Lecture 5: Back to Encryption
Plan

→ Review
  * Building blocks
  * CPA secure enc
  * MAC
→ GMAC analysis
→ Stretch break
→ CCA security & AE
  * Encrypt then MAC.

* Set 1 due Friday 5pm
* Reminder: Extensions
Primitives so far

One-way fn (OWF) - "hard to invert"
\[ f: \{0,1\}^n \rightarrow \{0,1\}^n \]

Pseudorand generator (PRG) - "small random \Rightarrow big \& \text{ pseudo-random string}"
\[ G: \{0,1\}^n \rightarrow \{0,1\}^{2n} \]

Pseudorand fn (PRF)
\[ F: \mathbb{R} \times \{0,1\}^n \rightarrow \{0,1\}^n \]
\[ F(k, \cdot) \] looks like a random fn when key is random & hidden!
\[ A \neq A \]

Pseudorand perm (PRP) - "block cipher"
\[ F: \mathbb{R} \times \{0,1\}^n \rightarrow \{0,1\}^n \]
\[ F^{-1}: \mathbb{R} \times \{0,1\}^n \rightarrow \{0,1\}^n \]
\[ \forall k \in \mathbb{R} \forall x \in \{0,1\}^n \]
\[ F^{-1}(k, F(k, x)) = x \]

"F(k, \cdot) looks like a random perm"!
\[ A(k) \neq A \]

Make sure you know & understand the formal deff's (see past notes)
All equally powerful in theory terms...

Theoretical concepts:
- OWF
- HILL
- GGM tree
- Luby-Rackoff

Intermediate concepts:
- Immediate
- Counter mode
- Immediate via "Switchover lemma"

Practical application:
- AES

Note: Will discuss how to build these things next time.
**Bigger tools**

**CPA-Secure encryption** *(weak/passive sec)*

\[ (\text{Enc}, \text{Dec}) \text{ over } \mathcal{K} \text{ is CPA secure if } \forall \text{ eff adv } \exists \text{ negl } \sigma \text{ st. } |Pr[A \text{ outputs } 1 \text{ when } b=0] - Pr[A \text{ outputs } 1 \text{ when } b=1]| \leq \sigma \]

![Diagram](image)

Even getting WEAK CPA security requires randomness!

Eg: encrypting SSH comm *PASS WORD*

Counter mode using PRF \( F: \mathcal{K} \times \{0,1\}^n \rightarrow \{0,1\}^n \)

\[
\text{Enc}(k, m_0 \| m_1 \| \cdots \| m_n) = \\
\text{IV} \leftarrow \{0,1\}^n \\
m = m_0 \| m_1 \| m_2 \cdots \\
\text{c} = (\text{IV}, F(k, IV) \| F(k, IV \| m_1) \| F(k, IV \| m_1 \| m_2) \cdots)
\]

Doesn't need to be multiple of d block size
Message Authentication Code

End schemes give no integrity protection? MAC does.

A MAC is a function:
\[ \text{MAC} : \mathcal{K} \times \mathcal{M} \rightarrow \{0,1\}^n \]

\[ k \in \mathcal{K} \]
\[ t_i \leftarrow \text{MAC}(k, m_i) \]
\[ m_i \]
\[ t_i \]
\[ (m^*, t^*) \]

Adv wins if
\[ \text{MAC}(k, m^*) = t^* \]
AND \[ m^* \notin \{m_1, \ldots, m_n\} \]

Note: Some MAC schemes have a separate alg.
Ver.(k, m, t) \rightarrow \{0,1\}
to check tags. GMAC does.
**GMAC (Simplified!)**

First, define

\[ \text{GHASH}(r, m_0 \| \ldots \| m_i) := \text{len}(m) + m_1 r + m_2 r^2 + \ldots + m_n r^n \in \text{GF}(2^{128}) \]

\[ \text{GMAC}((k, r), m) := \{ \text{IV} \leftarrow \{0,1\}^{128} \}

(\text{IV}, F(k, \text{IV}) \Theta \text{GHASH}(r, m) \}

\[ \text{GMACVer}((k, r), m, (\text{IV}, t)) := \{ F(k, \text{IV}) \Theta \text{GHASH}(r, m) = t \} \]

**Claim** If \( m \neq m' \)

\[ \Pr_{r \in \{0,1\}^{128}}[\text{GHASH}(r, m) = \text{GHASH}(r, m')] \leq \frac{n}{2^{128}} \]

**Idea:** \( \text{GHASH}(r, m) = \text{GHASH}(r, m') \)

\[ \Rightarrow (m_i - m_i') r + (m_i - m_i') r^2 + \ldots + (m_i - m_i') r^n = 0 \]

\[ \Rightarrow P(r) = 0 \]

Non-zero Poly of degree \( n \). At most \( n \) roots!

**Claim:** GMAC is secure MAC.

**Idea:** Adv has no info on \( r \). So all adv in are indep of \( m \). So adv sees w.p \( \frac{2^{128} n}{2^{128}} = \frac{2^{128}}{2^{128}} \) on \( q \) queries.
Stretch Break
CCA Security

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

What if adv can see decryptions?

How could adv learn any info on decrypted msg?

* B could reply w/ msg of varying len
* B could throw error
* B could reply in diff time
* B could perform other action

⇒ Two tasks
1. Stronger sec defn (CCA)
2. Stronger enc scheme.

Decryption routine for CCA scheme can output "Fail"
CCA: Definition

CCA Security Def:

$(Enc, Dec)$ is CCA secure if for all $\mathcal{A}$ s.t. $|\text{Pr}[W_0 = 1] - \text{Pr}[W_1 = 1]| \leq \text{negl}$.

Adv is very powerful here. AND adv's goal is very weak $\Rightarrow$ Strong security.

$\Rightarrow$ Strongest possible??? No...
CCA Observations

* CCA sec \implies\ CPA sec \implies\ CCA must be weak.
* CCA cts cannot be "malleable" at all
  \[ C^* \sim C^* \text{ ask for dec of } C^* \]
* CCA admits schemes that allow adv to cook up own cts

Authenticated encryption

(Enc, Dec) is AE if:
1) Is CPA secure and
2) Satisfies "cheat integrity"
   Adv wins if
   \[ C^* \neq \{ C_1, \ldots, C_n \} \]
   \[ \text{and } \text{Dec}(k, z^*) = \text{reject} \]

AE security \implies\ CCA security.

\[ \implies \text{ Msg integrity} \]

AE is "gold standard" for enc security.
\[ \implies \text{ AEAD = AE + associated (auth but not enc) data} \]
Constructing AE schemes

“Encrypt then MAC” → As easy as it sounds

- Independent keys for both parts (PRF)
- AES-GCM is standard
- ChaCha-Poly1305 is another

To decrypt:
1) Check MAC on it. Is bad, FAIL.
2) Then decrypt.

Encrypt-then-MAC is only safe way to combine enc & MAC
* AES-GCM = AES-CTR then GMAC
* Also common = ChaCha20 + Poly1305
* Well-designed crypto APIs handle this for you.

It’s possible to construct AE directly from PRF (OCB)
  ⇒ less common. Why
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 only CPA secure, adv cannot learn any info on result of decrypting adv-chosen ct
MAC - then encrypt & encrypt-and-MAC violate this!
Why MAC-then-encrypt is bad:

* Some enc schemes (CBC mode) require plaintext be multiple of block size, e.g. 16 bytes
  - Convenient & sometimes necessary
* Pad msg with n indicating "truncate n+1 bytes"

\[\text{msg} \rightarrow \text{MAC} \rightarrow \text{pad} \rightarrow \text{AES(k,)} \rightarrow \text{encrypt} \rightarrow \text{ct} \rightarrow \text{decrypt} \rightarrow \text{check padding} \rightarrow \text{Fail MAC}\]

If adv can learn whether padding is valid, learns one byte of msg?