Lecture 16: Public-key Encryption and Collision-Resistant Hash Functions

Public-key Encryption

- The receiver broadcasts $\mathrm{pk} = g^x$, where $x \xleftarrow{\$} \{0, \dots, |G| - 1\}$
- To send a message m, the sender sends the cipher text $(g^y, m \cdot \mathrm{pk}^y)$, where $y \xleftarrow{\$} \{0, \dots, |G| - 1\}$

Think: Should y be reused if the same sender wants to encrypt a new message m' or the same message m again?
Security Proof Intuition: $(g, g^x, g^y, g^{xy}) \approx^{(c)} (g, g^x, g^y, g^z)$ implies that the mask g^{xy} used in the encryption looks like a random mask g^z

Multiple Messages

- To send a second message m' the (possibly, new) sender sends $(g^{y'}, m' \cdot \mathrm{pk}^{y'})$, where $y' \xleftarrow{\$} \{0, \ldots, |G| - 1\}$

Security Proof Intuition: We have to show that $(g, g^x, g^y, g^{xy}, g^{y'}, g^{xy'}) \approx^{(c)} (g, g^x, g^y, g^z, g^{y'}, g^{z'})$. Do we need a new assumption or will DDH suffice?

DDH-based Proof of Security

Consider the following hybrids:

- $H^{(0)}$: $(g, g^x, g^y, g^{xy}, g^{y'}, g^{xy'})$
- $H^{(1)}$: $(g, g^x, g^y, g^z, g^{y'}, g^{xy'})$
- $H^{(2)}$: $(g, g^x, g^y, g^{xy}, g^{y'}, g^{z'})$

We prove that $H^{(0)} \approx^{(c)} H^{(1)}$ and $H^{(1)} \approx^{(c)} H^{(2)}$ using DDH. We shall show the first implication: DDH $\implies$ $H^{(0)} \approx^{(c)} H^{(1)}$. Second implication is left as an exercise.

Proof

- Suppose an efficient adversary $\mathcal{A}^*$ can distinguish the distribution $(g, g^x, g^y, g^{xy}, g^{y'}, g^{xy'})$ from the distribution $(g, g^x, g^y, g^z, g^{y'}, g^{z'})$
- Consider the algorithm $\widetilde{\mathcal{A}}$ that can distinguish (g, g^x, g^y, g^{xk}) from (g, g^x, g^y, g^z)
  - On input $(g, \alpha, \beta, \gamma)$, sample $y' \xleftarrow{\$} \{0, \dots, |G| - 1\}$
  - Output $\mathcal{A}^*(g, \alpha, \beta, \gamma, g^{y'}, \alpha^{y'})$
- Prove: If $\mathcal{A}^*$ distinguishes its two distributions with advantage ε then $\widetilde{\mathcal{A}}$ distinguishes its two distributions with advantage ε

Collision Resistance Hash Function

A family of functions $\mathcal{H} = \{h^{(1)}, \dots, h^{(k)}\}$ is called a collision resistant hash function family, if:

- For all $i \in \{1, \dots, k\}$ the function $h^{(i)} \colon D \to R$ and $|D| > |R|$
- The advantage of any efficient adversary $\mathcal{A}$ in the following game with the honest challenger is negligible
    - The honest challenger $\mathcal{H}$ samples $i \xleftarrow{\$} \{1, \dots, k\}$ and sends $h^{(i)}$ to the adversary $\mathcal{A}$
    - The adversary $\mathcal{A}$ replies back with (x, x')
    - The honest challenger outputs $z = 1$ if and only if $x \neq x'$ and $h^{(i)}(x) = h^{(i)}(x')$

Construction based on DL

Let G be a multiplicative group with generator g where Discrete Log is believed to be hard. For $y \in G$, define $h^{(y)}(b, x) = y^b g^x$, where $b \in \{0, 1\}$ and $x \in \{0, \ldots, |G| - 1\}$. Then we will show that $\mathcal{H} = \{h^{(y)} \colon y \in G\}$ is a CRHF.

Proof of Security

- Suppose $\mathcal{A}^*$ breaks the CRHF security property
- Suppose the $\mathcal{A}^*$ replies with distinct (b, x) and (b', x') as a collision

Claim

It is impossible to have $b = b'$

- If possible let $b = b'$
- Then it must be the case that $x \neq x'$
- Then we have $y^b g^x = y^{b'} g^{x'} \iff g^x = g^{x'} \iff x = x'$, a contradiction

Proof Continued

- So, in a successful collision it must be the case that $b \neq b'$
- Without loss of generality, assume that $b = 0$ and $b' = 1$
- So, we have $g^x = yg^{x'} \iff g^{(x-x')} = y$
- So, $x - x'$ is the discrete log of y, when $b = 0$
- Consider the following adversary $\widetilde{\mathcal{A}}$ against discrete log:
  - On input y send $h^{(y)}$ to $\mathcal{A}^*$
  - Receive (b, x) and (b', x') in reply
  - If $(b, x) \neq (b', x')$ and $h^{(y)}(b, x) = h^{(y)}(b', x')$ then:
    - If $b = 0$, return $(x - x') \mod |G|$
    - If $b' = 0$, return $(x' - x) \mod |G|$
  - Else return 0 (i.e., the algorithm could not find the discrete log)
- What is the probability that $\widetilde{\mathcal{A}}$ outputs the correct discrete logarithm?