# Recitation 8: Commitment Schemes and Secret Sharing Practice

## 1 Commitment Schemes

**Definition 1 (Commitment scheme)** *A commitment scheme between two parties is a pair of algorithms* $(\mathsf{Gen}, \mathsf{Com})$ *where* $\mathsf{Gen}(1^\lambda)$ *produces public parameters* $\mathsf{pp} \in \{0,1\}^{poly\lambda}$. $\mathsf{Com}(\mathsf{pp}, m, r)$ *for a message* $m \in \mathcal{M}$ *and randomness* $r \xleftarrow{R} \{0,1\}^\lambda$ *follows the two properties*

- *(Hiding) For all* $m_0, m_1 \in \mathcal{M}$

$$(\mathsf{pp}, \mathsf{Com}(\mathsf{pp}, m_0, r_0) \approx (\mathsf{pp}, \mathsf{Com}(\mathsf{pp}, m_1, r_1))$$

  *where* $r_0, r_1 \xleftarrow{R} \{0,1\}^\lambda$.

- *(Binding) For an adversary* $A$*,*

$$\Pr_{\mathsf{pp}}[A(\mathsf{pp}) = (m_0, m_1, r_0, r_1) m_0 \neq m_1 \wedge \mathsf{Com}(\mathsf{pp}, m_0, r_0) = \mathsf{Com}(\mathsf{pp}, m_1, r_1)] = \mathit{negl}(\lambda)$$

Both properties may be computational or statistical. Computational hiding means the two distributions are computationally indistinguishable, whereas statistical hiding means they are statistically indistinguishable. For computational binding, the adversaries are PPT, whereas statistical binding means the property holds for all-powerful adversaries. In other words, over the distribution of public parameters, the probability of an all-powerful adversary (which can compute exponential time algorithms) finding $m_0, r_0, m_1, r_1$ that collide is negligible. We'll see more of this in the examples.

### 1.1 Practice Problems

For the following commitment schemes, determine whether the scheme is (1) computationally binding, statistically binding, or neither and (2) computationally hiding, statistically hiding, or neither.

#### A CRHF-based scheme.

Recall the definition collision-resistant hash functions. To quote pset 1,
a family of functions $\{f_\lambda\}_{\lambda \in \mathbb{N}}$ is said to be **collision resistant** if it is polynomial-time computable, for every $\lambda \in \mathbb{N}$, $f_\lambda : \{0,1\}^* \to \{0,1\}^\lambda$, and for all polynomial-time adversaries $A$

there exists a negligible function $\mu$ such that for every $\lambda \in \mathbb{N}$,

$$\Pr\left[f_\lambda(x) = f_\lambda(x') \wedge x \neq x' \,:\, (x, x') \leftarrow A(1^\lambda)\right] \leq \mu(\lambda).$$

In other words, it is difficult to find distinct $x, x'$ such that $f_\lambda(x) = f_\lambda(x')$.

Let $f_\lambda : \{0,1\}^* \to \{0,1\}^\lambda$ be a CRHF family We define the commitment scheme to $m \in \{0,1\}^*$ as follows:

- $\mathsf{Gen}(1^\lambda) : f_\lambda \xleftarrow{R} \{f_\lambda\}_{\lambda \in \mathbb{N}}$
- $\mathsf{Com}(\mathsf{pp}, m, r) = f_\lambda(m)$.

**A PRG-based scheme.**

A **pseudorandom generator** (PRG) is an expanding function $G : \{0,1\}^n \to \{0,1\}^m$, $m = \text{poly}(n)$, which takes in a random seed $s \in \{0,1\}^n$ and outputs $G(s) \approx_c U_m$. Similar to the PRFs we learned about in lecture, outputs of $G$ are indistinguishable from random, but PRGs are not keyed.

Let $G : \{0,1\}^n \to \{0,1\}^{3n}$ be a PRG. We define the commitment scheme to a *1-bit message* $b$ as follows:

- $\mathsf{Gen}(1^n) : v \xleftarrow{R} \{0,1\}^{3n}$
- $\mathsf{Com}(v, b, r) = G(r) \oplus b \cdot v$.

**Answers.**

1. Computationally binding (follows from collision resistance), but not hiding (hash functions may reveal information about inputs).
2. Statistically binding (counting argument: $2^n$ inputs, $2^{3n}$ outputs), computationally hiding (follows from pseudorandomness).

# 2 Secret Sharing

In lecture this week, we learned about Shamir secret sharing. Recall the definition of $t$-out-of-$n$ secret sharing, and then we will review Shamir secret sharing by walking through an example on the board.

**Definition 2 ($t$-out-of-$n$ secret sharing)** *A $t$-out-of-$n$ secret sharing scheme scheme over message space $\mathcal{M}$ consists of a pair of efficient algorithms* $(\mathsf{Share}, \mathsf{Reconstruct})$ *such that:*

- $\mathsf{Share}$ *is a randomized algorithm that takes as input a message $m \in \mathcal{M}$ and outputs a n-tuple of shares $(s_1, \ldots, s_n)$.*

- Reconstruct *is a deterministic algorithm that given a t-tuple of shares outputs a message* $m \in \mathcal{M}$.

*The following two properties are required to be satisfied.*

- ***Correctness**. For every* $m \in \mathcal{M}$ *and every* $I = \{i_1, \dots i_t\} \subseteq [n]$ *of size* $t$*,*

$$\Pr_{(s_1,\dots,s_n)\leftarrow \mathsf{Share}(m)}[\mathsf{Reconstruct}(s_{i_1}, \dots, s_{i_t}) = m] = 1$$

- ***Security**. For every* $m, m' \in \mathcal{M}$ *and for every* $I \subseteq [n]$ *such that* $|I| < t$*,*

$$(s_i)_{i\in I} \equiv (s'_i)_{i'\in I}$$

*where* $(s_i)_{i\in[n]} \leftarrow \mathsf{Share}(m)$ *and* $(s'_i)_{i\in[n]} \leftarrow \mathsf{Share}(m')$

**Lagrange interpolation practice.**

Suppose we have a $3$-out-of-$11$ Shamir secret sharing scheme for a secret $\mu$ with $p = 31$ modulus. There are 11 total shares.

- Suppose we only have two shares $(1, 6)$, $(4, 21)$. Explain why the information we now have is independent of $\mu$.
- We now receive a third share $(12, 5)$. What is the secret?

*Definitions are adapted from 6.5610 2024 lecture notes.*