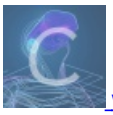
RSA加密

原创

一个小南瓜 于 2020-07-27 10:16:00 发布 258 收藏

分类专栏： web安全 CTF 文章标签： 密码学 算法 python



RSA加密

0x01、RSA算法的起源和简介

出自百度百科

1、起源：

RSA是1977年由罗纳德·李维斯特（Ron Rivest）、阿迪·萨莫尔（Adi Shamir）和伦纳德·阿德曼（Leonard Adleman）一起提出的。RSA就是他们三人姓氏开头字母拼在一起组成的。

2、简介：

RSA公开密钥密码体制是一种使用不同的加密密钥与解密密钥，“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。RSA是被研究得最广泛的公钥算法，从提出到现在已近三十年，经历了各种攻击的考验，逐渐为人们接受，普遍认为是目前最优秀的公钥方案之一。

0x02、RSA算法的基本原理

RSA是非对称加密

数学锁：f(m)=c 信息m通过加密函数f()得到密文c很简单，但是通过密文c得到信息m却很难，但是给与一定的提示信息（私钥）后这种逆运算就变得简单

取模运算：

```
a mod n ≡ b    a与b对模n同余
3 mod 2 ≡ 1    3与1对模2同余
```

信息传递过程：

$m^e \bmod N \equiv c$ 加密过程（简单），加密密钥（公钥）:(e,N)

$c^d \bmod N \equiv m$ 解密过程（难），解密密钥（私钥）：(d,N)

0x03、RSA算法描述

(实际应用中选取的素数都很大来保证加密的可靠性)

1、随机选择两个不相等的质数p和q，例如p=61,q=53

2、计算p和q的乘积N

N=61*53=3233

N的二进制的长度就是密钥的长度，3233的二进制一共有12位，所以这个密钥的长度就是12位。（实际应用中密钥的长度一般为1024位或2048位）

3、计算N的欧拉函数φ(N)

φ(N)=(p-1)(q-1)

φ(3233)=3120

4、随机选择一个整数e,也就是公钥中用来加密的数字

条件：1<e<φ(N),且e与φ(N)互质

此处，随机选择17（实际应用中，常常选择65537）

5、计算e对于φ(N)的模反元素d。也就是密钥当中用来解密的数字。

推导d和e的关系，欧拉定理

由：

$m^e \bmod N \equiv c$

$c^d \bmod N \equiv m$

推导出：

d=(K*φ(N)+1)/e

k的取值要保证d可以算出整数。

e和d的关系也可以写作：

ed≡1 mod φ(N) 也可以写作 ed mod φ(N) = 1

取d=2753 ,17*2753 mode 3120=1

6、将e和N封装成公钥，d和N封装为私钥

公钥为：(17,3233)

私钥为：(2753,3233)
这里要解释一下，≡是数论中表示同余的符号。公式中，≡符号的左边必须和符号右边同余，也就是两边模运算结果相同。显而易见，不管取什么值，符号右边1 mod φ(N)的结果都等于1；符号的左边d与e的乘积做模运算后的结果也必须等于1。这就需要计算出d的值，让这个同余等式能够成立。

RSA算法的保密性在于如何正确将N分解成两个整数素数,而整数分解在数学上是无解的问题

0x04、RSA算法在CTF中的应用

1、攻防世界easy_RSA

[https://adworld.xctf.org.cn/task/answer?type=crypto&number=5&grade=0&id=5114&page=1]:

题目：

```
在一次RSA密钥对生成中，假设p=473398607161，q=4511491，e=17
求解出d
```

解题脚本1（输出的d即为flag）：

```
import math
def getEuler(p1,q1):
    return (p1-1)*(q1-1)
def getDkey(e,Eulervalue):
    k=1
    while True:
        if ((Eulervalue*k)+1)%e==0:
            (d,m)=divmod(Eulervalue*k+1,e)
            return d
        k += 1
def Mingwen(c,d,n):
    return pow(c,d,n)
if __name__=='__main__':
    p=473398607161
    q=4511491
    d=getDkey(17,getEuler(p,q))
    print('私钥为:%d'%d)
```

解题脚本2：

```
import gmpy2
p =473398607161
q =4511491
e =17
phin =(p -1)*(q -1)
# print (gmpy2.invert(e, phin))
d =(gmpy2.invert(e, phin))
# d = mpz(125631357777427553)
d =125631357777427553
cipher =""
for char in str(d):
    cipher +=chr(ord('a')+int(char)-1)
    print(cipher)
```

参考文档：

[https://www.jianshu.com/p/ff2b538a77e2]

[https://adworld.xctf.org.cn/task/writeup?type=crypto&id=5114&number=5&grade=0&page=1]