

# Experiment 5

## BlowFish

**Name:** Sumeet Haldipur

**UID:** 2019130018

**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 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]

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

Original encoded message

## Blowfish – Symmetric Ciphers Online

Input type: Text

Input text:  
(plain)

If there is a will there is a way but the way may not be the right way

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



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:  
(plain)

kanye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!  

Encrypted text:

00000000	17 4c b9 39 3f 1d 89 cf 29 b8 65 2b 64 34 f0 36	. L <sup>1</sup> 9 ? . . Ĭ ) , e + d 4 ð 6
00000010	18 1f f1 eb 84 24 e2 bc e9 03 c5 be da 1a 1f ed	. . ñ ë . \$ â % é . Å % ú . . í
00000020	fa 70 d7 bf d4 ce ed 22 40 e3 ae 86 92 3c 1e 85	ú p × ç Ò Î í " @ ã ° . . < . 8
00000030	2a 82 41 b1 d8 5e c3 8b 8f 57 1a fb 03 c8 ea a3	* . A ± Ø ^ Ä 8 8 W . û . È ê £
00000040	bf 53 49 ce c3 f6 e5 ab	ç S I Î Æ ö å «

[\[Download as a binary file\] \[?\]](#)

Inactive

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

**Input type:** Text

**Input text:**  
(plain) If there is a will there is a way but the way may not be the right was

☒ Plaintext ☐ Hex Autodetect: ON | OFF



**Function:** BLOWFISH

**Mode:** ECB (electronic codebook)

**Key:**  
(plain) kanye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	17 4c b9 39 3f 1d 89 cf 29 b8 65 2b 64 34 f0 36	. L ¹ 9 ? . . Ĩ ) , e + d 4 ð 6
00000010	18 1f f1 eb 84 24 e2 bc e9 03 c5 be da 1a 1f ed	. . ñ ë . \$ à % é . Å ¼ Ú . . í
00000020	fa 70 d7 bf d4 ce ed 22 40 e3 ae 86 92 3c 1e 85	ú p × ¿ Ô Î í " @ ã ° . . < . 0
00000030	2a 82 41 b1 d8 5e c3 8b 8f 57 1a fb 03 c8 ea a3	* . A ± Ø ^ Ă 0 0 W . û . È ê £
00000040	4e 23 6e a1 a8 94 fa 9b	N # n ; ~ . ú .

[\[Download as a binary file\] \[?\]](#) Inactive

Last 16 characters of the encrypted message change. The rest of the message is the same.

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

**Input type:** Text

**Input text:**  
(plain)  
ef there is a will there is a way but the way may not be the right way

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



**Function:** BLOWFISH

**Mode:** ECB (electronic codebook)

**Key:**  
(plain)  
kanye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	94 7b aa a2 0e 1a 13 ac 29 b8 65 2b 64 34 f0 36	. { ð ¢ . . . ~ ) . e + d 4 ð 6
00000010	18 1f f1 eb 84 24 e2 bc e9 03 c5 be da 1a 1f ed	. . ñ ë . \$ â % é . Å ¤ Ú . . í
00000020	fa 70 d7 bf d4 ce ed 22 40 e3 ae 86 92 3c 1e 85	ú p × ¸ Ô Î Í " @ ã ® . . < . 0
00000030	2a 82 41 b1 d8 5e c3 8b 8f 57 1a fb 03 c8 ea a3	* . A ± Ø ^ Ã 0 0 W . û . È è £
00000040	bf 53 49 ce c3 f6 e5 ab	¿ S I Î Æ ö å «

[\[Download as a binary file\] \[?\]](#) Inactive

First 16 characters of the encrypted message changes. 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:

## Blowfish – Symmetric Ciphers Online

Input type: Text

Input text:  
(plain)

If there is a will there is a way but the way may not be the right wa

☒ Plaintext ☐ Hex Autodetect: ON | OFF

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:  
(plain)

kanye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt! ▶ 🔗

Encrypted text:

00000000	17 4c b9 39 3f 1d 89 cf 29 b8 65 2b 64 34 f0 36	. L ¹ 9 ? . . Ĩ ) . e + d 4 ð 6
00000010	18 1f f1 eb 84 24 e2 bc e9 03 c5 be da 1a 1f ed	. . ñ ë . \$ â % é . Å ¼ Ú . . í
00000020	fa 70 d7 bf d4 ce ed 22 40 e3 ae 86 92 3c 1e 85	ú p × ¿ Ô Î í " @ ã ° . . < . ☐
00000030	2a 82 41 b1 d8 5e c3 8b 8f 57 1a fb 03 c8 ea a3	* . A ± Ø ^ Æ ☐ ☐ W . û . È è £
00000040	64 18 80 b9 c7 31 29 f3	d . . ¹ Ç 1 ) ó

[\[Download as a binary file\] \[?\]](#) Inactive

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 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)

If there is a will there is a way but the way may not be the right way

☒ Plaintext ☐ Hex Autodetect: ON | OFF



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:  
(plain)

manye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!  

Encrypted text:

00000000	0a 11 7f 1b fb d8 d2 3b c2 5f 53 15 16 06 f5 25	. . . û ø ò ; Å _ S . . . ö %
00000010	fb 6a d4 19 34 89 47 f7 44 f0 40 fa b6 e6 b8 cc	û j Ô . 4 . G ÷ D ð @ ú ŷ æ . Ì
00000020	7b 2b a9 c0 9f b0 99 f8 f6 96 2a 5f 12 f1 35 c4	{ + @ À . ° . ø ö ð * _ . ñ 5 Ä
00000030	a7 64 6d cc 3a 0a be 6b 18 c5 0a d6 73 a9 d8 13	§ d m Ì : . ¼ k . Å . Ö s @ Ø .
00000040	18 49 d0 83 da 6e 1e 31	. I Ø . Ú n . 1

[\[Download as a binary file\] \[?\]](#)

Inactive

After changing one character in a key, entire encrypted message changes. 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?

Input type: Text

Input text:  
(hex)

```
17 4c b9 39 3f 1d 89 cf 29 b8 65 2b 64 34 f0 36
18 1f f1 eb 84 24 e2 bc e9 03 c5 be da 1a 1f ed
fa 70 d7 bf d4 ce ed 22 40 e3 ae 86 92 3c 1e 85
2a 82 41 b1 d8 5e c3 8b 8f 57 1a fb 03 c8 ea a3
bf 53 49 ce c3 f6 e5 ah
```

☐ Plaintext ☒ Hex Autodetect: ON | OFF

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:  
(plain)

manye

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt! ▶ 🔗

Decrypted text:

00000000	8e 46 c3 f8 0f ef 1c 3d 03 3e 26 e7 52 c7 72 97	. F Ã ø . i . = . > & ç R Ç r .
00000010	a4 aa 1c 8b 12 4e 86 8e bd 9d 6c c8 40 b4 9d 28	π ð . @ . N . . % @ 1 È @ ' @ (
00000020	94 fd 47 99 db 60 b7 e1 37 3c 58 dd f2 6d 04 7b	. ý G . Û ` . á 7 < X Ý ò m . {
00000030	b9 dc d5 a2 55 e1 49 63 35 d2 6d 8b a7 bf b9 5c	¹ Ü Ö ¢ U á I c 5 Ò m @ § ¿ º \
00000040	60 aa 94 08 40 a5 68 c9	` ð . . @ ¥ h É

[\[Download as a binary file\] \[?\]](#) Inactive

Does not resemble original message and a lot of special characters are seen.

## 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.