2^{-n}$. In Equation 7, a simple identity is used: Let a be an invariable quantity in x. Then, $\sum_{x \in X} a = |X|\, a$. Incorporating this identity into our case, we have: $\Pr[\textsc{Com}_n(u) = c] = 2^{-n} \sum_{u'} \Pr[\textsc{Com}_n(u) = c]$. Inequality 8 is obtained by applying the triangle inequality. Equation 9 uses the definition of statistical distance, as well as the fact that $\Delta(\textsc{Com}_n(u)\,;\textsc{Com}_n(u')) = 2^{-\delta n}$, as required by the statistical hiding of the commitment (cf. Equation 17 in Appendix A.3).

Stage 3. If $\hat{\Sigma}_n \neq \perp$, the protocol continues. Let h be the function that the verifier applies to its view to compute z_n. In other words, let $\hat{Z}_n \leftarrow h(V_2)$, where V_2 is either P_2 or S_2. Since $\Delta(P_2\,; S_2) \leq 2^{-\delta n}$, we can apply Fact 2 of Appendix A.2 to conclude that $\Delta(Z_n\,; Z'_n) \leq 2^{-\delta n}$.

Let F be the event that $Z_n = \pi_n^{-1}(Y_n)$, and F' be the event that $Z'_n = \pi_n^{-1}(Y'_n)$. Because $Y_n \sim Y'_n$ and $\Delta(Z_n\,; Z'_n) \leq 2^{-\delta n}$, it holds that $|\Pr[F] - \Pr[F']| \leq 2^{-\delta n}$. Now consider the following two cases:

1. If neither F nor F' happens: Both the simulator and the prover output $\perp$ and halt.

2. If both F and F' happen: The prover decommits by outputting ρ_n. The simulator has the preimage of y'_n, and therefore constructs the rest of the verifier's view identical to what the prover would do.

Notice that in both cases, the outputs of S and P are identical. Applying Lemma 5 of Appendix A.2, we get $\Delta(P_3\,; S_3) \leq 2^{-\delta n}$, which concludes the proof. ∎

3.3 Secure Authentication

In this section, we prove that Protocol 1 is a secure authentication protocol against *active attacks* in the smart-card model defined in Section 2.2. Contrary to the proof of zero-knowledge property given in the previous section, we have to assume that the input to the parties is chosen according by a PPT algorithm $G(1^n)$, as defined below:

$$\texttt{Let}\ (\mathsf{desc}(\pi_n), t_n) \leftarrow \textsc{GenP}(1^n),\ \text{and}\ \mathsf{desc}(\textsc{Com}_n) \leftarrow \textsc{GenC}(1^n)$$
$$\texttt{Define}\ i_n \stackrel{\text{def}}{=} (\mathsf{desc}(\pi_n), \mathsf{desc}(\textsc{Com}_n))$$
$$\texttt{OUTPUT}\ (i_n, t_n)$$

Theorem 2. *Let G be the algorithm defined above, and $\langle P, V\rangle$ be Protocol 1. Then, the triple (G, P, V) is a secure authentication protocol against* active attacks *in the smart-card model, assuming that* GenP *is a TDP generator, and* GenC *is a generator for statistically-hiding and computationally-binding commitments.*

It is straightforward to see that upon interacting with an honest prover P, the honest verifier V always accepts. Therefore, Protocol 1 has perfect completeness. It remains to prove that the soundness condition of Definition 1 holds as well. First recall the following notations[3]:

- ϵ_n: the hiding gap of the commitment scheme, which as defined by Equation 17, equals $2^{-\delta n}$.

- $\alpha_n \stackrel{\text{def}}{=} \mathbf{Adv}^{\textsc{Active}}_{\mathcal{A},(G,P,V)}(n)$: The advantage of $\mathcal{A}$ in mounting an active attack against the triple (G, P, V), as defined in Definition 1.

- $\beta_n \stackrel{\text{def}}{=} \mathbf{Adv}^{\textsc{Binding}}_{P^*,\textsc{GenC}}(n)$: Probability that P^* can break the binding property of a commitment generated by GenC, as defined in Definition 4.

[3]To prevent notational confusion, we chose the Greek letter corresponding to the first letter of the English name: α, β, and ι are mnemonics for **a**uthentication, **b**inding, and **i**nverting, respectively. Notice the difference between ι (Greek letter "iota") and i.

- $\iota_n \stackrel{\text{def}}{=} \mathbf{Adv}^{\text{INVERT}}_{M^{\mathcal{A}},\text{GEN4}}$: The advantage of $M^{\mathcal{A}}$ in inverting an element in the range of some TDP, as defined by Equation 12 in Appendix A.1. (The machines M and $\mathcal{A}$ will be defined below.)

Let $T_{\text{GENC}}(n)$ be an upper bound on the running time of $\text{GENC}(1^n)$. Moreover, for any oracle machine M, let $T_{M^{\mathcal{A}}}(n)$ be an upper bound on the running time of $M^{\mathcal{A}}$ on security parameter 1^n, including the total computation time of $\mathcal{A}$. Similarly, let $T_{\langle V^*,Q\rangle}(n)$ and $T_{\langle V,P^*\rangle}(n)$ be upper bounds on the total running time of the parties in the protocols $\langle V^*,Q\rangle$ and $\langle V,P^*\rangle$, respectively, when the security parameter is 1^n. The following lemma gives a direct relationship, in terms of the exact security [2], between the time and success probability of an active adversary against the authentication protocol, and the time and success probability of a TDP invertor.

Lemma 1. *There exists a PPT oracle machine M, such that for all $n \in \mathbb{N}$ and for any active PPT adversary $\mathcal{A} = (V^*, P^*)$ against the triple (G, P, V), where V^* interacts with P at most τ_n times, the following holds. If $\alpha_n > \beta_n + \tau_n\epsilon_n$ and $\beta_n \neq 1$, then:*

$$\iota_n \geq \left(\frac{\alpha_n - \beta_n - \tau_n\epsilon_n}{1-\beta_n}\right)^2 \ . \tag{10}$$

Furthermore, $T_{M^{\mathcal{A}}}(n) \leq 2\big(T_{\langle V^*,Q\rangle}(n) + T_{\langle V,P^*\rangle}(n)\big) + T_{\text{GENC}}(n)$.

Proof. Let M be the oracle Turing machine described in Algorithm 2. On a high level, M first tries to simulate a sequential prover for V^*, and then interacts with P^*. Consequently, if the adversarial coalition $\mathcal{A} = (V^*, P^*)$ succeeds in misrepresenting herself as the honest prover, M will invert the trapdoor permutation with probability related to the success probability of $\mathcal{A}$. Details follow.

Initially, M generates the description of a statistically-hiding commitment. It then simulates the execution of Protocol 1: First as an honest sequential prover denoted Q (see Definition 1), and next as an honest verifier. Finally, M tries to invert $\hat{y}_n$.

To simulate Q for V^*, algorithm M uses the SZK simulator S. As stated in Theorem 1, the statistical distance between the output of S and the real-world view of V^* is at most ϵ_n, in a *single execution*. A *hybrid argument* shows that this distance will increase to at most $\tau_n\epsilon_n$ in τ_n executions.

Consider stages 1 and 2 of Algorithm 2. Let st be the output generated by V^* before it halts, assuming V^* interacts with the real prover instead of the simulator. As stated above, the statistical distance between the random variables corresponding to st and st' is at most $\tau_n\epsilon_n$. Let α'_n be the success probability of P^* in breaking the authentication protocol, when its input is st' instead of st. An application of Fact 2 of Appendix A.2 shows that the output distribution of P^* on inputs st and st' are at most $\tau_n\epsilon_n$ far apart. We therefore get $|\alpha'_n - \alpha_n| \leq \tau_n\epsilon_n$, which guarantees $\alpha'_n \geq \alpha_n - \tau_n\epsilon_n$.

Let E_1 be the event that V^* succeeds in the impersonation attack, E_2 be the event that V^* successfully breaks the binding of the commitment scheme, and E_3 be the event that $M^{\mathcal{A}}$ does not output $\perp$ in step 2(e). By definition, $\Pr[E_1] = \alpha'_n$ and $\Pr[E_2] = \beta_n$. Since $\beta_n \neq 1$ by the premise, we can condition E_1 on $\overline{E_2}$. Using the law of total probability:

$$\begin{aligned}\alpha'_n = \Pr[E_1] &= \Pr[E_1 \cap E_2] + \Pr[E_1|\overline{E_2}]\Pr[\overline{E_2}] \\ &\leq \beta_n + \Pr[E_1|\overline{E_2}](1-\beta_n) \ .\end{aligned}$$

Now notice that $\Pr[E_3] = \Pr[E_1|\overline{E_2}]$, since $M^{\mathcal{A}}$ will not halt in step 2(e) if and only if V^* succeeds in impersonation without breaking the binding of the commitment. Therefore,

$$\Pr[E_3] \geq \frac{\alpha'_n - \beta_n}{1-\beta_n} \geq \frac{\alpha_n - \beta_n - \tau_n\epsilon_n}{1-\beta_n} \ .$$

By the premise, we know that the lower bound for $\Pr[E_3]$ is positive. Now notice that stage 3 of Algorithm 2 executes P^* on an independent input $(\hat{y}_n)$ chosen according to the same distribution as

Input: A pair $(\mathsf{desc}(\pi_n), \hat{y}_n)$ selected from the quadruple $(\mathsf{desc}(\pi_n), t_n, \hat{x}_n, \hat{y}_n) \leftarrow \textsc{Gen4}(1^n)$.

0. **Initialization:** Let $\mathsf{desc}(\textsc{Com}_n) \leftarrow \textsc{GenC}(1^n)$, and $i_n \leftarrow (\mathsf{desc}(\pi_n), \mathsf{desc}(\textsc{Com}_n))$.

1. **Simulate Q for V^*:** The algorithm M has black-box access to V^*, while it internally runs the SZK simulator $S(i_n)$. The goal is to simulate a sequential $Q(i_n, t_n)$ for V^*, such that the simulated output of V^* is indistinguishable from its real output (Q is defined by Definition 1).

   M keeps a flag F, indicating whether an instance of $S(i_n)$ is currently running (initially, $F = 0$). M also accepts the special message "New". Upon receiving this message, M replies with $\perp$ if $F = 1$. Otherwise, F is set to 1, and M will behave like $S(i_n)$ with fresh randomness. If S requires a message from V^*, M will obtain it from V^*. If S outputs any string, M will forward its most recent suffix to V^*. If S asks to rewind the verifier, M will rewind V^* to the state before S was spawned. If S halts, the flag F will be set to 0 again.

   As soon as V^* halts, M gets its output st', and proceeds to the next stage.

2. **Simulate V for P^*:** The algorithm M simulates V for P^* twice: (1) for some y_n whose corresponding x_n is chosen by M. This step is to obtain the value u_n; (2) for the specific $\hat{y}_n$, where M exploits the value u_n obtained in previous step.

   (a) Let $c_n \leftarrow P_r^*(i_n, st')$.
   (b) Let $x_n \leftarrow \textsc{Samp}(\mathsf{desc}(\pi_n))$ and $y_n \leftarrow \pi_n(x_n)$.
   (c) Let $\sigma_n \leftarrow P_r^*(i_n, st', y_n)$ and $u_n \leftarrow \sigma_n \oplus x_n$.
   (d) Let $\rho_n \leftarrow P_r^*(i_n, st', y_n, x_n)$.
   (e) If $c_n \neq \textsc{Com}_n(u_n\,; \rho_n)$, `OUTPUT` $\perp$ and halt.

3. **Invert $\hat{y}_n$:** If M did not halt, use u_n to invert $\hat{y}_n$:

   (a) Rewind P_r^* to step (c) and run it on $\hat{y}_n$. That is, let $\sigma_n^* \leftarrow P_r^*(i_n, st', \hat{y}_n)$.
   (b) Let $x_n^* \leftarrow \sigma_n^* \oplus u_n$. If $\hat{y}_n = \pi_n(x_n^*)$ then `OUTPUT` x_n^*; else `OUTPUT` $\perp$.

Algorithm 2: Description of algorithm M, which inverts $\hat{y}_n$ under π_n using black-box access to an *active* adversary $\mathcal{A} = (V^*, P^*)$ against Protocol 1.

y_n. Therefore, the probability that M does not output $\perp$ in step 3(b) equals $\Pr[E_3]$. Consequently,

$$\iota_n = (\Pr[E_3])^2 \geq \left(\frac{\alpha_n - \beta_n - \tau_n \epsilon_n}{1 - \beta_n}\right)^2 .$$

Finally, let us compute the running time of M. The running time of the initialization stage is at most $T_{\textsc{GenC}}(n)$. By Theorem 1, the simulator rewinds V^* at most once. Therefore, the running time of stage 1 is at most $2T_{\langle V^*, Q\rangle}(n)$. Lastly, notice that each of the stages 2 and 3 simulates a single execution of $\langle V, P^*\rangle$, and hence can be executed in at most $T_{\langle V, P^*\rangle}(n)$. Consequently, $T_{M^{\mathcal{A}}}(n) \leq 2\big(T_{\langle V^*, Q\rangle}(n) + T_{\langle V, P^*\rangle}(n)\big) + T_{\textsc{GenC}}(n)$, as required. ■

Proof of Theorem 2 is a straightforward consequence of Lemma 1:

Proof (Theorem 2). By assumption, GenP is a TDP generator, and GenP is a generator for a statistically-hiding and computationally-binding commitment. Therefore, for large enough n, the quantities ι_n, β_n, and ϵ_n are negligible in n. Furthermore, τ_n is always a polynomial in n, since V^* is a PPT algorithm.

Consequently, Equation 10 mandates that α_n be a negligible quantity in n, which implies that Protocol 1 is a secure authentication protocol against active attacks. ■

3.3.1 How to Interpret Lemma 1 for Practical Purposes

In practice, it is desirable to achieve a certain level of security, say 128-bit security. Below, we will interpret the meaning of a level of security, as well as how to achieve it based on the results of Lemma 1.

Let us first examine a simple case. Consider an algorithm which outputs the correct answer with probability p. If this algorithm is executed $1/p$ times, the probability that it outputs the correct answer is $1-(1-p)^{1/p} > 1-e^{-1} \approx 0.63$. Therefore, the success probability of such an algorithm is at least a constant (i.e., 63%) if it is executed $1/p$ times.

In cryptography, it is customary to compare the running times of algorithms with constant success probabilities. For instance, 128 bits of security means that no algorithm can break the scheme with constant success probability in less than 2^{128} steps.

Let us go back to the main question: How to interpret the results of Lemma 1? The crucial point is to differentiate between online and offline attacks. For instance, the adversary can try to invert the TDP offline, but to try her chance against the authentication scheme, she must be online. For this reason, α_n is sometimes called an *absolute constant*, which means it can be set regardless of the computational power of the adversary. This fact is best explained in [67, p. 190]:

> "The [...] probability of forgery is an absolute constant, and thus there is no need to pick [... a very small α_n, to] safeguard against future technological developments. In most applications, a security level of 2^{-20} suffices to deter cheaters. No one will present a forged passport at an airport, give a forged driver's license to a policeman, use a forged ill badge to enter a restricted area, or use a forged credit card at a department store, if he knows that his probability of success is only one in a million. [...] For national security applications, we can change the security level to 2^{-30}."

For a security level of 2^{-30}, the adversary has to present the forged smart card 2^{30} times to the verifier, to have a *constant* probability of masquerading. Assuming each authentication attempt takes only one second,[4] a success will be attainable (with constant probability) just once in 2^{30} seconds ≈ 34 years, regardless of the computational power of the adversary. By then, the adversary will probably be arrested due to fraud.

Similar to α_n, the advantage ϵ_n of the adversary in breaking the *hiding* property of the commitment is an absolute value. The reason is that the statistical hiding of the commitment holds regardless of the computational power of the adversary. Assume that the adversary is given the ability to verify the identity of a given smart card (that is, the adversary plays the role of V^*). Depending on the situation, the number of times the adversary may maliciously verify the smart card (i.e., the quantity τ_n) varies. A conservative choice is 2^{30}; that is, the adversary can pose itself as the real verifier for over a billion times. It seems that no real-life malicious verifier can reach this number, even if the smart card is stolen, and the adversary can verify the protocol for as many times as she wants. Now let $\epsilon_n \leq 2^{-61}$. For small enough values of β_n, this satisfies the premise of Lemma 1 that $\alpha_n > \beta_n + \tau_n\epsilon_n$.

Finally, we get to choose the values ι_n and β_n. Momentarily assume that β_n is negligible relative to $\alpha_n - \tau_n\epsilon_n \geq 2^{-31}$. Therefore, Lemma 1 presents an inverter with execution time $T_{M^{\mathcal{A}}}(n)$ and success probability $\iota_n \gtrapprox (\alpha_n - \tau_n\epsilon_n)^2 \geq 2^{-62}$. The lemma bounds $T_{M^{\mathcal{A}}}(n)$ by twice the time $\mathcal{A}$ can interact *online* with the honest provers and verifier. A real-world assumption is $T_{M^{\mathcal{A}}}(n) \leq 2^{25}$ bit operations. If the best known algorithm to invert the TDP has a complexity more than $2^{25}/2^{-62} = 2^{87}$, we can assume that the authentication protocol is secure. This is because the existence of an adversary

[4] Fiat and Shamir [67] suggest that the attacker can make at most 1000 forgery attempts per day. Thus, with a security level of 2^{-30}, she will succeed (with constant probability) in masquerading once in every 3000 years. Therefore, our assumption that the adversary can make a forgery attempt once per second is very conservative, but it shows that even with such power, she cannot succeed in a reasonable amount of time.

against the authentication protocol is translated (via Lemma 1) to the existence of an inverter against the TDP with success probability better than the best known algorithm, which is deemed impossible. In this paper, we assume 128-bit security; therefore, $\iota_n, \beta_n \leq 2^{-128}$.

Remark 4. The astute reader might ask why β_n is taken to be so small, while Lemma 1 does not seem to require such a small success probability. The reason is that β_n is an *offline* parameter. That is, the adversary may break the binding property offline (via preprocessing), and then attempt to attack the authentication protocol. The same holds for ι_n: The adversary can find the trapdoor offline, and then attack the authentication protocol. Therefore, the protocol designer must choose the parameters to foil offline attacks as well. It seems that a security level of 2^{100} or more is the recommended choice for the near future. We therefore picked the conservative 128-bit security. $\triangleleft$

4 An Efficient Instantiation of Protocol 1, Secure Against Quantum Attacks

In this section, we implement the commitment and the TDP used in Protocol 1, in such a way that the protocol remains secure against quantum attacks. Notice that the zero-knowledge property is already guaranteed to hold against infinitely powerful adversaries, and therefore we only focus on the security of the authentication protocol. At the end of this section, we give an overall estimate of the efficiency of our protocol, and compare it to other protocols in the literature. Definitions related to lattice problems are given in Appendix A.5.

4.1 Constructing the Commitment

Kawachi *et al.* [47, 68] suggest a lattice-based commitment. The computational-binding property of their scheme is based on the hardness of the SIS problem, while its statistical-hiding property holds unconditionally.

Given an integer n, let $m = m(n)$, and $q = q(n)$ be integers bounded by a polynomial in n. The generator $\textsc{GenC}$ of Kawachi *et al.*'s commitment scheme works as follows: On input 1^n, it outputs a matrix $\mathbf{A}$, chosen uniformly from $\mathbb{Z}_q^{n \times m}$. In this scheme, $\ell(n) = m/2$, and the distribution $\textsc{Rnd}_{\ell(n)}$ from which the commitment randomness is sampled is the uniform distribution over $\{0,1\}^{m/2}$.

To commit to a string $x \in \{0,1\}^{m/2}$, we first pick $r \leftarrow \textsc{Rnd}_{m/2}$. Let $\mathbf{x}$ and $\mathbf{r}$ denote the column vectors corresponding to x and r, respectively. Moreover, let $\mathbf{x} \,||\, \mathbf{r}$ denote the column vector obtained from appending $\mathbf{r}$ to $\mathbf{x}$. The commitment is then defined by:

$$\textsc{Com}_n(x) \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{x} \,||\, \mathbf{r}) \bmod q \ . \tag{11}$$

The following lemma is proven in [68, Lemma 5.3.2]:

Lemma 2. *The commitment defined above is:*

- *statistically hiding with statistical gap $2q^{-dn/4}$ if $m > 2n(1+d)\lg q$ for some positive constant d;*
- *computationally binding if collision-finding $\mathbf{SIS}^{\infty}_{q,m,n,1}$ is hard.*[5] *In other words, collision-finding $\mathbf{SIS}^{\infty}_{q,m,n,1}$ reduces to breaking the computational-binding property of the commitment.*

The second condition can be interpreted both theoretically and practically. In theory, the SIS problem is proven hard via an efficient reduction from worst-case SIVP to the average-case SIS. The most recent result is [69, Theorem 4], which gives the best current reduction. The theorem, cast for the case of $\mathbf{SIS}^{\infty}_{q,m,n,1}$, is as follows:

[5] [68] reduces $\mathbf{SIS}^{2}_{q,m,n,\sqrt{m}}$ to breaking the computational-binding property of the commitment, which is a weaker reduction. It also requires that q be a prime, but as we will see in Lemma 3, recent results relaxed this requirement.

Lemma 3. *For $q \geq \sqrt{m} \cdot n^{\Omega(1)}$, there is an efficient reduction from $\mathbf{SIVP}_{\omega(\sqrt{mn \log n})}$ to collision-finding $\mathbf{SIS}^{\infty}_{q,m,n,1}$ with non-negligible advantage.*

The lower bound given for q in Lemma 3 is essentially optimal, as the problem is trivially easy for $q \leq \sqrt{m}$ [69]. However, the scope of this reduction is limited to the *asymptotic* case. In practice, the SIS problem might be hard, regardless of whether other lattice problems can be efficiently reduced to it. The following formula, suggested in [70], gives a heuristic for the shortest SIS solution attainable by the best algorithm, assuming $m \geq \sqrt{n \lg q / \lg \hbar}$:

$$\min\{q, 2^{2\sqrt{n \lg q \lg \hbar}}\} ,$$

where $\hbar$ is the *hermit factor* of the algorithm. The current best algorithm, BKZ 2.0 [71], requires over 2^{128} steps to achieve $\hbar = 1.006$. Therefore, by setting $n = 128$, $q = 257$ and $m \geq \sqrt{n \lg q / \lg \hbar} \approx 345$, we can be sure that it is highly unlikely that current algorithms can find vectors shorter than 61 in the corresponding SIS problem, in less than 2^{128} steps. Consequently, if $\sqrt{m} < 61$, then the security against an adversary attacking the binding property is at least 128 bits.

On the other hand, m should be large enough to satisfy the statistical gap. Setting $m = 2560$ (which still satisfies $\sqrt{m} < 61$), we achieve a statistical gap of 2^{-62}, as desired (see Section 3.3.1).

Given the parameters $n = 128$, $m = 2560$, and $q = 257$, the size of the SIS matrix (i.e., the description of the commitment) will be $nm\lceil \lg q \rceil \approx 360$ KB. However, by defining the SIS over rings [72–76], one can reduce this size by a factor of n, and thus achieving a SIS matrix as small as 2.8 KB. (The ring setting requires n to be a power of 2, m to be a multiple of n, and $q \equiv 1 \pmod{2n}$ to be a prime. All requirements are satisfied by our choice of parameters.)

4.2 Constructing the TDP

There are several TDPs with conjectured security against quantum attacks. The oldest ones are McEliece [77] and Niederreiter [78], which are based on the coding theory (see [79] for more information). While McEliece and Niederreiter are sometimes called "encryption," they do not satisfy the semantic security property, and are actually TDPs.[6] McEliece and Niederreiter are dual to each other, in the sense that an attacker that breaks one can break another [82]. The precise assumptions underlying the security of the Niederreiter TDP is studied in [81], while [83] examines the practical security of both McEliece and Niederreiter: For 80-bit security, the size of $\mathsf{desc}(\pi_n)$ is 56 KB. It grows to 188 KB for achieving 128-bit security.

Another option is to use lattice-based TDPs. Micciancio and Peikert [84] examine how LWE and ring-LWE problems can be used to construct TDPs. However, based on their results, the size of $\mathsf{desc}(\pi_n)$ is prohibitively large for smart cards.

A third option is to incorporate a lattice-based encryption, instead of a TDP. The protocol and its proof of security should change minimally to reflect this modification. Recently, a very efficient set of parameters were proposed for the ring-LWE encryption [7,85]. Specifically, achieving 128-bit security is possible with $n = 256$, $q = 7,681$, and $s = 11.31$, for which the size of the system parameter and the public key is $n|q| \approx 416$ bytes, and the size of the secret key is $n|5s| \approx 192$ bytes. (Notice that n is a power of 2, and $q \equiv 1 \pmod{2n}$ is a prime, which is required for ring operations.)

4.3 Overall Analysis

In this section, we analyze the overall complexity of our protocol, and compare it to several other lattice-based authentication protocols. We picked protocols which have a ZK-like structure; that is, they are either ZK or obtained by executing some base ZK protocol in parallel). The list is not

[6] A semantically-secure variant of McEliece is proposed in [80], but the original McEliece is a TDP, as defined in [81, footnote 2].

Table 1: Comparison of several lattice-based authentication protocols at 128-bit security. (Quantities in bytes are underlined to be easily distinguishable from those in Kilobytes.)

Protocol	System Parameter Size	SK Size	PK Size	# of Passes	Comm. Complexity	Concurrently Secure?	ZK?
Protocol 1	3.2 KB	<u>192 B</u>	<u>416 B</u>	5	1.46 KB	No, but see Section 5	SZK
[47]	32 KB	<u>320 B</u>	<u>144 B</u>	3	281 KB	+	–
[48]	8 KB	2 KB	2 KB	3	7 KB	+	–
[50]	32 KB	<u>320 B</u>	<u>144 B</u>	5×30	144 KB	–	SZK
[52]	32 KB	9 KB	4 KB	5×30	157 KB	–	SZK

exhaustive; yet protocols not listed here are either too inefficient (such as [46]), or are similar to the protocols we mentioned here (e.g., [49] is similar to [47]).

Remark 5. There might be more efficient lattice-based authentication protocols for smart cards, such as simple challenge-response protocols which use lattice-based encryption or signature. However, as described in Section 1, the main idea of this paper is to propose an efficient zero-knowledge authentication protocol which can be implemented on smart cards for environments with tight security requirements. Therefore, we did not compare our protocol with those without a ZK-like structure. $\lhd$

Table 1 gives an overview of the comparison. Notice that for the sake of readability, some numbers are denoted in bytes, while others are in Kilobytes (= 1024 bytes). In protocols like [47], the base protocol is ZK, but the protocol designers use the parallel repetition which is not ZK anymore. In such cases, the table does not consider the protocol as ZK.

Below, we will describe our choice of parameters for each protocol. It is assumed that the protocols must satisfy 128-bit security, with soundness error at most 2^{-30}, and completeness error less than 2^{-20}.

- Our protocol (Protocol 1): We described the choice of parameters to get a secure commitment and a secure TDP in Sections 4.1 and 4.2, respectively. Specifically, $n_C = 128$, $m_C = 2560$, and $q_C = 257$ for the commitment (SIS) matrix, and $n_T = 256$, $q_T = 7,681$, and $s_T = 11.31$ for the TDP (LWE) matrix, with 128-bit message length. The communication complexity is therefore $n_C|q_C| + 2n_T|q_T| + m_C + 128 + n_T|5s_T| \approx 1.5$ KB.

- Kawachi *et al.* [47]: We set parameters similar to ours: $n = 128$, $m = 2560$, and $q = 257$. This protocol requires another commitment matrix, which should be able to commit to binary strings whose length is $M = n\lceil(\lg m!)/n\rceil + n|q| = 26,496$. For this, we pick a random matrix from $\mathbb{Z}_q^{n\times M}$. Since the soundness error of the base protocol is $\frac{2}{3}$, it is required to be repeated $t = 52$ times so that its soundness error is at most 2^{-30}. The communication complexity is $t(3n|p| + 2 + (2\lceil\lg m!\rceil + m)/3 + m|p|) \approx 281$ KB.

- Lyubashevsky [48]: Fig. 2 of [48] gives four sets of parameters for 80-bit security. We used the first set of parameters, but adjusted κ to achieve 128-bit security: $n = 512$, $m = 4$, $\sigma = 127$, $\kappa = 44$, and $p \approx 2^{32}$. The completeness error of the protocol is $1 - 1/e$. To make the the completeness error less than 2^{-20}, we must repeat the protocol for $t = 31$ times. [48, p. 610] gives a series of tricks to improve the efficiency, which we will incorporate here. The most important trick is to use a hash function such as the SHA-256 in the first step of the protocol. Using the notations of [48], the communication complexity is $256 \cdot t + |G^m| + |D_c| \approx 7$ KB.

- Cayrel *et al.* [50]: The public parameter and the prover's public and private keys are exactly like those in [47]; we therefore use the same parameters. The soundness error of the base protocol is almost $1/2$, and hence it must be repeated $t = 30$ times to achieve the 2^{-30} soundness error. The communication complexity of the protocol is $t(2n|q|+|q|+m|q|+1+n+(\lceil\lg m!\rceil+m)/2) \approx 144$ KB.

- Silva *et al.* [52]: The public parameter is exactly as in [50], but the prover's public and private keys differ. Again, the soundness error of the base protocol is almost $1/2$. The communication complexity of the protocol is $t(3n|q|+2|t|+1+m+m|q|+(\lceil \lg m! \rceil+m)/2) \approx 157$ KB. Interestingly, while this protocol is more efficient for 80-bit security than [50] (see [52]), it is less efficient at 128-bit security.

As Table 1 shows, the best previous protocol was [48]. Below, we will compare our protocol with this protocol.

In terms of communication complexity, Protocol 1 improves [48] by a factor of 5, while requiring less than half storage.

Since all protocols (except ours) mentioned in Table 1 are repeated several times, they need to perform many lattice-based computations. However, our protocol makes only two lattice operations: a SIS and an LWE-based encryption. Therefore, our protocol is much more efficient in terms of computation complexity. In particular, since the base protocol of [48] has similar operations to ours, and it is repeated 30 times, Protocol 1 improves [48] by a factor of 30.

The *theoretical* round complexity of Protocol 1 is 5, which is close to the best (i.e., 3). Let us now consider the *practical round complexity*: As described in Section 1, smart cards transmit data in units called the *Application Protocol Data Unit* (APDU), which can carry up to 255 bytes of data. Therefore, our protocol requires at least $\lceil \frac{1.46 \text{ KB}}{255 \text{ B}} \rceil = 6$ passes (rather than 5) to perform the authentication in practice. The round complexity of other protocols discussed above is much higher due to their high communication complexity. For instance, [48] requires at least $\lceil \frac{7 \text{ KB}}{255 \text{ B}} \rceil = 29$ passes (rather than 3).

The final point is that our protocol is statistical zero knowledge, while [48] is not. On the other hand, [48] is secure against concurrent attacks. While such attacks are *not* practical against real-world smart cards (because the smart card does not have enough resources to take part in multiple sessions simultaneously), it is certainly advantageous to study these attacks *in theory*. In the next section, we present a modification to our protocol to make it resilient to concurrent attacks.

Remark 6. As pointed out in Remark 1, the recent work of Boorghany and Jalili [4] shows the practicality of Protocol 1, by implementing it on a real smart card. Noting that no competitors in Table 1 (except our protocol) can be implemented practically, Boorghany and Jalili took a different approach for comparison: they first constructed authentication protocols from efficient lattice-based signature schemes [86, 87], and then implemented the results on a smart card.

The results of [4] show that on a Feitian FT-Java/H10CR Java Card, Protocol 1 is almost *three* times faster than the authentication protocol based on [87], which is in turn *twice* faster than the authentication protocol based on [86]. Furthermore, they showed that it takes almost *half a second* to execute Protocol 1 on an AVR ATxmega64A3 microcontroller.

As an added bonus, we mention that Protocol 1 is an *statistical* zero knowledge authentication protocol, while the authentication protocols based on [86, 87] are not even zero knowledge. $\triangleleft$

5 Modifying Protocol 1 to Thwart Concurrent Attacks

The zero-knowledge simulator of Protocol 1 does not work in the concurrent setting, since it *rewinds* the verifier. Diagram 1 of [25, p. 410] illustrates the difficulty that arises when dealing with rewinding simulators in the concurrent setting. This statement can be generalized to the extent of denying any *black-box* simulator for the protocol: [24] proves a logarithmic lower bound on the round complexity of black-box CZK protocols, while Protocol 1 is constant round.

Furthermore, the protocol is not known to remain a secure authentication protocol against concurrent attacks, since Algorithm 2 makes explicit use of the zero-knowledge simulator to simulate Q for V^*, and this simulator does not work in the concurrent setting.

In this section, we modify Protocol 1 in such a way that it remains a secure authentication protocol against concurrent attacks. As an added bonus, the modified protocol will remain SZK if executed sequentially.

A first idea is to modify the protocol such that the common input includes the descriptions of two TDPs instead of one, and the prover will then prove that he can invert either of them. This idea is similar to that of *OR proofs* [88], with one major difference: The OR proof is a transformation on *public-coin* ZK proofs, while Protocol 1 uses private coins. There are two objections against this approach: Firstly, an OR-proof reduces the efficiency of the protocol, and increases its communication complexity. Secondly, difficulties arise when dealing with private-coin protocols, and they cannot be easily transformed to OR proofs without making extra assumptions.[7]

A better idea, due to Damgård [65], is to use the concept of trapdoor commitments [89], also known as chameleon blobs [90] or equivocable commitments [91–93]. (Although the last reference explains definitional differences between these concepts, we will use "trapdoor commitment" as an umbrella term to refer to all of them). Informally, a trapdoor commitment is a commitment that satisfies an extra property: There is an algorithm which generates a "twisted" description of the commitment, along with a trapdoor. This description must be indistinguishable from an honestly generated description. Moreover, there exists an algorithm which can output a commitment, and then open it to any arbitrary string using the trapdoor.

We notice that trapdoor commitments can be constructed from ordinary commitments without making new assumptions. Section 2 of [94, Chapter 3] describes several such constructions.

Definition 2 (Non-interactive Statistically-Hiding *Trapdoor* Commitments). A pair of PPT algorithms $(\textsc{GenC}, \mathsf{Sim})$ is called a *non-interactive statistically-hiding trapdoor commitment*, if the following conditions hold:

1. $\textsc{GenC}$ is a generator for some non-interactive statistically-hiding commitment (recall Definition 4 in Appendix A.3).
2. For any $n \in \mathbb{N}$, and all $x \in \{0,1\}^n$, the statistical distance between the outputs of the left-and-right experiments below is at most $2^{-\mu n}$.

$\mathsf{desc}(\textsc{Com}_n) \leftarrow \textsc{GenC}(1^n)$	$(\mathsf{desc}(\widetilde{\textsc{Com}}_n), \tilde{t}_n) \leftarrow \mathsf{Sim}(\texttt{'Gen'}, 1^n)$
$r \leftarrow \textsc{Rnd}_{\ell(n)}$	$\tilde{c} \leftarrow \mathsf{Sim}(\texttt{'Commit'}, \mathsf{desc}(\widetilde{\textsc{Com}}_n))$
$c \leftarrow \textsc{Com}_n(x\,;r)$	$\tilde{r} \leftarrow \mathsf{Sim}(\texttt{'Decommit'}, \mathsf{desc}(\widetilde{\textsc{Com}}_n), \tilde{t}_n, \tilde{c}, x)$
$\texttt{OUTPUT}\ (\mathsf{desc}(\textsc{Com}_n), x, c, r)$	$\texttt{OUTPUT}\ (\mathsf{desc}(\widetilde{\textsc{Com}}_n), x, \tilde{c}, \tilde{r})$

○

Remark 7. In our protocol, we do not need to open the commitment to an arbitrary string x. Rather, we merely need to open it to a *randomly chosen* string. ◁

Define "Protocol 2" as the modified version of Protocol 1, which uses trapdoor commitments instead of ordinary ones. However, notice that Sim is not used in the real-life execution. It is only employed in the proof of security, as detailed later. Therefore, we continue to assume that in the real-life execution, $\textsc{Com}_n$ is generated honestly (i.e., via $\textsc{GenC}$). See the beginning of Section 3.1, where three methods for honest generation of $\textsc{Com}_n$ are suggested (out-of-band agreement, CRS, and TTP).

[7] In our setting, we required an assumption like the indistinguishability of the pair $(\pi_n^0(U_n), \pi_n^1(U_n))$ from $(\pi_n^0(U_n), \pi_n^1(U_n'))$, where π_n^0 and π_n^1 are independently generated TDPs. This assumption is much stronger that the non-invertibility of a single TDP, and we know few TDPs that satisfy this strong assumption.

Input: Same as Algorithm 2.

0. **Initialization:** Let
   $(\mathsf{desc}(\widetilde{\text{COM}}_n), \tilde{t}_n) \leftarrow \mathsf{Sim}(\texttt{'Gen'}, 1^n)$, and
   $i_n \leftarrow (\mathsf{desc}(\pi_n), \mathsf{desc}(\widetilde{\text{COM}}_n))$.

1. **Simulate Q for V^*:** Q keeps a set ID (initially empty), and accepts the special message $\text{NEW}(id)$. Upon receiving this message, Q checks whether $id \in ID$, and replies with $\perp$ if this is the case. Otherwise, Q sets $ID \leftarrow ID \cup \{id\}$, and *simulates* a new instance of the prover with fresh randomness and id as identifier, as follows: Upon receiving a message from V^* with id as its prefix, dispatch it to the simulated prover with identifier id. The simulated prover then makes a computation, and sends a message. Q then saves the state of this prover for later calls.

   The algorithm of the simulated prover with identifier id is described below:

   (a) Generate a commitment by computing $c'_n \leftarrow \mathsf{Sim}(\texttt{'Commit'}, \mathsf{desc}(\widetilde{\text{COM}}_n))$. Send (id, c'_n) to V^*.

   (b) Receive the challenge y'_n from V^*.

   (c) If $y'_n \notin \{0,1\}^n$, send $(id, \perp)$ to V^* and halt.
   Pick a random n-bit string σ'_n, and send (id, σ'_n) to V^*.

   (d) Receive z'_n from V^*.

   (e) If $y'_n \neq \text{EVAL}(\mathsf{desc}(\pi_n), z'_n)$, then send $(id, \perp)$ to V^* and halt.
   Else let $u''_n \leftarrow z'_n \oplus \sigma'_n$, and $\rho''_n \leftarrow \mathsf{Sim}(\texttt{'Decommit'}, \mathsf{desc}(\widetilde{\text{COM}}_n), \tilde{t}_n, c'_n, u''_n)$. Send (id, ρ''_n) to V^*.

As soon as V^* halts, M' gets its output st', and then proceeds exactly as steps 2 & 3 of Algorithm 2.

Algorithm 3: Description of algorithm M', which inverts $\hat{y}_n$ under π_n using black-box access to a *concurrent* adversary $\mathcal{A} = (V^*, P^*)$ against Protocol 2.

Since substituting an ordinary commitment with a trapdoor commitment does not change the real-life execution, the security proofs of Protocol 1 carries over to Protocol 2. In other words, Protocol 2 remains SZK when executed sequentially, and it is a secure authentication protocol against active adversaries. It remains to exploit the properties of trapdoor commitments to prove that Protocol 2 is a secure authentication protocol against *concurrent* adversaries.

Theorem 3. *Protocol 2 is a secure authentication protocol against concurrent PPT adversaries.*

Proof. Let $\mathcal{A} = (V^*, P^*)$ be a concurrent adversary. Recall from Definition 1 that in the concurrent setting, V^* can send the special message $\text{NEW}(id)$, to spawn a new instance of the prover with id as its identifier. Furthermore, to communicate with the prover whose identifier is id, the cheating verifier must prefix her messages with id.

We now construct an algorithm, similar to M (see Algorithm 2), which inverts its input under the TDP, given black-box access to $\mathcal{A}$. Let us call this algorithm M'. Contrary to M, the inverter M' should simulate a concurrent setting for V^* in the information gathering phase. The code for M' is given in Algorithm 3. Here is the ideas used by M':

- M' instantiates a trapdoor commitment instead of an ordinary one. Given the indistinguishability of the descriptions of $\widetilde{\text{COM}}_n$ and COM_n, the malicious verifier will notice the change with probability at most $2^{-\mu n}$.

- M' returns a random bit string σ'_n instead of $\sigma_n \stackrel{\text{def}}{=} u_n \oplus w_n$. As shown in *Stage 2* of the proof of

Theorem 1, the statistical distance between σ'_n and σ_n is at most $2^{-\delta n}$ (even when the rest of the view is given).

- If V^* reveals a correct pre-image z'_n of y'_n, the algorithm M' uses Sim to open the commitment c'_n as $z'_n \oplus \sigma'_n$, thus pretending to V^* that it had correctly sent the right pre-image in step (c).

Since M' does *not* rewind V^*, it does not suffer from the weakness of the ZK simulator, described at the beginning of this section.

Using the triangle inequality, the statistical distance between the view of V^* in the real and simulated executions is $\epsilon_n \leq 2^{-\mu n} + 2^{-\delta n}$ in a single execution. We can now apply Lemma 1, where $\tau_n = \text{poly}(n)$ is an upper bound on the number of prover instances that V^* spawns. The rest of the proof is similar to the proof of Theorem 2. ■

5.1 Constructing a Lattice-based Trapdoor Commitment

In Section 4.1, we described how lattice-based commitments can be constructed. This section modifies the construction to achieve lattice-based *trapdoor* commitments. Our main tool is the results of [84, 95, 96], which describes how to generate a random looking matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, in which a trapdoor is embedded.[8] Given this trapdoor, and a random vector $\mathbf{c} \in \mathbb{Z}_q^n$, one can efficiently sample a vector $\mathbf{z} \in \mathbb{Z}^m$ according to some narrow Gaussian distribution, such that $\mathbf{Az} \equiv \mathbf{c} \pmod q$. Let us explain the details.

Generating the description of the commitment. For any $n \in \mathbb{N}$, the output of $\textsc{GenC}(1^n)$ is a matrix $\mathbf{A}$, chosen randomly from $\mathbb{Z}_q^{n\times 2m}$. Here, q and $m \geq 2n \lg q$ are polynomially bounded in n. The algorithm $\textsc{GenC}$ also defines the distribution of the randomness to the commitment, which is a discrete Gaussian distribution $D_{\mathbb{Z}^m,s}$ with parameter $s \geq \omega(\sqrt{\log m})$. Let $\mathbf{A} = [\mathbf{A}_1 \,||\, \mathbf{A}_2]$, where $\mathbf{A}_1$ is the first m columns of $\mathbf{A}$, and $\mathbf{A}_2$ constitutes the remaining columns of $\mathbf{A}$.

Committing to a string $x \in \{0,1\}^m$. Let $\mathbf{x}$ be the m-dimensional column vector (with binary entries) corresponding to x. Pick a random vector $\mathbf{r} \leftarrow D_{\mathbb{Z}^m,s}$, and define the commitment as in Equation 11. It is proven in [99, Corollary 5.4] that except for an exponentially small fraction of $\mathbf{A}_2$'s, the quantity $\mathbf{A}_2\mathbf{r} \pmod q$ is statistically close to the uniform distribution over $\mathbb{Z}_q^n$. Therefore, $\textsc{Com}_n(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{x} \,||\, \mathbf{r}) \bmod q = \mathbf{A}_1\mathbf{x} + \mathbf{A}_2\mathbf{r} \pmod q$ is statistically close to uniform distribution over $\mathbb{Z}_q^n$. Consequently, for any two m-bit strings x_1 and x_2, the commitments to x_1 and x_2 are statistically close, and the commitment is statistically hiding.

Regarding the binding property, care must be taken as there is no theoretical limit on the length of $\mathbf{r}$. However, the probability that $\|\mathbf{r}\|_2 > Ls$ for any positive L is at most $e^{-\pi L^2}$. Therefore, the receiver (of the commitment protocol) can safely reject if the length of the revealed randomness exceeds Ls for some given L. In this approach, the reveal phase of the commitment may fail with an exponentially small probability (which results in an exponentially small *completeness* error in our protocol). Notice that the collision-finding SIS problem with $\beta = Ls + \sqrt{m} = L\omega(\sqrt{\log m}) + \sqrt{m}$ reduces to breaking the binding property of this commitment.

The trapdoor commitment. Algorithm $\mathsf{Sim}(\texttt{'Gen'}, 1^n)$ generates a special matrix $\widetilde{\mathbf{A}}_2 \in \mathbb{Z}_q^{n\times m}$, with the associated trapdoor $\tilde{\mathbf{t}}_\mathbf{n}$, as described in [84]. The parameters q and m are chosen properly. Micciancio and Peikert [84] describe a method in which $\widetilde{\mathbf{A}}_2$ is statistically indistinguishable from a uniformly chosen matrix. Next, Sim defines $\widetilde{\mathbf{A}} \stackrel{\text{def}}{=} [\widetilde{\mathbf{A}}_1 \,||\, \widetilde{\mathbf{A}}_2]$, where $\widetilde{\mathbf{A}}_1 \leftarrow_R \mathbb{Z}_q^{n\times m}$.

[8] After the preliminary version of this work [3], but independently of it, Blazy *et al.* [97] constructed a lattice-based trapdoor commitment. Similar to our results, they use the Micciancio–Peikert trapdoors [84], but they utilize the *smooth projective hash functions* [98] to construct their lattice-based trapdoor commitment.

$\mathsf{Sim}(\texttt{'Commit'}, \widetilde{\mathbf{A}})$ outputs a uniform element $\tilde{\mathbf{c}}$ in $\mathbb{Z}_q^n$. Since $\textsc{Com}_n(\mathbf{x})$ is statistically close to uniform for any $\mathbf{x} \in \mathbb{Z}_2^m$, the random variables $\tilde{\mathbf{c}}$ and $\textsc{Com}_n(\mathbf{x})$ are statistically close.

For any $\mathbf{x} \in \mathbb{Z}_q^m$ independent of $\tilde{\mathbf{c}}$, the algorithm $\mathsf{Sim}(\texttt{'Decommit'}, \widetilde{\mathbf{A}}, \tilde{\mathbf{t}}_{\mathbf{n}}, \tilde{\mathbf{c}}, \mathbf{x})$ works as follows: It first computes $\tilde{\mathbf{c}}_{\mathbf{2}} \stackrel{\text{def}}{=} \tilde{\mathbf{c}} - \widetilde{\mathbf{A}}_1\mathbf{x} \pmod q$, which is a uniform element in $\mathbb{Z}_q^n$ since $\tilde{\mathbf{c}}$ was picked uniformly. It then uses the trapdoor $\tilde{\mathbf{t}}_{\mathbf{n}}$ and the pre-image sampling of [84] to choose a vector $\tilde{\mathbf{r}}$ from the discrete Gaussian distribution $D_{\mathbb{Z}^m,s'}$, such that $\tilde{\mathbf{c}}_{\mathbf{2}} = \widetilde{\mathbf{A}}_2\tilde{\mathbf{r}} \pmod q$.

Notice that in order the parameters should be set in such a way that $D_{\mathbb{Z}^m,s}$ and $D_{\mathbb{Z}^m,s'}$ are statistically close. If possible, the best choice is $s = s'$.

6 Conclusions and Future Work

In this paper, we presented a general SZK authentication protocol, and proved its exact security. The protocol was then instantiated using lattice-based constructions, so as to remain secure against quantum attacks. We next modified the general protocol using trapdoor commitments, and proved that the modified protocol is secure against concurrent attacks. Finally, it was shown how the trapdoor commitment can be instantiated using lattice-based cryptography.

Below, we will try to present the most important direction for future research:

- Improving the security proofs to provide a tighter security reduction.
- Finding the parameters for the lattice-based trapdoor commitment to achieve a certain level of security.
- Modifying the protocol so that it resists resetting attacks, which are practical against smart cards.
- Discussing a practical implementation which is secure against side-channel attacks.
- Improving the protocol to support bilateral authentication.

A final direction is to examine whether the security proofs carry over to the case where the protocol is modified as follows. The prover authenticates himself to the verifier by proving an OR statement: Either he knows the trapdoor of the TDP, or he knows the trapdoor of the commitment. In this case, a single matrix $\mathbf{A}$ might be used for the construction of both the TDP (based on the LWE problem), and the trapdoor commitment (based on the SIS problem). The modification reduces the storage requirement for public and private keys.

References

[1] Oded Goldreich and Hugo Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, 25(1):169–192, 1996. See [61] for the conference version.

[2] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures—How to Sign with RSA and Rabin. In *Advances in Cryptology—EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer Berlin / Heidelberg, 1996.

[3] Mohammad Sadeq Dousti and Rasool Jalili. Efficient Statistical Zero-Knowledge Authentication Protocols for Smart Cards Secure Against Active & Concurrent Attacks. Cryptology ePrint Archive, 2013. Available from http://eprint.iacr.org/2013/709.

[4] Ahmad Boorghany and Rasool Jalili. Implementation and Comparison of Lattice-based Identification Protocols on Smart Cards and Microcontrollers. Cryptology ePrint Archive, 2014. Available from http://eprint.iacr.org/2014/078.

[5] Thijs Laarhoven, Michele Mosca, and Joop H. van de Pol. Solving the Shortest Vector Problem in Lattices Faster Using Quantum Search. In *Proceedings of the 5th International Workshop on Post-Quantum Cryptography (PQCrypto 2013)*, pages 83–101, Limoges, France, 2013. Springer.

[6] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[7] Richard Lindner and Chris Peikert. Better Key Sizes (and Attacks) for LWE-based Encryption. In *Proceedings of the 11th International Conference on Topics in Cryptology* (*CT-RSA 2011*), pages 319–339, San Francisco, California, USA, 2011. Springer. Full version is available at http://eprint.iacr.org/2010/613.

[8] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067, 2005. Available from http://eprint.iacr.org/2000/067. See [100] for the conference version.

[9] Phillip Rogaway. On the Role of Definitions in and Beyond Cryptography. In *Proceedings of the 9th Asian Computing Science Conference* (*ASIAN 2004*), pages 13–32, Chiang Mai, Thailand, 2004. Springer.

[10] Mihir Bellare and Phillip Rogaway. Entity Authentication and Key Distribution. In *Advances in Cryptology—CRYPTO '93*, pages 232–249, Santa Barbara, California, USA, 1993. Springer.

[11] Mihir Bellare and Phillip Rogaway. Provably Secure Session Key Distribution: The Three Party Case. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing* (*STOC '95*), pages 57–66, Las Vegas, Nevada, USA, 1995. ACM.

[12] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols (Extended Abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (*STOC '98*), pages 419–428, Dallas, Texas, USA, 1998. ACM.

[13] Victor Shoup. On Formal Models for Secure Key Exchange. Technical report, IBM Zurich Research Lab, 1999. Version 4 is available at http://eprint.iacr.org/1999/012.

[14] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated Key Exchange Secure against Dictionary Attacks. In *Advances in Cryptology—EUROCRYPT '00*, pages 139–155, Bruges, Belgium, 2000. Springer.

[15] Ran Canetti and Hugo Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Advances in Cryptology—EUROCRYPT '01*, pages 453–474, Innsbruck, Austria, 2001. Springer. Full version is available at http://eprint.iacr.org/2001/040.

[16] Ran Canetti and Hugo Krawczyk. Universally Composable Notions of Key Exchange and Secure Channels (Extended Abstract). In *Advances in Cryptology—EUROCRYPT '02*, pages 337–351, Amsterdam, The Netherlands, 2002. Springer. Full version is available at http://eprint.iacr.org/2002/059.

[17] Hugo Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol (Extended Abstract). In *Advances in Cryptology—CRYPTO'05*, pages 546–566, Santa Barbara, California, 2005. Springer. Full version is available at http://eprint.iacr.org/2005/176.

[18] Brian LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger Security of Authenticated Key Exchange. In *Proceedings of the 1st International Conference on Provable Security* (*ProvSec '07*), pages 1–16, Wollongong, Australia, 2007. Springer.

[19] Augustin P. Sarr, Philippe Elbaz-Vincent, and Jean-Claude Bajard. A New Security Model for Authenticated Key Agreement. In *Proceedings of the 7th International Conference on Security and Cryptography for Networks* (*SCN '10*), pages 219–234, Amalfi, Italy, 2010. Springer.

[20] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer, 2003.

[21] Kim-Kwang Raymond Choo, Colin Boyd, and Yvonne Hitchcock. Examining Indistinguishability-Based Proof Models for Key Establishment Protocols. In *Advances in Cryptology—ASIACRYPT '05*, pages 585–604, Chennai, India, 2005. Springer.

[22] Cas J. Cremers. Session-state Reveal Is Stronger Than Ephemeral Key Reveal: Attacking the NAXOS Authenticated Key Exchange Protocol. In *Proceedings of the 7th International Conference on Applied Cryptography and Network Security* (*ACNS '09*), pages 20–33, Paris-Rocquencourt, France, 2009. Springer.

[23] Cas Cremers. Examining Indistinguishability-Based Security Models for Key Exchange Protocols: The Case of CK, CK-HMQV, and eCK. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS '11)*, pages 80–91, Hong Kong, China, 2011. ACM.

[24] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-Box Concurrent Zero-Knowledge Requires $\tilde{\Omega}(\log n)$ Rounds (Extended Abstract). In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC '01)*, pages 570–579, Hersonissos, Greece, 2001. ACM. See [101] for the journal version.

[25] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent Zero-Knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Csomputing (STOC '98)*, pages 409–418, New York, NY, USA, 1998. ACM. See [102] for the conference version.

[26] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable Zero-Knowledge (Extended Abstract). In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC '00)*, pages 235–244, Portland, Oregon, USA, 2000. ACM.

[27] Boaz Barak. How to Go Beyond the Black-Box Simulation Barrier. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 106–115, Las Vegas, Nevada, USA, 2001. IEEE Computer Society.

[28] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[29] Louis C. Guillou and Jean-Jacques Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In *Advances in Cryptology—EUROCRYPT '88*, pages 123–128. Springer, 1988.

[30] Thomas Beth. Efficient Zero-Knowledge Identification Scheme for Smart Cards. In *Advances in Cryptology—EUROCRYPT '88*, pages 77–84, Davos, Switzerland, 1988. Springer.

[31] Kazuo Ohta and Tatsuaki Okamoto. A Modification of the Fiat-Shamir Scheme. In *Advances in Cryptology—CRYPTO '88*, pages 232–243, Santa Barbara, California, USA, 1990. Springer.

[32] Silvio Micali and Adi Shamir. An Improvement of the Fiat-Shamir Identification and Signature Scheme. In *Advances in Cryptology—CRYPTO '88*, pages 244–247, Santa Barbara, California, USA, 1990. Springer.

[33] Claus P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology—CRYPTO '89*, pages 239–252, Santa Barbara, California, USA, 1989. Springer. See [103] for the journal version.

[34] Jacques Stern. An Alternative to the Fiat-Shamir Protocol. In *Advances in Cryptology—EUROCRYPT '89*, pages 173–180, Houthalen, Belgium, 1989. Springer.

[35] Christoph G. Günther. An Identity-Based Key-Exchange Protocol. In *Advances in Cryptology—EUROCRYPT '89*, pages 29–37, Houthalen, Belgium, 1989. Springer.

[36] Adi Shamir. An Efficient Identification Scheme Based on Permuted Kernels (Extended Abstract). In *Advances in Cryptology—CRYPTO '89*, pages 606–609, Santa Barbara, California, USA, 1990. Springer.

[37] Ernest F. Brickell and Kevin S. McCurley. An Interactive Identification Scheme Based on Discrete Logarithms and Factoring (Extended Abstract). In *Advances in Cryptology—EUROCRYPT '90*, pages 63–71, Aarhus, Denmark, 1991. Springer. See [104] for the journal version.

[38] Marc Girault. An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In *Advances in Cryptology—EUROCRYPT '90*, pages 481–486, Aarhus, Denmark, 1991. Springer.

[39] Samy Bengio, Gilles Brassard, Yvo G. Desmedt, Claude Goutier, and Jean-Jacques Quisquater. Secure Implementation of Identification Systems. *Journal of Cryptology*, 4(3):175–183, 1991.

[40] Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Advances in Cryptology—CRYPTO '92*, pages 31–53, Santa Barbara, California, USA, 1992. Springer.

[41] Lidong Chen and Ivan Damgård. Security Bounds for Parallel Versions of Identification Protocols (Extended Abstract). In *Advances in Cryptology—EUROCRYPT '92*, pages 461–466, Balatonfüred, Hungary, 1992. Springer.

[42] M. Burmester, Y. Desmedt, and T. Beth. Efficient Zero-Knowledge Identification Schemes for Smart Cards. *The Computer Journal*, 35(1):21–29, 1992.

[43] Jacques Stern. A New Identification Scheme Based on Syndrome Decoding. In *Advances in Cryptology—CRYPTO '93*, pages 13–21, Santa Barbara, California, USA, 1994. Springer.

[44] Victor Shoup. On the Security of a Practical Identification Scheme. *Journal of Cryptology*, 12(4):247–260, 1999. See [105] for the conference version.

[45] Mihir Bellare and Adriana Palacio. GQ and Schnorr Identification Schemes: Proofs of Security against Impersonation under Active and Concurrent Attacks. In *Advances in Cryptology—CRYPTO '02*, pages 162–177, Santa Barbara, California, USA, 2002. Springer.

[46] Vadim Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *Public Key Cryptography* (*PKC 2008*), pages 162–179. Springer, 2008.

[47] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *Advances in Cryptology—ASIACRYPT 2008*, pages 372–389. Springer, 2008.

[48] Vadim Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Advances in Cryptology—ASIACRYPT 2009*, pages 598–616. Springer, 2009.

[49] Keita Xagawa and Keisuke Tanaka. Zero-Knowledge Protocols for NTRU: Application to Identification and Proof of Plaintext Knowledge. In *Proceedings of the 3rd International Conference on Provable Security—ProvSec 2009*, pages 198–213. Springer, 2009.

[50] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. Improved Zero-Knowledge Identification with Lattices. In *Proceedings of the 4th International Conference on Provable Security—ProvSec 2010*, pages 1–17. Springer, 2010.

[51] Pierre-Louis Cayrel and Pascal Véron. Improved Code-Based Identification Scheme, 2010. Unpublished manuscript. Available from `http://arxiv.org/abs/1001.3017v1`.

[52] Rosemberg Silva, Pierre-Louis Cayrel, and Richard Lindner. Zero-knowledge Identification based on Lattices with Low Communication Costs. In *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais* (*SBSEG*), 2011. Available from `http://www.cayrel.net/PublicationsCayrel/2011%20-%20ZK%20Id%20based%20on%20Lattices%20with%20Low%20Com%20Cost.pdf`.

[53] Yvo Desmedt, Claude Goutier, and Samy Bengio. Special Uses and Abuses of the Fiat-Shamir Passport Protocol (extended abstract). In *Advances in Cryptology—CRYPTO '87*, pages 21–39. Springer, 1988.

[54] Thomas Beth and Yvo Desmedt. Identification Tokens — or: Solving The Chess Grandmaster Problem. In *Advances in Cryptology—CRYPTO '90*, pages 169–176, Santa Barbara, California, USA, 1991. Springer.

[55] Mihir Bellare, Marc Fischlin, Shafi Goldwasser, and Silvio Micali. Identification Protocols Secure against Reset Attacks. In *Advances in Cryptology—EUROCRYPT 2001*, pages 495–511. Springer, 2001.

[56] Yair Oren. On the Cunning Power of Cheating Verifiers: Some Observations about Zero Knowledge Proofs (Extended Abstract). In Ashok K. Chandra, editor, *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '87*), pages 462–471, Los Angeles, California, USA, 1987. IEEE Computer Society Press. See [106] for the journal version.

[57] Daniele Micciancio and Salil P. Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In *Advances in Cryptology—CRYPTO 2003*, pages 282–298. Springer, 2003.

[58] Jacques Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[59] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *Proceedings of the 3rd International Symposium on Algorithmic Number Theory* (*ANTS-III*), pages 267–288. Springer, 1998.

[60] Rosemberg Silva, Antonio C. de A. Campello Jr., and Ricardo Dahab. LWE-based Identification Schemes. In *Proceedings of the 11th IEEE Information Theory Workshop* (*ITW 2011*), pages 292–296. IEEE Computer Society, 2011.

[61] Oded Goldreich and Hugo Krawczyk. On the Composition of Zero-Knowledge Proof Systems. In Mike Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming* (*ICALP '90*), volume 443 of *Lecture Notes in Computer Science*, pages 268–282, Warwick University, England, 1990. Springer. See [1] for the journal version.

[62] Toshiya Itoh and Kouichi Sakurai. On the Complexity of Constant Round ZKIP of Possession of Knowledge. In *Advances in Cryptology—ASIACRYPT '91*, pages 331–345, Fujiyoshida, Japan, 1991. Springer.

[63] Jonathan Katz. Which Languages Have 4-Round Zero-Knowledge Proofs? In *Theory of Cryptography—TCC 2008*, pages 73–88, New York, NY, USA, 2008. Springer.

[64] Tatsuaki Okamoto, David Chaum, and Kazuo Ohta. Direct Zero Knowledge Proofs of Computational Power in Five Rounds. In *Advances in Cryptology—EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 96–105. Springer Berlin / Heidelberg, 1991.

[65] Ivan Damgård. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. In *Advances in Cryptology—EUROCRYPT '00*, pages 418–430, Bruges, Belgium, 2000. Springer.

[66] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information. In *Proceedings of the 14th Annual ACM Symposium on Theory of Csomputing* (*STOC '82*), pages 365–377, New York, NY, USA, 1982. See [107] for the journal version.

[67] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Advances in Cryptology—CRYPTO '86*, pages 186–194, Santa Barbara, California, USA, 1987. Springer.

[68] Keita Xagawa. *Cryptography with Lattices*. PhD thesis, Tokyo Institute of Technology, Tokyo, Japan, 2010. Available from `http://xagawa.net/pdf/2010Thesis.pdf`.

[69] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with Small Parameters. In *Advances in Cryptology—CRYPTO 2013*, pages 21–39, Santa Barbara, California, USA, 2013. Springer. Full version is available at `http://eprint.iacr.org/2013/069`.

[70] Daniele Micciancio and Oded Regev. Lattice-Based Cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.

[71] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *Advances in Cryptology–ASIACRYPT 2011*, pages 1–20. Springer, 2011.

[72] Daniele Micciancio. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '02*), pages 356–365, Vancouver, British Columbia, Canada, 2002. IEEE Computer Society. See [108] for the journal version.

[73] Chris Peikert and Alon Rosen. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *Proceedings of the Third conference on Theory of Cryptography*, Theory of Cryptography—TCC '06, pages 145–166, New York, NY, USA, 2006. Springer.

[74] Vadim Lyubashevsky and Daniele Micciancio. Generalized Compact Knapsacks are Collision Resistant. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming* (*ICALP 2006*), pages 144–155. Springer, 2006.

[75] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFT: A Modest Proposal for FFT Hashing. In *Fast Software Encryption—FSE '08*, pages 54–72. Springer, 2008.

[76] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In *Advances in Cryptology—EUROCRYPT 2010*, pages 1–23, Nice, French Riviera, France, 2010. Springer.

[77] Robert J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN Progress Report 42–44*, pages 114–116, 1978.

[78] Harald Niederreiter. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory. Problemy Upravlenija i Teorii Informacii*, 15:159–166, 1986.

[79] Raphael Overbeck and Nicolas Sendrier. Code-Based Cryptography. In *Post-Quantum Cryptography*, pages 95–145. Springer, 2009.

[80] Kazukuni Kobara and Hideki Imai. Semantically Secure McEliece Public-Key Cryptosystems—Conversions For McEliece PKC. In *Public Key Cryptography* (*PKC 2001*), pages 19–35, Cheju Island, Korea, 2001. Springer.

[81] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More Constructions of Lossy and Correlation-Secure Trapdoor Functions. In *Public Key Cryptography* (*PKC 2010*), pages 279–295, 2010. See [109] for the journal version.

[82] Yuan Xing Li, Robert H. Deng, and Xin Mei Wang. On the Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[83] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and Defending the McEliece Cryptosystem. In *Proceedings of the 2nd International Workshop on Post-Quantum Cryptography* (*PQCrypto 2008*), pages 31–46, Cincinnati, Ohaio, USA, 2008. Springer.

[84] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Advances in Cryptology—EUROCRYPT 2012*, pages 700–718. Springer, 2012. Full version is available at `http://eprint.iacr.org/2011/501`.

[85] Norman Göttert, Thomas Feller, Michael Schneider, Johannes Buchmann, and Sorin Huss. On the Design of Hardware Building Blocks for Modern Lattice-Based Encryption Schemes. In *Proceedings of the 14th International Workshop Cryptographic Hardware and Embedded Systems* (*CHES 2012*), pages 512–529. Springer, 2012.

[86] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems. In *Cryptographic Hardware and Embedded Systems—CHES 2012*, pages 530–547, Leuven, Belgium, 2012. Springer.

[87] Léo Ducas, Alain Durmus, Tancrede Lepoint, and Vadim Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *Advances in Cryptology—CRYPTO 2013*, pages 40–56, Santa Barbara, California, USA, 2013. Springer.

[88] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology—CRYPTO '94*, pages 174–187. Springer, 1994.

[89] Uriel Feige and Adi Shamir. Zero Knowledge Proofs of Knowledge in Two Rounds. In *Advances in Cryptology—CRYPTO '89*, pages 526–544. Springer, 1989.

[90] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer and System Sciences* (*JCSS*), 37(2):156–189, 1988.

[91] Donald Beaver. Adaptive Zero Knowledge and Computational Equivocation. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (*STOC '96*), pages 629–638, Philadelphia, Pennsylvania, USA, 1996. ACM.

[92] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (*STOC '98*), pages 141–150, Dallas, Texas, USA, 1998. ACM.

[93] Giovanni Di Crescenzo and Rafail Ostrovsky. On Concurrent Zero-Knowledge with Pre-Processing. In *Advances in Cryptology—CRYPTO '99*, pages 485–502, Santa Barbara, California, USA, 1999. Springer.

[94] Marc Fischlin. *Trapdoor Commitment Schemes and Their Applications*. PhD thesis, Department of Mathematics, 2001. Available from `http://www.math.uni-frankfurt.de/~dmst/research/phdtheses/mfischlin.dissertation.2001.html`.

[95] Chris Peikert. Bonsai Trees (or, Arboriculture in Lattice-Based Cryptography). Cryptology ePrint Archive, 2009. Available from `http://eprint.iacr.org/2009/359`.

[96] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In *Advances in Cryptology—EUROCRYPT 2012*, pages 523–552. Springer, 2010. See [110] for the journal version.

[97] Olivier Blazy, Céline Chevalier, Léo Ducas, and Jiaxin Pan. Exact Smooth Projective Hash Function based on LWE. Cryptology ePrint Archive, 2013. Available from http://eprint.iacr.org/2013/821.

[98] Ronald Cramer and Victor Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Advances in Cryptology—Eurocrypt 2002*, pages 45–64, Amsterdam, The Netherlands, 2002. Springer.

[99] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. How to Use a Short Basis: Trapdoors for Hard Lattices and New Cryptographic Constructions, 2008. Available from http://people.csail.mit.edu/cpeikert/pubs/trap_lattice.pdf. See [111] for the conference version.

[100] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols (Extended Abstract). In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science* (*FOCS '01*), page 136, Washington, DC, USA, 2001. IEEE Computer Society. See [8] for the full version.

[101] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-Box Concurrent Zero-Knowledge Requires (Almost) Logarithmically Many Rounds. *SIAM Journal on Computing*, 32(1):1–47, January 2002. See [24] for the conference version.

[102] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent Zero-Knowledge. *Journal of the ACM* (*JACM*), 51(6):851–898, November 2004. See [25] for the conference version.

[103] Claus-Peter Schnorr. Efficient Signature Generation for Smart Cards. *Journal of Cryptology*, 4(3):239–252, 1991. See [33] for the conference version.

[104] Ernest F. Brickell and Kevin S. McCurley. An Interactive Identification Scheme Based on Discrete Logarithms and Factoring. *Journal of Cryptology*, 5(1):29–39, 1992. See [37] for the conference version.

[105] Victor Shoup. On the Security of a Practical Identification Scheme. In *Advances in Cryptology—EUROCRYPT '96*, pages 344–353, Saragossa, Spain, 1996. Springer. See [44] for the journal version.

[106] Oded Goldreich and Yair Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology*, 7:1–32, 1994. See [56] for the conference version.

[107] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Sciences* (*JCSS*), 28(2):270–299, 1984. See [66] for the conference version.

[108] Daniele Micciancio. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions. *Computational Complexity*, 16(4):365–411, 2007. See [72] for the conference version.

[109] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More Constructions of Lossy and Correlation-Secure Trapdoor Functions. *Journal of Cryptology*, 26(1):39–74, 2013. See [81] for the conference version.

[110] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. *Journal of Cryptology*, 25(4):601–639, 2012. See [96] for the conference version.

[111] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions (Extended Abstract). In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing* (*STOC 2008*), pages 197–206, Dallas, Texas, USA, 2008. ACM. See [99] for the full version.

[112] Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical Secrecy and Multi-Bit Commitments. Technical report, Basic Research in Computer Science (BRICS), University of Aarhus, Denmark, 1996. Available from http://www.brics.dk/RS/96/45/. See also [112] for the journal version.

[113] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing* (*STOC '05*), pages 84–93, Baltimore, Maryland, USA, 2005. ACM. See [113] for the journal version.

[114] Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical Secrecy and Multibit Commitments. *IEEE Transactions on Information Theory*, 44(3):1143–1151, 1998. See [114] for the technical report.

[115] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *Journal of the ACM*, 56(6):1–40, 2009. See [115] for the conference version.

[116] Michael O. Rabin. How To Exchange Secrets with Oblivious Transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981. Available from: `http://eprint.iacr.org/2005/187.pdf`.

[117] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*, volume 671 of *The Springer International Series in Engineering and Computer Science*. Springer, 2002.

[118] Salil Pravin Vadhan. *A Study of Statistical Zero-Knowledge Proofs*. PhD thesis, Department of Mathematics, 1999. Available from `http://groups.csail.mit.edu/cis/theses/salil-phd.ps`.

[119] Shai Halevi and Silvio Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In *Advances in Cryptology—CRYPTO '96*, pages 201–215. Springer, 1996.

[120] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (True) Complexity of Statistical Zero Knowledge. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing* (*STOC '90*), pages 494–502, Baltimore, Maryland, USA, 1990. ACM.

[121] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In *Advances in Cryptology—CRYPTO '92*, pages 390–420, Santa Barbara, California, USA, 1993. Springer.

[122] Shimon Even, Alan L. Selman, and Yacov Yacobi. The Complexity of Promise Problems with Applications to Public-Key Cryptography. *Information and Control*, 61(2):159–173, 1984.

[123] Daniele Micciancio. Efficient Reductions Among Lattice Problems. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2008, pages 84–93, San Francisco, California, 2008. Society for Industrial and Applied Mathematics (SIAM).

[124] Joop H. van de Pol. Lattice-Based Cryptography. Master's thesis, Department of Mathematics and Computer Science, 2011. Available from `http://www.cs.bris.ac.uk/pgrad/csjhvdp/files/ThesisJvdPol.pdf`.

[125] Kenneth Ireland and Michael Rosen. *A Classical Introduction to Modern Number Theory*, volume 84 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 1990.

[126] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing* (*STOC '09*), pages 333–342, Bethesda, Maryland, USA, 2009. ACM. Full version is available at `http://eprint.iacr.org/2008/481`.

[127] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical Hardness of Learning with Errors. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing* (*STOC '13*), pages 575–584, Palo Alto, California, USA, 2013. ACM. Full version is available at `http://arxiv.org/abs/1306.0281`.

A Omitted Definitions & Lemmas

A.1 Trapdoor One-Way Permutations (TDP)

Informally, a trapdoor one-way permutation is a permutation having three properties: (1) it is easy to compute, (2) it is hard to invert, and (3) there exists auxiliary information, such that it is easy to invert the permutation if the auxiliary information is known. A formal definition follows:

Definition 3 (Collection of Trapdoor One-Way Permutations). Let Π_n be a set of permutations, such that for any permutation $\pi_n \in \Pi_n$, we have $\mathsf{dom}(\pi_n) \subseteq \{0,1\}^n$. A family of such sets $\Pi = \{\Pi_n\}_{n\in\mathbb{N}}$ is called a *collection of trapdoor one-way permutations* (*TDP*) if there exist two PPT algorithms GenP and Samp, and two *deterministic* polynomial-time algorithms Eval and InvP, such that the following conditions hold:

1. **Easy to generate:** On input 1^n, algorithm GenP picks a permutation $\pi_n \in \Pi_n$, and outputs the description of π_n denoted $\mathsf{desc}(\pi_n)$, as well as the associated trapdoor t_n. In order to avoid mentioning 1^n explicitly in the input algorithms such as Samp, Eval, and InvP, we assume that $|\mathsf{desc}(\pi_n)| \geq n$.

2. **Easy to sample the domain:** On input $\mathsf{desc}(\pi_n)$, algorithm $\textsc{Samp}$ chooses an element from $\mathsf{dom}(\pi_n) \subseteq \{0,1\}^n$.

3. **Easy to evaluate:** On input $\mathsf{desc}(\pi_n)$ and $x \in \mathsf{dom}(\pi_n)$, the output of the algorithm $\textsc{Eval}$ is $\pi_n(x)$. If the input is malformed, $\textsc{Eval}$ returns a special symbol $\perp$, indicating failure.

4. **Easy to invert with the trapdoor:** On input $\mathsf{desc}(\pi_n)$, t_n, and $y \in \mathsf{range}(\pi_n)$, the algorithm $\textsc{InvP}$ outputs ${\pi_n}^{-1}(y)$. Moreover, if $y \notin \mathsf{range}(\pi_n)$, then $\textsc{InvP}(\mathsf{desc}(\pi_n), t_n, y)$ outputs a special symbol $\perp$, indicating failure.

5. **Hard to invert without the trapdoor:** For any PPT algorithm A, for every $c \in \mathbb{N}$, and for all sufficiently large n, the advantage of A:

$$\mathbf{Adv}^{\textsc{Invert}}_{A,\textsc{Gen4}} \stackrel{\text{def}}{=} \Pr\left[A(\mathsf{desc}(\pi_n), y) = x \;\middle|\; (\mathsf{desc}(\pi_n), t_n, x, y) \leftarrow \textsc{Gen4}(1^n)\right] \,, \tag{12}$$

is less than n^{-c}. The probability is taken over the random coins of A and $\textsc{Gen4}$, where the latter is defined on 1^n by the following experiment:

$$\begin{aligned}
&(\mathsf{desc}(\pi_n), t_n) \leftarrow \textsc{GenP}(1^n),\\
&x \leftarrow \textsc{Samp}(\mathsf{desc}(\pi_n)),\\
&y \leftarrow \textsc{Samp}(\mathsf{desc}(\pi_n), x)\\
&\texttt{OUTPUT}\ (\mathsf{desc}(\pi_n), t_n, x, y)\ .
\end{aligned}$$

○

In this paper, we make use of the Rabin's trapdoor one-way permutation [116] for counter-examples: Let $m = pq$ be a secure RSA modulus of size n, and let QR_m be the set of quadratic residues modulo m. The Rabin's TDP is defined as follows:

$$\begin{aligned}
\pi_n \colon QR_m &\to QR_m\\
x &\mapsto x^2 \pmod{m}\ .
\end{aligned}$$

A.2 Statistical Distance

Let X and Y be two discrete random variables. The *statistical distance* of X and Y, denoted $\Delta(X\,;Y)$, is defined as:

$$\Delta(X\,;Y) \stackrel{\text{def}}{=} \frac{1}{2}\sum_{s} \left|\,\Pr[X=s] - \Pr[Y=s]\right|\ .$$

Like any notion of "distance," the statistical distance satisfies the triangle inequality:

Fact 1 (Triangle Inequality). $\Delta(X\,;Z) \leq \Delta(X\,;Y) + \Delta(Y\,;Z)$ *for any three random variables* X*,* Y*, and* Z*.*

Let $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_n\}_{n\in\mathbb{N}}$ be two discrete distribution ensembles. We call $\mathcal{X}$ and $\mathcal{Y}$ *statistically indistinguishable* or *statistically close*, denoted $\mathcal{X} \stackrel{S}{\approx} \mathcal{Y}$, if there exists a constant $\delta > 0$, such that for all $n \in \mathbb{N}$, we have $\Delta(X_n; Y_n) \leq 2^{-\delta n}$.

Define the *joint support* of two random variables as the union of their supports; i.e., $[X, Y] = [X] \cup [Y]$.

It is well-known that processing cannot increase the statistical distance. Below, we will see two versions of this theorem. The first version only considers "bijective" procedures:

Lemma 4. *Let* X *and* Y *be two random variables with joint support* $\mathcal{S}$*, and let* $g\colon S \to S$ *be a deterministic bijection. Then,* $\Delta(g(X)\,; g(Y)) = \Delta(X\,;Y)$*.*

Proof. Since g is injective, $s' = g^{-1}(s)$ is defined for any $s \in S$. Moreover, because g is surjective, $g(s') \in S$ is equivalent to $s' \in S$. Therefore:

$$\begin{aligned}
\Delta(g(X)\,;g(Y)) &= \frac{1}{2}\sum_{s\in S} |\Pr[g(X)=s] - \Pr[g(Y)=s]| \\
&= \frac{1}{2}\sum_{s\in S} \left|\Pr[X=g^{-1}(s)] - \Pr[Y=g^{-1}(s)]\right| \\
&= \frac{1}{2}\sum_{g(s')\in S} |\Pr[X=s'] - \Pr[Y=s']| \\
&= \frac{1}{2}\sum_{s'\in S} |\Pr[X=s'] - \Pr[Y=s']| \\
&= \Delta(X\,;Y) \ .
\end{aligned}$$

■

The following fact is a generalization of Lemma 4, where g is no longer limited to deterministic bijections. The fact is formally stated and proven in, say [117, p. 159]:

Fact 2. *Let X and Y be two random variables with joint support $\mathcal{S}$, and let g be a possibly randomized function defined over $\mathcal{S}$. Then, $\Delta(g(X)\,;g(Y)) \leq \Delta(X\,;Y)$.*

Noting that the statistical distance is zero for identically distributed random variables, the following corollary is immediate.

Corollary 1. *If X and Y are identically distributed with joint support $\mathcal{S}$, then $f(X)$ and $f(Y)$ are identically distributed, for any (possibly randomized) function f defined over $\mathcal{S}$.*

Here's another useful fact, adapted from Fact 3.1.14 in [118, p. 39]. There's a typo in the statement of Fact 3.1.14 of [118], but its proof gives the correct version:

Fact 3. *Let X_0 and X_1 be independent, Y_0, and Y_1 be independent. Then $\Delta((X_0, X_1)\,;(Y_0, Y_1)) \leq \Delta(X_0\,;Y_0) + \Delta(X_1\,;Y_1)$.*

Lemma 5. *Let X and Y be two discrete random variables, and let E and E' be events defined over the probability spaces underlying X and Y, respectively. Assume that we have $|\Pr[E] - \Pr[E']| \leq \upsilon \in [0, 1)$, and the following two conditions hold:*

1. *$X \mid E \sim Y \mid E'$ if $\Pr[E] \neq 0$ and $\Pr[E'] \neq 0$; and*
2. *$X \mid \overline{E} \sim Y \mid \overline{E'}$ if $\Pr[\overline{E}] \neq 0$ and $\Pr[\overline{E'}] \neq 0$.*

Then $\Delta(X\,;Y) \leq \upsilon$, irrespective of the values of $\Pr[E]$ and $\Pr[E']$.

Proof. Let us first consider the special cases, i.e., $\Pr[E] \in \{0, 1\}$ or $\Pr[E'] \in \{0, 1\}$. Notice that by symmetry, we can examine only the case where $\Pr[E'] = 0$; the lemma for other cases follow similarly. Let $e \stackrel{\text{def}}{=} \Pr[E]$. From $|\Pr[E] - \Pr[E']| \leq \upsilon \in [0, 1)$, we get $e \leq \upsilon < 1$. Therefore, $\Pr[\overline{E}] = 1 - e \geq 1 - \upsilon > 0$ and $\Pr[\overline{E'}] = 1$, and it follows from condition 2 that $X \mid \overline{E} \sim Y \mid \overline{E'} \equiv Y$. Let S be the joint support of X and Y. Applying the law of total probability, for any $s \in S$ we have:

$$\begin{aligned}
\Pr[X=s] &= \Pr[E]\Pr[X=s \mid E] + \Pr[\overline{E}]\Pr[X=s \mid \overline{E}] \\
&= e\Pr[X=s \mid E] + (1-e)\Pr[Y=s \mid \overline{E'}] \\
&= \Pr[Y=s] + e\left(\Pr[X=s \mid E] - \Pr[Y=s]\right) \ .
\end{aligned}$$

Therefore,

$$|\Pr[X = s] - \Pr[Y = s]| = e\,|\Pr[X = s \mid E] - \Pr[Y = s]| \ ,$$

and:

$$\begin{aligned}
\Delta(X\,;Y) &= \frac{1}{2}\sum_{s\in S}|\Pr[X = s] - \Pr[Y = s]| \\
&= \frac{e}{2}\sum_{s\in S}\big|\Pr[X = s \mid E] - \Pr[Y = s]\big| \\
&\le \frac{e}{2}\Big(\sum_{s\in S}\Pr[X = s \mid E] + \sum_{s\in S}\Pr[Y = s]\Big) \\
&= \frac{e}{2}\cdot 2 = e \le \upsilon \ .
\end{aligned}$$

We now pertain to the general case, where $\Pr[E] \notin \{0,1\}$ and $\Pr[E'] \notin \{0,1\}$. Let $\delta \stackrel{\text{def}}{=} \Pr[E] - \Pr[E']$, and therefore $|\delta| \le \upsilon$. Notice that we have $\Pr[\overline{E'}] - \Pr[\overline{E}] = \delta$. By assumption, for any $s \in S$,

$$\Pr[X = s|E] = \Pr[Y = s|E'] \ , \tag{13}$$

$$\Pr[X = s|\overline{E}] = \Pr[Y = s|\overline{E'}] \ . \tag{14}$$

Multiplying both sides of Equations (13) and (14) by $\Pr[E] = \Pr[E'] + \delta$ and $\Pr[\overline{E}] = \Pr[\overline{E'}] - \delta$ respectively, we have:

$$\Pr[X = s, E] = \Pr[Y = s, E'] + \delta\cdot\Pr[Y = s|E'] \ , \tag{15}$$

$$\Pr[X = s, \overline{E}] = \Pr[Y = s, \overline{E'}] - \delta\cdot\Pr[Y = s|\overline{E'}] \ . \tag{16}$$

Adding both sides of Equations (15) and (16), and using the law of total probability, we obtain $\Pr[X = s] = \Pr[Y = s] + \delta\cdot\big(\Pr[Y = s|E'] - \Pr[Y = s|\overline{E'}]\big)$. Therefore,

$$\begin{aligned}
\Delta(X\,;Y) &= \frac{1}{2}\sum_{s\in S}|\Pr[X = s] - \Pr[Y = s]| \\
&= \frac{|\delta|}{2}\sum_{s\in S}\big|\Pr[Y = s|E'] - \Pr[Y = s|\overline{E'}]\big| \\
&\le \frac{|\delta|}{2}\Big(\sum_{s\in S}\Pr[Y = s|E'] + \sum_{s\in S}\Pr[Y = s|\overline{E'}]\Big) \\
&= \frac{|\delta|}{2}\cdot 2 = |\delta| \le \upsilon \ . \qquad\blacksquare
\end{aligned}$$

A.3 Commitments

A commitment scheme is a protocol between two entities, the *sender* (S) and the *receiver* (R). The protocol consists of two phases: The commitment phase, and the reveal phase. Informally, it is required that: (1) S and R accept at the end of both phases; (2) in the commitment phase, R learns nothing about the value S committed to, and (3) S cannot change this value in the reveal phase.

In this paper, we are only interested in commitments with *non-interactive* commitment and reveal phases. That is, S sends a single message in the commitment phase, and a single message in the reveal phase, but R does not send any messages during the whole protocol.

For notational simplicity, we assume that the commitment is performed on bit strings. Let $\textsc{Com}_n\colon \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ denote an efficient and *deterministic* algorithm defined, where $\ell(n)$ and $m(n)$ are polynomial in n.

For any $n \in \mathbb{N}$, let the description of $\textsc{Com}_n$ be generated by a PPT algorithm $\textsc{GenC}$. That is, $\mathsf{desc}(\textsc{Com}_n) \leftarrow \textsc{GenC}(1^n)$. In order to avoid mentioning 1^n explicitly in the input other algorithms, we assume that $|\mathsf{desc}(\textsc{Com}_n)| \geq n$. In general, the sender and the receiver will agree on $\mathsf{desc}(\textsc{Com}_n)$ prior to the main protocol, perhaps during an initial phase or via a trusted setup.

We also assume that $\mathsf{desc}(\textsc{Com}_n)$ includes the description of some random variable $\textsc{Rnd}_{\ell(n)}$ over $\{0,1\}^{\ell(n)}$. If we only specify the first input to $\textsc{Com}_n$, the second input will be chosen according to $\textsc{Rnd}_{\ell(n)}$. That is, given $x \in \{0,1\}^n$, we commit to x by first picking $r \leftarrow \textsc{Rnd}_{\ell(n)}$, and then computing $\textsc{Com}_n(x\,;r)$. Let $\textsc{Com}_n(x)$ denote the random variable induced by this process.

Definition 4 (Non-interactive Statistically-Hiding Commitments). A PPT algorithm $\textsc{GenC}$ is called a generator for a non-interactive *statistically hiding* (and *computationally binding*) commitment scheme, if the following conditions hold:

1. **Computational Binding:** No efficient algorithm can decommit to a value it did not commit to. Specifically, for any PPT algorithm A, any $c \in \mathbb{N}$, and all sufficiently large n, the following quantity:

$$\mathbf{Adv}^{\textsc{Binding}}_{A,\textsc{GenC}}(n) \stackrel{\text{def}}{=} \Pr\left[\begin{array}{l} \textsc{Com}_n(x\,;r) = \textsc{Com}_n(x'\,;r')\ , \\ \text{ and } x \neq x', \text{ and } x,x' \in \{0,1\}^n\ , \\ \text{ and } r,r' \in \{0,1\}^{\ell(n)} \end{array} \middle| \begin{array}{l} \mathsf{desc}(\textsc{Com}_n) \leftarrow \textsc{GenC}(1^n)\ , \\ (x,x',r,r') \leftarrow A(\mathsf{desc}(\textsc{Com}_n)) \end{array}\right]\ ,$$

   is less than n^{-c}, where the probability is taken over the coin tosses of A and $\textsc{GenC}$.

2. **Statistical Hiding:** Commitments to values of the same length n are statistically indistinguishable. That is, there exists a constant $\delta > 0$, such that for all $n \in \mathbb{N}$, any $\mathsf{desc}(\textsc{Com}_n) \in [\textsc{GenC}(1^n)]$, and all $x, x' \in \{0,1\}^n$:

$$\Delta\left(\textsc{Com}_n(x)\,;\textsc{Com}_n(x')\right) \leq 2^{-\delta n}\ . \tag{17}$$

We call $\textsc{Com}_n$ a *non-interactive statistically-hiding commitment scheme* if $\mathsf{desc}(\textsc{Com}_n) \in [\textsc{GenC}(1^n)]$.$\bigcirc$

Remark 8. Note that both of the binding and hiding properties are defined in a strong sense. A weaker binding property can be obtained by asking A to output, given $\mathsf{desc}(\textsc{Com}_n)$ and some $(x,r) \in \{0,1\}^n \times \{0,1\}^{\ell(n)}$, a pair $(x',r') \in \{0,1\}^n \times \{0,1\}^{\ell(n)}$, such that $x' \neq x$, and $\textsc{Com}_n(x',r') = \textsc{Com}_n(x,r)$. This definition is weaker since A must satisfy a harder condition: (x,r) is fixed a priori, and A is not free to choose it.

A weaker hiding property can be obtained by requiring that an *overwhelming* fraction (rather than *all*) of the support of $\textsc{GenC}(1^n)$ satisfy Equation 17. This is equivalent to requiring that Equation 17 holds over the random coins of $\textsc{GenC}(1^n)$ with overwhelming probability.

In this paper, we did *not* adopt the weaker definitions of hiding and binding for two reasons: (1) The well-known instances of the statistical-hiding commitments, such as [47, 114, 119], satisfy the strong variation, and (2) proving theorems are easier with the strong definition. $\triangleleft$

A.4 Zero Knowledge

Informally, a protocol $\langle V, P\rangle$ is called zero knowledge (ZK) for P (the prover), if at the end of the execution, party V (the verifier) does not learn anything about the private input of P, which she could not learn by herself before the start of the protocol. This is the case even if the verifier deviates from the protocol arbitrarily. We denote by V^* the party which may or may not follow the verifier's program.

In this paper, we are only interested in *cryptographic* protocols, where the strategy of *honest* parties can be implemented in probabilistic polynomial time, while possibly giving the honest parties an extra (secret) input. In the context of ZK protocols, only the honest prover is given this type of input. This paper uses the statistical variation of ZK protocols, where the protocol remains ZK even if the cheating party V^* is infinitely powerful. Moreover, we focus on the case where the simulator is black-box. Note, however, that while V^* might be unbounded, we will assume that all prover strategies (even the cheating ones) are PPT.

Before giving the actual definition of statistical zero-knowledge protocols, let us define some notation. It is a good idea to review the notation introduced in Section 2.1 as well. Define the *view* of a party participating in a protocol as whatever it sees during the protocol, including its input, randomness, and received messages. For instance, in the protocol $\langle V_r^*, P(y)\rangle(x)$, the view of V_r^* is $(x, r, m_1, \ldots, m_k)$, where $(m_1, \ldots, m_k)$ is the sequence of messages V_r^* receives from P. We denote this view by the random variable $\mathsf{View}_{V_r^*}^{P(y)}(x)$. Note that the randomness of V^* is fixed here. Let $\mathsf{View}_{V^*}^{P(y)}(x)$ denote the random variable describing $\mathsf{View}_{V_r^*}^{P(y)}(x)$ when r is chosen uniformly at random.

In addition, let S be the simulator, which is a PPT oracle machine; i.e., S can have black-box access to an oracle, which in this case is the machine V^*. This is denoted by $S^{V^*}(x)$, and it means that S can freely reset/rewind V^*, and load any desired randomness onto V^*'s random tape. Since V^* may need an a priori unbounded number of random coins, we will assume that S has two separate random tapes, one of which is fed directly into V^*, while the other is consumed by S itself (cf. [1,120]).

Definition 5 (Statistical Zero Knowledge). The protocol $\langle V, P(y)\rangle(x)$ is (*black-box*) *statistical zero-knowledge* (SZK) for P on some relation $R = \{(x, y)\}$ if there exists a PPT algorithm S (the *simulator*) and a constant $\delta > 0$, such that for all pairs $(x, y) \in R$ and any interactive function V^*, we have

$$\Delta\big(\mathsf{View}_{V^*}^{P(y)}(x)\,;S^{V^*}(x)\big) \leq 2^{-\delta|x|} \ ,$$

where the probabilities are taken over the internal coin tosses of P, V^*, and S. ○

Notice that Definition 5 is stronger than usually defined in the literature: (1) It quantifies over *all* verifier strategies, rather than merely over *PPT* verifiers. Therefore, the verifier may use an infinitely powerful strategy, even an uncomputable one. For this reason, we used the term "interactive function" instead of "interactive Turing machine" (see [121, Section 2]). (2) It allows the verifier strategy to depend on the common input x. (3) The definition is *not asymptotic*: statistical indistinguishability is required for any x, rather than for "sufficiently large" x. (4) The statistical distance is taken to be *exponentially small* in $|x|$, rather than only *negligible* in it.

Similar to [106, Theorem 3.2], it can be shown that the class of interactive proofs satisfying our black-box SZK is a subclass of those satisfying SZK with *auxiliary input*. The proof uses the analogy between the definition of black-box zero-knowledge in [106, p. 8] and Definition 5, where V^* can depend arbitrarily on x, and therefore any auxiliary input can be incorporated into its code.

A.5 Lattices

Consider n linearly-independent vectors $\mathbf{b_1}, \ldots, \mathbf{b_n}$ in $\mathbb{R}^n$. The set of all integral linear combinations of these vectors, i.e., the set $\left\{\sum_{i=1}^n x_i \mathbf{b_i} \mid x_i \in \mathbb{Z}\right\}$ is called a *lattice*. $\mathbf{b_1}, \ldots, \mathbf{b_n}$ are the *base vectors* of the lattice, and the matrix $\mathbf{B} = [\mathbf{b_1} \mid \cdots \mid \mathbf{b_n}]$ is the lattice *basis*. The lattice generated by the basis $\mathbf{B}$ is noted by $\Lambda \stackrel{\text{def}}{=} \Lambda(\mathbf{B}) \stackrel{\text{def}}{=} \left\{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\right\}$.

For $\mathbf{B} \in \mathbb{R}^{n\times n}$ and $i \in \{1, \ldots, n\}$, define the i^{th} minima $\lambda_i\big(\Lambda(\mathbf{B})\big)$ as the radius of the smallest n-dimensional ball including i independent lattice vectors. Note that $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n < +\infty$.

Several problems are conjectured to be hard on lattices, among which we mention a few. Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be a basis of rank n:

- **Shortest Vector Problem (SVP):** Find a non-zero shortest vector in the lattice; i.e., a vector of length $\lambda_1\big(\Lambda(\mathbf{B})\big)$.

  The *approximation version* $\mathbf{SVP}_\gamma$ asks for finding a non-zero lattice vector within the γ factor of the shortest vector; that is, a non-zero vector of Λ whose length is at most $\gamma\lambda_1\big(\Lambda(\mathbf{B})\big)$.

  The *gap version* $\mathbf{GapSVP}_\gamma$ is a *promise problem* [122]: Output "YES" if $\lambda_1 \le 1$, and output "NO" if $\lambda_1 > \gamma$.

- **Closest Vector Problem (CVP):** Given a target point $\mathbf{t} \in \mathbb{R}^n$, find a lattice point $\mathbf{u} \in \Lambda(\mathbf{B})$ such that $\|\mathbf{u} - \mathbf{t}\|$ is minimized.

  The *approximation version* $\mathbf{CVP}_\gamma$ asks for finding a lattice point $\mathbf{u} \in \Lambda(\mathbf{B})$ within γ distance of the nearest lattice point to $\mathbf{t}$. In other words, find $\mathbf{u} \in \Lambda(\mathbf{B})$ such that for all $\mathbf{v} \in \Lambda(\mathbf{B})$ we have $\|\mathbf{u} - \mathbf{t}\| \le \gamma\|\mathbf{v} - \mathbf{t}\|$.

  The *gap version* $\mathbf{GapCVP}_\gamma$ is a *promise problem* [122]: Output "YES" if there exists a lattice point $\mathbf{u}$ whose distance to $\mathbf{t}$ is at most 1. Output "NO" if the distance of $\mathbf{t}$ to any lattice point is more than γ.

- **Shortest Independent Vector Problem (SIVP):** Find n linearly independent lattice vectors $\mathbf{v_1}, \ldots, \mathbf{v_n}$, such that the quantity $\max_i \|\mathbf{v_i}\|$ is minimized.

  The *approximation version* $\mathbf{SIVP}_\gamma$ asks for finding a set of n linearly independent lattice vectors $\{\mathbf{v_1}, \ldots, \mathbf{v_n}\}$, such that $\max_i \|\mathbf{v_i}\| \le \gamma\lambda_n\big(\Lambda(\mathbf{B})\big)$.

The complexity of CVP, SVP, SIVP, and their corresponding approximation and gap versions are related to each other via *reductions*. For more information, see [117, 123] and [124, Section 3.1].

It is proven that lattice problems such as CVP or SVP, are NP-hard. Therefore, the best we can hope for is to solve the approximation versions of these problems. However, even solving these problems with an approximation factor of $n^{O(1/\log\log n)}$ is NP-hard. On the other hand, approximation to within a factor of $\sqrt{n/\log n}$ is not NP-hard, unless the polynomial hierarchy collapses. In general, cryptographic constructions reduce to lattice problems with a polynomial approximation factor (see [70] and the references thereof).

Note that all problems described above are *worst-case* problems. In cryptography, we need to rely on the hardness of the *average case* problems. For instance, a cryptosystem must be hard to break when the keys are chosen *randomly*. Below, we will see two such problems: SIS and LWE.

A class of lattices, with the property that $q\mathbb{Z}^n \subseteq L \subseteq \mathbb{Z}^n$ for some integer q, is called a *q-ary lattice*. This class has interesting applications in cryptography. One special sub-class of q-ary lattices—used in this paper—is described next. Let n, m, and q be positive integers. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, define the following set of points:

$$\Lambda_q^{\perp}(\mathbf{A}) \stackrel{\text{def}}{=} \Big\{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{Az} \equiv \mathbf{0} \pmod{q}\Big\} \ . \tag{18}$$

It can be shown that any *discrete additive subgroup* of any *finite dimensional vector space* over $\mathbb{R}$ is a lattice (see for example [125, page 327]). Therefore, $\Lambda_q^{\perp}(\mathbf{A})$ denotes an m-dimensional lattice, since it is a discrete additive subgroup of $\mathbb{R}^{m\times m}$. The following *average-case* problem is defined on this class of lattices:

Short Integer Solution (SIS): For a random matrix $\mathbf{A}$, find a "short" non-zero lattice point in the lattice defined by Equation 18. More specifically, define the problem $\mathbf{SIS}^p_{q,m,n,\beta}$ as follows: Given a random matrix $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n\times m}$, find a vector $\mathbf{z} \in \Lambda_q^{\perp}(\mathbf{A}) \setminus \{\mathbf{0}\}$, such that $\|\mathbf{z}\|_p \le \beta$.

We also define the *collision-finding* $\mathbf{SIS}^p_{q,m,n,\beta}$ problem as follows: Given a random matrix $\mathbf{A} \leftarrow_R \mathbb{Z}_q^{n\times m}$, find two distinct vectors $\mathbf{z_1}, \mathbf{z_2} \in \mathbb{Z}^m$, such that $\mathbf{Az_1} \equiv \mathbf{Az_2} \pmod{q}$ and $\|\mathbf{z_1}\|_p, \|\mathbf{z_2}\|_p \le \beta$.

The following relations hold between the SIS and collision-finding SIS problems:

- **If collision-finding $\mathbf{SIS}^p_{q,m,n,\beta}$ is hard, then $\mathbf{SIS}^p_{q,m,n,\beta}$ is hard.** Assume, to the contrary, that $\mathbf{SIS}^p_{q,m,n,\beta}$ is easy. Then we find a vector $\mathbf{z_1} \in \Lambda_q^{\perp}(\mathbf{A}) \setminus \{\mathbf{0}\}$, such that $\|\mathbf{z_1}\|_p \leq \beta$. Then, the vectors $\mathbf{z_1}$ and $\mathbf{z_2} = \mathbf{0}$ constitute an answer for the collision-finding $\mathbf{SIS}^p_{q,m,n,\beta}$, contradicting the premise.
- **If $\mathbf{SIS}^p_{q,m,n,\beta}$ is hard, then collision-finding $\mathbf{SIS}^p_{q,m,n,\beta/2}$ is hard.** Assume, to the contrary, that collision-finding $\mathbf{SIS}^p_{q,m,n,\beta}$ is easy. Then we find two distinct $\mathbf{z_1}, \mathbf{z_2} \in \mathbb{Z}^m$, such that $\mathbf{Az_1} \equiv \mathbf{Az_2} \pmod q$ and $\|\mathbf{z_1}\|_p, \|\mathbf{z_2}\|_p \leq \beta/2$. Define $\mathbf{z}$ as $\mathbf{z_1} - \mathbf{z_2}$. Notice that $\mathbf{z} \in \Lambda_q^{\perp}(\mathbf{A}) \setminus \{\mathbf{0}\}$. Applying the triangle inequality, we also get $\|\mathbf{z}\|_p \leq \|\mathbf{z_1}\|_p + \|\mathbf{z_2}\|_p \leq \beta$, contradicting the premise.

Fact 4. *$\mathbf{SIS}^2_{q,m,n,\beta}$ reduces to $\mathbf{SIS}^{\infty}_{q,m,n,\beta/\sqrt{m}}$ (and the same reduction holds for the respective collision-finding problems). This is because for any $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and any vector $\mathbf{z} \in \Lambda_q^{\perp}(\mathbf{A}) \setminus \{\mathbf{0}\}$, if $\|z\|_{\infty} \leq \beta/\sqrt{m}$, then $\|z\|_2 \leq \beta$.*

Another average-case lattice problem is called "learning with errors" or LWE. Specifically, for the security parameter n, let integers $m = m(n)$ and $q = q(n)$ be polynomial in n, and let χ be a probability distribution on $\mathbb{Z}_q$. The problem $\mathbf{LWE}_{q,m,n,\chi}$ is defined as follows: Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a linear system $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \pmod q$, find the secret vector $\mathbf{s}$, where the entries of the vector $\mathbf{e}$ are i.i.d. samples from χ. Regev [115] showed that if χ is a discrete Gaussian distribution with with standard deviation roughly $\alpha q \geq 2\sqrt{n}$, then there is an efficient *quantum* reduction from solving worst-case lattice problems with approximation factor $\tilde{O}(n/\alpha)$, to solving LWE. This result was later generalized to *classical* (PPT) reductions [126, 127].