**Experiment 5**

BlowFish

**Name:** Kashish Jain

**UID:** 2019130022

**Class:** TE Comps

**Aim:** To implement blowfish algorithm.

**THEORY**

**BLOWFISH ALGORITHM:**

[Blowfish is a symmetric-key](https://en.wikipedia.org/wiki/Symmetric-key_algorithm) [block cipher](https://en.wikipedia.org/wiki/Block_cipher), designed in 1993 by [Bruce Schneier](https://en.wikipedia.org/wiki/Bruce_Schneier) and included in many cipher suites and encryption products. Blowfish provides a good [encryption rate in software and no effective cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis) of it has been found to date.

Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative [to the aging DES](https://en.wikipedia.org/wiki/Data_Encryption_Standard) and free of the problems and constraints associated with other algorithms. At the time Blowfish was released, many other designs were proprietary, encumbered by [patents](https://en.wikipedia.org/wiki/Patent) or were commercial or government secrets. Schneier has stated that, "Blowfish is unpatented, and will remain so in all countries. The algorithm is hereby placed in the [public domain](https://en.wikipedia.org/wiki/Public_domain), and can be freely used by anyone."

[Notable features of the design include key-dependent S-boxes](https://en.wikipedia.org/wiki/S-box) and a highly complex [key schedule](https://en.wikipedia.org/wiki/Key_schedule).

Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use.

Blowfish uses:

* **blockSize**: 64-bits
* **keySize**: 32-bits to 448-bits variable size
* **number of subkeys**: 18 [P-array]
* number of rounds: 16
* **number of substitution boxes**: 4 [each having 512 entries of 32-bits each

Original encoded message

Graphical user interface, application

Description automatically generated with medium confidence

1)If you change one character at the end of the message, the encoded message changes in the following way:

Graphical user interface

Description automatically generated with low confidence

After changing the last character of plain text message, last 16 characters of the encrypted message changes, and the rest of the encrypted message remains same.

2)If you change one character at the beginning of the message, the encoded message changes as follows:

Graphical user interface, text, application

Description automatically generated

After changing the first character of plain text message, first 16 characters of the encrypted message changes, and the rest of the encrypted message remains same.

3)If you delete one character at the end of the message, the encoded message changes as follows:

Graphical user interface, text, application

Description automatically generated

After deleting last character of plain text message, last 16 characters of the encrypted message changes, and the rest of the encrypted message remains same.

Size still remains the same since ECB is used which is a block cipher.

4)If you change one character in a key, the encoded message changes as follows:

Graphical user interface, application

Description automatically generated with medium confidence

After changing one character in a key, entire encrypted message changes. Still the size of the encrypted message remains the same since the key length is the same.

**Github Links**

Repository

<https://github.com/kashishvjain/CSS-Lab>

5)Decrypt a message using a key with one character changed. Does it look anything like the original?

Graphical user interface, text, application

Description automatically generated

1. Here key used is donno and message is original encoded message.
2. It does not look like original message and decrypted message consists of lots of special characters.

**CONCLUSION**

1. Studied and understood the Blowfish algorithm.
2. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for both domestic and exportable use.
3. Blowfish is considered to be a block Cipher since changing one text alters that section of the block encryption.
4. It is also a symmetric cypher because it encrypts and decrypts with the same key. Any change in key causes the ciphered text to be incorrectly deciphered.