LG.1
3/8/2021

Today

- Recap: Definition of MACs (Message Authentication Codes)
- MACs + CPA secure encryption ⟹ CCA secure encryption
- Constructions of MACs
- Hash functions

Def: A MAC consists of a PPT alg MAC that takes as input a key K and a msg M and outputs $MAC(K,M)$. ← authenticates the msg M; often called "tag"

If MAC is randomized it is also associated w. a verification procedure that given (K, M, tag) outputs 0/1. ← We assume that MAC is deterministic and K is uniform (o.w need Gen alg)

A MAC is secure against adaptive chosen msg attacks if

$\forall$ PPT A $\exists$ negl function $\mu: \mathbb{N} \to [0,1]$ s.t. $\forall n \in \mathbb{N}$

$$\Pr\left[A^{MAC(K,\cdot)}(1^n) = (M, tag) : M \text{ was not queried } \& \; tag = MAC(K,M) \right] \leq \mu(n)$$

$K \leftarrow \{0,1\}^n$

if MAC is randomized then $Verify(K, M, tag) = 1$

- MACs are like digital signatures, where the former is in the secret key setting and the latter is in the public key setting (stay tuned...)

- tag cannot be too short, o.w. it can be guessed (typically 128 bits)

Thm: CPA secure encryption + secure MAC

$\Downarrow$

CCA secure encryption

Recall: An encryption scheme (Gen, Enc, Dec) is CPA (CCA) secure if $\forall$ PPT adv $\exists$ negl function $\mu: \mathbb{N} \to [0,1]$ s.t. $\forall n \in \mathbb{N}$ $\forall m_0, m_1 \in \mathcal{M}$

$$Pr\left[A^{Enc(K,\cdot)\ Dec(K,\cdot)}\left(Enc(K, m_b)\right) = b \right] \leq \frac{1}{2} + \mu(n)$$

Given only in CCA, in which case A cannot query decryption oracle with challenge CT

"Proof": Append to the CPA secure CT a MAC:

$$\underbrace{Enc(K, M)}_{CT} \quad \underbrace{MAC(k_2, CT)}_{tag}$$

Formally, given a CPA secure enc. scheme (Gen, Enc, Dec) & given MAC, define the following CCA secure scheme (Gen', Enc', Dec'):

$Gen'(1^n)$: Generate $K_1 \leftarrow Gen(1^n)$, $K_2 \leftarrow \{0,1\}^n$

output (K_1, K_2)

$Enc'((K_1,K_2), M)$: Compute $CT \leftarrow Enc(K_1, M)$

$tag \leftarrow MAC(K_2, CT)$

Output $CT' = (CT, tag)$

$Dec'((K_1,K_2), (CT,tag))$: Check if $tag = MAC(K_2, CT)$

If not output $\perp$.

Otherwise output $M = Dec(K_1, CT)$.

Intuitively, the reason this is CCA secure is that the decryption oracle is useless unless adv. knows a valid tag. But it knows a valid tag only if it got this CT from the enc oracle, in which case the decryption oracle is useless.

Hash Functions:

Def: A **hash family** is a family of functions $\mathcal{H} = \{h_s\}$ where

$h_s : \{0,1\}^* \to \{0,1\}^d$. $\forall n \in \mathbb{N}\ \forall s \in \{0,1\}^n$ $h_s : \{0,1\}^* \to \{0,1\}^{d=d(n)}$

($\{0,1\}^*$: arbitrary length; typically think of input space as $\{0,1\}^{>n}$ and output space $d(n) < n$)

Given s, x one can efficiently compute $h_s(x)$

s is called the seed, and is public.

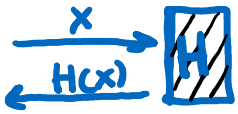
$h_s(x)$ is called the hash value.

Ideal Hash Function

Truly random function $H : \{0,1\}^* \to \{0,1\}^d$

A truly random function is not efficiently computable

so we give all parties oracle access to this ideal hash function



This idealized (theoretical) model is called the **Random Oracle Model** (ROM)

Many cryptographic primitives use hash functions, and these primitives are often proved to be secure in the ROM.

In practice the hash family is usually implemented with a single hash function

SHA 256 — secure Hash Algorithm

Desired properties

① One-Way: $\forall$ PPT $\mathcal{A}$ $\forall n \in \mathbb{N}$

H can consist of a single function per output length d.

$$\Pr_{s,x \leftarrow \{0,1\}^n}[\mathcal{A}(s, h_s(x)) \in h_s^{-1}(h_s(x))] = \text{negl}(d)$$

ROM: Time to invert $\Theta(2^d)$ ✓

② Collision-Resistant: $\forall$ PPT $\mathcal{A}$ $\forall n \in \mathbb{N}$

$$\Pr_{s \leftarrow \{0,1\}^n}[\mathcal{A}(s) = (x, x') \text{ s.t. } x \neq x' \text{ \& } h_s(x) = h_s(x')] = \text{negl}(d)$$

ROM: Time to find collisions $\Theta(2^{d/2})$ ✓ ← Birthday paradox: choose randomly $x_1, x_2, \ldots$ until a collision is found.

③ Target Collision-Resistant: $\forall$ PPT $\mathcal{A}$ $\forall n \in \mathbb{N}$ $\forall x \in \{0,1\}^n$

$$\Pr_{s}[\mathcal{A}(s, x) = x' \text{ s.t. } x' \neq x \text{ \& } h_s(x) = h_s(x')] = \text{negl}(d)$$

ROM: Time to find a target collision $\Theta(2^d)$ ✓

④ Pseudo Randomness: A PPT adversary cannot distinguish between oracle access to h_s & a truly random H.

Formally, $\forall$ PPT $\mathcal{A}$ $\forall n \in \mathbb{N}$

$$\left| \Pr_{s \leftarrow \{0,1\}^n}[\mathcal{A}^{h_s}(1^n) = 1] - \Pr[\mathcal{A}^{H}(1^n) = 1] \right| = \text{negl}(d)$$

where $H : \{0,1\}^n \to \{0,1\}^d$ is a random function.

ROM: ✓