

## Lab 8 Notes - Encryption

### Step 0: preparation

1. Install `openssl` by running this (or another command per your package manager)

```
sudo dnf install openssl
sudo apt install openssl
```

2. Create a plaintext file that we'll use for the rest of the project to encrypt

```
echo "some random super secret data txat i don't want mridul to read" >
plaintext
```

### Step 1: tr tool

After creating a plain text file we learned to use `tr` tool with a random key to encrypt and decrypt the files. `tr` command, basically, swaps the characters in the file with provided key, replacing all instances of `s` with `a`, for example.

```
tr 'a-z' 'qgvmftzyceolhsuwbjaxdnikpr' < plaintext > ciphertext-tr
tr 'qgvmftzyceolhsuwbjaxdnikpr' 'a-z' < ciphertext-tr > plaintext-tr
```

```
Terminal
~/Development/coen146/lab8 main !2 ?2 19:32:18
tr 'a-z' 'qgvmtzyceolhsuwbjxndikpr' < plaintext > ciphertext
19:32:25
tr 'qgvmtzyceolhsuwbjxndikpr' 'a-z' < ciphertext > plaintext-tr
19:32:32
cat plaintext && cat ciphertext && cat plaintext-tr

File: plaintext
1 some random super secret data txat i don't want mridul to read

File: ciphertext
1 ~ auhf jqsmuh adwfj afvjfx mqxq xkqx c mus'x iqsx hjcmdl xu jfqm

File: plaintext-tr
1 some random super secret data txat i don't want mridul to read

~/Development/coen146/lab8 main !2 ?2 19:32:35
```

## Step 2: openssl and aes

Using openssl we encrypted our text file with a 128-bit key using AES algorithm.

```
openssl enc -aes-128-ecb -e -in plaintext -out ciphertext-aes -k
00112233445566778899AABBCCDDEEFF
% openssl enc -aes-128-ecb -d -in ciphertext-aes -out plaintext-aes -k
00112233445566778899AABBCCDDEEFF
```

```
Terminal
~/Development/coen146/lab8 main !2 ?2 19:35:47
openssl enc -aes-128-ecb -e -in plaintext -out ciphertext-aes -k 00112233445566778899AABBCCDDEEFF
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

~/Development/coen146/lab8 main !3 ?2 19:35:58
openssl enc -aes-128-ecb -d -in ciphertext-aes -out plaintext-aes -k 00112233445566778899AABBCCDDEEFF
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

~/Development/coen146/lab8 main !3 ?3 19:36:16
cat plaintext && cat ciphertext-aes && cat plaintext-aes

File: plaintext
1 some random super secret data txat i don't want mridul to read

File: ciphertext-aes
1 ~ Salted___.%|J^ ^L^Wj
2 ~ ^]XEP7j"U(^X&k^W.'Ou\^tN^0

File: plaintext-aes
1 some random super secret data txat i don't want mridul to read

~/Development/coen146/lab8 main !3 ?3 19:36:25
```

### Step 3: openssl rsa tool and key encryption

This method allows us to use a public key and a password to decrypt the file. A private key is needed to encrypt the file but can also be used to decrypt it. We need to create a password that is used for decryption along with the public key when we create the private key.

```
openssl genrsa -aes128 -out privatekey 1024
openssl rsa -in privatekey -pubout > publickey
openssl rsautl -encrypt -inkey publickey -pubin -in plaintext -out
ciphertext-rsa
openssl rsautl -decrypt -inkey privatekey -in ciphertext-rsa -out
plaintext-rsa
```

```
Terminal
~/Development/coen146/lab8 main !6 ?4 19:41:34
openssl rsautl -encrypt -inkey publickey -pubin -in plaintext -out ciphertext-rsa
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.

~/Development/coen146/lab8 main !6 ?4 19:41:42
openssl rsautl -decrypt -inkey privatekey -in ciphertext -out plaintext-rsa
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
Enter pass phrase for privatekey:

~/Development/coen146/lab8 main !6 ?5 19:42:05
cat plaintext && cat ciphertext-aes && cat plaintext-aes

File: plaintext
1 some random super secret data txat i don't want mridul to read

File: ciphertext-aes
1 ~ Salted__@|J^ ^L^Wj
2 ~ ^]XEP7j"U(^X&k^W.'Ou\tN^0

File: plaintext-aes
1 some random super secret data txat i don't want mridul to read

~/Development/coen146/lab8 main !6 ?5 19:42:08
```

#### Step 4: digital signature using rsa

We can use the following commands to generate message hash, sign it, and verify the signature.

```
Terminal
~/Development/coen146/lab8 main !6 ?5 19:48:22
openssl sha256 -binary plaintext > plaintext.sha256

~/Development/coen146/lab8 main !5 ?5 19:48:23
xxd plaintext.sha256
00000000: fae8 1a4e b083 f838 3fb6 0084 df2d d9ea ...N...8?...-...
00000010: 21cc cb45 9840 e620 fef9 ea33 5140 852f !..E.@. ...3Q@./

~/Development/coen146/lab8 main !5 ?5 19:48:28
openssl rsautl -sign -inkey privatekey -in plaintext.sha256 -out plaintext.sig
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
Enter pass phrase for privatekey:

~/Development/coen146/lab8 main !5 ?5 19:48:36
openssl rsautl -verify -inkey publickey -in plaintext.sig -pubin -raw | xxd
The command rsautl was deprecated in version 3.0. Use 'pkeyutl' instead.
00000000: 0001 ffff ffff ffff ffff ffff ffff ffff .....
00000010: ffff ffff ffff ffff ffff ffff ffff ffff .....
00000020: ffff ffff ffff ffff ffff ffff ffff ffff .....
00000030: ffff ffff ffff ffff ffff ffff ffff ffff .....
00000040: ffff ffff ffff ffff ffff ffff ffff ffff .....
00000050: ffff ffff ffff ffff ffff ffff ffff ff00 .....
00000060: fae8 1a4e b083 f838 3fb6 0084 df2d d9ea ...N...8?...-...
00000070: 21cc cb45 9840 e620 fef9 ea33 5140 852f !..E.@. ...3Q@./

~/Development/coen146/lab8 main !5 ?5 19:48:40
ls
ciphertext      plaintext      plaintext.sha256 privatekey
ciphertext-aes  plaintext-aes  plaintext.sig   publickey
ciphertext-rsa  plaintext-rsa  plaintext-tr
```

## Step 5: calculating the hash

OpenSSL can be also used to calculate the hash

```
openssl dgst -sha256 plaintext
openssl dgst -md5 plaintext
```

```
Terminal
~/Development/coen146/lab8 main !5 ?5 19:52:41
openssl dgst -sha256 plaintext
SHA2-256(plaintext)= fae81a4eb083f8383fb60084df2dd9ea21cccb459840e620fef9ea335140852f

~/Development/coen146/lab8 main !5 ?5 19:52:50
openssl dgst -md5 plaintext
MD5(plaintext)= c4244a1850148eb926ac39d835583383

~/Development/coen146/lab8 main !5 ?5 19:52:57
ls
ciphertext      plaintext      plaintext.sha256 privatekey
ciphertext-aes  plaintext-aes  plaintext.sig   publickey
ciphertext-rsa  plaintext-rsa  plaintext-tr
```