CS 442
Introduction to Cryptography

Lecture 16: Authenticated Encryption

Instructor: Aarushi Goel

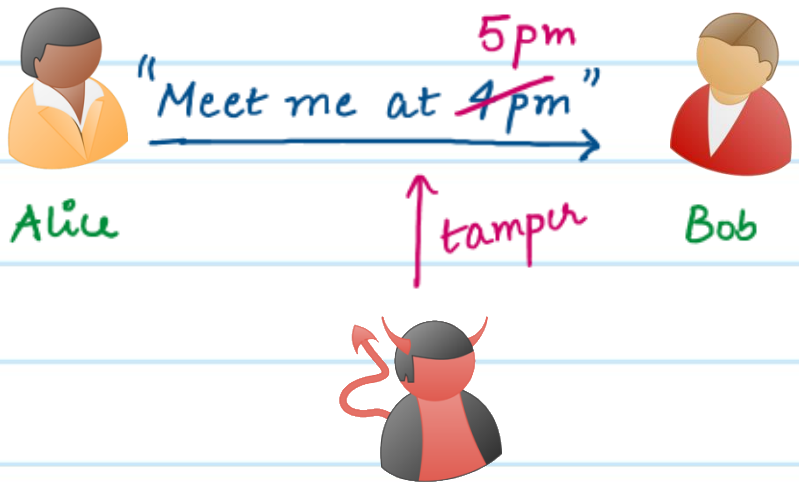
Spring 2026

Agenda

* Construction of MACs
* CCA - secure Encryption
* Authenticated Encryption.

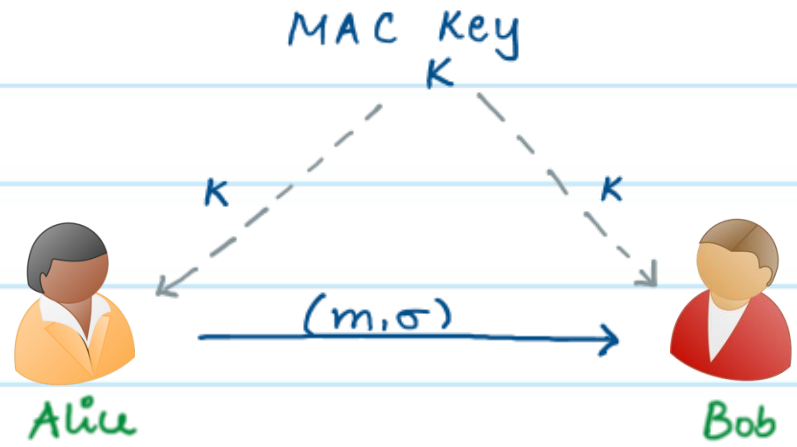
Message Integrity

* Encryption prevents an evesdropper from learning private messages exchanged between Alice and Bob over a public communication channel.
* Encryption does not prevent against stronger adversaries who can modify the message on the way, i.e., it does not help with message integrity.
* Message authentication codes (MACs) that are digital analogues of of physical signatures help ensure message integrity.



Message Authentication Codes (MACs)

* If the adversary tries of change $m \rightarrow m'$, they will also have to find a corresponding σ', such that $Verify(m, \sigma, K) \rightarrow yes$. This should be hard if they don't have K.



$Sign(m, K) \rightarrow \sigma$

$Verify(m, \sigma, K) \rightarrow yes/no$

Adversary

(Many-time)

Message Authentication Codes (MACs)

Definition: A one-time message athentication code scheme with message space $\mathcal{M}$, key space $\mathcal{K}$ and signature/tag space $\mathcal{S}$, comprises of the following algorithms:

* $\text{Keygen} \rightarrow K$: This algorithm samples a key $K \xleftarrow{\$} \mathcal{K}$.
* $\text{Sign}(m, K) \rightarrow \sigma$: On input a message $m \in \mathcal{M}$ & key $K \in \mathcal{K}$, it outputs a signature $\sigma \in \mathcal{S}$.
* $\text{Verify}(m, \sigma, K) \rightarrow b$: On input message $m \in \mathcal{M}$, key $K \in \mathcal{K}$ and signature $\sigma \in \mathcal{S}$, it outputs a bit $b \in \{0,1\}$ (where 1 means yes, 0 means no).

These algorithms must satisfy the following:

→ Correctness: $\forall K \in \mathcal{K}$, $m \in \mathcal{M}$, it holds that $\Pr[\text{Verify}(m, \text{Sign}(m, K), K) = 1] = 1$

→ Unforgeability: For all PPT adversaries, there exists a negligible function $\nu(\cdot)$, such that the following holds in the game below:

$$\Pr[\text{Adv wins}] \leq \nu(|K|)$$

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.

Construction of Many-time MAC Scheme

Let $\{f_K\}$ be a family of PRFs.

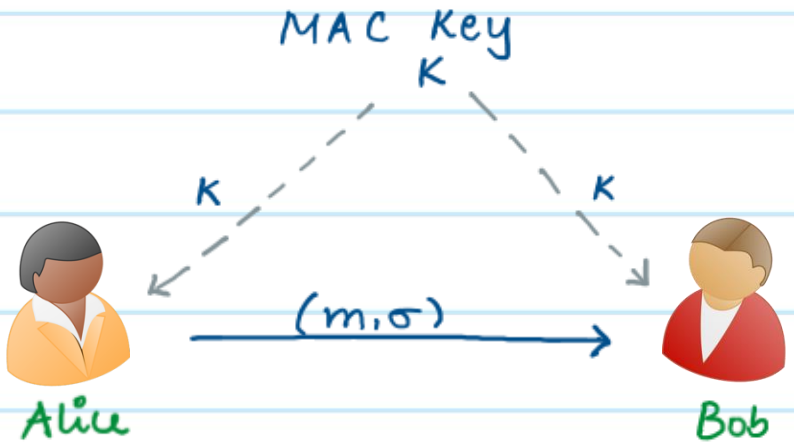
* Keygen: Sample a PRF Key K.
* Sign (m, K) : $\sigma = f_K(m)$
* Verify (m, K) : $b = \begin{cases} 1, & \text{if } \sigma = f_K(m) \\ 0, & \text{otherwise} \end{cases}$

* Correctness : straightforward
* Unforgeability : An adversary who can break unforgeability of this scheme (i.e., win in the chosen message attack (CMA)-game) should be able to distinguish between the outputs of this PRF and the outputs of a random function. But we know that no PPT adversary can do that except with negligible probability. Hence this scheme is unforgeable.

Message Authentication Codes (MACs)

* MACs only help with message integrity, they don't hide the message
* Encryption only hides the message, doesn't help with message integrity.

Q Can we somehow combine encryption and MACs to get both privacy and integrity?



Sign(m,K) → σ

Verify(m,σ,K) → yes/no

Adversary

Authenticated Encryption

* An encryption scheme that can both hide the message and cannot be tampered with.

Secrecy + Integrity

~~CPA-security~~ unforgeability

Not sufficient

→ Recall that we came up with the definition of CPA-security to only account for adversaries who can only eavesdrop.

→ Now we are dealing with stronger adversaries who can modify/maul the ciphertexts.

→ Such adversaries might be able to violate privacy of CPA-secure encryption

How? e.g. Read about padding oracle attacks (chapter 3.7.2 in Introduction to Modern Cryptography) on PRF-based encryption in CBC mode.

If an adversary is able to modify the ciphertexts and learn whether the receiver (Bob) was able to decrypt the modified ciphertexts or not, they may be able to recover the original message. — This is the main idea behind padding oracle attacks

* Therefore CPA-security is not sufficient for guaranteeing secrecy in authenticated encryption.
* We need a stronger notion of secrecy — CCA-security
  (chosen ciphertext attack)

Authenticated Encryption :

Secrecy + Integrity
↙ ↘
CCA-security unforgeability

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' →

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' →

Chosen-Ciphertext Attack (CCA-Security)

* In addition to the encryption oracles in CCA-security, we also give the adversary access to a decryption oracle.
* The adversary can use this decryption oracle to decrypt any polynomial number of ciphertexts of its choice, except the challenge ciphertext itself.
* CCA security intuitively captures non-malleability, i.e., if the adversary tries to modify a given ciphertext, the result is either an *invalid ciphertext* or one that decrypts to a plaintext having no relation to the original one.

Q Does this realistically model any real-world attack? Do real-world adversaries really have such decryption oracles?

1. Depending on the application, they might.
2. Or even if they don't, they might be able to "influence" what gets decrypted and try to learn partial information from that.

   example 1: padding oracle attacks

   example 2: Country A intercepts a conversation between an enemy country B and their armed forces. Country A modifies the encrypted message and tries to deduce the original message based on their behavior.

* CCA-security is the strongest definition to ensure privacy in all such scenarios.

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)

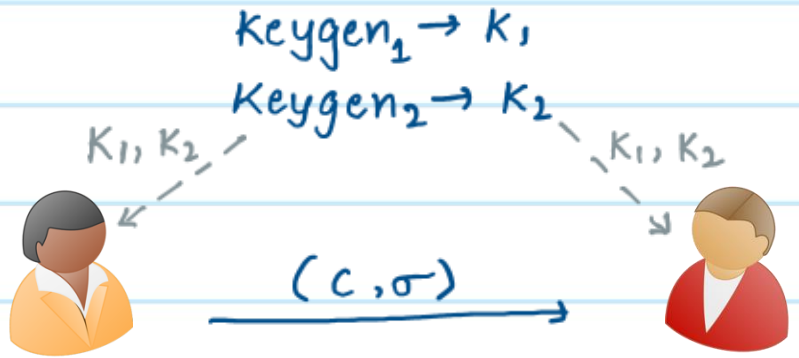
Q How should we combine CPA-secure encryption & CMA-secure MACs?

Attempt 1:

Encrypt - and - 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 MAC scheme



$c = \text{Enc}(K_1, m)$

$\sigma = \text{Sign}(K_2, m)$

$m = \text{Dec}(K_1, c)$

$b = \text{Verify}(K_2, m, \sigma)$

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

Q Is this a good strategy for designing an authenticated encryption scheme? No!

Why?

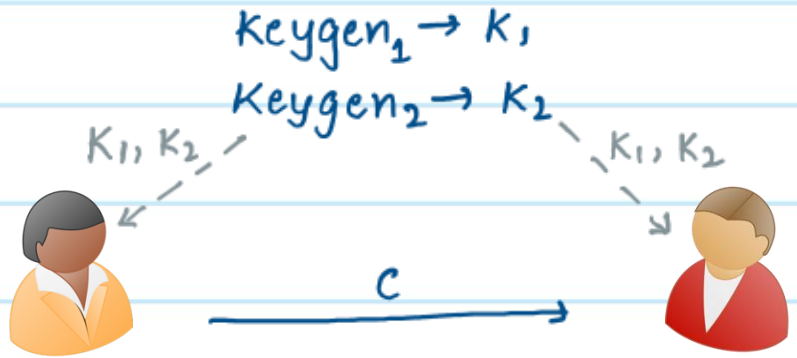
* σ doesn't even try to hide the message.
* It's possible that the MAC scheme reveals the message in the clear.

Attempt 2:

MAC - then - Encrypt

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 MAC scheme



$\sigma = \text{Sign}(K_2, m)$

$c = \text{Enc}(K_1, m \| \sigma)$

$m \| \sigma = \text{Dec}(K_1, c)$

$b = \text{Verify}(K_2, m, \sigma)$

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

Q Is this a good strategy for designing an authenticated encryption scheme? No!

Why?

* c is only a CPA-secure encryption of $m \| \sigma$. It may be susceptible to chosen-ciphertext attacks.
* Eg. There are two-sources of decryption failure here:
    1. c itself may be an ill-formed ciphertext
    2. σ may not be valid MAC for m.

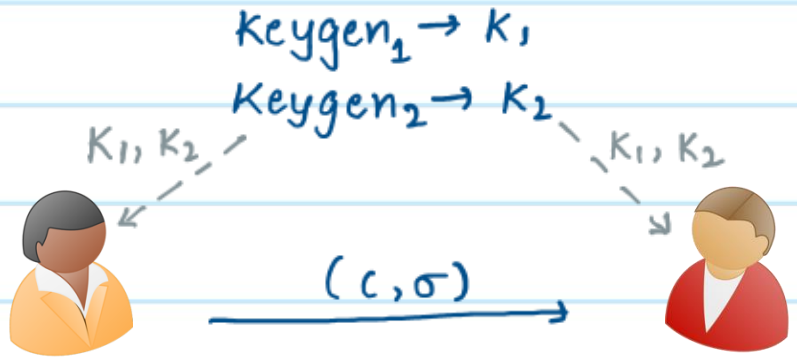
Consider an adversary who can distinguish between these two error messages can launch a padding oracle style attack.
Such attack was successfully carried out on configurations of IPSec that use MAC-then-encrypt.

Attempt 3:

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 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!