

University of Regina

Department of Computer Science

Winter 2019

CS 834 - Fundamentals of Computer System Security

Lab 2 - Symmetric Encryption

Submitted to Dr. Habib Louafi

By

**Chao Zhang
&
Betrand Nnamdi**

Regina, March 18, 2019

I. Setup

Creating a text file and name it: **message.txt**

```
vagrant@victim: ~
victin: virtual machine match the version of VirtualBox you have installed on
victin: your host and reload your VM.
victin:
victin: Guest Additions Version: 5.1.38
victin: VirtualBox Version: 6.0
==> victin: Setting hostname...
==> victin: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> victin: flag to force provisioning. Provisioners marked to run always will still run.

C:\CS834-Labs\Lab-2>vagrant ssh
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Mar 18 16:35:04 2019 from 10.0.2.2
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Mar 18 16:35:04 2019 from 10.0.2.2
vagrant@victim: $ sudo nano message.txt_
```

Adding the following content to the file and save it:

```
GNU nano 2.5.3 File: message.txt

CS 834 - Computer System Security
Lab 2
March 18, 2019
Group Members: Chao Zhang & Betrand Nnamdi

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text
^X Exit        ^R Read File   ^_ Replace     ^U Uncut Text
```

II. 3DES Encryption Algorithm

A. 3DES with operation mode CBC

a. Encrypt the message (the output is .enc):

The command used:

```

Select vagrant@victim: ~
vagrant@victim:~$
vagrant@victim:~$
vagrant@victim:~$ openssl enc -des-cbc -in message.txt -out message_DES_CBC.enc
enter des-cbc encryption password:
Verifying - enter des-cbc encryption password:
vagrant@victim:~$ ls
message_DES_CBC.enc message.txt
vagrant@victim:~$ nano message_DES_CBC.enc

```

screenshot of encrypted message:

```

vagrant@victim: ~
GNU nano 2.5.3 File: message_DES_CBC.enc
Salted__^L< ^@>^Q0 I 0 5 C &_!9 ` ^U+8/g x ^?=^Z?X< ^W@p s f
a u 0 ;^? '5 I ^@ o ^F ^YN
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

b. Decrypt the message (the output is .dec):

The command used:

```

vagrant@victim: ~
openssl: command not found
vagrant@victim:~$ openssl enc -d -des-cbc -in message_DES_CBC.enc -out message_DES_CBC.dec
enter des-cbc decryption password:
vagrant@victim:~$ ls
message_DES_CBC.dec message_DES_CBC.enc message.txt
vagrant@victim:~$ nano message_DES_CBC.dec

```

screenshot of decrypted message:

```

vagrant@victim: ~
GNU nano 2.5.3 File: message_DES_CBC.dec
CS 834 - Computer System Security
Lab 2
March 18, 2019
Group Members: Chao Zhang & Bertrand Nnamdi
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line

```

B. 3DES with operation mode ECB

a. Encrypt the message (the output is .enc):

The command used:

```
vagrant@victim: ~  
vagrant@victim: $  
vagrant@victim: $ nano message.txt  
vagrant@victim: $ vagrant@victim: $ openssl enc -des-ecb -in message.txt -out message_DES_ECB.enc  
enter des-ecb encryption password:  
Verifying - enter des-ecb encryption password:  
vagrant@victim: $ ls  
message_DES_CBC.dec message_DES_CBC.enc message_DES_ECB.enc message.txt  
vagrant@victim: $
```

screenshot of encrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_DES_ECB.enc  
Salted__ ^HHA [ : o ^@ j;F C _x^G ^?; ; .M 1w ` ^Kz [^Y ^N ^EaD^Y X!R ^Pp< 8a^A  
$  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^G Cur Pos  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

b. Decrypt the message (the output is .dec):

The command used:

```
vagrant@victim: ~  
vagrant@victim: $ ls  
message_DES_CBC.dec message_DES_CBC.enc message_DES_ECB.enc message.txt  
vagrant@victim: $ nano message_DES_ECB.enc  
vagrant@victim: $ vagrant@victim: $ ls  
message_DES_CBC.dec message_DES_CBC.enc message_DES_ECB.enc message.txt  
vagrant@victim: $ openssl enc -d -des-ecb -in message_DES_ECB.enc -out message_DES_ECB.dec  
enter des-ecb decryption password:
```

screenshot of decrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_DES_ECB.dec  
CS 834 - Computer System Security  
Lab 2  
March 18, 2019  
Group Members: Chao Zhang & Betrand Nnamdi  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^G Cur Pos  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

C. 3DES with operation mode CFB

a. Encrypt the message (the output is .enc):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ ls  
message_DES_CBC.dec message_DES_CBC.enc message_DES_ECB.dec message_DES_ECB.enc message.txt  
vagrant@victim:~$  
vagrant@victim:~$ openssl enc -des-cfb -in message.txt -out message_DES_CFB.enc  
enter des-cfb encryption password:  
Verifying - enter des-cfb encryption password:  
vagrant@victim:~$ ls  
message_DES_CBC.dec message_DES_CFB.enc message_DES_ECB.dec message_DES_ECB.enc  
message_DES_CBC.enc message_DES_ECB.dec message.txt  
vagrant@victim:~$
```

screenshot of encrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_DES_CFB.enc  
Salted__$Q+1 $ B ^W "GA $+iK  
^D<  
^ ^ b^H ^R R s^D ^joLh3^[^W ^B ^l ?^_ ^A^u#; J =(<^^>L %H Z^W ^ON  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^G Cur Pos  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^I To Spell ^_ Go To Line
```

b. Decrypt the message (the output is .dec):

The command used:

```
vagrant@victim: ~  
message_DES_CBC.dec message_DES_CFB.enc message_DES_ECB.dec  
message_DES_CBC.enc message_DES_ECB.dec message.txt  
vagrant@victim:~$ nano message_DES_CFB.enc  
vagrant@victim:~$  
vagrant@victim:~$ ls  
message_DES_CBC.dec message_DES_CFB.enc message_DES_ECB.dec  
vagrant@victim:~$ openssl enc -d -des-cfb -in message_DES_CFB.enc -out message_DES_CFB.dec  
enter des-cfb decryption password:  
vagrant@victim:~$ ls
```

screenshot of decrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_DES_CFB.dec  
CS 834 - Computer System Security  
Lab 2  
March 18, 2019  
Group Members: Chao Zhang & Bertrand Nnamdi  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^I To Spell
```

III. AES Encryption Algorithm

A. AES with operation mode CBC and key size 128

a. Encrypt the message (the output is .enc):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ openssl enc -aes-128-cbc -in message.txt -out message_AES_128.enc  
enter aes-128-cbc encryption password:  
Verifying - enter aes-128-cbc encryption password:  
vagrant@victim:~$ ls  
message_AES_128.enc  message_DES_CBC.enc  message_DES_CFB.enc  message_DES_ECB.enc  
message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  message.txt  
vagrant@victim:~$
```

screenshot of encrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_AES_128.enc  
Salted__P H LM^D G^0 ^Ug^A$mUy ^U4^0 .^N^I> ^R ^K u^U E ^W ^@I ^KtD ^N &C <$^C 't ! e0W mJlx r =3 =^D ^T  
g e ^?)J ^X^> 9  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^I To Spell ^G Go To Line ^V Prev Page ^N Next Page ^_ First Line  
^_ Last Line
```

b. Decrypt the message (the output is .dec):

The command used:

```
vagrant@victim: ~  
message_AES_128.enc  message_DES_CBC.enc  message_DES_CFB.enc  message_DES_ECB.enc  
message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  message.txt  
vagrant@victim:~$ openssl enc -d -aes-128-cbc -in message_AES_128.enc -out message_AES_128.dec  
enter aes-128-cbc decryption password:  
vagrant@victim:~$ ls  
message_AES_128.dec  message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  message.txt  
message_AES_128.enc  message_DES_CBC.enc  message_DES_CFB.enc  message_DES_ECB.enc  
vagrant@victim:~$
```

screenshot of decrypted message:

```
vagrant@victim: ~  
GNU nano 2.5.3 File: message_AES_128.dec  
CS 834 - Computer System Security  
Lab 2  
March 18, 2019  
Group Members: Chao Zhang & Betrand Nnamdi  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^I To Spell ^G Go To Line
```

B. AES with operation mode CBC and key size 192

a. Encrypt the message (the output is .enc):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ ls  
message_AES_128.dec message_DES_CBC.dec message_DES_CFB.dec message_DES_ECB.dec message.txt  
message_AES_128.enc message_DES_CBC.enc message_DES_CFB.enc message_DES_ECB.enc  
vagrant@victim:~$ openssl enc -aes-192-cbc -in message.txt -out message_AES_192.enc  
enter aes-192-cbc encryption password:  
Verifying - enter aes-192-cbc encryption password:  
vagrant@victim:~$ ls  
message_AES_128.dec message_DES_CBC.dec message_DES_CFB.dec message.txt  
message_AES_128.enc message_DES_CBC.enc message_DES_ECB.dec  
message_AES_192.enc message_DES_CFB.dec message_DES_ECB.enc  
vagrant@victim:~$ _
```

screenshot of encrypted message:

```
GNU nano 2.5.3 File: message_AES_192.enc  
Salted__ ^U 4P a ^B Z +> ^0 !j^Rq+ W '1 1 0 k ^Y Z ^\ lp^B z =s)W #?^_ ^S R^Q p z 9l =^P 0 ^?c ^\ $1~  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^8 Cur Pos ^U Prev Page ^H First Line  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell ^_ Go To Line ^N Next Page ^L Last Line
```

b. Decrypt the message (the output is .dec):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ ls  
message_AES_128.dec message_AES_192.enc message_DES_CBC.dec message_DES_CFB.dec message_DES_ECB.dec  
message_AES_128.enc message_DES_CBC.dec message_DES_CFB.dec message_DES_ECB.dec message.txt  
vagrant@victim:~$ openssl enc -d -aes-192-cbc -in message_AES_192.enc -out message_AES_192.dec  
enter aes-192-cbc decryption password:  
vagrant@victim:~$ ls  
message_AES_128.dec message_AES_192.dec message_DES_CBC.dec message_DES_CFB.dec message_DES_ECB.dec message.txt  
message_AES_128.enc message_AES_192.enc message_DES_CBC.dec message_DES_CFB.dec message_DES_ECB.dec  
vagrant@victim:~$ _
```

screenshot of decrypted message:

```
GNU nano 2.5.3 File: message_AES_192.dec  
CS 834 - Computer System Security  
Lab 2  
March 18, 2019  
Group Members: Chao Zhang & Betrand Nnamdi  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify  
^X Exit ^R Read File ^M Replace ^U Uncut Text ^T To Spell
```


C. AES with operation mode CBC and key size 256

a. Encrypt the message (the output is .enc):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ ls  
message_AES_128.dec  message_AES_192.enc  message_DES_CFB.dec  message_DES_ECB.enc  
message_AES_128.enc  message_DES_CBC.dec  message_DES_CFB.enc  message.txt  
message_AES_192.dec  message_DES_CBC.enc  message_DES_ECB.dec  
vagrant@victim:~$ openssl enc -aes-256-cbc -in message.txt -out message_AES_256.enc  
enter aes-256-cbc encryption password:  
Verifying - enter aes-256-cbc encryption password:  
vagrant@victim:~$ ls  
message_AES_128.dec  message_AES_192.enc  message_DES_CBC.dec  message_DES_ECB.dec  
message_AES_128.enc  message_AES_256.enc  message_DES_CFB.dec  message_DES_ECB.enc  
message_AES_192.dec  message_DES_CBC.dec  message_DES_CFB.enc  message.txt  
vagrant@victim:~$
```

screenshot of encrypted message:

```
GNU nano 2.5.3 File: message_AES_256.enc  
Salted__  
JC ^RM^C Z^F S ^X>< Z_ s ^A3 D ^X 02 .^G t ^_S^U^Ee ^^ _ K $^M ^Q/ M ! 1D m h ^G &Z " 9 >I ; 5  
  
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^G Cur Pos  ^V Prev Page  ^_ First Line  
^X Exit      ^R Read File  ^_ Replace  ^U Uncut Text  ^I To Spell  ^_ Go To Line  ^U Next Page  ^_ Last Line
```

b. Decrypt the message (the output is .dec):

The command used:

```
vagrant@victim: ~  
vagrant@victim:~$ ls  
message_AES_128.dec  message_AES_192.dec  message_AES_256.enc  message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  
message_AES_128.enc  message_AES_192.enc  message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  message.txt  
vagrant@victim:~$ openssl enc -d -aes-256-cbc -in message_AES_256.enc -out message_AES_256.dec  
enter aes-256-cbc decryption password:  
vagrant@victim:~$ ls  
message_AES_128.dec  message_AES_192.dec  message_AES_256.dec  message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  message.txt  
message_AES_128.enc  message_AES_192.enc  message_AES_256.enc  message_DES_CBC.dec  message_DES_CFB.dec  message_DES_ECB.dec  
vagrant@victim:~$
```

screenshot of decrypted message:

```
GNU nano 2.5.3 File: message_AES_256.dec  
CS 834 - Computer System Security  
Lab 2  
March 18, 2019  
Group Members: Chao Zhang & Betrand Nnandi  
  
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^G Cur Pos  
^X Exit      ^R Read File  ^_ Replace  ^U Uncut Text  ^I To Spell  ^_ Go To Line
```