1 Introduction

In this lecture, we will treat two main parts. During the first part we continue the proof that the SIS-function $f_{\mathbf{A}}$ is a one-way function by the 'two-to-one lemma'. The second part treats a new topic, still related to cryptography: a commitment scheme with SIS as underlying problem.

2 Preliminaries

In all subsequent sections, we will use implicitly that everything is parametrized by n. So, $m, q \in \text{poly}(n)$, for example. We will use the following notation $\tilde{\forall} x \in X, P(x)$ for

$$\mathop{\mathbb{P}}_{x \in X}[P(x)] \geq 1 - \text{negl}(n).$$

Given any distribution D on X, we write $\tilde{\forall} x \leftarrow D, P(x)$ for

$$\mathop{\mathbb{P}}_{x \in D}[P(x)] \geq 1 - \text{negl}(n).$$

Note the implicit parametrization in n. We call two distributions A, B on a finite set X statistically close, denoted $A \approx_s B$, whenever

$$\Delta(A, B) := \frac{1}{2} \sum_{x \in X} |\mathbb{P}[A = x] - \mathbb{P}[B = x]| < \text{negl}(n).$$

3 One-Way function from SIS (continued)

In the previous lecture notes, we defined the function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$. In this section, we are going to prove that this function is actually a one-way function, when choosing the right parameters. According to the two-to-one lemma, we need to prove that — for certain parameters — for almost all keys, the function $f_{\cdot}$ is two-to-one (or two-to-one).

The most straightforward strategy to prove that $f_{\cdot}$ is two-to-one is by using an upper bound for the covering radius $\mu(\Lambda_q^{\perp}(\mathbf{A}))$. To obtain this, one could start by bounding λ_1 from below, hence λ_n from above, and therefore bounding the covering radius. We will do this using duality and transference theorems.

LEMMA 1 *$\Lambda_q(\mathbf{A}) = q \cdot \left(\Lambda_q^{\perp}(\mathbf{A}^T)\right)^*$, i.e. $\Lambda_q(\mathbf{A})$ and $\Lambda_q^{\perp}(\mathbf{A}^T)$ are dual lattices up to q-scaling.*

Exercise 1 Prove above lemma.

LEMMA 2 *Let q be prime. $\tilde{\forall} \mathbf{A}$, $\lambda_1(\Lambda_q(\mathbf{A})) \geq O(q^{1-n/m})$.*

PROOF: Set $S = \{\mathbf{y} + q\mathbb{Z}^m \mid \mathbf{y} \in \mathbb{Z}^m \text{ and } \|\mathbf{y}\| \leq B\}$ for the set of cosets in $\mathbb{Z}^m$. Then $|S| \leq (2B+1)^m$, and therefore, for any non-zero vector $\mathbf{x} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$,

$$\mathop{\mathbb{P}}_{\mathbf{A}}[\mathbf{A}^T\mathbf{x} \bmod q \in S] = |S|/q^m \leq (2B+1)^m q^{-m}.$$

By the union bound, we have

$$\mathop{\mathbb{P}}_{\mathbf{A}}[\lambda_1(\Lambda_q(\mathbf{A})) \leq B] = \mathop{\mathbb{P}}_{\mathbf{A}}[\exists \mathbf{x} \in \mathbb{Z}^n \text{ s.t. } \mathbf{A}^T\mathbf{x} + q\mathbb{Z}^m \in S] \leq |\mathbb{Z}^n/q\mathbb{Z}^n| \cdot \mathop{\mathbb{P}}_{\mathbf{A}}[\mathbf{A}^T\mathbf{x} + q\mathbb{Z}^m \in S]$$

$$= |S| \cdot q^{n-m} \leq (2B+1)^m q^{n-m}.$$

Choosing $2B + 1 = 2^{-n/m} q^{1-n/m} = O(q^{1-n/m})$ yields that $\lambda_1(\Lambda_q(\mathbf{A})) \geq O(q^{1-n/m})$ with error probability 2^{-n}. □

LEMMA 3 *Let $q = p^k$ be a prime power. $\tilde{\forall}\mathbf{A}$, $\lambda_1(\Lambda_q(\mathbf{A})) \geq O(q^{1-n/m})$.*

PROOF: The shortest nonzero lattice vector $\mathbf{y}$ in $\Lambda_q(\mathbf{A})$ is of the form $\mathbf{y} = \mathbf{A}^T\mathbf{x}$, with $\mathbf{x} \in \mathbb{Z}^n \backslash p\mathbb{Z}^n$. Otherwise, if $\mathbf{x} \in p\mathbb{Z}^n$, we could divide the vector $\mathbf{y}$ by p to obtain a shorter vector. For any fixed $\mathbf{x} \in \mathbb{Z}^n \backslash p\mathbb{Z}^n$, the vector $\mathbf{A}^T\mathbf{x} + q\mathbb{Z}^m$ is a random coset in $\mathbb{Z}^m$ (for random $\mathbf{A}$). Using the same set $S = \{\mathbf{y} + q\mathbb{Z}^m \mid \mathbf{y} \in \mathbb{Z}^m \text{ and } \|\mathbf{y}\| \leq B\}$ as before, we obtain

$$\mathop{\mathbb{P}}_{\mathbf{A}}[\lambda_1(\Lambda_q(\mathbf{A})) \leq B] = \mathop{\mathbb{P}}_{\mathbf{A}}[\exists \mathbf{x} \in \mathbb{Z}^n \backslash p\mathbb{Z}^n \text{ s.t. } \mathbf{A}^T\mathbf{x} + q\mathbb{Z}^m \in S] \leq |\mathbb{Z}_q^n \backslash p\mathbb{Z}_q^n| \cdot \mathop{\mathbb{P}}_{\mathbf{A}}[\mathbf{A}^T\mathbf{x} + q\mathbb{Z}^m \in S]$$

$$= (q^n - (q/p)^n)|S| \cdot q^{-m} \leq (2B+1)^m q^{n-m}.$$

Therefore, the same bound applies as in Lemma 2. □

From transference, we know that $\lambda_1(\Lambda)\mu(\Lambda^*) \leq m$. Applying both Lemma 1 and 2, we obtain

$$\tilde{\forall}\mathbf{A} : \mu(\Lambda_q^\perp(\mathbf{A})) \leq qm/\lambda_1(\Lambda_q(\mathbf{A})) \leq O(mq^{n/m}).$$

For any lattice Λ (with dimension ≥ 1) and any $\mathbf{x} \in \text{span}(\Lambda)$ it holds that $|4\mu(\Lambda) \cdot \mathcal{B}_2^n \cap (\Lambda + \mathbf{x})| \geq 2$. So, for $\Lambda = \Lambda_q^\perp(\mathbf{A})$, there exists a $\mathbf{x}' \in \mathbf{x} + \Lambda_q^\perp(\mathbf{A})$ such that $\mathbf{x}' \neq \mathbf{x}$ and $\|\mathbf{x}'\| \leq 4\mu(\Lambda)$. This rewrite as $f_\mathbf{A}(\mathbf{x}') = f_\mathbf{A}(\mathbf{x})$ for $\mathbf{x} \neq \mathbf{x}'$. Therefore, for any $\mathbf{x} \in \mathbb{Z}_q^m$, there exists a $\mathbf{x}' \neq \mathbf{x}$ such that $f_\mathbf{A}(\mathbf{x}') = f_\mathbf{A}(\mathbf{x})$ and $\|\mathbf{x}'\| \leq O(m \cdot q^{n/m})$. From this, we obtain the following.

THEOREM 4 *The SIS-based function family $f_\cdot : \mathbb{Z}_q^{n \times m} \times \{-\beta, \ldots, \beta\}^n \to \mathbb{Z}_q^m$ with parameter $\beta = O(m \cdot q^{n/m})$ is a one-way function, assuming that $\texttt{SIS}_{n,m,q,\beta}$ is hard.*

4 Commitment scheme

4.1 Introduction

A commitment protocol is one of the most basic cryptographic protocols. As often in cryptography, such a protocol is explained best in the context of two players, two people that want to interact which each other in some sense. In this case, the players have different roles. One of the players is the so-called committer, which we call Cody, and the other is the verifier, called Vera.

The committer Cody has the following role. He wants to bind himself to a certain decision, but doesn't want to reveal this decision right away, but later. Note that this binding means that Cody can't change the decision made.

The role of the verifier Vera is more abstract; her task is to check whether the committer Cody really plays fair game, e.g. that he doesn't change his decision.

People often imagine these roles in the following context. The committer Cody and verifier Vera are geographically far apart. Cody writes his decision on paper, puts the paper in a locker

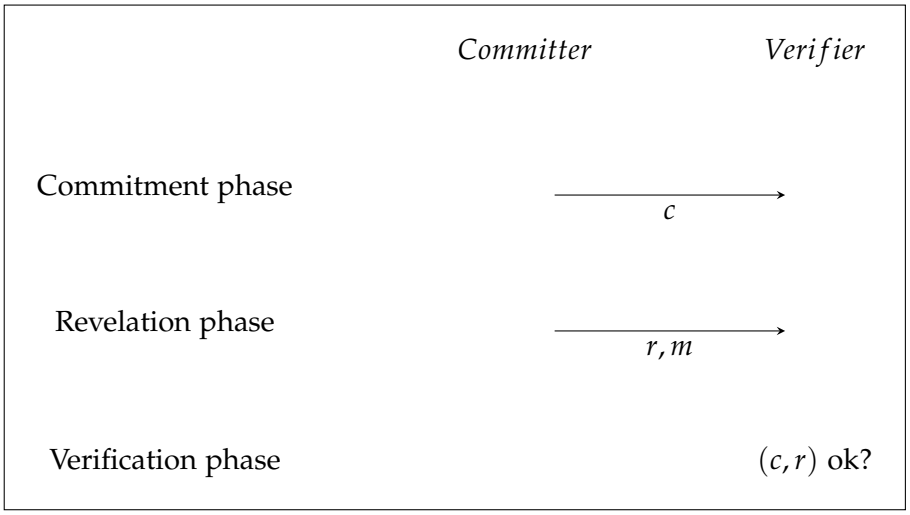

Figure 1: Schematic description of a commitment scheme

safe, locks the safe, and sends the safe to Vera, while keeping the key of the locker with himself. Vera now receives a closed locker, where she doesn't have the key of. Therefore Vera cannot see the decision of Cody. When Cody wants to reveal his decision, he sends the key to Vera, allowing her to open the locker and check the decision.

In the above story, Cody can't change his decision, and Vera can't see the decision before Cody sends her the key. In a real-life setting, people don't send safe lockers, but use cryptography. How does a commitment scheme look like, cryptographically?

In figure 1, a very broad description of a commitment scheme is depicted. Generally, there are three phases; the commitment phase, where the committer commits to his decision, the revelation phase, where the committer reveals his decision, and the verification phase, in which the verifier checks whether the committer played fair game. In the commitment phase, the committer Cody sends his commitment string c. The idea is that the verifier Vera cannot learn anything from c about the decision of Cody. Later, when Cody wants to reveal his decision, he sends the string r and his decision m to Vera. With this extra information, Vera can check whether Cody really committed to decision m in the commitment phase. The idea is that the commitment c could only reveal the decision m, and no other decision. This is called *binding*; in a later section this notion is made precise.

4.2 Definition

In the following definition, we denote by PK a space of public keys, by $\mathcal{C}$ a commitment string space and by R a space of random strings with distribution D. The star in $\{1\}^*$ is the Kleene star.

Definition 5 (Commitment scheme) *A commitment scheme is a triple of algorithms:*

- $\texttt{Keygen} : \{1\}^* \to PK$*, the (public) key generation function. It has (formally) as input a string of the form* 1^n*, allowing the generation function to use a* $\text{poly}(n)$ *amount of time.*
- $\texttt{Commit} : PK \times \{0,1\} \times R \to \mathcal{C}$*, the commitment function. It has as input a public key, a bit (to which the committer commits) and some random string, generated according to the distribution* D*. The output is an element in the commitment string space.*

- $\texttt{Verif} : PK \times \{0,1\} \times R \times C \to \{0,1\}$, *the verifying function. It has as input a public key, a bit (which the verifier tries to verify), a string from R and the commitment string.*

Definition 6 *A commitment scheme is called*

- ***correct*** *if* $\tilde{\forall} pk \leftarrow KeyGen(1^n), \forall \mu, \tilde{\forall} r$ *it holds that* $\texttt{Verif}(pk, \mu, r, c) = 1$, *where* $c = \texttt{Commit}(pk, \mu, r)$.
- ***statistically hiding*** *if* $\tilde{\forall} pk \leftarrow KeyGen(1^n)$ *holds*

$$\texttt{Commit}(pk, 0, r) \approx_s \texttt{Commit}(pk, 1, r) \text{ over the distribution } r \leftarrow D \text{ on } R.$$

- ***computationally binding*** *if* $\forall$ *probabilistic polynomial time algorithms* $\mathcal{A}$ *and* $\tilde{\forall} pk \leftarrow KeyGen(1^n)$ *holds*

$$\mathbb{P}\left[\texttt{Verif}(pk, 0, r_0, c) = 1 \text{ and } \texttt{Verif}(pk, 0, r_1, c) = 0 \;\middle|\; (r_0, r_1, c) \leftarrow \mathcal{A}(pk)\right] \leq negl(n)$$

In other words, the statistically hiding property ensures that, even with infinite computational ressources, it is impossible to distinguish a commitment of 0 from a commitment of 1 given only the public key (but not the randomness r). The computationnally binding property ensures that, with limited comutaional ressources, it is impossible to produce a commitment that can be opened to both values 0 and 1: the prover can not change his mind after having provided the commitment.

4.3 Construction of a commitment scheme from SIS

An example of a lattice-based commitment scheme can be obtained by considering SIS-related function $f_{\mathbf{A}} = \mathbf{Ax} \bmod q$. One obtains such a scheme by putting the triple of functions $\texttt{Keygen}, \texttt{Commit}$ and $\texttt{Verif}$ as follows.

The key generating function $\texttt{KeyGen}$ takes as input 1^n and outputs a matrix (that serves as public key) $\mathbf{A} =: pk$ uniformly random from $\mathbb{Z}_q^{n \times m}$, where m is a parameter whose value will be decided later. For the random set R and its distribution D, put $R = \mathbb{Z}^{m-1}$ and $D = D_{\mathbb{Z}^{m-1},\sigma}$ the discrete Gaussian distribution on $\mathbb{Z}^{m-1}$, where $\sigma \in \text{poly}(n)$.

The commitment function is defined as follows: $\texttt{Commit}(pk = \mathbf{A}, \mu, r) := \mathbf{A} \cdot \begin{pmatrix} \mu \\ \mathbf{r} \end{pmatrix} \bmod q$, where $\mu \in \{0,1\}$. Here, $\begin{pmatrix} \mu \\ \mathbf{r} \end{pmatrix}$ is the vector that is obtained by concatenating; $(\mu|\mathbf{r})$.

To verify the commitment on input (pk, μ, r, c), check whether $\mathbf{A} \cdot \begin{pmatrix} \mu \\ \mathbf{r} \end{pmatrix} = c \bmod q$ and $\| \begin{pmatrix} \mu \\ \mathbf{r} \end{pmatrix} \| \leq \beta$. If this is both true, set $\texttt{Verif}(pk, \mu, r, c) = 1$, otherwise 0.

Lemma 7 *For appropriate parameters, above scheme is correct, statistically hiding and computationally binding, assuming that the* $\texttt{SIS}_{n,m,q,2\beta}$ *is hard.*

Proof:

- (Correctness) Choose $\beta \in \text{poly}(n)$ in such a way that $\sigma \ll \beta \ll q$. Then, by construction, $\mathbf{A} \cdot \begin{pmatrix} \mu \\ \mathbf{r} \end{pmatrix} = c \bmod q$, and with overwhelming probability (tail bound of the discrete Gaussian distribution), $\|\mathbf{r}\| \leq \beta - 1$ and therefore $\|(m|\mathbf{r})\| \leq \beta$.

- (Computationally binding) Suppose a probabilistic polynomial time algorithm is able to find (on input $\mathbf{A}$) a triple $(\mathbf{r}_0, \mathbf{r}_1, c)$ such that $\mathbf{A}\begin{pmatrix}0\\ \mathbf{r}_0\end{pmatrix} = c = \mathbf{A}\begin{pmatrix}1\\ \mathbf{r}_1\end{pmatrix} \bmod q$ with non-negligible probability. Then the vector $\mathbf{v} = \begin{pmatrix}1\\ \mathbf{r}_0 - \mathbf{r}_1\end{pmatrix}$ is a $\mathtt{SIS}_{n,m,q,2\beta}$ solution. Therefore, this adversary solves SIS with these parameters, which is a contradiction, as we assumed that this was hard.

- (Statistically hiding) The goal is to prove that $\tilde{\forall}\mathbf{A} = pk \leftarrow \mathtt{KeyGen}(1^n)$ holds $\mathtt{Commit}(pk, 0, r) \approx_s \mathtt{Commit}(pk, 1, r)$, i.e., that the statistical distance is negligible. Decompose A into a first column $\mathbf{a}_0$ and the rest of the matrix $\mathbf{A}'$: $\mathbf{A} = (\mathbf{a}_0|\mathbf{A}')$. Our aim is to prove that

$$\Delta = \frac{1}{2}\sum_{c\in\mathcal{C}} |\mathbb{P}[\mathbf{A}'r = c] - \mathbb{P}[\mathbf{A}'r + a_0 = c]| \leq \mathrm{negl}(n).$$

  Define $\Lambda_q^{\perp c} = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{x} \equiv c \bmod q\}$. Then, by construction

$$\mathbb{P}_{r\leftarrow D_{\mathbb{Z}^{m-1},\sigma}}[\mathbf{A}r = c] = \frac{\rho_\sigma(\Lambda_q^{\perp c}(\mathbf{A}'))}{\rho_\sigma(\mathbb{Z}^{m-1})}$$

  If $\sigma \geq \eta_\varepsilon(\Lambda_q^{\perp c}(\mathbf{A}))$, the smoothing parameter of $\Lambda_q^{\perp c}(\mathbf{A}')$, then we know that (informally) the cumulative weight of the Gaussians of any coset of $\Lambda_q^{\perp c}(\mathbf{A}')$ is the same, up to a factor $(1 \pm \varepsilon)$ [Lecture 8, Lemma 5]. In particular, $\rho_\sigma(\Lambda_q^{\perp(c-a_0)}(\mathbf{A}')) \in [1-\varepsilon, 1+\varepsilon] \cdot \rho_\sigma(\Lambda_q^{\perp c}(\mathbf{A}'))$. Therefore

$$\Delta = \frac{1}{2}\sum_{c\in\mathcal{C}} |\mathbb{P}[\mathbf{A}'r = c] - \mathbb{P}[\mathbf{A}'r + a_0 = c]| = \frac{1}{2\rho_\sigma(\mathbb{Z}^{m-1})}\sum_{c\in\mathcal{C}} |\rho_\sigma(\Lambda_q^{\perp(c-a_0)}(\mathbf{A}')) - \rho_\sigma(\Lambda_q^{\perp c}(\mathbf{A}'))|$$

$$\leq \frac{1}{2\rho_\sigma(\mathbb{Z}^{m-1})}\sum_{c\in\mathcal{C}} \varepsilon \cdot \rho_\sigma(\Lambda_q^{\perp c}(\mathbf{A}')) \leq \varepsilon/2$$

  In order to know the parameter choice for σ, we need to estimate $\eta_\varepsilon(\Lambda_q^{\perp c}(\mathbf{A}))$ with $\varepsilon \in \mathrm{negl}(n)$. This is because σ needs to be larger than the smoothing parameter.

□

Exercise 2 Provide a lower bound for $\eta_\varepsilon(\Lambda_q^{\perp c}(\mathbf{A}))$, using Lemma 2 from this lecture, and [Lecture 7, Theorem 14].