CSC 591, Homework 4

Fatema Olia - 200253671
folia@ncsu.edu

Dec 2, 2018

Commitment Schemes

1. **Definition of Commitment Scheme:**[1]
   A polynomial-time machine Com is called a commitment scheme it there exists some polynomial $l(\,.\,)$ such that the following two properties hold:
   1. Binding: For all $n \in N$ and all $v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^{l(n)}$ it holds that $Com(v_0, r_0) \neq Com(v_1, r_1)$.
   2. Hiding: For every n.u. p.p.t. distinguisher D, there exists a negligible function ϵ such that for every $n \in N$ and $v_0, v_1 \in \{0,1\}^n$, D distinguishes the following distributions with probability at most $\epsilon(n)$:
      - $\{r \leftarrow \{0,1\}^{l(n)} : Com(v_0, r)\}$
      - $\{r \leftarrow \{0,1\}^{l(n)} : Com(v_1, r)\}$

2. For a scheme to be statistically hiding, the commitment of two different messages are the same, i.e. $Com(m_1) = Com(m_2)$
   For a scheme to be statistically binding, there are no two messages for which the commitments are the same, i.e. $Com(m_1) \neq Com(m_2)$.
   Thus, since both statistically hiding and statistically binding are contradictions, a commitment scheme cannot be both statistically hiding and statistically binding.

3.

1. **Theorem:**

If the DDH assumption holds in $\mathbb{G}$ then this scheme is hiding.

Assumption:

1. Towards a contradiction assume that there is a PPT adversary A_{hiding} that is able to distinguish the commitments of m_0 and m_1 and wins the hiding game with probability $\frac{1}{2} + p(n)$
2. There exists a PPT adversary A_{ddh} that has access to an oracle that returns the tuple (g, g_1, g_2, g_3) where $g_1 = g^a$, $g_2 = g^b$ and $g_3 = g^c$ where $c = ab$ or $c = z$.

Observation:

Consider another commitment scheme Π' similar to the El Gammal commitment scheme where the adversary has zero advantage, i.e. the adversary wins with probability $\frac{1}{2}$

$\underline{\Pi'(\mathbb{G}, q, g, h)}$:

The commuter picks a random $u \leftarrow \mathbb{Z}_q$ and a random $z \leftarrow \mathbb{Z}_q$ and $Com(m, u) = (g^u, g^m g^z)$

Reduction:

The adversary A_{ddh} queries the oracle to receive the tuple (g, g_1, g_2, g_3). Then it runs A_{hiding} in order to try and win the DDH game.

$\underline{A_{ddh}(g, g_1, g_2, g_3)}$:

1. Activate A_{hiding} and put $h = g_2$. Make $(\mathbb{G}, q, g, h)$ public.
2. Accept messages m_0, m_1 from A_{hiding}.
3. put $g^u = g_1$
4. Since $h = g_2$, put $h^u = g_3$
5. pick a bit b, calculate g^{m_b}
6. return $(g_1, g^{m_b} g_3)$
7. If A_{hiding} outputs $b* = b$ then output 1,
   else output 0.

Analysis:

A_{ddh} receives the tuple (g, g_1, g_2, g_3) from the oracle.

Here $g_1 = g^a$, so if we consider $u = a$ then $g^u = g_1$.

Also since $h \leftarrow \mathbb{G}$, we can put $h = g_2$. It means $h = g^b$ where b is some value in $\mathbb{G}$.

Thus, when we calculate h^u, we are actually calculating g^{ab} and so we can put $h^u = g_3$.

<u>Case 1:</u> If $g_3 = g^{ab}$ then this is exactly El Gammal commitment scheme.

Thus,

$$Pr[A_{ddh}\text{wins DDH game}] = Pr[A_{hiding}\text{ wins hiding game}] = \frac{1}{2} + p(n)$$

<u>Case 2:</u> If $g_3 = g^z$ then this is exactly the scheme Π'.

Thus,

$$Pr[A_{ddh}\text{wins DDH game}] = Pr[A_{hiding}\text{ wins hiding game}] = \frac{1}{2}$$

However, since the DDH assumption is true, A_{ddh} cannot win the DDH game with a non negligible probability. Thus, our initial assumption must be false and so $Pr[A_{hiding}\text{wins hiding game}] = \frac{1}{2} + negl(n)$

2. Since $h \leftarrow \mathbb{G}$, there is some value x for which $g^x = h$.
   If we know the discrete log of h, it would mean we can find the value of x, i.e. $x = log_g h$.
   During the commitment, the prover sends $(g^u, g^{m_b}h^u)$.
   We can then calculate $(g^u)^x$ which is h^u (Since $h^u = (g^x)^u$).
   Thus, we can get the value of g^{m_b}. Now we can calculate g^{m_0} and g^{m_1} and figure out whether the prover has committed message m_0 or m_1.
   Thus, $Pr[A_{hiding}$ wins hiding game$] = 1$ and the scheme does not remain computationally hiding.

Zero Knowledge Proofs

The protocol is as follows:

1. P chooses $r \leftarrow_{\$} \mathbb{Z}_n^*$ and sends $\alpha \leftarrow r^e$ to$V$
2. V chooses $\beta \leftarrow_{\$} \{0,1\}$ and sends it to P
3. P computes $\gamma \leftarrow rx^{\beta}$ and sends it to$V$
4. V accepts the proof if $\gamma^e = \alpha y^{\beta}$

Completeness:

The equation that the verifier V checks is $\gamma^e = \alpha y^{\beta}$. The scheme is functional if this equation is valid.

We know $\gamma = rx^{\beta}$

Thus, $\gamma^e = r^e(x^e)^{\beta}$

Since $\alpha = r^e$ and $x^e = y \bmod n$, $\gamma^e = \alpha(y)^{\beta}$

Thus the equations valid and the scheme has the property of completeness.

Soundness:

1. Since the prover $P*$ is an interactive state machine we can generate multiple transcripts by running the state machine as follows:
   - Activate $P*$
   - Receive α (where $\alpha \leftarrow r^e$)
   - Input β (where $\beta \leftarrow_{\$} \{0,1\}$)
   - Receive γ (where $\gamma \leftarrow rx^{\beta}$)
     Referring to the proof of completeness above, we know that this is an accepting transcript, i.e. $\gamma^e = \alpha y^{\beta}$

     Now, if we rewind $P*$, since it is an interactive state machine, it is initialised with the same r. Also, since it is running with the same initial state, we shall generate a transcript for the same secret x.
   - Rewind $P*$
   - Receive α (where $\alpha \leftarrow r^e$)
   - Input β' (where $\beta' \leftarrow_{\$} \{0,1\}$ and $\beta' \neq \beta$)

- Receive γ' (where $\gamma' \leftarrow rx^{\beta'}$)
  Referring to the proof of completeness above, we know that this is an accepting transcript, i.e. $\gamma'^{e} = \alpha y^{\beta'}$

  Since $\beta' \neq \beta$ we get $\gamma' \neq \gamma$. Thus, we can interact with P^* to obtain two transcripts from a prover that has the same first message. In this case x is the initial message which remains constant

2. The two accepting transcripts are (α, β, γ) and $(\alpha, \beta', \gamma')$.
   We know $\gamma = rx^{\beta}$ and $\gamma' = rx^{\beta'}$
   Since we, as the verifier, input β, we know the values of β and β'. Assume we have sent $\beta = 0$ and $\beta' = 1$.
   Then, $\gamma = r$ and $\gamma' = rx$.
   Thus we can find the secret x by calculating $\dfrac{\gamma'}{\gamma}$.
3. The transcript in this protocol is as follows:
   $P \xrightarrow{\alpha} V$
   $P \xleftarrow{\beta} V$
   $P \xrightarrow{\gamma} V$
   Thus the transcript is (α, β, γ)

4. The simulator is as follows:
   <u>$Sim(n, e, y)$</u>:
   1. choose $\gamma \leftarrow_{\$} \mathbb{Z}_n^*$
   2. choose $\beta \leftarrow_{\$} \{0,1\}$
   3. set $\alpha = \dfrac{\gamma^e}{y^{\beta}}$

   Thus, to check completeness, $\gamma^e = \alpha y^{\beta}$
   Substituting α from our simulation: $\gamma^e = \dfrac{\gamma^e}{y^{\beta}} y^{\beta}$
   Thus the simulation outputs a valid transcript.

5. α should be uniformly distributed over $\mathbb{Z}_n^*$. In our simulation, since we have chosen a uniform γ over $\mathbb{Z}_n^*$, the result of $\dfrac{\gamma^e}{y^{\beta}}$ is also uniform over $\mathbb{Z}_n^*$. Thus the value of α is valid.

Similarly, since $\gamma \leftarrow rx^{\beta}$, γ is also a value that is uniformly distributed over $\mathbb{Z}_n^*$. Thus our value for γ is valid.
Since the verifier has to input β and it can be either 0 or 1, the value of β in the simulator is also valid.
Thus, the transcript given in output by the simulator is distributed identically to the real transcript computed via the interaction between prover and verifier

References:

[1] Rafael Pass, Abhi Shelat, "Knowledge" in *A Course in Cryptography,* 3rd ed., p. 126.