现代密码学 作业 3

陈庆之 2021011819

2023 年 6 月 21 日

1 公钥密码算法实现

代码见 `main.py`。

我们实现了一个 RSA-2048 加密/解密算法，可以生成合法的密钥对，并用指定的密钥对将至多 255 个字符长度的字符串加密或将对应的密文解密。

1.1 生成密钥对

我们使用 **Miller-Rabin** 算法，检验 40[1]轮判断质数。在实际使用中，这个轮数下生成密钥对的时间几乎无法感知，可忽略不计。

Miller-Rabin 算法的基本思路是结合整除性检验和平方根检验。我们先迭代 10 个小素数进行特判，之后将待检验的整数 n 写成 $m \times 2^k$ 的形式。接着我们选取随机数 $a \in [2, n-2]$（这是为了防止平方根法直接结束），然后计算 $y = a^r \mod n$，并连续 $k-1$ 次平方 y。如果某次平方后的结果为-1，意味着平方根检验成功，因为下一次平方就成为 1；如果某次平方后的结果为 1，意味着平方根检验失败（因为上一次结果肯定不为-1），n 没有通过检验。如果成功通过了 40 轮检验，就认为 n 是一个强伪素数。

生成密钥对的过程是：随机生成两个大 (1023 位二进制) 质数相乘得到 n，然后寻找与 $\phi = (p-1)(q-1)$ 互素的小 (16 位二进制) 质数 e，最后计算 $d = e^{-1} \mod n$，将 (n, e) 作为公钥，(n, d) 作为私钥。

1.2 加密

为了成功加密，我们要求由信息生成的 $m \leq n$。因此，我们将输入的字符串对应的 ASCII 码视作一个至多 2040 位的二进制数作为 m，然后计算 $m^e \mod n$，以十六进制形式输出。

[1]参考了这篇讨论。

1.3 解密

将十六进制格式的整数输入作为 c，计算 $c^d \mod n$，然后每 2 位十六进制数对应 1 位 char，依次输出。

上述所有数的计算均使用 gmp 库实现。

1.4 使用方法及运行实例

需确保正确安装 gmp 库，编译时添加`-lgmp -lgmpxx`。（或直接使用附上的 CMakeList.txt 文件，以 CMake 形式安装）

程序会提示输入 d(ecrypt), e(ncrypt) 或 g(enerate)，输入其他内容将直接停止。

参考的密钥对及明、密文见 `example.txt`。

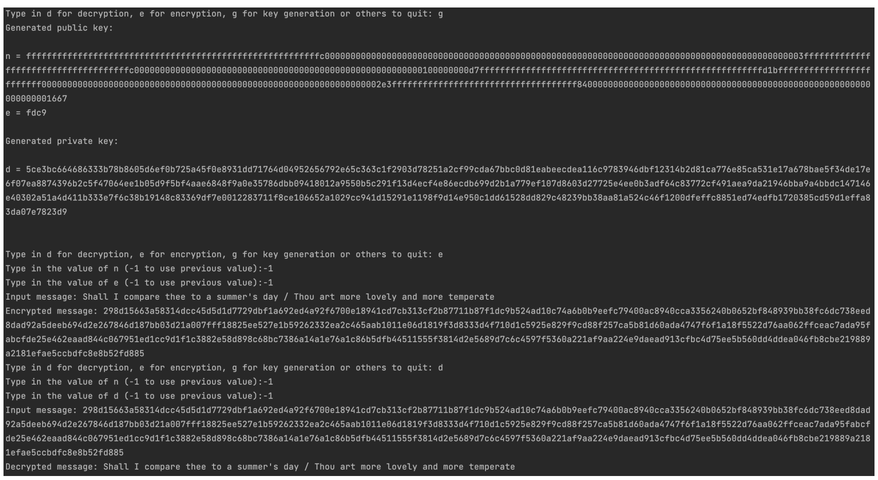

图 1: 运行截图

2 公钥密码算法计算

1. 我们对 $\forall x \in \mathbb{Z}_{11}$，计算如下表：

x	0	1	2	3	4	5	6	7	8	9	10
$x^3+x+6 \pmod{11}$	6	8	5	3	8	4	8	4	9	7	4
是否是二次剩余			是	是		是		是	是		是
$z^3 \pmod{11}$			4	5		9		9	3		9

因此所有的点为：

(2, 4), (2, 7), (3, 5), (3, 6), (5, 9), (5, 4),

(7, 9), (7, 2), (8, 3), (8, 8), (10, 9), (10, 2).

2. 证明：依次计算 $(2,7)^i$，得到下表：

i	$(2,7)^i$	i	$(2,7)^i$
0	O	7	(7, 2)
1	(2, 7)	8	(3, 5)
2	(5, 2)	9	(10, 9)
3	(8, 3)	10	(8, 8)
4	(10, 2)	11	(5, 9)
5	(3, 6)	12	(2, 4)
6	(7, 9)		

可以看出，i 取遍 $[0,13)\cap\mathsf{Z}$ 时，$(2,7)^i$ 取遍 $E_11(1,6)$ 上的所有点。因此 $\alpha=(2,7)$ 是本原元。

3. 自选密钥 $a=4$, 计算 $Q=a\alpha=4\times(2,7)=(10,2)$ 作为公钥, 4 作为私钥。加密时, 计算 $kP=3\times(2,7)=(8,3), kQ=3\times(10,2)=(2,4)$, 验证 kP,kQ 各分量不为零, 因此密文 $C=(kP,m+kQ)=((8,3),(2,4)+(5,2))=((8,3),(2,7))$.

3 数字签名算法

注意到两个消息的签名中，$(\alpha^k \mod p)=\gamma=23972$ 相同，又由于 α 依定义是本原元，因此 k 相同。由加密过程，我们知道

$$\begin{cases}\delta_1=(x_1-a\gamma)k^{-1} \mod (p-1)\\ \delta_2=(x_2-a\gamma)k^{-1} \mod (p-1)\end{cases} \tag{1}$$

据此我们可以构造

$$\begin{cases}x_1=a\gamma+\delta_1 k \mod (p-1)\\ x_2=a\gamma+\delta_2 k \mod (p-1)\end{cases} \tag{2}$$

相减得

$$x_1-x_2=k(\delta_1-\delta_2) \mod (p-1), \tag{3}$$

即

$$9421=k\times 10915 \mod 31846. \tag{4}$$

使用计算器可以求得 $10915^{-1}=22317 \mod 31846$, 进而知道

$$k = 22317 \times 9421 = 210248457 = 1165 \mod 31846. \tag{5}$$

为计算 a, 注意到

$$\delta_1 = (x_1 - a\gamma)k^{-1} \mod (p-1), \tag{6}$$

化简为

$$\gamma a = x_1 - \delta_1 k \mod (p-1) \tag{7}$$

代入数据:

$$23972 \times a = 8990 - 31396 \times 1165 = 23704 \mod 31846 \tag{8}$$

考虑到 $23972^{-1} \mod 31846$ 不存在，同除以 2[2]得

$$11986 \times a = 11852 \mod 15923 \tag{9}$$

使用上面的计算器知道 $11986^{-1} = 182 \mod 15923$，因此

$$a = 182 \times 11852 \mod 15923 = 7459. \tag{10}$$

[2]这么做的合法性是因为：若 $\exists \lambda \in \mathbb{Z}, \lambda a = \lambda r(\mod p)$, 则 $\exists k \in \mathbb{Z}$, 故 $\lambda a = k\lambda p + \lambda r$, 则 $a = kp + r$, 即 $a = r \mod p$.