信息安全专业-工程实践 3

密码算法工程实践选题要求

成都信息工程大学网络空间安全学院

2024 年 4 月

目　录

信息安全 2022 级《工程实践 3(密码算法工程实践)》选题注意事项

注 **1**：根据每道选题要求，完成工程实践 3；

注 **2**：选题时 1 人 1 题，并按时提交《教学实习报告》、《密码算法工程实践报告》、《答辩记录及评分表》和密码算法工程实践源代码；

注 **3**：每个班同一选题不能超过 4 位同学，每个班同一选题不能使用同一编程语言（C 和 C++算作同一种）；

注 **4**：按照规定要求到指定地点参加答辩考核（具体答辩时间和地点由指导老师通知）；

注 **5**：如果是开发功能完整的程序/软件/工具（包括友好的图形化操作界面、基于 C/S 结构），可以调用一些密码函数库实现的相关功能；可用组件的图形化界面设计，也可 Dos 界面，算法可调用已有编程接口完成，也可自行编写密码算法代码。

注 **6**：可选择 Java、VC++，C#，Python，Perl、PHP 等任意一种编程语言实现。也可以选择手机移动端 APP 开发语言，在手机移动端实现，通过移动端 APP 展示相应功能。

1. 基于 DES 算法的文件加解密工具 （5 小题）

采用 DES 算法对文件进行加解密。

功能要求:

（1） 要求采用 DES 算法，采用 ECB、CBC、CTR、OFB 或者 CFB 其中一种模式，对中英文文件进行加密。

（2） 采用 DES 算法对中英文文件进行解密。

算法实现要求:

（1） 可自行编写 DES 算法，也可以调用编译器或者加密包自带接口完成算法加密和解密。

（2） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求:

可以可视化界面，也可以是 Dos 界面

输入和输出要求:

(1) 文件可以输入任意格式文件，至少能加密 1K 以上的文件。(在工程实践中要有 1K 以上文件加密和解密测试用例)

(2) 显示加密和解密文件的时间

(3) 加密前，读取文件内容并在屏幕打印出来。(若文件内容多，则至少读取文件内容中，前 200 明文字符中文或者英文显示)。

(4) 加密后，把密文内容在屏幕上打印出来。(若密文太长，则密文可以以文件方式保存，前前 200 密文字符中文或者英文显示)

题目列表:

1.1 **基于 ECB 模式-DES 算法的文件加解密工具**

1.2 **基于 CBC 模式-DES 算法的文件加解密工具**

1.3 **基于 OFB 模式-DES 算法的文件加解密工具**

1.4 **基于 CFB 模式-DES 算法的文件加解密工具**

1.5 **基于 CTR 模式-DES 算法的文件加解密工具**

2. 基于 AES 算法的文件加解密工具 （5 小题）

采用 AES 算法对文件进行加解密。

功能要求:

（1） 要求采用 AES 算法，采用 ECB、CBC、CTR、OFB 或者 CFB 其中一种模式，对中英文文件进行加密。

（2） 采用 AES 算法对中英文文件进行解密。

算法实现要求:

（1） 可自行编写 AES 算法，也可以调用编译器或者加密包自带接口完成算法加密和解密。

（2） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求:

可以可视化界面，也可以是 Dos 界面

输入和输出要求:

(1) 文件可以输入任意格式文件，至少能加密 1K 以上的文件。(在工程实践中要有 1K 以上文件加密和解密测试用例)

(2) 显示加密和解密文件的时间

(3) 加密前，读取文件内容并在屏幕打印出来。(若文件内容多，则至少读取文件内容中，前 200 明文字符中文或者英文显示)。

(4) 加密后，把密文内容在屏幕上打印出来。(若密文太长，则密文可以以文件方式保存，前前 200 密文字符中文或者英文显示)

题目列表:

2.1 基于 ECB 模式-AES 算法的文件加解密工具

2.2 基于 CBC 模式-AES 算法的文件加解密工具

2.3 基于 OFB 模式-AES 算法的文件加解密工具

2.4 基于 CFB 模式-AES 算法的文件加解密工具

2.5 基于 CTR 模式-AES 算法的文件加解密工具

3. 基于 SM4 算法的文件加解密工具

采用 DES 算法对文件进行加解密。

功能要求:

（1） 要求采用 SM4 算法，采用 ECB、CBC、CTR、OFB 或者 CFB 其中一种模式，对中英文文件进行加密。

（2） 采用 SM4 算法对中英文文件进行解密。

算法实现要求:

（1） 可自行编写 SM4 算法，也可以调用编译器或者加密包自带接口完成算法加密和解密。

（2） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求：

可以可视化界面，也可以是 Dos 界面

输入和输出要求：

（1）文件可以输入任意格式文件，至少能加密 1K 以上的文件。（在工程实践中要有 1K 以上文件加密和解密测试用例）

（2）显示加密和解密文件的时间

（3）加密前，读取文件内容并在屏幕打印出来。（若文件内容多，则至少读取文件内容中，前 200 明文字符中文或者英文显示）。

（4）加密后，把密文内容在屏幕上打印出来。（若密文太长，则密文可以以文件方式保存，前前 200 密文字符中文或者英文显示）

题目列表：

3.1 基于 ECB 模式-SM4 算法的文件加解密工具

3.2 基于 CBC 模式-SM4 算法的文件加解密工具

3.3 基于 OFB 模式-SM4 算法的文件加解密工具

3.4 基于 CFB 模式-SM4 算法的文件加解密工具

3.5 基于 CTR 模式-SM4 算法的文件加解密工具

4. 基于 CS 模式的 Hash 算法文件完整性校验工具（共 5 题）

实现基于 CS 模式的 MD5、SHA-1、SHA-256、SHA-384、SHA-512 的文件完整性校验工具。

功能要求：

（1） 要求实现 CS 模式网络编程

（2） 客户端完成对文件 Hash 值的计算功能，并将 Hash 值和文件通过网络传送给服务器断。

（3） 服务器端接收端客户端的文件和 Hash 值，然后计算接收到文件的，用同一 Hash 函数计算得到 Hash 值，跟接收到的 Hash 比较，若一致，则验证通过。

算法实现要求：

（1） 可自行编写 Hash 算法，也可以调用编译器或者加密包自带接口完成算法 Hash 算法。

（2） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

（3） Hash 算法可选 MD5、SHA-1、SHA-256、SHA-384、SHA-512、SM3 任一 Hash 算法

界面要求:

可以可视化界面，也可以是 Dos 界面

输入和输出要求:

（1） 客户端把文件 Hash 值显示出来。

（2） 服务器端把接收到 Hash 值，以及文件内容显示出来。并把校验结果显示出来。

题目列表:

4.1 基于 CS 模式的 MD5 算法文件完整性校验工具

4.2 基于 CS 模式的 SHA-1 算法文件完整性校验工具

4.3 基于 CS 模式的 SM3 算法文件完整性校验工具

4.4 基于 CS 模式的 SHA-256 算法文件完整性校验工具

4.5 基于 CS 模式的 SHA-512 算法文件完整性校验工具

5. 基于 RSA 算法的文件加密和解密工具（共 1 题）

实现 **RSA 算法的文件加密和解密工具**

功能要求:

1）实现 RSA 公钥和私钥产生。要求私钥的长度至少 1024 比特。

2）用 RSA 公钥对中英文文件加密

3）用 RSA 私钥对中英文文件解密

算法实现要求:

（1） 可自行编写 RSA 算法，也可以调用编译器或者加密包自带接口完成算法 RSA 算法。

（2） RSA 算法私钥长度和模数 n 至少 1024 比特长度，即要求实现大数计算（建议可直接调用编译器大数库实现）

（3） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求:

可以可视化界面，也可以是 Dos 界面

输入和输出要求:

(1) 文件可以输入任意格式文件，至少能加密 1K 以上的文件。(在工程实践中要有 1K 以上文件加密和解密测试用例)

(2) 显示加密和解密文件的时间

() 加密前，读取文件内容并在屏幕打印出来。(若文件内容多，则至少读取文件内容中，前 200 明文字符中文或者英文显示)。

（4）加密后，把密文内容在屏幕上打印出来。(若密文太长，则密文可以以文件方式保存，前前 200 密文字符中文或者英文显示)

题目列表:

5.1 基于 RSA 算法的文件加密和解密工具

6. 基于 RSA 算法的文件签名和验证工具（共 1 题）

实现 RSA 算法的文件签名和验证工具

功能要求:

1）实现 RSA 公钥和私钥产生。要求私钥的长度至少 1024 比特。

2）用 RSA 私钥对中英文文件签名 （建议签名前可调用 Hash 算法计算摘要值，再计算签名）

3）用 RSA 公钥对中英文文件验证

算法实现要求:

（1） 可自行编写 RSA 算法，也可以调用编译器或者加密包自带接口完成算法 RSA 算法。

（2） RSA 算法私钥长度和模数 n 至少 1024 比特长度，即要求实现大数计算（建议可直接调用编译器大数库实现）

（3） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求:

可以可视化界面，也可以是 Dos 界面

输入和输出要求:

（1）文件可以输入任意格式文件，至少能加密 1K 以上的文件。(在工程实践中要有 1K 以上签名和验证测试用例)

（2）显示签名和验证文件的时间

（3）签名前，读取文件内容并在屏幕打印出来。(若文件内容多，则至少读取文件内容中，前 200 明文字符中文或者英文显示)，若签名前采用 Hash 函数，则把文件 Hash 值给输出出来。

（4）验证后，把验证结果在屏幕上打印出来。

题目列表:

6.1 基于 RSA 算法的文件签名和验证工具

7. 基于 CS 模式的 DH 密钥协商算法

功能要求:

（1） 要求实现 CS 模式

（2） 客户端实现产生一个奇数 p，判断是否是素数。素数要求至少 1024 比特，可调用编译器给定的大素性判断接口，或者自行编写米勒罗斌素性检测算法检测。求得模 p 的一个原根 g，原根要求至少 512 比特；并产生一个随机数 a，随机数至少 512 比特.然后计算密钥协商值 Sa，然后把 p，g，Sa 发送给服务器端。等待服务器端消息，若接收到服务器 Sb 值，然后并计算出协商密钥 K。

（2）服务器端收到 p，g，Sa，并产生一个随机数 a，随机数至少 512 比特.然后计算密钥协商值 Sb，然后把 Sb 发送给客户端。并计算出协商密钥 K。

算法实现要求：

（1）模幂运算、大数计算、原根判断、素性检测调用编译器或者加密包自带接口完成。（建议可直接调用编译器大数库实现）

（2） 编程语言不限，可选择 C、C++、Java、Python 等任意一种编程语言实现

界面要求：

可以可视化界面，也可以是 Dos 界面

输入和输出要求：

（1） 在界面把大素数 p，原根 g，计算的 Sa，Sb，以及协商密钥的值 k 显示出来。

题目列表：

7.1 基于 CS 模式的 DH 密钥协商算法