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:

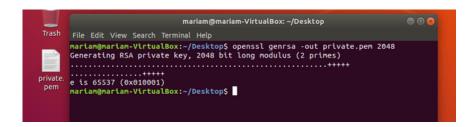
- file1.txt =Plaintext.txt
- file2.txt=Ciphertext.txt

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

```
mariam@mariam-VirtualBox:~/Desktop$ openssl rsa -in private.pem -out public.pem -outform PEM -pubout writing RSA key

public.pem mariam@mariam-VirtualBox:~/Desktop$ []
```

Display Private.pem content: Display Public.pem content:

IIIEogIBAAKKAQEAPHII/BWI9blEbRSsAIM4W60IQQUCLgnzyqHQ3hZPQcyh38QV
fLAUZYKUVXN01GrJ6hMHvJf2lLer/J19CEFCcvsd0Unc4ol4Lmbkk09Q0n3kf
sF3kTqd4/Sh19RXx+1310AZF47aVvTQeAMTC9bLf0b8fMNvgddW8rn900A9CC
xHLUX-Jh7NP/bSMB1CxeBXrH4KnVnV8d0V18hbtsA8JRfpGf*nv+Vf846Bqyzyu
J0naB9N2IQgo(y11dgAnV7)GN1yAqnSskx6GIHQ07x5k0n3kjzv13gBHM3BB
t+cundkuthflpLaNV11013jJlAGglBoJWNcWIDAQABA0IBACTSJ3bMLvHcRNm
q0PBAbJGHeAyKSjktsJeKvyLXVxVXNBxlKCt6u901V1evJXVcQXCQHHydtG18WM
/JZA100tQMC3Xh0ZdAvyJKoa4spZzebLyjhJnexLArxch2kjCuhnhqKxNkSZUpK
lu1RPZcMv3A9fbvmySnXv1pKn9JJUNO/tVJLOPchLnuqIQ8kLSMECU0vJh59kfx1
VMBBLC+1ZYJ3TQKbyHzTW74Jg)5cggzxYM8bLFtSAyg2pv/obfTvXPzmM62T
99St6+SXQECV7J2J76zm68XpNt00JJCCIJmQnAFJpoLHKXXxxbCUnnvtKpOhne
Mqu16ECGYEAZPhR9ZSw9hLKsB-Bww1coWf5/STfSk-wHEAUXCY0KBIC1024053
tauhnbWnQjJTAZwu8-B2QJUWP/GNSUfSkpnfLAByCQ03yThtGRE-YZU9ejpMHH
TckyHECE105EbF18TfGMCWCZTBBMCJBGCSSBW0-Eugh7_SwmlCAJGQCYEAWGC
FILAGZmW13UKCSEJ0FBgpruchV/SLNVUPTIV2Lg7Hl7eJ1V3PGAVSSPck

```
Rey

m-VirtualBox:~/Desktop$ cat public.pem

PUBLIC KEY----

phkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApHiI/BWV9blEbRS

zyqHQ3hZPqcyh38QVsfLAuZYKUVxm0iGrj8hNHvjf2Ller/J

l4LwnbkkD09QDn3kfi5FJkTqd4/ShIjBXx1+JIJeAZfaYuvT

/Hgd0WBrm9oD40PCpixHLlVs/h7NP/zbSMRICzw8XrN4KnVn

pGfYw+VfB46BqyzyutJDnaB9NZUgq/XjI1dgANY7VjGNjvAq

Dm3Njzvr3pEMW3aBo8i+cundkuthTipLaAVT1013jjlA3Ggl

BLIC KEY-----
```

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 beryptjs

To hash a password::

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$20$JxBYBY12ofqRELJOC9POqeZawGIQ17prtUYhQy4TeFFbeCUwKGua.
```

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
```