other stream ciphers -

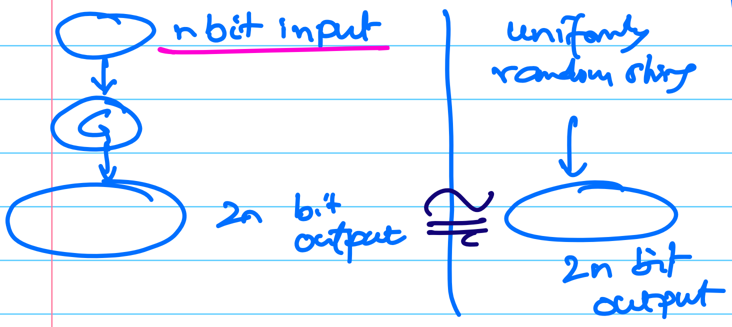
A5/1 , A5/3 ....

Sosemamak, Trivium, ...

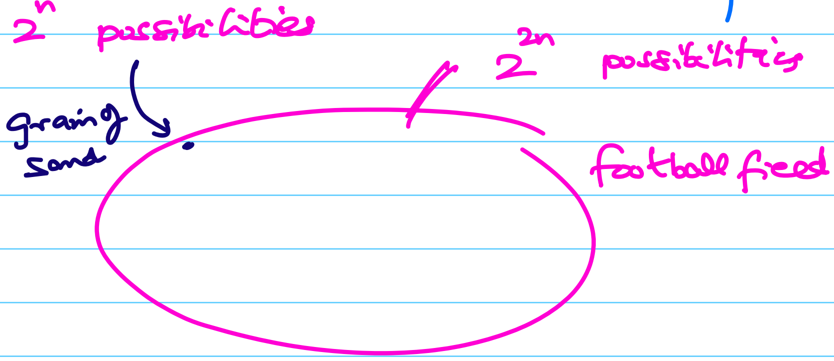
Kargil War - Musharraf ↔ Deputy army chief

Jan 21, 2025

Stream cipher :



from a PRG G
(i) length extending
(ii) deterministic
(iii) output of G on a randomly chosen input "looks" uniformly random



$\approx_c$ computationally indistinguishable

Stream cipher from a PRG

Secret key = $k \xleftarrow{\$} \{0,1\}^n$

$m \in \{0,1\}^\ell$ where $\ell > n$

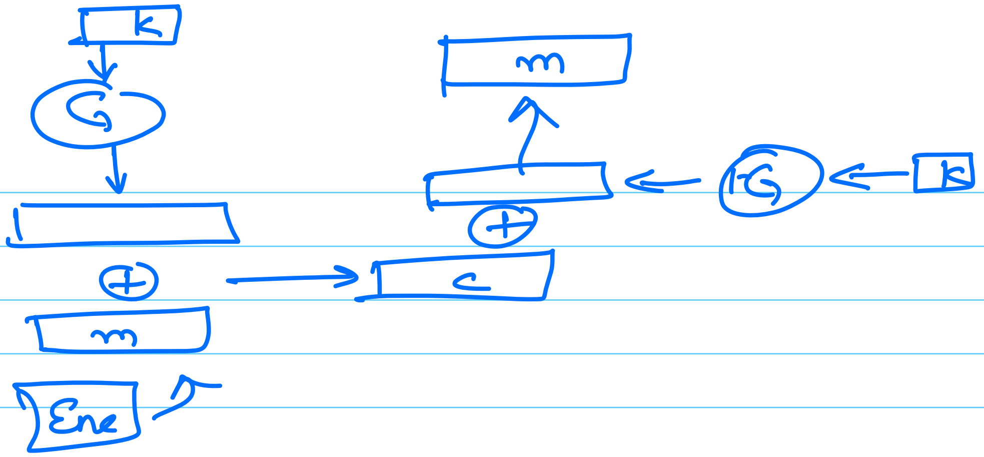
Enc: $c = m \oplus G(k)$

where $G: \{0,1\}^n \to \{0,1\}^\ell$

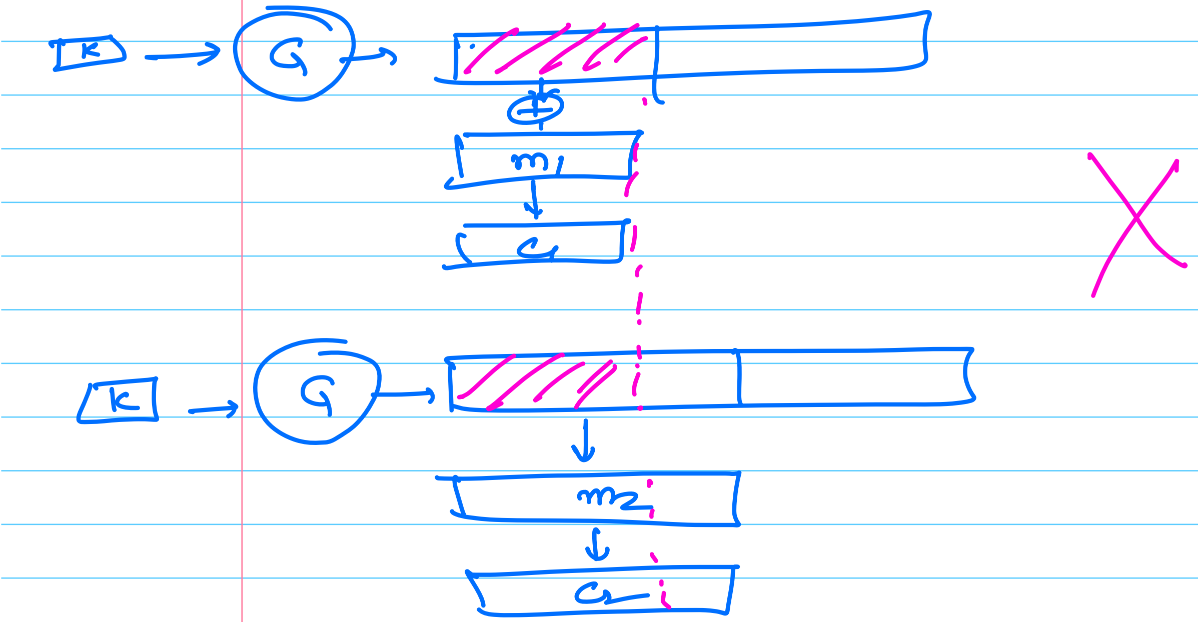
Indirt of C_1 & C_2

$C_1 = m_1 \oplus G(k)$, $C_2 = m_2 \oplus G(k)$

Dec: $m = c \oplus G(k)$

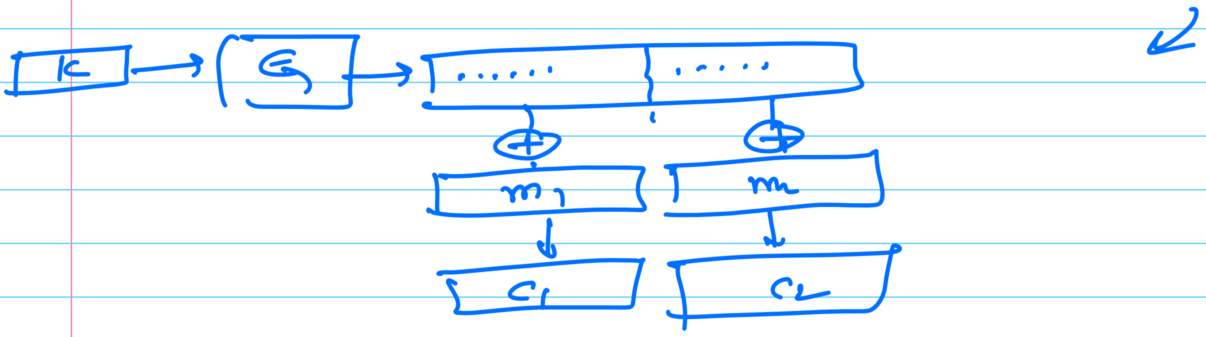


Suppose we had 2 (or more) messages to encrypt.



Instead:

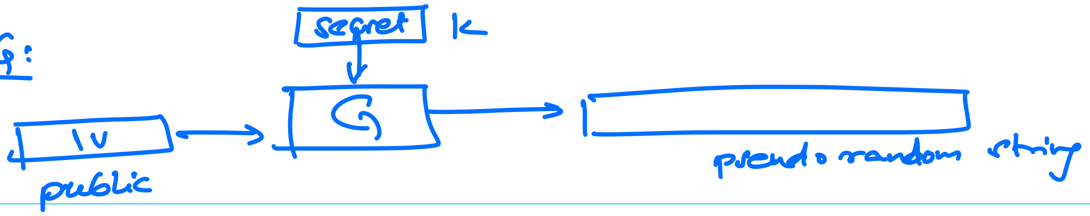
Synchronous Stream Cipher



Initialization Vector (IV)

- random string
- it can be public (it need not be hidden)

PRG:



Stream cipher:

Encrypt message m_1 with key k

(i) generate a random string IV_1

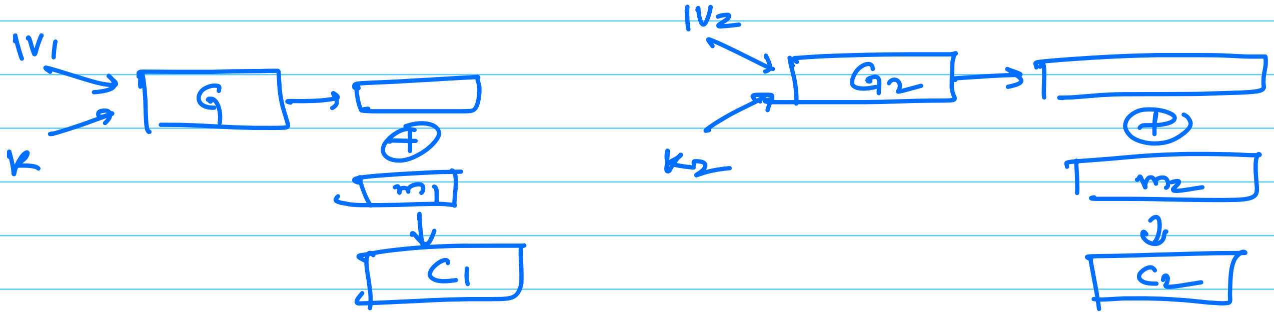
(ii) $G(IV_1, k) \oplus m_1 = c_1$

(iii) cipher text $= (IV_1, c_1)$

Decryption:

cipher text $= (\alpha, \beta)$

Decryption $= G(\alpha, k) \oplus \beta$



IV based PRG:

Security requirement

$G(IV_1, k)$ & $G(IV_2, k)$ are comp. indist.

$\simeq_c$

random string

$G(\text{known value}, \text{unknown key}) \simeq_c$ rando

WEP – design IEEE 802.11 standard for wireless communication

Wired Equivalent Privacy

Based on RC4

old design by Ronald Rivest

3 byte IV = 24 bits

2^{24} possibilities

16,000,000

IV repeat problem is less likely

Key in RC4 was upto 16 bytes = 128 bits

Export control law in US

Only 40 bit keys were permitted = 5 bytes

Designers of WEP:

key will be between 5 bytes to 13 bytes

+ 3 bytes of IV

= 8 bytes to 16 bytes

Export

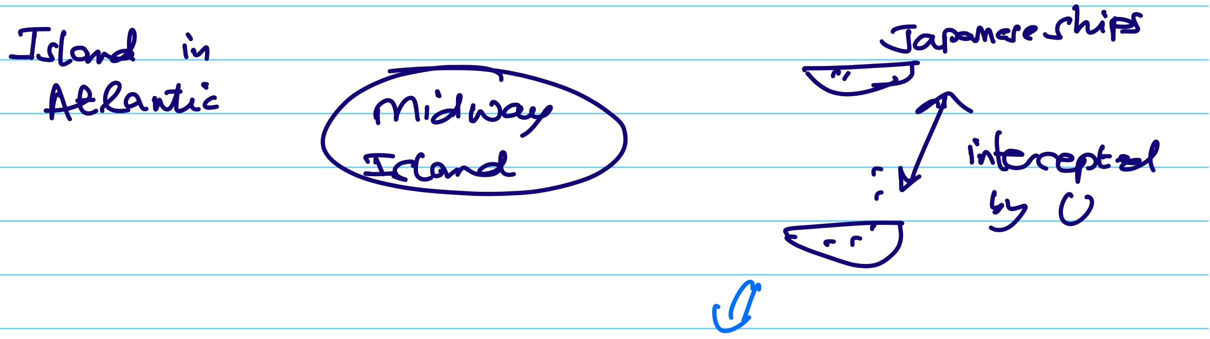
Internal use

RC4 was very weak –

Deprecated – not expected to be used anywhere

So far: only the ciphertext is available to the attacker

WW II:



AZ is to be attacked

Plm: desalination is used for drinking water

spread news: desalination plant at midway island is brokendown

Attacker can not only observe ciphertexts but can also choose plaintexts

CPA (Chosen Plaintext Attack)

Challenger | Adv

$k \xleftarrow{\$} \{0,1\}^n$

$Enc(k, \cdot)$
$Dec(k, \cdot)$ } setup

Learning

← $m_0, m_1, m_2, \ldots$

→ $c_0, c_1, c_2, \ldots$

challenge

← (m_0^*, m_1^*)

$b \xleftarrow{\$} \{0,1\}$ | where $|m_0^*| = |m_1^*|$

$c^* = Enc(k, m_b^*) \longrightarrow$ analyse

predicts $b' \in \{0,1\}$

Adv wins if $(b' = b)$

CPA secure encryption

Observation: no deterministic encryption algo. can be CPA secure.

Why? we show an CPA attack against any deterministic encryption algo.

Learning phase

Chall. Adv.

$\longleftarrow$

m_0, m_1 $(m_0 \neq m_1)$

$\longleftarrow$

$\longrightarrow c_0, c_1$

Challenge

2n bit inputs

m_0^* m_1^*

$\longleftarrow (m_0, m_0), (m_0, m_1)$

$\longrightarrow (c_1, c_2)$

Check if $(c_1 == c_2)$

yes m_0^* no m_1^*

Pr of winning = 1.