CS 171: Discussion Section 10 (April 8)

1 Which Tasks Become Easy With Bilinear Maps?

Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map for which the *decisional bilinear Diffie-Hellman* (DBDH) problem is hard.

1. For each of the following computational problems, indicate whether the following problems are hard:

   (a) DDH in $\mathbb{G}$

   (b) CDH in $\mathbb{G}$

   (c) DDH in $\mathbb{G}_T$

2. Will the Diffie-Hellman key-exchange protocol be secure if we use group $\mathbb{G}$? How about if we use $\mathbb{G}_T$?

Solution

1. Summary: We will show that DDH in $\mathbb{G}$ is easy to solve with the help of the bilinear map $e(\cdot)$. But the other problems listed above are hard. Next, the Diffie-Hellman key exchange protocol will be secure if it uses $\mathbb{G}_T$, but insecure if it uses $\mathbb{G}$. The protocol is secure if it uses a group for which DDH is hard.

2. Let us recall the DBDH problem:

   Definition 1.1 (Decisional Bilinear Diffie-Hellman Problem).

   $\mathsf{DBDH}(n, \mathcal{A})$*:*

   (a) The challenger samples the parameters of the bilinear map:

   $$\mathsf{pp} = (\mathbb{G}, \mathbb{G}_T, q, g, e) \leftarrow \mathcal{G}(1^n)$$

   (b) The challenger samples $a, b, c, r \leftarrow \mathbb{Z}_q$ independently and also samples $\beta \leftarrow \{0, 1\}$. Then they give the adversary the inputs:

   $$(\mathsf{pp}, g^a, g^b, g^c, e(g, g)^{abc+r\beta})$$

   (c) $\mathcal{A}$ outputs a guess β' for β.

   (d) The output of the game is 1 (win) if $\beta' = \beta$ and 0 (lose) otherwise.

   *We say that **the DBDH problem is hard** if for all PPT adversaries $\mathcal{A}$,*

   $$\left| \Pr[\mathsf{DBDH}(n, \mathcal{A}) \to 1] - \frac{1}{2} \right| \le \mathsf{negl}(n)$$

3.

   Claim 1.2. *DDH in $\mathbb{G}$ is easy.*

Proof. DDH in $\mathbb{G}$ can be solved efficiently as follows:

(a) The DDH challenger samples $x, y \leftarrow \mathbb{Z}_q$ independently and sends the adversary $(\mathbb{G}, q, g, g^x, g^y, g^z)$, where either $z = x \cdot y \mod q$ or $z \leftarrow \mathbb{Z}_q$.

(b) The adversary computes $e(g^x, g^y) = e(g, g)^{x \cdot y}$ and $e(g, g^z) = e(g, g)^z$ and checks whether:

$$e(g, g)^{x \cdot y} = e(g, g)^z \tag{1.1}$$

If so, the adversary guesses that $z = x \cdot y \mod q$. If not, they guess that $z \leftarrow \mathbb{Z}_q$.

The adversary will win the DDH game with probability $1 - \mathsf{negl}(n)$. $e(g, g)$ is a generator for $\mathbb{G}_T$, so equation 1.1 is satisfied if and only if $z = x \cdot y \mod q$. The only way the adversary can lose is if $z \leftarrow \mathbb{Z}_q$ happens to produce $z = x \cdot y \mod q$, and this occurs with negligible probability. □

4.

Claim 1.3. *CDH in $\mathbb{G}$ is hard.*

Proof.

(a) If CDH in $\mathbb{G}$ were easy, then we could use the CDH attacker $\mathcal{A}_{\mathsf{CDH}}$ to solve the DBDH problem.

(b) Here is a construction of an adversary for the DBDH game $\mathcal{A}_{\mathsf{DBDH}}$:
$\mathcal{A}_{\mathsf{DBDH}}$:
   i. $\mathcal{A}_{\mathsf{DBDH}}$ receives inputs $(\mathsf{pp}, g^a, g^b, g^c, g^{abc+r\beta})$.
   ii. They run $\mathcal{A}_{\mathsf{CDH}}(\mathbb{G}, q, g, g^a, g^b)$ which outputs h.
   iii. They check whether
   $$e(g^a, g^b) = e(g, h)$$
   which is equivalent to checking whether $h = g^{ab}$. If not, they sample and output $\beta' \leftarrow \{0, 1\}$ and halt. If so, they continue.
   iv. Then they check whether
   $$e(h, g^c) = e(g, g^{abc+r\beta})$$
   When $h = g^{ab}$, this is equivalent to checking whether $abc = abc + r\beta$. If so, they output $\beta' = 0$. If not, they output $\beta' = 1$.

(c) The point of checking whether $e(g^a, g^b) = e(g, h)$ is to determine whether $h = g^{ab}$. The two conditions are equivalent. $\mathcal{A}_{\mathsf{CDH}}$ will compute $h = g^{ab}$ with non-negligible probability.

(d) If $h = g^{ab}$, then checking whether $e(h, g^c) = e(g, g^{abc+r\beta})$ will correctly decide the value of β with probability $1 - \mathsf{negl}(n)$.
If $h = g^{ab}$, then $e(h, g^c) = g^{abc}$. The condition $e(h, g^c) = e(g, g^{abc+r\beta})$ will pass if and only if $abc = abc + \beta \cdot r$.
Then the only way that $\beta' \neq \beta$ is if $r = 0$, but this only occurs with negligible probability.

(e) On the other hand, if $h \neq g^{ab}$, then $\mathcal{A}_{\mathsf{DBDH}}$ is unable to learn any useful information about β, so they guess randomly ($\beta' \leftarrow \{0,1\}$). This guess is correct with probability $\frac{1}{2}$.

(f) In total, the success probability of $\mathcal{A}_{\mathsf{DBDH}}$ at guessing β is:

$$\begin{aligned}\Pr[h = g^{ab}] \cdot \big(1 - \mathsf{negl}(n)\big) + \big(1 - \Pr[h = g^{ab}]\big) \cdot \frac{1}{2} &= \frac{1}{2} + \Pr[h = g^{ab}] \cdot \Big(1 - \frac{1}{2} - \mathsf{negl}(n)\Big) \\ &= \frac{1}{2} + \mathsf{non\text{-}negl}(n) \cdot \Big(\frac{1}{2} - \mathsf{negl}(n)\Big) \\ &= \frac{1}{2} + \mathsf{non\text{-}negl}(n)\end{aligned}$$

(g) In summary, we've shown that if CDH in $\mathbb{G}$ is easy, then DBDH is easy. That's a contradiction because we are told that DBDH is hard. Therefore, CDH in $\mathbb{G}$ in actually hard.

□

5.

Claim 1.4. *DDH in $\mathbb{G}_T$ is hard.*

Proof.

(a) If DDH in $\mathbb{G}_T$ were easy, then we could use the DDH attacker $\mathcal{A}_{\mathsf{DDH}}$ to solve the DBDH problem. Without loss of generality, let us assume that if DDH is easy in $\mathbb{G}_T$, then $\Pr[\mathcal{A}_{\mathsf{DDH}}$ is correct$] \geq \frac{1}{2} + \mathsf{non\text{-}negl}(n)$.

(b) Here is a construction of an adversary for the DBDH game $\mathcal{A}_{\mathsf{DBDH}}$:

$\mathcal{A}_{\mathsf{DBDH}}$:

i. $\mathcal{A}_{\mathsf{DBDH}}$ receives inputs $(\mathsf{pp}, g^a, g^b, g^c, g^{abc+r\beta})$.

ii. They compute $e(g^a, g^b) = e(g,g)^{ab}$ and $e(g, g^c) = e(g,g)^c$.

iii. They run $\mathcal{A}_{\mathsf{DDH}}(\mathbb{G}_T, q, e(g,g), e(g,g)^{ab}, e(g,g)^c, e(g,g)^{abc+r\beta})$, which correctly decides whether $abc = abc + r\beta$ with non-negligible advantage.

iv. If $\mathcal{A}_{\mathsf{DDH}}$ says that $abc = abc + r\beta$, then $\mathcal{A}_{\mathsf{DBDH}}$ outputs $\beta' = 0$. Otherwise, they output $\beta' = 1$.

(c) As long as $r \neq 0$ and $\mathcal{A}_{\mathsf{DDH}}$ correctly decides whether $abc = abc + r\beta$, then $\mathcal{A}_{\mathsf{DBDH}}$ correctly guesses β.

Then:

$$\begin{aligned}\Pr[\mathcal{A}_{\mathsf{DBDH}} \text{ succeeds}] &\geq \Pr[\mathcal{A}_{\mathsf{DDH}} \text{ succeeds}] \cdot \Pr[r \neq 0] = \Pr[\mathcal{A}_{\mathsf{DDH}} \text{ succeeds}] \cdot \big(1 - \mathsf{negl}(n)\big) \\ &= \Pr[\mathcal{A}_{\mathsf{DDH}} \text{ succeeds}] - \mathsf{negl}(n) \\ &\geq \frac{1}{2} + \mathsf{non\text{-}negl}(n)\end{aligned}$$

(d) In summary, we've shown that if DDH in $\mathbb{G}_T$ is easy, then DBDH is easy. That's a contradiction because we are told that DBDH is hard for this bilinear map. Therefore, DDH in $\mathbb{G}_T$ is actually hard.

□

6.

Corollary 1.5. *The following problems are also hard: discrete log in* $\mathbb{G}$*, CDH in* $\mathbb{G}_T$ *and discrete log in* $\mathbb{G}_T$*.*

Proof sketch:

(a) For any group, DDH is hard $\implies$ CDH is hard $\implies$ discrete log is hard.

(b) For group $\mathbb{G}_T$, we know that DDH is hard, so CDH and discrete log are also hard.

(c) For group $\mathbb{G}$, we know that CDH is hard, so discrete log is also hard.

□

2 Bounded Collusion Identity-Based Encryption

In lecture 18, we used a bilinear map to construct IBE (identity-based encryption). Here, we will use DDH and a random oracle $H : \mathbb{Z}_q \to \mathbb{Z}_q$ to construct a weaker version of IBE that is secure if the attacker only receives a single $\mathsf{sk}_{\mathsf{ID}}$.

A random oracle is a truly random function that all parties have query access to. In this problem, H is sampled uniformly at random from all functions mapping $\mathbb{Z}_q \to \mathbb{Z}_q$. Random oracles are idealized objects, and they don't exist in the real world. In practice, we replace random oracles with sufficiently complex hash functions, such as SHA-256.

Let the IBE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be constructed as follows:

1. $\mathsf{Setup}(1^n)$:
   (a) Sample the parameters of a cyclic group $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$. Let $\mathsf{pp} = (\mathbb{G}, q, g)$.
   (b) Sample $a, b \leftarrow \mathbb{Z}_q$ independently. Compute $h_0 = g^a$ and $h_1 = g^b$.
   (c) Output $\mathsf{mpk} = (\mathsf{pp}, h_0, h_1)$ and $\mathsf{msk} = (\mathsf{pp}, a, b)$.
2. $\mathsf{KeyGen}(\mathsf{msk}, \mathsf{ID})$:
   (a) Let $\mathsf{ID} \in \mathbb{Z}_q$.
   (b) Compute $r = H(\mathsf{ID})$ and $s = a \cdot r + b \mod q$.
   (c) Output $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{ID}, s)$.
3. $\mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, m)$:
   (a) Let $m \in \mathbb{G}$.
   (b) Compute $r = H(\mathsf{ID})$.
   (c) Sample $y \leftarrow \mathbb{Z}_q$.
   (d) Output $\mathsf{ct} = (g^y, h_0^{y \cdot r} \cdot h_1^y \cdot m)$.
4. $\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{ct})$: TBD

It is implied that all functions can make queries to H.

Questions:

1. Fill in $\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{ct})$, and prove that any valid ciphertext will be decrypted correctly.

   Solution

   $\mathsf{Dec}(\mathsf{sk}_{\mathsf{ID}}, \mathsf{ct})$:
   (a) Parse ct as $\mathsf{ct} = (c_0, c_1)$.
   (b) Compute $r = H(\mathsf{ID})$ and $s = a \cdot r + b \mod q$.
   (c) Compute and output $m = c_0^{-s} \cdot c_1$

Any valid ciphertext will be decrypted correctly because:

$$\begin{aligned}\mathsf{Dec}\big[\mathsf{sk}_{\mathsf{ID}}, \mathsf{Enc}(\mathsf{mpk}, \mathsf{ID}, m)\big] &= c_0^{-s} \cdot c_1 \\ &= g^{-yar-yb} \cdot h_0^{yr} \cdot h_1^{y} \cdot m \\ &= g^{-yar-yb} \cdot g^{yar} \cdot g^{yb} \cdot m \\ &= m\end{aligned}$$

□

2. Show that Π is not a CPA-secure IBE scheme.

Solution

(a) The adversary queries $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ on two different ID's: They obtain

$$(\mathsf{ID}_1, s_1) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{ID}_1)$$
$$(\mathsf{ID}_2, s_2) \leftarrow \mathsf{KeyGen}(\mathsf{msk}, \mathsf{ID}_2)$$

(b) The adversary computes $r_1 = H(\mathsf{ID}_1)$ and $r_2 = H(\mathsf{ID}_2)$ and sets up the following linear system:

$$\begin{cases} s_1 = r_1 \cdot a + b \mod q \\ s_2 = r_2 \cdot a + b \mod q \end{cases}$$

The unknown variables are (a, b). If $r_1 \neq r_2$ (which occurs with probability $1 - \frac{1}{q} = 1 - \mathsf{negl}(n)$), this system is full-rank.

(c) The adversary solves the system for (a, b).

(d) Now the adversary knows $\mathsf{msk} = (\mathsf{pp}, a, b)$, so they can decrypt any ciphertext and break CPA security.

□

It turns out that any adversary that breaks the CPA-security of this IBE scheme needs to make at least 2 queries to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$. This IBE scheme is CPA-secure against any adversary that never makes more than 1 query to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$.