Cryptographic Hash Functions & Message Authentication Codes

2023.10.18

Jiageng Chen
Tutor: Zheng Chen

Security Model (IND-CCA)

Setup: The challenger chooses a random key K.

Phase 1: The adversary can choose any M for encryption query and choose any CT for decryption query.

Challenge: The adversary can choose any two different messages M_0 and M_1 . The challenger chooses a random c and computes the challenge ciphertext CT^*=Enc(M_c , K), which is given to the adversary.

Phase 2: The adversary can choose any M for encryption query and choose any CT different from CT^* for decryption query.

Guess: The adversary returns the guess c' and wins if $c' = c$.

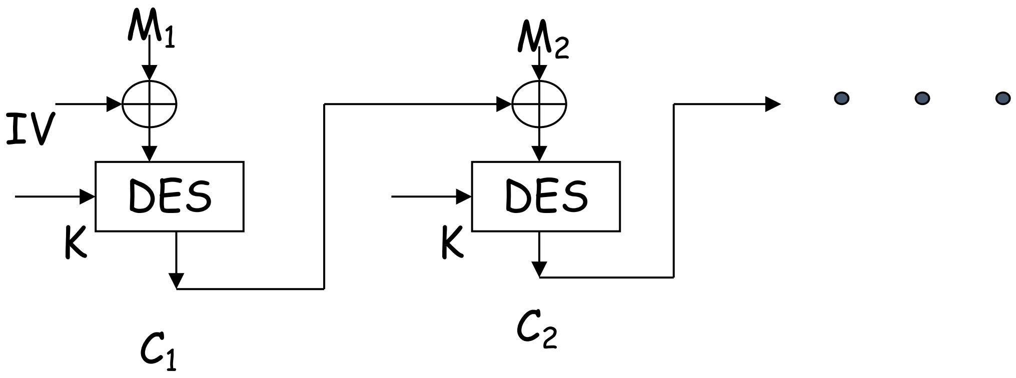
We say that the encryption is secure if every P.P.T adversary can only win the game with **negligible** probability defined as

$$Adv_A^{IND-CCA} = Pr[c' = c] - \frac{1}{2}$$

Q1: Cipher Block Chaining (CBC), n=2

An **initiation vector (IV = C_0)** is used for randomization

- CT = (C_0 C_1 C_2), where C_1 = DES(K, $M_1 \oplus C_0$), C_2 = DES(K, $M_2 \oplus C_1$)



The ciphrtext for (M_1, M_2) is CT = (C_0 C_1 C_2)

Suppose that the above encryption is IND-CPA secure. Prove that it is not IND-CCA secure.

Q1: Cipher Block Chaining (CBC), n=2

An **initiation vector (IV = C_0)** is used for randomization

- $CT = (C_0, C_1, C_2)$, where $C_1 = DES(K, M_1 \oplus C_0)$, $C_2 = DES(K, M_2 \oplus C_1)$

The ciphertext for (M_1, M_2) is $CT = (C_0, C_1, C_2)$

Suppose that the above encryption is IND-CPA secure. Prove that it is not IND-CCA secure.

The adversary sends (M_0, M_1) for challenge, where

$$M_0 = (M_{00}, M'_{00}),\ M_1 = (M_{11}, M'_{11}),\ M_{00} \neq M_{11}$$

Let the challenge ciphertext be $CT^* = (C_0^*, C_1^*, C_2^*)$. Then, the adversary chooses a random C_3 and query the decryption on $= (C_0^*, C_1^*, C_2^*)$. The challenger should return a decryption result (M_a, M_b), where M_a must be M_{00} or M_{11}. Then, CCA is broken.

Hash Functions (Security Definitions)

- A hash function (algorithm) is denoted by h: $\{0, 1\}^* \rightarrow \{0, 1\}^n$

Adversary's Target: Given a hash function h, find $x \neq x'$ such that h(x)=h(x').

1.Collision Resistance:

Given a hash function h: $\mathcal{X} \rightarrow \mathcal{Y}$, there is no efficient mechanism (P.P.T. adversary) to find $x, x' \in$X such that $x' \neq x$ and h(x') = h(x).

3.Pre-Image Resistance:

Given a hash function h:$\mathcal{X} \rightarrow \mathcal{Y}$ and y $\in \mathcal{Y}$, there is no efficient mechanism (P.P.T. adversary)to find $x \in$ X such that y = h(x).

Q2： Help Alice

Bob wants to prove that collision *resistance* ***doesn’t*** **imply** pre-image *resistance.*

He constructs the following hash function h using $g(x)$: $\{0,1\}^* \rightarrow \{0,1\}^n$, which is a collision-resistant hash function. Bob claims that he can prove “*doesn’t* imply pre-image ” with this example, but Alice cannot understand.

How to convince Alice?

$$h(x) = \begin{cases} 1 \| x & \text{if } |x| = n \\ 0 \| g(x) & \text{otherwise} \end{cases}$$

Q2： Help Alice

Bob wants to prove that collision *resistance doesn't* imply pre-image *resistance.*

How to convince Alice?

$$h(x) = \begin{cases} 1 \,\|\, x & \text{if } |x| = n \\ 0 \,\|\, g(x) & \text{otherwise} \end{cases}$$

We need to:

1. Prove that this is a collision-resistant hash function. Suppose we can find x and x'.

- x and x' have the same length n. Then, h(x) = 1|| x, h(x') = 1|| x' (not equal)
- x has length n and x' has length ≠ n, then, the output cannot be equal.
- x and x' are length ≠ n, we cannot have h(x) = h(x'). Otherwise, g(x) = g(x').

2. Prove that this is not a pre-image resistant hash function. This is easy because when given y =1|| x, it is easy to compute its pre-image x.