**Department of Information Technology**

**Subject: Cryptography and Network Security Lab (DITL504)**

**Semester: V (TEIT)**

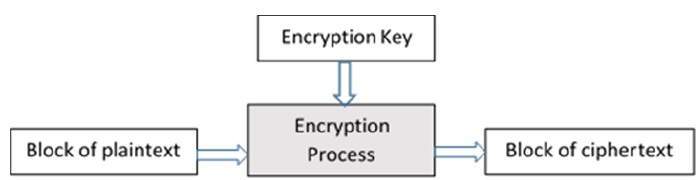
**Practical No 4**

**Aim:**

Analysis of Modern Block Ciphers (use crypt APIs)

**Theory:**

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.



**Analysis:**

1. Use crypt API to encrypt/decrypt a plaintext block using AES, DES, Blowfish, RC5
2. Avalanche Effect : Change in Plaintext
3. Avalanche Effect : Change in key

**Observations:**

Note: the key used was 1 which is 1 padded by required amount of bytes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RC4 | Blowfish | AES-128 | DES |
| plaintext | 123456 | 123456 | 123456 | 123456 |
| ciphertext | RyuhSx0i | /P5oH3xt0fA= | geG2qd3T1MxlM3Nj3aLY7A== | U6VKCXo4URE= |
| Etime | 305800 ns | 825900 ns | 698800 ns | 701299 ns |
| Dtime | 201200 ns | 252200 ns | 168000 ns | 171001 ns |

Avalanche Effect ::

Changes in Plaintext: (key is constant: 1 (padded))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm > | RC4 | Blowfish | AES | DES |
| Ciphertext  (  plaintext:  123456  ) | RyuhSx0i | /P5oH3xt0fA= | geG2qd3T1MxlM3Nj3aLY7A== | U6VKCXo4URE= |
| Ciphertext  (  plaintext:  122456  ) | RyugSx0i | npPV0aDEzd4= | pbY+uvTJtM64Div0fDtw6g== | 9QJrXZOU9hU= |
| No of bits changed | 1 | 15 # | 7 | 8 |

Changes in Key: (plaintext is constant: 123456)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Avalanche effect (1 bit change in key) | RC5 | Blowfish | AES | DES |
| Ciphertext  (key : 1) | RyuhSx0i | /P5oH3xt0fA= | geG2qd3T1MxlM3Nj3aLY7A== | U6VKCXo4URE= |
| Ciphertext  (key: 2) | 0uYI1iTU | tFGP9W2JBvY= | fSDSOOvU7xy1WiQN6mjDDQ== | yPuwqmsWdho= |
| No of bits changed |  |  |  |  |

**Conclusion:** based on amount of time taken for encryption/decryption comment wrt to performance

Which algo exhibits better avalanche effect wrt to change in plaintext

Which algo exhibits better avalanche effect wrt to change in key