Journal of Security and Safety Technology 安防技术, 2018, 6(4), 67-73
Published Online December 2018 in Hans. http://www.hanspub.org/journal/jsst
https://doi.org/10.12677/jsst.2018.64010

Hans汉斯

Study and Implementation of Simple Key Management Service

Chengxia Liu, Ying Cai, Yanfang Fan

School of Computing, Beijing Information and Technology University, Beijing
Email: cecilia7812@163.com

Received: Nov. 17th, 2018; accepted: Dec. 4th, 2018; published: Dec. 11th, 2018

Abstract

Through studying the basic principle of key management service (KMS), this paper designs a simple KMS system which composed of server side and user side. The primary key is stored on the server side, and the data key is encrypted by the primary key. Users only store the encrypted data key, and when they need to use the data key to do data encryption, the KMS service will be called to decrypt the encrypted data key. Finally, a simple key management service model is implemented and tested.

Keywords

Key Management Service, Primary Key, Data Key, Encryption

一种简单的密钥管理服务研究及实现

刘城霞，蔡　英，范艳芳

北京信息科技大学计算机学院，北京
Email: cecilia7812@163.com

收稿日期：2018年11月17日；录用日期：2018年12月4日；发布日期：2018年12月11日

摘　要

本文通过研究密钥管理服务(KMS)的基本原理，设计了一个由服务端和用户端构成的简单的KMS。在服务端存放主密钥，然后使用用户指定的主密钥完成对数据密钥的加密。用户只部署加密后的数据密钥，然后在需要使用时调用KMS服务解密数据密钥，来实现密钥的安全管理。最后实现了一个简单的密钥管理服务模型并进行了测试。

文章引用: 刘城霞, 蔡英, 范艳芳. 一种简单的密钥管理服务研究及实现[J]. 安防技术, 2018, 6(4): 67-73.
DOI: 10.12677/jsst.2018.64010

关键词

密钥管理服务，主密钥，数据密钥，加密

1. 引言

密钥管理服务是现在常用于云上的一种安全易用的密钥托管服务，它的作用是提供用户简单的、安全的使用密钥的服务，从而达到保密的目的。现在的密钥管理服务多是基于云的，比如阿里云的密钥管理服务[1]，华为云的密钥管理服务[2]，腾讯云的密钥管理服务[3]等等，也有许多学者对各种密钥管理都进行了若干研究[4] [5] [6]，本文要研究的是用传统服务器端及客户端模拟阿里云的密钥管理服务的 KMS 系统。

2. 密钥管理服务的功能设计

一个 KMS(密钥管理服务)的信封加密过程如下：首先用户通过 KMS 服务创建一个主密钥；其次，用户通过密钥管理服务(KMS)产生数据密钥(包括明文的数据密钥和密文的数据密钥)传递给用户；再次，用户使用明文数据密钥来加密文件，产生密文文件；最后，用户将密文数据密钥和密文文件一同保存到持久化存储设备或服务中，等解密时使用。而解密过程是这样的：首先用户先从持久化存储设备或服务中读取密文数据密钥和密文文件；其次用户通过 KMS 服务来解密密文数据密钥，获得明文数据密钥；最后用户使用明文数据密钥解密文件。本文将模拟信封加密的简单密钥管理服务中的信息加解密过程，它是基于多个客户端/一个服务器(C/S)模式的。服务器端用于实现同客户端通信、创建主密钥、生成数据密钥(明文数据密钥和密文数据密钥)以及解密密文数据密钥的功能。客户端用于同服务端通信、实现对明文文件的加密和对密文文件的解密。它的具体功能模块设计，如图 1 所示：

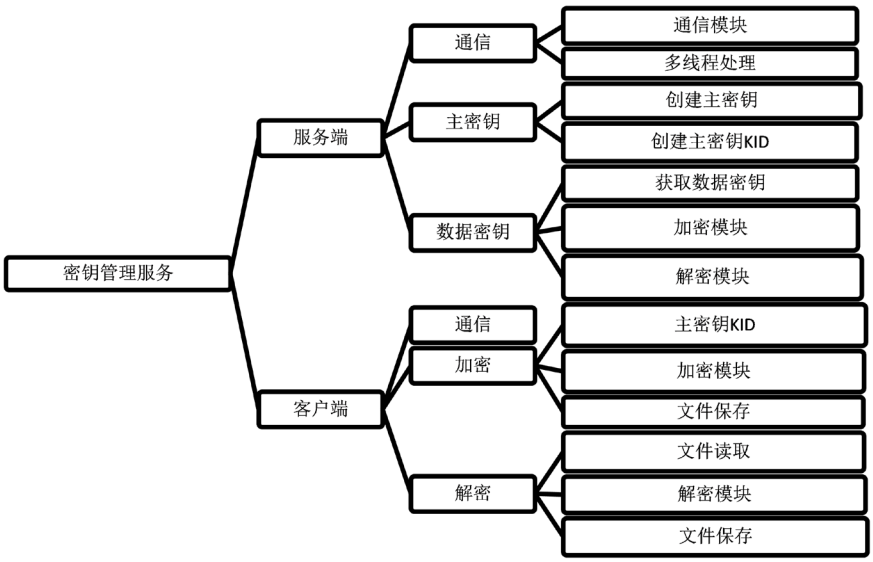

Figure 1. Analog key management service function module diagram
图 1. 模拟密钥管理服务功能模块图

服务端可以调用多线程处理模块来处理多个客户端的请求；可以创建主密钥，生成长度明文数据密钥；用主密钥加密生成密文数据密钥以及解密密文数据密钥模块来生成明文数据密钥。客户端可以保存主密钥 KID；判断是否有可以使用的主密钥；寻找客户端最近创建的主密钥；对明文文件的加密；找到与密文文件对应的密文数据密钥；对密文文件的解密；文件的读取以及文件的保存等。

2.1. 服务器端加解密部分

由于模拟信封加密的简单密钥管理服务的设计与实现是基于一个服务器与多个客户端模式的，所以在服务器加入了线程处理。服务端加密的流程如下：

1) 服务端监听端口，等待客户端连接。

2) 有客户端连接，与客户端建立连接。

3) 读取客户端的输入(创建主密钥)，服务器调用 createCMK()方法创建主密钥，然后生成 KID，并向客户端返回创建成功和 KID 或创建失败的信息。

4) 服务器读取客户端的输入(获取数据密钥)，服务器调用 getDatakey()方法创建明文数据密钥，然后调用 encrypt()方法对明文数据密钥进行加密，得到密文数据密钥，并向客户端返回明文数据密钥和密文数据密钥。

5) 等待客户端关闭连接，然后服务器

对应的服务端解密的流程如下：

1) 服务端监听端口，等待客户端连接。

2) 有客户端连接，与客户端建立连接。

3) 读取客户端的输入(密文数据密钥和 KID)，然后服务器得到主密钥，再将密文数据密钥和主密钥作为 decrypt(String data, String key)方法的参数，产生明文数据密钥，并向客户端返回该明文数据密钥。

4) 等待客户端关闭连接，继续监听端口。

2.2. 客户端加解密部分

客户端加密的流程如下：

1) 与服务器建立连接。

2) 向服务器发送请求(创建主密钥)，接受服务器的返回信息(创建成功+KID 或创建失败)，保存 KID。

3) 如果创建主密钥成功，向服务器发送请求(获取数据密钥)，接受服务器的返回信息(明文数据密钥和密文数据密钥)。

4) 输入明文或打开明文文件，然后使用明文、明文数据密钥作为 encrypt()方法的参数对明文进行加密，得到密文。

5) 保存密文文件和密文数据密钥。

6) 如果想创建新的数据密钥继续加密，跳转到步骤 3)。如果想使用刚创建的数据密钥加密新文件，跳转到步骤 4)。如果加密完成，跳转到步骤 7)。

7) 关闭密钥管理服务客户端的加密界面。

客户端解密的流程如下：

1) 与服务器建立连接。

2) 打开 KID 文件、密文文件。

3) 根据密文文件找到对应的密文数据密钥文件。然后向服务器发送请求(密文数据密钥+KID)，接受服务器的返回信息(明文数据密钥)。

4) 使用密文文件、明文数据密钥作为 decrypt()方法的参数对密文文件进行解密。

5) 保存明文文件。如果继续解密，跳转到步骤 2)。如果解密完成，跳转到步骤 6)。

6) 关闭密钥管理服务客户端的解密界面。

3. KMS 功能详细设计

3.1. 创建主密钥

主密钥可以直接用来加密少量的数据，但通常会被当做产生数据密钥的输入参数来生成可以加密数据的数据密钥。主密钥是长度为 8 的字符串，可以通过产生随机数的方法来生成主密钥。伪代码如下：

```
String createCMK( ) {
定义一个变量，类型为 StringBuffer ，存放每次生成的主密钥;
for (int i = 0; i < 8; i++) {
    定义一个整型变量;
    产生 0~57 的随机数，并将其赋值给整型变量;
    if (整型变量 <= 9) {
        将整型变量(0~9)转为 char 型变量(0~9)
        将 char 型变量追加到 StringBuffer 类型的变量字符序列中;
     }
    else if (整型变量 < 33) {
        将整型变量(10~33)转为 char 型变量(A~Z);
        将整型变量(33~57)转为 char 型变量(a~z);
        将 char 型变量追加到 StringBuffer 类型的变量字符序列中;
    }
}
      定义一个字符串型变量，变量名为 cmk;
      将 StringBuffer 类型的变量转换成字符串型，并将其赋值给定义好的变量 cmk;
      返回 字符串型变量 cmk;
}
```

主密钥与主密钥 KID 应该是一一对应的。服务器通过对主密钥做变换生成 KID。用户向服务器发送主密钥 KID，服务端可以根据 KID 找到对应的主密钥。主密钥和主密钥 KID 之间的转化通过使用 BASE64 编码方法来完成。将主密钥进行编码之后可以得到主密钥 KID，将主密钥 KID 解码也可以得到主密钥。生成主密钥 KID 的流程图，如图 2 所示。

3.2. 获取数据密钥 GetDataKey

用户可以使用这个密钥进行本地数据的加密，会返回一个明文的数据密钥。然后调用 encrypt()方法生成数据密钥加密后的密文。数据密钥的加密是用与主密钥 KID 对应的主密钥来加密的。密文的数据密钥可以调用 decrypt()进行解密得到明文的数据密钥。每获取一次数据密钥，所产生的数据密钥都应该是互不相同的，也就是说一次一密。伪代码如下：

```
String getDatakey(){
定义一个 StringBuffer 类型变量，用来存放产生的数据密钥;
创建一个新的随机数生成器，用来生成每一位数据密钥;
for (int i = 0; i < 8; i++) {
```

定义一个整型变量，用来存放产生的随机数；
将产生的随机数转换成 char 型，并将其添加到 StringBuffer 类型变量中；
}
定义一个 String 类型变量 edk ；
将 StringBuffer 类型变量转换成 String 类型，并将其赋值给 edk;
返回 String 类型变量 edk;
}

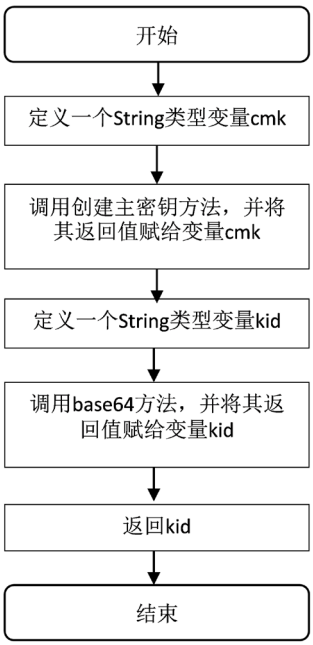

Figure 2. Flow chart of KMS server generating master key KID method
图 2. KMS 服务器生成主密钥 KID 方法的流程图

3.3. 获取密文文件与密文数据密钥的对应关系

当用户进行解密操作时，需要用到待解密的密文文件、与密文文件对应的密文数据密钥和对应的主密钥 KID 文件。当进行加密操作时，用户每保存一次密文文件，系统就会自动生成一个存有密文文件和密文数据密钥对应关系的文件。当用户进行解密操作时，只要用户找到需要解密的密文文件，系统就可以自动读取存有密文文件和密文数据密钥对应关系的文件找到与之对应的密文数据密钥。并且由于只有用户本人的计算机上有存有密文文件和密文数据密钥对应关系的文件，假如有人窃取了文件和主密钥 KID，也无法解密该文件。

进行加密操作时，自动创建一个存有密文文件和密文数据密钥对应关系的文件。然后进行解密操作时，输入需要解密的密文文件，自动读取存有密文文件和密文数据密钥对应关系的文件找到与之对应的密文数据密钥。

4. 加解密过程测试

4.1. 加密过程

客户端加密测试结果如图 3 所示：

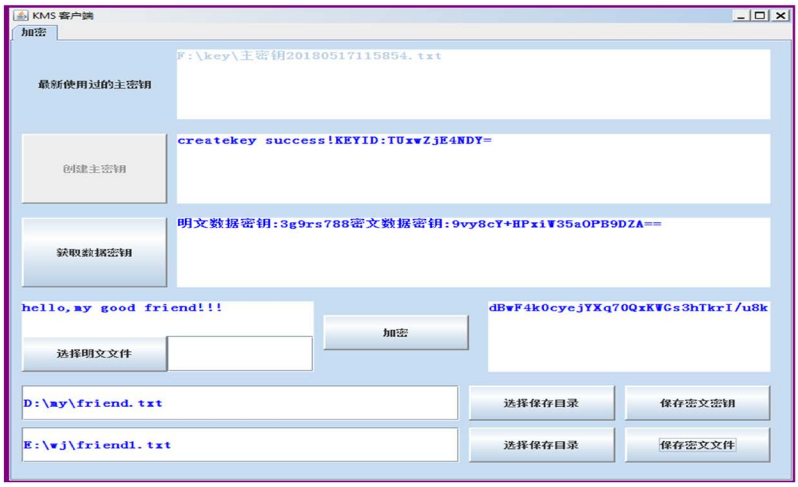

Figure 3. KMS client encryption test result diagram
图 3. KMS 客户端加密测试结果图

4.2. 解密过程

客户端解密测试结果如图 4 所示：

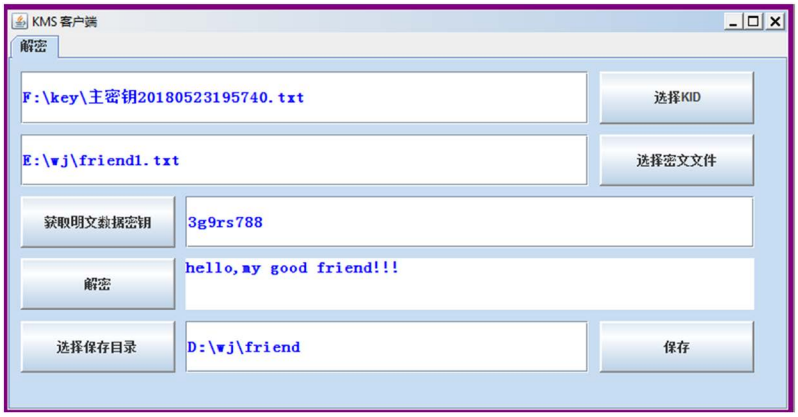

Figure 4. KMS client decryption test results
图 4. KMS 客户端解密测试结果

5. 总结

该简单 KMS 是模拟的阿里云的密钥管理服务的过程，主要通过生成密钥、信封加密、密钥解密等来完成密钥的管理和加解密过程。模拟过程中用的传统密钥进行的，但不论是传统密钥还是其他密钥(比如量子密钥)，只要将接口统一，过程是不变的。

项目基金

本项目得到 2018 网络文化与数字传播北京市重点实验室开放课题资助；中央引导地方科技发展专项：量子通信技术创新与行业应用项目资助(编号：Z171100004717002)。

参考文献

[1] 阿里云 KMS 参考文档[EB/OL]. https://www.alibabacloud.com/help/zh/product/28933.htm

[2] 华为云 KMS 参考文档[EB/OL]. https://www.huaweicloud.com/product/kms.html
[3] 腾讯云 KMS 参考文档[EB/OL]. https://cloud.tencent.com/product/kms#wiki
[4] 谢立军. 云计算环境下密钥管理系统设计与优化[D]: [硕士学位论文]. 郑州: 解放军工程大学, 2013.
[5] 张勇. 密钥管理中的若干问题研究[D]: [博士学位论文]. 上海: 华东师范大学, 2013.
[6] 潘维勇, 袁聿卿. 在云计算环境下密钥应用服务的管理设计与实现[J]. 信息技术, 2015(33): 13-14, 16.