BLOCK CIPHERS

B Tech CSE- IV Sem

TCS 491

Introduction to Cryptography

Jan- Jun 2022

Introduction:

- Digital data is represented in strings of binary digits (bits) unlike alphabets. Modern cryptosystems need to process this binary strings to convert into another binary string.
- Based on how these binary strings are processed, a symmetric encryption schemes can be classified into −
- Block Ciphers
- Stream Ciphers

- Both the Stream Cipher and Block Cipher belong to a part of the symmetric key cipher. Both of these are basically the methods that one can use for converting any given set of plain texts into a Cipher text. There’s a major difference between Block Cipher and Stream Cipher. The Block Cipher primarily performs the conversion of plain text into a cipher text by taking one block of plain text at a given time. On the other hand, the Stream Cipher can perform this kind of conversion by taking a single byte of plain text at a given time.

Block Ciphers

- In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of ciphertext bits. The number of bits in a block is fixed. For example, the schemes DES and AES have block sizes of 64 and 128, respectively.

Stream Ciphers

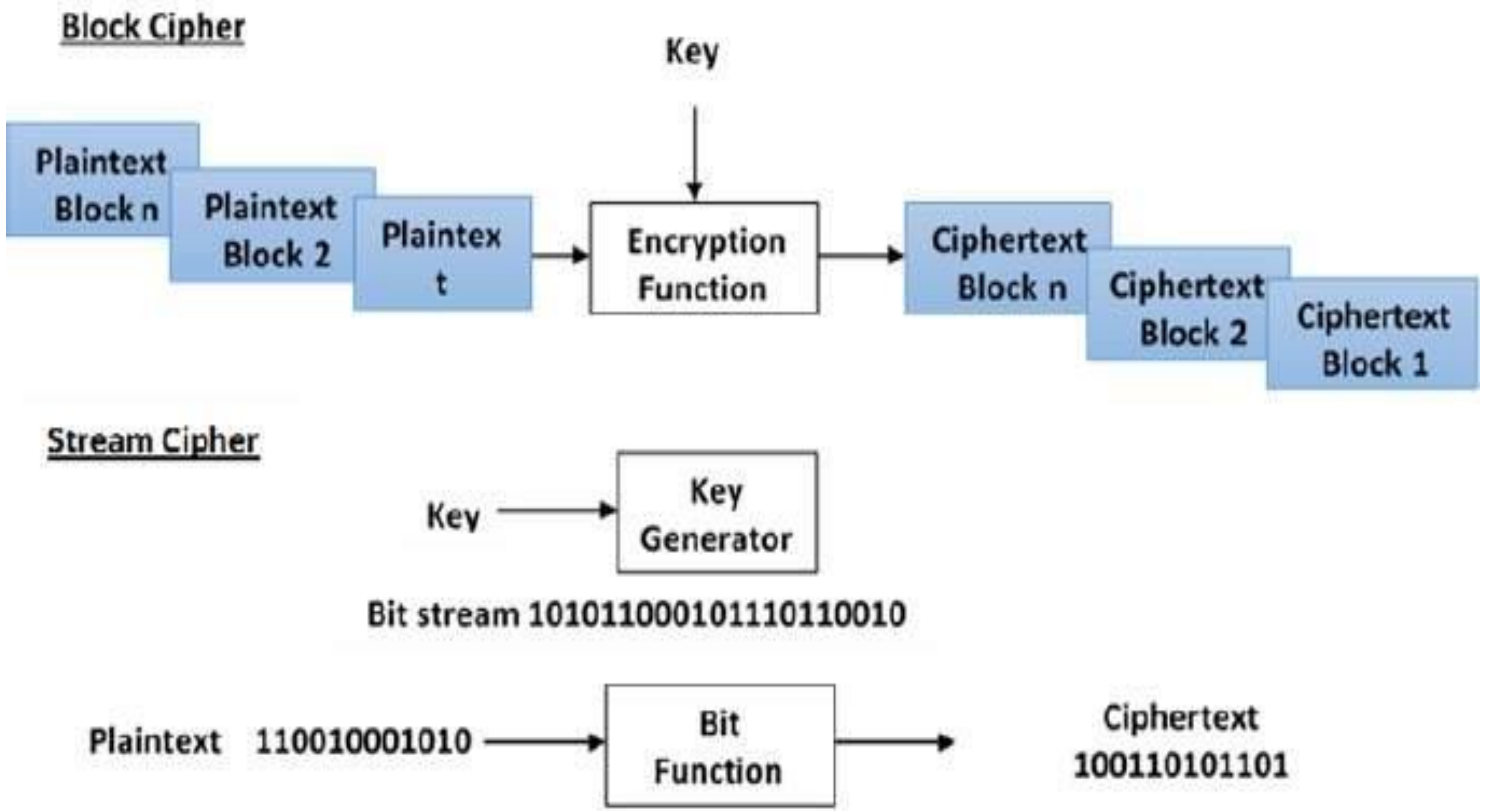
- In this scheme, the plaintext is processed one bit at a time i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of ciphertext. Technically, stream ciphers are block ciphers with a block size of one bit.

Difference Between Block Cipher and Stream Cipher

Parameters	**Block Cipher**	**Stream Cipher**
Definition	It is a type of encryption that performs the conversion of a plain text by taking a single block at a given time.	It is a type of encryption that performs the conversion of plain text by taking a byte of plain text at a time.
Principle	It makes use of both the diffusion and confusion principles for the conversion (used later in encryption).	It only makes use of the confusion principle to carry out the process of conversion.
Conversion of Bits	The Block Cipher converts one block of plain text at a given time. Thus, it is capable of converting comparatively more bits than the Stream Cipher. As a result, it converts about 64 bits at a time or more.	The Stream Cipher, on the other hand, is capable of converting a minimum of 8 bits at any given time.
Algorithm	The Block Cipher makes use of the ECB (Electronic Code Block) and the CBC (Cipher Block Chaining) algorithms to perform the encryption of plain text.	The Stream Cipher, on the other hand, makes use of the Output Feedback (OFB) and the Cipher Feedback (CFB) algorithms to perform the process of conversion.

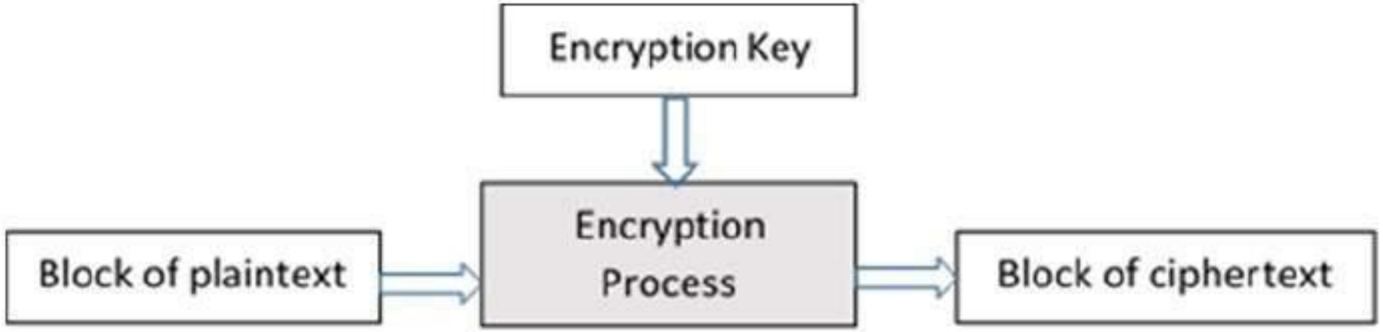
Difference Between Block Cipher and Stream Cipher

Parameters	Block Cipher	Stream Cipher
Decryption	The Block Cipher generally encrypts a combination of more bits. Thus, the process of decryption or reverse encryption becomes way more complex here as compared to the Stream Cipher.	The Stream Cipher, on the other hand, makes use of the process of XOR for encryption. One can easily reverse it to plain text at any given time.
Implementation	The primary implementation of the Block Cipher takes place in the Fiestal Cipher.	The main implementation of the Stream Cipher, on the other hand, takes place in the Vernam Cipher.
Complexity	The Block Cipher has a comparatively simple structure and design.	The Stream Cipher has a comparatively more complicated and complex design of its own.
Speed	The Block Cipher is very slow as compared to the Stream Cipher.	The Stream Cipher, on the other hand, works comparatively much faster than the Block Cipher.
Working Techniques	The Block Cipher basically works on the techniques (of transportation) like the columnar transportation technique, the rail-fence technique, etc.	The Stream Cipher, on the other hand, works on various substitute techniques such as the Polygram Substitution Cipher, the Caesar Cipher, etc.


Block Cipher
Key
Plaintext Block n
Plaintext Block 2
Plaintex t
Encryption Function
Ciphertext Block n
Ciphertext Block 2
Ciphertext Block 1
Stream Cipher
Key
Key Generator
Bit stream 10101100010111011001 0
Plaintext 110010001010
Bit Function
Ciphertext 100110101101

Block Cipher

- The basic scheme of a block cipher is depicted as follows −



A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The choice of block size does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length.

Block Size

- Though any size of block is acceptable, following aspects are borne in mind while selecting a size of a block.
- **Avoid very small block size** − Say a block size is m bits. Then the possible plaintext bits combinations are then 2^m. If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of ‘dictionary attack’ by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.
- **Do not have very large block size** − With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.
- **Multiples of 8 bit** − A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits.

Padding in Block Cipher

- Block ciphers process blocks of fixed sizes (say 64 bits). The length of plaintexts is mostly not a multiple of the block size. For example, a 150-bit plaintext provides two blocks of 64 bits each with third block of balance 22 bits. The last block of bits needs to be padded up with redundant information so that the length of the final block equal to block size of the scheme. In our example, the remaining 22 bits need to have additional 42 redundant bits added to provide a complete block. The process of adding bits to the last block is referred to as **padding**.
- Too much padding makes the system inefficient. Also, padding may render the system insecure at times, if the padding is done with same bits always.

Block Cipher Schemes

- There is a vast number of block ciphers schemes that are in use. Many of them are publically known. Most popular and prominent block ciphers are listed below.
- **Digital Encryption Standard (DES)** − The popular block cipher of the 1990s. It is now considered as a ‘broken’ block cipher, due primarily to its small key size.
- **Triple DES** − It is a variant scheme based on repeated DES applications. It is still a respected block ciphers but inefficient compared to the new faster block ciphers available.
- **Advanced Encryption Standard (AES)** − It is a relatively new block cipher based on the encryption algorithm **Rijndael** that won the AES design competition.
- **Twofish** − This scheme of block cipher uses block size of 128 bits and a key of variable length. It was one of the AES finalists. It is based on the earlier block cipher Blowfish with a block size of 64 bits.
- **Serpent** − A block cipher with a block size of 128 bits and key lengths of 128, 192, or 256 bits, which was also an AES competition finalist. It is a slower but has more secure design than other block cipher.

- **The CAST Block Cipher −** The CAST Block Cipher is an advancement of the DES block cipher, introduced in Canada by Carlisle Adams and Stafford Tavares. The name of the cipher appear to be after the initials of the inventors. The CAST algorithm has 64 bit block size and has a key of size 64 bits.
- CAST is based on the Feistel structure to perform the substitution permutation network. The authors defines that they need the Feistel mechanism, as it is well considered and free of basic structural weaknesses.

- **Blowfish** − Blowfish is a 64-bit block cipher introduced by Bruce Schneier. Blowfish was designed for quick ciphering on 32-bit microprocessors. Blowfish is also solid and has a variable key length which can be enhanced to 448 bits.
- Blowfish is accessible for applications where the key does not modify generally like communication links or file encryptor. However for software like packet switching or as a one-way hash function, it is not proper.
- Blowfish is not perfect for smart cards, which needed even more compact ciphers. Blowfish is quicker than DES when performed on 32-bit microprocessors.

- **IDEA** − IDEA stands for International Data Encryption Algorithm. It is another block cipher. It works on 64 bit data blocks and the key is 128 bit long. It was introduced by Xuejia Lai and James Massey, and named IDEA in 1990, after changing and enhancing the original proposal of the cipher based on the seminal work on Differential cryptanalysis by Biham and Shamir.
- The design principle behind IDEA is the combining of arithmetical operations from different algebraic sets. These arithmetical operations are simply performed both in hardware and software.
- IDEA has a very simple key scheduling. It creates the 128 bit key and divides it into eight 16 bit blocks. The first six blocks are utilized for the first round, while the remaining two are to be utilized for the second round. Thus the whole 128 bit key is given a rotation for 25 steps to the left and again divided into eight blocks.
- The first four blocks are utilized as the remaining subkeys for the second round, while the last four blocks are to be utilized for the third round. The key is given a left shift by 25 bits, and the another subkeys are acquired. The procedure is continued till the end of the algorithm.
- It is a sufficiently strong block cipher with a block size of 64 and a key size of 128 bits. A number of applications use IDEA encryption, including early versions of Pretty Good Privacy (PGP) protocol. The use of IDEA scheme has a restricted adoption due to patent issues.

- **RC5 −** RC5 was invented by Ron Rivest. It is a block encryption algorithm depends on the symmetric key. The main feature of this it is quite fast as it needs only primitive computer operations. It enables a variable number of rounds and variable bit size key to insert flexibility.
- Another benefit of using RC5 is that it needed less memory for implementation. This feature allows RC 5 to be used for several purposes such as desktop operation, smart cards, etc.

Thank You
&
Best Wishes