基于 SHA256 和 RSA 的数字签名

智能 1502 陈东宁 0918150203

1. RSA 算法实现

在尝试了自己动手造车轮，用 C++编写大数类和素性检测函数，以及尝试使用 C 及 C++的附加大数类库（譬如 MPUINT, GMP, GUN Cryto, boost 等）之后，终于意识到自己编写的类和函数过于低效，而附加类库又过于庞杂。

在经过无数次失败的尝试后，决定转用标准库自带了大数类的 JAVA。

果然，JAVA 原生的大数库使用简便而且高效。剩下的只需要自己编写 RSA 的算法实现即可。终于让我意识到了 C++的局限性。

不过，这个过程也不是一无所获。查阅了大量的资料，让我深刻地了解了 RSA 算法和素性检测算法，同时也意识到了大数运算的复杂性。

2. SHA 函数

相较于 RSA，SHA 的实现过程就没那么坎坷了，但也绝非易事。在 Wikipedia 上查找了 SHA256 的伪代码实现，将之转化为 JAVA 代码。

第一次运行，计算英文 Hello 的 hash 值并于网上的在线 hash 值计算软件比对，发现不一致。遂查看网页的源代码，然后逐条执行，与

自己的程序比对，反复纠错。其中还遇到了负数右移的问题。在查找资料后，将右移>>改为无符号右移>>>，配合其他一些地方的改动，终于顺利解决。

3. 签名函数

与前两者比较起来，签名函数是最轻松的。基本没遇到什么问题，按着公式顺利解决。

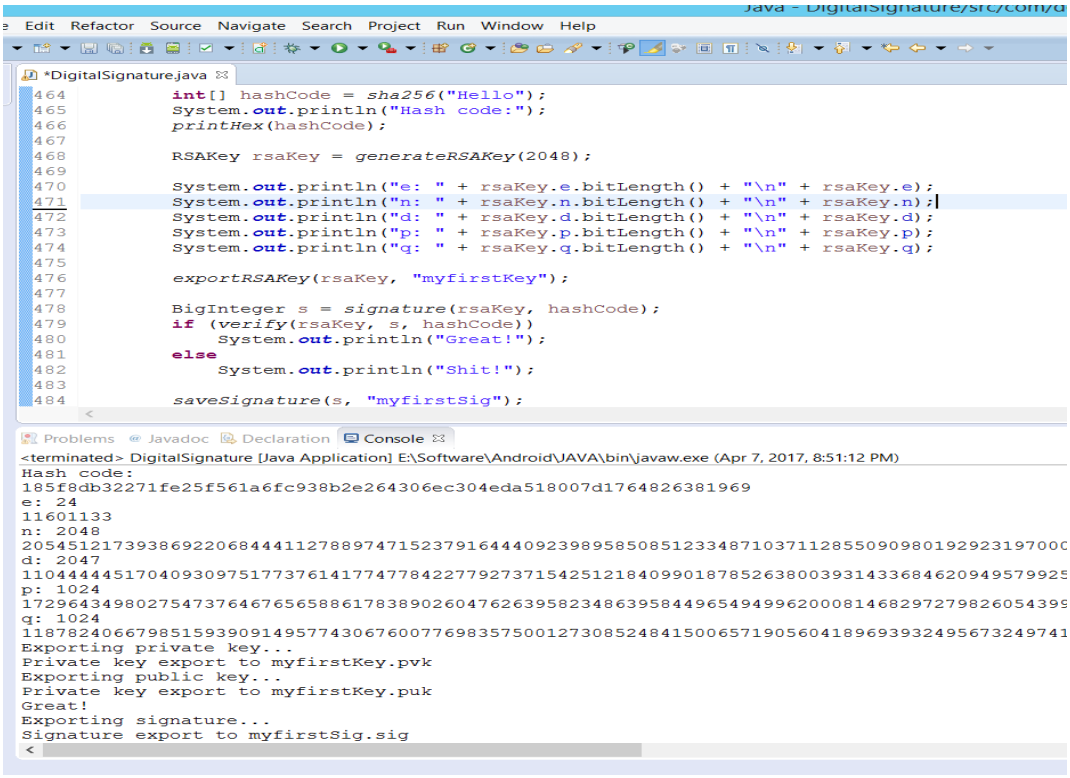


图 1　主要功能通过测试

正确计算出 hash 值，成功生成并导出 RSA 密钥，并对文件进行签名，导出签名

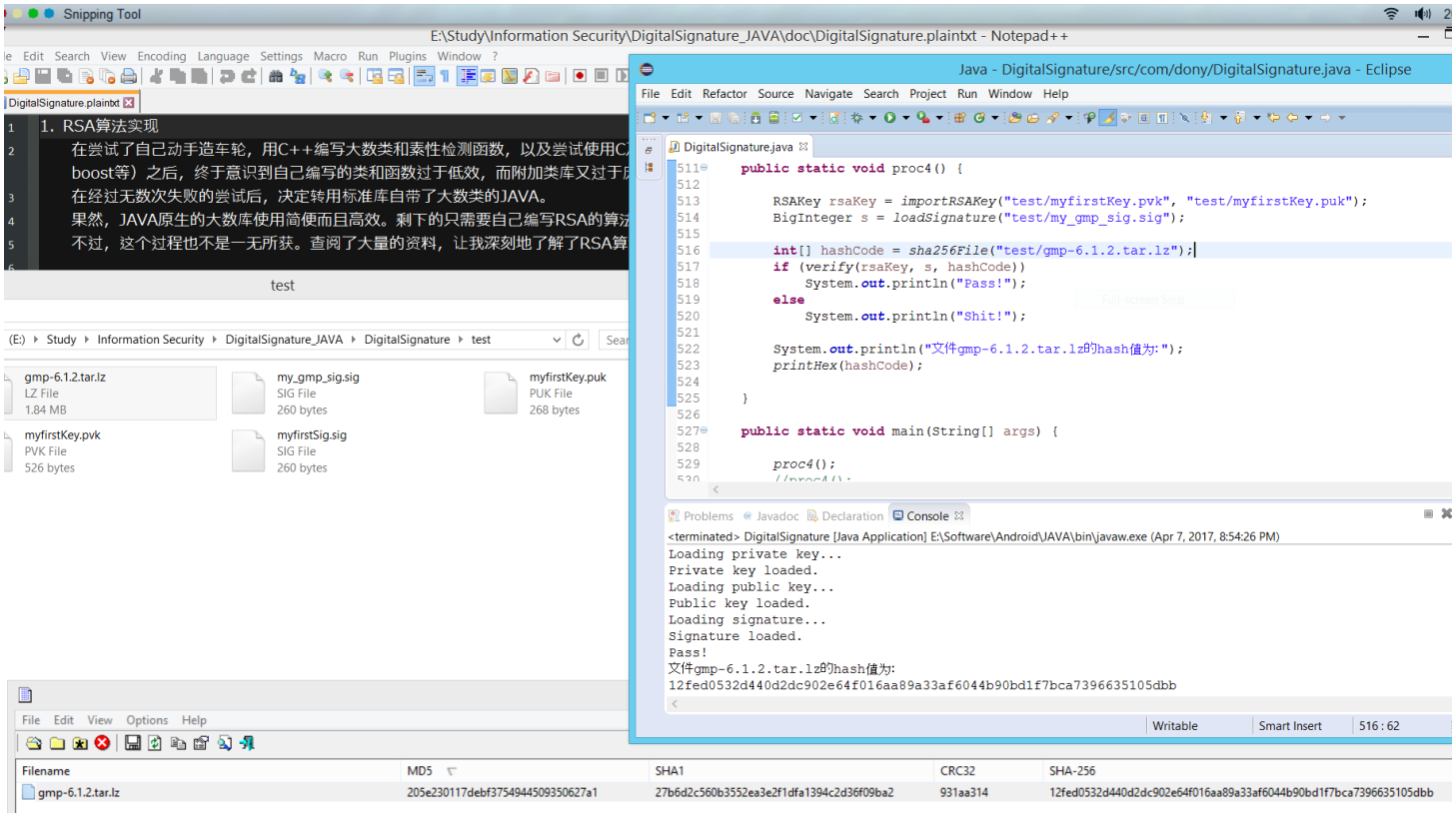
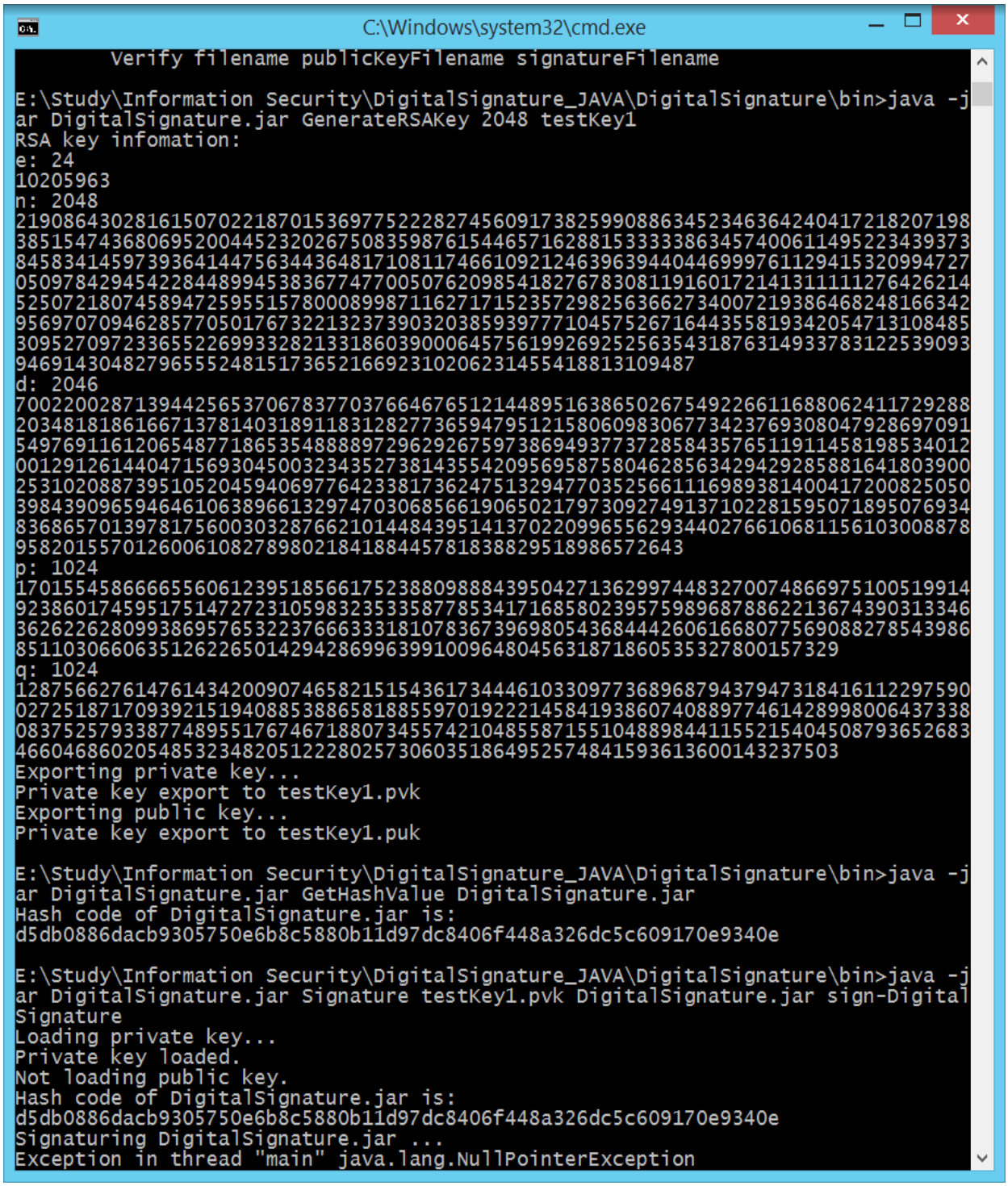


图 2　验证签名功能通过测试

成功读取 RSA 密钥，文件 hash 值正确，签名验证通过



生成 RSA 密钥

e 为公钥

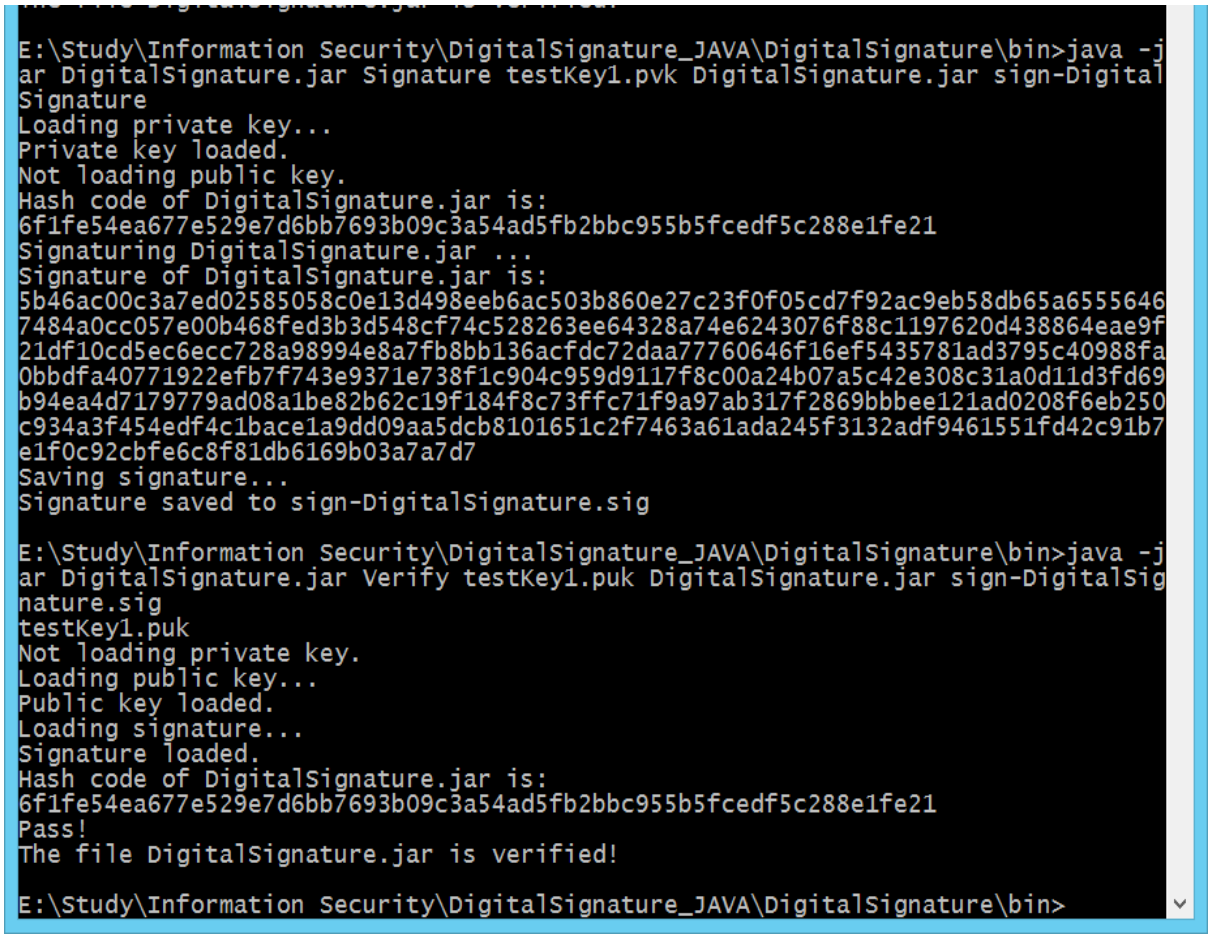
n 为模数

d 为私钥

p，q 为所用的素数

计算文件 hash 值

图 3 在控制台生成 RSA 密钥和计算文件 hash 值
(由于控制台不支持中文因此输出英文)



用生成的 RSA 私钥对文件签名

用生成的 RSA 公钥对签名进行验证

图 4 用生成的 RSA 密钥签名文件并验证通过