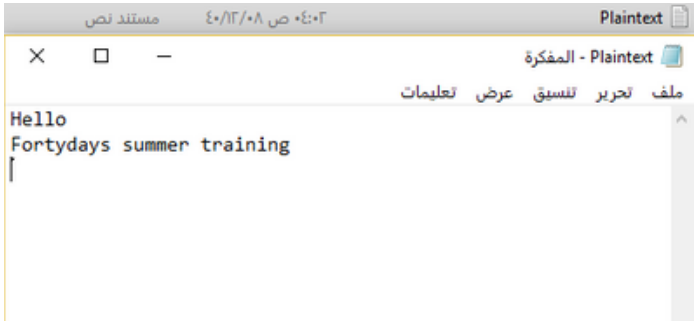
Encryption Algorithms :

Done By : OpenSSL



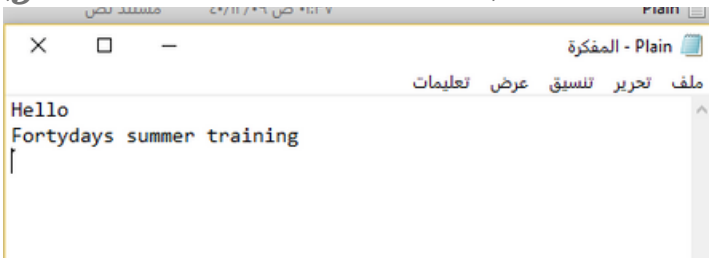
Creating Plaintext file

For Encrypt File :

enc -aes-256-ecb -a -salt -in file1.txt -out file2.txt

For Decrypt File :

enc -d -aes-256-ecb -a -salt -in file2.txt -out file1.txt



Note :

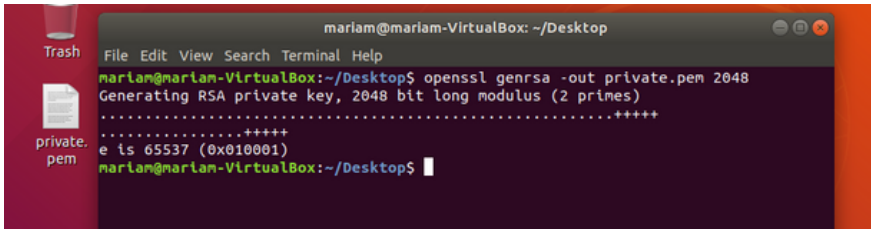
- file1.txt =Plaintext.txt
- file2.txt=Ciphertext.txt

Encryption Algorithms :

On ubuntu

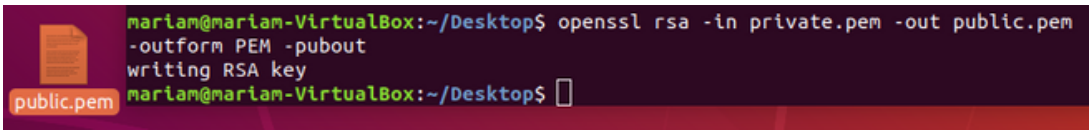
1-Generate Private Key:

openssl genrsa -out private.pem 2048



2-Generate Public Key :

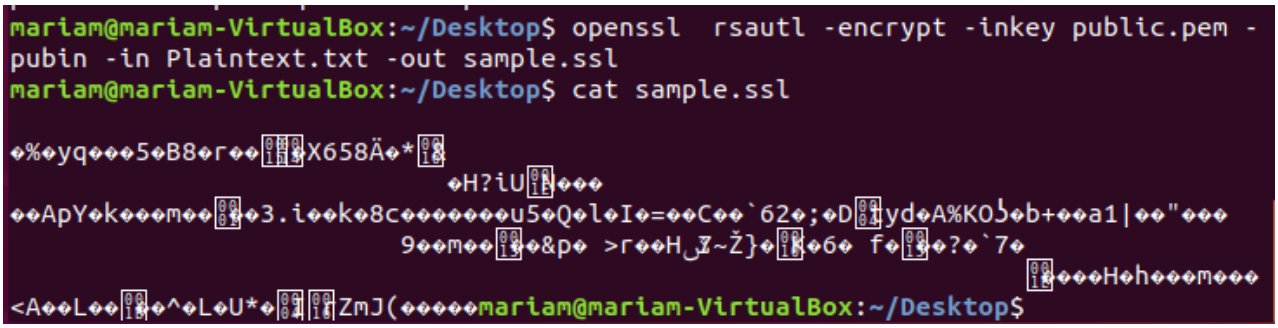
openssl rsa -in private.pem -out public.pem -outform PEM -pubout



Display Private.pem content :

Display Public.pem content :

3-Encrypt File called Plaintext.txt :



Encryption Algorithms :

On ubuntu

4-Decrypt File called Plaintext.txt into Hello.txt :

```
mariam@mariam-VirtualBox:~/Desktop$ openssl  rsautl -decrypt -inkey private.pem
-in sample.ssl -out hello.txt
mariam@mariam-VirtualBox:~/Desktop$ cat hello.txt
Hello fortydays summer training
mariam@mariam-VirtualBox:~/Desktop$
```

Hash Function :

Done By : node.bcrypt.js

Install via NPM

$ npm install bcryptjs

To hash a password: :

sync

Plaintext her = My own password "MariamSafar "

```
const bcrypt = require('bcrypt');
const saltRounds = 10;
const myPlaintextPassword = 's0/\/\P4$$w0rD';
const someOtherPlaintextPassword = 'not_bacon';
```

Encrypt String in javascript/NodeJS by using this Technique (auto-gen a salt and hash):

```
bcrypt.hash(myPlaintextPassword, saltRounds, function(err, hash) {
  // Store hash in your password DB.
});
```

```
const bcrypt = require('bcrypt')

let saltRounds =10

let myString ='MariamSafar'


bcrypt.hash(myString, saltRounds,(err, hash) => {


if(!err){
console.log(hash)
} else {
        console.log('Error',err)
}
})
```

```
ms@TOSHIBA1 MINGW64 ~/Desktop/Hacker task (master)
$ node pass.js
$2b$20$JxBYBY12ofqRELJOC9POqeZawGIQl7prtUYhQy4TeFFbeCUwKGua.
```

Hash password : Successfully generated

Hash Function :

To check a password :

```
// Load hash from your password DB.
bcrypt.compare(myPlaintextPassword, hash, function(err, res) {
    // res == true
});
bcrypt.compare(someOtherPlaintextPassword, hash, function(err, res) {
    // res == false
});
```

```
bcrypt.compare('MariamSafar', '$2b$20$JxBYBY12ofqRELJOC9P0qeZawGIQ17prtUYhQy4TeFFbeCUwKGua.' ,(err,res) => {
    if (!err){
        console.log('Password Correct:',res)
    }else{
        console.log('Error:',err)
    }
})
```

```
ms@TOSHIBA1 MINGW64 ~/Desktop/Hacker task (master)
$ node pass.js
Password Correct: true
```

```
bcrypt.compare('mariam12', '$2b$20$JxBYBY12ofqRELJOC9P0qeZawGIQ17prtUYhQy4TeFFbeCUwKGua.' ,(err,res) => {
    if (!err){
        console.log('Password Correct:',res)
    }else{
        console.log('Error:',err)
    }
})
```

```
ms@TOSHIBA1 MINGW64 ~/Desktop/Hacker task (master)
$ node pass.js
Password Correct: false
```