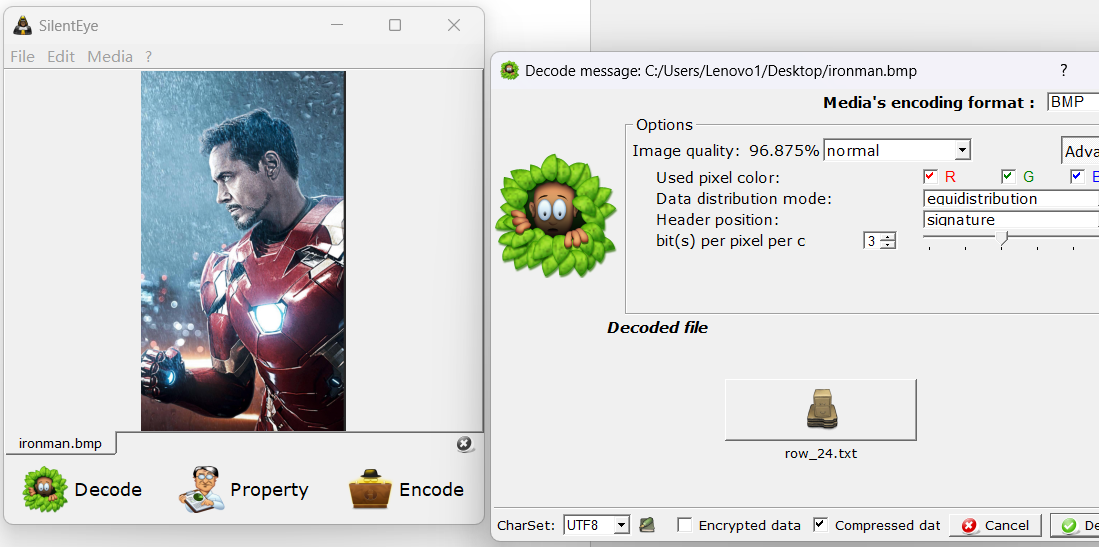
# ISCC2024 WriteUp

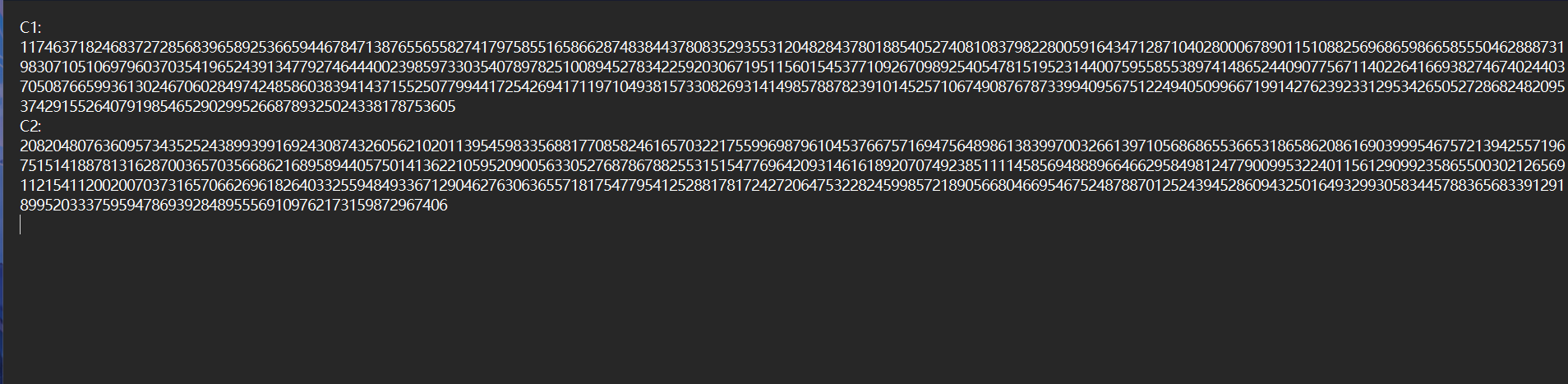
AkyOI 李承达 [3433778745@qq.com](mailto:3433778745@qq.com)

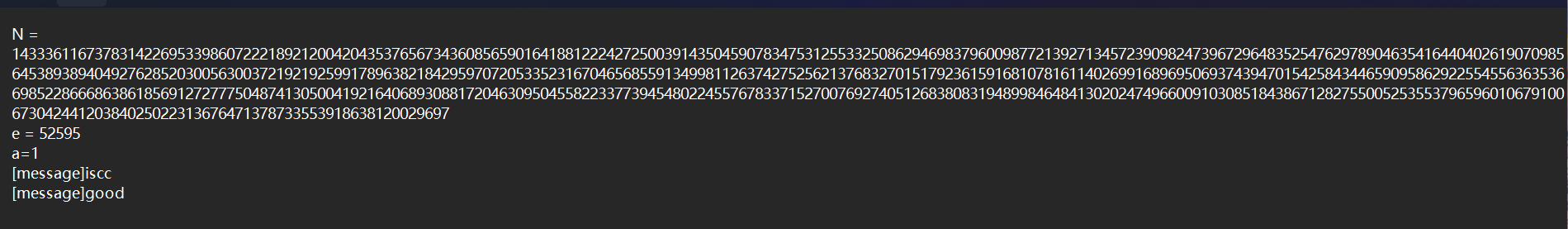
## Misc+钢铁侠在解密

## 解题思路



用工具提取，得到C1，C2





加上之前纸条的线索，之后写个sage脚本跑一下就能得到flag了

## Exp

# sage -python gtx.py

def calcuate(n1, n2):

    # calculate the value

    if 2 \* n2.degree() <= n1.degree() or n1.degree() == 1:

        return 1, 0, 0, 1

    # calculate the value

    flag\_resdssss = n1.degree() // 2

    # calculate the value

    taa, baa = n1.quo\_rem(x ^ flag\_resdssss)

    # calculate the value

    tbb, bbb = n2.quo\_rem(x ^ flag\_resdssss)

    reg0, reg2, reg3, reg4 = calcuate(taa, tbb)

    n3 = reg0 \* n1 + reg2 \* n2

    n4 = reg3 \* n1 + reg4 \* n2

    n5, val222 = n3.quo\_rem(n4)

    tdd, bdd = n4.quo\_rem(x ^ (flag\_resdssss // 2))

    tee, bee = val222.quo\_rem(x ^ (flag\_resdssss // 2))

    S00, S01, S10, S11 = calcuate(tdd, tee)

    RET00 = S01 \* reg0 + (S00 - n5 \* S01) \* reg3

    RET01 = S01 \* reg2 + (S00 - n5 \* S01) \* reg4

    RET10 = S11 \* reg0 + (S10 - n5 \* S11) \* reg3

    RET11 = S11 \* reg2 + (S10 - n5 \* S11) \* reg4

    return RET00, RET01, RET10, RET11

# calculate the gcd

def gcd\_function0(n1, n2):

    # calculate the gcd

    # calculate the gcd

    n5, n6 = n1.quo\_rem(n2)

    if n6 == 0:

        return n2

    reg0, reg2, reg3, reg4 = calcuate(n1, n2)

    n3 = reg0 \* n1 + reg2 \* n2

    n4 = reg3 \* n1 + reg4 \* n2

    if n4 == 0:

        return n3.monic()

    n5, n6 = n3.quo\_rem(n4)

    if n6 == 0:

        return n4

    return gcd\_function0(n4, n6)

# calculate the flag

PR.<x>=PolynomialRing(Zmod(NNNNNNNN))

# num1 and num2 are the two numbers

num1=11746371824683727285683965892536659446784713876556558274179758551658662874838443780835293553120482843780188540527408108379822800591643471287104028000678901151088256968659866585550462888731983071051069796037035419652439134779274644400239859733035407897825100894527834225920306719511560154537710926709892540547815195231440075955855389741486524409077567114022641669382746740244037050876659936130246706028497424858603839414371552507799441725426941711971049381573308269314149857887823910145257106749087678733994095675122494050996671991427623923312953426505272868248209537429155264079198546529029952668789325024338178753605

num2=2082048076360957343525243899399169243087432605621020113954598335688177085824616570322175599698796104537667571694756489861383997003266139710568686553665318658620861690399954675721394255719675151418878131628700365703566862168958944057501413622105952090056330527687867882553151547769642093146161892070749238511114585694888966466295849812477900995322401156129099235865500302126569112154112002007037316570662696182640332559484933671290462763063655718175477954125288178172427206475322824599857218905668046695467524878870125243945286094325016493299305834457883656833912918995203337595947869392848955569109762173159872967406

# NNNNNNNNN is the value of N

NNNNNNNN=14333611673783142269533986072221892120042043537656734360856590164188122242725003914350459078347531255332508629469837960098772139271345723909824739672964835254762978904635416440402619070985645389389404927628520300563003721921925991789638218429597072053352316704656855913499811263742752562137683270151792361591681078161140269916896950693743947015425843446590958629225545563635366985228666863861856912727775048741305004192164068930881720463095045582233773945480224557678337152700769274051268380831948998464841302024749660091030851843867128275500525355379659601067910067304244120384025022313676471378733553918638120029697

# x is the variable

val222 = 52595

# val1 and val0 are the two values

val1 = 1769169763

val0 = 1735356260

# x is the variable

global0 = (x\*2^32+val1)^val222 - num1

global1 = (x\*2^32+val0)^val222 - num2

# calculate the gcd

X=584734024210292804199275855856518183354184330877

resultttt = gcd\_function0(global0,global1)

# print the flag

flag\_resdssss = -resultttt.monic().coefficients()[0]

# print the flag

print(bytes.fromhex(hex(flag\_resdssss)[2:]).decode())