CS 171: Problem Set 10

Due Date: April 25, 2024 at 8.59pm via Gradescope

1 Proof of Decryption (10 Points)

We will construct a zero-knowledge proof system for DDH triples. This can be used to prove that a given El Gamal ciphertext was decrypted correctly without revealing the secret decryption key.

Let $\mathsf{pp} = (\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ be a group in which DDH is hard. Let $\mathcal{L}$ be the language of DDH triples for this group:

$$\mathcal{L} = \{(\mathsf{pp}, g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}}) : \mathsf{c} = \mathsf{a} \cdot \mathsf{b} \mod q\}$$

Given an instance $x = (\mathsf{pp}, g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}}) \in \mathcal{L}$, let the corresponding witness be $w = \mathsf{b}$. The witness provides a simple way to verify that $x \in \mathcal{L}$:

$$R(x, w) = \begin{cases} 1 \text{ if} & g^w = g^{\mathsf{b}} \text{ and } (g^{\mathsf{a}})^w = g^{\mathsf{c}} \\ 0 & \text{else} \end{cases}$$

We can also prove that $x \in \mathcal{L}$ without revealing the witness to the verifier. To do so, we will construct a zero-knowledge proof below.

A Zero-Knowledge Protocol for $\mathcal{L}$:

- Inputs: The prover P takes inputs $(1^\lambda, x, w)$ and the verifier V takes inputs $(1^\lambda, x)$. $x = (\mathsf{pp}, g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}})$, and $w \in \mathbb{Z}_q$.
- P samples $\mathsf{x} \leftarrow \mathbb{Z}_q$, computes $g^{\mathsf{t}} = (g^{\mathsf{a}})^{\mathsf{x}}$, and sends $(g^{\mathsf{x}}, g^{\mathsf{t}})$ to V. Note that $\mathsf{t} = \mathsf{a} \cdot \mathsf{x} \mod q$.
- V samples $\mathsf{y} \leftarrow \mathbb{Z}_q$ and sends y to P.
- P computes $\mathsf{z} = w \cdot \mathsf{y} + \mathsf{x}$ and sends z to V.
- V checks that:
  1. $g^{\mathsf{z}} = (g^{\mathsf{b}})^{\mathsf{y}} \cdot g^{\mathsf{x}}$, and
  2. $(g^{\mathsf{a}})^{\mathsf{z}} = (g^{\mathsf{c}})^{\mathsf{y}} \cdot g^{\mathsf{t}}$

  If both checks pass, then the verifier accepts the proof. Otherwise, they reject.

Questions:

1. Show that this proof system satisfies completeness and soundness.
2. Show that this proof system satisfies honest-verifier zero-knowledge.

The definitions of completeness, soundness, and honest-verifier zero-knowledge are given in Discussion 11.

Solution

1. **Claim 1.1 (Completeness)** *If $R(x, w) = 1$, and if the prover and verifier follow the protocol honestly, then the verifier will accept the proof with probability 1.*

   Proof

   (a) $R(x, w) = 1$ if and only if $w = \mathsf{b}$, and $\mathsf{a} \cdot \mathsf{b} = \mathsf{c} \mod q$.

   (b) In this case, the verifier's checks will pass:

      i. $g^{\mathsf{z}} = g^{\mathsf{by}+\mathsf{x}} = (g^{\mathsf{b}})^{\mathsf{y}} \cdot g^{\mathsf{x}}$

      ii. $(g^{\mathsf{a}})^{\mathsf{z}} = g^{\mathsf{aby}+\mathsf{ax}} = g^{\mathsf{cy}+\mathsf{ax}} = (g^{\mathsf{c}})^{\mathsf{y}} \cdot g^{\mathsf{t}}$

   (c) Therefore, the verifier will accept the proof with probability 1.

   ■

2. **Claim 1.2 (Soundness)** *If $x \notin \mathcal{L}$, then when any adversarial prover P^* interacts with the honest verifier $V(1^\lambda, x)$, the probability that the verifier accepts the proof is $\mathsf{negl}(\lambda)$.*

   Proof

   (a) If $x \notin \mathcal{L}$, then $\mathsf{c} \neq \mathsf{ab} \mod q$.

   (b) Let the adversarial prover's first message be $(g^{\mathsf{x}}, g^{\mathsf{t}})$ for some $\mathsf{t} \in \mathbb{Z}_q$.[1]

   (c) The verifier accepts if and only if:

   $$\mathsf{z} = \mathsf{by} + \mathsf{x} \mod q$$
   $$\mathsf{az} = \mathsf{cy} + \mathsf{t} \mod q$$

   (d) We will show that the verifier accepts only if $\mathsf{y} = \frac{\mathsf{ax}-\mathsf{t}}{\mathsf{c}-\mathsf{ab}} \mod q$. To see why, let's do algebra on the equations above:

   $$\begin{aligned} \mathsf{az} &= \mathsf{aby} + \mathsf{ax} \mod q \\ \mathsf{az} &= \mathsf{cy} + \mathsf{t} \mod q \\ \mathsf{cy} + \mathsf{t} &= \mathsf{aby} + \mathsf{ax} \mod q \\ (\mathsf{c} - \mathsf{ab})\mathsf{y} &= \mathsf{ax} - \mathsf{t} \mod q \\ \mathsf{y} &= \frac{\mathsf{ax} - \mathsf{t}}{\mathsf{c} - \mathsf{ab}} \mod q \end{aligned}$$

   Note that we don't encounter a divide-by-zero error because $\mathsf{c} - \mathsf{ab} \neq 0 \mod q$.

   (e) Note that $(\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{t})$ are determined by the end of the prover's first message, before y is sampled. Then:

   $$\Pr_{\mathsf{y} \leftarrow \mathbb{Z}_q}\left[\mathsf{y} = \frac{\mathsf{ax} - \mathsf{t}}{\mathsf{c} - \mathsf{ab}} \mod q \middle| (\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{t})\right] = \frac{1}{q} = \mathsf{negl}(\lambda)$$

   This means that the verifier will accept the proof with probability $\leq \mathsf{negl}(\lambda)$.

[1] We can't assume that $\mathsf{t} = \mathsf{ax} \mod q$ because the prover is dishonest.

■

3. **Claim 1.3 (Honest-Verifier Zero-Knowledge)** *Let $x \in \mathcal{L}$, and let the prover and verifier follow the protocol honestly. Then there exists a simulator* Sim *such that* $\mathsf{view}(V; 1^\lambda, x, w)$ *is identically distributed to* $\mathsf{Sim}(1^\lambda, x)$.

   Proof

   (a) The verifier's view is the list of their inputs and the messages sent to and from the verifier during the protocol:

   $$\mathsf{view}(V; 1^\lambda, x, w) = \left[(1^\lambda, \mathsf{pp}, g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}}), (g^{\mathsf{x}}, g^{\mathsf{t}}, \mathsf{y}, \mathsf{z})\right]$$

   (b) $\underline{\mathsf{Sim}^V(1^\lambda, x)}$:

      i. Sample $\mathsf{y}, \mathsf{z} \leftarrow \mathbb{Z}_q$ independently and uniformly at random.

      ii. Compute

      $$g^{\mathsf{x}} = g^{\mathsf{z}} \cdot (g^{\mathsf{b}})^{-\mathsf{y}} \tag{1.1}$$
      $$g^{\mathsf{t}} = (g^{\mathsf{a}})^{\mathsf{z}} \cdot (g^{\mathsf{c}})^{-\mathsf{y}} \tag{1.2}$$

      iii. Output

      $$(1^\lambda, \mathsf{pp}, g^{\mathsf{a}}, g^{\mathsf{b}}, g^{\mathsf{c}}), (g^{\mathsf{x}}, g^{\mathsf{t}}, \mathsf{y}, \mathsf{z})$$

4. We will argue that the distribution of $(\mathsf{y}, \mathsf{z}, g^{\mathsf{x}}, g^{\mathsf{t}})$ is the same in the real and simulated protocols.

5. <u>In the simulated protocol:</u> $(\mathsf{x}, \mathsf{t}, \mathsf{y}, \mathsf{z})$ have the following distribution: for a given $(\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{t}, \mathsf{y}, \mathsf{z})$,

   $$\Pr[X = \mathsf{x}, T = \mathsf{t}, Y = \mathsf{y}, Z = \mathsf{z}] = \begin{cases} \frac{1}{q^2} & \text{if eqs. 1.1 and 1.2 are satisfied} \\ 0 & \text{otherwise} \end{cases}$$

   The randomness comes from Y, Z, which are independent and uniformly random.

6. <u>In the real protocol:</u>

   (a) $(\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{t}, \mathsf{y}, \mathsf{z})$ will satisfy equations 1.1 and 1.2. This is because

   $$\mathsf{z} = \mathsf{b} \cdot \mathsf{y} + \mathsf{x}$$
   $$\mathsf{x} = \mathsf{z} - \mathsf{b} \cdot \mathsf{y}$$

   and

   $$\mathsf{t} = \mathsf{a} \cdot \mathsf{x} = \mathsf{a} \cdot \mathsf{z} - \mathsf{a} \cdot \mathsf{b} \cdot \mathsf{y}$$
   $$= \mathsf{a} \cdot \mathsf{z} - \mathsf{c} \cdot \mathsf{y}$$

   (b) x and y are sampled independently and uniformly at random. And for a given $(\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{y})$: (t, z) take the unique values that satisfy equations 1.1 and 1.2.

(c) Therefore, for a given $(\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{x}, \mathsf{t}, \mathsf{y}, \mathsf{z})$:

$$\Pr[X = \mathsf{x}, T = \mathsf{t}, Y = \mathsf{y}, Z = \mathsf{z}] = \begin{cases} \frac{1}{q^2} & \text{if eqs. 1.1 and 1.2 are satisfied} \\ 0 & \text{otherwise} \end{cases}$$

The randomness comes from X, Y, which are independent and uniformly random.

7. We've show that the distribution of $(\mathsf{x}, \mathsf{t}, \mathsf{y}, \mathsf{z})$ is the same in the real protocol and the simulated protocol.

   That means the distribution of $\mathsf{view}(V; 1^\lambda, x, w)$ is identical to the distribution of $\mathsf{Sim}^V(1^\lambda, x)$, so the protocol satisfies honest-verifier zero-knowledge. ■

■

2 Hiding and Binding For KZG Commitments (15 Points)

In discussion 11, we showed that the basic KZG commitment protocol is not hiding because the Commit function is deterministic. In section 2.1 below, we give a modified version of the scheme in which the Commit function is randomized.

Question: Prove that the commitment scheme given in section 2.1 satisfies the notions of hiding and polynomial binding given in section 2.2, assuming that the d-discrete log problem is hard.

2.1 A Randomized Polynomial Commitment Scheme

1. $\mathsf{Gen}(1^n)$:
   (a) Let d be polynomial in n.
   (b) Set up a bilinear map by sampling
   $$\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$$
   (c) Sample $h \leftarrow \mathbb{G}$ and $\tau \leftarrow \mathbb{Z}_q^*$.
   (d) Finally, output
   $$\mathsf{params} = \left(\mathsf{pp}, g^{\tau}, g^{(\tau^2)}, \ldots, g^{(\tau^d)}, h, h^{\tau}, h^{(\tau^2)}, \ldots, h^{(\tau^d)}\right)$$
2. $\mathsf{Commit}(\mathsf{params}, f)$:
   (a) Let f be a polynomial $\in \mathbb{Z}_q[X]$ of degree $\leq d$:
   $$f(X) = \sum_{i=0}^{d} \alpha_i \cdot X^i$$
   where every $\alpha_i \in \mathbb{Z}_q$.
   (b) Sample a polynomial $r \in \mathbb{Z}_q[X]$ of degree $\leq d$ uniformly at random. In other words, sample $\beta_0, \ldots, \beta_d \leftarrow \mathbb{Z}_q$ independently and uniformly at random, and let
   $$r(X) = \sum_{i=0}^{d} \beta_i \cdot X^i$$
   (c) Compute and output the commitment:
   $$\begin{aligned}\mathsf{com} &= \prod_{i=0}^{d} \left(g^{(\tau^i)}\right)^{\beta_i} \cdot \prod_{i=0}^{d} \left(h^{(\tau^i)}\right)^{\alpha_i} \\ &= g^{r(\tau)} \cdot h^{f(\tau)}\end{aligned}$$

   Note: We also define $\mathsf{Commit}(\mathsf{params}, f; r)$ to take the random polynomial r as input, rather than sampling r internally.

2.2 Definitions

Hiding basically says that $\mathsf{Commit}(f, \mathsf{params})$ doesn't reveal any information about f. The definition of hiding resembles the definition of CPA security.

Definition 2.1 (Hiding)

$\underline{\mathsf{Hiding\text{-}Game}(n, \mathcal{A})}$*:*

1. *The challenger samples* $\mathsf{params} \leftarrow \mathsf{Gen}(1^n)$ *and sends* params *to the adversary* $\mathcal{A}$*.*
2. $\mathcal{A}$ *outputs two polynomials* $f_0, f_1 \in \mathbb{Z}_q[X]$ *of degree* $\leq d$*.*
3. *The challenger samples* $b \leftarrow \{0, 1\}$ *and computes:* $\mathsf{com}^* = \mathsf{Commit}(\mathsf{params}, f_b)$*. They send* com^* *to* $\mathcal{A}$*.*
4. $\mathcal{A}$ *outputs a guess* b' *for* b*. The output of the game is* 1 *if* $b' = b$ *and* 0 *otherwise.*

The commitment scheme is ***hiding*** *if for any PPT adversary* $\mathcal{A}$*,*

$$\Pr[\mathsf{Hiding\text{-}Game}(n, \mathcal{A}) \to 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

Next, we'll consider a notion called polynomial binding, which says that the adversary cannot find two inputs to Commit that produce the same commitment. This resembles the definition of collision-resistance.

Definition 2.2 (Polynomial Binding)

$\underline{\mathsf{Polynomial\text{-}Binding\text{-}Game}(n, \mathcal{A})}$*:*

1. *The challenger samples* $\mathsf{params} \leftarrow \mathsf{Gen}(1^n)$ *and sends* params *to the adversary* $\mathcal{A}$*.*
2. $\mathcal{A}$ *outputs two pairs* (f_0, r_0) *and* (f_1, r_1)*, where* f_0, r_0, f_1, r_1 *are polynomials* $\in \mathbb{Z}_q[X]$ *of degree* $\leq d$*.*
3. *The output of the game is* 1 *if* $f_0 \neq f_1$*, and*

$$\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$$

Otherwise, the output of the game is 0*.*

The commitment scheme satisfies ***polynomial binding*** *if*

$$\Pr[\mathsf{Polynomial\text{-}Binding\text{-}Game}(n, \mathcal{A}) \to 1] \leq \mathsf{negl}(n)$$

Finally, we will prove polynomial binding using the hardness of the following problem.

Definition 2.3 (A Variant of Discrete Log)
$\underline{d\text{-}\mathsf{Discrete\text{-}Log}(n, \mathcal{A})}$*:*

1. *Let* d *be polynomial in* n*.*

2. *The challenger samples* $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$ *as well as* $\tau \leftarrow \mathbb{Z}_q$. *Then they send the adversary:* $\left(\mathsf{pp}, g^{\tau}, g^{(\tau^2)}, \ldots, g^{(\tau^d)}\right)$

3. *The adversary* $\mathcal{A}$ *outputs a guess* τ' *for* τ. *The output of the game is* 1 *if* $\tau' = \tau$ *and* 0 *otherwise.*

The d*-discrete-log problem is hard if for any PPT adversary* $\mathcal{A}$*,*

$$\Pr[d\text{-}\mathsf{Discrete\text{-}Log}(n, \mathcal{A}) \to 1] \leq \mathsf{negl}(n)$$

Note that if the d-discrete-log problem is hard, then in addition, the regular discrete log problem is hard for $\mathbb{G}$.

Solution

1. **Claim 2.4** *The commitment scheme is hiding.*

   Proof

   (a) Key Idea: $g^{r(\tau)}$ is uniformly random over the randomness of r, so $g^{r(\tau)}$ masks the value of $h^{f(\tau)}$.

   (b) For any given τ, $r(\tau)$ is uniformly random in $\mathbb{Z}_q$, over the randomness of r. Then for any poynomial f and any parameters params, the output of $\mathsf{Commit}(\mathsf{params}, f)$ is uniformly random in $\mathbb{G}$ due to the randomness of r.

   (c) Then the commitment $\mathsf{com}^* = \mathsf{Commit}(\mathsf{params}, f_b)$ is actually independent of b. In this case, the adversary's probability of correctly guessing b is exactly $\frac{1}{2}$. Therefore, the scheme is hiding.

   ■

2. **Claim 2.5** *The commitment scheme satisfies polynomial binding.*

   Proof

   (a) If the d-discrete log problem is hard, then in addition, the regular discrete log problem is hard for $\mathbb{G}$. Assume toward contradiction that there is an adversary $\mathcal{A}_{\mathsf{Binding}}$ that breaks polynomial binding. Then we will use $\mathcal{A}_{\mathsf{Binding}}$ to construct two adversaries:

   - An adversary $\mathcal{B}$ that solves the discrete log problem in $\mathbb{G}$ to find the $x \in \mathbb{Z}_q$ for which $h = g^x$.
   - An adversary $\mathcal{C}$ that solves the d-discrete log problem by computing τ from params.

   We will show that one of these adversaries succeeds with non-negligible probability. This is a contradiction because discrete log and d-discrete log are hard. Therefore, there is no adversary that breaks binding.

   (b) Construction of $\mathcal{B}$:

   i. The discrete log challenger samples $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. They send (pp, g^x) to $\mathcal{B}$.

   ii. $\mathcal{B}$ sets $h = g^x$. Then they sample $\tau \leftarrow \mathbb{Z}_q^*$ and compute

   $$\mathsf{params} = \left(\mathsf{pp}, g^{\tau}, g^{(\tau^2)}, \ldots, g^{(\tau^d)}, h, h^{\tau}, h^{(\tau^2)}, \ldots, h^{(\tau^d)}\right)$$

   iii. They run $\mathcal{A}_{\mathsf{Binding}}$ on input params. With non-negligible probability, $\mathcal{A}_{\mathsf{Binding}}$ outputs (f_0, r_0) and (f_1, r_1) such that $f_0 \neq f_1$, and $\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$.

   iv. Compute and output

   $$x = \frac{r_1(\tau) - r_0(\tau)}{f_0(\tau) - f_1(\tau)}$$

(c) If $\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$, and $f_0(\tau) \neq f_1(\tau)$, then

$$\begin{aligned} g^{r_0(\tau)} \cdot h^{f_0(\tau)} &= g^{r_1(\tau)} \cdot h^{f_1(\tau)} \\ h^{f_0(\tau)-f_1(\tau)} &= g^{r_1(\tau)-r_0(\tau)} \\ x \cdot [f_0(\tau) - f_1(\tau)] &= r_1(\tau) - r_0(\tau) \mod q \\ x &= \frac{r_1(\tau) - r_0(\tau)}{f_0(\tau) - f_1(\tau)} \mod q \end{aligned}$$

In this case, $\mathcal{B}$ solves the discrete log problem.

3. Construction of $\mathcal{C}$:

(a) The d-discrete-log challenger samples $\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$ as well as $\tau \leftarrow \mathbb{Z}_q$. Then they send the adversary:

$$\left(\mathsf{pp}, g^\tau, g^{(\tau^2)}, \ldots, g^{(\tau^d)}\right)$$

(b) The adversary $\mathcal{C}$ samples $x \leftarrow \mathbb{Z}_q$ and computes

$$h = g^x, \quad h^\tau = (g^\tau)^x, \quad \ldots, \quad h^{(\tau^d)} = \left(g^{(\tau^d)}\right)^x$$
$$\mathsf{params} = \left(\mathsf{pp}, g^\tau, g^{(\tau^2)}, \ldots, g^{(\tau^d)}, h, h^\tau, h^{(\tau^2)}, \ldots, h^{(\tau^d)}\right)$$

(c) They run $\mathcal{A}_{\mathsf{Binding}}$ on input params. With non-negligible probability, $\mathcal{A}_{\mathsf{Binding}}$ outputs (f_0, r_0) and (f_1, r_1) such that $f_0 \neq f_1$, and $\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$.

(d) If $f_0 \neq f_1$, then compute the roots of the polynomial $f_0(X) - f_1(X)$. For each root τ', check whether $g^{\tau'} = g^\tau$. If so, output τ'.

4. If $f_0 \neq f_1$, then $f_0(X) - f_1(X)$ is a non-zero polynomial of degree $\leq d$. Therefore, it has at most d roots. If $f_0(\tau) = f_1(\tau)$, then τ is one of those roots, so $\mathcal{C}$ will find τ and win the d-discrete-log game.

5. One of the following events occurs with non-negligible probability:

(a) $\mathcal{A}_{\mathsf{Binding}}$ outputs (f_0, r_0) and (f_1, r_1) such that $f_0 \neq f_1$, and $\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$, and $\underline{f_0(\tau) \neq f_1(\tau)}$.

(b) $\mathcal{A}_{\mathsf{Binding}}$ outputs (f_0, r_0) and (f_1, r_1) such that $f_0 \neq f_1$, and $\mathsf{Commit}(\mathsf{params}, f_0; r_0) = \mathsf{Commit}(\mathsf{params}, f_1; r_1)$, and $\underline{f_0(\tau) = f_1(\tau)}$.

If the first event has non-negligible probability, then $\mathcal{B}$ breaks the hardness of discrete log. If the second event has non-negligible probability, then $\mathcal{C}$ breaks the hardness of d-discrete log.

In either case, we've arrived at a contradiction because both discrete log in $\mathbb{G}$ and d-discrete log are assumed to be hard. Therefore, there does not exist a PPT adversary $\mathcal{A}_{\mathsf{Binding}}$ that breaks the polynomial binding of the commitment scheme. ■

■

3 Course Evaluation (Extra Credit: 2 Points)

Complete your course evaluation for this course. You can write as much or as little as you want. Include a screenshot of the submission receipt when you submit this assignment to Gradescope to prove that you've finished your evaluation.