山东大学 软件 学院

《 密码学原理与实践 》实验教学大纲

编写人：侯孟波　　审定人：林丰波

编制时间：2024.5.15　　审定时间：2024.6.10

一、课程基本信息

课程名称	密码学原理与实践				
英文名称	Cryptographic Principles and Practice				
课程编码	**sd03032530**				
开课单位	软件学院				
实验类型	□专业基础实验　□专业实验√　□综合实验 □创新实验　□开放实验				
课程性质	□必修√　□选修				
实验类别	□独立设课　□非独立设课√				
学分	0.5	总学时		实验学时	16
适用专业	软件工程(网络空间安全新工科)				
先修课程	程序设计语言 操作系统 计算机网络				
课程网站					

二、课程描述

（不超过 200 字，须提供中、英文对照描述）

在计算机和网络应用广泛普及的今天，密码学理论与技术日益受到重视。以密码学代表的信息安全理论和技术可以为信息安全提供较为全面的解决方案。本课程除了理论学习，配置了相关实验题目，内容涉及对称算法和公钥算法的具体实现和接口编程，数字证书系统、电子邮件安全和 VPN 系统配置部署，开源密码安全软件包的接口应用开发等。通过实验学习，使学生真实感受密码学的应用实践，培养动手能力和安全分析能力。

With the pervasive development of computer network technology and applications, more and more attentions are paying to the theory, technology, and application resolutions related to the information security issues. Cryptography can be used to construct the technology solutions for solving this issues. This course tries to provide a variety of experiments, including program and interface implementation of symmetric cipher, asymmetric cipher, digital certificate authority system, email security and VPN system deployment and configuration, programming with the open source cryptographic library. Through this experiments, the students can improve their security analysis and practice ability.

三、课程性质和教学目标

【教学目标】

本试验内容作为密码学理论与实践课程的配套上机实验，目的在于让学生在学习和掌握理论知识的同时，通过一系列和理论相配套的实验题目来验证理论的可实现性和实现方法与相关应用，通过实验，了解应用系统对安全的基本需求和基本解决方案，从算法编程到安全系统的安装，配置和部署，实现具体应用环境下的安全方案分析与应用。并通过密码函数库的编程实现一些基本的应用开发，为以后的安全实践打下良好的专业编程能力基础。

【教学要求】

巩固和加深对信息安全和密码学基本知识的理解，提高综合运用课程知识的能力。培养学生自学参考书籍，查阅手册、图表和文献资料的能力，培养对开源安全系统搜索、下载、编译、安装、配置、部署和运行维护的全方位能力，着力提高在开源系统接口基础上的二次开发能力。

四、课程教学内容及学时分配

实验一 题目：VMWARE 虚拟机和 Windows 操作系统安装（2 学时，选做）

实验内容：在个人微机基础上安装 VMWARE 虚拟机软件，并能够在其中安装用于实验环境的 WINDOWS 操作系统（或者 VISUAL BOX 虚拟机和 LINUX 操作系统），掌握虚拟机软件安装步骤和配置过程，了解虚拟机工作原理。

实验室（或个人笔记本系统）微机，内存 4G 以上，硬盘 30G 以上，Intel i5 以上 CPU 处理器。

实验要点：虚拟机内操作系统软件安装过程中注意虚拟机内存最小配置 2G 以上，网络桥接模式，Windows 操作系统要安装 CA 认证模块以备后用。

实验二 题目：对称加密算法实验 （8 学时，必做）

实验内容：下载 OPENSSL(或者 GMssl、libtomcrypt 函数库等开源 C（C++）语言或其他语言密码函数库)开源软件，编译链接形成相应操作系统下的函数库，通过阅读说明，了解其中 DES 和 AES 算法的函数接口，在原始加密函数（ECB\CBC\CFB\OFB）基础上进行封装，实现一组能直接加密解密任意长度消息的加解密函数接口，并在此基础上实现一个采用口令密钥演化方法针对任意文件口令加密加密和解密命令行程序。该程序可以对操作系统中的任意大小和类型的文件用口令加密得到密文文件，并能够通过口令解密得到明文文件。

进而学习其他函数，较熟练调用接口开发其他应用。

开源软件地址：

www.openssl.org OPENSSL 开源软件（说明见实验指导书）

www.gmssl.org 国密密码函数库软件（国家标准）

gitee.com/fotg/libtomcrypt LibTomcrypt 开原密码函数库

github.com/libtom/libtomcrypt LibTomcrypt 开原密码函数库

对称分组密码算法在程序实现时一般首先被抽象为这样一个基本函数：

Function（IN，OUT，K， ENC/DEC）

其中 IN 是固定长度的输入分组（DES 算法是 8 字节，AES 是 16 字节），OUT 是同样长度的输出分组，密钥 K 是随机比特串（DES 是 56 比特，AES 是 128 比特），Function 是相应的函数名称。

OpenSSL 中实现的对称算法非常多，重点关注三个：分组密码算法 DES、AES 和流密码算法 RC4。

以 OPENSSL 实现的函数为例，简单讲解（版本变化函数名可能会变化）：

1． DES 算法函数接口

DES 算法的基本函数是 ECB 操作模式对应的函数 DES_ecb_encrypt()。

该函数把一个 8 字节明文分组 input 加密成为一个 8 字节密文分组 output。参数中密钥结构 ks 是用函数 DES_set_key()准备，而密钥 key 是用随机数算法产生的 64 个随机比特。参数 enc 指示是加密还是解密。该函数每次只加密一个分组。

```
void DES_ecb_encrypt(const_DES_cblock *input,DES_cblock *output,
DES_key_schedule *ks,int enc);

int DES_set_key(const_DES_cblock *key,DES_key_schedule *schedule);
```

DES 算法 CBC 操作模式加解密函数是 DES_ncbc_encrypt()。参数 length 指示输入字节长度。如果长度不是 8 字节的倍数，则会被用 0 填充到 8 字节倍数。因此，输出可能比 length 长，而且必然是 8 字节的倍数。

```
void DES_ncbc_encrypt(const unsigned char *input,unsigned char
*output, long length, DES_key_schedule *schedule, DES_cblock *ivec,
int enc);
```

DES 算法 CFB 操作模式加解密函数是 DES_cfb_encrypt()。参数 length 指示输入字节长度。参数 numbits 则指示了 CFB 每次循环加密多少明文比特，也即密文反馈的比特数目。ivec 是初始向量，被看做第 0 个密文分组，是不用保密但应随机取值的 8 个字节。如果在一次会话中数次调用 DES_cfb_encrypt()，则应该记忆 ivec。由于 CFB 模式中每次 DES 基本操作只加密 numbits 比特明文，因此如果 numbits 太小则效率太低。

```
void DES_cfb_encrypt(const unsigned char *in, unsigned char *out,
int numbits, long length, DES_key_schedule *schedule, DES_cblock
*ivec, int enc);
```

另有一个 numbit 是 64 比特的版本，既高效又没有填充的麻烦，推荐使用。num 中的返回值指示了 ivec 中的状态，是和下次调用衔接的。

```
void DES_cfb64_encrypt(const unsigned char *in, unsigned char *out,
long length, DES_key_schedule *schedule, DES_cblock *ivec, int *num,
int enc);
```

OFB 和 CFB 类似，也有两个函数，用法一样。

void DES_ofb_encrypt(const unsigned char *in,unsigned char *out,int numbits,long length,DES_key_schedule *schedule,DES_cblock *ivec);

void DES_ofb64_encrypt(const unsigned char *in,unsigned char *out,long length,DES_key_schedule *schedule,DES_cblock *ivec,int *num);

2. AES 算法函数接口

典型参数的 AES 的基本操作是把 128 比特明文在 128 比特密钥指引下加密成 128 比特密文。OpenSSL 中关于 AES 的函数名和参数接口和 DES 的雷同。相关函数名如下：

int AES_set_encrypt_key();
int AES_set_decrypt_key();
void AES_ecb_encrypt();
void AES_cbc_encrypt();
void AES_cfb128_encrypt();
void AES_ofb128_encrypt();

3. RC4 算法函数接口

RC4 密码算法是流算法，也叫序列算法。流算法是从密钥作为种子产生密钥流，明文比特流和密钥流异或即加密。RC4 算法由于算法简洁，速度极快，密钥长度可变，而且也没有填充的麻烦，因此在很多场合值得大力推荐。

OpenSSL 中 RC4 算法有两个函数: RC4_set_key()设置密钥，RC4()加解密。可以把 RC4 看作异或，因此加密两次即解密。

void RC4_set_key(RC4_KEY *key, int len, const unsigned char *data);
void RC4(RC4_KEY *key, unsigned long len, const unsigned char *indata, unsigned char *outdata);

实验三 题目：公钥加密算法实验 （8 学时，必做）

实验内容：使用开源软件接口，采用 RSA 算法和对称算法相结合的方法实现针对任意大小和类型的文件内容的保密性、和数字签名及验证。

要求：能够生成用户的 RSA 密钥对，采用对称算法生成对称加密的密钥并用公钥加密，然后用对称密钥加密文件内容；采用公钥解密对称密钥，并用对称密钥解密文件。文件加密时要求生成签名，解密文件时要求能够验证签名。

本实验（1）掌握公钥算法的使用方法，（2）掌握混合密码体制的工作原理。

实验过程指导：

不同于对称加密算法中加密和解密使用同样的密钥，公钥算法分为加密密钥 K1 和解密密钥 K2 两部分，而且从 K1 很难计算推导出 K2。这样就可以保密 K2 而公布 K1，从而大大简化了密钥管理。习惯上 K1 称为公钥，K2 称为私钥。

加密使用公钥，解密使用私钥。ENC（P，K1）= C　DEC（C，K2）= P

RSA 加密算法的步骤是这样的：

（1） 找两个随机大素数 p 和 q；

（2） 计算模 n=pq 和 Euler 函数 Φ(n) =(p-1)(q-1)；

（3） 选取数 e 后用扩展 Euclid 算法求数 d 满足 ed≡1 mod Φ(n)；

（4） 保密私钥 K2=(d, n)，发布公钥 K1=(e, n)；

（5） 加密明文 p 时，计算密文 c = p^e mod n；

（6） 解密 c 时，计算 p = c^d mod n。

RSA 算法也可以用来签名：

（7） 对消息 m，其签名 s = m^d mod n；

（8） 验证（m，s）即判断 m =? s^e mod n。

下面介绍 OpenSSL 中 RSA 算法的函数接口。

1. RSA 密钥产生

RSA 密钥产生函数 RSA_generate_key()，需要指定模长比特数 bits 和公钥指数 e。另外两个参数为 NULL 即可。

```
RSA *  RSA_generate_key(int bits, unsigned long e, void (*callback)
(int,int,void *),void *cb_arg);
```

目前对于长达 663 比特的 RSA 模数已经有成功分解的先例，因此当前典型的应用场合使用 1024 比特模长的 RSA 算法，此时一个分组是 128 字节。

如果从文件中读取密钥，可使用函数 PEM_read_bio_PrivateKey()/PEM_read_bio_PUBKEY()，其中 EVP_PKEY 中包含一个 RSA 结构，可以引用。

EVP_PKEY *PEM_read_bio_PrivateKey(BIO *bp, EVP_PKEY **x, pem_password_cb *cb, void *u);

2. RSA 加密和解密

RSA 加密函数 RSA_public_encrypt()使用公钥部分，解密函数 RSA_private_decrypt()使用私钥。填充方式常用的有两种 RSA_PKCS1_PADDING 和 RSA_PKCS1_OAEP_PADDING。出错时返回-1。输入必须比 RSA 钥模长短至少 11 个字节（在 RSA_PKCS1_PADDING 时？）。输出长度等于 RSA 钥的模长。

int RSA_public_encrypt(int flen, const unsigned char *from,unsigned char *to, RSA *rsa,int padding);

int RSA_private_decrypt(int flen, const unsigned char *from,unsigned char *to, RSA *rsa,int padding);

3. RSA 签名和验证

签名使用私钥，验证使用公钥。RSA 签名操作是把被签署消息的散列值编码后用私钥加密，因此函数中参数 type 用来指示散列函数的类型，一般是 NID_md5 或 NID_sha1。正确情况下返回 0。

int RSA_sign(int type, const unsigned char *m, unsigned int m_length, unsigned char *sigret, unsigned int *siglen, RSA *rsa);

int RSA_verify(int type, const unsigned char *m, unsigned int m_length, unsigned char *sigbuf, unsigned int siglen, RSA *rsa);

4. 设计：签名并加密

在公钥体制中，每个人都有一对公钥私钥，可以使用这对密钥加密保护自己的文件。通常使用对称算法加密文件，而用公钥算法加密对称算法的密钥，此即混合密码体制。一般使用随机数产生函数（OpenSSL 库中有）产生随机比特用做对称密钥 Key，选择使用 DES/AES/RC4 等算法加密文件正文，而使用 RSA 算法加密对称密钥。公式化表示为：

RSA（Key，K1）|| RC4（Message，Key）

解密时先用私钥解密得到 Key，再用 Key 解密得到 Message。

如果想对消息签名以发现变化或防篡改，通常则使用私钥对原文的散列值进行签名，签名值和原文一起存放：

Message||RSA（Message，K2）

签名和加密可以结合起来，比如先用自己的私钥签名，再用对方的公钥加密，

这就是 PGP 加密邮件系统的基本思路。

关于私钥的保密问题。私钥一般使用口令保护（参见前一个实验），并且操作系统往往也限制对私钥密文的访问。

实验四　题目：微软 CA 组件安装和配置（2 学时，选做)

实验目的是通过安装和配置虚拟机的 WINDOWS2000　SERVER 操作系统中集成的 CA 组件,实现浏览器证书和 WEB 证书的签发,进一步了解微软公司的 PKI 技术框架,并能够使用该模块,在实际系统中进一步应用(也可以在安装 WINDOWS 2000 SERVER 的时候一并选择安装).实验内容包括 1．安装 CA 组件。2．配置 CA 组件。3．申请证书、签发证书、下载证书等。

证书授权服务器是 Windows Server 的一个附件，它放在 Windows Server 的安装盘上。默认安装 WINDOWS 操作系统时是不安装的(除非特别选择)。它可让你为建立和管理 X509 版本 3 的数字证书创建一个自定制的服务以作证书之用。你可以为 Internet 或者公司的内部网创建服务器证书，从而可让你的组织完全控制它自己的证书管理策略。

实验五 题目： EJBCA 系统的安装和配置 （8 学时，选做）

通过 EJBCA 系统的安装和配置，一方面了解大型软件的基本安装步骤，更重要的是了解和掌握作为一个典型的大型 CA 系统软件，其功能模块是如何构成的，实现的基本功能有哪些，掌握其优缺点，并能根据目前的应用环境，写出一个结构性分析报告，尽量提出进一步完善的方面，使之更适合实际应用的需要。本实验要求学生独立完成 EJBCA 系统的各个模块的安装和配置，并能完成多种数字证书的申请、签发、下载、注销，黑名单的生成、下载等 CA 基本功能。其中实验所需的具体软件环境和所需的软件相对较多，VMWARE 虚拟硬件环境下的 WINDOWS 操作系统，作为基本安装环境。EJBCA 完全采用 Java 编写，能够在任何采用 J2EE 服务器的平台上运行。开发和测试是在 Linux 和 Windows 上进行的。以 Windows XP 操作系统平台的安装、部署与应用为例说明（由于开源软件更新问题，请参考网站最新版本说明）：

0. ejbca_3_0_2.zip (http://ejbca.sourceforge.net)，EJBCA 系统源代码
1. j2sdk-1_4_1_01-windows-i586.exe (http://www.sun.com/)
2. jboss-3.2.5.zip (http://www.jboss.org/)
3. jce_policy-1_4_2.zip (http://www.sun.com/)
4. apache-ant-1.6.2-bin.zip

具体安装、配置、运行和维护参见网站 http://ejbca.sourceforge.net 下载文件中的说明部分。

实验六 题目：安全电子邮件的配置和使用（2 学时，选做）

实验目的是利用前面实验中的 CA 系统,颁发安全电子邮件证书,并在一个支持电子邮件证书的邮件客户端系统中安装配置,实现加密和签名电子邮件系统。

实验内容包括：（1）通过 OPENSSL 系统的命令行工具生成两个不同电子邮箱的邮件证书（2）配置 OUTLOOK 等支持加密签名功能的邮件客户端系统，实现签名电子邮件和加密签名电子邮件的发送与接收。

实验七 题目：安全 WEB 访问的配置和使用（2 学时，选做）

利用 OpenSSL 系统的工具,颁发浏览器 SSL 客户端证书以及 WEB SERVER 证书,并在一个支持 SSL 安全连接的 WEB 系统(如 IIS,APACHE)中安装配置,实现基于数字证书安全访问的 WEB 系统。实验内容包括：（1）申请并获取安装浏览器 SSL-CLIENT 证书。（2）申请并获取安装 WEB 服务器证书。（3）配置 WEB 服务器，实现 SSL 单向认证 WEB 登录和 SSL 双向认证 WEB 登录。

默认情况下我们所使用的 HTTP 协议是没有任何加密措施的，所有的消息全部都是以明文形式在网络上传送的，恶意的攻击者可以通过安装监听程序来获得我们和服务器之间的通讯内容。通过 SSL（Security Socket Layer）安全机制使用数字证书，建立 SSL 安全通道后，只有 SSL 允许的客户才能与 SSL 允许的 Web 站点进行通信，并且在使用 URL 资源定位器 时，输入 https://，而不是 http://。SSL（加密套接字协议层）位于 HTTP 层和 TCP 层之间，建立用户与服务器之间的加密通信，确保所传 递信息的安全性。SSL 是工作在公共密钥和私人密钥基础上的，任何用户都可以获得公共密钥来加密数据，但解密数据必须要通过相应的私人密钥。使用 SSL 安 全机制时，首先客户端与服务器建立连接，服务器把它的数字证书与公共密钥一并发送给客户端，客户端随机生成会话密钥，用从服务器得到的公共密钥对会话密钥 进行加密，并把会话密钥在网络上传递给服务器，而会话密钥只有在服务器端用私人密钥才能解密，这样，客户端和服务器端就建立了一个惟一的安全通道。

实验八 题目：OpenVPN 配置和使用（**4 学时，选做**）

VPN 的优点是对应用层透明，因此特别方便部署。终端用户自己部署 VPN 典型的有三种方式：IPSec 方式、基于 SSL 的 OpenVPN 方式、PPTP/L2TP 等拨号方式。其中 IPSec 方式是 IPv6 强制要求的。

OpenVPN 是一套开源软件的 VPN 实现，PPTP 在 Windows 中支持。本实验的目的是（1）掌握 OpenVPN 的部署和使用方法（2）了解其它 VPN 的部署和使用方法。实验内容包括（1）OpenVPN 软件的安装（2）基于共享秘密的 OpenVPN 配置（3）基于公钥证书的 OpenVPN 配置部署。

五、每年更新实验项目

每年在实验 1 至实验 8 更换 20%新的题目并增加部分选作题目。

六、考核及成绩评定方式

【考核内容】操作结果+报告

【成绩评定】必选题目实验代码和结果 占 60%，实验报告占 40%

七、教材及参考书目

【教材】 讲义《网络空间安全专业课程实验指导书》