Admin: PSet #2 due March 12
Project descriptions due March 23

Today: Symmetric encryption (stream cipher)
Authentication (MACs).

Readings: Serious cryptography:
Ch. 5 & 7.

Recall: 1. Block ciphers:

- Encrypts blocks of fixed length.
- Goal: Indistinguishable from pseudorandom permutation (ideal cipher).
- Even if it ideal cipher, only secure if msg is "random".

2. Mode of operations

Allows to encrypt msgs of arbitrary length & msg do not need to be random.
**Counter Mode (CTR)**

\[
\begin{align*}
X_0 & \quad X_1 & \quad X_N & \quad \text{output} \\
E_{K_0} & \quad E_{K_1} & \quad \ldots & \quad E_{K_N} \\
M_0 & \quad M_1 & \quad \ldots & \quad M_N \\
C_0 & \quad C_1 & \quad \ldots & \quad C_N
\end{align*}
\]

**Stream Cipher**: Generates pseudorandom bits from the key, and encrypts plaintext by xoring with the pseudorandom bits (a la one-time pad).

CTR is a stream cipher.

Stream ciphers can handle arbitrary length msgs without padding or ciphertext stealing methods.

**Cipher Block Chaining Mode (CBC)**

\[
\begin{align*}
\text{IV} & \quad \oplus & \quad M_0 & \quad \oplus & \quad M_1 & \quad \oplus & \quad M_2 & \quad \text{output} \\
E_{K_0} & \quad \quad \downarrow & \quad E_{K_1} & \quad \quad \downarrow & \quad E_{K_2} & \quad \quad \downarrow & \quad \ldots & \quad \downarrow \\
C_0 & \quad \quad \downarrow & \quad C_1 & \quad \quad \downarrow & \quad C_2
\end{align*}
\]

- If msg is not of length which is multiple of block length, need to pad or ciphertext stealing
* Are these mode of operations secure?

**Claim:** If block cipher is indistinguishable from ideal cipher then these encryption schemes are secure against chosen plaintext attacks (CPA).

**Goal:** Security against chosen ciphertext attacks (CCA) (probabilistic poly time)

**Def:** An encryption scheme is **CCA-secure** if a (PPT) adversary can win the following game w.p. $\leq \frac{1}{2} + \epsilon$ for some "negligible" function $\epsilon$.

Let $K$ be randomly chosen key.

Let $E_k$ denote encryption alg w. key $K$.

Let $D_k$ denote decryption alg w. key $K$.

**Game:**

- Adv is given Black-Box access to $E_k & D_k$

**Phase 1**

- Adv outputs two msgs $M_0, M_1$ of same length (and oracle information $S$)

- Adv is given $C = E_k(M_b)$ for random $b \in \{0,1\}$.

& is given Black-Box access to $E_k & D_k$ (except on $C$), & is given the state $S$.

- Adv outputs bit $\hat{b}$ & wins iff $\hat{b} = b$.

**CPA-Game:** Same except Adv is never given oracle to $D_k$ (only $E_k$).
\( b - b \) is called the \textbf{advantage} of \textit{Adv}.

The encryption scheme is \textit{CCA-secure} (resp. \textit{CPA-secure}) if \( \textit{Adv} \) has advantage in the \textit{CCA-game} (resp. \textit{CPA-game}) is negligible.

**Thm:** CBC & CTR are not \textit{CCA-secure}.

**Pf:** \( \textit{Adv} \) picks \( m_0 = 0^n \) & \( m_1 = 1^n \).

Given \( C \in E_k(m_b), \)

let \( C' = 1^{st} \) half of the bits of \( C \)

Since \( C' \neq C \), \( \textit{Adv} \) is allowed to ask \( D_k \)
to decrypt \( C' \), which gives \( 1^{st} \) half bits of \( m_b \),

revealing \( b \).

How do we design \textit{CCA-secure} schemes?

1. Construct a scheme that is only \textit{CPA-secure}

   \( \text{Recall: CBC & CTR are CPA-secure if underlying block cipher} \) \textit{is indistinguishable from ideal cipher}.

2. Add authentication.
Message Authentication Code (MAC)

Provides integrity (authenticity), not confidentiality.

Alice \( M, K \) \( \rightarrow \) Bob \( K \)

- Allows Bob to verify that \( M \) originated from Alice, & arrived unmodified.
- Alice & Bob need to share a secret key.
- Orthogonal to confidentiality, typically we do both (encrypt & append MAC for integrity).

Secure MAC

Goal: Security against adaptive chosen msg attack:

Adv is given pairs \((M_i, \text{MAC}_k(M_i))\) to msgs \( M_i \) of its choice, and cannot generate new \( M^* \) with valid \( \text{MAC}_k(M^*) \).
If MAC has t bits then Adv can guess w.p. $2^{-t}$
so t should be large enough.

Thm: CPA secure enc. scheme + MAC $\Rightarrow$ CCA secure enc. scheme.

How to Construct a MAC

Two common methods:

1. From hash functions (HMAC)

2. From block ciphers (CBC-MAC).

Historically, MACs constructed from block ciphers. Constructing from hash sume is more efficient.

MAC from hash function

1st Attempt: $MAC_k(m) = h(k||m)$

Possibly
* Vulnerable to length extension attacks:

$\left[ \text{Given } h(k||m) \text{ one can eff compute } h(k||m||m') \right]$ $\left[ \text{for any } m' \right]$

This vulnerability exists for Merkle-Damgard constructions (SHA-256 & SHA-512 vulnerable to this attack).

Read "Keying Hash Functions for Message Authentication" Bellare-Cametti-Krawczyk.
**2nd Attempt:** Secret-Suffix construction

\[ \text{MAC}_K(m) = h(m \| k) . \]

Length extension attacks don't work.

possibly insecure if attacker knows a collision for \( h \): \( h(m_1) = h(m_2) \)

For SHA-256 (or Merkle-Damgard)

\[ h(m) = h(m_1) \implies h(m_1 \| k) = h(m_2 \| k) . \]

**3rd Attempt:** HMAC construction (used in IPSec, SSH, TLS)

\[ \text{HMAC}_K(m) = h(k_1 \| h(k_2 \| m)) \]

where \( k_1 = k \oplus \text{opad} \)

\[ k_2 = k \oplus \text{ipad} \]

\[ \{ \text{opad & ipad fixed const.} \} \]

This construction is formally analyzed and is proven that if the hash function is "secure" then HMAC is secure.
MAC from block ciphers

Recall CBC mode of operation

\[ \begin{align*}
M_1 & \rightarrow E_k \rightarrow C_1 \\
M_2 & \rightarrow E_k \rightarrow C_2 \\
M_3 & \rightarrow E_k \rightarrow C_3 \\
\vdots
\end{align*} \]

CBC-\( \text{MAC}_k(M) \): Encrypt \( M \) w. \( IV = 0 \)
& output last cipher.

Insecure!

Given single block msg \( M_1 \) & tag \( T_1 = E_k(M_1) \)
& single block msg \( M_2 \) & tag \( T_2 = E_k(M_2) \)

\( T_2 \) is tag of \( M_1 || M_2 \oplus T_1 \).

The fix: C-MAC Process last block differently
all blocks are processed with \( K_1 \) & last block is processed w. \( K_2 \).
Desai [CRYPTO 2000]

Succinct & efficient CCA-secure enc. scheme

[LUFE (unbalanced Feistel Encryption)]

\( M = M_1 \ldots M_N \) (long) sequence of \( b \)-bit blocks.
\( K = (K_1, K_2, K_3) \) Three indep. keys for block ciphers

\( \text{ENC}_K(M) : \)

1. Compute \( (\tau, c_1, \ldots, c_N) \) using CTR mode enc. w. secret key \( K_1 \)
   \[
   \tau \leftarrow \{0,1\}^b \\
   x_i = E_{K_1}(\tau + i) \\
   c_i = m_i \oplus x_i
   \]

2. Compute CMAC of \( (c_1, \ldots, c_N) \) w.r.t. secret keys \( K_2, K_3 \)
   \[
   z_0 = 0^b \\
   z_i = E_{K_2}(\oplus c_i \oplus z_{i-1}) \quad i \in [N-1] \\
   z_N = E_{K_3}(c_N \oplus z_{N-1}) \quad \text{last block uses } K_3
   \]

3. Let \( \sigma = \tau \oplus z_N \)

output \( (c_1, \ldots, c_N, \sigma) \).
Encryption can be done in single pass over data ("online" property), but decryption requires two passes:
- First to compute $Z_N$ (CMAC of $(C_1 \ldots C_N)$),
- Compute $r = \sigma \oplus Z_N$
- Then decrypt $(r, C_1 \ldots C_N)$ to get $M$.

- Provides CCA-security
  Does not provide authenticity

- Note "unbalanced Feistel structure"

Enc. Called: Unbalanced Feistel Encryption (UFE)

- Length of ciphertext $| (C, \sigma) | = | M | + 1 | r |$