C语言实现RSA算法 _

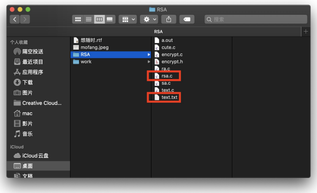
2020-07-05 18:21 7880 1 8 T 9862 19:43 ~ 32:52

算法

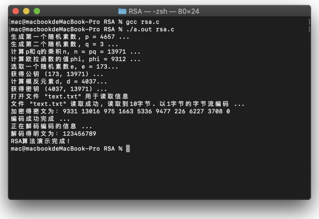
```c
c，m = c^d Mod n
 */
int decode(int c, int d, int n) {
    return modpow(c, d, n);
}

/**
 * Encode the message of given length, using the public key (exponent, modulus)
 * The resulting array will be of size len/bytes, each index being the encryption
 * of "bytes" consecutive characters, given by m = (m1 + m2*128 + m3*128^2 + ..),
 * encoded = m^exponent mod modulus
 * 使用公钥(指数、模数)对给定长度的消息进行编码)
 得到的数组将是大小为len/字节，每个索引是由m=(m1m2*128m3*128^2.)给出的“字节”连续字符的加密，编码=m^指数mod模量
 */
int* encodeMessage(int len, int bytes, char* message, int exponent, int modulus) {
    int *encoded = (int *)malloc((len/bytes) * sizeof(int));
    int x, i, j;
    for(i = 0; i < len; i += bytes) {
        x = 0;
        for(j = 0; j < bytes; j++) x += message[i + j] * (1 << (7 * j));
        encoded[i/bytes] = encode(x, exponent, modulus);
#ifndef MEASURE
        printf("%d ", encoded[i/bytes]);
#endif
    }
    return encoded;
}

/**
 * Decode the cryptogram of given length, using the private key (exponent, modulus)
 * Each encrypted packet should represent "bytes" characters as per encodeMessage.
 * The returned message will be of size len * bytes.
 * 使用私钥(指数、模数)解码给定长度的密码)
 每个加密的数据包应该按照编码消息表示“字节”字符。
 返回的消息大小为len*字节。
 */
int* decodeMessage(int len, int bytes, int* cryptogram, int exponent, int modulus) {
    int *decoded = (int *)malloc(len * bytes * sizeof(int));
    int x, i, j;
    for(i = 0; i < len; i++) {
        x = decode(cryptogram[i], exponent, modulus);
        for(j = 0; j < bytes; j++) {
            decoded[i*bytes + j] = (x >> (7 * j)) % 128;
#ifndef MEASURE
            if(decoded[i*bytes + j] != '\0') printf("%c", decoded[i*bytes + j]);
#endif
        }
    }
    return decoded;
}

/**
 * Main method to demostrate the system. Sets up primes p, q, and proceeds to encode and
 * decode the message given in "text.txt"
 * 系统降级的主要方法。 设置素数p，q，并开始编码和
 *解码“text.txt”中给出的消息；
 */
int main(void) {
    int p, q, n, phi, e, d, bytes, len;
    int *encoded, *decoded;
    char *buffer;
    FILE *f;
    srand(time(NULL));
    while(1) {
        p = randPrime(SINGLE_MAX);
        printf("生成第一个随机素数, p = %d ... ", p);
        getchar();

        q = randPrime(SINGLE_MAX);
        printf("生成第二个随机素数, q = %d ... ", q);
        getchar();

        n = p * q;
        printf("计算p和q的乘积n, n = pq = %d ... ", n);
        if(n < 128) {
            printf("Modulus is less than 128, cannot encode single bytes. Trying again ... ");
            getchar();
        }
        else break;
    }
    if(n >> 21) bytes = 3;
    else if(n >> 14) bytes = 2;
    else bytes = 1;
    getchar();

    phi = (p - 1) * (q - 1);
    printf("计算欧拉函数的值phi, phi = %d ... ", phi);
    getchar();

    e = randExponent(phi, EXPONENT_MAX);
    printf("选取一个随机素数e, e = %d...\n获得公钥 (%d, %d) ... ", e, e, n);
    getchar();

    d = inverse(e, phi);
    printf("计算模反元素d, d = %d...\n获得密钥 (%d, %d) ... ", d, d, n);
    getchar();

    printf("打开文件 \"text.txt\" 用于读取信息\n");
    f = fopen("text.txt", "r");
    if(f == NULL) {
        printf("Failed to open file \"text.txt\". Does it exist?\n");
        return EXIT_FAILURE;
    }
    len = readFile(f, &buffer, bytes); /* len will be a multiple of bytes, to send whole chunks伦将是多个字节，以发送整个块 */
    fclose(f);

    printf("文件 \"text.txt\" 读取成功, 读取到%d字节. 以%d字节的字节流编码 ... ", len, bytes);
    getchar();
    printf("加密得密文为:");
    encoded = encodeMessage(len, bytes, buffer, e, n);
    printf("\n编码成功完成 ... ");
    getchar();


    printf("正在解码编码的信息 ... ");
    getchar();
    printf("解码得明文为:");
    decoded = decodeMessage(len/bytes, bytes, encoded, d, n);


    printf("\nRSA算法演示完成!\n");

    free(encoded);
    free(decoded);
    free(buffer);
    return EXIT_SUCCESS;
}
```

代码没有半点问题，跑就是了；

需要注意的是，运行代码需要新建一个.txt文档，里面存放你的明文，目前我试过支持数字和英文，中文不支持，其他还没试过。把这个.txt文档和代码放在一个路径里，注意代码里的文件名和你的文件名，现在是text.txt；



这是运行结果，可以参考下；



```
mac@macbookdeMacBook-Pro RSA % gcc rsa.c
mac@macbookdeMacBook-Pro RSA % ./a.out rsa.c
生成第一个随机素数，p = 4657 ...
生成第二个随机素数，q = 3 ...
计算p和q的乘积n, n = pq = 13971 ...
计算欧拉函数的值phi, phi = 9312 ...
选取一个随机素数e, e = 173...
获得公钥 (173, 13971) ...
计算模反元素d, d = 4037...
获得密钥 (4037, 13971) ...
打开文件 "text.txt" 用于读取信息
文件 "text.txt" 读取成功，读取到10字节. 以1字节的字节流编码 ...
加密得密文为: 9331 13016 975 1663 5336 9477 226 6227 3708 0
编码成功完成 ...
正在解码编码的信息 ...
解码得明文为: 123456789
RSA算法演示完成！
mac@macbookdeMacBook-Pro RSA % 
```

以上为所有内容，🏄冲浪🏄‍♀码字不容易！！！

~~您的赞就是最好的肯定🙆‍♂~~

```
//还是那句话，有什么问题，欢迎指正！！！
```

__EOF__

本文作者： 栁旍
本文链接： https://www.cnblogs.com/myth67/p/13247074.html
关于博主： 评论和私信会在第一时间回复。或者直接私信我。
版权声明： 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！
声援博主： 如果您觉得文章对您有帮助，可以点击文章右下角【**推荐**】一下。

你要灿若星海

分类: 算法

推荐该文 关注博主 收藏本文 分享微信

MoreKing
粉丝 - 3 关注 - 4
+加关注

8 0

« 上一篇： 模运算
» 下一篇： Mac下使用C语言生成和使用动态链接库

posted @ 2020-07-05 18:21 MoreKing 阅读(7880) 评论(1) 编辑 收藏 举报

登录后才能查看或发表评论 · 立即 登录 或者 逛逛 博客园首页

编辑推荐：
· 你为什么不应该过度关注go语言的逃逸分析
· 在C#中基于Semantic Kernel的检索增强生成（RAG）实践
· 数据库系列：主从延时优化
· 一次彻底讲清如何处理mysql 的死锁问题
· 我被 .NET8 JIT 的一个BUG反复折磨了半年之久

阅读排行：
· 404的众包平台 · 也许是园子商业化的未来
· C#/.NET/.NET Core技术前沿周刊 | 第 10 期（2024年10.14-10.20）
· Awesome Tools · 程序员常用高效实用工具、软件资源精选 · 办公效率提升利器！
· count(*)、count(1)哪个更快？面试必问：通宵整理的十道经典MySQL必问面试题
· 推荐一款专为Nginx设计的图形化管理工具: Nginx UI！

[你的踏板车要滑向哪里 ❤ 如今这个广场是你的坟墓]

This blog has running : 1390 d 9 h 15 m 24 s ღゝ◡╹)ノ♡