CS 171: Discussion Section 9 (April 1)

1 Group Operations

Definitions: Let $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ be the description of a cyclic group for which the discrete log problem is hard. $|\mathbb{G}| = q \approx 2^n$, and $g \in \mathbb{G}$ is a generator of $\mathbb{G}$. Next, let $h \in \mathbb{G}$ be an arbitrary group element, and sample $a, x, y \leftarrow \mathbb{Z}_q$ independently and uniformly.

Question: For each of the following tasks, describe how it can be performed efficiently (in $\mathsf{poly}(n)$ time) or prove that it cannot be performed efficiently. For each task, assume that you are given $(\mathbb{G}, q, g)$, the parameters of the group.

1. Given x, g, compute g^x.
2. Sample a uniformly random element of $\mathbb{G}$.
3. Given h, compute h^{-1}.
4. Given a, y, g, g^x, compute $g^{a \cdot x - y}$.
5. Given $a, g^{a \cdot x}$, compute $a \cdot x$.

Solution

1. This can be done efficiently. The naive algorithm requires $x = O(q)$ multiplications: $g^x = \Pi_{i \in [x]} g$. However, there is a more-efficient algorithm based on repeated squaring that requires $O(\log q)$ multiplications.

   (a) Repeated Squaring Algorithm: Compute $g^{(2^i)}$ for every $i \in \{0, \ldots, \log(q) - 1\}$ as follows:
      i. $g^1 = g$
      ii. $g^2 = g^1 \cdot g^1$
      iii. $g^4 = g^2 \cdot g^2$
      iv. $g^8 = g^4 \cdot g^4$
      v. Etc.

      This requires $\log(q) - 1$ multiplications in total.

   (b) Then to compute g^x: write x in binary as $\vec{x} \in \{0,1\}^{\log q}$. Let $\vec{x}_0$ be the lowest-order bit, and let $\vec{x}_{\log(q)-1}$ be the highest-order bit. Then compute

   $$g^x = \prod_{i : \vec{x}_i = 1} g^{(2^i)}$$

   This requires $O(\log q)$ multiplications.

2. This can be done efficiently. Sample $x \leftarrow \mathbb{Z}_q$ and compute $h = g^x$.

3. This can be done efficiently. Compute h^{q-1} (using repeated squaring). Note that $h^{q-1} = h^{-1}$ because $h \cdot h^{q-1} = h^{q-1} \cdot h = h^q = 1$.

   Here, we used the fact that $h^q = 1$.

Note: Cyclic groups have the following useful property:

$$g^x = g^{x \mod q}$$

for any $g \in \mathbb{G}$ and any $x \in \mathbb{Z}$, where $q = |\mathbb{G}|$.

4. This can be done efficiently.
   (a) Compute $g^{a \cdot x} = (g^x)^a$ using repeated squaring.
   (b) Compute g^{-y}.
   (c) Compute $g^{a \cdot x - y} = g^{a \cdot x} \cdot g^{-y}$

5. No efficient algorithm can succeed at this task with non-negligible probability. This follows from the hardness of discrete log.

   Proof.

   (a) Key Ideas: We can turn this task into the discrete log problem mainly by renaming our variables. We will also need the fact that a is independent of $a \cdot x$ (due to the randomness of x), so a gives no useful information to the discrete log adversary.
   (b) Let $y = a \cdot x$, and let $h = g^{a \cdot x}$. Since a and x are independent and uniformly random, then a and h are statistically close to independent and uniformly random.
   (c) Now with this new notation, the problem that $\mathcal{A}$ solves is the following:
   Modified Discrete Log Game:
      i. The challenger samples $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$. Then they sample $a \leftarrow \mathbb{Z}_q$. If $a = 0$, then $h = 1$. Otherwise, sample $h \leftarrow \mathbb{G}$. Finally, they give the adversary $\mathcal{A}$ the variables $(\mathbb{G}, q, g, h, a)$.
      ii. $\mathcal{A}$ outputs $y' \in \mathbb{Z}_q$.
      iii. The output of the game is 1 if $h = g^{y'}$, and the output is 0 otherwise.
   (d) Reduction to discrete log: We will show that if there exists a PPT adversary $\mathcal{A}$ for which the *Modified Discrete Log Game* outputs 1 with non-negligible probability, then we can construct an adversary $\mathcal{B}$ that wins the *Discrete Log Game* with non-negligible probability. This is a contradiction because discrete log is hard relative to $\mathcal{G}$, so in fact, $\mathcal{A}$ cannot win the *Modified Discrete Log Game* with greater than negligible probability.
   (e) Construction of $\mathcal{B}$ (the discrete log adversary):
      i. The discrete log challenger samples $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $h \leftarrow \mathbb{G}$. Then they give $\mathcal{B}$ the variables $(\mathbb{G}, q, g, h)$.
      ii. $\mathcal{B}$ samples $a \leftarrow \mathbb{Z}_q$ and runs $\mathcal{A}(\mathbb{G}, q, g, h, a)$ until $\mathcal{A}$ outputs y'. $\mathcal{B}$ also outputs y'.
   (f) $\mathcal{B}$ simulates the *Modified Discrete Log Game* up to negligible statistical error. This is because in the *Modified Discrete Log Game*, a is statistically close to independent of $(\mathbb{G}, q, g, h)$, and h is statistically close to uniformly random.
   That means with non-negligible probability, $\mathcal{A}$ and $\mathcal{B}$ will output a y' such that $h = g^{y'}$. This is the answer that $\mathcal{B}$ needs to win the *Discrete Log Game*, so $\mathcal{B}$ wins the *Discrete Log Game* with non-negligible probability.

□

□

2 Another PKE Construction from DDH

Consider the following public-key encryption scheme, which is based on El Gamal encryption.

1. $\mathsf{Gen}(1^n)$: Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Then compute $h = g^x$. Next,

$$\text{let } \mathsf{pk} = (\mathbb{G}, q, g, h)$$
$$\mathsf{sk} = (\mathbb{G}, q, g, x)$$

2. $\mathsf{Enc}(\mathsf{pk}, m)$: Let $m \in \{0, 1\}$. First, sample $y \leftarrow \mathbb{Z}_q$. Next:

   (a) If $m = 0$, compute and output the following ciphertext:

   $$c = (c_1, c_2) = (g^y, h^y)$$

   (b) If $m = 1$, then sample $z \leftarrow \mathbb{Z}_q$ and output the following ciphertext:

   $$c = (c_1, c_2) = (g^y, g^z)$$

3. $\mathsf{Dec}(\mathsf{sk}, c)$: TBD

Questions:

1. Fill in the algorithm $\mathsf{Dec}(\mathsf{sk}, c)$ so that the scheme is efficient and correct, up to negligible error.

2. Prove that this encryption scheme is CPA-secure if DDH is hard.

Solution

Part 1: Decryption

1. $\mathsf{Dec}(\mathsf{sk}, c)$: Check whether $c_1^x = c_2$. If so output 0, and if not output 1.

2. This encryption scheme is clearly efficient.

3. Now we will show correctness:

   Claim 2.1. *For any* $(\mathsf{pk}, \mathsf{sk}, m)$,

   $$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m] \geq 1 - \mathsf{negl}(n)$$

   where the probability is over the randomness of Enc.

   Proof. First, if $c = \mathsf{Enc}(\mathsf{pk}, 0)$, then

   $$c_1^x = (g^y)^x = g^{x \cdot y} = (g^x)^y = h^y = c_2$$

   Then $\mathsf{Dec}(\mathsf{sk}, c)$ will output 0.

Second, if $c = \mathsf{Enc}(\mathsf{pk}, 1)$, then $c_2 = g^z$. Decryption will be incorrect only if $c_1^x = c_2$. In this case, $g^{x \cdot y} = g^z$, so $x \cdot y = z \mod q$. Next, since z is uniformly random,

$$\Pr_z[x \cdot y = z \mod q] = \frac{1}{q} = \mathsf{negl}(n)$$

In summary, over the randomness of Enc:

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, 1)) = 0] = \mathsf{negl}(n)$$

□

Part 2: CPA security

Claim 2.2. *If DDH is hard relative to $\mathcal{G}$, then the encryption scheme defined above satisfies CPA security.*

Proof.

1. Key ideas: An adversary that tries to break DDH is given (g^x, g^y, g^z) and must distinguish whether $z = x \cdot y \mod q$ or $z \leftarrow \mathbb{Z}_q$. These two cases correspond to encryptions of 0 and 1 respectively. If there were a CPA adversary that could tell whether the challenge ciphertext encrypts 0 or 1, then they could be used to break DDH.

2. Overview: Assume toward contradiction that there's a PPT adversary $\mathcal{A}$ that breaks the CPA security of the encryption scheme. Then we will use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that wins the DDH game with non-negligible advantage. This is a contradiction because no PPT adversary can win the DDH game with non-negligible advantage. Therefore, our assumption was false and in fact, the encryption scheme is CPA-secure.

3. Let us require that $\mathcal{A}$'s challenge messages are always $m_0 = 0$ and $m_1 = 1$. This is without loss of generality. The intuition is that there are only two possible messages $\{0, 1\}$ to choose from.

4. Construction of $\mathcal{B}$ (the DDH adversary):

   (a) The DDH challenger samples $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$, and also samples $x, y \leftarrow \mathbb{Z}_q$ independently. Then they either set $z = x \cdot y \mod q$ or sample $z \leftarrow \mathbb{Z}_q$. Finally, they give $\mathcal{B}$ the values $(\mathbb{G}, q, g, g^x, g^y, g^z)$.

   (b) $\mathcal{B}$ will simulate the CPA security game. They set $\mathsf{pk} = (\mathbb{G}, q, g, g^x)$. Then they run the CPA adversary $\mathcal{A}$ on input pk.

   (c) When $\mathcal{A}$ outputs two challenge messages $m_0 = 0$ and $m_1 = 1$, $\mathcal{B}$ ignores them and returns $c^* = (g^y, g^z)$.

   (d) When $\mathcal{A}$ outputs a bit b', $\mathcal{B}$ outputs b' as well.

5. For any given $b \in \{0, 1\}$, let $\mathsf{CPA}(\mathcal{A}, b)$ be the CPA game in which the challenge ciphertext is always an encryption of m_b, and the output of the game is whatever bit b' that $\mathcal{A}$ outputs.

Since $\mathcal{A}$ breaks CPA security,

$$\left| \Pr[\mathsf{CPA}(\mathcal{A}, 0) \to 1] - \Pr[\mathsf{CPA}(\mathcal{A}, 1) \to 1] \right| \geq \mathsf{non\text{-}negl}(n)$$

Furthermore, when the DDH challenger sets $z = x \cdot y \mod q$, $\mathcal{B}$ ends up simulating $\mathsf{CPA}(\mathcal{A}, 0)$, and when the DDH challenge samples $z \leftarrow \mathbb{Z}_q$, $\mathcal{B}$ ends up simulating $\mathsf{CPA}(\mathcal{A}, 1)$. Therefore, $\mathcal{B}$ distinguishes these two cases with non-negligible advantage.

□

□