枢钥管理系统

范学鹏

August 29, 2022

1 需求分析

在 Fidelius 及基于 Fidelius 的系统中，枢钥是十分重要的概念。在某些情况下，用户可以自己生成并管理相应的枢钥。然而，在面向企业的使用场景下，终端用户管理枢钥是复杂且低效的。为了降低用户的使用成本，引入枢钥管理系统。

2 枢钥管理

2.1 关于枢钥

枢钥是一对基于椭圆曲线生成得公私钥（枢公钥 P^S，枢私钥 S^S）。在 Fidelius 中，提供了 `yterminus` 命令行工具。用户可以使用 `yterminus` 在非可信环境下生成。同时，Fidelius 也提供了 Javascript 库，Python 库，以便在具体的业务系统上操作枢钥。

枢钥可以使用上述工具生成，并直接以文件的形式存储，我们称这种方式为*直接方式的枢钥*; 也可以使用硬件设备进行管理，例如 OpenGPG 智能卡（OpenGPG 物理密钥，下文简称*智能卡*），我们称这种方式为*智能卡方式的枢钥*。由于无法从智能卡中提取私钥，相比于通过 USB 闪存盘存取

Figure 1: 各种不同的 OpenGPG 智能卡。智能卡有额外的 ID 标识。

私钥，对枢私钥的保护更周全。在智能卡中，无需将私钥暴露给其他程序，只能将数据交给物理密钥进行解密处理，这些智能卡的样子如图 1 所示。

在面向企业的使用场景下，既存在以文件的方式存储枢私钥的情况，也存在使用智能卡形式存储枢私钥的情况。支持智能卡，需要基于 OpenGPG 进行二次开发。同时也需要考虑“信创”或国密标准下的智能卡。

2.2 枢钥管理系统

枢钥管理系统定义三个角色：

- 用户：是枢钥管理的对象，用户的唯一标识为 `UID`;
- 管理员：根据枢钥管理系统提供的操作界面，对用户进行管理;
- 业务系统：依赖于枢钥管理系统提供的服务，提供业务的外部系统，例如数据服务平台。

管理员角色可以是虚拟的，也可以是物理的。虚拟管理员是指可以通过在业务系统中内置相应的管理功能，例如，自动为每个用户生成枢私钥并注册。物理管理员则是有实际的管理或运维人员担任，为此需要提供相应的命令行工具或图形化操作方式。

枢钥管理包括以下几个功能，如图 2 所示。

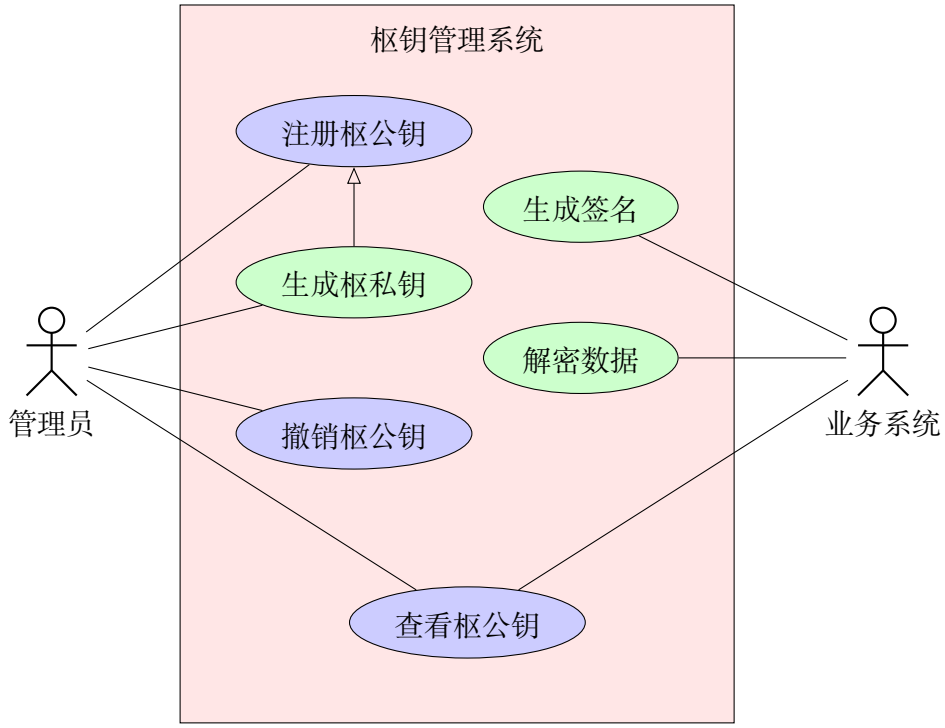

Figure 2: 枢钥管理系统用例图。图中绿色用例需要使用枢私钥，由于枢私钥可能是智能卡方式的，因此绿色用例是可选的。

枢钥管理系统需要同时考虑直接方式的枢钥和智能卡方式的枢钥。在直接方式下，系统需要管理枢私钥; 在智能卡方式下，系统仅管理公钥。系统中两种方式是并存的。

枢钥管理系统不提供签名验证及数据加密功能，这两个功能在有枢公钥的情况下，可以自行完成。

3 系统功能

本节我们对图 2 中描述的各个用例的功能进行描述。

3.1 注册枢公钥

输入: 用户标识 UID, 枢公钥 P^S，以及枢公钥的相关信息，这些相关信息包括是否为智能卡、系统是否持有枢私钥、智能卡 ID、智能卡种类等。

输出： 系统中存储枢公钥及相关信息。

描述： 系统需要在对用户的身份进行核实（如实名认证）后，调用该功能。一个用户可以有多个枢公钥。

3.2 生成枢私钥

输入: 用户标识 UID。

输出: 系统中存储枢私钥。

描述： 在生成枢私钥后，需要立刻注册相应的枢公钥，注册枢公钥成功后，返回相应的公钥。注意，私钥对用户是不可见的。为了保证枢私钥的安全，可以使用额外的数据安全技术。

3.3 撤销枢公钥

输入： 用户标识 UID，枢公钥。

输出： 相应的枢公钥标识为不可用。

描述： 用户的智能卡遗失、怀疑私钥泄漏时，应该使用该功能撤销相应的枢公钥。

3.4 生成签名

输入： 用户标识 UID, 枢公钥，以及需要签名的内容（哈希）。

输出： 返回签名信息。

描述： 该功能仅在系统存储了相应的枢私钥的情况下可用，如果没有枢私钥，应该返回错误信息。

3.5 解密数据

输入： 用户标识 UID, 枢公钥，以及需要解密的内容。

输出： 返回原文。

描述： 该功能仅在系统存储了相应的枢私钥的情况下可用，如果没有枢私钥，应该返回错误信息。为了保证原文不被泄露，应该在可信的数据传输通道（如 HTTPS）上使用该功能。

3.6 查看枢公钥

分为两个功能，查看一个用户标识下的所有枢公钥，以及查看一个枢公钥对应的用户标识。

4 未来工作

一般而言，密钥相关的系统会给密钥设定一个有效时间，标识在该时间内，密钥是可用的。该功能作为可选项，可以作为未来系统的扩展。