0368.4162: Introduction to Cryptography Ran Canetti

Lecture 11

12th January 2009 Fall 2008

Scribes: D. Widder, E. Widder

Today's lecture topics

- Introduction to cryptographic protocols
- Commitments

1 Cryptographic Protocols

1.1 Motivation

Imagine two millionaires arguing who is richer but no one is willing to let the other one know how much money he has.
Let M_A denote the amount of money owned by millionaire A and M_B the amount of money owned by millionaire B.
We need to design a protocol that will enable them to calculate the function: $M_A > M_B$ such that A doesn't get further information about M_B and B doesn't get further information about M_A besides the one bit function output. This problem was suggested in [Yao '82].
How can we use the primitives that we have so far to design such a protocol?
Up until now, we designed primitives that enable security against "outsiders".
Now we want two participants to collaborate in order to compute a certain function, even though they do not trust *each other*.
Generally , we wish to be able to perform a joint calculation of many participents who don't trust each other.

Some concrete problems:

- Electronic voting
- Auctions
- Online shopping
- Gambling games
- Contract signing
- Secured database lookup
- Secured remote computations

When dealing with this set of problems we will have to:

1. Understand and formalize the security requirements of such tasks.
2. Design protocols that realize these tasks.
3. Find ways to rigorously assert or analyze the security properties of these protocols against the security requirements.

2 Commitment Schemes

The first problem we will discuss is Commitment.

2.1 Motivation

Assume Alice claims she can predict next week's lottery winning numbers.
Alice wants to convince Bob that she really knows but she doesn't want Bob to use this information.
In a physical world, Alice can write the numbers on a piece of paper, put it in a safety box, lock it and give it to Bob for safekeeping.
On the next week, Alice will give Bob the key so Bob can verify that Alice was right.
The two main properties here:

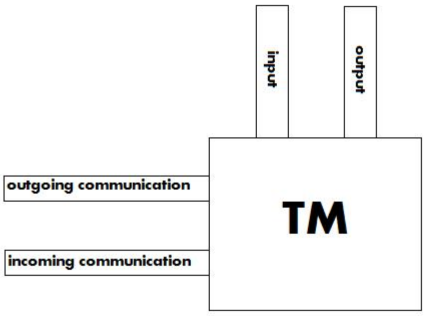
- Secrecy : Until Alice gives the key to Bob, she is certain that Bob does not know her prediction.
- Binding : As soon as Bob has the box, he knows that Alice's prediction is fixed and cannot be changed.

A commitment scheme is a protocol aimed at giving the same type of guarantees in a digital way. More specifically, a commitment scheme is a protocol for two participants: Sender and Receiver. This protocol consists of two phases:

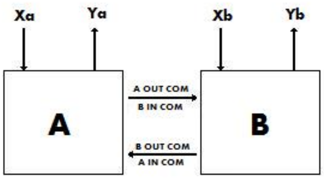
1. Commitment phase - The sender commits to a value.
2. Reveal phase - The value is revealed and checked.

2.2 Reminder - Formalism of two party protocols

We can think of an interactive algorithm as a Turing Machine with two extra tapes (These TM's are called ITM - Interactive Turing Machine).



For a protocol $P = (A, B)$, let $[A, B](x_A, x_B) = (y_A, y_B)$ denote the output of protocol P on inputs x_A for A, x_B for B. Here y_A is the output of A and y_B is the output of B.



For reactive protocols in which the parties iteratively get inputs and generate outputs, we define:

$$[A, B](x_A^1, x_B^1; x_A^2, x_B^2; ...) = (y_A^1, y_B^1; y_A^2, y_B^2; ...)$$

Where y_A^i, y_B^i are the outputs of $[A, B]$ in round i when given x_A^i, x_B^i as inputs.

2.3 Definitions

A commitment is formalized as a reactive two party protocol (C, R).
Let M denote the message domain and 1^n is the security parameter.
ϵ denotes the empty string.
The protocol has two iterations:

1. The commitment phase :
   Input:
   C gets $m \in M$, 1^n
   R gets 1^n
   Output:
   C outputs ϵ
   R outputs $"committed"$

2. The reveal phase :
   Input:
   C gets $"open"$
   R gets ϵ
   Output:
   C outputs ϵ
   R outputs m' or $\perp$

As usual , there are two approaches for security definition:
By defining a game (IND) or by an ideal model (SEM). We will present both this approaches for defining commitment scheme security:

Definition 1. *IND secured commitment scheme (C, R) is IND-secure if:*

1. **Completeness:** For all $m \in M$:

$$Prob\big[[C, R]\big((m, 1^n), 1^n; "open", \epsilon\big) = \big(\epsilon, "committed"; \epsilon, m\big)\big] = 1$$

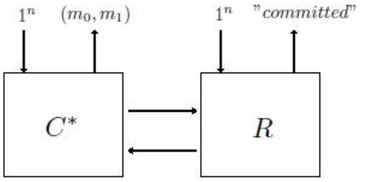
2. **Secrecy:** For all $m_0, m_1 \in M$ $(m_0 \neq m_1, |m_0| = |m_1| = n)$ and for all polynomial time R^*:

$$Prob\big[b \leftarrow \{0, 1\}; [C, R^*]\big((m_b, 1^n), (m_0, m_1))\big) = (\epsilon, b)\big] < \tfrac{1}{2} + \nu(n)$$

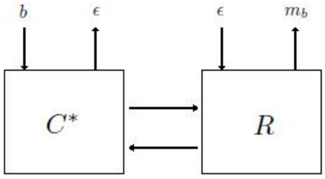
3. **Binding:** For all polynomial time C^*:

$$Prob\big[b \leftarrow \{0,1\}; m_0 \neq m_1; [C^*, R]\big(1^n, 1^n; b, \epsilon\big) = \big((m_0, m_1), "committed"; \epsilon, m_b\big)\big] < \frac{1}{2} + \nu(n)$$

In the binding requirement, we capture the requirement that C^* cannot "open" a commitment to more than one value as follows: C^* announces two possible openings of the commitment, and later it is asked to make R open a random one of the two and can win w.p. at most $\frac{1}{2} + \nu(n)$.



Commitment phase



Reveal phase

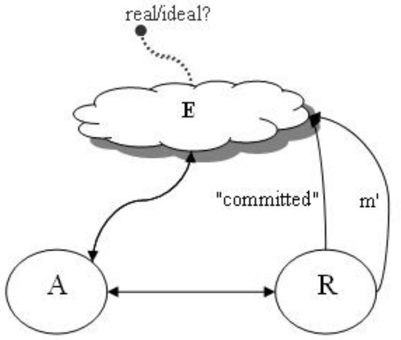
Note that the secrecy requirement is for some efficient R^* not only for R that follows the protocol and the binding requirement is for some efficient C^* not only for C that follows the protocol.
When the requirement for secrecy (binding) holds even for computationally unbounded R^* (C^*), we say that the protocol has statistical secrecy (binding).

Definition 2. *SEM secured commitment scheme*

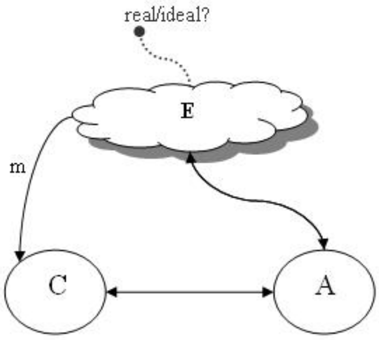
This definition is called in the literature Universally Composable Commitment. We will learn more about the theory behind this notion in the next semester.

- **Formalization of a real commitment system** $Real_{E,A,(C,R)}$

  The adversary A decides whether it controls C or R

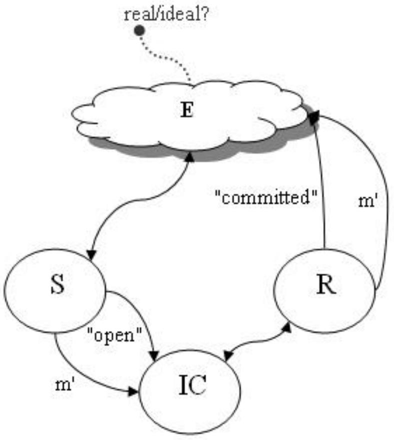
  If A controls C, it communicates with R in place of C while talking freely with E. At the end of this sequence R potentially outputs $"committed"$.
  A then communicates with R in place of C. At the end of this sequence R outputs a message m'.



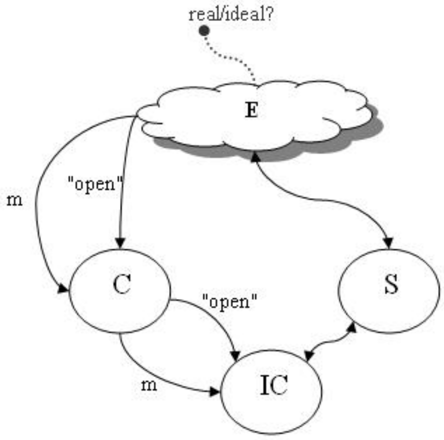
If A controls R, C gets m from the environment and then it communicates with A in place of R. At the same time A interacts freely with E.
When it chooses, E generates some output. (This output can be thought of as a prediction whether E is in the real or ideal interaction).
Let $Real_{E,A,(C,R)}$ be the ensemble which describes the output of E in a real interaction.

- **Formalization of an ideal commitment system** $Ideal_{E,S,IC}$

The simulator S decides whether it controls C or R at the beginning.
S and the party not controlled by it are interacting with an ideal commitment service, denoted as IC.



When S simulates an adversary controlling C, it generates and sends m' to IC. IC in turn communicates with R and at the end of the sequence R outputs "$committed$" to the environment. Then S sends the "$open$" message to IC. At the end of the sequence R outputs m'. Throughout , S interacts with E freely.



When S simulates an adversary controlling R, C gets a message m from the environment and sends m to IC. IC in turn gives "committed" to S. When E gives input "open" to the committer, C, C forwards "open" to IC. In turn , IC outputs m to S.

Note that the ideal commitment provides perfect secrecy and binding:

1. As soon as R received the output "committed" , it is guaranteed that there's only one value, m, that will be opened.
2. As long as C haven't started the reveal phase S doesn't learn anything about m

Let $Ideal_{E,S,IC}$ be the ensemble which describes the output of E in an ideal interaction.

A commitment scheme (C, R) is SEM-secured if for every polynomial time bounded adversary A there exists a polynomial time bounded adversary S such that for every polynomial time bounded environment E:

$$Real_{E,A,(C,R)} \approx Ideal_{E,S,IC}$$

It turns out that SEM security is not equivalent to IND security. We have:

- $SEM \Rightarrow IND$.
- SEM is stronger and has many nice properties that IND doesn't have (such as composability).
- SEM is not realizable without adding some assumptions to the model.
- In fact, there are several notions of security in the literature between SEM and IND where each notion provides a different set of properties and guarantees.

2.4 Constructions

We will see three commitment schemes , and prove that they are IND-secure. We remark that none of these schemes is SEM-secure. We will see SEM-secure schemes in the next semester.

2.4.1 OWP Based Commitment Scheme

Let f be a one-way permutation (OWP) with hard core predicate B.
A commitment scheme for one bit $m \in \{0, 1\}$:

- Commitment phase:

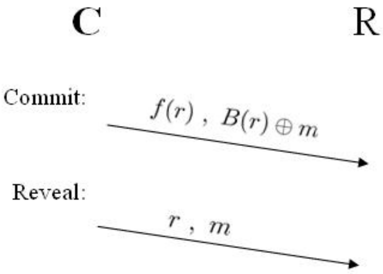
  C : Randomly selects $r \in \{0, 1\}^n$, and sends $y = f(r), b = B(r) \oplus m$

  R : Outputs "$committed$"

- Reveal phase:

  C : Sends r, m

  R : Verifies $f(r) = y$ and $B(r) \oplus b = m$. If verification succeeded output m, otherwise output $\perp$



IND **Security of the scheme**

- Completeness : follows immediately.
- Secrecy : A polynomial adversary who can guess $m \in \{0, 1\}$ with non negligible probability can by the same guess compute $B(x)$ contradicting B being a hard core predicate.
- Binding : f is a permutation , Thus x, m are uniquely determined by y, b. So binding holds even for unbounded C^*

2.4.2 Discrete-Log Based Commitment Scheme [Pedersen '91]

Let G be a group of prime order $p \approx 2^n$.
A commitment scheme for $m \in [1...p]$:

- Commitment phase:

  R : Randomly selects $g, h \in G$ generators of G , and sends g, h

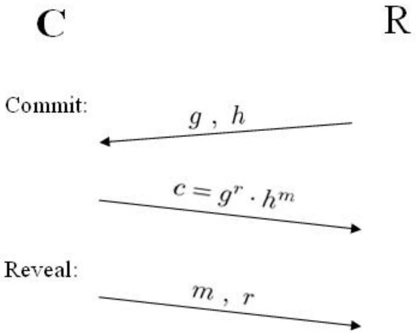
  C : Randomly selects $r \in [1...p]$, and sends $c = g^r \cdot h^m$

  R : Outputs "$committed$"

- Reveal phase:

  C : Sends m, r

  R : Verifies $g^r \cdot h^m = c$. If verification succeeded output m, otherwise output $\perp$



IND **Security of the scheme**

- Completeness : follows immediately.
- Secrecy : r is randomly chosen so $c = g^r \cdot h^m$ is a random element of G , independently of m. Thus we get perfect secrecy (even against unbounded R^*).
- Binding : Assume the existence of C^* that wins the game in the binding requirement of IND-security definition. w.p. $\geq \frac{1}{2} + \epsilon$.
  Construct an efficient algorithm A that computes DL in G w.p. $\geq 2 \cdot \epsilon$.
  A receives g, h as input and needs to compute x such that $g^x = h(mod\, p)$
  A interacts with C^* as the receiver in the above game:
  1. A sends g, h to C^*
  2. A receives m_0, m_1, c
  3. A sends $b = 0$ to C^*
  4. w.p. $\geq \frac{1}{2} + \epsilon$, A receives r_0 such that $c = g^{r_0} \cdot h^{m_0}$
  5. A "rewinds" C^* till right after step 2 and sends $b = 1$ to C^*
  6. w.p. $\geq \frac{1}{2} + \epsilon$, A receives r_1 such that $c = g^{r_1} \cdot h^{m_1}$
  7. A computes and outputs $\frac{r_0 - r_1}{m_1 - m_0}$

  $c = g^{r_0} \cdot h^{m_0} = g^{r_1} \cdot h^{m_1}$ holds w.p. $\geq 1 - (\frac{1}{2} - \epsilon) - (\frac{1}{2} - \epsilon) = 2 \cdot \epsilon$, and then $h = g^{\frac{r_0 - r_1}{m_1 - m_0}}$

Note the new technique here: A uses the fact that it has the code of C^* to "rewind" it and re-run it from a previous state. This technique will be very useful in the future.

2.4.3 Decisional Diffie-Helman Based Commitment Scheme

Let G be a group of prime order $p \approx 2^n$ and let g be a generator of G.
A commitment scheme for $m \in [1...p]$:

- Commitment phase:

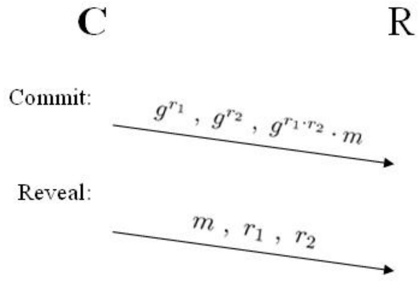
  C : Randomly selects $r_1, r_2 \in [1..p]$, and sends $a = g^{r_1}, b = g^{r_2}, c = g^{r_1 \cdot r_2} \cdot m$

  R : Outputs "$committed$" and records a, b, c

- Reveal phase:

  C : Sends m, r_1, r_2

  R : Verifies $a = g^{r_1}, b = g^{r_2}, c = g^{r_1 \cdot r_2} \cdot m$. If verification succeeded output m, otherwise output $\perp$



IND **Security of the scheme**

- Completeness follows immediately.
- Secrecy:
  Assume towards contradiction that there exists an efficient R^* that wins the game defined in the secrecy property of IND with probability $\geq \frac{1}{2} + \epsilon$.
  Construct an efficient algorithm D that distinguishes between the following two distributions with non negligible probability: $S_1 = (g, g^r, h, h^r)$ and $S_2 = (g, g^r, h, h^s)$ where r, s are randomly chosen from G.
  Algorithm D:

  1. D receives a sample from the sampling oracle : (g, g^r, h, h^s)
  2. D randomly selects $b \leftarrow \{0, 1\}$ and generates $m_0 \neq m_1 \in \{0, 1\}^n$
  3. D interacts with R^* taking C's role and sends $a = g^r, b = h, c = h^s \cdot m_b, m_0, m_1$
  4. R^* outputs a bit b'
  5. D outputs 1 iff $b = b'$

  $$(1) Prob[D(S_1) = 1] = Prob[R^* \; wins \; the \; game \; against \; C] \geq \tfrac{1}{2} + \epsilon$$

  (1) follows because $a = g^r$, $b = h := g^{r_1}$ and $c = h^r \cdot m_b = g^{r_1 \cdot r} \cdot m_b$ so D acts like C defined in the protocol.

  $$(2) Prob[D(S_2) = 1] = Prob[R^* \; wins \; the \; game \; against \; random \; commitments] = \tfrac{1}{2}.$$

  (2) follows because when sampling from S_2: $c = h^s \cdot m_b = g^{r_1 \cdot s} \cdot m_b$ the view of R^* is independent of b.

  $$|(1) - (2)| \geq \tfrac{1}{2} + \epsilon - \tfrac{1}{2} = \epsilon$$

  This contradicts the DDH assumption.
- **Binding:**
  R receives $a = g^{r_1}, b = g^{r_2}$ and $c = g^{r_1 \cdot r_2} \cdot m$.
  Because g is a generator a, b uniquely determines r_1, r_2 respectively and that uniquely determines m from c.
  We get unconditional binding, even against unbounded C^*.

2.5 Composing commitment schemes in cryptographic protocols

Consider an example for an auction protocol based on an IND-secured commitment scheme:

The auction protocol based on a commitment scheme:

There are n players who want to give an offer for a digital camera.

- Each player commits to an offer.
- Each player opens his commitment to the auction center.
- Auction center verifies the commitments and the highest offer wins the auction.

Suppose that we use the commitment scheme based on DDH as the underlying commitment.
As shown, this commitment scheme is IND-secured, however it's not enough for composing it in the auction protocol:
A malicious player can eavesdrop the communication of another player, so he sees at the commitment phase: (a, b, c).
The player can then compute $2 \cdot c = g^{r_1 \cdot r_2} \cdot 2 \cdot m$ and commit to the auction center.
On the reveal phase it eavesdrops again and fetches r_1, r_2, m.
The player finally reveals his commitment by sending $r_1, r_2, 2 \cdot m$, and gets the camera.

In other words , the above scheme is *malleable*. (Recall that we had a similar issue in the context of encryption schemes).
This is one example for the fact that IND-security is not always composable in cryptographic protocols. We note that the above definition of semantic security of commitment does guarantee composability with any other protocol. For this reason it is often called Universally Composable commitment. We'll learn more about this notion in the next semester.