Understanding Symmetric Key Cryptography

Mridul Nandi

Indian Statistical Institute, Kolkata

mridul@isical.ac.in

ACM School
ISI Kolkata

What do we want to Achieve in Cryptography

- Hide content - Privacy.
  - Encryption.
- Ensure originality of content - Integrity.
  - Signature/Message Authentication Code (MAC).
- Authorizing Right Person - (Identity) Authenticity.
  - Identity-based Authentication/MAC.
- Combination of above.
  - Signcryption (Signature + Encryption) and Authenticated Encryption (Authentication and Encryption)

What is Encryption

- Symmetric Key.
  - Key K, $\text{Enc}(K, M) = C$ and $\text{Dec}(K, C) = M$.
  - Minimum Condition: $\text{Dec}(K, \text{Enc}(K, M)) = M$.
  - AES, DES, counter mode encryption, CBC encryption etc.
- Public Key.
  - Key is a pair (PK, SK). $\text{Enc}(PK, M) = C$ and $\text{Dec}(SK, C) = M$.
  - Minimum Condition: $\text{Dec}(SK, \text{Enc}(PK, M)) = M$.
  - RSA, Elgamal Encryption etc.

Symmetric Key Encryption

- One time padding (classical): Two simple ways to encrypt.
  1. $M \oplus K = C$.
  2. $C = M + K \mod N$ for some predetermined large N.
- Achieves perfect secrecy - Given ciphertext no information about the message is leaked.

$$\text{Exercise}: \ Pr[M = m | C = c] = Pr[M = m]$$

- So, we are done, right?

Symmetric Key Encryption

- One time padding (classical): Two simple ways to encrypt.
  1. $M \oplus K = C$.
  2. $C = M + K \mod N$ for some predetermined large N.
- Achieves perfect secrecy - Given ciphertext no information about the message is leaked.

$$\text{Exercise}: \; Pr[M = m | C = c] = Pr[M = m]$$

- So, we are done, right?
- No, we have some issues:
  1. performance issue: Key size is as large as message size.
  2. security issue: Key-recovery. Leaks information of messages (while encrypting more than once).

Stream Cipher

- Stream Cipher Encryption: Classical and efficient encryption (for arbitrary sized message).

$$C = \mathsf{G}(K, |M|) \oplus M.$$

- $\mathsf{G} : \{0,1\}^k \times \mathbb{N} \to \{0,1\}^*$ such that for all positive integer ℓ and $K \in \{0,1\}^k$, $\mathsf{G}(K, \ell) \in \{0,1\}^\ell$.

Examples

1. Popular: RC4, SNOW etc.
2. ECRYPT Stream Cipher Project - eStream - 2004-2008.
3. HC-128, Rabbit, Salsa20/12 and SOSEMANUK for software.
4. Grain v1, Micky 2.0 and Trivium for hardware.

Pseudorandom Bit Generator (PRBG)

- Notation: U_n is a random (uniformly distributed) n-bit string.
- For all ℓ, $\mathrm{G}(U_k, \ell)$ should be close to U_ℓ.
- Can we have any such G?

Definition

$\mathrm{G} : \{0,1\}^k \times \mathbb{N} \to \{0,1\}^*$ is called (t, ϵ, ℓ)-PRBG if for all algorithm D runs in time t,

$$|\Pr(D(U_\ell) = 1) - \Pr(D(G(U_k, \ell)) = 1)| \leq \epsilon.$$

- PRBG would solve large key issue. Still cannot use more than once (leaks information of messages).

Classical vs. IV based Stream Ciphers

Classical

- Classical Streamcipher (e.g. RC4, SNOW) generates key stream in online manner.
- Need to hold current internal secret state to make use of multiple times.
- If random position key-stream can be generated efficiently, we can still use.

IV-based stream cipher

- IV-based Stream Cipher Encryption (e.g. Trivium, Grain)

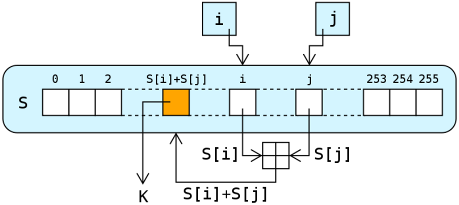
$$(C, IV) = \text{PRBG}(IV \| K) \oplus M$$

- Can we decrypt? What is IV? How it is generated? etc.

RC4 Stream Cipher

```
for i from 0 to 255
S[i] := i
endfor
j := 0
for i from 0 to 255
j := (j + S[i] + key[i mod keylength]) mod 256
swap values of S[i] and S[j]
endfor
```

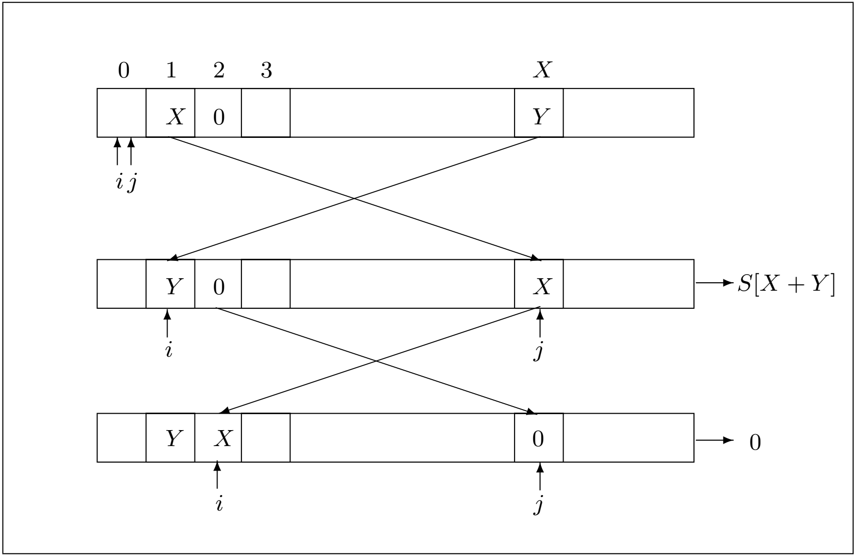
RC4 Stream Cipher



i := 0, j := 0
while GeneratingOutput:
i := (i + 1) mod 256
j := (j + S[i]) mod 256
swap values of S[i] and S[j]
K := S[(S[i] + S[j]) mod 256]
output K
endwhile

Mantin-Shamir Attack on RC4 Stream Cipher

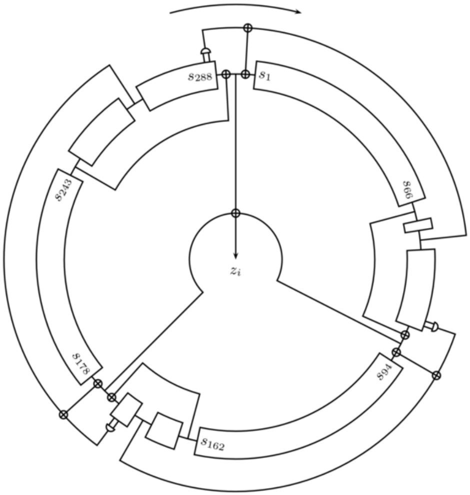
Exercise: The second byte output is 0 with prob $\frac{2}{256}$ (bias).



Attack on WEP based RC4 Stream Cipher

- WEP - the link-layer security protocol.
- It uses RC4.
- Key recovery attack.
- Applies to Broadcast Situation - Multiple RC4 encryption of a same message.
- Snow, ZUC (used in mobile device).

Trivium Stream Cipher (1152-rounds/288-state)



Counter Mode Encryption

- No need to hold current internal secret state, moreover we can directly compute key-stream.
- Random position key-stream can be generated efficiently.
- Pseudorandom function.

$$f : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n$$

such that for all distinct m-bit strings $x_1, \ldots, x_s$,

$$f(U_k, x_1), \ldots, f(U_k, x_m)$$

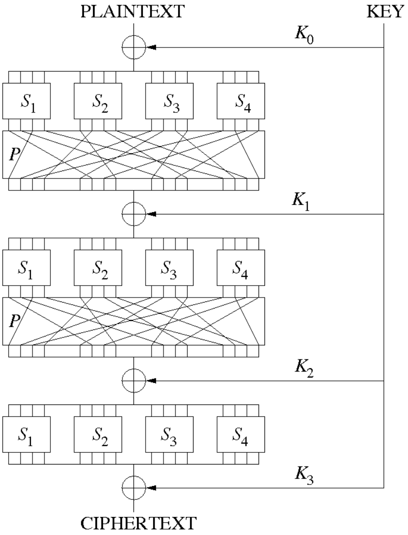
look computationally close uniform distribution over $\{0,1\}^{ns}$.

Counter Mode Encryption

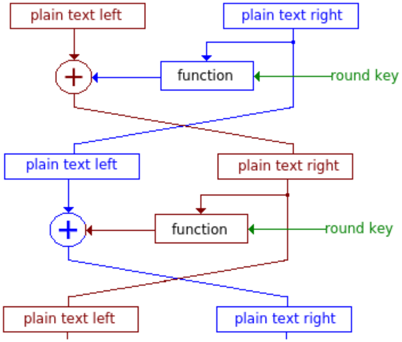
- Let $\log L = \ell$ (maximum message size is nL bits).
- Let $K \in \{0,1\}^k$ random secret key.
- Choose $IV \in \{0,1\}^{m-\ell}$ (distinct for each encryption).
- We can define key-stream (like stream cipher) of counter mode as
$$f(K, IV\|0)\| \cdots f(K, IV\|\ell' - 1)$$
where $(\ell' - 1)n < |M| \leq \ell' n$.
- Instead of keyed function, keyed permutation (also called block cipher) is more popular (here $m = n$).

SPN (e.g. AES) and Feistel (e.g., DES)

SPN three rounds



Feistel cipher two rounds



Security Notions of Symmetric Key Encryption

- Need to describe (i) Power and (ii) Goal of an adversary.
- Power: Only ciphertext, both plaintext and ciphertext. Number of such texts. Access of encryption/decryption function etc.
- Goal: Key recovery, message recovery, some information about message, distinguishing from ideal encryption (?) etc.

Distinguishing Game

Distinguishing a real keyed construction from an ideal object.

Symmetric Key Primitives

Distinguishing Game

Distinguishing a real keyed construction from an ideal object.

1. PRF or Pseudorandom function.
   - Indistinguishable from (uniform) random function.

Symmetric Key Primitives

Distinguishing Game

Distinguishing a real keyed construction from an ideal object.

1. PRF or Pseudorandom function.
   - Indistinguishable from (uniform) random function.
2. PRP or Pseudorandom permutation.
   - Indistinguishable from (uniform) random permutation by only making forward queries.

Symmetric Key Primitives

Distinguishing Game

Distinguishing a real keyed construction from an ideal object.

1. PRF or Pseudorandom function.
   - Indistinguishable from (uniform) random function.
2. PRP or Pseudorandom permutation.
   - Indistinguishable from (uniform) random permutation by only making forward queries.
3. SPRP or Strong Pseudorandom permutation.
   - Indistinguishable from (uniform) random permutation by only making forward and backward (i.e., inverse) queries.

Attacks on Feistel (Will be studied)

- If we use same key in each round.
- Chosen plaintext attack on 2 rounds.
- Chosen ciphertext on 3 rounds.

Security: 3 round is secure against chosen plaintext and 4 round is secure against chosen plaintext and ciphertext adversaries.

ECB Encryptions

- $C = \mathrm{BC}(K, M)$. Works for small message (64/128-bit). DES-56, AES-128/196/256 etc.
- ECB (Electronic Code-book Encryption) for larger messages

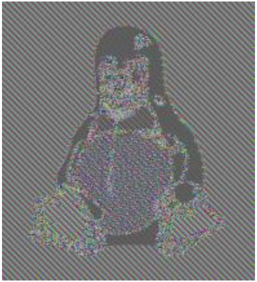
$$C_1 = \mathrm{BC}(K, M_1), \ldots, C_l = \mathrm{BC}(K, M_l).$$

- If block repeats ciphertext repeats - some impression reveals.

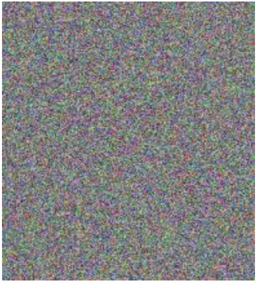
Drawback of ECB

Original Message

Encrypted under ECB



A Secure Encryption



OCB and CBC Encryption

- OCB resolves this.

$$
\begin{aligned}
C_1 &= \mathrm{BC}(K, M_1 \oplus \Delta) \oplus \Delta \\
&\vdots \\
C_l &= \mathrm{BC}(K, M_l \oplus 2^{l-1}\Delta) \oplus 2^{l-1}\Delta.
\end{aligned}
$$

- Δ is generated from IV (e.g. $BC(IV) = \Delta$.

1. CBC Encryption - sequential, simple
2. Lightweight encryption.
3. Variants of CBC encryption (different feedback)