

**实 验 报 告**

**（ 2021/ 2022 学年 第 一 学期）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 课程名称 | 网络信息安全 | | | | | |
| 实验名称 | 使用OpenSSL进行加密操作 | | | | | |
| 实验时间 | 2021 | 年 | 11 | 月 | 15 | 日 |
| 指导单位 | 计算机学院信息安全系 | | | | | |
| 指导教师 | 吴礼发 | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 学生姓名 | 任远哲 | 班级学号 | B19031614 |
| 学院(系) | 计算机学院 | 专 业 | 信息安全 |

**实 验 报 告**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **实验名称** | | 使用OpenSSL进行加密操作 | | | | | **指导教师** | | 吴礼发 |
| **实验类型** | | 实验 | | **实验学时** | | 2 | **实验时间** | | 2021.11.15 |
| 1. **实验目的和要求**   1.在VMWare中开启Linux虚拟机，运行Openssl  2.对照ssl\_commands.doc文档，执行所有加密操作 | | | | | | | | | |
| 二、**实验环境(实验设备)**  1. Linux  2. Openssl | | | | | | | | | |
| **三、实验内容**  **1. 使用AES加密算法对文本加密，对不同参数加密后的密文进行截图、比较**  **（1）使用base64编码和不使用base64编码**  明文如下    **使用base64编码加密**  enc -aes-256-cbc -a -salt -in lincoln.txt -out cipher\_base64.enc  使用密码：123456  生成密文如下    解密  enc -d -aes-256-cbc -a -in cipher\_base64.enc  **不使用base64编码加密**  enc -aes-256-cbc -salt -in lincoln.txt -out cipher\_nobase64.enc  使用密码：123456  生成密文如下    解密  enc -d -aes-256-cbc -in cipher\_nobase64.enc  **（2）使用CBC模式和使用ECB模式**  上文使用的事CBC模式  下面使用ECB模式  enc -aes-256-ecb -a -salt -in lincoln.txt -out cipher\_base64\_ecb.enc  使用密码：123456  生成密文如下    enc -aes-256-cbc -salt -in lincoln.txt -out cipher\_nobase64\_ecb.enc  使用密码：123456  生成密文如下    **2．对于较长的文本使用AES-CBC base64编码加密，生成密文文件（使用 –out cihper.txt参数），打开密文cipher.txt，分别修改（不要删除或增加，只是修改）密文最开始的1个字符、中间任意1个字符、最后1个字符，再进行解密。分别看看三次解密会有什么问题，截图比较。**  **还是对lincoln.txt使用AES-CBC base64编码加密**  enc -aes-256-cbc -a -salt -in lincoln.txt –out cihper.txt    加密结果    **修改密文最开始的1个字符，得到cipher1.txt**  enc -d -aes-256-cbc -a -in cihper1.txt  对cihper1.txt尝试解密    **修改密文最中间的1个字符，得到cipher2.txt**  enc -d -aes-256-cbc -a -in cihper2.txt  对cihper2.txt尝试解密    **修改密文最开始的1个字符，得到cipher3.txt**  enc -d -aes-256-cbc -a -in cihper3.txt  对cihper3.txt尝试解密    **3．生成1024位的RSA私钥和公钥，粘贴到下方。为什么私钥会比公钥长？**  生成私钥： genrsa -out mykey.pem 1024    利用私钥生成公钥： rsa -in mykey.pem –pubout -out pubkey.pem    **为什么私钥会比公钥长？**  RSA体制下私钥和公钥是可以互换的。但人们之所以习惯把长的作为私钥，短的作为公钥，是为了方便公钥的分发过程，也可以缩短加密用的时间。  **4．以自己的学号建立文本文件，内容任意，如“B16040740.txt”,使用上一步（3）生成的私钥对“B16040740.txt”进行签名，将签名信息直接粘贴到下方。再使用对应的公钥进行验证操作，对验证结果截图保存到下方。（截图需要能看清学号信息，不得使用他人学号文件进行操作，否则按抄袭处理）**  文本文件内容    **对B19031614.txt进行签名**  dgst -sha1 -sign mykey.pem -out B19031614.sha1 B19031614.txt  **签名结果：**  ┯氎L庺?M?袴8?m陓鬥凮?t遗#\*Y­?K{)?齡\*m~鰘?新頽m阷施鉮1?瀱檗k矽0氳x礍?g=E?Э鏥H嬗 SD]a觻=0???腺轗?牺择p]?    **使用对应的公钥进行验证操作**  dgst -sha1 -verify pubkey.pem -signature B19031614.sha1 B19031614.txt  **验证结果**  **5．完成openssl\_commands.doc其他剩余所有操作**  连接到不同类型的启用SSL的服务器  s\_client -connect mail.163.com:443 | | | | | | | | | |
| **四、实验小结（包括问题和解决方法、心得体会、意见与建议等）**  问题和解决方法：  问题：为什么使用AES-CBC base64编码加密时，修改最开始的一个字符会导致解密出错  解决方法：通过询问laos1以及查阅资料，在WEB中，加密协议，相关算法、参数是通过握手协议进行交换得到的，但在加解密软件很多是放在密文文件的某个部分。如果破坏了一些关键参数比如初始化向量会导致一点也解密不出来。  心得体会：  这次实验，我通过windows上的绿色版openssl，学会了经典的对称加密算法和公钥密码算法，以及数字签名与验证。纸上得来终觉浅，绝知此事要躬行，通过实验才能更好理解书本上的知识。在前面密码学的学习中，因为缺少实验，我学的仅仅是数论，直到做了这个实验，我才从实际应用的层面上去了解密码学。同一种加密算法，在使用base64或不使用base64编码；使用cbc或使用ebc，结果都是很不相同。  不同于ascill编码，Base64是一种基于64个可打印字符来表示二进制数据的表示方法，Base64编码要求把3个8位字节（3\*8=24）转化为4个6位的字节（4\*6=24），之后在6位的前面补两个0，形成8位一个字节的形式。 如果剩下的字符不足3个字节，则用0填充，输出字符使用‘=’，因此编码后输出的文本末尾可能会出现1或2个‘=’。  CBC模式的全称是Cipher Block Chaining模式（密文分组链接模式）。在CBC模式中，首先将明文分组与前一个密文分组进行XOR运算，然后再进行加密。假设CBC模式加密的密文分组中有一个分组损坏了。在这种情况下，只要密文分组的长度没有发生变化，则解密时最多只有2个分组受到数据损坏的影响。假设CBC模式的密文分组中有一些比特缺失了，那么此时即便只缺失1比特，也会导致密文分组的长度发生变化，此后的分组发生错位，这样一来，缺失比特的位置之后的密文分组也就全部无法解密。  意见与建议：  无。 | | | | | | | | | |
| **五、指导教师评语** | | | | | | | | | |
| **成 绩** |  | | **批阅人** | |  | | **日 期** |  | |