PESU Center for
Information Security,
Forensics and
Cyber Resilience

Welcome to

PES University

Ring Road Campus, Bengaluru

10 June 2020

PESU Center for
Information Security,
Forensics and
Cyber Resilience

APPLIED CRYPTOGRAPHY

Private key systems

Lecture 2

Pseudo Random Numbers

generating a sequence of numbers

Pseudorandom generators

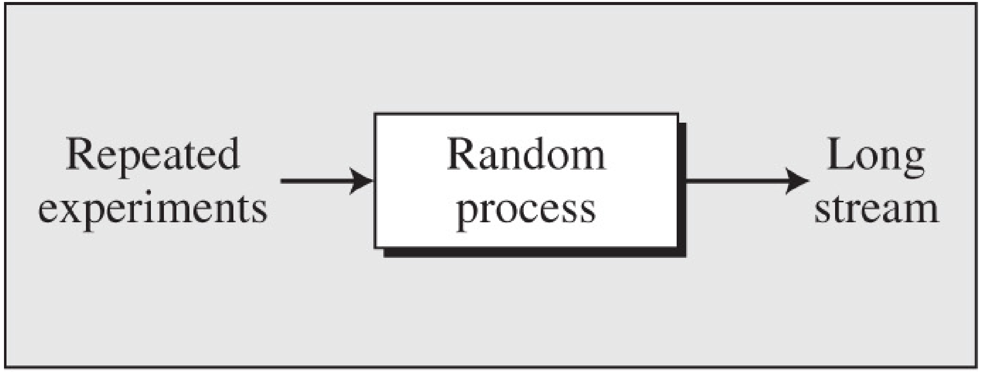
- A pseudorandom generator G is an efficient deterministic algorithm for transforming a short, uniform string called seed into a longer "uniform-looking" output string
- Let G: $\{0,1\}^n \rightarrow \{0,1\}^l$ be a function and define Dist to be the distribution on l-bit string obtained by choosing a uniform s$\in \{0,1\}^n$ and outputting G(s)

Randomness

- Truly random
- Pseudo random

Truly random

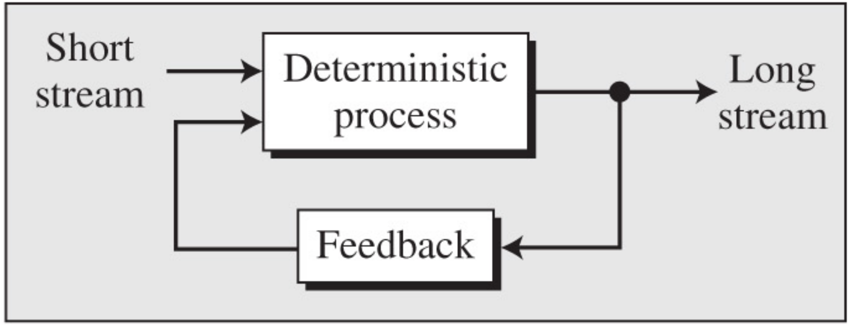
- TRG G’ is a randomised algorithm
- Output $R \in \{0,1\}^L$
- Uniformly randomly output a bit string of length L bits



a. TRGN

Pseudorandom generators

- G should be an ***efficient algorithm***
- ***Expansion:*** $L > l$
- ***Pseudo randomness:*** No efficient statistical test should significantly separate an output of G from L bit truly random generator



b. PRNG

Generator Algorithm

- Function G is called a **pseudorandom generator**.
- G is a deterministic algorithm

Input: s$\in \{0,1\}^{l}$

Output: G(s) $\in \{0,1\}^{L}$

Note: G should be a polynomial function of a security parameter (efficient).

Seed

- Kept secret
- Chosen uniformly

ALGORITHM 3.16
Constructing G_ℓ from $(\mathsf{Init}, \mathsf{GetBits})$

Input: Seed s and optional initialization vector IV
Output: $y_1, \ldots, y_\ell$

$\mathsf{st}_0 := \mathsf{Init}(s, IV)$
for $i = 1$ to ℓ:
  $(y_i, \mathsf{st}_i) := \mathsf{GetBits}(\mathsf{st}_{i-1})$
return $y_1, \ldots, y_\ell$

Encrypting long messages using short keys

$$M = K = C = \{0,1\}^L \quad G\{0,1\}^l \rightarrow \{0,1\}^L \qquad l < L$$

Consider mϵM$= \{0,1\}^L$

Encryption done by computing m$\oplus$G(s).

s is the seed = $\{0,1\}^l$

- A computationally bounded adversary will not be able to distinguish between G(s) and uniformly random string from $\{0,1\}^L$
- Both l and L are polynomial functions of security parameter n

PRG indistinguishability game

1. Hypothetical verifier:

   Challenges the distinguisher by a string or a sample of length L bits.

2. Distinguisher D

   Distinguish apart a sample generated by the pseudorandom generator from a sample generated by a truly random generator

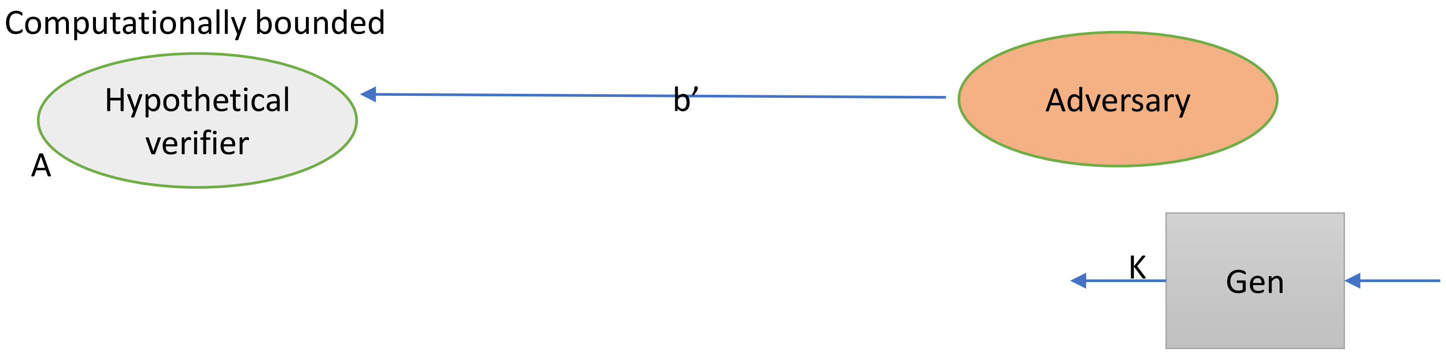
Hypothetical verifier

Uses two method to generate string of L bits

1. Truly random number generator (b=0) $y_R \in \{0,1\}^L$
2. Pseudorandom number generator (b=1) $s\{0,1\}^l \rightarrow y_p \in \{0,1\}^L$

And sends the y bits to the distinguisher

The indistinguishability experiment



Distinguisher D

Should find how y bits are generated

If for every distinguisher D participating in this experiment

$$\boldsymbol{pr(D\ outputs\ b' = b) \le \frac{1}{2} + negl(n)}$$

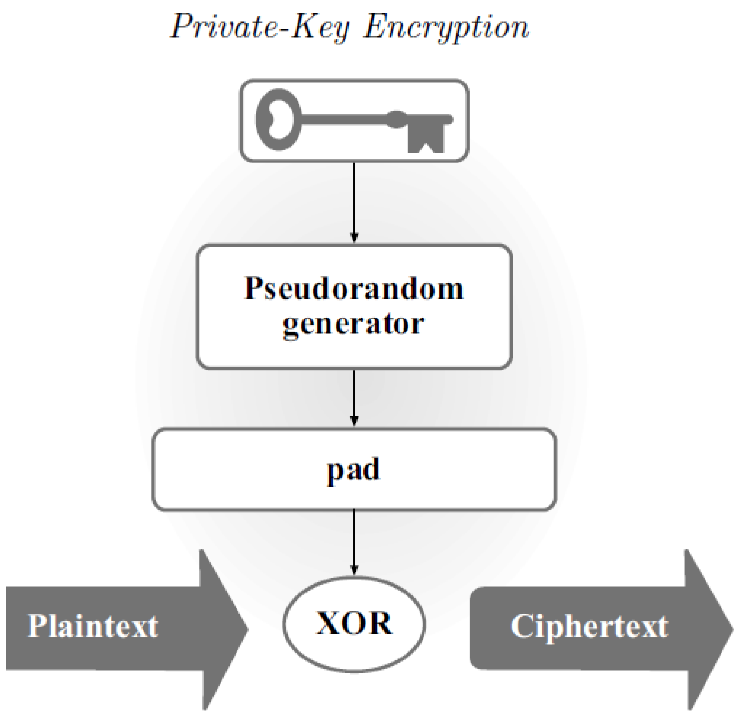
$$\boldsymbol{pr(D\ outputs\ b' = 1|b = 0)}$$

-

$$\boldsymbol{pr(D\ outputs\ b' = 1|b = 1) \le negl(n)}$$

Probability of D labelling y as outcome of PRG given that y is generated by TRG - Probability of D labelling y as outcome of PRG given that y is generated by PRG $\le negl(n)$

Encryption with a Pseudorandom generator



Pseudorandom Functions

- PRF does not require a one-to-one mapping between the input space and output space.
- A Pseudo Random Permutation is a PRF that happens to have the property that every element in the input domain has a single associated member in the output co-domain
- PRP is a bijection function (one-to-one mapping)

PRP (pseudorandom permutation)

- PRP is invertible.

*Let $F, F^{-1} : \{0,1\}^{\lambda} \times \{0,1\}^{blen} \rightarrow \{0,1\}^{blen}$ be deterministic and efficiently computable functions. Then F is a **pseudorandom permutation (PRP)** if for all keys $k \in \{0,1\}^{\lambda}$, and for all $x \in \{0,1\}^{blen}$, we have:*

- $F^{-1}(k, F(k,x)) = x.$

For example initial permutation and final permutation in DES

Thank you

Next Class

- ☞ Mandatory reading for the next class
  - ☞ https://www.coursera.org/lecture/symmetric-crypto/feistel-cipher-YgMcO

S Rajashree

Computer Science and Engineering

PES University, Bengaluru

PESU Center for Information Security, Forensics and Cyber Resilience

PESU Center for Internet of Things