Re

FAKE

下载附件，是一个elf文件，扔IDA，加密逻辑很清晰，

```
puts("Give me your true flag:");
__isoc99_scanf("%50s", s);
if ( strlen(s) != 36 )
{
  puts("Wrong length.");
  exit(0);
}
for ( i = 0; i <= 35; ++i )
  v4[i] = s[i];
if ( (unsigned int)sub_401216((__int64)v4) == 1 )
  puts("Ohhhhhhhhhh!");
else
  puts("Wrong flag. Keep looking!");
return 0LL;
```

进到sub_401216里，好家伙，这么多行，反正是对输入的36个字符进行计算，那么解个方程?

```
from sympy import *
a0,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20,a21,a2
2,a23,a24,a25,a26,a27,a28,a29,a30,a31,a32,a33,a34,a35 = symbols('a0,a1,
a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20,a21,a22,a23,
a24,a25,a26,a27,a28,a29,a30,a31,a32,a33,a34,a35')
f1=-35 * a2 + 89 * a24 + -49 * a16 + -19 * a4 + 88 * a25 + -7 * a30 + a27 + -33
* a20 + -23 * a23 + 90 * a14 + -99 * a10 + 30 * a29 + -37 * a1 + -58 * a33 + 17
* a7 + 26 * a31 + -20 * a12 + -56 * a26 + 70 * a19 + 29 * a0 + -42 * a17 + 67 *
a35 + 11 * a6 + 66 * a15 + 53 * a11 - 53 * a3 + 63 * a32 - 65 * a21 + 9 * a9 -
50 * a28 - 48 * a8 - 70 * a22 + 48 * a13 - 68 * a34 - 14 * a5 - 67 * a18 - -874
f2=-41 * a25 + -47 * a16 + a14 + 67 * a34 + -20 * a1 + 47 * a33 + -79 * a19 +
-17 * a6 + 30 * a5 + (a18 *64) + -57 * a15 + 90 * a21 + 57 * a4 + -63 * a13 + 13
* a2 + 10 * a35 + -56 * a8 + 56 * a31 + -40 * a0 + -91 * a27 + 57 * a23 + 62 *
a10 + 90 * a9 + -92 * a17 + -5 * a29 + 60 * a22 - 13 * a12 + 5 * a28 - 63 * a32
+ 5 * a26 + 70 * a3 + 41 * a7 + 71 * a24 + 15 * a11 + 42 * a20 - 21163
f3=73 * a26 + 50 * a20 + 88 * a35 + 49 * a1 + 34 * a23 + 58 * a29 + 69 * a24 +
28 * a13 + 77 * a19 + 2 * a7 + -53 * a10 + -61 * a4 + 12 * a17 + 93 * a11 + -13
* a32 + 53 * a9 + 29 * a16 + -77 * a14 + 77 * a33 + 74 * a34 + -100 * a30 - 99 *
a27 - 87 * a25 + 36 * a6 + 59 * a3 + 81 * a21 + 28 * a28 + 7 * a0 + 54 * a22 - 5
* a31 - 41 * a2 + 5 * a18 - 93 * a15 + 10 * a12 - 27 * a8 + 24 * a5 - 45615
f4=-17 * a23 + 53 * a21 + 51 * a13 + 58 * a17 + -52 * a10 + -77 * a7 + 86 * a11
+ -77 * a16 + -100 * a18 + -61 * a2 + -46 * a20 + -61 * a6 + -46 * a19 + 51 * a4
+ -76 * a34 + -17 * a31 + 8 * a28 + 94 * a30 + 23 * a8 + -61 * a29 + -52 * a35 +
81 * a33 + -44 * a1 + 75 * a32 + -9 * a24 - 96 * a12 + 5 * a26 + 2 * a25 + 31 *
a22 + 43 * a15 - 2 * a0 - 92 * a14 + 13 * a5 - 99 * a3 + 63 * a27 + 8 * a9 -
-37017
f5=-25 * a4 + 91 * a0 + -43 * a32 + 17 * a13 + 9 * a15 + (a1 *64) + 9 * a16 + 59
* a3 + -29 * a14 + 32 * a18 + -69 * a26 + -81 * a33 + -69 * a9 + 60 * a19 + -35
* a21 + 40 * a11 + -44 * a7 + 78 * a22 + 68 * a28 + 70 * a29 + 3 * a2 + 61 * a6
+ 37 * a35 - 36 * a27 + 40 * a34 + 23 * a17 + 81 * a12 + 69 * a30 - 9 * a23 - 75
* a25 - 62 * a20 + 56 * a31 + 96 * a5 + 69 * a8 + 80 * a10 + 99 * a24 - 72092
```

```
f6=51 * a32 + -63 * a21 + 78 * a24 + -76 * a28 + 48 * a18 + 93 * a1 + 66 * a29 +
-86 * a27 + -3 * a0 + -79 * a26 + -20 * a8 + 90 * a6 + 6 * a30 + 47 * a16 + 50 *
a20 + 51 * a23 + -13 * a33 + -86 * a13 + 32 * a31 + -89 * a2 + 79 * a11 + -41 *
a7 + -56 * a14 + 54 * a19 - 96 * a34 - 34 * a25 - (a4 *64) - 36 * a35 + 48 * a10
- 39 * a5 + 20 * a3 + 61 * a17 - 56 * a15 - 97 * a22 + 96 * a9 - 61 * a12 -
-27809
f7=-44 * a35 + -26 * a30 + -36 * a4 + (a6 *64) + -65 * a26 + -46 * a8 + -33 *
a14 + -45 * a1 + -32 * a23 + -74 * a28 + 76 * a33 + 91 * a5 + 83 * a19 + 99 *
a32 + 98 * a7 + 22 * a34 + 83 * a13 + -13 * a0 + -66 * a11 + -25 * a2 + -9 * a31
+ 35 * a25 + 31 * a18 - 95 * a21 + 37 * a22 - 74 * a16 + 17 * a12 - 27 * a24 +
11 * a3 + 83 * a9 - 60 * a27 + 77 * a20 + 96 * a15 - 23 * a10 - 5 * a29 - 73 *
a17 - 9604
f8=-79 * a10 + 82 * a8 + -88 * a3 + -26 * a11 + 76 * a1 + 69 * a27 + -51 * a14 +
78 * a33 + -11 * a18 + -83 * a5 + 70 * a20 + -36 * a22 + -57 * a31 + 32 * a25 +
55 * a16 + 42 * a24 + -93 * a26 + 69 * a17 + 84 * a12 + 9 * a23 + -34 * a32 +
-84 * a2 + -18 * a7 + 60 * a29 - 99 * a30 - a0 + 24 * a21 - 36 * a4 + 9 * a35 +
89 * a15 + 72 * a19 + 86 * a13 - 8 * a28 + (a6 *64) + 19 * a9 + 71 * a34 - 25498
f9=-70 * a8 + 50 * a30 + 26 * a16 + 65 * a32 + -62 * a34 + 79 * a10 + -82 * a27
+ -16 * a29 + -60 * a1 + -80 * a15 + -54 * a2 + 75 * a6 + -31 * a17 + 29 * a12 +
28 * a28 + 57 * a14 + -68 * a4 + 86 * a0 + 82 * a13 + -20 * a11 + -18 * a23 + 88
* a18 + -57 * a25 + 94 * a9 - 51 * a5 - 58 * a7 - 2 * a3 + 94 * a31 - 6 * a21 -
59 * a19 + 25 * a20 - 66 * a35 - 62 * a24 + 89 * a26 + 12 * a22 - 86 * a33 -
-10472
f10=31 * a14 + -11 * a2 + 30 * a10 + 72 * a8 + 72 * a29 + -49 * a34 + 83 * a18 +
-63 * a0 + -88 * a4 + -91 * a20 + -11 * a17 + 38 * a3 + 53 * a35 + 31 * a5 + -75
* a22 + 14 * a26 + -7 * a24 + -7 * a31 + 77 * a23 + -46 * a6 + 47 * a19 + 48 *
a33 + 74 * a1 - 24 * a30 + 87 * a9 + 33 * a11 + 86 * a28 + 37 * a21 - 97 * a27 -
30 * a13 - 59 * a16 + 5 * a7 - 3 * a15 + 13 * a12 - 73 * a32 - 56 * a25 - 6560
f11=59 * a23 + -81 * a8 + 49 * a0 + -52 * a3 + 84 * a2 + 26 * a25 + -74 * a20 +
-97 * a6 + 14 * a15 + 77 * a30 + -66 * a28 + -89 * a12 + -95 * a13 + -70 * a10 +
-27 * a1 + -85 * a22 + -66 * a34 + -91 * a4 - 5 * a19 - 94 * a29 - 24 * a35 - 7
* a32 + 63 * a5 - 49 * a14 - 96 * a18 - 100 * a7 + 81 * a16 + 70 * a11 + 3 * a21
+ 28 * a24 - 14 * a9 + 59 * a17 + 24 * a31 - 25 * a27 + 20 * a33 - 77 * a26 -
-69431
f12=98 * a28 + 2 * a7 + 70 * a19 + 53 * a35 + 34 * a6 + 30 * a5 + 55 * a23 + 69
* a10 + 60 * a2 + -69 * a25 + 33 * a20 + 55 * a24 + 69 * a18 + 83 * a15 + -19 *
a13 + 22 * a21 + a16 + -53 * a22 + -58 * a4 + -63 * a29 - 91 * a26 + 28 * a34 +
5 * a3 + 35 * a8 + 27 * a1 - 31 * a27 + 10 * a12 + 84 * a33 + 24 * a14 + 42 *
a11 - 2 * a17 + 68 * a32 + 21 * a0 - a9 + 60 * a31 - 60 * a30 - 54106
f13=-88 * a22 + -96 * a3 + (a18 *64) + -61 * a15 + -92 * a13 + 50 * a8 + 90 *
a29 + 32 * a4 + -97 * a27 + 14 * a14 + a30 + 97 * a7 + 15 * a31 + -96 * a21 + 38
* a19 + -81 * a9 + -68 * a12 + 89 * a20 + 33 * a2 + 70 * a34 + 79 * a28 + -80 *
a35 + 76 * a10 - 38 * a5 + 5 * a6 + 60 * a11 - 8 * a26 - 59 * a23 + 9 * a24 + 34
* a17 - 60 * a1 + 98 * a25 + 48 * a16 - 17 * a32 - 96 * a0 + 54 * a33 - -8292
f14=-58 * a32 + 53 * a8 + -57 * a30 + -29 * a24 + -28 * a26 + -7 * a3 + 49 * a17
+ 60 * a25 + -45 * a16 + -50 * a0 + -98 * a28 + -92 * a12 + -22 * a23 + 33 * a13
+ 57 * a31 - 15 * a5 + 36 * a29 - 88 * a15 + 12 * a21 + 71 * a14 - 48 * a35 + 79
* a34 - 5 * a19 + 68 * a33 - 2 * a4 - 82 * a10 - 16 * a18 - 98 * a22 - (a27 *64)
+ 32 * a11 + 73 * a2 - 38 * a20 + 27 * a9 - 7 * a7 - 30 * a1 - 35 * a6 - -44677
f15=-86 * a28 + -69 * a33 + -31 * a21 + 91 * a15 + 91 * a8 + 58 * a16 + -91 *
a23 + 4 * a31 + -43 * a18 + -36 * a29 + 60 * a14 + 29 * a20 + -85 * a19 + 71 *
a27 + -22 * a11 + 95 * a9 + 19 * a12 + -20 * a30 + 6 * a34 + 49 * a6 + 13 * a10
- 23 * a13 + 17 * a7 - 79 * a22 + 12 * a3 - 7 * a4 - 12 * a2 - 78 * a17 - 56 *
a5 + 59 * a26 + 18 * a32 - 87 * a35 - 30 * a24 + 54 * a25 - 5 * a1 - 94 * a0 -
-17772
```

```
f16=-18 * a18 + -76 * a34 + 65 * a3 + -78 * a13 + -71 * a26 + -44 * a23 + 61 *
a7 + 63 * a1 + 9 * a16 + -17 * a9 + -93 * a12 + -85 * a20 + -73 * a35 + -87 *
a24 + -80 * a25 + -87 * a4 + 56 * a27 + -89 * a21 + 52 * a15 + 97 * a0 + -11 *
a19 + -94 * a10 + -92 * a29 + -20 * a17 - 95 * a5 - 13 * a8 + 80 * a31 - a33 +
37 * a30 + (a32 *64) + 11 * a22 - 39 * a14 + 80 * a6 - 33 * a11 - 62 * a2 - 74 *
a28 - -77151
f17=43 * a33 + 76 * a32 + -4 * a26 + 59 * a6 + -85 * a34 + 69 * a29 + 77 * a31 +
-95 * a8 + 75 * a16 + 54 * a20 + 65 * a22 + -9 * a9 + -61 * a24 + -45 * a25 + 47
* a4 + 31 * a5 + 36 * a23 + 20 * a13 - 40 * a2 - (a12 *64) - 40 * a14 + 81 * a10
- 35 * a0 - 12 * a27 + 35 * a30 + 63 * a15 - 65 * a19 + 31 * a18 - 42 * a35 + 33
* a11 - 19 * a3 + 65 * a21 - 78 * a7 - 48 * a28 - 77 * a17 - 11531
f18=67 * a23 + 95 * a5 + -37 * a9 + -71 * a25 + 33 * a32 + 96 * a14 + 47 * a31 +
-92 * a27 + -51 * a1 + -25 * a28 + -12 * a0 + 55 * a17 + 35 * a20 + 76 * a13 +
-73 * a15 + 84 * a12 + -72 * a3 + 71 * a24 + -41 * a7 + 28 * a8 + -93 * a34 +
-63 * a30 + 35 * a6 - 38 * a10 - 4 * a16 + 99 * a11 + 10 * a4 - 98 * a35 - 9 *
a18 + 22 * a21 - 6 * a26 + 82 * a2 - 6 * a33 - 13 * a29 + 25 * a22 - 35 * a19 -
4538
f19=98 * a16 + (a21 *64) + 93 * a8 + 78 * a2 + 56 * a15 + -51 * a6 + -17 * a26 +
-50 * a25 + -76 * a3 + -65 * a28 + -36 * a31 + 88 * a34 + 79 * a27 + 87 * a24 +
-52 * a29 + -72 * a13 + -17 * a23 + 54 * a0 + 45 * a10 + -17 * a33 + -49 * a4 +
-34 * a17 + 87 * a7 + -41 * a18 + 2 * a30 + -81 * a11 + 37 * a35 - 46 * a9 + 25
* a32 - 45 * a14 - 30 * a12 + 83 * a19 + 24 * a1 + 77 * a20 - 62 * a5 + 67 * a22
- 33735
f20=30 * a1 + -29 * a7 + -71 * a30 + 61 * a10 + -25 * a4 + 82 * a32 + 62 * a22 +
-40 * a34 + 90 * a3 + -36 * a14 + 37 * a17 + -21 * a19 + 55 * a21 + -70 * a26 +
92 * a6 + 75 * a31 + -35 * a29 + -50 * a25 + 8 * a33 + -74 * a13 + 34 * a35 + 29
* a24 + -10 * a15 + -75 * a16 + 24 * a18 + 98 * a0 + 41 * a20 - 54 * a28 - 5 *
a23 - 66 * a9 + 3 * a5 - 66 * a2 + 15 * a12 - 74 * a27 + 31 * a8 - 68 * a11 -
-7107
f21=-79 * a24 + -83 * a29 + 55 * a35 + -75 * a8 + 77 * a31 + 84 * a22 + -34 *
a13 + -13 * a15 + -11 * a19 + 28 * a17 + 98 * a9 + -69 * a3 + (a25 *64) + -66 *
a7 + -71 * a6 + 75 * a34 + 19 * a32 + -94 * a33 - 72 * a18 + 35 * a26 - 32 * a27
+ 76 * a1 + 80 * a28 + 66 * a10 + 3 * a12 - 99 * a14 + 17 * a30 - 94 * a0 + 12 *
a2 + 61 * a20 - 24 * a23 + 62 * a11 + 37 * a16 - 65 * a21 - 2 * a4 - 90 * a5 -
-17028
f22=-17 * a21 + -34 * a26 + (a15 *64) + 43 * a11 + 39 * a23 + 68 * a33 + -58 *
a13 + 21 * a1 + 19 * a19 + 96 * a8 + 24 * a3 + -76 * a2 + -94 * a16 + -37 * a4 +
-31 * a7 + -65 * a0 + -23 * a22 + 80 * a24 + -48 * a20 + -42 * a32 + 47 * a9 -
95 * a6 - 10 * a35 - 30 * a34 - 67 * a12 + 81 * a14 - 21 * a27 + 65 * a18 + 60 *
a25 + 31 * a17 - 20 * a31 - 32 * a30 - 83 * a28 + 20 * a5 - 3 * a29 + 7 * a10 -
-21641
f23=-52 * a7 + -82 * a23 + 14 * a27 + 52 * a6 + 67 * a11 + a3 + -37 * a30 + -76
* a0 + -82 * a22 + -92 * a24 + 53 * a20 + -90 * a5 + 3 * a34 + 93 * a2 + 77 *
a25 + -40 * a16 + -59 * a26 + -91 * a15 + 55 * a9 + -84 * a35 + -46 * a12 + -41
* a31 + -55 * a8 + 97 * a32 + 56 * a19 - 15 * a13 - 93 * a4 + 37 * a33 - 88 *
a18 - 16 * a10 + a14 + 48 * a17 - 80 * a21 + 17 * a29 - 94 * a28 - 12 * a1 -
-71317
f24=46 * a9 + -26 * a0 + 42 * a30 + 22 * a25 + -89 * a19 + 34 * a32 + -12 * a29
+ -16 * a7 + 22 * a18 + -52 * a31 + -71 * a21 + -55 * a5 + -76 * a4 + -94 * a10
+ -79 * a26 + 95 * a28 + 58 * a3 + -85 * a13 + -74 * a27 + -35 * a16 + 68 * a2 +
84 * a11 + -25 * a23 + -91 * a33 + -87 * a14 + -65 * a34 + 23 * a20 + -91 * a15
+ 34 * a12 + 53 * a1 - 16 * a24 + 83 * a22 + 5 * a17 - 71 * a6 + 41 * a35 + 68 *
a8 - -41387
f25=-59 * a10 + 35 * a4 + -53 * a8 + -18 * a1 + 9 * a32 + -45 * a9 + -97 * a12 +
-19 * a19 + -87 * a3 + 89 * a27 + 54 * a5 + 59 * a22 + 95 * a17 + 62 * a26 + 6 *
a20 + (a14 *64) + -50 * a13 + -95 * a30 + -68 * a16 + 10 * a0 - a2 - a28 + 17 *
a18 - 76 * a6 - 24 * a23 - 76 * a11 + 33 * a34 - 98 * a31 - 60 * a29 - 74 * a35
+ 31 * a7 + 50 * a24 + 25 * a21 - 83 * a33 + 25 * a25 + 52 * a15 - -30463
```

```
f26=-68 * a5 + 77 * a21 + -34 * a15 + -42 * a35 + 30 * a28 + -54 * a30 + -53 *
a20 + 98 * a33 + 70 * a32 + 99 * a19 + -27 * a25 + 84 * a34 + -73 * a14 + -54 *
a7 + -45 * a26 + -97 * a18 + 40 * a10 + 73 * a27 + -55 * a11 + 52 * a29 + -29 *
a13 + 32 * a24 + -80 * a0 + -79 * a17 + -39 * a6 + 88 * a1 + 44 * a2 - 50 * a3 -
2 * a22 - 44 * a31 - 62 * a8 - 51 * a4 + 12 * a16 - 55 * a9 + 40 * a12 + 76 *
a23 - -14435
f27=20 * a11 + -64 * a1 + 74 * a30 + 16 * a16 + -83 * a33 + 16 * a2 + -17 * a17
+ -28 * a8 + 9 * a7 + 86 * a20 + 70 * a13 + -76 * a19 + -31 * a28 + 77 * a14 +
48 * a15 + -78 * a31 + -82 * a26 + 69 * a3 + 70 * a5 + 95 * a6 - 60 * a4 + 30 *
a27 + 3 * a29 - 29 * a32 + 5 * a24 + 55 * a0 + 36 * a23 - 90 * a22 + 37 * a35 +
78 * a34 - 62 * a10 + 46 * a9 + 63 * a21 - 39 * a12 - (a18 *64) - 27 * a25 -
23472
f28=97 * a21 + -71 * a2 + -7 * a19 + -55 * a3 + 85 * a16 + -97 * a34 + -29 * a27
+ -79 * a32 + 50 * a28 + 7 * a0 + 92 * a6 + -57 * a24 + -89 * a11 + -47 * a5 +
-39 * a30 + (a8 *64) + -63 * a12 + -46 * a9 + -82 * a17 + 39 * a23 + 58 * a13 -
81 * a1 + 33 * a29 + 89 * a7 - 14 * a20 + 97 * a33 + 10 * a35 - 46 * a14 + 81 *
a4 + 89 * a15 + 81 * a22 - 44 * a31 - 60 * a10 - 20 * a26 + 18 * a18 + 91 * a25
- 7913
f29=-41 * a10 + 67 * a29 + -22 * a15 + -79 * a12 + -18 * a2 + 23 * a20 + -20 *
a14 + (a19 *64) + 91 * a5 + 49 * a17 + 52 * a18 + -89 * a25 + -93 * a35 + -70 *
a11 + -45 * a24 + 88 * a30 + 92 * a31 + 44 * a26 + -5 * a1 + -48 * a6 + -16 *
a22 + 88 * a32 + 91 * a33 + 82 * a28 + 98 * a8 - 63 * a13 - 8 * a9 - a16 - 4 *
a27 - 47 * a4 - 6 * a7 + 84 * a3 - 6 * a34 + 69 * a21 - 4 * a23 - 80 * a0 -
23824
f30=-28 * a35 + -72 * a6 + -46 * a32 + 99 * a20 + -69 * a22 + -94 * a12 + -35 *
a8 + -29 * a0 + 89 * a11 + 61 * a7 + -92 * a31 + 99 * a21 + 27 * a16 + -48 * a24
+ -51 * a4 + -39 * a25 + 84 * a30 + 34 * a14 + -73 * a17 + -92 * a18 + 72 * a2 -
14 * a13 - a19 + 2 * a9 + 3 * a29 - 61 * a33 - 6 * a26 - 57 * a15 - 8 * a27 - 29
* a10 - 2 * a23 - 23 * a34 + 41 * a3 + 42 * a28 - -13865
f31=90 * a19 + 58 * a30 + 74 * a12 + 22 * a29 + -29 * a20 + -49 * a22 + 88 * a14
+ -51 * a24 + 44 * a21 + 28 * a13 + 62 * a25 + 85 * a8 + -66 * a32 + 43 * a10 +
32 * a33 + 75 * a34 + 44 * a1 + 49 * a28 + -21 * a26 + 60 * a4 + -40 * a0 + -98
* a15 + -37 * a9 + 78 * a16 + 96 * a35 - 84 * a18 - 2 * a7 + 43 * a2 - 28 * a6 -
77 * a3 - 30 * a17 - 95 * a5 + 5 * a23 + 85 * a31 + 5 * a27 + 47 * a11 - 50179
f32=30 * a1 + -92 * a26 + 4 * a29 + -89 * a20 + 57 * a15 + -66 * a0 + -85 * a12
+ 91 * a35 + -68 * a2 + -100 * a25 + -88 * a18 + 46 * a33 + 50 * a31 + -85 * a4
+ -92 * a6 + -54 * a7 + 83 * a23 + -25 * a24 + -91 * a5 + 85 * a10 + -15 * a16 -
59 * a27 - 91 * a8 + 73 * a32 + 44 * a19 + 5 * a34 + 68 * a14 - 32 * a21 - 26 *
a30 - 56 * a17 - 95 * a3 - 16 * a13 - 76 * a11 - 48 * a9 - 88 * a22 + 65 * a28 -
-75429
f33=85 * a30 + -35 * a24 + -58 * a18 + 16 * a12 + -45 * a7 + 49 * a35 + 8 * a11
+ 54 * a22 + -33 * a4 + 4 * a17 + -49 * a20 + -71 * a13 + -23 * a23 + -19 * a21
+ 62 * a2 + -41 * a19 + 46 * a15 + 5 * a1 + -2 * a5 + 88 * a9 + 84 * a16 + 77 *
a6 - 6 * a26 + 51 * a33 - 96 * a31 + 59 * a14 - 62 * a8 - 55 * a25 - 32 * a34 +
69 * a32 - 48 * a28 - 21 * a27 + 31 * a0 - 98 * a10 - 96 * a29 - 71 * a3 -
-18764
f34=88 * a12 + 84 * a34 + 66 * a24 + 99 * a16 + -78 * a32 + -88 * a11 + -21 *
a35 + 25 * a18 + -81 * a19 + 74 * a20 + -50 * a2 + -46 * a21 + 28 * a15 + -100 *
a5 + 53 * a28 + -93 * a9 + -69 * a1 + -61 * a0 + 26 * a8 + -66 * a6 + -66 * a27
- 42 * a4 + 89 * a33 - 30 * a31 - 45 * a22 + 13 * a14 - 29 * a3 + 33 * a10 + 54
* a23 + 18 * a30 - 39 * a29 + 15 * a13 + 83 * a26 - 28 * a7 + 2 * a25 - -20428
f35=10 * a31 + (a28 *64) + 97 * a5 + -7 * a27 + 62 * a14 + 60 * a24 + 27 * a34 +
-11 * a10 + -97 * a22 + 14 * a33 + -43 * a11 + 40 * a20 + 31 * a13 + 44 * a29 +
-68 * a3 + -36 * a1 + -38 * a9 + -7 * a12 + a26 + -50 * a6 + 59 * a8 + 88 * a30
+ 46 * a0 - 34 * a15 + 10 * a4 + 84 * a18 + 13 * a7 + 14 * a25 - 5 * a16 - 31 *
a32 - 48 * a19 - 55 * a35 - 96 * a2 - 83 * a23 - 11973
```

```
f36=-45 * a26 + -10 * a35 + -40 * a9 + 97 * a10 + 6 * a22 + 58 * a34 + 4 * a31 +
55 * a21 + -99 * a4 + -57 * a8 + 2 * a7 + 57 * a24 + -54 * a25 + 39 * a29 + -91
* a1 + -32 * a20 + -30 * a11 + 16 * a12 + 45 * a17 + 90 * a32 + 26 * a5 - 59 *
a28 + 7 * a2 - 88 * a3 + 36 * a15 - 73 * a6 - 6 * a27 + 99 * a13 - 96 * a0 - 72
* a16 + 27 * a19 + 79 * a23 - 28 * a18 - 90 * a30 - 6 * a33 + 58 * a14 - -23186
print(solve([f1,
f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,f17,f18,f19,f20,f21,f22,f23,
f24,f25,f26,f27,f28,f29,f30,f31,f32,f33,f34,f35,f36]))
```

稍微整理一下得到一个flag：hgame{@FAKE_flag!-do_Y0u_know_SMC?}，提交试试？不对。

于是搜索SMC，得到以下内容

自修改代码（Self-modifying code）是指程序在运行期间（Run time）修改自身指令。可能的用途有：病毒利用此方法逃避杀毒软件的查杀，反静态分析，反盗版[1]，单片机程序升级。

哦豁，这玩意要动态调试？试试，于是链接Linux，发现输入上面得到的flag就对了！看来不能动态调试。于是回到start函数

```
void __fastcall __noreturn start(__int64 a1, __int64 a2, void (*a3)(void))
{
  __int64 v3; // rax
  int v4; // esi
  __int64 v5; // [rsp-8h] [rbp-8h] BYREF
  char *retaddr; // [rsp+0h] [rbp+0h] BYREF

  v4 = v5;
  v5 = v3;
  _libc_start_main((int (__fastcall *)(int, char **, char **))main, v4, &retaddr, (void (*)(void))init, fini, a3, &v5);
  __halt();
}
```

里面的函数挨个看看，于是在init函数中的funcs_406BD9中发现

```
funcs_406BD9    dq offset sub_401210
                dq offset sub_406A11
```

进一步到sub_406A11，到了这个函数

```
int sub_406A11()
{
  int result; // eax
  char buf[100]; // [rsp+0h] [rbp-70h] BYREF
  int fd; // [rsp+64h] [rbp-Ch]
  char *i; // [rsp+68h] [rbp-8h]

  fd = open("/proc/self/status", 0);
  read(fd, buf, 0x64uLL);
  for ( i = buf; *i != 84 || i[1] != 114 || i[2] != 97 || i[3] != 99 || i[4] != 101 || i[5] != 114; ++i )
    ;
  result = atoi(i + 11);
  if ( !result )
    result = sub_40699B();
  return result;
}
```

嗯，继续看sub_40699B，

```
__int64 sub_40699B()
{
  __int64 result; // rax
  unsigned int i; // [rsp+Ch] [rbp-4h]

  mprotect(&dword_400000, 0x10000uLL, 7);
  for ( i = 0; ; ++i )
  {
    result = i;
    if ( i > 0x43E )
      break;
    *((_BYTE *)sub_401216 + (int)i) ^= byte_409080[i];
  }
  return result;
}
```

看来这就是SMC的内容了，于是写IDA脚本：

```
#include <idc.idc>

static main()
{
    auto addr = 0x401216;
    auto addr2=0x409080;
    auto i = 0;
    for(i=0;i<=0x43E;i++)
    {
        PatchByte(addr+i,Byte(addr+i)^Byte(addr2+i));
    }
}
```

先把函数sub_401216(就是之前的加密函数)undefine，然后运行脚本，再在401216处把db转成代码，并建立新函数，得到SMC后的加密函数

```
 87  for ( i = 0; i <= 5; ++i )
 88  {
 89    for ( j = 0; j <= 5; ++j )
 90    {
 91      for ( k = 0; k <= 5; ++k )
 92        v4[6 * i + j] += v2[6 * k + j] * *(_DWORD *)(4LL * (6 * i + k) + a1);
 93    }
 94  }
 95  for ( l = 0; l <= 5; ++l )
 96  {
 97    for ( m = 0; m <= 5; ++m )
 98    {
 99      if ( v4[6 * l + m] != v3[6 * l + m] )
100        v10 = 0;
101    }
102  }
103  return v10;
104}
```

主要逻辑是这个样子，并且其中的v2和最终的v3是已知的，在第一个嵌套的for循环中是对v4赋值，赋值内容是v2和a1的矩阵乘法，这个解密过程好麻烦啊，于是直接解方程（python的sympy真好用）

```
from sympy import *
v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v210,v211,v212,v213,v214,v215,v216,v217,
v218,v219,v220,v221,v222,v223,v224,v225,v226,v227,v228,v229,v230,v231,v232,v233,
v234,v235=symbols("v20,v21,v22,v23,v24,v25,v26,v27,v28,v29,v210,v211,v212,v213,v
214,v215,v216,v217,v218,v219,v220,v221,v222,v223,v224,v225,v226,v227,v228,v229,v
230,v231,v232,v233,v234,v235")
v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,v310,v311,v312,v313,v314,v315,v316,v317,
v318,v319,v320,v321,v322,v323,v324,v325,v326,v327,v328,v329,v330,v331,v332,v333,
v334,v335=symbols("v30,v31,v32,v33,v34,v35,v36,v37,v38,v39,v310,v311,v312,v313,v
314,v315,v316,v317,v318,v319,v320,v321,v322,v323,v324,v325,v326,v327,v328,v329,v
330,v331,v332,v333,v334,v335")
a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a110,a111,a112,a113,a114,a115,a116,a117,
a118,a119,a120,a121,a122,a123,a124,a125,a126,a127,a128,a129,a130,a131,a132,a133,
a134,a135=symbols("a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a110,a111,a112,a113,a
114,a115,a116,a117,a118,a119,a120,a121,a122,a123,a124,a125,a126,a127,a128,a129,a
130,a131,a132,a133,a134,a135")
v30 = 55030
v31 = 61095
v32 = 60151
v33 = 57247
v34 = 56780
v35 = 55726
v36 = 46642
```

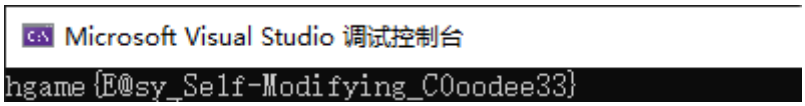
```
v37 = 52931
v38 = 53580
v39 = 50437
v310 = 50062
v311 = 44186
v312 = 44909
v313 = 46490
v314 = 46024
v315 = 44347
v316 = 43850
v317 = 44368
v318 = 54990
v319 = 61884
v320 = 61202
v321 = 58139
v322 = 57730
v323 = 54964
v324 = 48849
v325 = 51026
v326 = 49629
v327 = 48219
v328 = 47904
v329 = 50823
v330 = 46596
v331 = 50517
v332 = 48421
v333 = 46143
v334 = 46102
v335 = 46744
v20 = 104
v21 = 103
v22 = 97
v23 = 109
v24 = 101
v25 = 123
v26 = 64
v27 = 95
v28 = 70
v29 = 65
v210 = 75
v211 = 69
v212 = 95
v213 = 102
v214 = 108
v215 = 97
v216 = 103
v217 = 33
v218 = 45
v219 = 100
v220 = 111
v221 = 95
v222 = 89
v223 = 48
v224 = 117
v225 = 95
v226 = 107
v227 = 111
v228 = 110
```

```
v229 = 119
v230 = 95
v231 = 83
v232 = 77
v233 = 67
v234 = 63
v235 = 125
f1=v30-+v20*a10+v26*a11+v212*a12+v218*a13+v224*a14+v230*a15
f2=v31-+v21*a10+v27*a11+v213*a12+v219*a13+v225*a14+v231*a15
f3=v32-+v22*a10+v28*a11+v214*a12+v220*a13+v226*a14+v232*a15
f4=v33-+v23*a10+v29*a11+v215*a12+v221*a13+v227*a14+v233*a15
f5=v34-+v24*a10+v210*a11+v216*a12+v222*a13+v228*a14+v234*a15
f6=v35-+v25*a10+v211*a11+v217*a12+v223*a13+v229*a14+v235*a15
f7=v36-+v20*a16+v26*a17+v212*a18+v218*a19+v224*a110+v230*a111
f8=v37-+v21*a16+v27*a17+v213*a18+v219*a19+v225*a110+v231*a111
f9=v38-+v22*a16+v28*a17+v214*a18+v220*a19+v226*a110+v232*a111
f10=v39-+v23*a16+v29*a17+v215*a18+v221*a19+v227*a110+v233*a111
f11=v310-+v24*a16+v210*a17+v216*a18+v222*a19+v228*a110+v234*a111
f12=v311-+v25*a16+v211*a17+v217*a18+v223*a19+v229*a110+v235*a111
f13=v312-+v20*a112+v26*a113+v212*a114+v218*a115+v224*a116+v230*a117
f14=v313-+v21*a112+v27*a113+v213*a114+v219*a115+v225*a116+v231*a117
f15=v314-+v22*a112+v28*a113+v214*a114+v220*a115+v226*a116+v232*a117
f16=v315-+v23*a112+v29*a113+v215*a114+v221*a115+v227*a116+v233*a117
f17=v316-+v24*a112+v210*a113+v216*a114+v222*a115+v228*a116+v234*a117
f18=v317-+v25*a112+v211*a113+v217*a114+v223*a115+v229*a116+v235*a117
f19=v318-+v20*a118+v26*a119+v212*a120+v218*a121+v224*a122+v230*a123
f20=v319-+v21*a118+v27*a119+v213*a120+v219*a121+v225*a122+v231*a123
f21=v320-+v22*a118+v28*a119+v214*a120+v220*a121+v226*a122+v232*a123
f22=v321-+v23*a118+v29*a119+v215*a120+v221*a121+v227*a122+v233*a123
f23=v322-+v24*a118+v210*a119+v216*a120+v222*a121+v228*a122+v234*a123
f24=v323-+v25*a118+v211*a119+v217*a120+v223*a121+v229*a122+v235*a123
f25=v324-+v20*a124+v26*a125+v212*a126+v218*a127+v224*a128+v230*a129
f26=v325-+v21*a124+v27*a125+v213*a126+v219*a127+v225*a128+v231*a129
f27=v326-+v22*a124+v28*a125+v214*a126+v220*a127+v226*a128+v232*a129
f28=v327-+v23*a124+v29*a125+v215*a126+v221*a127+v227*a128+v233*a129
f29=v328-+v24*a124+v210*a125+v216*a126+v222*a127+v228*a128+v234*a129
f30=v329-+v25*a124+v211*a125+v217*a126+v223*a127+v229*a128+v235*a129
f31=v330-+v20*a130+v26*a131+v212*a132+v218*a133+v224*a134+v230*a135
f32=v331-+v21*a130+v27*a131+v213*a132+v219*a133+v225*a134+v231*a135
f33=v332-+v22*a130+v28*a131+v214*a132+v220*a133+v226*a134+v232*a135
f34=v333-+v23*a130+v29*a131+v215*a132+v221*a133+v227*a134+v233*a135
f35=v334-+v24*a130+v210*a131+v216*a132+v222*a133+v228*a134+v234*a135
f36=v335-+v25*a130+v211*a131+v217*a132+v223*a133+v229*a134+v235*a135
print(solve([f1,
f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12,f13,f14,f15,f16,f17,f18,f19,f20,f21,f22,f23,
f24,f25,f26,f27,f28,f29,f30,f31,f32,f33,f34,f35,f36]))
```

其中那些v什么和a什么是直接去方括号得到的。运行后得到

```
{a10: 104, a11: -103, a12: -97, a13: -109, a14: -101, a15: -123, a110: -95, a111: -83, a16: 69, a17: -64, a18: -115, a19
: -121, a112: 101, a113: -49, a114: -102, a115: -45, a116: -77, a117: -111, a118: 100, a119: -105, a120: -102, a121: -12
1, a122: -105, a123: -110, a124: 103, a125: -95, a126: -67, a127: -48, a128: -111, a129: -111, a130: 100, a131: -101, a1
32: -101, a133: -51, a134: -51, a135: -125}
Press any key to continue . . .
```

于是简单整理一下得flag



crypto

LikiPrime

直接脚本

```
from libnum import n2s,s2n

def gcd(a, b):   #求最大公约数
    if a < b:
        a, b = b, a
    while b != 0:
        temp = a % b
        a = b
        b = temp
    return a
def egcd(a,b):          #扩展欧几里得算法
    if a==0:
        return  (b,0,1)
    else:
        g,y,x=egcd(b%a,a)
        return (g,x-(b//a)*y,y)

def modinv(a,m):
    g,x,y=egcd(a,m)
    if g!=1:
        raise Exception('modular inverse does not exist')
    else:
        return x%m
if __name__ == '__main__':
    p =2**4253-1
    q =2**4423-1
    e = 65537
    d =modinv(e,(p-1)*(q-1))
```

```
    c
=274723046568693436449294087151532486839286897535313022378461495861932265596585247262668570469354159917080211702308916965347523963081589482678834480055194378990383244668002602101123854420784609037376693496552570502646685579670193045681644760033717120740476318593563813142202757973702239047641509560473188385514368034765804147555539511736976425888570287326879454013818994639479102268406402310797771276301050540373899230975720916201036158560040085923566525805554530408680304457848239630848124678469204220181024483649406818090874753222026480450001943693105807712298143832230105142486359659078087421422741826949936653889611564621526352962261094509459611197854942339809099340477546417035747304101520216859917015333692830074762521596335176301051252876884312164901291398051157669729305382198068677072871396301532176831880030907761920077930151057094846403428175672447021775760982351719578572401855257562249133948084612906093680161051555407672984218440844966768906294766977545875699525312078595752518782730038477411233741832248694580584615509961254001903540028967209593334479006669050737815450878202536673494253315217540329764335135501636746813055910654267325677417888425967588789481878200383381350037822494324414556426263062299310297854297886057151698168811002608704427494270398810285740113422081879305121465497284624729384486924042269774783148988004427702322075980549456826529670334936684481553746296771604868731689677425518661861112554221348792878903397684855860863804648153355939388600998198808118398319659223657792977654749811101087819365277573572395556838299580474502740218350144950620659087760126464884345642008510498854636972731246125693314988766562203816332002933503243437889863070276727879267986089985706383225156589922164389585151874961615612334049404427210575996338071547332195506960774004491011707124920689629962269768257079600997923163902494964153192564267189261217550917625775587466088192250048456869551577722122130117288496516081987452433117295265880011964262689100523055585126630282179419769307967711845044069897157484221584128103974182407553885172154865857141473065117496801000582559794780041356513213292042115672524100778125761067035645619886377867254247369267789140221428188031625458572695412941813385401518593038753855900185205489943065983187789786963430408980404289094050861336769181010547830918793539878878289472188608274669667268060236698571830119219904491840391114975734448194634476277255823635122615871665697144074660363358742180644828413968242962580182821316651583215693982452743159433370039743897072216855095688844123698745297458491116059452828366633503448889148896657676191573646532434154
    n =p*q
    m = pow(c,d,n)
    print(n2s(m))
```

得

```
b'hgame{Mers3nne^Pr!Me^re411y_s0+5O-1i7t1e!}'
Press any key to continue . . .
```