

# RSA 和 DES 加密实验

张海斌\*

2019 年 11 月

## 目录

1 界面说明	1
2 RSA 实现	1

## 1 界面说明

如Figure 1，数据输入界面和之前的 DES 实验的前端界面一致。手动输入 DES 密钥、和初始偏移量，并选择加解密方法和要加密的文件，最后点击”开始加密/解密”按钮即可加密。不同之处在于结果显示多了很多 RSA 的加解密的结果，如Figure 2：RSA 公钥、RSA 私钥 分别用于对 DES 密钥进行加密得到密文，和将密文解密得到原始的 DES 密钥。还有 RSA n 表示 RSA 取模的大小，以及 RSA 加密和 RSA 解密 分别对应用 RSA 加密 DES 密钥和解密 RSA 密文得到的结果。

在访问该前端网页的时候，会自动获取新的随机密钥对并显示在界面上。提交 DES 加/解密操作之后会显示出 DES 加解密的结果，同时也会显示出 RSA 对 DES 密钥进行加解密的结果。操作成功提示界面也略有更改，如Figure 3。

运行环境需要 Python 和 Flask。执行 run.bat 或 run.sh 脚本即可运行。示例输入数据见 example 文件夹里。text.txt 为明文，key.txt 为 DES 密钥，iv.txt 是 CBC 加密的初始偏移量，ciphertext.txt 是 DES 加密后的结果，将它用通过 RSA 加密算法传输的 DES 密钥解密后的结果与 text.txt 明文相同，说明 DES 和 RSA 的算法都是正确的。

## 2 RSA 实现

RSA 的 Python 实现代码在 rsa.py 文件中。主要函数功能有快速幂取模、扩展欧几里得算法、Miller-Robin 检测质数、获取一个指定位数的质数。在这之上有三个用于进行 RSA 加解密的函数：生成密钥对的函数 generateKeys、对明密文加解密的函数 encryptRSA 和 decryptRSA。

质数检测函数中使用多次 Miller-Robin 的方法检测质数，保证几乎可以完全确定是质数。检测次数可以通过修改变量 primeTestRound 的值改变。在生成 RSA 密钥对时 e 的选取，我直接选择一个 16 位的质数作为 e 的值。

由于 64 位的 DES 密钥开头可能是 0 开头，导致最后 RSA 解密出的结果长度小于 64 位，所以我用 0 扩充 RSA 解密的结果到 64 位来避免加解密后长度不一致的问题。

---

\* 学号 17307130118

DES + RSA Lab

密钥

00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

CBC 加密初始偏移量

00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

加密

解密

text.txt

Browse

开始加密/解密

结果

01001111 11100000 01010111 10000110 01110001 00000100 00000001 11100011  
10011110 00100011 10110100 10011000 01010100 00011110 10001111 10111001  
11110100 11111100 10110111 10110011 00101110 00111111 01000110 01011110  
10101110 01000010 11101011 11101101 11001100 11101101 01010010 10110010  
00111101 11100111 00001001 10101101 00010000 00101110 11010101 10000000

RSA 公钥

1110101011110101

RSA 私钥

100101001101110100111010110000100101100100011110001110000001010  
01010101000001011100111101111010011011111111100001001100010010  
000000011110111101010110001100010001101000000011000101001101001  
01000011001111111000111100110010101101111000100001111010011110  
1

Figure 1: 前端界面

RSA 公钥	1110101011110101
RSA 私钥	100101001101110100111010110000100101100100011110001110000001010 01010101000001011100111101111010011011111111100001001100010010 000000011110111101010110001100010001101000000011000101001101001 010000110011111110001111001100101011011111000100001111010011110 1
RSA n	100110101001110101001011111010111110101101110000100110101010110011 111111010001111111001100101101000010100111001001101110010000000011 00011110101101000000111110101000000111101111001111101111011100110 011101000011100001100111001110001000110011111000010110011
RSA 加密	111111111011110010010010010011010010001000011010100100001010001 011010101001100101110100111101000000011010101111010101111100001 100101100100101001001101100110111000011110100111110110010000101 000100101001101011011111111001111000001011001101011000101001010 01
RSA 解密	000100110011010001010111011110011001101110111100110111111111000 1

Figure 2: RSA 加解密结果

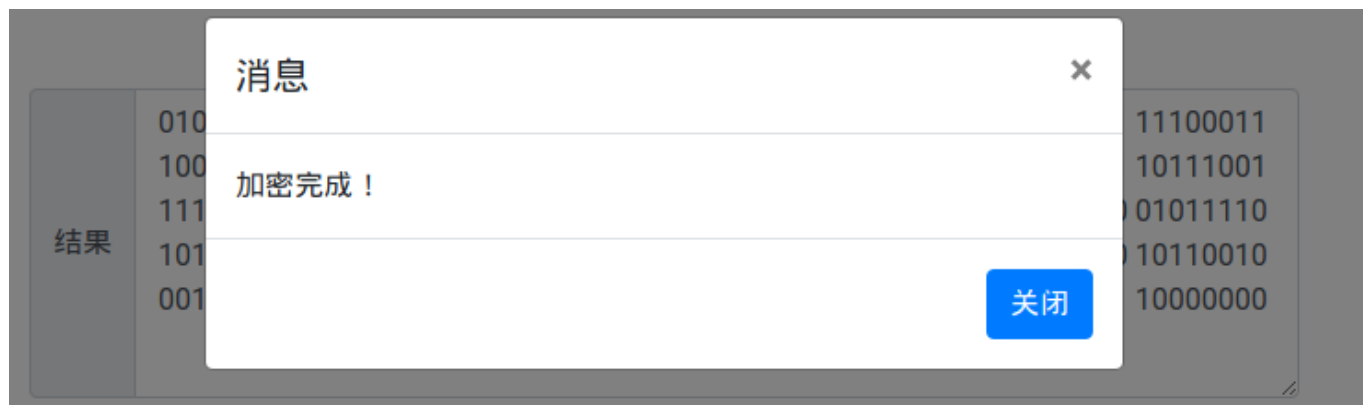


Figure 3: 提示界面