计算机网络安全与管理

RC4 的实现

姓名	易俊泉
班级	软件 92 班
学号	2194411245
电话	18813517223
Email	1302190004@stu.xjtu.edu.cn
日期	2022-6-4

RC4 的实现

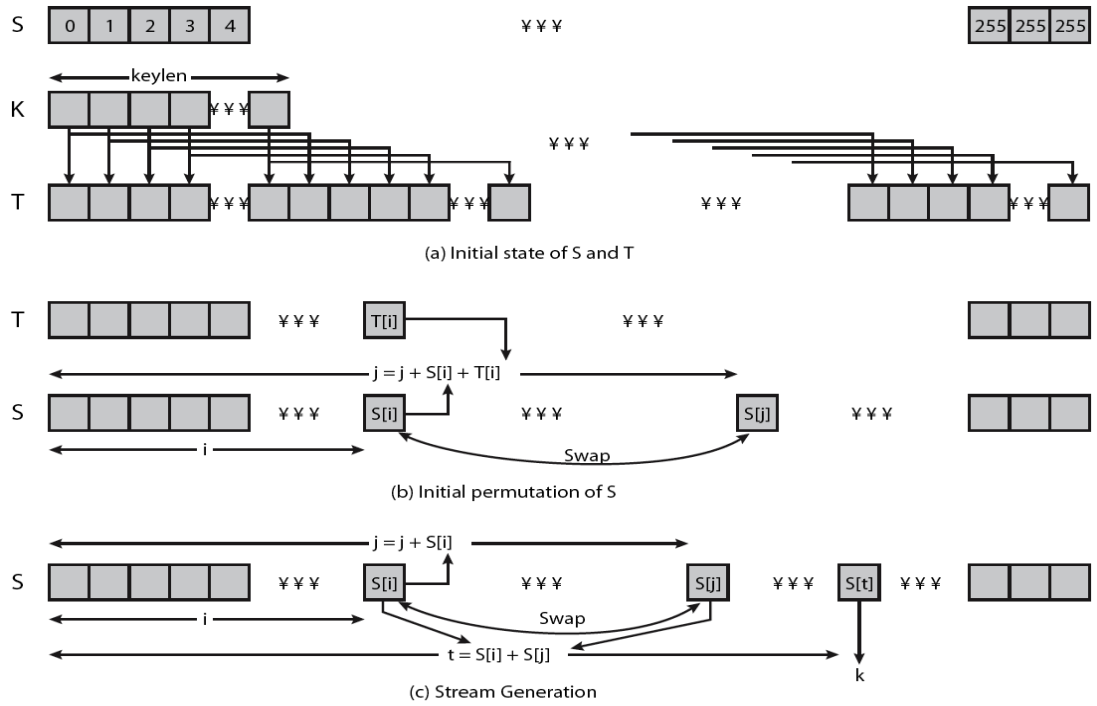
一、 实验要求

① 编码实现 RC4，简单分析代码
② 加解密个人信息，包含姓名学号，等相关信息

二、 实验原理[1]

RC4 是 RonRivest 为 RSA 公司在 1987 年设计的一种流密码。它是一个可变密钥长度、面向字节操作的流密码。该算法以随机置换作为基础。分析显示该密码的周期很可能大于 10^{100}。每输出 1 字节的结果仅需要 8~16 条机器操作指令，软件实现的该密码运行很快。RC4 应用很广，例如，它用于作为 EEE802.l1 无线局域网标准一部分的 WEP(WiredEquivalentPrivacy)协议和新的 WiFi 受保护访问协议(WPA)中。作为可选，它也被用于 SecureShell(SSH)和 Kerberos 中。在当时 RC4 作为 RSA 公司的商业机密并没有公开。直到 1994 年 9 月，RC4 算法才通过 Cypherpunks 匿名邮件列表匿名地公开于 Internet 上。

RC4 算法非常简单，易于描述：用 1~ 256 个字节(8~2048 位)的可变长度密钥初始化一个 256 个字节的状态向量 S，S 的元素记为 S [0]，S [1]，…，S [255]，从始至终置换后的 S 包含从 0~255 所有的 8 位数。对于加密和解密，字节 k 是从 S 的 255 个元素中按一种系统化的方式选出的一个元素生成的。每生成一个 k 值，S 中的元素个体就被重新置换一次。



图片 1 算法逻辑结构

① 初始化 *S*

开始时，S 中元素的值按升序被置为从 0~255，即 S[0]，S[1]，…，S[255]。同时建立一

[1] 该内容均参考《密码编码学与网络安全（第七版）》

个临时向量 T。如果密钥 K 的长度为 256 字节，则将 K 赋给 T。否则若密钥长度为 keylen 个字节(keylen<256),则将 K 的值赋给 T 的前 keylen 个元素，并循环重复用 K 的值赋给 T 剩下的元素，直到 T 的所有元素都被赋值。这些预操作可被概括为如下：

```
/*初始化*/
for i = 0 to 255 do
    S[i] = i
    T[i] = K[i mod keylen];
```

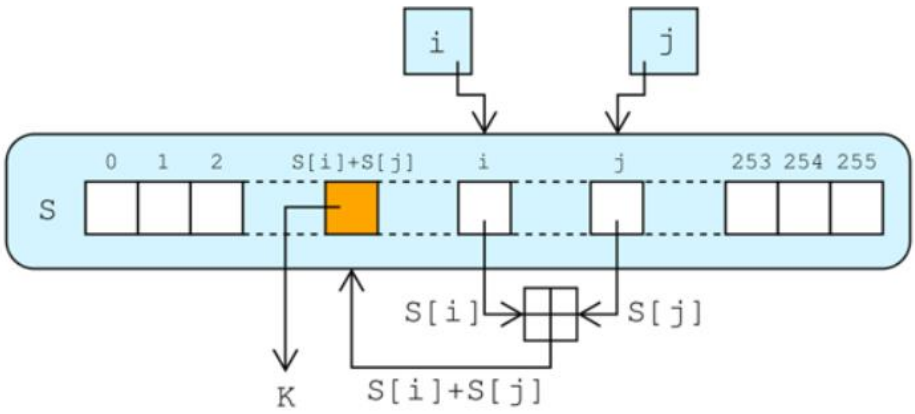
然后用 T 产生 S 的初始置换，从 S[0]到 S[255],对每个 S[i]，根据由 T[i]确定的方案，将 S[i]可置换为 S 中的另一字节。

```
/*S的初始置换*/
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256)
    swap (S[i], S[j])
```

因为对 S 的操作仅是交换，所以唯一的改变就是置换。S 仍然包含所有值为 0~255 的元素。

② 密钥流的产生

向量 S 一旦完成初始化，输入密钥就不再被使用。密钥流的生成过程是，从 S[0]到 S[255]，对每个 S[i]，根据 S 的当前配置，将 S[i]与 S 中的另一字节置换。当 S[255]完成置换后，操作继续重复从 S[0]开始。



图片 2 密钥流的产生

```
/*密钥流的生成*/
i = j = 0
while(true)
    i = (i + 1) (mod 256)
    j = (j + S[i]) (mod 256)
    swap(S[i], S[j])
    t = (S[i] + S[j]) (mod 256)
    k=S[t];
```

加密中，将 k 的值与明文的下一字节异或；解密中，将 k 的值与密文的下一字节异或。

三、 代码说明

这里我使用 java 来实现 RC4。

首先定义了几个变量

```
private int[] S;//S盒
private char[] K;//密钥
private char[] data;//数据
private char[] key_stream;//密钥流
```

① 初始化 *S*

```
/**
* 初始化S
*/
public void initS() {
    S = new int[256];
    int[] T = new int[256];
    //初始化
    for (int i = 0; i < 256; i++) {
        S[i] = i;
        T[i] = K[i % K.length];
    }
    //S的初始置换
    int j = 0;
    for (int i = 0; i < 256; i++) {
        j = (j + S[i] + T[i]) % 256;
        swap(S, i, j);
    }
}
```

② 密钥流的产生

根据 S 盒生成与明文长度相同的秘钥流。

```
/**
* 产生密钥流
*/
public void generateKeyStream() {
    int i = 0, j = 0;
    for (int q = 0; q < data.length; q++) {
        i = (i + 1) % 256;
        j = (j + S[i]) % 256;
        swap(S, i, j);
        int t = (S[i] + S[j]) % 256;
        key_stream[q] = (char) (S[t]);
    }
}
```

③ 加密与解密

由于异或运算的特性，使得加密与解密过程一致。**如果输入的是明文，输出的就是密文；如果输入的是密文，输出的就是明文**。调用过程如下：

```java
/**
* 加密
* @param key       密钥
* @param plaintext 明文
* @return 密文
*/
public String encrypt(String key, String plaintext) {
    return useRC4(key, plaintext);
}

/**
* 解密
* @param key        密钥
* @param ciphertext 密文
* @return 明文
*/
public String decrypt(String key, String ciphertext) {
    return useRC4(key, ciphertext);
}
```

```java
/**
* 使用RC4
* @param key    密钥
* @param string 明文/密文
* @return 密文/明文
*/
private String useRC4(String key, String string) {
    this.data = string.toCharArray();
    this.key_stream = new char[data.length];
    this.K = key.toCharArray();
    //初始化S
    initS();
    //产生密钥流
    generateKeyStream();
    StringBuffer result = new StringBuffer();
    //逐字加密数据
    for (int i = 0; i < data.length; i++) {
        result.append((char) (string.charAt(i) ^ key_stream[i]));
    }
    //输出结果
    return result.toString();
}
```

四、 实验结果

测试数据如下：

```
易俊泉 2194411245 软件92 西安交通大学。我把我的整个灵魂都给你，连同它的怪癖，耍小脾气，忽明忽暗，一千八百种坏毛病。它真讨厌，只有一点好，爱你。--from 王小波
```

测试代码如下

```java
String key = "love";
String plaintext = readFile("src/input.txt");
RC4 test = new RC4();
System.out.println("明文为：");
```

```
System.out.println(plaintext);
System.out.println("使用RC4加密后的密文：");
String ciphertext = test.encrypt(key, plaintext);
System.out.println(ciphertext);
System.out.println("使用RC4解密该密文得到：");
String plaintext_rc4 = test.decrypt(key, ciphertext);
System.out.println(plaintext_rc4);
```

实验结果如下：

```
明文为:
易俊泉 2194411245 软件92 西安交通大学。我把我的整个灵魂都给你，连同它的怪癖，耍小脾气，忽明忽暗，一千八百种坏毛病。它真讨厌，只有一点好，爱你。--from 王小波
使用RC4加密后的密文:
轁京2:V覃嫼丯邎妢孎囯戠扎拡癖敌丹炊鲑達绫佹￥蝾咲家瘸悁瘢；聂峀脐氛ο亻曚忿昲₩叀印汦瘾褃均殃畱ク密辠诣危歴梤仦炁妫＂牪欿ぶ-,§.-3Y珧屯汞
使用RC4解密该密文得到:
易俊泉 2194411245 软件92 西安交通大学。我把我的整个灵魂都给你，连同它的怪癖，耍小脾气，忽明忽暗，一千八百种坏毛病。它真讨厌，只有一点好，爱你。--from 王小波
```

综上，该实验能成功实现简单的 RC4 加密解密。