Lecture A1:

Stream Ciphers and RC4 algorithm

4th Year Course- CCSIT, UoA

Lecture goals

- To discuss stream ciphers
- To review RC4 stream cipher algorithm

Stream Ciphers (1)

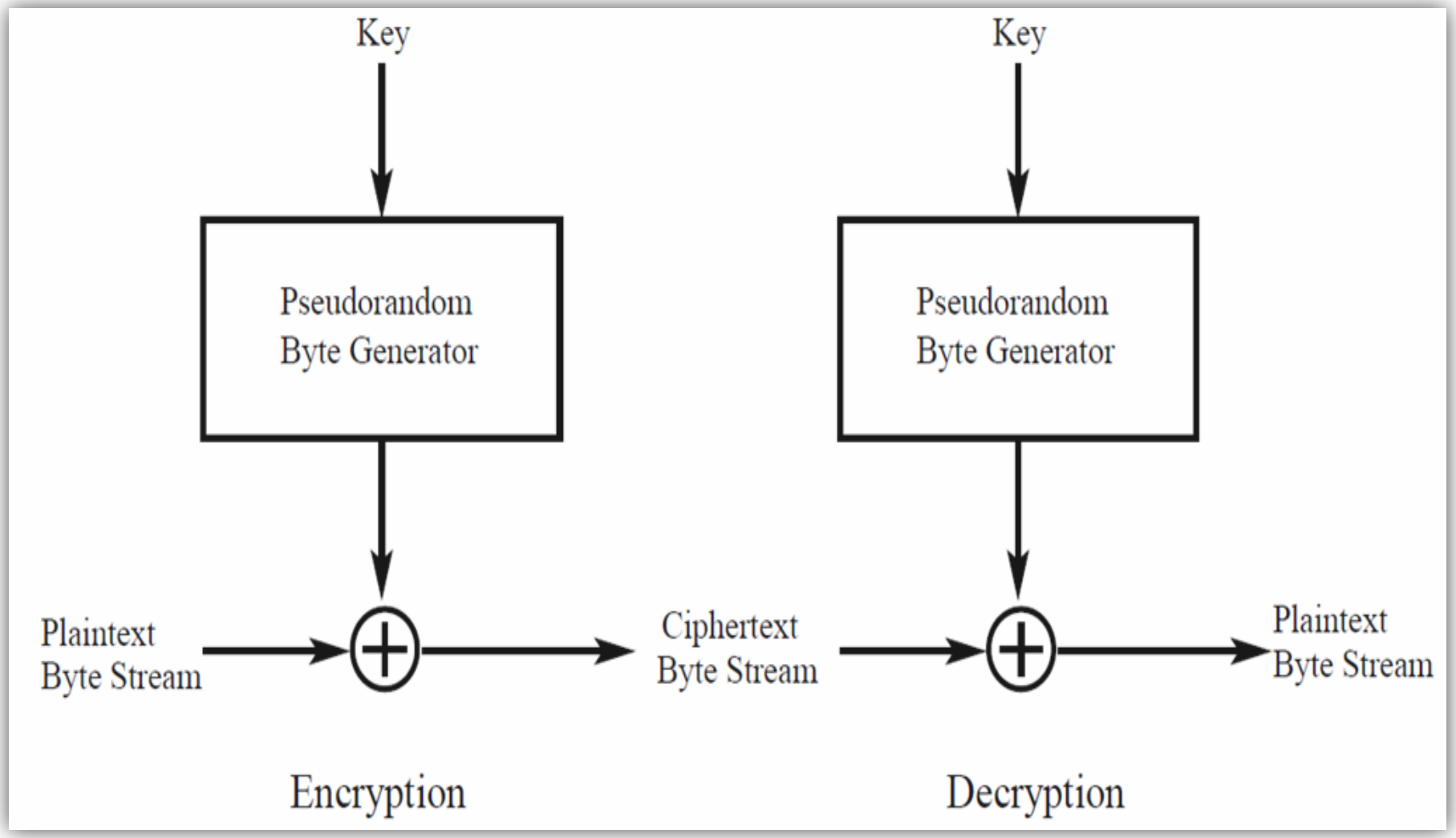
- Previously we showed how a block cipher, when used in the CFB and OFB modes, can be deployed as a stream cipher.
- So this is a good time to focus on what is generally meant by a stream cipher and to talk about **real stream cipher** algorithms.
- A typical stream cipher encrypts plaintext one byte at a time.
- The **main processing step** in a general stream cipher is the generation of a stream of pseudorandom bytes starting with the encryption key.

Stream Ciphers (2)

- As a new byte of plaintext shows up for encryption, a new byte of the pseudorandom stream also becomes available at the same time and this happens on a continuous basis.
- Obviously, each different encryption key will result in a different stream of pseudorandom bytes. But for a given encryption key, the stream of pseudorandom bytes will be the same at the both the encryption end and the decryption end of a data link.

Stream Ciphers (3)

- Encryption itself is as simple as it can be. You just XOR the byte from the pseudorandom stream with the plaintext byte to get the encrypted byte.
- You generate the same pseudorandom byte stream for decryption. The decryption itself consists of XORing the received byte with the pseudorandom byte.
- The encryption is shown in the left half and the decryption in the right half of the figure in the next slide.


Key
Pseudorandom
Byte Generator
Plaintext
Byte Stream
Encryption
Key
Pseudorandom
Byte Generator
Ciphertext
Byte Stream
Plaintext
Byte Stream
Decryption

Stream Ciphers (4)

- For a stream cipher to be secure, the pseudorandom sequence of bytes should have as long a period as possible. (Note that every pseudorandom number generator produces a seemingly random sequence that eventually repeats.) The longer the period, the more difficult it is to break the cipher.
- Within the periodicity limitations of a pseudorandom byte sequence generator, the sequence should be as random as possible. From a statistical point, that means that all of the 256 8-bit patterns should appear in the sequence equally often

Stream Ciphers (5)

- Additionally, the byte sequence should be as uncorrelated as possible. This means, for example, that for any two given bytes, the probability of their appearing together should be no greater than what is dictated by their appearance as individual bytes.
- The pseudorandom byte sequence is a function of the encryption key. To foil brute-force attacks, the encryption key should be as long as possible, subject to, of course, all the other practical constraints. A desirable key length these days is 128 bits.
- With a properly designed pseudorandom byte generator, a stream cipher for a given key length can be as secure as a block cipher using keys of the same length.

Stream Ciphers (6)

- The next section presents pseudorandom byte generation for the **RC4 stream cipher**. [For more details about the subject of pseudorandom number generation for general cryptographic applications, Please refer to the textbook.]
- As you would expect, a stream cipher is particularly appropriate for **audio and video streaming**. A stream cipher could also be used for **browser-web-server links**. A block cipher, on the other hand, may be more appropriate for file transfer, etc.