密码学实验实验报告四

18374480-黄翔

2021 年 4 月 11 日

1 实验目的

1. 理解 Fesitel 结构及 DES 算法的原理，并掌握加解密流程
2. 了解弱密钥与半弱密钥
3. 对算法进行深入理解和思考，开始尝试算法的优化与改良

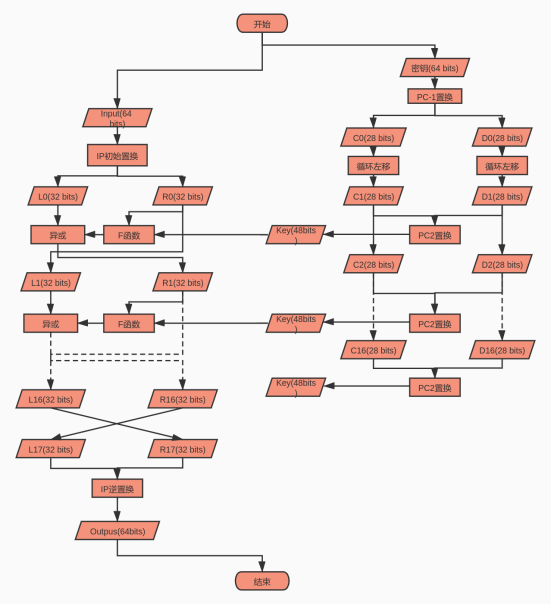
2 实验环境

python 3.9.1+

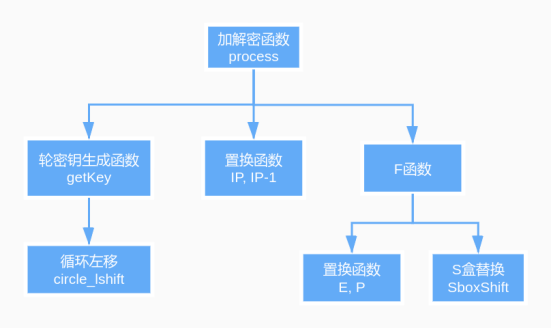
3 实验内容

3.1 DES 加解密算法

3.1.1 算法流程图



3.1.2 函数调用关系



3.1.3 算法伪代码

Algorithm 1 DES 加密

Input: 明文 $P(64\text{ bits})$，密钥 $K(64\text{ bits})$

Output: 密文 $C(64\text{ bits})$

1: **function** $getRoundKey(K)$
2: $roundKey = []$
3: $K(56bits) \leftarrow$ 对 K 做 $PC1$ 置换
4: $C_0, D_0 \leftarrow K[0:28], K[28:]$
5: **for** $i \leftarrow 1$ to 16 **do**
6: $C_i, D_i \leftarrow C_{i-1}, D_{i-1}$ 循环左移一位到两位
7: $K_i = C_i D_i$
8: $K_i \leftarrow$ 对 K_i 做 $PC2$ 置换
9: $roundKey \leftarrow K_i$
10: **end for**
11: **return** $roundKey$
12: **end function**
13: **function** $Encrypt(P, K)$
14: $roundKey \leftarrow getRoundKey(K)$
15: $P \leftarrow$ 对 P 做 IP 初始置换
16: $L_0, R_0 \leftarrow P[0:32], P[32:]$
17: **for** $i \leftarrow 1$ to 16 **do**
18: $eR \leftarrow$ 对 R_{i-1} 做 E 扩展置换
19: $inputS \leftarrow eR \oplus roundKey[i]$
20: $outputS \leftarrow$ 对 $inputS$ 做 S 盒代替
21: $pR \leftarrow$ 对 $outputS$ 做 P 盒置换
22: $L_i \leftarrow R_{i-1}$
23: $R_i \leftarrow L_{i-1} \oplus pR$
24: **end for**
25: $C \leftarrow R_{16}L_{16}$
26: $C \leftarrow$ 对 C 做 IP^{-1} 置换
27: **return** C
28: **end function**

Algorithm 2 DES 解密

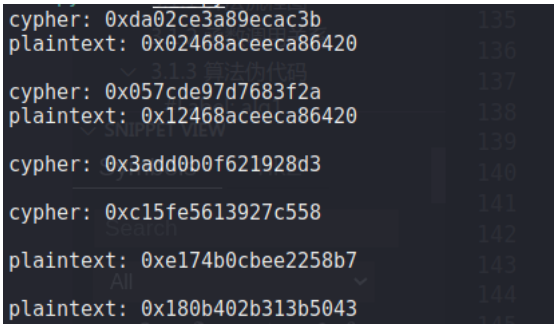
Input: 密文 C(64 bits)，密钥 K(64 bits)

Output: 明文 P(64 bits)

1: **function** $getRoundKey(K)$
2: $roundKey = []$

```
    K(56bits) ← 对 K 做 PC1 置换
    C_0, D_0 ← K[0 : 28], K[28 :]
    for i ← 1 to 16 do
        C_i, D_i ←C_{i−1}, D_{i−1} 循环左移一位到两位
        K_i = C_i D_i
        K_i ← 对 K_i 做 PC2 置换
        roundKey ← K_i
    end for
    return roundKey
end function
function Decrypt(C, K)
    roundKey ← getRoundKey(K)
    将 roundKey 中元素顺序倒置
    C ← 对 C 做 IP 初始置换
    L_0, R_0 ← C[0 : 32], C[32 :]
    for i ← 1 to 16 do
        eR ← 对 R_{i−1} 做 E 扩展置换
        inputS ← eR ⊕ roundKey[i]
        outputS ← 对 inputS 做 S 盒代替
        pR ← 对 outputS 做 P 盒置换
        L_i ← R_{i−1}
        R_i ← L_{i−1} ⊕ pR
    end for
    P ← R_16 L_16
    P ← 对 P 做 IP^{−1} 置换
    return P
end function
```

3.1.4 测试样例及结果截图



3.1.5 总结

DES 雪崩效应的观察 DES 算法当明文或密钥发生微笑改变时，将会对密钥产生很大的影响。

明文	密文	相差位
0x02468aceeca86420	0xda02ce3a89ecac3b	32 bits
0x12468aceeca86420	0x057cde97d7683f2a	

表 1: 明文雪崩效应

密钥	密文	相差位
0x0f1571c947d9e859	0xda02ce3a89ecac3b	30 bits
0x1f1571c947d9e859	0xee92b50606b62b0b	

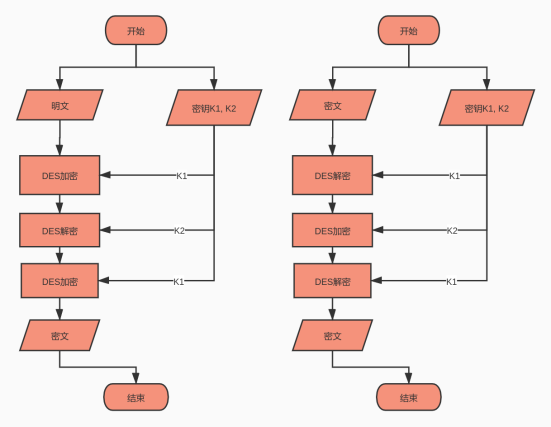
表 2: 密钥雪崩效应

混淆与扩散 DES 函数中存在许多置换表，如 P, E 置换表，将每个字节的影响扩散。S 盒替换则是非线性部分，用于增加混淆。

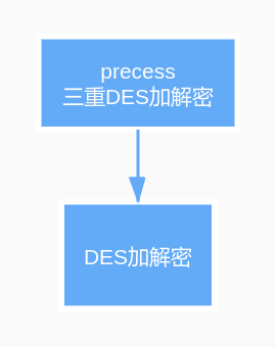
Fesitel 密码结构 编写代码时能明确感受到 Fesitel 密码结构的优势。由于加密与解密结构想同，只需要在解密部分改变轮密钥的输入顺序，就可以实现解密，这极大的简化了 DES 加解密的结构。

3.2 双倍密钥三重 DES 算法

3.2.1 算法流程图



3.2.2 函数调用关系图



3.2.3 算法伪代码

Algorithm 3 三重 DES 加密

Input: 明文 $P(64\text{ bits})$，密钥 $K(128\text{ bits})$

Output: 密文 $C(64\text{ bits})$

1: $Key_1 \leftarrow K[0:64]$

2: $Key_2 \leftarrow K[64:128]$

3: $temp1 \leftarrow DES\_encrypt(P, Key_1)$

4: $temp2 \leftarrow DES\_decrypt(temp1, Key_2)$

5: $cypher \leftarrow DES\_encrypt(temp2, Key_1)$

6: **return** $C \leftarrow cypher$

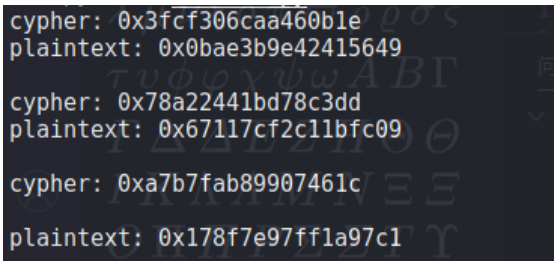
Algorithm 4 三重 DES 解密

Input: 密文 P(64 bits)，密钥 K(128 bits)

Output: 明文 C(64 bits)

1: $Key_1 \leftarrow K[0:64]$
2: $Key_2 \leftarrow K[64:128]$
3: $temp1 \leftarrow DES\_decrypt(C, Key_1)$
4: $temp2 \leftarrow DES\_encrypt(temp1, Key_2)$
5: $plain \leftarrow DES\_decrypt(temp2, Key_1)$
6: **return** $P \leftarrow plain$

3.2.4 测试样例及结果截图



3.2.5 总结

安全性 三重 DES 使用两个不同密钥三次加密，可以提升已知明文的攻击代价，防止中间相遇攻击。对三重 DES 穷举攻击代价是 2^{112} 数量级，用差分密码分析的代价按指数增长，

3.3 弱密钥与半弱密钥

3.3.1 原理

弱密钥 由 DES 轮密钥生成流程不难发现，K_L 只来自于 C_i 的 28 个元素，K_R 只来自于 D_i 的 28 个元素，且每轮密钥对应的 C_i，D_i 中元素只是顺序改变。因此，只要 C_i，D_i 的元素都相同，则每轮选出的轮密钥相同。注意到 DES 密钥有 8 位奇偶校验位不影响密钥的生成。根据 DES 密钥扩展算法可以的到弱密钥如下：

弱密钥
0x0101010101010101
0xFEFEFEFEFEFEFEFE
0xE0E0E0E0F1F1F1F1
0x1F1F1F1F0E0E0E0E
0x0000000000000000
0xFFFFFFFFFFFFFFFF
0xE1E1E1E1F0F0F0F0
0x1E1E1E1E0F0F0F0F

表 3: DES 弱密钥

半弱密钥 当 C_i 与 D_i 元素是已 2 为周期重复出现时，产生的轮密钥只有两种可能值，取决于该轮左移奇数还是偶数。观察 C_i，D_i 的循环左移，当第 i 轮左移奇数 1 时，第 $16-i$ 轮左移偶数 2。当对 C_i，D_i 互换时，产生的轮密钥顺序也会互换. 据此，可以得到半弱密钥对如表 4

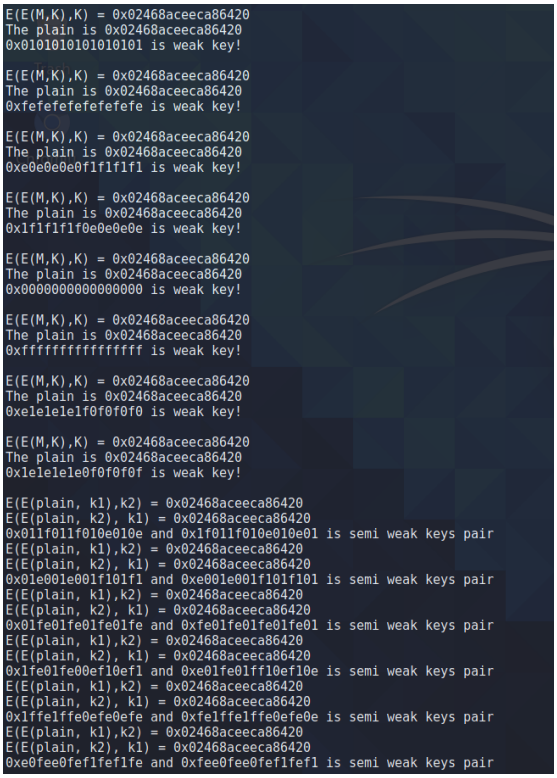
K_1	K_2
0x011F011F010E010E	0x1F011F010E010E01
0x01E001E001F101F1	0xE001E001F101F101
0x01FE01FE01FE01FE	0xFE01FE01FE01FE01
0x1FE01FE00EF10EF1	0xE01FE01FF10EF10E
0x1FFE1FFE0EFE0EFE	0xFE1FFE1FFE0EFE0E
0xE0FEE0FEF1FEF1FE	0xFEE0FEE0FEF1FEF1

表 4: DES 半弱密钥对

明密文角度

1. 弱密钥 K: 由于每轮密钥相等，$E_K(E_K(M)) = M$
2. 半弱密钥 K_0, K_1 :由于产生的轮密钥顺序正好相反,有 $E_{K_2}(E_{K_1}(M)) = E_{K_1}(E_{K_2}(M)) = M$

3.3.2 测试样例及结果截图



3.3.3 总结

由于 DES 密钥扩展时两部分元素间没有扩散，导致会出现不安全的密钥。这也反映了扩散对密码算法的重要性

4 总结

通过本次实验，我更加理解了 Fesitel 结构及 DES 算法的原理，并掌握了加解密流程。同时，通过附加实验了解了三重 DES 并了解了弱密钥与半弱密钥及其出现原因，对混淆与扩散的重要性理解更加提升。

5 思考题

1. 目前对 DES 的攻击包括穷举攻击，差分攻击，线性攻击等。目前超级计算机等对 64bit 密钥 DES 的攻击只需数十分钟。攻击方法中如 Improved Davies 攻击方法计算复杂度为 2^{50}，成功率为 50%

2. 目前的技术力下，计算机的计算能力急速发展。一台 PC 可以破坏 DES 的时间大约是一年，如果多台 PC 并行工作，时间将大大缩短。增大密钥长度虽然能有效防止穷举攻击，但是会增大存储难度，而目前已经有大量更好的 DES 替代方法。