Cryptography
ECE5632 - Spring 2024

Lecture 3A

Dr. Farah Raad

Lecture Topic

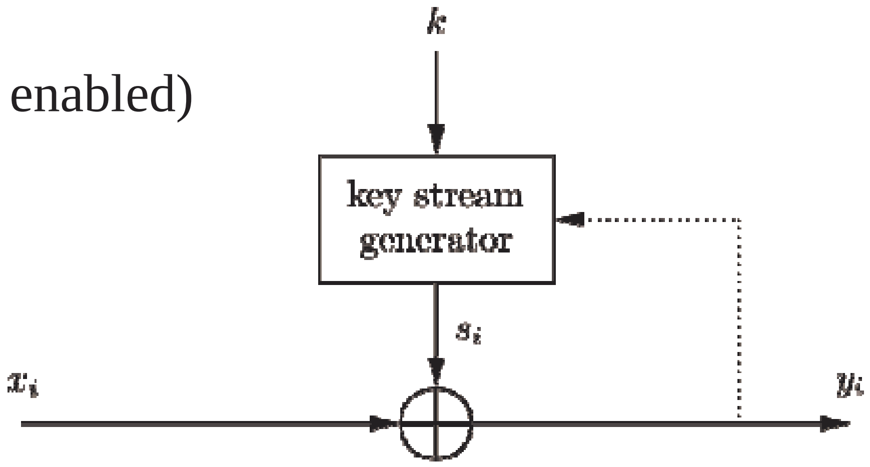
Stream and Block Ciphers

Synchronous vs. Asynchronous Stream Cipher

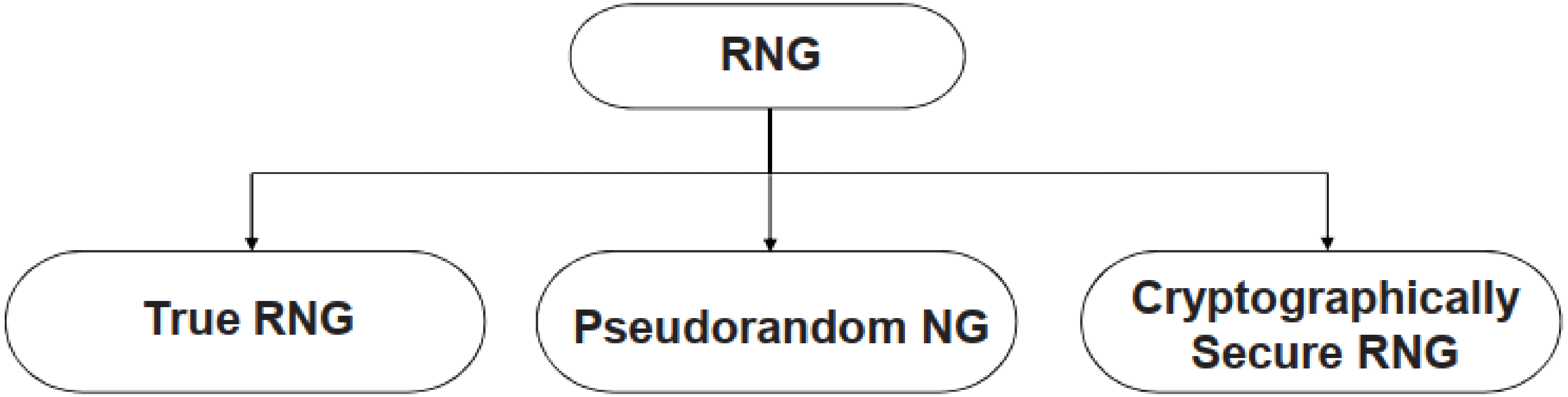
- **Security of stream cipher depends entirely on the key stream *si* :**
  - Should be **random** , i.e., Pr(*si* = 0) = Pr(*si* = 1) = 0.5 = 50%
  - Must be **reproducible** by sender and receiver

- **Synchronous Stream Cipher**
  - Key stream depend only on the key (and possibly an initialization vector)

- **Asynchronous Stream Ciphers**
  - Key stream depends also on the ciphertext (dotted feedback enabled)



Random number generators (RNGs)



Types of RNGs
a. True Random Number Generators (TRNG)
b. Pseudorandom Number Generators (PRNG)
c. Cryptographically Secure Pseudorandom Number Generators (CSPRNG)

True Random Number Generators (TRNG)

- Based on physical random processes:

  Some examples: coin flipping, dice rolling, semiconductor noise, radioactive decay, air turbulence, thermal noise, mouse movement, clock jitter of digital circuits.
- Output stream s_i should have good statistical properties:

  $\Pr(s_i = 0) = \Pr(s_i = 1) = 50\%$ (often achieved by post-processing)
- Output can neither be predicted nor be reproduced
- Typically used for generation of keys, nonces (used only-once values) and for many other purposes
- Problem: Truly random, i.e., cannot be recreated

Pseudorandom Number Generator (PRNG)

- Generate sequences from initial seed value
  - Typically, output stream has good statistical properties (meaning their output approximates a sequence of true random numbers).
  - Output can be reproduced and can be predicted
- PRNs are computed, i.e., they are deterministic. Often computed in a recursive way:

$$s_0 = seed$$
$$s_{i+1} = f(s_i, s_{i-1}, \ldots, s_{i-t})$$

$$s_0 = \text{seed}$$
$$s_{i+1} \equiv a\,s_i + b \bmod m, \quad i = 0, 1, \ldots$$

a, *b*, *m* are integer constants

- Note that PRNGs are not random in a true sense because they can be computed and are thus completely deterministic.
- A widely used

Example: *rand() function* in ANSI C:

$$s_0 = 12345$$
$$s_{i+1} = 1103515245 s_i + 12345 \bmod 2^{31}$$

❖ **Most PRNGs have bad cryptographic properties!**

Cryptographically Secure Pseudorandom Number Generators (CSPRNG)

- CSPRNGs are a special type of PRNG.
- PRNGs with an additional property:
  - Output must be **unpredictable**

- **More precisely:** Given *n* consecutive bits of output *si* , the following output bits *Sn+1* cannot be predicted (in polynomial time).

i.e, given n output bits of the key stream si ,si+1 ,...,si+n-1

it is computationally infeasible to compute the subsequent bits si+n ,si+n+1 ,...

- No polynomial time algorithm that can predict the next bit sn+1 with better than 50% chance of success.

- Needed in cryptography, in particular for stream ciphers
- Remark: There are almost no other applications that need unpredictability, whereas many (technical) systems need PRNGs.

Unconditional Security

A cipher (cryptosystem) is unconditionally or information-theoretically secure if it cannot be broken even with infinite computational resources.

One-Time Pad (OTP)

A stream cipher for which:

1) Key stream s_0 , s_1 , s_2 ,... is generated by a TRNG.
2) Every key stream bit s_i is only used once.
3) Key stream is only known to the legitimate communicating parties.

Encryption: $e_{k_i}(x_i) = x_i \oplus k_i$.

Decryption: $d_{k_i}(y_i) = y_i \oplus k_i$

❖ **OTP is unconditionally secure if and only if the key *k_i.* is used once!**

One-Time Pad (OTP)

➢ **For every ciphertext bit we get an equation of this form :**

Unconditionally secure cryptosystem:

$$y_0 = x_0 \oplus k_0$$

$$y_1 = x_1 \oplus k_1$$

$$\vdots$$

Every equation is a linear equation with two unknowns

$\Rightarrow$ for every y_i are $x_i = 0$ and $x_i = 1$ equiprobable!

$\Rightarrow$ This is true iff $k_0, k_1, \ldots$ are independent, i.e., all k_i have to be generated truly random

$\Rightarrow$ It can be shown that this systems can *provably* not be solved.

Drawbacks of OTP

1. **Key generation:**
   a) Obtaining a TRNG is difficult (but doable).
   b) Single use means the key is as long as the message, and can't be reused. Very impractical.
   e.g., For encryption of a 400 MB file, we'd need 8.400 = 3.2 Gbit of key. Can't be reused for another file.
2. **Key distribution becomes very complicated**, with very large keys that can't be reused and must be eventually destroyed.

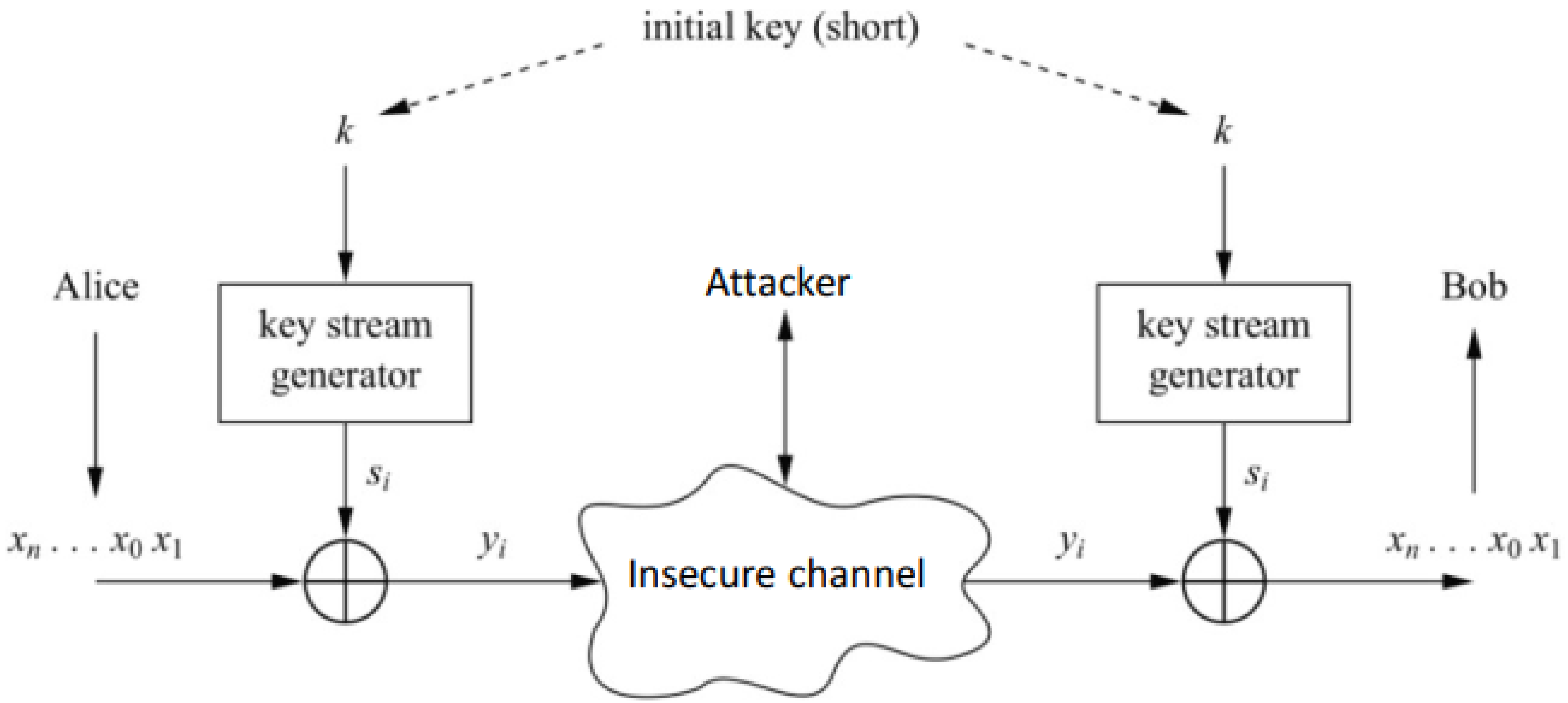
❖ For almost all applications the OTP is **impractical** since the key must be as long as the message! (Imagine you have to encrypt a 1GByte email attachment.)

Computational Security

A cipher is computationally secure if it meets one or both of the following criteria:

- The cost of breaking the cipher exceeds the value of the encrypted information.
- The time required to break the cipher exceeds the useful lifetime of the information.

Practical Stream Ciphers

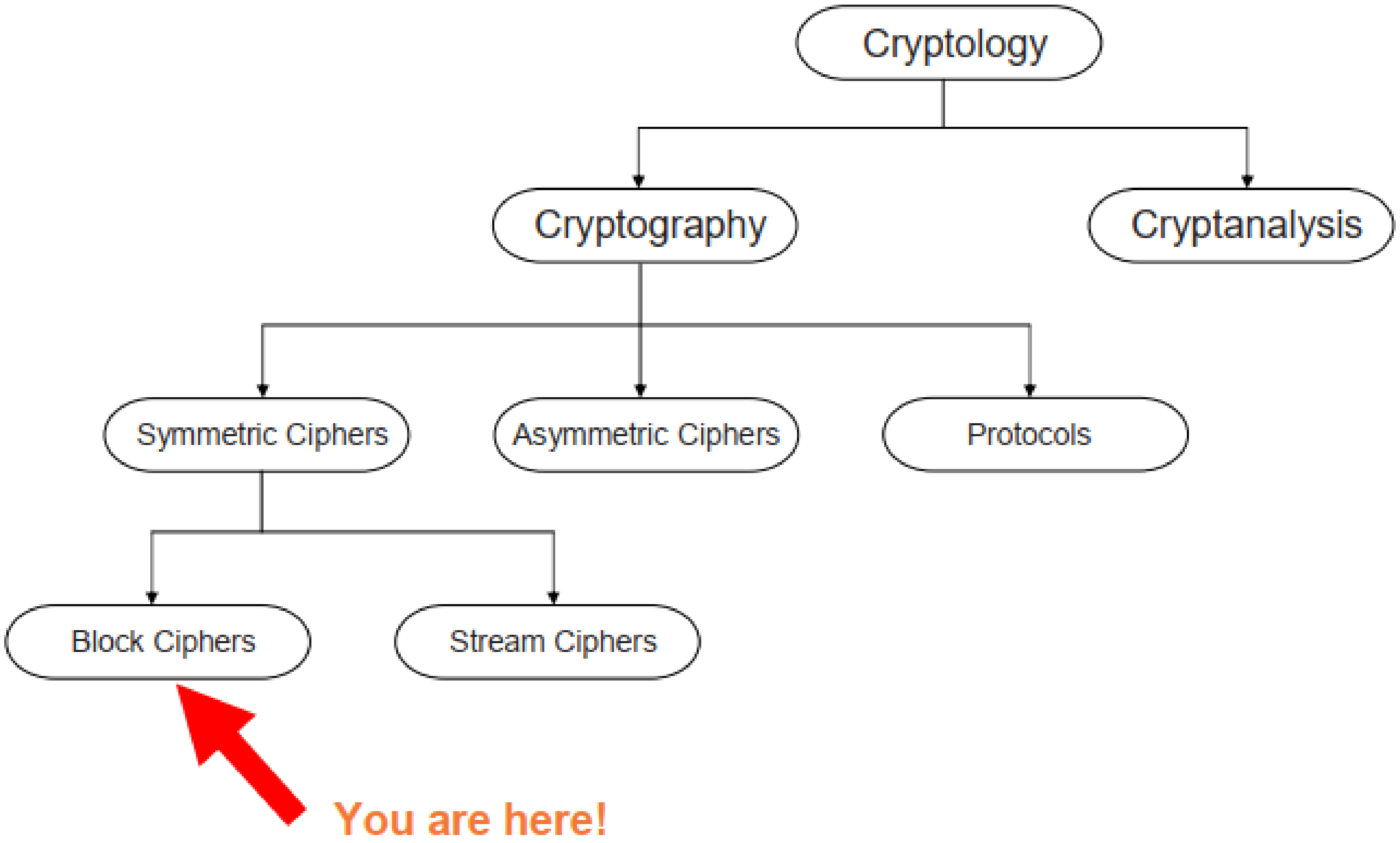


Practical Stream Ciphers

- Get key streams from PRNGs:
  - e.g., Using linear feedback shift registers (LFSR). Not cryptographically secure.

- Get key streams from CSPRNGs:
  - Using combinations of several LFSRs and nonlinear components.
  - Using block ciphers as building blocks.

Main Areas of Cryptography



Block Ciphers

- *Simplified Data Encryption Standard (SDES)*
- *Data Encryption Standard (DES)*
- *Advanced Encryption Standard (AES)*

Strong Block Encryption

- In 1945, Claude Shannon defined two basic operations to achieve strong encryption:
  • **Confusion:** an encryption operation where the relationship between key and ciphertext is hidden.
  Today, a common element for achieving confusion is **substitution**, which is found in DES

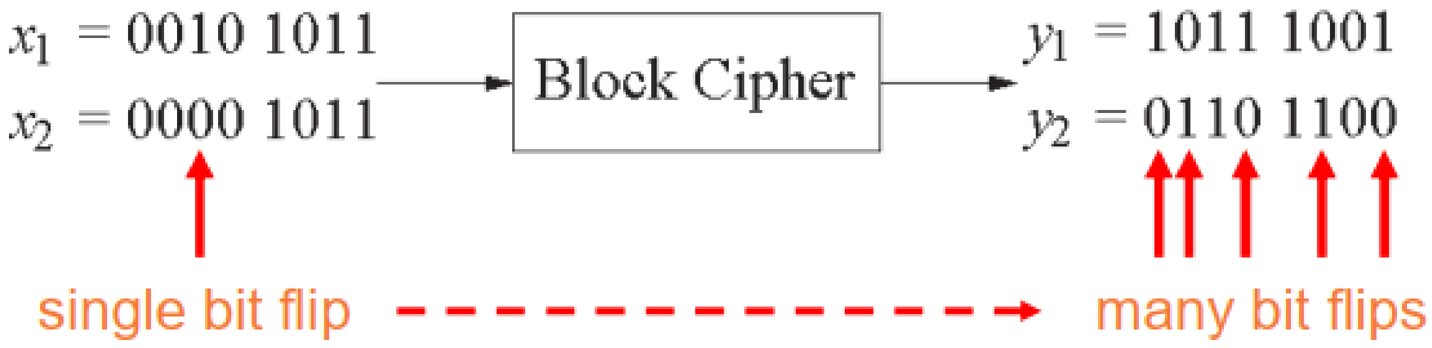
  • **Diffusion:** an encryption operation where the influence of one plaintext bit is spread over many ciphertext bits. with the goal of hiding statistical properties of the plaintext.
  A simple diffusion element is the **bit permutation**, which is frequently used within DES.

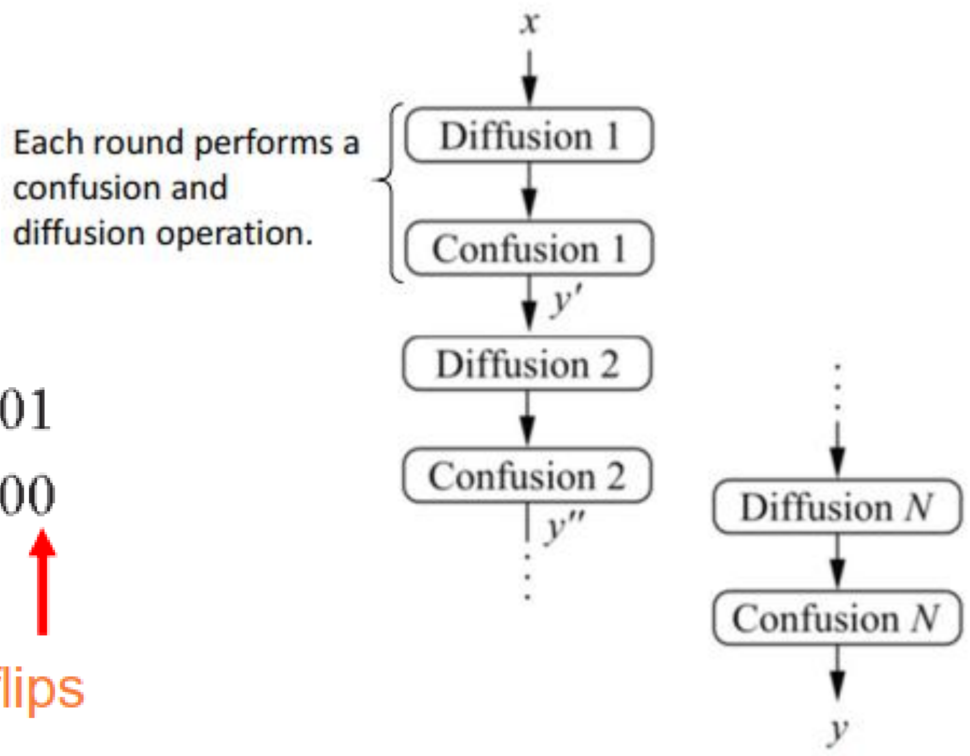
- Both operations by themselves cannot provide security. The idea is to concatenate confusion and diffusion elements to build so called ***product ciphers***

Product Ciphers

- Most of today's block ciphers are *product ciphers* as they consist of rounds which are applied repeatedly to the data.
- Can reach excellent diffusion: **changing of one bit of plaintext results** *on average* in the **change of half the output bits**.

Example: Let's assume a small block cipher with a block length of 8 bits. Encryption of two plaintexts *x*1 and *x*2, which differ only by one bit, should roughly result in something as shown :





"Principle of an N-round product cipher"

Thank You!

See You next Lectures!!

Any Question?

THE FIRST BRITISH HIGHER EDUCATION IN EGYPT

26th July Mehwar Road Intersection with Wahat Road, 6th of October City, Egypt

Tel: +202383711146 **Fax:** +20238371543 **Postal code:** 12451

Email: info@msa.eun.eg **Hotline:** 16672 **Website:** www.msa.edu.eg