[原创]CTF 2019Q2 readyu crackme 设计思路 精

readyu 8

2019-5-4 19:34

本题算法主要涉及公钥密码学的基础知识。 Win32程序， 无壳， 无Anti。 暂定取名 “一石二鸟”。

(1) 序列号
本题唯一序列号SN为：
KCTFREADYKXXXX154839617191505636852651380494876561909439231580657846179615950521527828825 4

(2) 方程
算法基于二次剩余与离散对数， 建立了2个方程 (1) (2) 见下文。并且模同一个素数。所以暂定取名 “一石二鸟”。
P=2^255-19 是一个素数, base16 或者base10如下：
P(hex)=7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFED
P(dec)=57896044618658097711785492504343953926634992332820282019728792003956564819949

序列号格式为两段字符， + 表示字符串合并：
SN = “s1” + “XXXX” + “s2”
s1是大写字母串， s2是十进制数字串。 数值上， 1 < s1，s2 < P

其中, s1,s2 是未知量， d是s1变换而来的未知量。
s1逆序变换，加上最小位调整， 然后base25解码得到d. (Y,G)内置于程序之中。

64*(s2-s1)^4 + (s2-s1)^2 + 3 = s1 mod P ... (1)
G^d = Y mod P ... (2)
G,Y 是已知数, 范围 [1, P-1]:
Y = 100
G = 923019785897501829962985797741152795455089947830751080921052096734695860003 9

(3) 解法

解法：首先解方程2， 得到d ， 转换为s1 ； 然后再解方程1 得到s2。

(3.1) 解方程(2)
方程(2)解释如下：
d = base25.reverse(s1) ....(3)

随机验证任何一对，结果都是一样的。
(Y,G) =
"100,923019785897501829962985797741152795455089947830751080921052096734695860003 9",
"101,504142217673520837656134985246745908443338237202556564324905578667724886003 4",
"102,383776841641129146692018316507568135510722233145922882179299471582835322702 68",
"103,134361955335197786716481208657431780104316970224006703849095150019704006450 91",

比如 Y=100,
G=923019785897501829962985797741152795455089947830751080921052096734695860003 9

G^d = Y mod P ，P是256bit的大素数。

GDLOG
Implementation of the GNFS for discrete logarithm problem in GF(p)
https://sourceforge.net/projects/gdlog/

但本题，程序里给出了条件， d 转化为base25字符时，最长为10个字节。
所以数量级极大地缩小， 只有47bits。

也就是d < limit, limit = 25^10 = 95367431640625 。
因此，用 pollard kangaroo 算法求解 d, 范围为 band = (1, 95367431640625) 只有47 bits。
kangaroo算法是解决区间band = (a, b)上离散对数问题很有效的方法,在平均意义下需要进行2 * √|a-b|次群操作。
期望步数 2 * (limit^0.5)= 2* 9765625, 也就是大概2000万次。经测试， 大概30-60秒之间。

我们得到:
d(dec) = 79821823136933
d(hex)= 4898F769D4A5
base25table=ABCDEFGHKJILMNOPQRSTUVWYZ
是26个大写字母表，(扣除X, 并且K，I互换位置)
得到 d.base25 = UYDAERFTCK

逆序后得到： s = d.base25.reverse = KCTFREADYU
为了防止被猜测到s1， 低位做了修正，修正值 diff = 'U' - 'K' = 10：
注册码里的采用KCTFREADYK，
s1 = KCTFREADYK + diff = KCTFREADYU
s1(hex) = 4B435446524541445955

(3.2) 解方程(1)

(3.2.1)
首先，根据一般的二次剩余方程， 可以配方以后求解：
a*x^2 + bx + c = 0 mod p
两边乘以4a配方:
4a^2*x^2 + 4a*bx + 4ac = 0 mod p

方程变为
(2a*x + b)^2 = b^2 - 4ac
令delta = b^2 - 4ac, 就可以归结为求 sqroot(delta) mod p：
x1 = (-b - sqroot(delta))/(2a) mod p
x2 = (-b + sqroot(delta))/(2a) mod p
注意，这里的除法是模P的逆。

(3.2.2)
方程(1)解释如下：
64*(s2 - s1)^4 + (s2-s1)^2 + 3 - s1 = 0 mod P

F(x)是一个4次多项式：F(x) = a4*x^4 + a2*x^2 + a0
这里取x = s2 - s1, a4 = 64, a2 = 1, a0 = 3

F(x) 可以简化为一个二次剩余方程，
64*(s2 - s1)^4 + (s2-s1)^2 + 3 = s1 mod P ... (4)

64*r^2 + r + 3 = s1 mod p ,
=> 64*r^2 + r = (s1 - 3) mod p

两边乘以4a, 加上1:
4*64*64*r^2 + 256*r + 1 = 256*(s1-3) + 1 mod p

s1 = 4B435446524541445955, s1-3 = 4B435446524541445952
乘以256就是左移一个字节：

(3.2.3)
方程简化为：
(128r + 1)^2 = 4B435446524541445952 01 mod p ...(5)
r = (sqroot(4B435446524541445952 01, P) - 1)/128

第一步，方程(5)求解首先求得两个解r1,r2。
Y = 4B435446524541445952 01
sqroot(Y, P)， RDLP 求解。

Two sqroots of Y (mod P), in HEX BASE.
G1-258B783A22015B08A6C64FB55644BAACCDA201473D4B6786821056707C680B58
G2-5A7487C5DDFEA4F75939B04AA9BB4553325DFEB8C2B498797DEFA98F8397F495

再求得：
r1=2D4B16F0744402B6114D8C9F6AAC8975599B44028E7A96CF0D0420ACE0F8D010
r2=1CB4E90F8BBBFD49EEB273609553768AA664BBFD71856930F2FBDF531F072FE5

然后r1,r2分别求解二次剩余，各有两个解，所以一共有4个解。

第二步， 解出 x^2 = r mod p , 可用RDLP求解。
root(r1)=
x2-3CCA260F45B79993C67F35F7A716B28BBA591BA35593C8DEB9B959C2CE43AE21
x3-4335D9F0BA48666C3980CA0858E94D7445A6E45CAA6C37214646A63D31BC51CC

root(r2)=
x4-7C93A389F44E31BC25D90165624292389B47C2C27F60286FF627A6FC3DB84BC4
x1-036C5C760BB1CE43DA26FE9A9DBD6DC764B83D3D809FD79009D85903C247B429

这4个解记作x1,x2,x3,x4, 恰好每一个都分布在(1, P)的4个区间之一。
(0-1/4), (1/4-1/2), (1/2 - 3/4), (3/4, 1)

然后： s2 = s1 + X , 也有4个s2， 并转为10进制表示。
最后合并：

SN = "s2" + "XXXX" + "s1"

4个解为：
SN_X1 = KCTFREADYKXXXX154839617191505636852651380494876561909439231580657846179615950521527828825 4
SN_X2= KCTFREADYKXXXX27495936700183671733408543181646240981077460232127048216208422649817010276214
SN_X3= KCTFREADYKXXXX30400107918474425978376949322697712945557532100693234514359350753673330338593
SN_X4= KCTFREADYKXXXX56347648446743041343258978699395188307540600017013704268771613898275062326553

题目取最小的解：
x < P/4，SN_X1 为有效答案。

[公告]看雪.纽盾 KCTF 2019晋级赛Q3攻击方规则，9月10日开赛，华为P30 Pro、iPad、kindle等你来拿！

最后于 2019-6-24 18:07 被kanxue编辑 ，原因:

上传的附件：

keygenme_2019q2_readyu.rar（38.53kb，11次下载）

最新回复 (5)

readyu 8 2019-5-4 19:36 2楼 0

解压密码为 readyu2019q2

附件是 基于miracl运算库的 Pollard's kangaroos 方法求 离散对数源代码以及编译好可运行的demo，验证输出如下:
(比赛结束以后公布)

算法描述见 wiki :
https://en.wikipedia.org/wiki/Pollard%27s_kangaroo_algorithm

LIMIT64 = 95367431640625 , LEAPS= 9765626
solve discrete logarithm problem - using Pollard's kangaroos
find d in: y = g^d mod n, given(y, g , n), if: d < 64 bits

y= 100
g= 923019785897501829962985797741152795455089947830751080921052096734695860003 9
n= 57896044618658097711785492504343953926634992332820282019728792003956564819949

setting trap ....

trap set! jumps = 9765626
Time cost 18 seconds 362ms

speed 531 K/s

Gotcha! Time cost 39 seconds 266 ms

jumps = 21326452, speed = 543 K/s

Discrete log: d =
79821823136933

最后于 2019-5-5 09:57 被readyu编辑，原因:

上传的附件:

dlp_kangaroo_src_2019q2.rar（298.41kb，16次下载）

netwind 13 2019-5-5 12:30 3楼 0

初步检查正常

readyu 8 2019-6-24 10:49 4楼 0

补充说明:
G^d = Y mod P　　... (2)
d=4898F769D4A5
这个d碰巧有个特点: d = ED970F * 4E390B
因此题目里加上一个条件判断：d不含0xFFFF 内的小素数因子。

最新回复 (5)

看场雪 3 2019-6-25 10:58 5 楼 0

数学功底真好

感觉这才是好出题方式，值得学习

勇士小蓝

内容

回帖 表情 高级回复

返回