

1 SSL 传输安全入门实验

主要内容：windows 下 Openssl 的部署与测试；命令行加密、签名操作；SSL 基础编程

1.1 windows 下 Openssl 的部署与测试

1.1.1 部署过程

1) 方法一

- Win64OpenSSL-1_1_1g.exe
- 64 位 windows 版的专用安装文件，不借助 activePerl 就能安装

2) 方法二

- 基于 openssl-1.1.1g.tar.gz 官网提供的版本安装
- 可供 Windows、Linux、Mac OS 等系统使用，不同系统的具体安装方法可在解压后查看 install 文件
- 实验提供了 windows 系统下安装官方包必要的工具 ActivePerl5 的安装包

1.1.2 测试

1) 进入 openssl 专属命令行（省去每次输入 openssl）

openssl

2) 版本和编译参数

openssl version -a

```
C:\Users\Yang Cheng>openssl version -a
OpenSSL 1.1.1g 21 Apr 2020
built on: Tue Apr 21 14:53:00 2020 UTC
platform: VC-WIN64A
options: bn(64,64) rc4(16x,int) des(long) idea(int) blowfish(ptr)
compiler: cl /Z7 /Fdssl_static.pdb /Gs0 /GF /Gy /MD /W3 /wd4090 /nologo /O2 -DL_ENDIAN -DOPENSSL_PIC -DOPENSSL_CPUID_OBJ
-DOPENSSL_IA32_SSE2 -DOPENSSL_BN_ASM_MONT -DOPENSSL_BN_ASM_MONT5 -DOPENSSL_BN_ASM_GF2m -DSHA1_ASM -DSHA256_ASM -DSHA512
2_ASM -DKECCAK1600_ASM -DRC4_ASM -DMD5_ASM -DAESNI_ASM -DVPAES_ASM -DGHASH_ASM -DECP_NISTZ256_ASM -DX25519_ASM -DPOLY1305
5_ASM -D_USING_V110_SDK71 -D_WINSOCK_DEPRECATED_NO_WARNINGS -D_WIN32_WINNT=0x0502
OPENSSLDIR: "C:\Program Files\Common Files\SSL"
ENGINESDIR: "C:\Program Files\OpenSSL\lib\engines-1_1"
Seeding source: os-specific
```

3) 查看支持的子命令

openssl help

openssl help [cmd]

openssl help rand

```
C:\Users\Yang Cheng>openssl help rand
Usage: rand [flags] num
Valid options are:
  -help           Display this summary
  -out outfile    Output file
  -rand val       Load the file(s) into the random number generator
  -writerand outfile Write random data to the specified file
  -base64         Base64 encode output
  -hex            Hex encode output
  -engine val     Use engine, possibly a hardware device
```

4) SSL 密码组合列表

openssl ciphers

```
C:\Users\Yang Cheng>openssl ciphers
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA:ECDHE-RSA-AES256-SHA:DHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA:RSA-PSS-AES256-GCM-SHA384:DHE-PSS-AES256-GCM-SHA384:RSA-PSS-CHACHA20-POLY1305:DHE-PSS-CHACHA20-POLY1305:ECDHE-PSS-CHACHA20-POLY1305:AES256-GCM-SHA384:PSK-AES256-GCM-SHA384:PSK-CHACHA20-POLY1305:RSA-PSK-AES128-GCM-SHA256:DHE-PSK-AES128-GCM-SHA256:AES128-GCM-SHA256:PSK-AES128-GCM-SHA256:AES256-SHA256:AES128-SHA256:ECDHE-PSK-AES256-CBC-SHA384:ECDHE-PSK-AES256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:RSA-PSK-AES256-CBC-SHA384:DHE-PSK-AES256-CBC-SHA384:RSA-PSK-AES256-CBC-SHA:DHE-PSK-AES256-CBC-SHA:AES256-SHA:PSK-AES256-CBC-SHA384:PSK-AES256-CBC-SHA:ECDHE-PSK-AES128-CBC-SHA256:ECDHE-PSK-AES128-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:RSA-PSK-AES128-CBC-SHA256:DHE-PSK-AES128-CBC-SHA256:RSA-PSK-AES128-CBC-SHA:DHE-PSK-AES128-CBC-SHA:AES128-SHA:PSK-AES128-CBC-SHA256:PSK-AES128-CBC-SHA
```

5) 测试所有算法速度

openssl speed

6) 测试 RSA 速度

openssl speed rsa

```
C:\Users\Yang Cheng>openssl speed rsa
Doing 512 bits private rsa's for 10s: 153198 512 bits private RSA's in 9.98s
Doing 512 bits public rsa's for 10s: 2052185 512 bits public RSA's in 10.02s
```

- 注意网上查资料解读命令行显示信息的含义

- 注意 rsa 签名数据量等于公私钥的长度

- 记录

- 密钥长度（512、1024、2048 三种）、密钥类型（公钥、私钥）、数据量、时长

- 密钥长度（512、1024、2048 三种）、签名或验签类型、数据量、时长

7) 测试 AES 速度

```
openssl speed aes
```

```
C:\Users\Yang Cheng>openssl speed aes
Doing aes 128 cbc for 3s on 16 size blocks: 29041072 aes 128 cbc's in 3.02s
Doing aes 128 cbc for 3s on 64 size blocks:
```

- 注意网上查资料解读命令行显示信息的含义
- 记录
 - 密钥长度、工作模式、数据量、时长

1.1.3 实验报告


- 1) 给出 RSA、AES 速度测试汇总表格
- 2) 分别对比 RSA、AES 加密速度和解密速度，给出结论
- 3) 与至少 1 位其他同学比较上述汇总表格，并判断谁的主机运算速度更快
- 4) （可选）根据 openssl speed 命令的能力，可以考虑设计一种工具评测主机性能，想想其测试的是主机哪方面能力，并给出该工具的设计方案

1.2 命令行加密、签名操作

本部分实验课上仅给出提示（删除详细指令，仅给出指令名称和主要参数），要求学生通过网络和帮助文件进行探索，实验课后给出参考步骤。

1.2.1 文件加密与解密

- 1) 生成随机数作为对称算法的密钥



```
sym256.key - Notepad
```

File Edit Format View Help

y- 闕 - 樑€, 啞鯉jW慧L 龠f5 箠 蘄r?

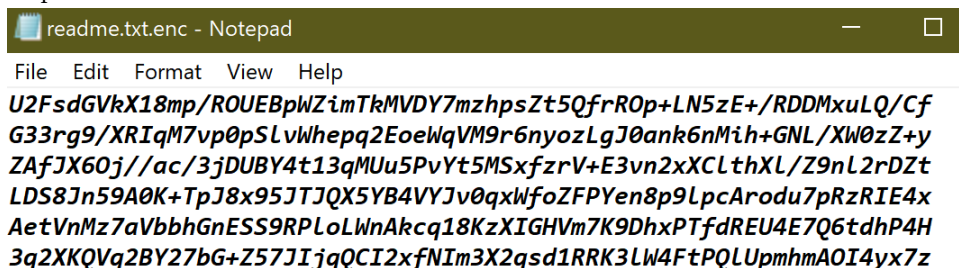
提示：rand 命令，注意生成的随机数长度要满足后续使用的密码算法要求（比如 AES256）

- 2) 加密文件示例

plaintext



ciphertext

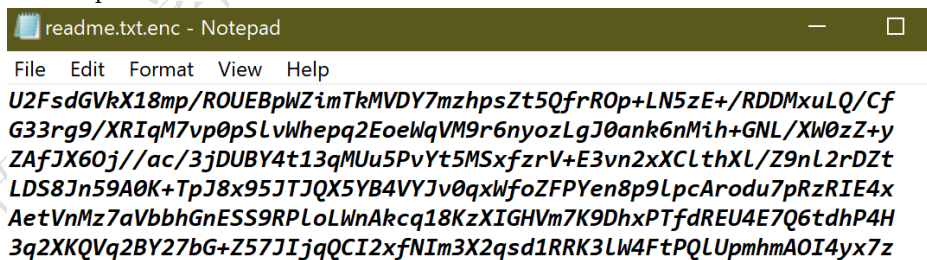


提示

- enc 命令
- 被加密的信息最好是文本，这样方便后续比对加解密是否成功
- 注意密钥参数的设定

3) 解密文件示例

- ciphertext



- plaintext

```

readme.txt - Notepad
File Edit Format View Help
openssl 开发手册.chm:
包含了较为详细的openssl 使用教程，对openssl 的各基础功

1.1.1版本官方文档:
https://www.openssl.org/docs/man1.1.1/
里面包含了各函数的说明等内容。
    
```

- 提示
 - enc 命令
 - 注意被解密的信息是前面被加密后的结果
 - 注意密钥参数的设定

4) 记录

- 打开原文件和加密后文件，判断是否加密成功
- 打开原文件和解密后文件，判断是否解密成功

1.2.2 计算文件摘要

1) 计算文件的 SHA256 值

- 提示
 - dgst 命令

2) 修改文件 1bit 或 1 字节，重新计算文件的 SHA256 值

- 提示
 - dgst 命令

3) 记录

- 两次 SHA256 计算的结果，以及改动后变化的比特位数，及其占总比特位数的比例

1.2.3 RSA 密钥生成与管理

1) 产生 RSA 密钥对

File Edit Format View Help

-----BEGIN RSA PRIVATE KEY-----

MIICXQIBAAKBgQC3L6vxaCb+FKjcLUrCvn4/EL+P2+DNvH
El xYGncmVI 9Y2nv

• 提示

- genrsa 命令
- 注意打算生成的私钥长度

2) 取出 RSA 公钥方法 1

文件(F) 编辑(E) 查看(V) 帮助(H)

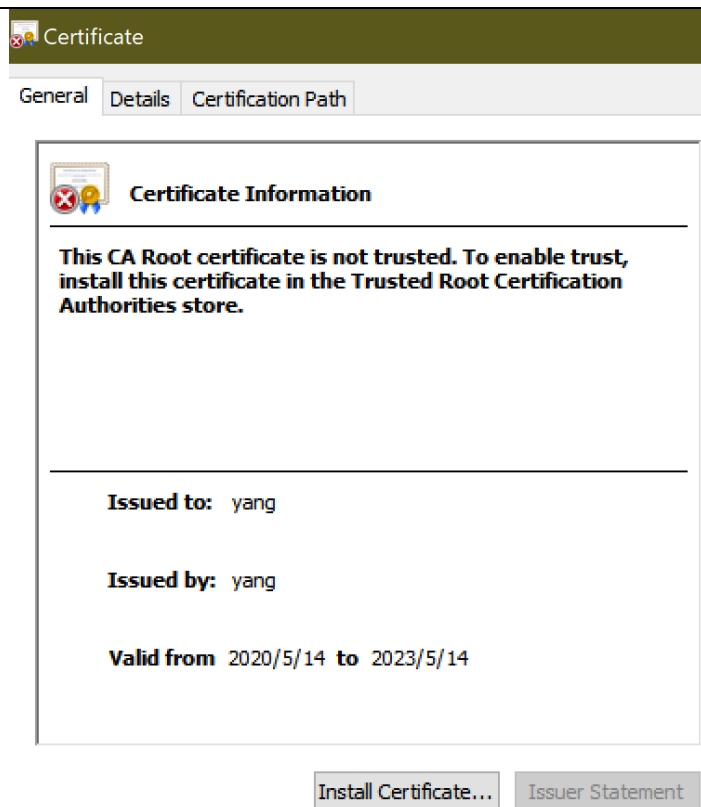
-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKE
2+DNvHn1O7t7aHn1vdtzi1FoFl xYGncmVI 9Y2nv

• 提示

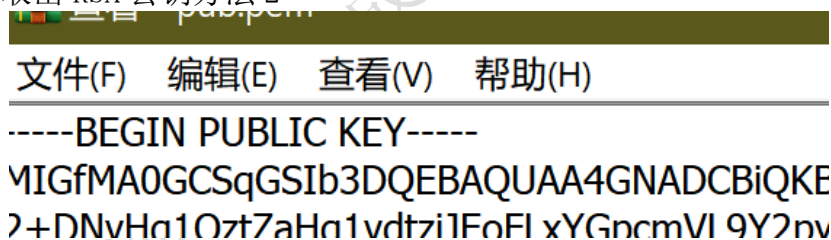
- rsa 命令
- 注意利用前面生成的私钥来获取公钥

3) 生成公钥证书



- 提示
 - req 命令
 - 注意 pem 和 crt 文件格式的区别

4) 取出 RSA 公钥方法 2



- 提示
 - x509 命令
 - 从公钥证书中获取公钥

5) 记录

- 用文本编辑器打开 priv.pem, pub.pem, 记录内容
- 双击生成的证书文件查看信息, 截图

Apink - Remember.mp4.sign - Notepad

File Edit Format View Help

欵h鸞^貶 簪?<跣占藥瑪驢藪L釧%?

- 提示
 - `dgst` 命令
 - 注意不加`-hex` 选项，否则无法验证
 - 注意选择私钥文件和文件格式

- 原文件验证

- 提示
 - Verified OK
 - Verification Failure
- 修改文件验证
- 对签名的文件进行任意改动后重新验证
 - Verified OK
 - Verification Failure
 - 提示
 - 修改视频文件可以采用 winhex、ultraedit 等类似工具进行修改

- 原文件验证、修改文件验证的结果

1.2.5 实验报告

1) 给出完整命令

- 生成随机数作为对称算法的密钥
- 加密文件、解密文件
- 计算文件的 SHA256 值
- 产生 RSA 密钥对、取出 RSA 公钥
- 签名、验证签名

2) 给出记录信息

- 给出生成公钥、私钥内容
- 给出证书文件内容信息截图
- 给出文件加密是否成功、解密是否成功的结论
- 给出文件摘要两次 SHA256 计算的结果，以及改动后变化的比特位数，及其占总比特位数的比例
- 给出文件签名是否成功，验证是否成功的结论，修改原文件后验证是否成功的结论

3) （选作）对实验过程中提示的警告信息（WARNING）进行分析，优化实验过程及其安全性

- 比如密钥生成方法-iter、-pbkdf2

4) （选作）基于上述实验，可以考虑设计一种对任意媒体（可以先以图像为例）文件边预览边签名和验证，尝试给出该工具的设计方案

1.3 SSL 基础编程（以 devc 为例）

1.3.1 devc 环境下配置 openssl

1) <https://blog.csdn.net/wingrez/article/details/96799346>

1.3.2 vs 环境下配置 openssl

1.3.3 代码中对 openssl 库的引入以及基本的文件加解密

1.3.4 基本的面向连接 Socket 通信，传输一个完整的文件

1.3.5 基于 SSL 的安全通信，双方身份鉴别并保密传输一个完整文件

1.3.6 实验报告

1) 完整程序源代码（不需要工程文件和各种临时文件）、程序流程图以及执行截图

1.4 其他语言下的 OPENSSL 参考

1.4.1 Python: PyOpenSSL

1) <https://pypi.org/project/pyOpenSSL/>

1.4.2 Golang: cgo 调用 OPENSSL 包

1) <https://www.jianshu.com/p/5518d82b3f4f>

1.4.3 PHP: openssl 扩展

1) <https://blog.csdn.net/zl834205311/article/details/99551657>

1.4.4 JAVA：直接调用 OPENSSL

1) <https://my.oschina.net/u/3695687/blog/1542125>