CTF之Crypto学习笔记(一)

原创

3tefanie丶zhou 于 2021-10-14 20:42:06 发布 2121 收藏

分类专栏： CTF题型总结 文章标签： 安全

本文链接：https://blog.csdn.net/luochen2436/article/details/120771081

文章目录

原理

加密

$$密文 = 明文^{E} mod N$$

$$公钥 = (E, N)$$

随便找出两个 整数 q 和 p （q，p互素，即：公因数只有1）

求出n = q * p

φ(n)= （p-1）*（q-1） 欧拉公式

公钥 e： 随机取，要求：e 和 φ(n) 互素（公因数只有 1）； 1< e < φ(n)）；

私钥 d： ed ≡ 1 （mod φ(n) ） （ed 除以 φ(n) 的 余数 为 1 ）

RSA加密其实是对明文的E次方后除以N后求余数的过程。只要知道E和N任何人都可以进行RSA加密了，所以说E、N是RSA加密的密钥，也就是说E和N的组合就是公钥，用(E,N)来表示公钥。

E：加密指数

解密

$$明文 = 密文^{D} mod N$$

$$私钥 = (D, N)$$

前面说到，RSA的加密过程是对明文的E次方后除以N后求余数的过程

那么，RSA解密过程则是密文进行D次方后除以N的余数，即明文
我们知道D和N就能进行解密密文了，所以D和N的组合就是私钥
D：解密指数

RSA的加密方式和解密方式是一样的，变化的是加密指数与解密指数的不同

加密是求 E次方的mod N
解密是求 D次方的mod N

公钥	(E，N)
私钥	(D，N)
密钥对	(E，D，N)
加密	$密文 = 明文^{E} mod N$
解密	$明文 = 密文^{D} mod N$

解密过程详细解读

4. 生成密钥对

既然公钥是（E，N），私钥是（D，N）所以密钥对即为（E，D，N）但密钥对是怎样生成的？步骤如下：

1. 求N
2. 求L（L为中间过程的中间数）
3. 求E
4. 求D

4.1 求N

准备两个质数p，q。这两个数不能太小，太小则会容易破解，将p乘以q就是N

$$N = p * q$$

4.2 求L

L 是 p - 1 和 q - 1的最小公倍数，可用如下表达式表示

$$L = lcm\,(\,p\,-\,1\,,\,q\,-\,1\,)$$

4.3 求E

E必须满足两个条件：E是一个比1大比L小的数，E和L的最大公约数为1
用gcd(X,Y)来表示X，Y的最大公约数则E条件如下：

1 < E < L

gcd（E，L）=1

之所以需要E和L的最大公约数为1是为了保证一定存在解密时需要使用的数D。现在我们已经求出了E和N也就是说我们已经生成了密钥对中的公钥了。

4.4 求D

数D是由数E计算出来的。D、E和L之间必须满足以下关系：

1 < D < L

E * D mod L = 1

只要D满足上述2个条件，则通过E和N进行加密的密文就可以用D和N进行解密。
简单地说条件2是为了保证密文解密后的数据就是明文。
现在私钥自然也已经生成了，密钥对也就自然生成了。
小结下：

求N	N= p * q；p，q为质数
求L	L=lcm（p - 1，q - 1）；L为p - 1、q - 1的最小公倍数

求E	1 < E < L，gcd（E，L）=1；E，L最大公约数为1（E和L互质）
求D	1 < D < L，E * D mod L = 1

5 实践下吧

我们用具体的数字来实践下RSA的密钥对对生成，及其加解密对全过程。为方便我们使用较小数字来模拟。

5.1 求N

我们准备两个很小对质数，
p = 17
q = 19
N = p * q = 323

5.2 求L

L = lcm（p - 1， q - 1）= lcm(16，18) = 144
144为16和18对最小公倍数

5.3 求E

求E必须要满足2个条件：1 < E < L，gcd（E，L）=1
即1 < E < 144，gcd（E，144） = 1
E和144互为质数，5显然满足上述2个条件
故E = 5

此时**公钥=(E，N) = （5，323）**

5.4 求D

求D也必须满足2个条件：1 < D < L，E * D mod L = 1
即1 < D < 144，5 * D mod 144 = 1
显然当D = 29 时满足上述两个条件
1 < 29 < 144
5 * 29 mod 144 = 145 mod 144 = 1
此时**私钥= （D，N）= （29，323）**

5.5 加密

准备的明文必须时小于N的数，因为加密或者解密都要mod N其结果必须小于N
假设明文 = 123
则 密文 = 明文$^{E} mod N = 123^{5} mod 323 = 225$

5.6 解密

明文 = 密文$^{D} mod N = 225^{29} mod 323 = 123$
解密后的明文为123。

好了至此RSA的算法原理已经讲解完毕，是不是很简单?

做一道曾经望而生畏的简单题

题目：

```
p = 26224880018227704065019205543990658047 9
q = 26285499423932282854792559548751991555 1
e = 65533
n = p*q
c = 27565231154623519221597938803435789010285480123476977081867877272451638645710
```

根据题目给出的信息，已知加密指数e,密文c，模n,以及p和q
由此编写解密脚本

```
import gmpy2
from Crypto.Util.number import *
from binascii import a2b_hex,b2a_hex



p = 26224880018227704065019205543990658047 9
q = 26285499423932282854792559548751991555 1

e = 65533
n = p*q

c = 27565231154623519221597938803435789010285480123476977081867877272451638645710
phi = (p-1)*(q-1)     #求ϕ(n)，ϕ(n)=(p-1)(q-1)
d = gmpy2.invert(e,phi)    #求e对于模n的逆元，即解密指数d
m = pow(c,d,n)      #m=c^d mod n，m为10进制格式
print(long_to_bytes(m))    #m的字符串格式
```

flag{B4by_Rs4}

参考链接：
https://blog.csdn.net/vhkjhwbs/article/details/101160822

https://blog.csdn.net/vanarrow/article/details/107846987?utm_medium=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~default-1.no_search_link&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~default-1.no_search_link

【请不要把陌生人的些许善意，视为珍惜的隗宝，却把身边亲近人全部付出，当作天经地义的事情，对其视而不见】