昵称： shelmean
园龄： 6年5个月
粉丝： 5
关注： 1
+加关注

< 2024年8月 >

日	一	二	三	四	五	六
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

network(7)
Python(4)
C(2)
work(1)
operating system(1)
encryption(1)

随笔分类

C(3)
Linux(4)
加密(1)
链接装载(1)
网络(2)
无线(1)

随笔档案

2022年5月(1)
2022年4月(6)
2021年4月(1)
2021年1月(1)
2018年12月(1)
2018年11月(1)
2018年8月(3)
2018年7月(1)
2018年3月(1)

文章分类

Network(4)
Python(4)
正则表达式(1)

阅读排行榜

1. 字符0、数字0和 '\0'（14959）
2. RC4加密算法(11035)
3. Linux export 命令(2217)
4. 无线网络密码破解初体验(2051)
5. Linux中ip地址结构和ip地址的转换(1991)

评论排行榜

1. 无线网络密码破解初体验(4)
2. 字符0、数字0和 '\0'（1）
3. 《归档与压缩》(1)

最新评论

1. Re:无线网络密码破解初体验

@matrin_kms 信道不一致，应该设置下信道就好了，我最近没用电脑，没法操作...

--shelmean

2. Re:无线网络密码破解初体验

@matrin_kms sudo airodump-ng -c 6 --bssid 24:6F:8C:80:16:60 -w mercury.cap wlan0mon 这个抓取到的...

--shelmean

3. Re:无线网络密码破解初体验

—>>>然后等手机重连，可以看到抓取到了握手报文
请问这个窗口的内容是在哪个终端显示的?

RC4加密算法

什么是RC4？

RC4加密算法是大名鼎鼎的RSA三人组中的头号人物Ron Rivest在1987年设计的密钥长度可变的流加密算法簇，之所以称其为簇，是由于其核心部分的S-box长度可为任意，但一般为256字节。

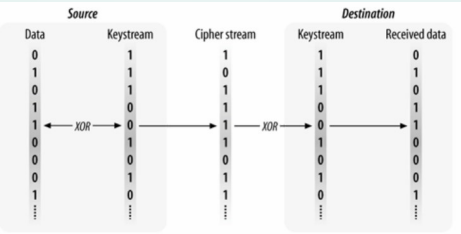
在密码学中，RC4（来自Rivest Cipher 4的缩写）是一种流加密算法，密钥长度可变。它加解密使用相同的密钥，因此也属于对称加密算法。所谓对称加密，就是加密和解密的过程是一样的。RC4是有线等效加密（WEP）中采用的加密算法，也曾经是TLS可采用的算法之一。

RC4已经成为一些常用的协议和标准的一部分，如1997年的WEP和2003/2004年无线卡的WPA; 和1995年的SSL，以及后来1999年的TLS。让它如此广泛分布和使用的主要因素是它不可思议的简单和速度，不管是软件还是硬件，实现起来都十分容易。

基本原理

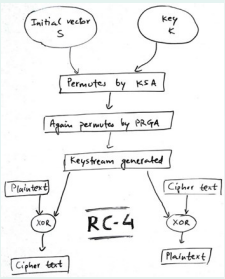
对明文使用同一个密钥异或两次最后得到的是原文

- 加密：原文和Keystream进行异或得到密文
- 解密：密文和Keystream进行异或得到原文



图片来源：《802.11无线网络权威指南》第 5 章 图5-1 串流密码锁的一般运作程序

流程图解



图片来源：https://www.biaodianfu.com/rc4.html

生成秘钥流（KeyStream）

从上图可以看出来，RC4加密原理很简单，只需要一个KeyStream与明文进行异或即可，密钥流的长度和明文的长度是对应的。RC4算法的的主要代码还是在于如何生成秘钥流。

密钥流的生成由两部分组成：

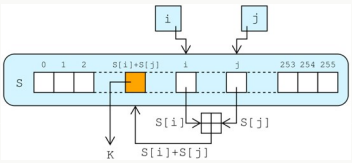
1. KSA（the Key-Scheduling Algorithm）
2. PRGA(the Pseudo-Random Generation Algorithm)

利用Key生成S盒——The key-scheduling algorithm (KSA)

```
/* 得到S-box */
int i = 0;
for (i = 0; i < 256; i++) {
    S[i] = i;
    T[i] = puc_key[i % key_length];
}

for (i = 0; i < 256; i++) {
    j = (j + S[i] + T[i]) % 256;
    swap_uchar(&S[i], &S[j]); //交换S[i]和S[j]
}
```

利用S盒生成密钥流——The pseudo-random generation algorithm(PRGA)



图片来源：https://www.biaodianfu.com/rc4.html

```
/* 生成密钥流 Keystream */
int i = 0;
int j = 0;
int t = 0;
unsigned long k = 0;

for (k = 0; k < ul_data_length; k++) {
    i = (i + 1) % 256;
    j = (j + puc_sbox[i]) % 256;
    swap_uchar(&puc_sbox[i], &puc_sbox[j]);
    t = (puc_sbox[i] + puc_sbox[j]) % 256;
    puc_key_stream[k] = puc_sbox[t];
}
```

代码实现

```
#include<stdio.h>
#include<string.h>

#define SBOX_LEN 256

#define rc4_encrypt rc4_crypt
#define rc4_decrypt rc4_crypt

static inline void swap_uchar(unsigned char *puc_x, unsigned char *puc_y)
{
    *puc_x = *puc_x ^ *puc_y;
    *puc_y = *puc_x ^ *puc_y;
    *puc_x = *puc_x ^ *puc_y;
}

void hexdump(unsigned char *puc_data, int length)
{
    int i = 0;

    for (i = 0; i < length; i++) {
        printf("%02X", puc_data[i]);
        if (i && (i + 1) % 16 == 0) {
            putchar('\n');
        }
    }
    printf("\n");
}

/**
 * 利用Key生成S盒
 * the Key-Scheduling Algorithm
 */
static void rc4_ksa(unsigned char *puc_sbox, unsigned char *puc_key, int key_length)
{
    int i = 0;
    int j = 0;
    char tmp[SBOX_LEN] = {0};
```

--matrin_kms

4. Re:无线网络密码破解初体验

大佬, 您好! 我也想学习一下, 试试破解自己的WiFi, 但是我现在碰到这个问题: 23:55:31 wlan0mon is on channel 8, but the AP uses channe...

--matrin_kms

5. Re:IP地址分类

好久没更新博客了，坐等更新

--check_check

```c
    for (i = 0; i < SBOX_LEN; i++) {
        puc_sbox[i] = i;
        tmp[i] = puc_key[i % key_length];
    }

    for (i = 0; i < SBOX_LEN; i++) {
        j = (j + puc_sbox[i] + tmp[i]) % SBOX_LEN;
        swap_uchar(&puc_sbox[i], &puc_sbox[j]); //交换puc_sbox[i]和puc_sbox[j]
    }
}

/**
 * 利用S盒生成密钥流
 * The pseudo-random generation algorithm(PRGA)
 */
static void rc4_prga(unsigned char *puc_sbox, unsigned char *puc_key_stream, unsigned long ul_data_length)
{
    int i = 0;
    int j = 0;
    int t = 0;
    unsigned long k = 0;

    for (k = 0; k < ul_data_length; k++) {
        i = (i + 1) % SBOX_LEN;
        j = (j + puc_sbox[i]) % SBOX_LEN;
        swap_uchar(&puc_sbox[i], &puc_sbox[j]);
        t = (puc_sbox[i] + puc_sbox[j]) % SBOX_LEN;
        /* 为了更清晰看到rc4算法流程 此处保存keystream. 不直接进行XOR运算 */
        puc_key_stream[k] = puc_sbox[t];
    }
}

/* 加解密 */
void rc4_crypt(unsigned char *puc_data, unsigned char *puc_key_stream, unsigned long ul_data_length)
{
    unsigned long i = 0;

    /* 把PRGA算法放在加解密函数中可以不需要保存keystream */
    for (i = 0; i < ul_data_length; i++) {
        puc_data[i] ^= puc_key_stream[i];
    }
}

int main(int argc, char *argv[])
{
    unsigned char sbox[SBOX_LEN] = {0};
    char key[SBOX_LEN] = {"abcdefghijklmnopqrstuvwxyz"}; //秘钥内容随便定义
    char data[512] = "1sRJ@.0 1vfvr#9527";
    unsigned char puc_keystream[512] = {0};
    unsigned long ul_data_length = strlen(data);

    printf("key=%s, length=%d\n\n", key, strlen(key));
    printf("Raw data string:%s\n", data);
    printf("Raw data hex:\n");
    hexdump(data, ul_data_length);

    /* 生成S-box */
    rc4_ksa(sbox, (unsigned char *)key, strlen(key));

    /* 生成keystream并保存,S-box也会被更改 */
    rc4_prga(sbox, puc_keystream, ul_data_length);

    printf("S-box final status:\n");
    hexdump(sbox, sizeof(sbox));

    printf("key stream:\n");
    hexdump(puc_keystream, ul_data_length);

    /* 加密 */
    rc4_encrypt((unsigned char*)data, puc_keystream, ul_data_length);

    printf("cipher hexdump:\n");
    hexdump(data, ul_data_length);

    /* 解密 */
    rc4_decrypt((unsigned char*)data, puc_keystream, ul_data_length);

    printf("decypt data:%s\n", data);

    return 0;
}
```

运行示例:

```
┌──(shelmean㉿ubuntu)-[~/rc4]
└─$ ./rc4
key=abcdefghijklmnopqrstuvwxyz, length=26

Raw data string:1sRJ@.0 1vfvr#9527
Raw data hex:
6C73524A402E30206C76667672233935
3237
S-box final status:
0F6FB31DDB7F1C9C91B760E83B2FF7B3
3F49485A942603DEA95F463386711A55
DC6DF03D05975EAA41D94E2DAF135059
09ED42C6984706 7BCCB75BD7747D0C4D
2B0E844F9A5168535273113 54C77219E
FD179F02297C18A06A7572BF2CC5A108
E1FBE7A819C0DFA28DFF8FD262D4BAAD
6E3CA34476340804BCA580E5F4B42A20
2232FA153090EECB9B56A600EFB2A7A4
2EF5C1E3AEF93882B0B6B87E376C668E
541F7925144585C8C4963AC9B5B1248A
E4F11E16B9E289E8CD3EEC23D19D694A
BDC2B8FED0FCABCA7057F30DEAE901C7
ACE067B827B5929366139F25D120A8C
2831CED3FBD55CCF9995436436C3D681
58DA65DD1B781063D807F66B404B8B7A

key stream:
2393B785D3C35D4824B588D0C0159423
1EEA
cipher hexdump:
4FE0E5CF93ED6D6848F3EEA6B236AD16
2CDD
decypt data:1sRJ@.0 1vfvr#9527
```

参考

> 《802.11无线网络权威指南》 第5章
>
> 百度百科 RC4
>
> https://www.biaodianfu.com/rc4.html

分类: 加密

标签: encryption

好文要顶 关注我 收藏该文 微信分享

shelmean
粉丝 - 5 关注 - 1
+加关注

1 推荐 0 反对

« 上一篇： Linux中ip地址结构和ip地址的转换
» 下一篇： 无线网络密码破解初体验

posted @ 2021-01-15 11:47 shelmean 阅读(11035) 评论(0) 编辑 收藏 举报