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:

$$M \rightarrow \boxed{E_K} \rightarrow C$$

- Encrypts blocks of fixed length.

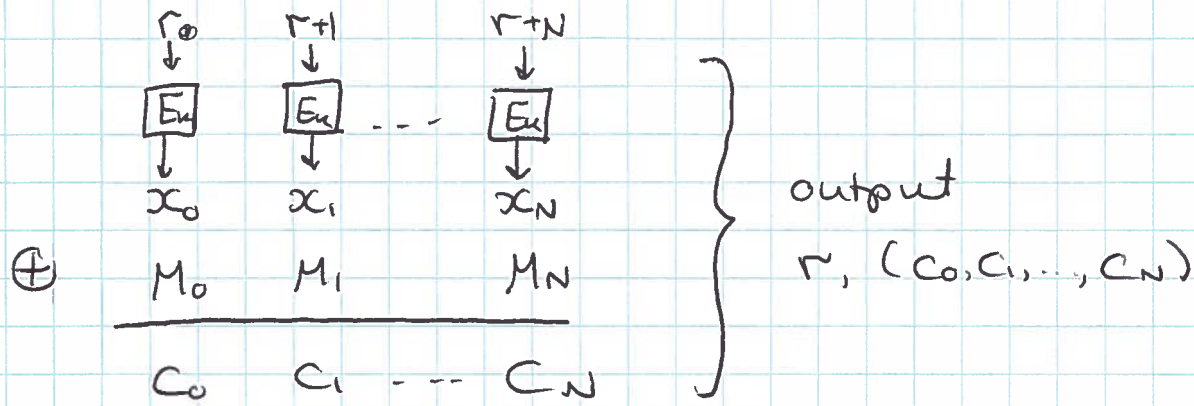
- Goal: Indistinguishable from pseudorandom permutation (ideal cipher).

- Even if it ideal cipher, only secure if msg is "random".

② Mode of operations

Allows to encrypt msgs of arbitrary length & msgs do not need to be random.

Counter Mode (CTR)

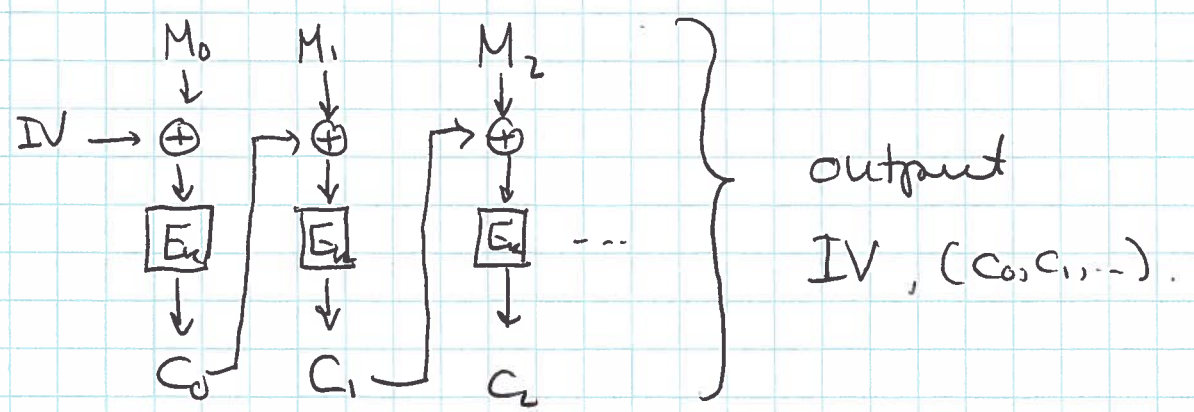


Stream Cipher: Generates pseudorandom bits from the key, and encrypts plaintext by xoring w. the pseudorandom bits (a la one-time pad).

CTR is a stream cipher.

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

Cipher Block Chaining Mode (CBC)



- If msg is not of length which is multiple of block length, need to pad or ciphertext stealing

* Are these mode of operations secure?

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

<u>Goal</u>: Security against chosen ciphertext attacks (CCA).

<u>Def</u>: An encryption scheme is <u>CCA-secure</u> if a (PPT) (probabilistic poly time) adversary can win the following game w.p. $\leq \frac{1}{2} + \varepsilon$ for some "negligible" function ε.

Let K be randomly chosen key.

Let E_K denote encryption alg w. key K.

Let D_K denote decryption alg w. key K.

<u>Game</u>: Phase I
- Adv is given Black-Box access to E_K & D_K
- Adv outputs two msgs M_0, M_1 of same length (and state information S)
- Adv is given $C \leftarrow E_K(m_b)$ for random $b \leftarrow \{0,1\}$.

& 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 $\hat{b} = b$.

<u>CPA-Game</u>: Same except Adv is never given oracle to D_K (only E_K).

$\hat{b}-b$ is called the <u>advantage</u> of Adv.

The encryption scheme is CCA-secure (resp. CPA-secure) if $\forall$ Adv its advantage in the CCA-game (resp. CPA-game) is negligible.

<u>Thm</u>: CBC & CTR are <u>not</u> CCA secure.

<u>Pf</u>: Adv picks $m_0 = 0^N$ & $m_1 = 1^N$.

Given $C \leftarrow E_k(m_b)$,

let $C' = 1^{st}$ half of the bits of C

Since $C' \neq C$, adv is allowed to ask D_k to decrypt C', which gives 1st half bits of m_b, revealing b.

$\blacksquare$

<u>How do we design CCA-secure schemes?</u>

① Construct a scheme that is only CPA secure

(<u>Recall</u>: CBC & CTR are CPA-secure if underlying block cipher is indistinguishable from ideal cipher)

② Add authentication.

Message Authentication Code (MAC)

Provides integrity (authenticity), not confidentiality.

Alice M, K $\xrightarrow{M,\ MAC_K(M)}$ Bob K

authentication tag of M.

Bob recomputes $MAC_K(M)$ & verifies that it agrees w. what he received. If not, reject msg.

- 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 $(M_i, MAC_K(M_i))$ to msgs M_i of its choice, and ~~cannot~~ cannot generate new M^* with valid $MAC_K(M^*)$.

If MAC has t bits then Adv can guess w.p. 2^{-t}

so t should be large enough.

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

How to Construct a MAC

Two common methods:

① From hash functions (HMAC) — Read "Keying Hash Functions for Message Authentication" Bellare-Canetti-Krawczyk

② From block ciphers (CBC-MAC)

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

MAC from hash function

1'st Attempt: $MAC_k(m) = h(k \| m)$ — secret prefix construction.

* Possibly Vulnerable to length extension attacks:

Given $h(k\|m)$ one can eff compute $h(k\|m\|m')$ for any m'.

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

2nd Attempt: Secret-Suffix construction

$$MAC_K(m) = h(m \| K).$$

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)

$$h(m_1) = h(m_2) \implies h(m_1 \| K) = h(m_2 \| K).$$

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

$$HMAC_K(m) \triangleq h(K_1 \| h(K_2 \| m))$$

where $K_1 = K \oplus \underbrace{opad}_{\text{outer padding}}$

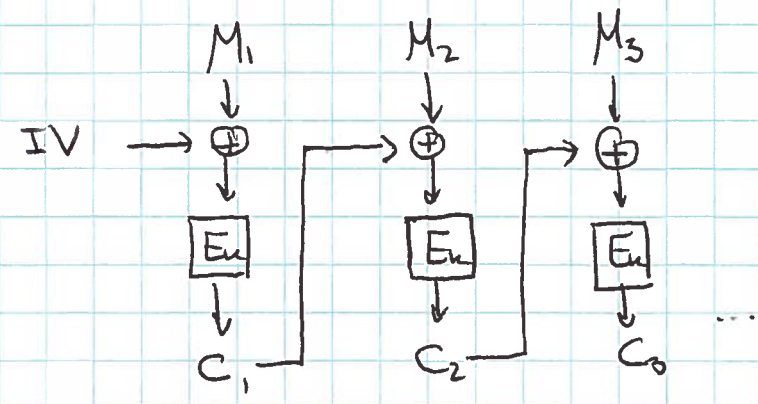
$K_2 = K \oplus \underbrace{ipad}_{\text{inner padding}}$

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



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

<u>Insecure</u>!

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 $M_1 \| M_2 \oplus T_1$

<u>The fix</u>: <u>CMAC</u> 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

[UFE (unbalanced ~~cipher~~ 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

$Enc_K(M)$:

① Compute $(r, C_1 \ldots C_N)$ using CTR mode enc. w. secret key K_1

$r \leftarrow \{0,1\}^b$

$X_i = E_{K_1}(r + i)$

$C_i = M_i \oplus X_i$

② Compute CMAC of $(C_1, \ldots C_N)$ w.r.t. secret keys K_2, K_3

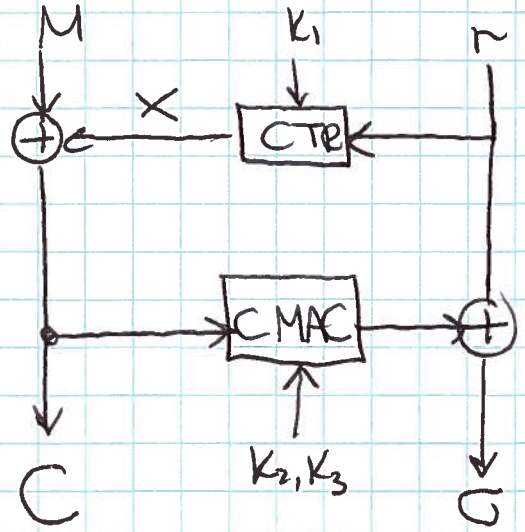
$Z_0 = 0^b$

$Z_i = E_{K_2}(C_i \oplus Z_{i-1})$ $i \in [N-1]$

$Z_N = E_{K_3}(C_N \oplus Z_{N-1})$ ← last block uses K_3

③ Let $\sigma = r \oplus Z_N$

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 $|(c, \sigma)| = |M| + |r|$