CS 442
Introduction to Cryptography

Lecture 17: Example Problems on MACs, CPA, CCA and AE

Instructor: Aarushi Goel

Spring 2026

Agenda

* Example problems on
  - MACs
  - CPA and CoA-secure encryption
  - CCA-secure encryption
  - AE using encrypt-then-MAC

* HW3 is due on April 5

Many-time message Authentication game:

Adversary

Challenger

Key Gen $\rightarrow$ K

$m_i \in M$

polynomial number of queries

σ_i

Sign $(m_i, K) \rightarrow \sigma_i$

(m, σ)

Adv wins if:

① $\forall i$, $m_i \neq m$

② Verify $(m, \sigma, K) = 1$

* This game is also called chosen message attack. (CMA)
* These MACs are also called CMA-secure MACs.

MACs # 1

Q Let $\{f_K\}_K$ be a family of PRFs, where $f_K : \{0,1\}^n \to \{0,1\}^n$ and $K \in \{0,1\}^n$.
Determine whether the following MAC scheme satisfies many time unforgeability?

- KeyGen $\to K$: Sample $K \xleftarrow{\$} \{0,1\}^n$
- Sign $(m, K) \to \sigma$: Let $m = m_1 \| m_2$ with $|m_1| = |m_2| = n$.

  Output $\sigma = f_K(m_1) \| f_K(f_K(m_2))$.

A NO! Consider the following adversary:

Adv

$m = m_1 \| m_2$

$\sigma = \sigma_1 \| \sigma_2$

$m' = m_2 \| m_1$

$\sigma' = \sigma_3 \| \sigma_4$

$m'' = m_1 \| m_1$, $\sigma'' = \sigma_1 \| \sigma_4$

Ch

We know that $\sigma = \sigma_1 || \sigma_2$ is such that $\sigma_1 = f_K(m_1)$, $\sigma_2 = f_K(f_K(m_2))$

similarly $\sigma' = \sigma_3 || \sigma_4$ is such that $\sigma_3 = f_K(m_2)$, $\sigma_4 = f_K(f_K(m_1))$

a valid MAC on message $m'' = m_1 || m_1$ is $\sigma'' = f_K(m_1) || f_K(f_K(m_1))$

$= \sigma_1 || \sigma_4$.

Hence this adversary succeeds in forging a valid MAC on a new message.

$\Rightarrow$ This MAC scheme does not achieve unforgeability.

MACs #2

Q Let $\{f_K\}_K$ be a family of PRFs, where $f_K : \{0,1\}^n \to \{0,1\}^n$ and $K \in \{0,1\}^n$.
Determine whether the following MAC scheme satisfy unforgeability?

- KeyGen $\to K$: Sample $K \xleftarrow{\$} \{0,1\}^n$
- Sign $(m, K) \to \sigma$: Let $m = m_1, \ldots, m_\ell$, where each $m_i \in \{0,1\}^{n/2}$. Sample $r \xleftarrow{\$} \{0,1\}^n$.
  compute $t = f_K(r) \oplus f_K(\langle 1 \rangle, m_1) \oplus \cdots \oplus f_K(\langle \ell \rangle, m_\ell)$. Set $\sigma = (r, t)$.

A Nope! Consider the following adversary:

Adv

Ch

$m = m_1, \ldots, m_\ell$ →

← $\sigma = (r, t)$

$m', \sigma' = (r_1 \| m_{r_1}, t)$ →

Let $r = r_1 \| r_2$

$\forall i \in [\ell]$, if $i \neq r_1$,
set $m_i' = m_i$
else set $m_{r_1}' = r_2$

set $m' = m_1', \ldots, m_\ell'$

* We know that $\sigma = (r, t)$ is such that

$$\sigma = f_K(r) \oplus f_K(\langle 1 \rangle \| m_1) \oplus \dots \oplus f_K(\langle \ell \rangle \| m_\ell)$$

$$= f_K(r_1 \| r_2) \oplus f_K(\langle 1 \rangle \| m_1) \oplus \dots \oplus f_K(r_1 \| m_{r_1}) \oplus \dots \oplus f_K(\langle \ell \rangle \| m_\ell)$$

$$= f_K(r_1 \| m_{r_1}) \oplus f_K(\langle 1 \rangle \| m_1) \oplus \dots \oplus f_K(r_1 \| r_2) \oplus \dots \oplus f_K(\langle \ell \rangle \| m_\ell)$$

* This is a valid MAC on message $m_1, \dots, r_2, \dots, m_\ell$, using randomness $r_1 \| m_{r_1}$

* Hence this adversary succeeds in forging a valid MAC on a new message.

$\Rightarrow$ This MAC scheme does not achieve unforgeability.

MACs #3

Q Consider the following alternate definition of unforgeability for MACs. Let's call this *strong unforgeability*.

Adversary

Challenger

KeyGen → K

$m_i \in M$

polynomial number of queries

Sign $(m_i, K) \rightarrow \sigma_i$

σ_i

(m, σ)

Adv wins if:

① $\forall i$, $(m_i, \sigma_i) \neq (m, \sigma)$

② Verify $(m, \sigma, K) = 1$

1. Is a MAC scheme that satisfies unforgeability also guaranteed to satisfy strong unforgeability?

No! unforgeability implies that no PPT adversary can find a valid (m, σ) pair, such that the adversary never previously queried to obtain a MAC on m. This does not rule out the possibility of a adversary forging a new MAC on some previously queried message. While this is not sufficient to break unforgeability, it is a valid attack on strong unforgeability.

2. Is a MAC scheme that satisfies strong unforgeability also guaranteed to satisfy unforgeability?

Yes! This is straightforward.

Chosen - Plaintext - Attack (CPA - Security)

Adversary

Challenger

KeyGen → K

query polynomial no. of times { $m \rightarrow$, $\leftarrow c$ } Enc(K, .)

$m_0, m_1 \in \mathcal{M}$ →

← $c_b = Enc(K, m_b)$

can include previously queried messages

$b \xleftarrow{\$} \{0,1\}$

query polynomial no. of times { $m \rightarrow$, $\leftarrow c$ } Enc(K, .)

b' →

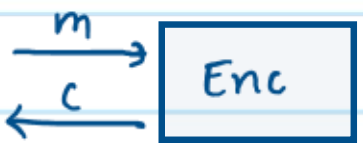
CPA-Security #1

Q Let $\{f_K\}_K$ be a family of PRFs, where $f_K : \{0,1\}^n \to \{0,1\}^n$ and $K \in \{0,1\}^n$. Determine whether the following is a CPA secure encryption.

- KeyGen $\to K$: Sample $K \xleftarrow{\$} \{0,1\}^n$
- Enc $(m, K) \to c$: Output $c = m \oplus f_K(0^n)$.
- Dec $(c, K) \to m$: Output $m = c \oplus f_K(0^n)$.

A No! Consider the following adversary:

Adv



Ch

(m_0, m_1)

$b \xleftarrow{\$} \{0,1\}$

c_b

b'

if $c_b = m_0 \oplus c \oplus m$,
set $b' = 0$
else set $b' = 1$

It is easy to see that this adversary will be able to successfully guess $b'=b$ with probability 1.

$\Rightarrow$ Hence this scheme is not CPA-secure.

Q Is it a COA-secure encryption?

A Yes! Why?

This scheme is very similar to the pseudorandom one-time pad encryption.

Chosen-Ciphertext-Attack (CPA-Security)

Adversary

Challenger

query polynomial no. of times

m → Enc(K,.)

← c

c → Dec(K,.)

← m

KeyGen → K

$m_0, m_1 \in \mathcal{M}$ →

← $c_b = Enc(K, m_b)$

$b \xleftarrow{\$} \{0,1\}$

query polynomial no. of times

m → Enc(K,.)

← c

c → Dec(K,.)

← m

cannot include c_b.

b' →

CCA-Security #1

Q Let $\{f_K\}_K$ be a family of PRFs, where $f_K : \{0,1\}^n \to \{0,1\}^n$ and $K \in \{0,1\}^n$. Determine whether the following is a CCA-secure encryption.

- KeyGen $\to K$: Sample $K \xleftarrow{\$} \{0,1\}^n$
- Enc $(m, K) \to c$: Sample $r \xleftarrow{\$} \{0,1\}^n$. output $c = (r, m \oplus f_K(r))$
- Dec $(c, K) \to m$: Parse $c = (c_1, c_2)$. output $m = c_2 \oplus f_K(c_1)$

A NO! Consider the following adversary:

Adv

Ch

(m_0, m_1) →

← c_b

$m \xleftarrow{\$} \{0,1\}^n$

Let $c_b = (c_1, c_2)$

$(c_1, m \oplus c_2)$ →

← m'

Dec

if $m' \oplus m = m_0$, set $b' = 0$

else set $b' = 1$

b' →

$b \xleftarrow{\$} \{0,1\}$

We know that $c_b = \underbrace{(r}_{c_1}, \underbrace{m_b \oplus f_K(r))}_{c_2}$

$Dec((r, m \oplus m_b \oplus f_K(r)), K) \rightarrow m \oplus m_b$

It is now easy to see that this adversary will be able to successfully guess $b' = b$ with probability 1.

$\Rightarrow$ Hence this scheme is not CCA-secure.

Authenticated Encryption

Definition: An authenticated encryption scheme must be CCA-secure and unforgeable. → defined similarly to MACs.

Construction: CPA secure encryption + CMA-secure MAC

↓

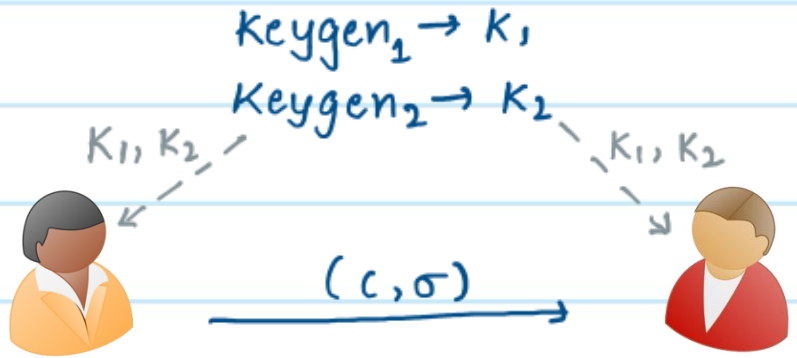
Authenticated Encryption
(CCA-secure & unforgeable)

Q How should we combine CPA-secure encryption & CMA-secure MACs?

Encrypt-then-MAC

Let $(\text{Keygen}_1, \text{Enc}, \text{Dec})$ be a CPA-secure Encryption Scheme.

Let $(\text{Keygen}_2, \text{Sign}, \text{Verify})$ be a many-time strongly unforgeable MAC scheme



$c = \text{Enc}(k_1, m)$

$\sigma = \text{Sign}(k_2, c)$

$m = \text{Dec}(k_1, c)$

$b = \text{Verify}(k_2, c, \sigma)$

if $b = 1$, accept m, else abort.

Q Is this a good strategy for designing an authenticated encryption scheme? Yes!

We need to show that this scheme is unforgeable and CCA secure.

* Unforgeability: follows from strong unforgeability of the underlying MAC scheme

* CCA-Security: Recall that in the CCA-security game, the adversary gets access to both a decryption oracle & an encryption oracle. From CPA-security of the underlying encryption scheme, we know that the encryption oracle is not useful for the adversary in breaking security of this scheme. Now observe that the strong unforgeability of the MAC scheme also pretty much renders the decryption oracle useless. This is because every ciphertext (c,σ) that the adversary might potentially query to the decryption oracle, it either already knows the decryption or will be an invalid (c,σ) pair because of strong unforgeability of σ.

Exercise: Think where does this argument break if we use a MAC scheme that is only unforgeable, not strongly unforgeable.