

Abhishek Chopra
2019130009
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.

Input text:
(plain)

The obsession for perfection

☒ Plaintext ☐ Hex Autodetect: ON | OFF

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:
(plain)

toggle

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	c9 bc 0c 98 10 d2 53 9b 82 cc 06 0d 52 ca 34 5a	É % . 0 . 0 S . . İ . . R Ê 4 Z
00000010	41 cb 1a b8 22 01 ae 33 dd a3 48 18 d0 e6 af 38	A Ê . , " . * 3 Ÿ £ H . Ð æ ` 8

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

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

Input type: Text

Input text:
(plain)

The obsession for perfection

☒ Plaintext ☐ Hex Autodetect: ON | OFF

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:
(plain)

toggle

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	c9 bc 0c 98 10 d2 53 9b 82 cc 06 0d 52 ca 34 5a	É % . 0 . 0 S . . İ . . R Ê 4 Z
00000010	41 cb 1a b8 22 01 ae 33 f0 3c 78 fe df e8 63 32	A Ê . , " . * 3 ð < x þ ß è c 2

[Download as a binary file] [?] Inactive

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)

Fhe obsession for perfection

1

☒ Plaintext ☐ Hex

Autodetect: **ON** | **OFF**

Function:

BLOWFISH

Mode:

ECB (electronic codebook)

Key:
(plain)

toggle

☒ Plaintext ☐ Hex

> Encrypt!

> Decrypt!

▶

🔗

Encrypted text:

00000000

9b d9 d0 c1 66 dd 5e 28 82 cc 06 0d 52 ca 34 5a

00000010

41 cb 1a b8 22 01 ae 33 dd a3 48 18 d0 e6 af 38

.

Ù

Đ

Á

f

Ý

^

(

.

İ

.

.

R

Ê

4

Z

A

Ě

.

,

"

.

®

3

Ý

£

H

.

Đ

æ

˘

8

[Download as a binary file! ?]

Inactive

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?

Input type: Text

Input text: (plain) The obsession for perfectio

☒ Plaintext ☐ Hex Autodetect: ON | OFF



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) toggle

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	c9 bc 0c 98 10 d2 53 9b 82 cc 06 0d 52 ca 34 5a	É ¼ . 0 . 0 S . . İ . . R Ê 4 Z
00000010	41 cb 1a b8 22 01 ae 33 4f ad 42 5a 4c b5 65 40	A Ě . , " . ® 3 0 . B Z L µ e @

[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)
The obsession for perfection

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



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key:
(plain)
toyg1e

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt!

Encrypted text:

00000000	84 ec 08 80 10 5a a9 79 98 9f e8 67 ad c0 6b 8c	. ì . . . Z @ y . è g . À k .
00000010	ad 6f d3 c1 7b 8a 7b 7c 1c 4e 8f b3 43 a3 33 17	. o Ó Á { . { . N ³ C £ 3 .

[Download as a binary file] [?]

Inactive

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)

c9	bc	0c	98	10	d2	53	9b	82	cc	06
0d	52	ca	34	5a						
41	<u>cb</u>	1a	b8	22	01	ae	33	<u>dd</u>	a3	48
18	d0	e6	<u>af</u>	38						

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

Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) toggle

☒ Plaintext ☐ Hex

> Encrypt! > Decrypt! ▶ 🔗

Decrypted text:

00000000	54	68	65	20	6f	62	73	65	73	73	69	6f	6e	20	66	6f	T	h	e	o	b	s	e	s	s	i	o	n	f	o	
00000010	72	20	70	65	72	66	65	63	74	69	6f	6e	00	00	00	00	r	p	e	r	f	e	c	t	i	o	n

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

Decryption with the original key

Input type: Text

Input text: (hex)

c9	bc	0c	98	10	d2	53	9b	82	cc	06
0d	52	ca	34	5a						
41	<u>cb</u>	1a	b8	22	01	ae	33	<u>dd</u>	a3	48
18	d0	e6	<u>af</u>	38						

☐ Plaintext
 ☒ Hex
 Autodetect: ON | OFF



Function: BLOWFISH

Mode: ECB (electronic codebook)

Key: (plain) toygle

☒ Plaintext
 ☐ Hex

> Encrypt!
 > Decrypt!

Decrypted text:

00000000	d6	ea	05	8d	ec	a2	93	dd	cb	8d	4c	81	62	e9	f1	d9	Ö	ê	.	▯	ì	¢	.	Ý	Ě	▯	L	.	b	é	ñ	Ù
00000010	2c	5f	f7	20	79	f8	4a	4e	3e	9d	7e	d6	bb	34	11	3b	,	_	÷	y	ø	J	N	>	▯	~	Ö	»	4	.	;	

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

Decryption with changed key.

No, it does not 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.