if we pick any random string in the output set, it is not likely to have been produced by this algo

$$\Pr\left(\begin{array}{c}\text{a random string} \\ \text{from output is} \\ \text{actually produced} \\ \text{from this algo.}\end{array}\right) = 1/2^n$$

Brute force attack will win with prob $\simeq 1$

— to prevent this, we will <u>not</u> allow attacker this exponential power

---

16-Jan-25

PRG - Pseudo-random generator (G)

<u>Idea:</u> a PRG takes a short seed and produces a long output which "<u>looks like</u>" random

& PRG is a deterministic function.

$G : \{0,1\}^s \longrightarrow \{0,1\}^\ell$ where <u>expansion</u> $\ell > s$

should be "looking like" random

Adv A : PPT : <u>Probabilistic</u> Polynomial Time algorithm

(i) internal coin tosses

(ii) also gets a random string as input

<u>Real world</u>	<u>Ideal world</u>
$s \xleftarrow{\$} \{0,1\}^n$	$y \xleftarrow{\$} \{0,1\}^\ell$
$y = G(s)$	

A : PPT algorithm which is going to distinguish between the two worlds

: Distinguisher for PRG G

$A(x) \to 1$ if x is random

$\to 0$ if x is generated by G.

if

$$\Pr\left(\begin{array}{c} A(G(s)\to 1) \\ \text{where } s \xleftarrow{\$} \{0,1\}^n \end{array}\right) - \Pr\left(\begin{array}{c} A(y)\to 1: \\ \text{where } y \xleftarrow{\$} \{0,1\}^\ell \end{array}\right) \le negl(n)$$

then G is a PRG.

<u>Example 1.</u>

$G(s) = s||s$ $\quad$ n bit $\to$ 2n bit

(1) <u>Que:</u> is G a PRG ?

<u>Adv:</u> $A(x) = A(x_1||x_2)$ where $|x_1| = |x_2|$

check if $x_1 = x_2$

if yes $\to$ output 0

else $\to$ output 1.

Dist prob $= 1 - 1/2^n$ = not negligible

(2) $\boxed{G \text{ is a given PRG}}$: $\{0,1\}^n \to \{0,1\}^{2n}$

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

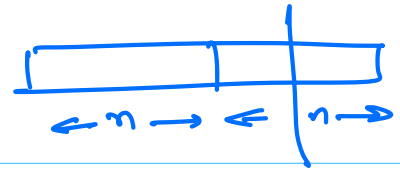
$G(x) = y_1||y_2$ where $y_1, y_2 \in \{0,1\}^n$

$G'(x) = G(y_1)||G(y_2)$ $\leftarrow$

$G' : \{0,1\}^n \to \{0,1\}^{4n}$

<u>Que:</u> is G' a PRG ?

③ G is a PRG : $\{0,1\}^n \to \{0,1\}^{2n}$

$G(x) =$ 

$G'(x) =$ first $\frac{3n}{2}$ bits of $G(x)$

<u>Why did we not permit exponential time adversary?</u>

$$G(\overset{\{0,1\}^n}{s}) \longrightarrow \{0,1\}^{\ell} = y$$

<u>Adv gets y</u>

create a table of output sequences

$T = \{ G(00\cdots0), G(0\cdots1), G(0\cdots10), \cdots$
$\cdots \}$ Size 2^n
where each entry is of length ℓ bits

When Adv gets a string x

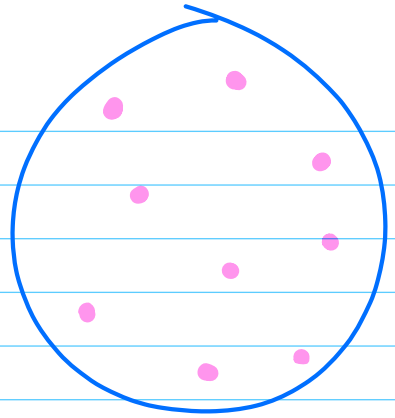
Adv checks if $x \in T$ or not

if $x \in T$. if $x \notin T$ ✓

Adv says PRG Adv say random

$\Pr$ (of A distinguishing) $= 1 - \frac{2^n}{2^\ell}$

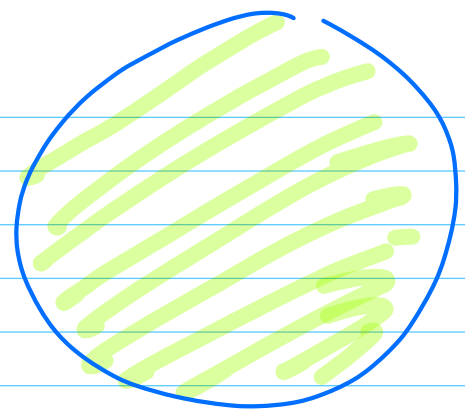
PRG:



ℓ bit outputs

coverage $= 2^n$

Random



coverage $= 2^\ell$

Do we have PRG's ?

- We don't know
- we believe they exist

OWF : (One-way function)

$$x \xrightarrow[\text{Easy}]{f} f(x)$$

$$x \xleftarrow{\times} f(x) \quad \text{Hard}$$

Trapdoor OWF

$$x \xrightarrow{\text{Easy}} f(x)$$

$$x \xleftarrow{\times} f(x) \quad \text{Hard}$$

But

$$y \xleftarrow{\text{Easy}} f(x,y)$$

y: trapdoor

OWF $\Longleftrightarrow$ PRG

Using PRG to Create Encryption Algorithm

$m \leftarrow$ plaintext $\in \{0,1\}^\ell$

secret key $k \leftarrow \{0,1\}^n$ $\qquad \ell > n$

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

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

Stream cipher

- description of the PRG

Microsoft Word → encrypt a file with a password

key ← f(password)

ciphered doc = PRG(key) + Word Doc

Earlier - RC4 was used by microsoft

RC4:

State array of size 256 byte

Init (1) Initialization: $S[i] = i$

KSA (2) Use key to randomize the state

key is of size 8 bytes to 16 bytes

eg. $K = k_0, k_1, k_2, k_3, k_4, k_6, k_7$

PRGA (3) PRG key is not used anymore

State → one byte at a time and jumble up the state again

RC4 was one of the most used stream ciphers in history

- Used in Microsoft Word, ... Lotus notes,
- IEEE 802.11 Wireless protocol

WEP

(Wired Equivalent Privacy)

other stream ciphers –

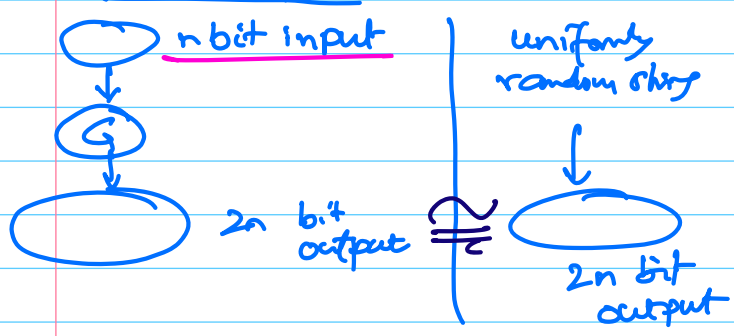
A5/1 , A5/3 ....

Sosemanuk, Trivium, ...

Kargil war – Musharraf ↔ Deputy army chief

Jan 21, 2025

Stream cipher :

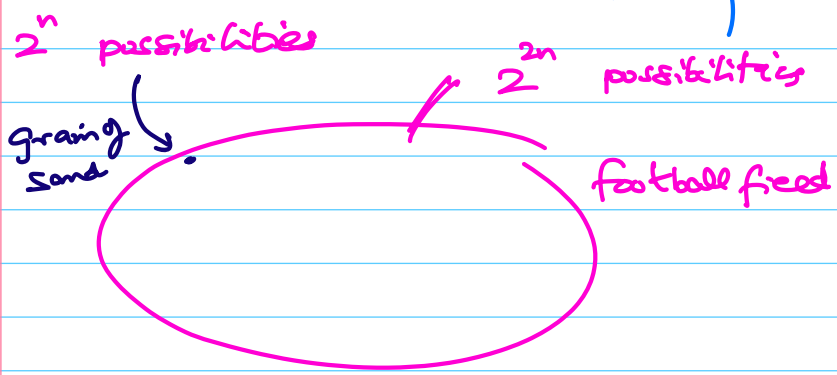


from a PRG G

(i) length extending

(ii) deterministic

(iii) output of G on a randomly chosen input "looks" uniformly random



$\cong_c$

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

Indist of c_1 & c_2

$c_1 = m_1 \oplus G(k)$, $c_2 = m_2 \oplus G(k)$

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