

Divya Shah  
2019130057  
TE Comps

## Lab 5: Blowfish Encryption

### 1. Objective

This lab will give you the chance to experiment with an online encryption tool. You will encode a message and send it to someone else in the class, who will decode it when you supply the secret key. Note that this particular tool is of limited use in a security context, since the plaintext of the message is sent to and from the encryption web site! However, it could be used to prevent people from reading your email. A similar tool downloaded and running on your computer would provide a greater level of security. Some email clients even provide support for automatic encryption and decryption of all messages.

The [tool](#) we will use implements the [Blowfish](#) cipher system. Blowfish is a public domain algorithm designed and released by Bruce Schneier, a noted security expert. Although it was originally designed in 1993, it remains in use and no compromising errors are known in its design

### Laboratory Task: Testing Blowfish

Go to the [encryption tool](#) web site and try it out. Enter a short key phrase and a longer piece of text to be encoded. Then submit and see what your text looks like when encrypted.

```
41 12 dd 88 a9 13 83 f3 85 52 3a 04 3f d2 28 91  
44 ea df 9b f9 6c 43 cc 1d 8c 94 9b c7 1e 21 dc
```

Hexadecimal Numbers

```
A . Ý © . . ó R : . ? Ò (  
D ê ß . ù l C Ì . . . . Ç . ! Û
```

Encrypted Text

**Input type:** Text

**Input text:**  
(plain)  
Hello, This is a good time

☒ Plaintext ☐ Hex Autodetect: ON | OFF

**Function:** BLOWFISH

**Mode:** ECB (electronic codebook)

**Key:**  
(plain)  
qwertyuiop

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	41	12	dd	88	a9	13	83	f3	85	52	3a	04	3f	d2	28	91	A	.	Ý	®	®	.	.	ó	®	R	:	.	?	Ò	(	®
00000010	44	ea	df	9b	f9	6c	43	cc	1d	8c	94	9b	c7	1e	21	dc	D	ê	ß	.	ù	1	C	İ	.	.	.	Ç	.	!	Ü	

[Download as a binary file] [?]

Inactive

Try the following experiments and note how they change the output:

1. Change one character at the end of the message. How much of the encoded message changes?

Input type:

Text

Input text:  
(plain)

Hello, This is a good timq

1

☒ Plaintext
☐ Hex

Autodetect: ON | OFF

Function:

BLOWFISH

Mode:

ECB (electronic codebook)

Key:  
(plain)

qwertyuiop

☒ Plaintext
☐ Hex

> Encrypt!

> Decrypt!

Encrypted text:

00000000

41 12 dd 88 a9 13 83 f3 85 52 3a 04 3f d2 28 91

00000010

44 ea df 9b f9 6c 43 cc b2 6e 8a 4b 4d 3d 71 c4

[Download as a binary file] [?]

Inactive

Hexadecimal

41 12 dd 88 a9 13 83 f3 85 52 3a 04 3f d2 28 91

44 ea df 9b f9 6c 43 cc b2 6e 8a 4b 4d 3d 71 c4

Encrypted text

A . Ý © . . ó R : . ? Ò (

D ê ß . ù l C Ì º n . K M = q Ä

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

2. Change one character at the beginning of the message. How much of the encoded message changes?

Input type:

Text

Input text:  
(plain)

Mello, This is a good time

☒ Plaintext ☐ Hex

Autodetect: ON | OFF

Function:

BLOWFISH

Mode:

ECB (electronic codebook)

Key:  
(plain)

qwertyuiop

☒ Plaintext ☐ Hex

> Encrypt!

> Decrypt!

▶

🔗

Encrypted text:

00000000    cb f1 e8 f0 d3 31 e5 8c 85 52 3a 04 3f d2 28 91    Ě ñ è ð Ó 1 å . R : . ? ò (

00000010    44 ea df 9b f9 6c 43 cc 1d 8c 94 9b c7 1e 21 dc    D ê ß . ù l C Ĩ . . . . Ç . ! Ü

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

cb f1 e8 f0 d3 31 e5 8c 85 52 3a 04 3f d2 28 91	Ě ñ è ð Ó 1 å . R : . ? ò (
44 ea df 9b f9 6c 43 cc 1d 8c 94 9b c7 1e 21 dc	D ê ß . ù l C Ĩ . . . . Ç . ! Ü
Hexadecimal	Encrypted text

The first 16 characters of the encrypted message changes.

3. Delete one character at the end of the message. How much of the encoded message changes?

41 12 dd 88 a9 13 83 f3 85 52 3a 04 3f d2 28 91  
44 ea df 9b f9 6c 43 cc d7 02 51 9f 7c f0 72 5c

A . Ý © . . ó R : . ? Ò (   
D ê ß . ù l C Ì × . Q . | ð r \

**Input type:** Text

**Input text:**  
(plain) Hello, This is a good tim

☒ Plaintext ☐ Hex Autodetect: ON | OFF



**Function:** BLOWFISH

**Mode:** ECB (electronic codebook)

**Key:**  
(plain) qwertyuiop

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	41 12 dd 88 a9 13 83 f3 85 52 3a 04 3f d2 28 91	A . Ý © . . ó R : . ? Ò (
00000010	44 ea df 9b f9 6c 43 cc d7 02 51 9f 7c f0 72 5c	D ê ß . ù l C Ì × . Q .   ð r \

[Download as a binary file] [?]

Inactive

The last 16 characters of the encrypted message changes, the rest of the encrypted message remains same.

4. Change one character in the key. How much of the encoded message changes?

Input type: Text

Input text: (plain) Hello, This is a good time

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

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) qwertiuiop

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt! ▶ 🔗

Encrypted text:

00000000	7e 69 3a 2a 59 20 8a 8f a1 d7 4e e4 4a f6 08 f3	~ i : * Y . ı x N ä J ö . ó
00000010	92 4a d0 fd 12 21 d6 9d e7 17 ba e6 6f 2b bb 5d	. J Đ ý . ! Ö ç . ° æ o + » ]

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

```
7e 69 3a 2a 59 20 8a 8f a1 d7 4e e4 4a f6 08 f3
92 4a d0 fd 12 21 d6 9d e7 17 ba e6 6f 2b bb 5d
```

```
~ i : * Y . ı x N ä J ö . ó
. J Đ ý . ! Ö ç . ° æ o + » ]
```

The entire encrypted message changes significantly.

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

Input type:

Text

Input text:  
(hex)

4112dd88a91383f385523a

043fd22891

44eaddf9bfc43cc1d8c94

9bc71e21dc

☐ Plaintext
☒ Hex

Autodetect: ON | OFF

Function:

BLOWFISH

Mode:

ECB (electronic codebook)

Key:  
(plain)

qwertyuiop

☒ Plaintext
☐ Hex

> Encrypt!

> Decrypt!

Decrypted text:

00000000

48 65 6c 6c 6f 2c 20 54 68 69 73 20 69 73 20 61

00000010

20 67 6f 6f 64 20 74 69 6d 65 00 00 00 00 00 00

H

e

l

l

o

,

T

h

i

s

i

s

a

g

o

o

d

t

i

m

e

.

.

.

.

.

.

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

Decryption with the original key

Input type: Text

Input text: (hex)

41	12	dd	88	a9	13	83	f3	85	52	3a
04	3f	d2	28	91						
44	ea	df	9b	f9	6c	43	cc	1d	8c	94
9b	c7	1e	21	dc						

☐ Plaintext
 ☒ Hex
 Autodetect: ON | OFF



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) qwertiuiop

☒ Plaintext
 ☐ Hex

> Encrypt!
 > Decrypt!

Decrypted text:

00000000	0c db 41 ba ff 44 10 4b 80 e8 16 a3 ec 9d 87 4a	. Ő A º ŷ D . K . è . f i . J
00000010	bc ef 70 57 f5 58 31 be 42 2f 09 04 94 e3 a1 cf	% i p W ö X 1 % B / . . . ä i İ

[Download as a binary file] [?]

Inactive

Decryption with changed key.

No it doesnt look anything like the original.

## Conclusion:

It is understood that blowfish is a block cipher because changing of one text changes that part of block encryption. It can also be understood that it is a symmetric cipher because it encrypts and decrypts using the same key. Any change in key does not decipher the ciphered text properly.

Github Link: <https://github.com/Divya-127/CSS-Lab/tree/main/Exp5>