6.876/18.426J Advanced Topics in Cryptography April 16, 2003

Lecture 17: Mutually Independent Commitments

Scribe: Shien Jin Ong

Guest lecturer: TA

1 Introduction, Commitment Schemes

This lecture is based on the paper, "Mutually Independent Commitments" by Moses Liskov, Anna Lysyanskaya, Silvio Micali, Leonid Rezyin, and Adam Smith, which appeared in ASIACRYPT 2001.

There are two main kinds of commitment schemes.

1. Computationally binding and perfectly hiding commitments.

2. Perfectly binding and computationally hiding commitments.

Computationally binding and perfectly hiding commitments can be constructed from GM-secure public-key encryptions. The main idea is to send an encryption of the message together with the public key. To decommit, we reveal the random bits used to encrypt the message. In fact, one-way functions suffice to construct computationally binding and perfectly hiding commitments.

Perfectly binding and computationally hiding commitments can be constructed from claw-free permutations. We present a scheme due to Pederson which relies on the hardness of DLP (the hardness of DLP implies the existence of claw-free permutations).

Let S be the sender and R be the receiver.

Input: Security parameter, k.

1. R: Sends a random k-bit prime p, a random generator $g \leftarrow \mathbb{Z}_p^*$, a proof that g is a generator (e.g. factorization of $p-1$), and a random $h \leftarrow \mathbb{Z}_p^*$

2. S: To commit $x \in \{0,1\}$, choose a random $r \leftarrow \mathbb{Z}_p$ and send $C(x) = g^r h^x \pmod{p}$.

3. S: To decommit, send r.

This scheme is perfectly hiding because g generates the whole group $\mathbb{Z}_p^*$. The computational binding property comes from the assumption that finding an α such that $g^\alpha \equiv h \pmod{p}$ is hard for most random choices of h.

2 Independence in Commitments

Imagine an auction bidding scenario where A commits to a price p and then B commits to another price. Although A's commitment to p does not reveal anything, a commitment scheme like Pederson's would allow B to commit to $p+1$ just by seeing $C(p)$. We would like to prevent this from happening, thus we need to construct commitments scheme that are independent. There are three model of independence that we will discuss.

2.1 Mutually Independent Announcement

In a mutually independent announcement, we want A to commit to a message first and B to commit second. Our goal is to ensure that A and B's commitments are not correlated, i.e. independent. We omit the formal definition, which is in the paper by Liskov et. al.

We describe the scheme. Let C be a perfectly binding and computationally hiding commitment scheme.

1. A chooses a random $a \leftarrow \{0,1\}^k$ and commits to $C(a)$.
2. B chooses a random $b \leftarrow \{0,1\}^k$ and commits to $C(b)$.
3. B decommits b.
4. A decommits a.

The binding and hiding property of A and B is preserved. We focus on the non-correlation property.

Even if A is dishonest, the value of a cannot be correlated to b since A commits first. Even if B is dishonest, b cannot be correlated to a since that would mean that we could find distinguish the commitment $C(a)$ from commitments to other values. We stress that B needs to decommit first since otherwise B could just copy A's commitment and let $C(b) = C(a)$ and decommits by copying A's decommitment.

2.2 Mutually Independent Commitment

There are certain scenarios where we want A to be the first party to commit and also be the first party to decommit. As an extension to the mutually independent announcement, we can define mutually independent commitment to capture this property. Again, we omit the formal definition, which is in the paper by Liskov et. al.

Two-round protocol. We present a simple two-round protocol based on the assumption that subexponentially hard one-way permutations exist. Under this assumption, we can construct subexponentially hard noninteractive commitment schemes. By two-round, we mean that the commitment stage takes 2 rounds.

What do we mean by subexponentially hard? This means that there exist an $\varepsilon > 0$ such that any adversary running in time less than 2^{n^ε} cannot break the commitment (or cannot invert the one-way permutation).

Let C be a 2^{n^ε}-hard, noninteractive, perfectly binding commitment scheme. By brute force, we can break the commitment scheme C in time 2^{n^d} for some $d > \varepsilon$. We construct a mutually independent commitment scheme as follows.

Input: Security parameter, k.

1. Let $K = k^{2d/\varepsilon}$. A choose a random $a \leftarrow \{0,1\}^k$ and commits to $C(a;K)$ (commitment to a with security parameter K).
2. B chooses to a random $b \leftarrow \{0,1\}^k$ and commits to $C(b;k)$ (commitment to a with security parameter k).

Either A or B is allowed to decommit first.

We show that B cannot correlate b to the value of a. First, observe that we can obtain b from $C(b;k)$ in $2^{k^d} = 2^{K^{\varepsilon/2}}$ time. Then if b is correlated to a, we can break the commitment $C(a;K)$ in $O(2^{K^{\varepsilon/2}})$ time, contradicting the assumption that C is a 2^{n^ε}-hard noninteractive commitment scheme.

Three-round protocol. Notice that our assumption is strong since we need a subexponentially hard one-way permutation. We can construct a three-round mutually independent commitment scheme using the assumption that dense cryptosystems exist. By three-round, we mean that the commitment stage takes 3 rounds. Dense cryptosystems are such that the public key can be chosen at random from the uniform distribution on $\{0,1\}^n$ and still be secure with high probability. *Exercise:* Prove that dense cryptosystem exist under the RSA hardness assumption.

Let $p(k)$ be the length of public key of the cryptosystem with security parameter k. Let C be a perfectly binding, non-interactive commitment scheme.

Input: Security parameter, k.

1. A chooses a random $r_A \leftarrow \{0,1\}^{p(k)}$ and sends $C(r_A)$.

2. B chooses to a random $b \leftarrow \{0,1\}^k$ and commits to $C(b)$. In addition, B chooses a random $r_B \leftarrow \{0,1\}^{p(k)}$ and sends r_B.

3. A computes $PK = r_A \oplus r_B$ and sends $E(PK; a)$ and PK. Note that A does not open the commitment to r_A at this step.

The reveal stage is where A opens the encryption of a, decommits to r_A, and where B decommits to b, checks if $r_A \oplus r_B = PK$.

Clearly b cannot be correlated to a. Why cannot a be correlated to b? The proof details are in the paper, but roughly speaking, the ability to correlate a to b would allow one to break the semantic security of the commitment to b.