جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

College of science
Department of Cyber Security

Lecture 2:

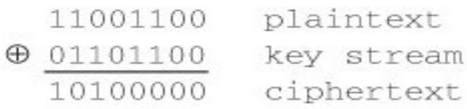
Stream Cipher Structure
Important element for design a stream cipher
Types of stream ciphers

Stage :2

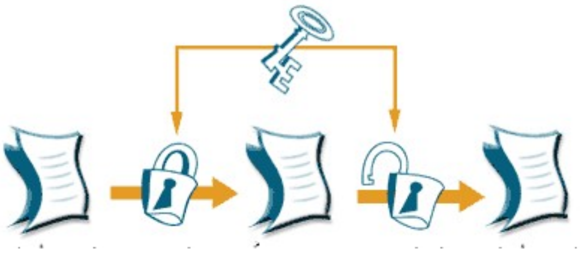
م.م مصطفى امير صبري

2.1 Stream Cipher Structure

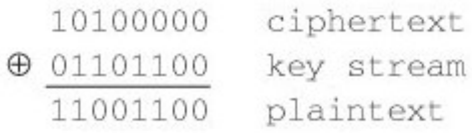
A typical stream cipher encrypts plaintext one byte at a time; although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time. In the following figure is a representative diagram of stream cipher structure. In this structure a key is input to a **pseudorandom** bit generator that produces a stream of 8-bit numbers that are apparently random. For now, we simply say that a pseudorandom stream is one that is unpredictable without knowledge of the input key. The output of the generator, called a keystream, is combined one byte at a time with the plaintext stream using the bitwise exclusive-OR (XOR) operation. For example, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting ciphertext byte is



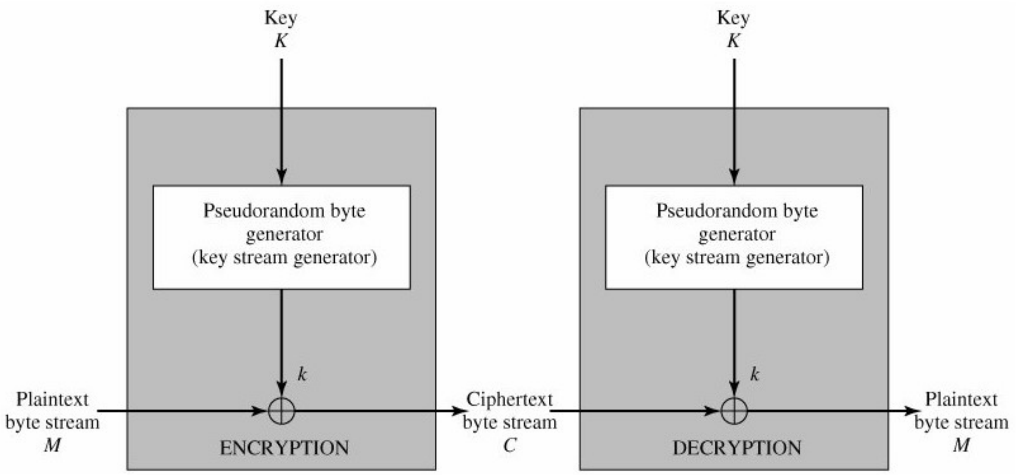
```
  11001100   plaintext
⊕ 01101100   key stream
  10100000   ciphertext
```



Decryption requires the use of the **same** pseudorandom sequence:



```
  10100000   ciphertext
⊕ 01101100   key stream
  11001100   plaintext
```


Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream
M
ENCRYPTION
Ciphertext byte stream
C
Key
K
Pseudorandom byte generator (key stream generator)
k
DECRYPTION
Plaintext byte stream
M

Stream Cipher Diagram

***Note*:** The stream cipher is similar to the **one-time pad** discussed .The difference is that a one-time pad uses a **genuine**random number stream, whereas a stream cipher uses a **pseudo**random number stream.

2.2 Important element for design a stream cipher

1. The **encryption** sequence should have a **large period.** A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. The longer the

period of repeat the more difficult it will be to do cryptanalysis. This is essentially the same consideration that was discussed with

reference to the Vigenère cipher, namely that the longer the keyword the more difficult the cryptanalysis.

2. The **keystream** should approximate the properties of a true **random** number stream as close as possible. For example, there should be an approximately equal **numbers of 1s and 0s**. If the keystream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often. The more random-appearing the keystream is, the more randomized the ciphertext is, making cryptanalysis more difficult.
3. the **output** of the pseudorandom number generator is conditioned on the value of the **input key**. To guard against brute-force attacks, the key needs to be sufficiently long. The same considerations as apply for block ciphers are valid here. Thus, with current technology, a key length of at **least 128 bits** is desirable.

Note: Pseudorandom Number Generators (PRNGs)

Cryptographic applications typically make use of algorithmic techniques for random number generation. These algorithms are deterministic and therefore produce sequences of numbers that are not statistically random. However, if the algorithm is **good**, the resulting sequences will pass many reasonable tests of randomness. Such numbers are referred to as **pseudorandom numbers**.

2.3 Types of stream ciphers

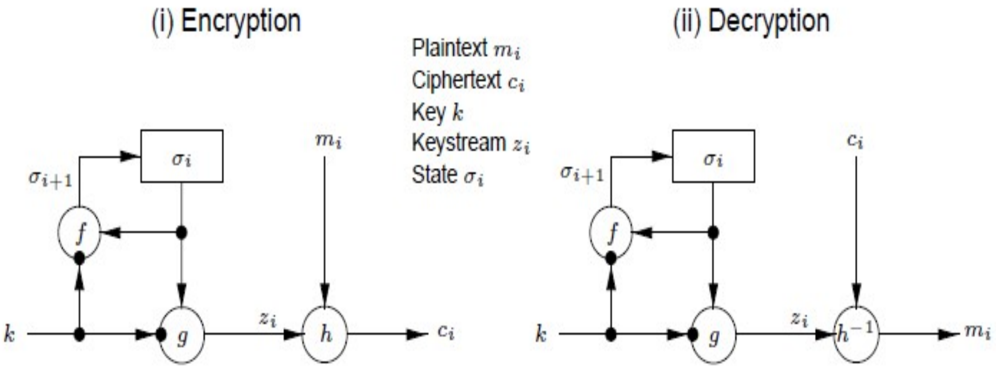
A stream cipher generates successive elements of the **keystream** based on an **internal** state. This state is updated in essentially two ways: if the state changes independently of the **plaintext or ciphertext** messages, the cipher is classified as a synchronous stream cipher. By contrast, selfsynchronising stream ciphers update their state based on previous ciphertext digits.

❖ Synchronous stream ciphers

Definition: A *synchronous* stream cipher is one in which the keystream is generated **independently** of the plaintext message and of the ciphertext. The encryption process of a synchronous stream cipher can be described by the equations:

$$\begin{aligned} \sigma_{i+1} &= f(\sigma_i, k), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned}$$

where σ_0 is the **initial** state and may be determined from the key **k**, **f** is the next-state function, **g** is the function which produces the key stream zi, and **h** is the output function which combines the keystream and plaintext **mi** to produce cipher text **ci**. The encryption and decryption processes are depicted in following figure. The **OFB** mode of a block cipher



- **Note** (*properties of synchronous stream ciphers*)

(i) ***synchronization requirements***. In a synchronous stream cipher, both the sender and receiver must be *synchronized* – using the **same key** and **operating at the same position (state)** within that key – to allow for proper decryption. If synchronization is lost due to ciphertext digits being inserted or deleted during transmission, then decryption **fails** and can only be restored through additional techniques for re-synchronization. Techniques for re-synchronization include re-initialization, placing special markers at regular intervals in the ciphertext, or, if the plaintext contains enough redundancy, trying all possible keystream offsets.

(ii) ***No error propagation.*** A ciphertext digit that is modified (but not deleted) during transmission does **not affect** the decryption of **other** ciphertext digits.

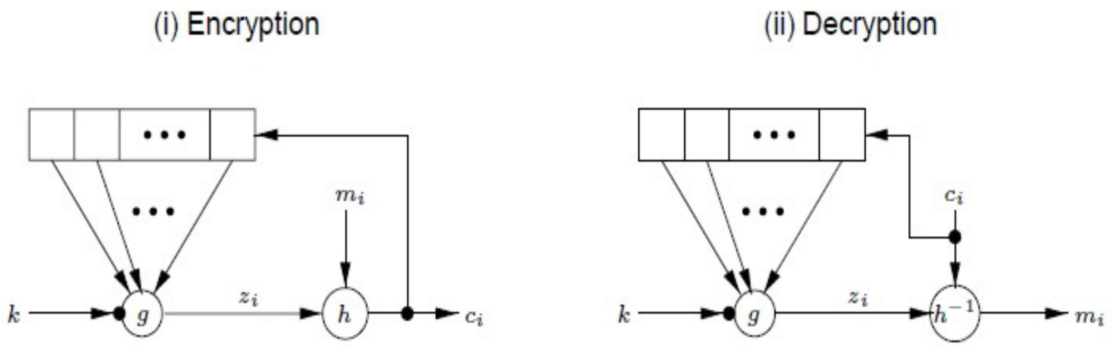
(iii) ***Active attacks.*** As a consequence of property (i), the insertion, deletion, or replay of ciphertext digits by an active adversary causes immediate loss of synchronization, and hence might possibly be detected by the decrypted.

❖ Self-synchronizing stream ciphers

Definition: A *self-synchronizing* or *asynchronous* stream cipher is one in which the **keystream** is generated as a function of the key and a **fixed number of previous ciphertext** digits. The encryption function of a selfsynchronizing stream cipher can be described by the equations:

$$\begin{aligned} \sigma_i &= (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned}$$

where σ_0 = (c−t; c−t+1; .... ; c−1) is the (non-secret) *initial state*, **k** is the *key*, **g** is the function which produces the *keystream* **zi**, and **h** is the *output function* which combines the **keystream and plaintext** mi to produce ciphertext ci. The encryption and decryption processes are depicted in following Figure. The most common presently-used self synchronizing stream ciphers are based on block ciphers in 1-bit cipher feedback mode


(i) Encryption

(ii) Decryption

Note (*properties of self-synchronizing stream ciphers*)

(i) ***self-synchronization***. Self-synchronization is possible if ciphertext digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding **ciphertext characters**. Such ciphers are capable of re-establishing proper decryption automatically after loss of synchronization, with only a fixed number of plaintext characters unrecoverable.

(ii) ***Limited error propagation.*** Suppose that the state of a selfsynchronization stream cipher depends on **t** previous ciphertext digits. If a single ciphertext digit is **modified** (or even deleted or inserted) during transmission, then decryption of up to **t** subsequent ciphertext digits may be **incorrect**, after which correct decryption resumes.

(iii) ***Active attacks.*** Property (ii) implies that any modification of ciphertext digits by an active adversary causes several other ciphertext digits to be decrypted incorrectly, thereby improving (compared to synchronous stream ciphers) the likelihood of being detected by the decryptor. As a consequence of property (i), **it is more difficult** (than for synchronous stream ciphers) to **detect insertion, deletion, or replay of ciphertext digits** by an active adversary. This illustrates that additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees.

(iv) ***Diffusion of plaintext statistics.*** Since each plaintext digit influences the entire following ciphertext, the statistical properties of the plaintext are dispersed through the ciphertext. Hence, self-synchronizing stream ciphers may be more **resistant** than synchronous stream ciphers against attacks based on plaintext redundancy.