

南开大学

数据安全课程实验报告

在OpenSSL中进行数据签名及验证



学院：网络空间安全学院

专业：信息安全

学号：2113997

姓名：齐明杰

班级：信安2班

1 实验目的

参照教材2.3.6，实现在OpenSSL中进行数据签名及验证这一实验。

2 实验原理

公钥基础设施（PKI, Public Key Infrastructure），是一种遵循既定标准的密钥管理平台,它能够为所有网络应用提供加密和数字签名等密码服务及所必需的密钥和证书管理体系，简单来说，PKI就是利用公钥理论和技术建立的提供安全服务的基础设施。

数字证书是指在互联网通讯中标志通讯各方身份信息的一个数字认证，人们可以在网上用它来识别对方的身份。在PKI体系中，建有证书管理机构CA (Certificate Authority)。CA中心的公钥是公开的，因此由CA中心签发的内容均可以验证。

密钥的生存周期包括：密钥的产生和登记、密钥分发、密钥更新、密钥撤销、密钥销毁等。在产生密钥后，公钥需要在PKI中登记，并通过CA中心的私钥签名后形成公钥证书。由于CA中心的公钥公开，用户可以方便的对公钥证书进行验证，进而用户可以通过公钥证书来互相交换自己的公钥。进而，PKI作为安全基础设施，能够提供身份认证、数据完整性、数据保密性、数据公正性、不可抵赖性和时间戳六种安全服务。

PKI的应用非常广泛，其为网上金融、网上银行、网上证券、电子商务、电子政务等网络中的数据交换提供了完备的安全服务功能。

OpenSSL库提供了相关的基本功能支撑。

3 实验过程

由于该实验基于第2.2.4节的实例，即实验2.1，因此我先完成实验2.1的内容，再接着完成该实验。

3.1 OpenSSL安装

输入以下命令来安装openssl:

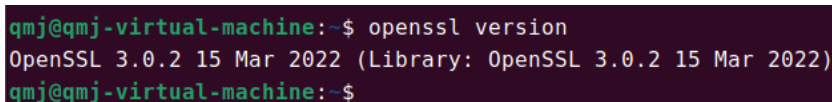
```
1 | sudo apt install openssl
```



```
qmj@qmj-virtual-machine: ~  
qmj@qmj-virtual-machine:~$ sudo apt install openssl  
[sudo] qmj 的密码:  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树... 完成  
正在读取状态信息... 完成  
openssl 已经是最新版 (3.0.2-0ubuntu1.12)。  
openssl 已设置为手动安装。  
升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 100 个软件包未被升级。  
qmj@qmj-virtual-machine:~$
```

显示已经安装，输入以下命令来查看版本:

```
1 | openssl version
```



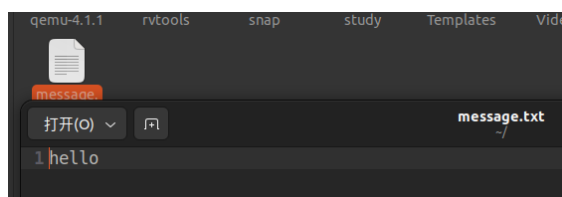
```
qmj@qmj-virtual-machine:~$ openssl version  
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)  
qmj@qmj-virtual-machine:~$
```

3.2 使用OpenSSL命令加解密文件

3.2.1 文件加密

使用 `aes-128-cbc` 对 `message.txt` 文件进行加密并使用 `base64` 编码，输出到 `ciphertext.txt`。假设128位密钥为 `a3171d177d1ce97ebc644ea3ff826b4e`，初始向量 `8bc65f2f883f95eea10b6f940cc805f6`。

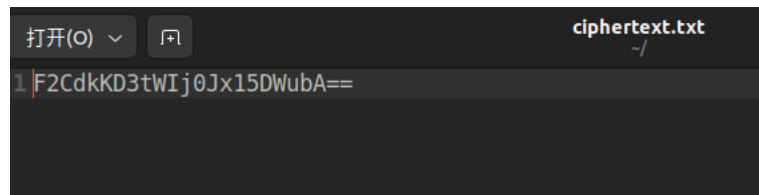
创建一个文件 `message.txt` :



输入命令:

```
1 | openssl enc -e -aes-128-cbc -in message.txt -out ciphertext.txt -K  
a3171d177d1ce97ebc644ea3ff826b4e -iv 8bc65f2f883f95eea10b6f940cc805f6 -  
base64
```

得到加密文件 `ciphertext.txt` :



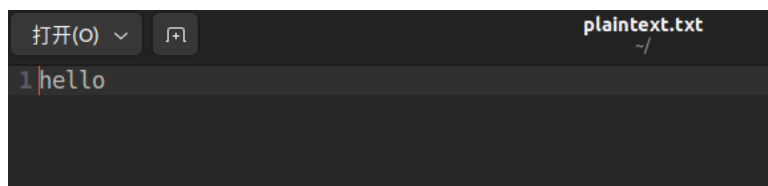
3.2.2 文件解密

对 `ciphertext.txt` 进行 `base64` 解码并解密, 结果输出到 `plaintext.txt`。

输入命令:

```
1 | openssl enc -d -aes-128-cbc -in ciphertext.txt -out plaintext.txt -K  
a3171d177d1ce97ebc644ea3ff826b4e -iv 8bc65f2f883f95eea10b6f940cc805f6 -  
base64
```

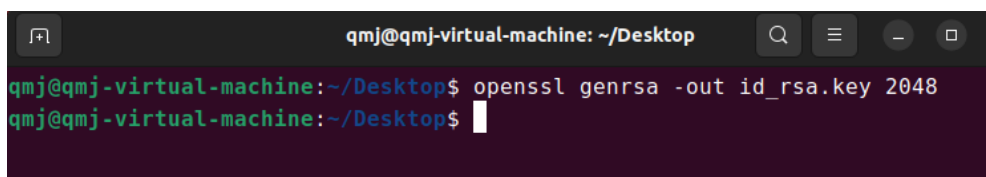
得到解密文件 `plaintext.txt` :



3.3 使用OpenSSL命令签名并验证

3.3.1 生成2048位密钥, 存储到文件id_rsa.key

```
1 | openssl genrsa -out id_rsa.key 2048
```



```
打开(O)  id_rsa.key 保存
1 -----BEGIN PRIVATE KEY-----
2 MIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCGRhrwXUNMPTBI
3 16fR0xeild8o+0TP2FjBqLVMVkJBrC+riadUU4Vot7euy4X+4JrJtHkAaPULeP
4 XwhqKAVRD5vhf9jWP0JDCRrk1CB0g+ZioVsWwpqVm38CgDJgBlrCKqdv4j9qVjDM
5 ivp5NnFONTEVQTtQGQJytqSka58/B3FL81esqWt2omXk6cxUEi9IKU6wYVWwp33N
6 nMelFu72G0UnKgxMjkmWP1YJhGNN9U11GLG00DJ+E6VRPaCNNm9KMdkMF8Kn2J7I
7 F/QW8wpKxcGpgp21ZPk96AJLZ2FUw+gJmK5CGWT4UhoU0cWA7fXQZYMdYIK4TUKX
8 iXpkdlHpAgMBAECggEAA0v9UkCbJ0sQmUAowP5xwaN1HLQKPqk42NhqS+KJSEqz
9 hJzjoA5gyW577x6JTgkh+/B228u+qTw2SEH3qRNAH6fAX7Ny3yRvmUK2nobVC/Ph
0 tixFioSMILmfGgSbxC2xlcXHK90U7WnyVmt5CuViAZxnnTWY9F5jKf0MYL4rz5ye
1 QRwdVg+z+fVXsYw6c+wHrDgTd8A008dXbuUGADYzopSWCzfsnaiPw0uJzoLPRe5u
2 xtw90R1Ri2MnIKrETu2EniWvpmIkuGyc+rK73yJrUZFXWf3rsBbV7pUh9RZWievQ
3 r04LfoHj21bsJmAAyAQA7baND0Zfwo0CE8Ewk2oBwKBgQC1djs2EBHu/lpv+FnZ
4 99n6D998ovrxcBZQYD2YcdvgVJ4GJA+WH0Jhz4K/klvE9outRsV1JDb1j6e0yttA
5 qpeN2qo+HfZ5p6jCmWccudPS06dJs19JQZAI+Z32MjPIZUPLenXAEuotij0BC2gj
6 TL5byeEWiX9mDjEnctMySybkwKBgQDMtGwZEjmQyuo7u9xofvg0XTPVlbVHaXPt
7 6GnruwTesVbquKMx0etA43XCf51N6owffuzCiZQUXfJz9YzzaWHNrAtwBlQ6IeJ
8 vJMPayzwImMF0LATIJQRIL8NdSoBA5knqHxlvJkCrE/Y7//BJtwY/3Yhx0saWMD
9 82Hfdo6UEwKBgQCbrPLfTqoT/uJXbRiophnajg0NScu/nImIKwis4AFiUzW+zvEQ
0 ONxiTUQ+VbGLKF5sKwR9NDRkfPnDgMTYLYGvkQi8gbgSZmPBN0AvNL0k2IGMGPaG
1 QljvWjbKIAeMxFOExDGATrlwmD/XVZHhuDn7RRZ130AMreA9l6TY0DhQQKBgDH3
2 dE9mWHDUzvpvr6w+Gd3GRknDQj2NKwQ0cBRfUR/+j6W5rfQGP6bGsE+mG9uG2978
3 UXzxZErEqAW12qZiCzgfDEA0nN9X8wWw5xqjYWW8gJaWcu/Q3+JLSanAhgkA/HZp
4 FkR+216aEYvUkF/yQuyCecrppc625rlnJiPCr6jLAoGAA2ggdgj105KooxZuGxw0
```

3.3.2 根据私钥文件，导出公钥文件id_rsa.pub

```
1 | openssl rsa -in id_rsa.key -out id_rsa.pub -pubout
```

```
qmj@qmj-virtual-machine: ~
qmj@qmj-virtual-machine:~$ openssl rsa -in id_rsa.key -out id_rsa.pub -pubout
writing RSA key
qmj@qmj-virtual-machine:~$
```

```
打开(O)  id_rsa.pub 保存
1 -----BEGIN PUBLIC KEY-----
2 MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAKRoA8F1DTD0wSNen0TsX
3 opXfKPtEz9hYwapbzFZFSQawvq4mnVFIuFaLe3rsuF/uCaybR5AGj1C3j18IaigF
4 UQ+b4X/Y1j9CQwka5NQgdIPmYqFbFsKaLZt/AoAyYAZawiqnb+I/alYwzIr6eTZx
5 TjUxUE7UBkCcrakpGufPwDxZfNXrKlrdqJl50nMVBIVsCL0sGFVsKd9zZzHpX7u
6 9htFJyoMZiZJl9WCYRjTfVNdRixjjgyfh0lUT2gjTZvSjHZDBfCp9ieyBf0FvMK
7 SsXBqYKdtWT5PegCZWdhVMPoCTJ0Qh1k+FiaFDnFg0310GWJg8iCuE1JF4l6ZHZR
8 6QIDAQAB
9 -----END PUBLIC KEY-----
```

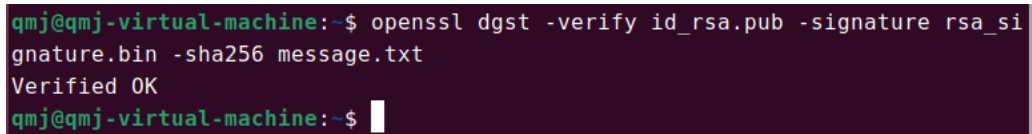
3.3.3 使用私钥对文件message.txt进行签名，输出签名到message.sha256

```
1 | openssl dgst -sign id_rsa.key -out rsa_signature.bin -sha256 message.txt
```



3.3.4 使用公钥验证签名

```
1 | openssl dgst -verify id_rsa.pub -signature rsa_signature.bin -sha256  
message.txt
```



```
qmj@qmj-virtual-machine:~$ openssl dgst -verify id_rsa.pub -signature rsa_si  
gnature.bin -sha256 message.txt  
Verified OK  
qmj@qmj-virtual-machine:~$
```

4 实验心得

经过本次实验，我对openssl的使用有了进一步的了解。我充分了解了使用openssl进行加密解密，签名和验签的过程。同时，对RSA非对称加密体系有了实践上的认知，知道了在实践中RSA的广泛使用。