Lecture 8: Authenticated Encryption
Plan

→ Review: CPA Security
  - Why CPA is insufficient
  - Authenticated encryption
    * Encrypt then MAC
    * CCA security

→ Recap of symmetric-key primitives
Encryption with a shared secret

\((\text{Enc}, \text{Dec})\)

\[ k \xrightarrow{\text{ct} \leftarrow \text{Enc}(k,m)} m \leftarrow \text{Dec}(k,m) \]
Recap: CPA Security

$Enc : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

**Challenger**

- $k \in \mathcal{K}$
- $M_i (1), M_i (2)$
- $|M_i (1)| = |M_i (2)|$
- $Enc(k, M_i (b))$

**Adv**

- $b' \in \{0, 1\}$
- $b' \neq b$

$Enc$ scheme is CPA secure if no adv can distinguish world $b=0$ from world $b=1$.

**CPA Secure encryption from PRF $F : \mathcal{K} \times [\mathcal{M}] \rightarrow \{0,1\}^*$**

$Enc(k, m) :=$

- Decryption essentially same as encryption.
- $F(k, r_1) F(k, r_2) F(k, r_3) \ldots F(k, r_n)$
- "IV" RANDOMIZED!
CPA Security is not enough.

Example: SSH server using CPA-secure enc

Key points

1) Adv can do lots of damage w/o learning encrypted msg. e.g. msg decrypts to garbage.

2) App-level failures can leak msg.

How could A learn that failure occurred?
* B could reply w/ msg of varying len
* B could throw error
* B could reply in diff time
* B could perform other action
Example: CBC Padding Oracle

* CBC is a “mode of operation” like AES-GCM
* Essentially, deprecated
* Required msg to be padded to multiple of 128 bits/16 bytes

Simplified?

Last block

\[
\text{this is the msg} + 4 \\
\text{Put 4 in the last byte to indicate 4 padding bytes}
\]

In CBC mode, decryptor would
* Decrypt ciphertext
* Check whether padding well formed
* IS not, throw error

With a few queries per byte, can recover msg
First, find end of message.

Next, learn last byte.

**Problem:** Recipient acted on unauthenticated data.
Authenticated encryption ("Gold standard" sec def)

**Syntax:** \((\text{Enc}, \text{Dec})\) as before.

The type sig of decryption routine is now

\[
\text{Dec} : K \times C \rightarrow M \cup \{\bot\}
\]

\(= \text{fail. no msg output}\)

\((\text{Enc}, \text{Dec})\) is AE if:

1) Is CPA-secure and
2) Satisfies "ciphertext integrity"

\[\begin{align*}
\text{Adv can't cook up new valid ciphertexts} \\
\text{Valid ciphertexts}
\end{align*}\]

\(\text{Adv wins if}\)
\(c^* \not\in \{c_1, \ldots, c_n\}\)
and \(\text{Dec}(k, c^*) = \text{reject}\)

Enc scheme has ciphertext integrity if \(V\) eff adversarial

\[Pr[A \text{ wins ctext int. game}] < "negl."\]

**AE Security \(\Rightarrow\) CCA Security.** (strong)

**" \Rightarrow\) Msg integrity**

AE is "gold standard" for enc security.

\(\Rightarrow\) AEAD = AE + associated (auth but not enc) data
Constructing AE schemes

“Encrypt then MAC” → As easy as it sounds

- Works with a “Strong MAC” (Game 9.1 in Boneh-Shoup)
  - Given many \( (m,t) \) pairs on chosen msgs
    hard to cook up a new valid \( (m',t') \) pair.

- Independent keys for both parts (PRF)

- AES-GCM is standard
  GTR mode + GMAC

- ChaCha-Poly1305 is another

To decrypt:
1) Check MAC on \( \texttt{ct} \) \( \text{first} \). Is bad, FAIL.
2) Then decrypt.

(\( \text{Don’t even peek at msg before checking MAC.} \))

Sanity check: Why does enc-then-MAC provide ciphertext integrity?

* To get decryptor to decrypt, must produce new \( (\texttt{ct},\text{tag}) \) pair

* Not possible by MAC security!"
Encrypt-then-MAC is the safe way to combine enc & MAC

* AES-GCM = AES-CTR then GMAC
* Also common = ChaCha20 + Poly1305mac
* Well-designed crypto APIs handle this for you.

It's possible to construct AE directly from $\text{PRP}(\text{AES})$

- OCB mode is one example
- Can be faster than generic encrypt & MAC (+ OCB is!)
- Why don't we use it? Sad story 😞

What do people often mess up?

- Same key for enc & MAC
- MAC doesn't cover whole ctext (e.g. IV)
- Provide data to application before checking MAC on entire ctext
CCA Security

CPA-secure: Adv can see encryption of msg of its choice

What if adv can see decryptions?

Principle of CCA Sec DeS:

- Adv shouldn't be able to dist enc of m₀ from m₁
- Even if it can ask for enc of many msgs of its choice
  AND
- Can ask for decryption of any ctext except answers to prior enc queries.
CCA: Definition

\[ \text{Chal}(b) = \begin{cases} \text{Encryption queries} & \text{if } b = 0 \\ \text{Decryption queries} & \text{if } b = 1 \end{cases} \]

Let \( W_0 \) be the event that \( \text{Adv} \) outputs 1 in world \( b \in \{0,1\} \).

\[ W_0 \]

\[ \text{CCA Security Def.} \]

(\( \text{Enc}, \text{Dec} \)) is CCA secure if \( \forall e \exists \text{adv} A \exists \epsilon \text{ negl} \) s.t.

\[ |Pr[W_0] - Pr[W_1]| \leq \text{negl} \]

\( \text{Adv} \) is very powerful here. AND \( \text{Adv}'s \) goal is very weak \( \Rightarrow \) Strong security.

\[ \Rightarrow \text{Strongest possible??? No...} \]
Sanity Check:
Why does CPA + Ctext integrity \[=\] CCA Security?

Idea: * Ctext integrity means that all decryption queries will output "fail".
* Then we're back to CPA game.
* CPA says attacker can't win.

CCA Observations
* CCA security \[\Rightarrow\] CPA security \[\Rightarrow\] CCA must be non-deterministic/stateful.
* CCA cts cannot be "malleable" at all.
* C^* \[\sim\] C^* ask for dec of C^*
Bad Ideas

MAC-then-encrypt
- Many many attacks (SSL)
- Basic idea: "padding oracle"

Encrypt-and-MAC
- Used in SSL (old versions)

Fundamental idea:
If enc scheme is CCA secure
secure adv cannot learn any
info on result of decrypting adv-chosen ct

MAC-then-encrypt & encrypt-and-MAC don't
guarantee in general.
Before we leave symmetric-key crypto, I wanted to mention a few other concepts you might hear.

So far

**OWF**

$$f: \{0,1\}^n \rightarrow \{0,1\}^m$$

Given \(y = f(x)\) s.t. \(x \in \{0,1\}^n\)

hard to find \(x'\) s.t. \(S(x') = y\)

**PRF:**

$$F: K \times \{0,1\}^n \rightarrow \{0,1\}^n$$

\(F(K, .)\) "looks like" a random \(fn\) from \(\{0,1\}^n \rightarrow \{0,1\}^n\)
**PRG:** \[ G : \{0,1\}^n \rightarrow \{0,1\}^{100n} \]

Stretch a short random string into a long pseudo-random string.

\[ \{ G(s) : s \in \{0,1\}^n \} \subseteq \{ r : r \in \{0,1\}^{100n} \} \]

Can build from PRF \( F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \)

\[ G(s) := (F(s,0) || F(s,1) || \cdots || F(s,99)) \]

Pseudorandom by PRF security

**PRP:**

Pair \( F, F^{-1} : \mathcal{X} \times \{0,1\}^n \rightarrow \{0,1\}^n \)

1. \( F \) is PRF
2. \( \forall k \in \mathcal{X} \forall x \in \{0,1\}^n \)
   \[ x = F^{-1}(F(k,x)) \]

* AES is actually a PRP, → why?
* N.B. \( F(k, \cdot) \) cannot have collisions!
* Use AES as PRF... okay until adv sees \( 2^{n/2} \) blocks → Birthday!
All equally powerful in theory terms...

**THEORY**

OWF \(\rightarrow\) HILL \(\rightarrow\) GGM tree \(\rightarrow\) Luby-Rackoff \(\rightarrow\) PRP

**PRACTICE**

Immediate via "Switching Lemma"

Counter mode

Immediate

CTR mode

Chacha20

Switching Lemma

AES