CSE 206A: Lattice Algorithms and Applications Fall 2019

Commitment Schemes

Instructor: *Daniele Micciancio* Scribe: *Yiwen Song*

1 Commitment Schemes

1.1 Definition

A **commitment scheme** is a cryptographic protocol that allows a *Sender* to commit a chosen value (or statement) while keeping it hidden to the *Receiver*, and the *Receiver* has the ability to reveal the committed value later. Usually a commitment scheme could be divided into two phases.

In the commit phase, the *Sender* holds a message m, picks a random key k and encodes (Commit) the message with k and some randomness r. The encoding result c is called a *commitment*, which is sent to the *Receiver* in this phase.

In the reveal phase, the *Sender* sends k to the *Receiver*. The *Receiver* can **Open** the commitment c using k, and then use **Check** to determine whether to **accept** or **reject** that commitment.

procedure Initialize

$m \xleftarrow{\$} \mathcal{M}$

$k \xleftarrow{\$} \mathcal{K}$

procedure Check(k, m, r, c)

$return\ \ (\textbf{Commit}(k, m, r) = c)$

Commit Phase:

procedure Send(m)

$r \xleftarrow{\$} \{0,1\}^s$

$c \leftarrow \textbf{Commit}(k, m, r)$

$return\ \ c$

procedure Receive(c)

$return\ \ c$

Reveal Phase:

procedure Send(k)

$return\ \ k$

procedure Receive(k)

$(m', r') \leftarrow \textbf{Open}(k, c)$

$if\ \ (\textbf{Check}(k, m', r', c) = true)$

$\quad return$ **accept**

$else$

$\quad return$ **reject**

1.2 Security Properties

A good commitment scheme should satisfy the following two security properties.

- **Hiding**: Receiving a commitment c should give the receiver no information about message m. Which means for $\forall\, m_0, m_1$, let

$$P_0 \sim \{(k,c)|k \xleftarrow{\$} \mathcal{K},\ c \xleftarrow{\$} \textbf{Commit}(k, m_0)\}$$

$$P_1 \sim \{(k,c)|k \xleftarrow{\$} \mathcal{K},\ c \xleftarrow{\$} \textbf{Commit}(k, m_1)\}$$

  Then the distribution of P_0 and P_1 should be statistically close over $(\mathcal{K}, \mathcal{C})$.

- **Binding**: Once the key $k \in \mathcal{K}$ is chosen in the first phase, the sender cannot send another key $k' \neq k$ to the receiver in the second phase. Moreover, in a computational view, a commitment scheme **Commit** is said to be (t, ϵ)-*secure*, if $\forall k \in \mathcal{K}$, for all **Open** algorithms $A \in \mathcal{A}$ that run in time t and output $A(k, c) = (m_0, m_1, r_0, r_1)$, we have

$$Pr\left[m_0 \neq m_1, \textbf{Commit}(k, m_0, r_0) = c = \textbf{Commit}(k, m_1, r_1)\right] < \epsilon$$

2 Lattice Revisit

Recall the *Leftover Hash Lemma* that was proven by Impagliazzo, Levin and Ruby in [1].

Lemma 1 *(Leftover Hash Lemma): Let $X \subset \{0,1\}^m$, $|X| \geq 2^l$. Let $e > 0$, and let H be an almost universal family of hash functions mapping m bits to $l-2e$ bits. Then the distribution $(h, h(x))$ is quasi-random within $\frac{1}{2^e}$ (on the set $H \times \{0,1\}^{l-2e}$), where h is chosen uniformly at random from H, and x uniformly from X.*

Proof. Please refer to [2]. ☐

Claim 1 *Let $A \in \mathbb{Z}_q^{n\times m}$, $x \in \{0,1\}^m$. If $m \geq 2n\, lg\, q$, then the distribution (A, Ax) is quasi-random within $\frac{1}{2^e}$, where $e \geq n\, lg\, q - \frac{n}{2}$*

Proof. Using Lemma 1. ☐

Claim 2 *Let $A \in \mathbb{Z}_q^{n\times m}$. If $m = 3n\, lg\, q$, then it's hard to find a short vector $v \in \Lambda_q^{\perp}(A)$ with $||v|| \leq \beta$, where $\beta = \sqrt{3n\, lg\, q}$.*

Proof. Please refer to []. ☐

3 An Example

Definition 1 *$f_A : x \to Ax \bmod q$, where $A \in \mathbb{Z}_q^{n\times m}$, $D(f_A) = \{0,1\}^m \subset \mathbb{Z}_q^m$*

Now let's construct a specific commitment scheme using lattice. Let $\mathcal{K} = \mathbb{Z}_q^{n\times m}$ and $\mathcal{M} = D(f_A)$. Let $k = A \xleftarrow{\$} \mathcal{K}$. Assume that $m = 3n\,lg\,q$, define the **Commit** function as follows:

$$\begin{aligned} &\textbf{Commit}(A, msg \in \{0,1\}^m, r \in \{0,1\}^{2m}) \\ =\ & f_A(msg, r) \\ =\ & A \begin{bmatrix} msg \\ r \end{bmatrix} \ mod\ q \end{aligned}$$

We are going to prove that this commitment scheme satisfies both the hiding property and the binding property.

3.1 Hiding Property

The scheme is said to satisfy the binding property, if, for $\forall m_0, m_1 \in \mathcal{M}$, the distribution ensembles $\{(A,c)|A \xleftarrow{\$} \mathcal{K},\ c \xleftarrow{\$} \textbf{Commit}(A, m_0)\}$ and $\{(A,c)|A \xleftarrow{\$} \mathcal{K},\ c \xleftarrow{\$} \textbf{Commit}(A, m_1)\}$ are computationally indistinguishable.

Since $m = 3n\,lg\,q$, we could write $A = (A_0, A_1, A_2)$, where $A_i \in \mathbb{Z}_q^{n\times n\,lg\,q}\,(i = 0,1,2)$. Therefore, for $\forall msg \in \mathcal{M}, \forall r \in \{0,1\}^{2m}$, we have

$$(A, c) = (A_0, A_1, A_2, A_0 \cdot msg + (A_1, A_2) \cdot r)$$

From Claim 1, we know that $(A_1, A_2) \cdot r$ is close to a uniform distribution over $\mathbb{Z}_q^n$, so $A_0 \cdot msg + (A_1, A_2) \cdot r$ is a shift copy of a uniform distribution.

Since $A = (A_0, A_1, A_2)$ is randomly chosen from $\mathcal{K}$, so for $\forall m_0, m_1 \in \mathcal{M}$, the distribution of $\{(A_0, A_1, A_2, A_0 \cdot m_0 + (A_1, A_2) \cdot r)|A \in \mathbb{Z}_q^{n\times 3n\,lg\,q}, r \in \{0,1\}^{2m}\}$ and $\{(A_0, A_1, A_2, A_0 \cdot m_1 + (A_1, A_2) \cdot r)|A \in \mathbb{Z}_q^{n\times 3n\,lg\,q}, r \in \{0,1\}^{2m}\}$ are computationally indistinguishable, which means, the commitment scheme satisfies the hiding property.

3.2 Binding Property

From the previous discussion, we know that the scheme is said to be (t, ϵ)-*secure*, if for any adversary $\mathcal{A}$ using **Open** algorithm with running time less than t, given the input $A \in \mathbb{Z}_q^{n\times 3n\,lg\,q}$ and $c \in \mathbb{Z}_q^n$, which outputs (m_0, m_1, r_0, r_1). The advantage

$$Adv(\mathcal{A}) = Pr\left[A \begin{bmatrix} m_0 \\ r_0 \end{bmatrix} = A \begin{bmatrix} m_1 \\ r_1 \end{bmatrix}\right] = Pr\left[A \begin{bmatrix} m_0 - m_1 \\ r_0 - r_1 \end{bmatrix} = \mathbf{0}\ mod\ q\right]$$

is less than ϵ.

From Claim 2, we know that since $m = 3n\, lg\, q$, it's hard to find short vectors in $\Lambda_q^{\perp}(A)$, which means if the sender encodes m_0 in the commit phase, it's hard for the adversary to find another message m_1 that is close to m_0 and could also pass the **Check** procedure. Therefore, this commitment scheme is computationally binding.

References

[1] Russell Impagliazzo, Leonid A Levin, and Michael Luby. "Pseudo-random generation from one-way functions". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. ACM. 1989, pp. 12–24.

[2] Russell Impagliazzo and David Zuckerman. "How to recycle random bits". In: *30th Annual Symposium on Foundations of Computer Science*. IEEE. 1989, pp. 248–253.