0 Commitment Schemes

a) (Definition of a Commitment Scheme). Write the formal definition (algorithms) of a commitment scheme. What are the two properties that a commitment scheme must satisfy?

Definition- An algorithm is a commitment scheme if $\exists$ a PPT algorithm $\ell(\cdot)$ for which the following two properties hold:

- **Binding**: $\forall\ n \in N$ and $\forall\ v_0, v_1 \in \{0,1\}^n$ and $r_0, r_1 \in \{0,1\}^{\ell}(n)$ it holds that $Commitment(v_0, r_0) \neq Commitment(v_1, r_1)$

- **Hiding**: $\forall$ PPT Distingusher D $\exists$ a negligible function $negl(n)$ such that $\forall$ $n \in N$ and $v_0, v_1 \in \{0,1\}^n$, D distinguishes the two commitments with at most $negl(n)$ probability.

b) (Impossibility of Commitment Scheme). Provide an informal argument for the fact that a commitment scheme **cannot** be both *statistically* hiding and *statistically* binding.

Answer-

- For a scheme to be statistically hiding, we have that the commitment of two different messages $\in \{m_1, m_2\}$ are the same, $\implies$ $Commitment(m_1) = Commitment(m_2)$

- For a scheme to be statistically binding, there are no two messages for which the the commitment is equal, $\implies$ $Commitment(m_1) \neq Commitment(m_2)$

Thus, since, the two points are contradictory, we cannot have a commitment scheme to be both statistically*hiding*and *binding*.

1 Commitment Schemes

(ElGamal Commitment Scheme). Let $\mathbb{G}$ be a group of order q, with generator g. Assume that the DDH assumption holds in $\mathbb{G}$. Let $h \leftarrow \mathbb{G}$ be an element of $\mathbb{G}$ sampled uniformly at random. $\mathbb{G}, q, g, h$ are publicly known to all parties. Consider the following procedures.

- Commitment Procedure. To commit to a message $m \in \mathbb{Z}_q$, the committer picks a random $u \leftarrow \mathbb{Z}_q$, and compute $(g^u, g^m h^u)$. Let us define $\mathsf{Com}(m, u) = (g^u, g^m h^u)$.
- Opening. To open a commitment, simply reveal (m, u).

This scheme is perfectly binding. There cannot exist $(m, u), (m', u') \leftarrow \mathbb{Z}_q \times \mathbb{Z}_q$ such that $\mathsf{Com}(m, u) = \mathsf{Com}(m', u')$. This scheme is computationally hiding. To prove hiding of this scheme we need to use the assumption that DDH assumption is true in $\mathbb{G}$.

1. (Hiding Proof by Reduction). Prove hiding of the commitment by showing a reduction to the DDH assumption. Namely, show that: if there exists a PPT adversarial receiver $\mathcal{A}_{\mathsf{hiding}}$ that is able to distinguish commitments of m_0 from commitments of m_1, then this adversary can be used to distinguish a DDH tuple from a random tuple. Recall that the DDH assumption says that given the tuple (g, g^a, g^b, g^c) any polynomial time adversary $\mathcal{A}_{\mathsf{ddh}}$ cannot tell whether $c = ab$ or c is an exponent chosen uniformly at random.

   Note. Your reduction $\mathcal{A}_{\mathsf{ddh}}$ takes in input a tuple (g, g_1, g_2, g_3), nothing else. The goal of the reduction is to use that tuple to generate the commitment for the receiver $\mathcal{A}_{\mathsf{hiding}}$.

   Hint. In the proof, the reduction is allowed to choose all parameters used in the commitment scheme.

   Reduction $\mathcal{A}_{\mathsf{ddh}}(g, g_1, g_2, g_3)$

   (a) ...
   (b) ...
   (c) ...
   (d) ...
   (e) Output

Theorem: If the DDH assumption holds in $\mathbb{G}$, then the commitment scheme is hiding.

Proof by Contradiction: If the commitment scheme is not hiding then $\exists$ a PPT algorithm $\mathcal{A}_{hiding}$ that is able to distinguish between the $Commit(m_0)$ and $Commit(m_1)$ with non-negligible probability $\frac{1}{2} + p(n)$

Assumption: $\exists$ PPT Adversary $\mathcal{A}_{DDH}$ that has oracle access that returns $g,\ g_1 = g^a,\ g_2 = g^b$ and $g_3 = g^{a \cdot b}$ or $g_3 = g^z$

Given Information: Assume another commitment scheme $\tilde{\Pi}$ similar to the El Gamal scheme where the adversary wins probability $= \frac{1}{2}$ Where the scheme is defined as follows:

- $\tilde{\Pi}(\mathbb{G}, q, g, h)$
  - Pick a random $u \leftarrow \mathbb{Z}_q$
  - Pick a random $z \leftarrow \mathbb{Z}_q$
  - Compute $Com(m, u) = (g^u, g^m \cdot g^z)$

Reduction: The adversary $\mathcal{A}_{DDH}$ queries the oracle to receive $g,\ g_1,\ g_2,\ g_3$, then it activates $\mathcal{A}_{hiding}$ to win the DDH Game.

$\mathcal{A}_{\mathsf{ddh}}(g, g_1, g_2, g_3)$

(a) $h = g_2$, $\mathbb{G}, q, h, g$ is made public

(b) $g^u = g_1$, $h^u = g_3$

(c) Pick a bit $b \in \{0, 1\}$

(d) Compute g^{m_b}

(e) Return $g_1, g^{m_b} \cdot g_3$

(f) If $\tilde{b} \neq b$ output 0 else output 1

Case Analysis:

- **Case 1:** If $g_3 = g^{a \cdot b}$, then the view is exactly like the El Gamal Commitment Scheme. Therefore we have the following that

  $Pr[\mathcal{A}_{DDH}\ wins\ DDH\ Game] = PR[\mathcal{A}\ wins\ hiding\ game] = \frac{1}{2} + p(n)$

- **Case 2:** If $g_3 = g^z$, then the view is exactly like the scheme $\tilde{\Pi}$. Therefore we have the following that

  $Pr[\mathcal{A}_{DDH}\ wins\ DDH\ Game] = PR[\mathcal{A}\ wins\ hiding\ game] = \frac{1}{2}$

But we assumed that the DDH assumption is *true*, therefore $\mathcal{A}_{DDH}$ cannot win the DDH game with non-negligible probability. Thus, the assumption must be false, since this is a contradiction. Taking Case 1 and Case 2 together we get $p(n)$ which is non-negligible, but cannot be so. Therefore we have that $PR[\mathcal{A}\ wins\ hiding\ game] = \frac{1}{2} + negl(n)$

2. What happens if the receiver knows $\log_g h$?

Since we have that $h \leftarrow \mathbb{G}$, there is some value for which we have that $g^x = h$ which would mean that if know $\log_g h$, we can find the value of x.

During the commitment, the prover sends $g^u, g^{m_b} \cdot h^u$, but then since we know the discrete log we can calculate $(g^u)^x$ which is equal to $h^u = (g^x)^u$.

Thus we can find the value of g^{m_b}, simultaneously calculate values of g^{m_0} and g^{m_1} and distinguish if the commitment is of m_0 or m_1.

Thus wining the game with probability 1 i.e., $Pr[\mathcal{A}\ wins\ game] = 1$

2 Zero Knowledge Proofs

The Guillou-Quisquater identification scheme is based on the RSA problem. It is an honest verifier zero-knowledge proof that the prover knows x such that $x^e = y \mod n$ where n is an RSA modulus. The **public information** is $\mathsf{pk} = (n, e, y)$ and the **corresponding secret** is x. 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 To prove completeness, we need to show that the equation that the verifier checks is indeed correct. Show that the above mentioned zero knowledge proof is complete.

Information Given: we have that V accepts the proof if $\gamma^e = \alpha y^\beta$

- We know that $\gamma \leftarrow rx^\beta$
- Therefore, we have that $\gamma^e = r^e \cdot (x^e)^\beta$
- We also know that $\alpha = r^e \implies \gamma^e = \alpha \cdot (x^e)^\beta$
- We also know that $x^e = y\ modn \implies \gamma^e = \alpha \cdot (y)^\beta$

Thus we can say the set of equations are valid and has the property of completeness.

Soundness To prove soundness, we want to show that if we have 2 accepting transcripts with the same first message, then we can extract the secret of the prover. Therefore this is a proof that the prover can convince the verifier only if she knows the secret.

1. How can we obtain 2 accepting transcripts from a prover that have the same first message? Recall, the proof is a mental experiment, so we can execute the prover as many times as we want.

Answer: It is given that the prover is interactive, and can be run multiple times. Therefore we can generate several transcripts *trans* as follows:

- Activate prover
- Get $\alpha \leftarrow r^e$
- Send $\beta \leftarrow_{\$} \{0, 1\}$
- Get back $\gamma \leftarrow r \cdot x^{\beta}$

From the proof on the previous page, we know that the *trans* is valid. Now we can use the *snapshot* (rewind) of the prover, to initialize it with the same r and secret x.

- Activate prover after rewinding
- Get $\alpha \leftarrow r^e$
- Send $\tilde{\beta} \leftarrow_{\$} \{0, 1\}$
  - Note: $\beta \neq \tilde{\beta}$
- Get back $\tilde{\gamma} \leftarrow r \cdot x^{\tilde{\beta}}$

From the proof on the previous page, we know that the *trans* is valid. Since we had that $\beta \neq \tilde{\beta}$, we get that $\tilde{\gamma} \neq \gamma$.

Thus we can interact with the prover to obtain two *trans* with the same first message x.

2. Assume that we obtained 2 accepting transcripts: (α, β, γ) and $(\alpha, \beta', \gamma')$. Show how you can extract the secret x.

Attack:
We know that $\gamma = r \cdot x^{\beta}$ and $\gamma' = r \cdot x^{\beta'}$. We also know the values for β and β'.

- Send $\beta = 0$, $\beta' = 1$
- Then we have that $\gamma = r$, $\gamma' = r \cdot x$

Thus we can simply calculate the secret by calculating $\dfrac{\gamma'}{\gamma}$

Zero knowledge Show a simulator that can compute an accepting transcript without knowing the secret. Your simulator must run in polynomial time. The input of the simulator is only the theorem $\mathsf{pk} = (n, e, y)$.

3. What is the transcript in this protocol?

We have the following the messages exchanged:

- Prover $\xrightarrow{\alpha}$ Verifier
- Prover $\xleftarrow{\beta}$ Verifier
- Prover $\xrightarrow{\gamma}$ Verifier

Thus the $trans = (\alpha, \beta, \gamma)$

4. Write the simulator :

$\underline{\mathsf{Sim}(n, e, y)}$

(a) Choose $\gamma \xleftarrow{\$} \mathbb{Z}_n^*$

(b) Choose $\beta \xleftarrow{\$} \{0, 1\}$

(c) Compute $\alpha = \dfrac{\gamma^e}{y^{\beta}}$

To check for compltleness, we know that $\gamma^e = \alpha \cdot y^{\beta}$. We have from the (c) that $\alpha = \dfrac{\gamma^e}{y^{\beta}}$. Thus on substituting we get, $\gamma^e = \dfrac{\gamma^e}{y^{\beta}} \cdot y^{\beta}$ which is a valid $trans$.

5. Argue (informally) that the transcript given in output by the simulator is distributed identically to the real transcript computed via the interaction between prover and verifier.

- For α we know that α should be uniformly distributed over $\mathbb{Z}_n^*$. We also have that $\alpha = \dfrac{\gamma^e}{y^\beta}$, thus the result of $\dfrac{\gamma^e}{y^\beta}$ is also uniform in $\mathbb{Z}_n^*$. Thus the value of α is valid.
- For $\beta \in \{0, 1\}$, thus even the value of β in the simulator is valid.
- We have that $\gamma \leftarrow r \cdot x^\beta$, thus we know that γ is distributed uniformly over $\mathbb{Z}_n^*$ which is valid. Similarly, the result of

Hence we have that the transcript given in output by the simulator is distributed identically to the real transcript computed via the interaction between prover and verifier.