同济大學

| 文档状态 | 保密级别 | |
|---|---|---|
| [√]草稿 | 文档编号 | |
| [ ]修订 | 管理部门 | 同济大学信息安全原理 |
| [ ]发布 | 修订年月 | |
| | 版本号 | |

# OpenPGP 系统项目

# 需求规格说明书

| 修订人签字 | 审核人签字 | 批准人签字 |
| --- | --- | --- |
| 日期： | 日期： | 日期： |

# 变更履历

| 序号 | 变更日期 | 版本 | 变更位置 | 变更原因 | 修订人 | 审核人 | 批准人 |
|---|---|---|---|---|---|---|---|
| 1 | 建立初稿 | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

说明：“变更原因”主要是分为：

1. 建立初稿
2. 内容修订
3. 正式发布

# 目录

# 1. 文档介绍

## 1.1. 编写目的

本文档描述软件产品需求规格说明书（SRS）的目的是：

1）定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；

2）提供性能要求、初步设计和用户影响的信息，作为软件人员进行软件结构设计和编码的基础；

3）作为软件总体测试的依据。

## 1.2. 文档范围

OppePGP 系统需求规格说明书主要包含了该系统整体需求及功能性需求的详细介绍。

## 1.3. 读者对象

- 开发人员
- 用户

## 1.4. 术语与缩写解释

| 缩写、术语及符号 | 解释 |
|---|---|
| PGP | 由 Phil Zimmermann 开发的商业软件 |
| OpenPGP | 由 PGP 团队提议的一种安全协议 |
| 加密 | 将可读的明文文件进行编码，生成不可读密文文件的过程 |
| 解密 | 将不可读的密文文件进行解码，还原为可读的明文文件的过程 |
| 签名 | 签名后，所有用户可以验证发件人身份以及查看文件是否被篡改 |
| 数字证书 CA | 数字证书可以在通讯中标志各方身份信息 |

## 1.5. 参考资料

| 序号 | 文档名称 | 文档编号 | 版本 | 发布日期 |
|---|---|---|---|---|
| 1 | OpenPGP（PGP/GPG）深入浅出，完全入门指南 | | | |
| 2 | RFC4880 OpenPGP Message Format | | | |

# 2. 项目介绍

## 2.1. 项目说明

项目名称：OpenPGP 系统。

任务提出者：同济大学信息安全原理课程。

开发者：1953246 方若愚。

## 2.2. 项目背景

PGP （Pretty Good Privacy）由 Phil Zimmermann 在上世纪 90 年代开发，是一个能为数据通信提供加密和验证功能的程序。

OpenPGP 是 PGP Inc.向 IETF 提议制定的一种统一标准，定义了加密消息、签名、私钥和用于交换公钥的证书统一标准。

OpenPGP 协议定义的加密标准安全性较高，本次项目需要基于 OpenPGP 协议实现一个本地文件夹加密。

## 2.3. 项目目标

- 对目标文件实现对存储者和调阅者的基于 pgp 的真实性认证和文件加密；
- 使文件的安全性不依赖于本地系统，即
  a) 本地其他非授权用户（即便是系统管理员）无法以可理解的方式读出该文件夹中文件的内容
  b) 对处理过程中可能涉及的临时存储至少实现可靠的敏感信息残留覆盖
- 在 linux 或 MS Windows 上实现该协议的一个 C++实现案例，本次项目在 Windows 10 上实现。包括软件设计文档、源代码及注释，可执行安装包、自测用例和测试分析报告。第三方资源及其说明。

# 3. 需求说明

## 3.1. 整体需求

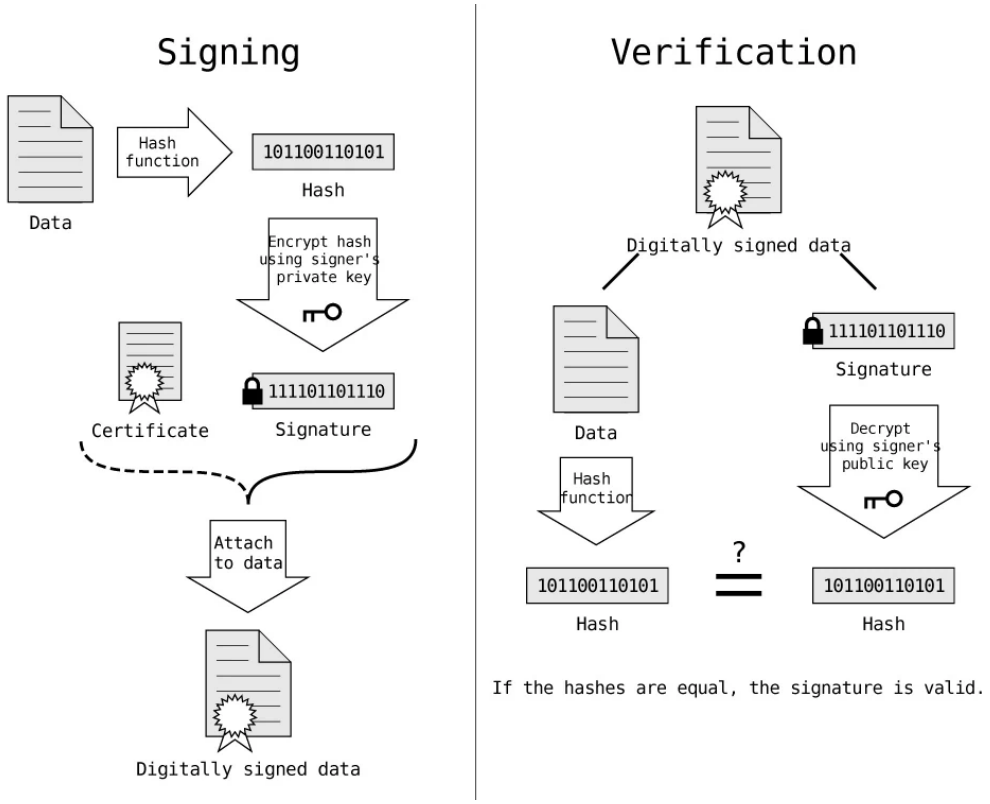
实现一个具有交互页面的文件加密软件，包括对文件进行加密和解密，用户签名，以及证书交换功能。系统中，每个用户都有一个公钥和私钥，其中公钥是公开的，私钥仅自己可以访问。

- 用户能够用自己的私钥对文件进行签名；

为了降低签名的时间，用户 A 利用哈希函数对文件 file 计算消息摘要 hash 后，用自己私钥 pri 对消息摘要进行签名，并将签名附加到消息中；

- 用户能利用发送者用户的公钥验证文件是否由该用户发出，以及文件是否被篡改；

  用户 B 收到签名后的文件后，利用用户 A 的公钥 pub 可以得到消息摘要 hash，然后再将接收到的消息计算消息摘要 hash-cal，如果 hash 和 hash-cal 相等则验证成功；

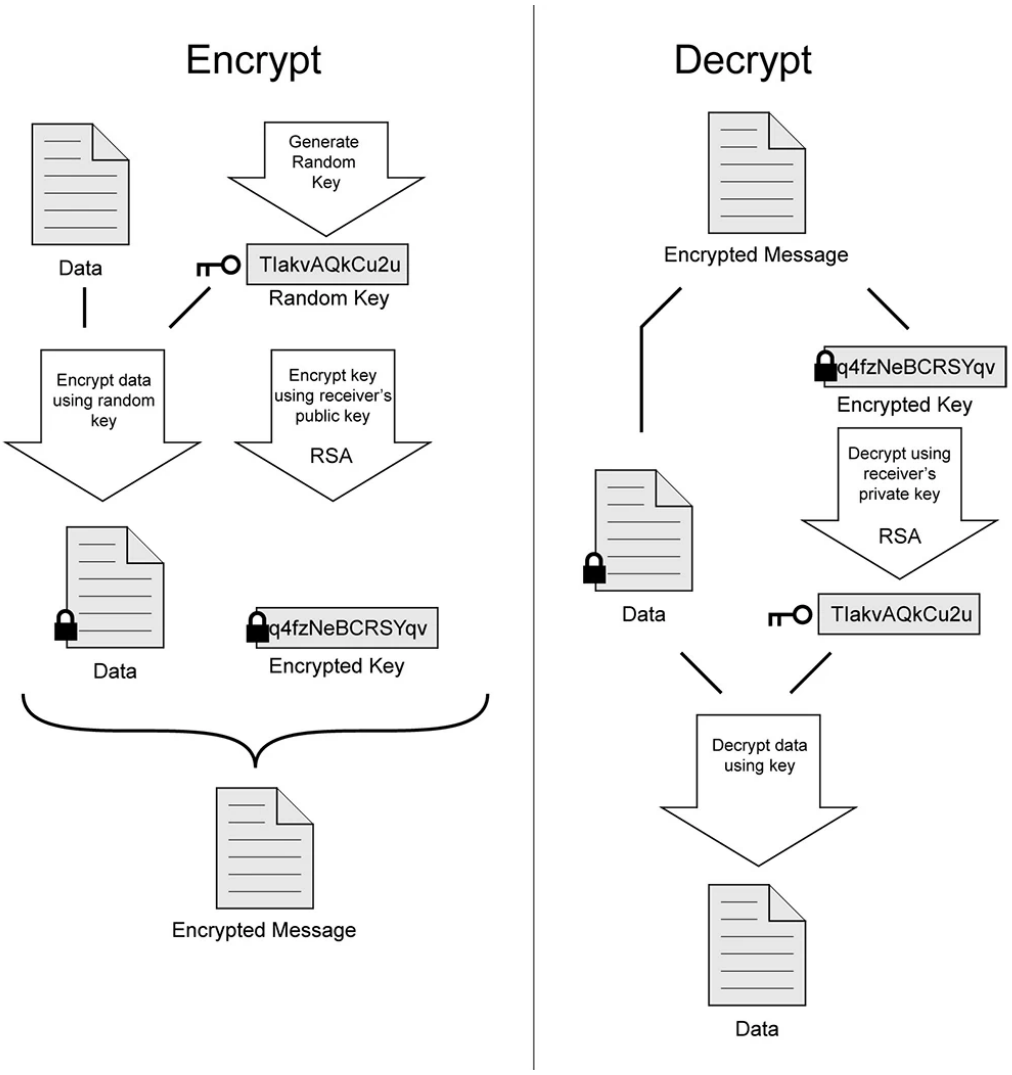



- 用户能对自己存储的文件进行加密，并设置权限；

  用户 A 对用户 B 开放权限的方式是：随机生成一个会话密钥 key，然后用 key 对文件 filename 进行加密，然后利用用户 B 的公钥 pub 对 key 进行加密，然后生成加密后的文件 filename.A，和加密后的 key 文件 filename.A.key；

- 用户能够阅读自己权限内的加密文件；

  用户 B 能够读取他权限内的文件 filename.A：B 先用自己的私钥 pri 对会话密钥文件 filename.A.key 进行解密得到 key，然后用 key 对 filename.A 进行解密可以得到原文件。




- 私钥保护功能；

  为了防止私钥被直接读取，

- 缓存保护；

## 3.2. 功能需求

### 3.2.1. 需求编号规则

需求编号： (模块名称)+（功能点）

### 3.2.2. 总体模块划分

主要分为认证机构模块、密码学模块、用户管理模块和菜单模块。

# 4. 功能性需求

## 4.1. 密码学算法模块 CRYPTO

| 模块名称 | 密码学算法 | | | |
|---|---|---|---|---|
| 模块简介 | 该模块手动实现非对称密码学算法 RSA 和对称密码学算法 TripleDES，以及利用 RSA 和 SHA1 哈希算法实现的证书和签名方案 | | | |
| 模块功能列表 | | | | |
| 序号 | 一级功能 | | 二级功能 | |
| | 功能名称 | 功能编号 | 功能名称 | 功能编号 |
| 1 | 非对称加密算法 | CRYPTO-01 | RSA 加密 | CRYPTO-01-1 |
| | | | RSA 解密 | CRYPTO-01-2 |
| | | | RSA 数字签名 | CRYPTO-01-3 |
| | | | RSA 签名验证 | CRYPTO-01-4 |
| 2 | 哈希算法 | CRYPTO-02 | SHA1 消息摘要 | CRYPTO-02-1 |
| 3 | 对称加密算法 | CRYPTO-03 | DES 加密 | CRYPTO-03-1 |
| | | | DES 解密 | CRYPTO-03-2 |

### 4.1.1. 非对称加密算法 CRYPTO-01

提供 RSA 相关的算法操作，包括加密解密，数字签名和认证，以及证书上传和吊销

#### 4.1.1.1. RSA 加密 CRYPTO-01-1

该部分利用公钥实现对数据的加密。

#### 4.1.1.2. RSA 解密 CRYPTO-01-2

该部分利用私钥实现对数据的解密。

#### 4.1.1.3. RSA 数字签名 CRYPTO-01-3

该部分利用私钥实现 RSA 数字签名。

#### 4.1.1.4. RSA 数字签名验证 CRYPTO-01-4

该部分利用公钥实现 RSA 数字签名验证。

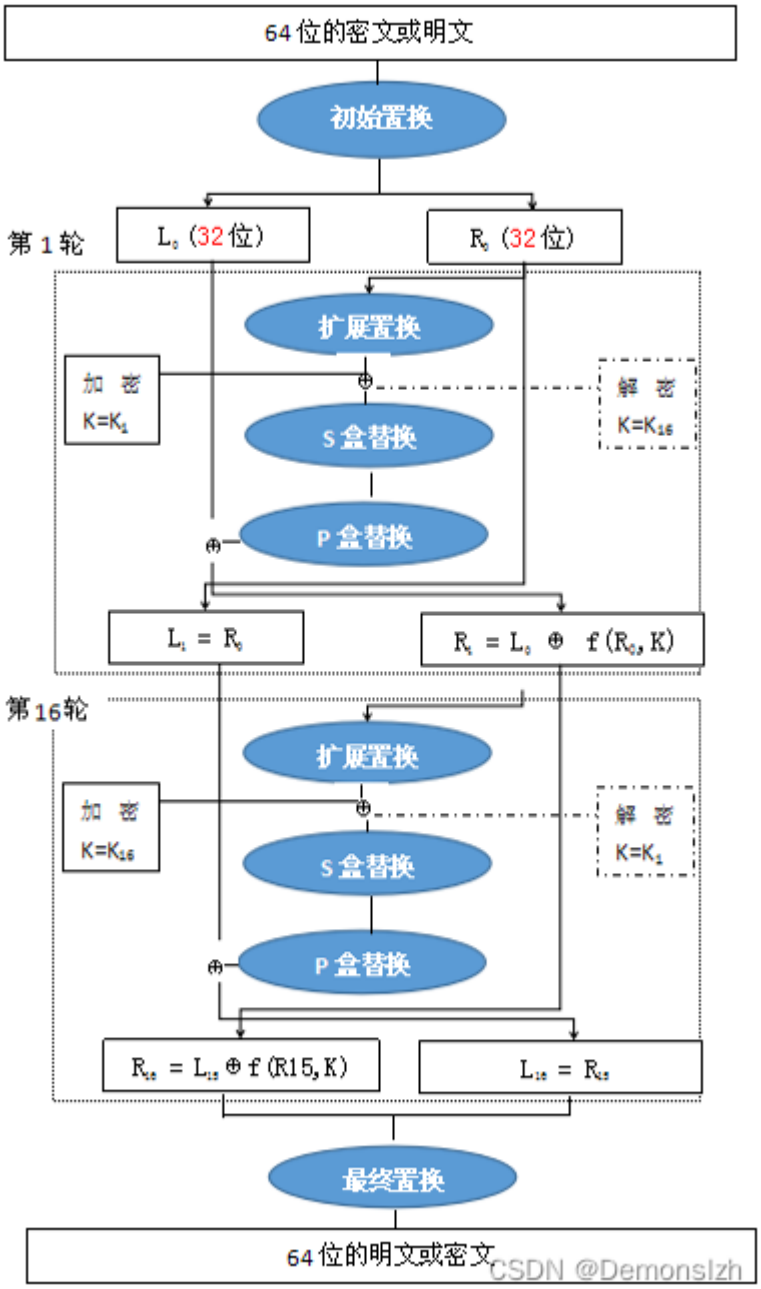
## 4.1.2. 哈希算法 CRYPTO-02

### 4.1.2.1. SHA1 消息摘要 CRYPTO-02-1

该部分实现计算消息摘要的 SHA1 算法，可以计算文件的摘要，也可以计算字符串的摘要。根据 SHA1 算法规定，该算法返回一个 160 比特的二进制串。

## 4.1.3. 对称加密算法 CRYPTO-03

本模块主要实现了 DES 加密解密算法。算法框架如下：



#### 4.1.3.1. DES 加密 CRYPTO-03-1

实现 ECB 模式下的 DES 加密算法，采用零填充的方式。输入文件路径和密钥进行加密

#### 4.1.3.2. DES 解密 CRYPTO-03-2

实现 ECB 模式下的 DES 解密算法。根据输入的路径和密钥进行解密

## 4.2. 认证机构模块 CA

| 模块名称 | 认证机构模块 | | | |
|---|---|---|---|---|
| 模块简介 | 实现认证机构，为系统用户提供申请证书和验证证书的功能，维护一个密钥服务器 | | | |
| 模块功能列表 | | | | |
| 序号 | 一级功能 | | 二级功能 | |
| | 功能名称 | 功能编号 | 功能名称 | 功能编号 |
| 1 | 证书管理 | CA-01 | 颁发证书 | CA-01-1 |

CA 模块需要先为自己设置一对公钥和私钥，为了多次启动程序不会更改私钥和公钥，每次程序初始化公钥私钥时需要读取公钥私钥的文件。

此外，该模块维护一个存储证书的数据库，为了简化处理，若 CA 的根目录为 root，则数据库用文件夹 root/CERT/ 来表示，该目录下的每一个文件形如 id.cert，其中 id 是用户的 id

### 4.2.1. 证书管理 CA-01

#### 4.2.1.1. 颁发证书

对用户进行验证后，为该用户分配公钥私钥对，并生成公钥的签名，创建证书文件。

## 4.3. 用户管理模块 USER

| 模块名称 | 用户模块 | | | |
|---|---|---|---|---|
| 模块简介 | 实现每个用户可以进行的操作 | | | |
| 模块功能列表 | | | | |
| 序号 | 一级功能 | | 二级功能 | |
| | 功能名称 | 功能编号 | 功能名称 | 功能编号 |
| 1 | 用户管理 | USER-01 | 用户注册 | USER-01-1 |
| | | | 用户登录 | USER-01-2 |
| 2 | 用户操作 | USER-02 | 证书验证 | USER-02-1 |
| | | | 文件签名 | USER-02-2 |
| | | | 文件签名验证 | USER-02-3 |
| | | | 文件加密 | USER-02-4 |

| | | | | |
|---|---|---|---|---|
| | | | 文件解密 | USER-02-5 |

该模块的作用是管理所有用户，通过维护用户的登录密码来判断当前用户的真实身份。

## 4.3.1. 用户管理 USER-01

### 4.3.1.1. 用户注册

在数据库中创建用户的登录文件，并为用户向 CA 申请公钥私钥对和证书。其中用户的密码以 SHA1 码的形式存储。

### 4.3.1.2. 用户登录

输入用户名和密码后，对密码进行验证。

## 4.3.2. 用户操作 USER-02

### 4.3.2.1. 证书验证 USER-02-1

调用密码学模块，对证书文件进行验证。验证的原理是，从证书中提取出内容和签名，然后计算出内容的摘要，然后再利用 CA 公钥还原签名，比对摘要是否相同，相同则验证通过。

### 4.3.2.2. 文件签名 USER-02-2

调用密码学模块，计算文件的哈希值，并对哈希进行签名，然后将计算好的哈希值添加在文件最后一行备注。

### 4.3.2.3. 验证签名 USER-02-3

调用密码学模块，验证文件的签名是否正确，并还原出签名前的文件。

### 4.3.2.4. 文件加密 USER-02-4

调用密码学模块，在输入用于对称加解密的密钥后，用 RSA 加密并保存为.PGP 文件，然后用该密钥实现 DES 加密。

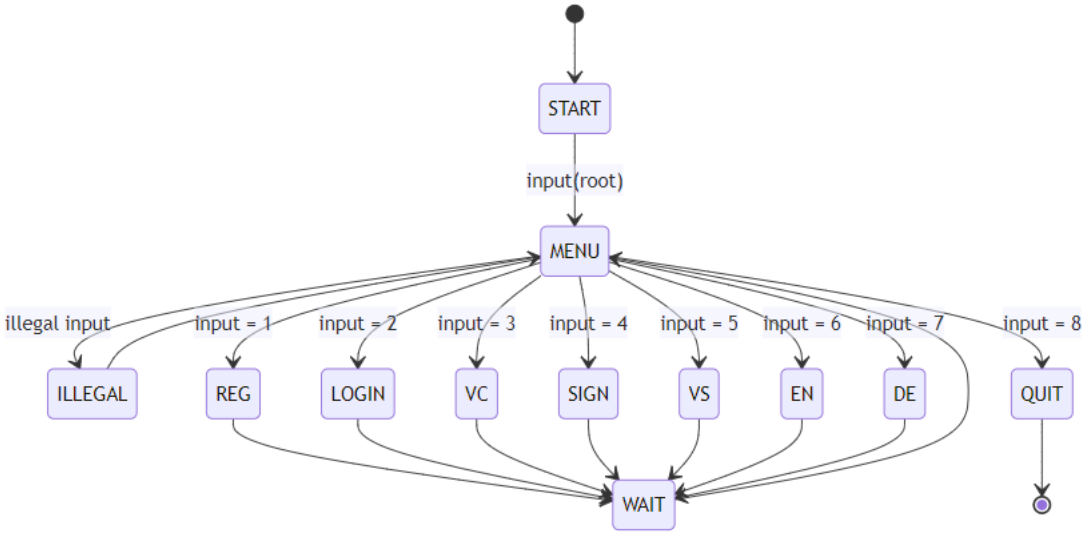
### 4.3.2.5. 文件解密 USER-02-5

调用密码学模块，读取.PGP 文件后，用 RSA 解密并获取用于对称加解密的密钥，然后用该密钥实现 DES 解密。

## 4.4. 菜单模块 MENU

| 模块名称 | 菜单模块 | | | |
|---|---|---|---|---|
| 模块简介 | 实现用户的操作界面，提供交互接口 | | | |
| 模块功能列表 | | | | |
| 序号 | 一级功能 | | 二级功能 | |
| | 功能名称 | 功能编号 | 功能名称 | 功能编号 |
| 1 | 控制模块 | MENU-01 | 状态机 | CA-01-1 |

该模块的作用是与用户实现交互，并根据输入维护一个状态机。状态机如下




### 4.4.1.1. 控制模块 MENU-01

### 4.4.1.2. 状态机

该部分根据用户的交互，建立输入，并控制机的状态转移。