Experiment 5

BlowFish

Name: Ameya Jangam
UID: 2019130025
Class: TE Comps

Aim: To implement blowfish algorithm.

THEORY

BLOWFISH ALGORITHM:

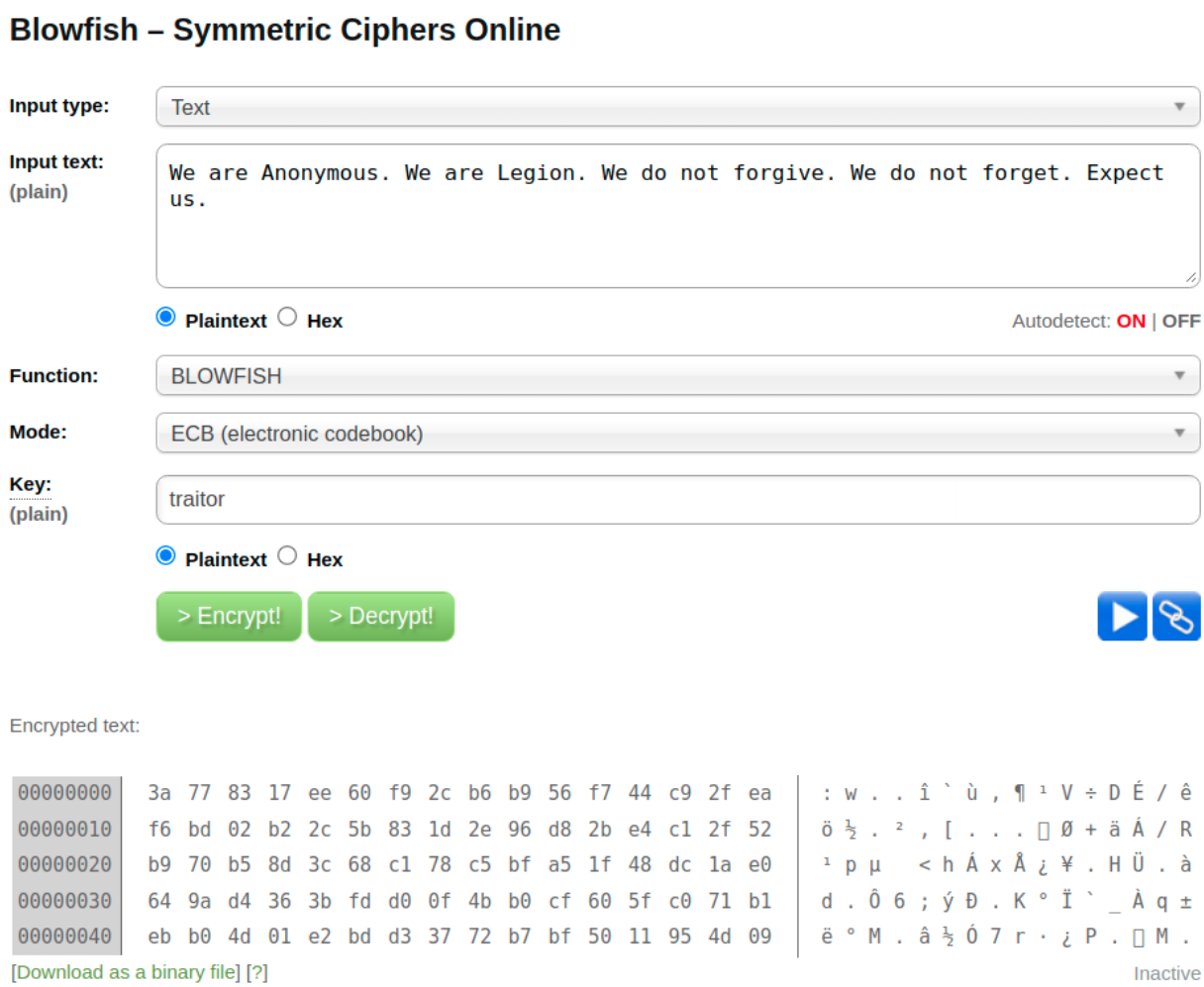
Blowfish is a symmetric-key block cipher, designed in 1993 by Bruce Schneier and included in many cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date.
Schneier designed Blowfish as a general-purpose algorithm, intended as an alternative to the aging DES 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 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, and can be freely used by anyone."
Notable features of the design include key-dependent S-boxes and a highly complex 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 small and large use cases.

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

- I used http://blowfish.online-domain-tools.com/ for the experiment.

Original encoded message



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

Blowfish – Symmetric Ciphers Online

Input type: Text

Input text: (plain)

```
We are Anonymous. We are Legion. We do not forgive. We do not forget. Expect
um.
```

Plaintext **Hex** Autodetect: **ON** | **OFF**

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) traitor

Plaintext **Hex**

> Encrypt! > Decrypt!

Encrypted text:

```
00000000  3a 77 83 17 ee 60 f9 2c b6 b9 56 f7 44 c9 2f ea  : w . . î ` ù , ¶ ¹ V ÷ D É / ê
00000010  f6 bd 02 b2 2c 5b 83 1d 2e 96 d8 2b e4 c1 2f 52  ö ½ . ² , [ . . . □ Ø + ä Á / R
00000020  b9 70 b5 8d 3c 68 c1 78 c5 bf a5 1f 48 dc 1a e0  ¹ p µ   < h Á x Å ¿ ¥ . H Ü . à
00000030  64 9a d4 36 3b fd d0 0f 4b b0 cf 60 5f c0 71 b1  d . Ô 6 ; ý Ð . K ° Ï ` _ À q ±
00000040  eb b0 4d 01 e2 bd d3 37 a3 49 a9 d6 0b c5 31 a7  ë ° M . â ½ Ó 7 £ I © Ö . Å 1 §
```

[Download as a binary file] [?] Inactive

After changing the last character of a plain text message, the last 16 characters of the encrypted message change, and the rest of the encrypted message remains the same.
2)If you change one character at the beginning of the message, the encoded message changes as follows:

Blowfish – Symmetric Ciphers Online

Input type: Text

Input text: (plain)

```
Te are Anonymous. We are Legion. We do not forgive. We do not forget. Expect
um.
```

(●) **Plaintext** ( ) **Hex** Autodetect: **ON** | **OFF**

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) traitor

(●) **Plaintext** ( ) **Hex**

> Encrypt! > Decrypt!

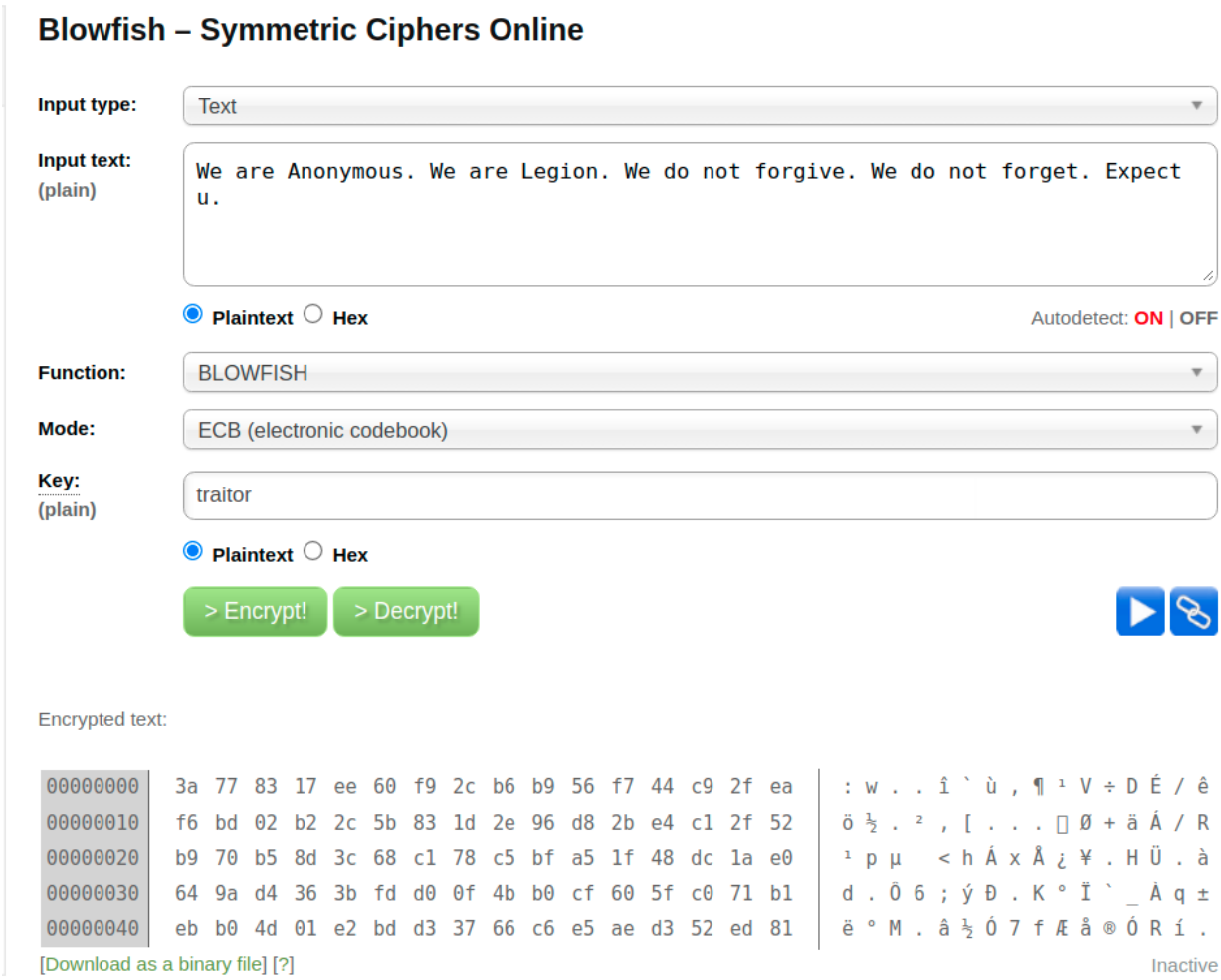
Encrypted text:

```
00000000  b2 9b 6a af 06 d6 8c f4 b6 b9 56 f7 44 c9 2f ea  ² . j ¯ . Ö . ô ¶ ¹ V ÷ D É / ê
00000010  f6 bd 02 b2 2c 5b 83 1d 2e 96 d8 2b e4 c1 2f 52  ö ½ . ² , [ . . . □ Ø + ä Á / R
00000020  b9 70 b5 8d 3c 68 c1 78 c5 bf a5 1f 48 dc 1a e0  ¹ p µ   < h Á x Å ¿ ¥ . H Ü . à
00000030  64 9a d4 36 3b fd d0 0f 4b b0 cf 60 5f c0 71 b1  d . Ô 6 ; ý Ð . K ° Ï ` _ À q ±
00000040  eb b0 4d 01 e2 bd d3 37 a3 49 a9 d6 0b c5 31 a7  ë ° M . â ½ Ó 7 £ I © Ö . Å 1 §
```

[Download as a binary file] [?] Inactive

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

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



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

Blowfish – Symmetric Ciphers Online

Input type: Text

Input text: (plain)

```
We are Anonymous. We are Legion. We do not forgive. We do not forget. Expect us.
```

Plaintext **Hex** Autodetect: **ON** | **OFF**

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) draitor

Plaintext **Hex**

> Encrypt! > Decrypt!

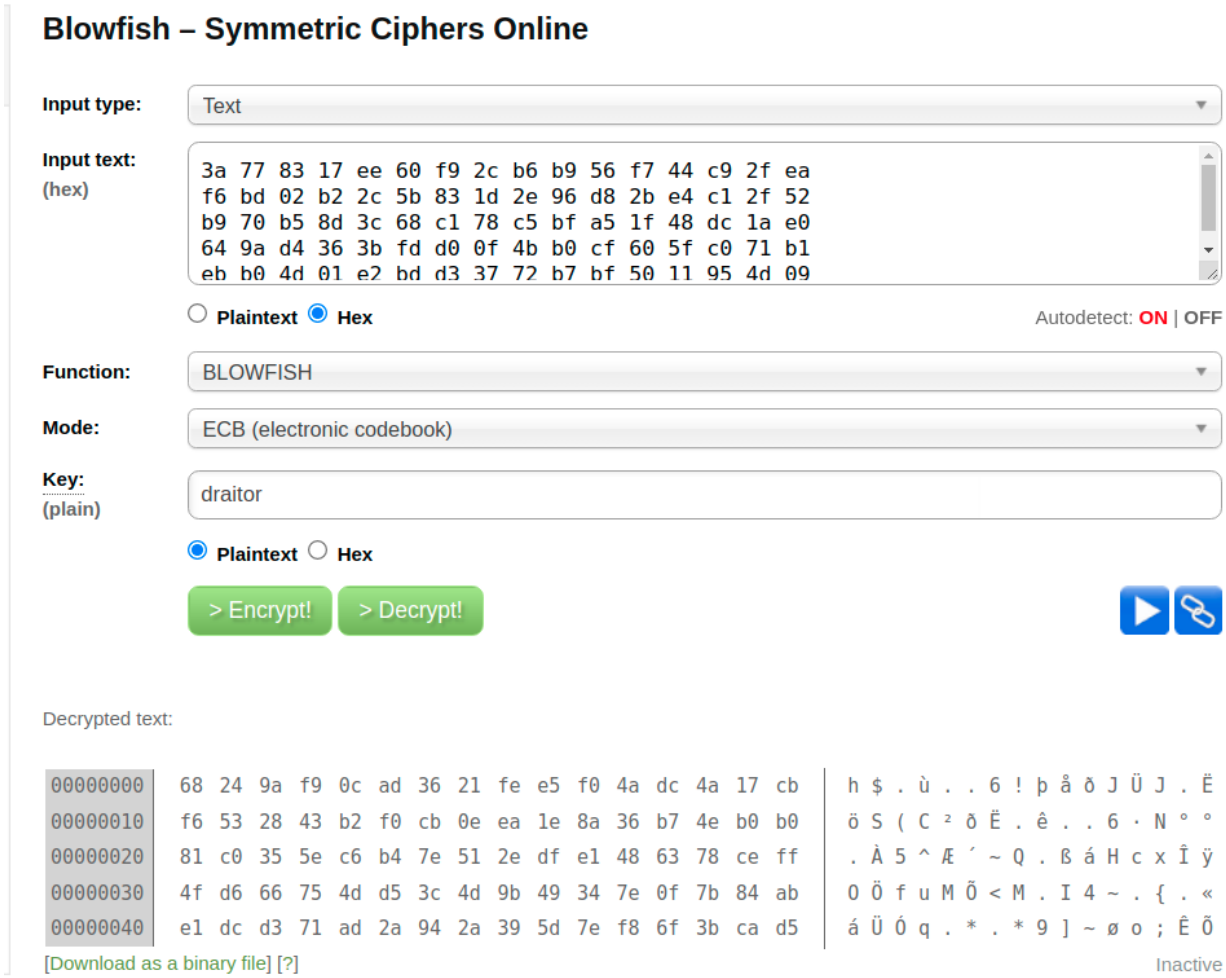
Encrypted text:

```
00000000  86 6d ae 73 5f da 6e d7 9b a2 49 6e 56 19 45 d6  . m ® s _ Ú n × . ¢ I n V . E Ö
00000010  50 64 63 f6 ed 05 01 3d f4 c3 a7 2a d1 f5 3d cd  P d c ö í . . = ô Ã § * Ñ õ = Í
00000020  56 ac 14 2f 11 99 75 23 fe 54 3a 55 a2 57 bd 95  V ¬ . / . . u # þ T : U ¢ W ½ □
00000030  f3 71 00 fb 63 94 17 b2 31 61 1e ee 3b 37 9f ad  ó q . û c . . ² 1 a . î ; 7 . .
00000040  18 f6 65 61 ab 1f 3a 74 10 24 62 41 19 fb fa 68  . ö e a « . : t . $ b A . û ú h
```

[Download as a binary file] [?] Inactive

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

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



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

Encoding text which will be sent via mail

Input type: Text

Input text: (plain)

```
That's cool! But can you do this?
```

Plaintext Hex

Autodetect: ON | OFF

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) ameya

Plaintext Hex

> Encrypt! > Decrypt!

Encrypted text:

```
00000000  e8 ae 91 49 cc e8 b2 71 17 b8 c0 0d a8 29 04 cf  | è ® ▯ I Ì è ² q . ¸ À . ¨ ) . Ï
00000010  f4 f3 dd 26 06 e0 54 e3 26 bd 3d e8 2b 10 37 a9  | ô ó Ý & . à T ã & ½ = è + . 7 ©
00000020  78 2a ab 5c 3f e1 d3 99                          | x * « \ ? á Ó .
```

[Download as a binary file] [?]

Inactive

Decrypt2 External Inbox ×

Ameya Jangam

to me

```
e8 ae 91 49 cc e8 b2 71 17 b8 c0 0d a8 29 04 cf
f4 f3 dd 26 06 e0 54 e3 26 bd 3d e8 2b 10 37 a9
78 2a ab 5c 3f e1 d3 99
```

Reply Forward

Blowfish – Symmetric Ciphers Online

Input type: Text

Input text: (hex)

```
e8 ae 91 49 cc e8 b2 71 17 b8
c0 0d a8 29 04 cf
f4 f3 dd 26 06 e0 54 e3 26 bd
3d e8 2b 10 37 a9
78 2a ab 5c 3f e1 d3 99
```

○ **Plaintext** ◉ **Hex** Autodetect: **ON** | **OFF**

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) ameya

◉ **Plaintext** ○ **Hex**

> Encrypt! > Decrypt!

Decrypted text:

```
00000000  54 68 61 74 27 73 20 63 6f 6f 6c 21 20 42 75 74  | That's cool! But
00000010  20 63 61 6e 20 79 6f 75 20 64 6f 20 74 68 69 73  | can you do this
00000020  3f 00 00 00 00 00 00 00                          | ?.......
```

[Download as a binary file] [?] Inactive

Checkout ?

CONCLUSION

1. I explored most of the variations that I could in the blowfish algorithm, that is by altering the text with the key in different ways and also altering with the key and observing the decryption output.

2. Since the algorithm takes a variable-length key, from 32 bits to 448 bits, making it ideal for a variety of use cases.

3. Blowfish is considered to be a block Cipher since changing one text alerts that section of the block encryption.
4. Because it encrypts and decrypts with the same key, it is called a symmetric cipher. Any key change causes the ciphered text to be decoded erroneously.