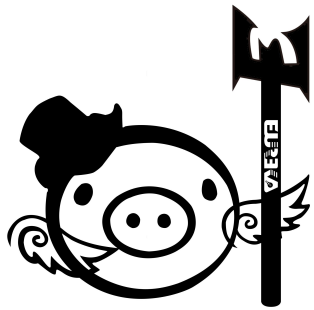
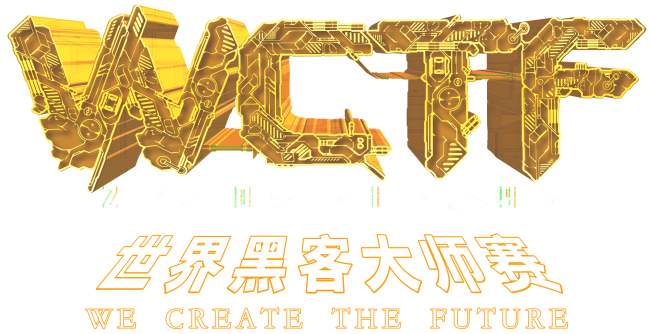
WCTF 2019 Challenge Toy Solution

Team r3kapig China

July 7, 2019





WCTF 2019 Challenge Toy Solution

Written in front

Overview

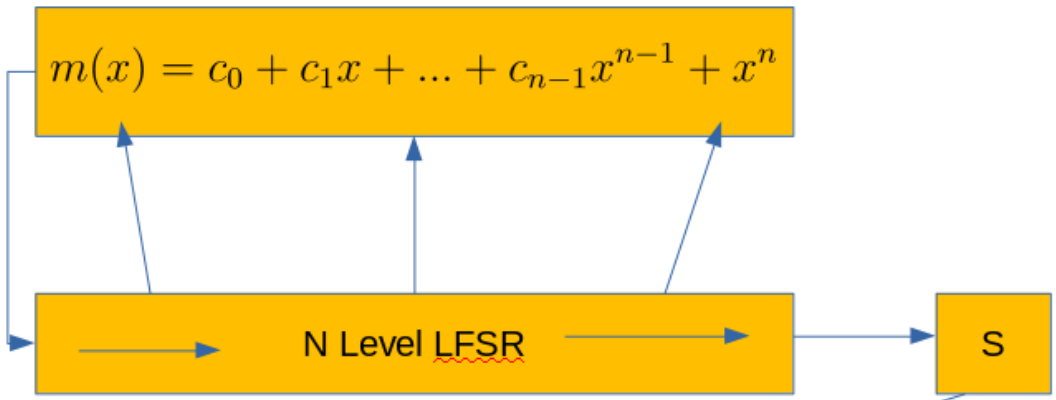
This a reverse+crypto challenge. We use SSE to achive a famous stream cipher which is named Toyocrypt. The player needs to reverse the binary and recognizes the algorithm. After that, the player need to attack the crypto method with the algebraic attack and recover the initial status of lfsr, which is the flag.

WCTF 2019 Challenge Toy Solution

Challenge Analysis

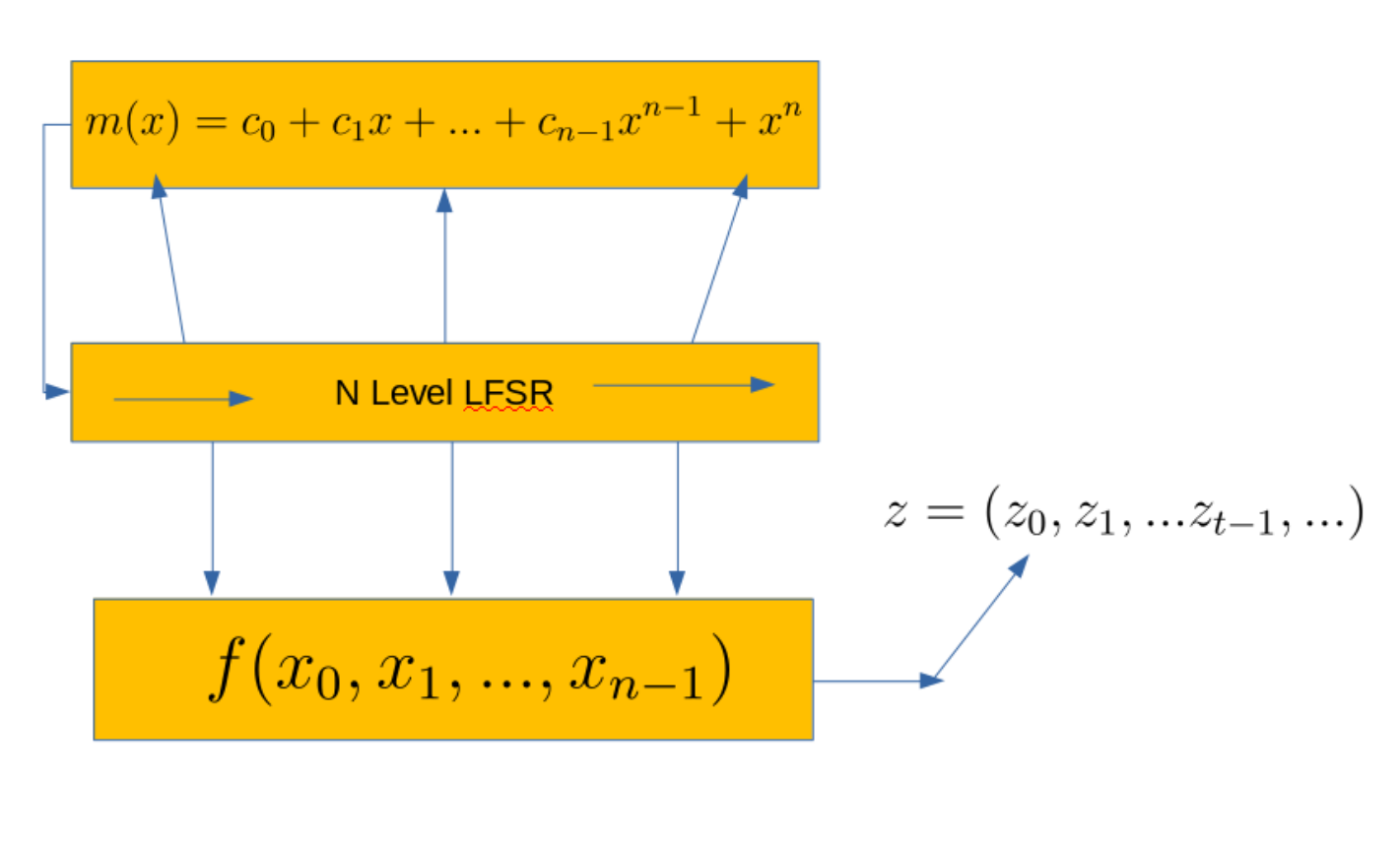
The dilemma Toy is drived from the algorithm Toyocrypt, which is a submission to the Japanese government Cryptrec call for cryptographic primitives.
To work out this problem, some simple reverse work is needed first. It can be found that we used some SSE instructions to implement a crypto algorithm. Actually, there are only two important parts of the whole process. One is LFSR function, and the other is nonlinear output combine function.

Feedforward model based on LSRF



$$s_{t+n} = s_t + c_1 s_{t+1} + ... + c_{n-1} s_{n-1+t} + c_n s_{n+t}, t \geq 0$$

Feedforward model based on LSRF

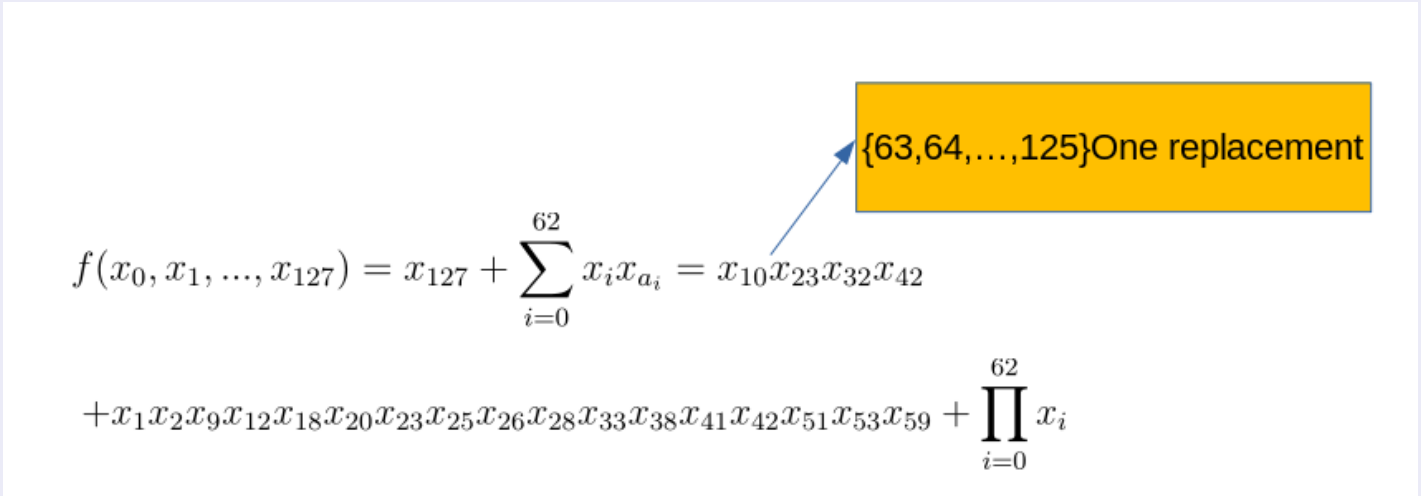


Reverse job is not very hard. SSE is interesting to achive the Toyocrypt. Some jobs, just like xor, are easy to achive. But some jobs, just like l-shift, are hard to achive. After that we can see the all part of the toyocrypt with some changes.
When reverse work is done, you will find that the essence of the program is to implement a toyocrypt cryptographic algorithm with some small changes.
The original nonlinear output function is like next page:

WCTF 2019 Challenge Toy Solution

Challenge Analysis

Toyocrypto is a submission to the Japanese government Cryptrec call for cryptographic primitives.
128 Level LFSR
Feedforward model based!



17 Mono

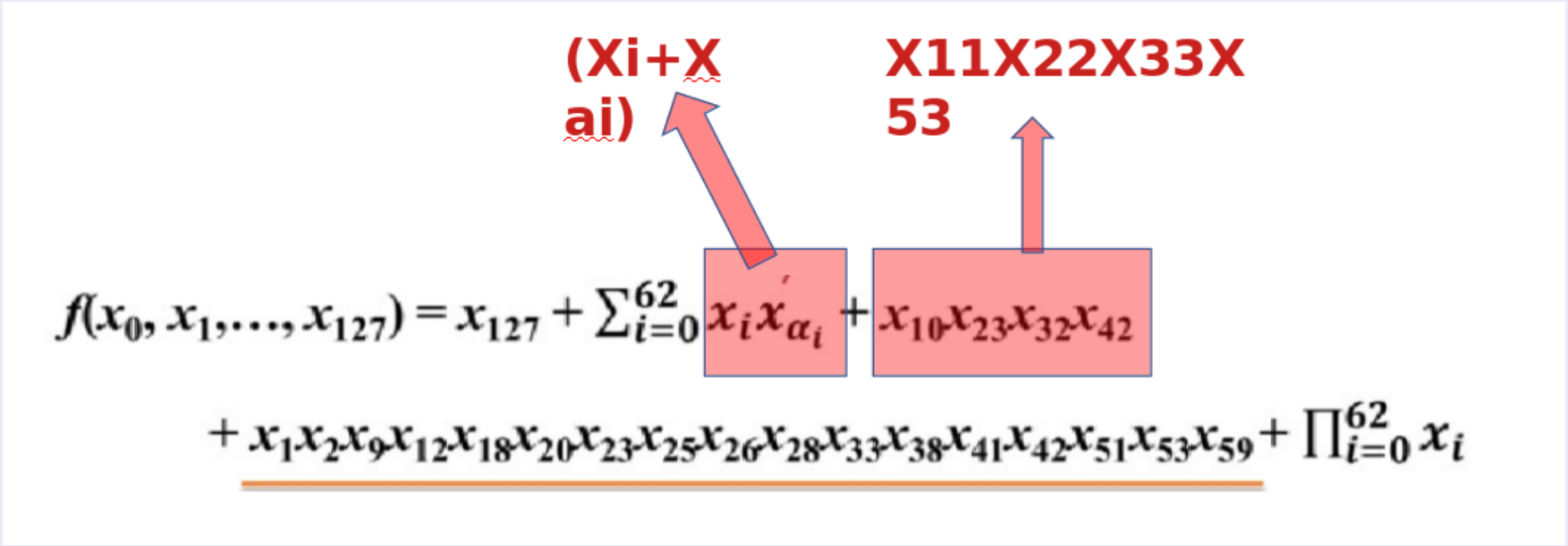
.....

But we made some changes to it.

WCTF 2019 Challenge Toy Solution

Challenge Analysis

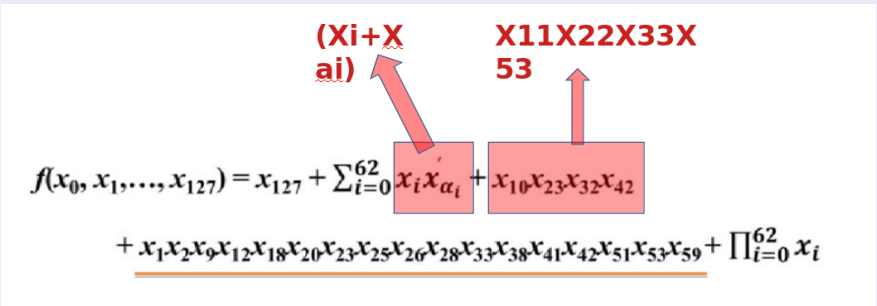
But we made some changes to it.Look Like Picture!



WCTF 2019 Challenge Toy Solution

Challenge Analysis

Some parts of toyocrypt have been changed by us. The first part was the quaternion in the feedforward function.
We changed it from $x_{10} * x_{23} * x_{32} * x_{42}$ to $x_{11} * x_{22} * x_{33} * x_{53}$.
So when we attack, we can't Directly use the existing attack methods of toyocrypt without understanding them. The second part that has been changed was the '&' in the quadratic term which is changed to ".
By doing this we reduce all of the cubic term in the algebraic attack In this way, the player can solve the problem with his personal notebook without consuming a lot of computing resources.



To workout the initial state, we need to crack this algorithm. The two mainly ways to crack the stream cipher is algebraic attack and correlation attack.
Correlation attack is a powerful attack method on many stream ciphers, just like A5-1, and so on. However, in this algorithm, it can't be cracked by correlation attack. So, to crack this cipher, we need implement the algebraic Attack.

WCTF 2019 Challenge Toy Solution

Challenge Analysis

First, let's look at the original feedforward function. If we multiply an annihilator, the high order terms in the equation will be reduced to 0.
The annihilator could be $x_{23} - 1$ or $x_{42} - 1$.
So, each outputbit can generate 2 equations like these. And these equations' highest order is 3.
To achive the attack, first we need to compute the annihilators. Because of the quaternion is $x_{11} * x_{22} * x_{33} * x_{53}$, so the anihilators are $(x_{33} - 1)$ and $(x_{53} - 1)$.

$$f(s_0,\ldots s_{127}) = s_{127} + \sum_{i=0}^{62} s_{\alpha_i} s_i + s_{10}s_{23}s_{32}s_{42} + s_1s_2s_9s_{12}s_{18}s_{20}s_{23}s_{25}s_{26}s_{28}s_{33}s_{38}s_{41}s_{42}s_{51}s_{53}s_{59} + \prod_{i=0}^{62} s_i$$

The picture $S = x$

Algebraic attack: original

In the equation, the left and right sides are multiplied by the annihilator at the same time. The equation will be converted into the following form
Annihilator: $(x_{23}-1)$ or $(x_{42}-1)$

$$f(x_0, x_1, \ldots, x_{127}) = x_{127} + \sum_{i=0}^{62} x_i x_{\alpha_i} + x_{10}x_{23}x_{32}x_{42}$$

$$+ x_1x_2x_9x_{12}x_{18}x_{20}x_{23}x_{25}x_{26}x_{28}x_{33}x_{38}x_{41}x_{42}x_{51}x_{53}x_{59} + \prod_{i=0}^{62} x_i$$

$$(x_{23}-1) * outputbit = x_{127} * (x_{23}-1) + \sum(x_i * x_{a_i} * (x_{23}-1)), i = range(0, 63)$$

$$(x_{42}-1) * outputbit = x_{127} * (x_{42}-1) + \sum(x_i * x_{a_i} * (x_{23}-1)), i = range(0, 63)$$

Algebraic attack: original

$128 : 0 - X_{128}$

$C(128, 2) : x_i x_j$

$C(128, 3) : x_i x_j x_k$

$128 + C(128, 2) + C(128, 3) = 349623$

Tips1

s[i]=sum(Init_status & magic[i])

Tips2

Each bitmap can be computed from 128 bitmaps before and the polyprim.

```
mask = 0xb64e4d3fa8e7331bd871fa30d46d4dba
magic = []
tmp = (1 << 128)
for i in range(128):
    tmp = tmp >> 1
    magic.append(tmp)
controller=bin(mask)[2:]
for i in range(128, 8256):
    i_magic = 0
    for point in range(128):
        if controller[point]=='1':
            i_magic ^= magic[i-128+point]
    magic.append(i_magic)
```

WCTF 2019 Challenge Toy Solution

Challenge Analysis

In algebraic attack, we need to do Gauss elimination. But there are quadratic terms here. So we define all quadratic terms as new unknown vars.

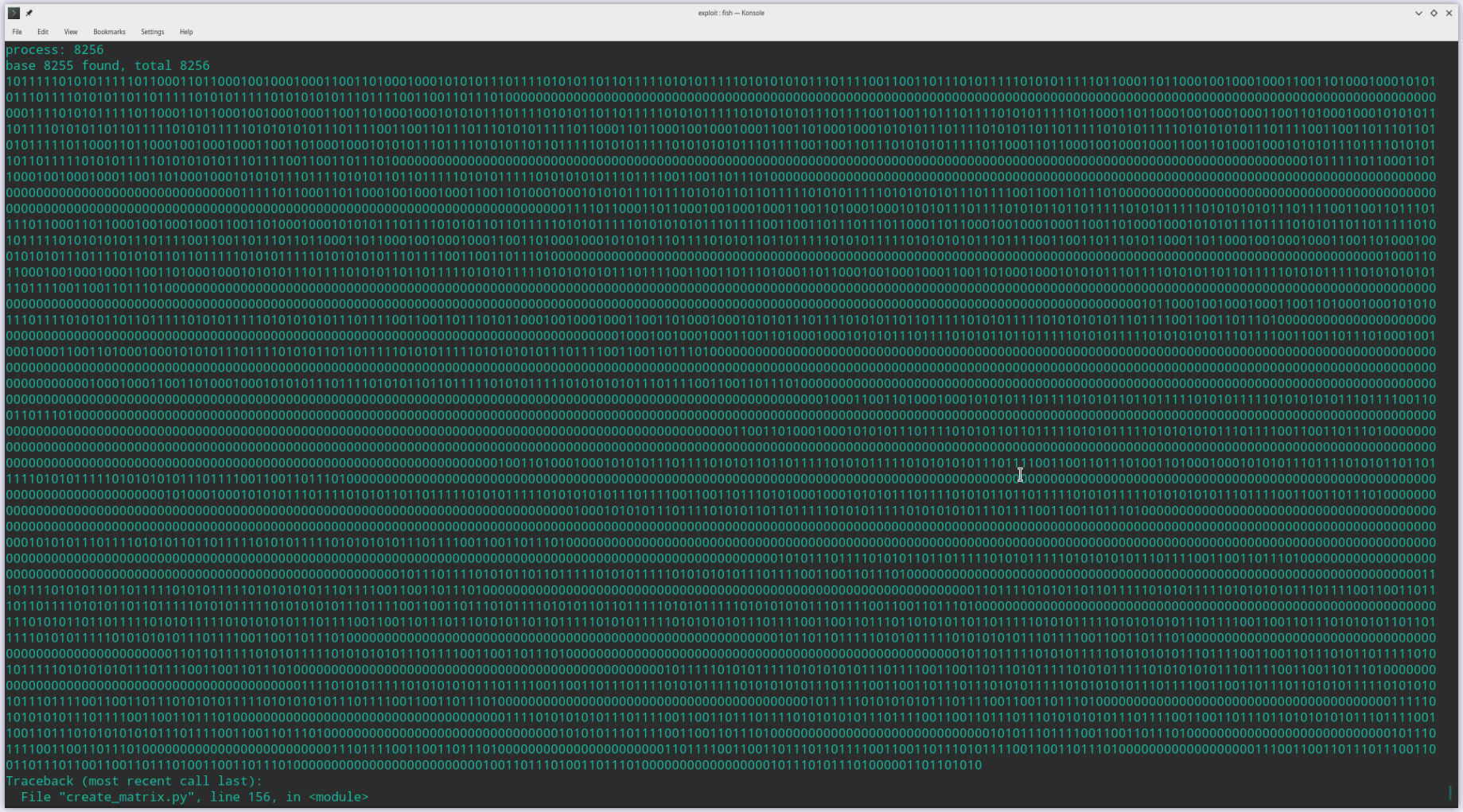
```
ttable = []
for i in range(128):
    tmp=[]
    for j in range(128):
        tmp.append(-1)
    ttable.append(tmp)
point = 128
for i in range(127):
    for j in range(i + 1, 128):
        ttable[i][j] = point
        ttable[j][i] = point
        point += 1
alreadylist=[]
for i in range(128):
    for j in range(128):
        assert ttable[i][j]==ttable[j][i]
        assert ttable[i][j]<8256
        if ttable[i][j] not in alreadylist and ttable[i][j]!=-1:
            alreadylist.append(ttable[i][j])
assert len(alreadylist)==8256-128
for i in range(128):
    assert ttable[i][i]==-1
```

Thus there are 128+C(128,2) unknown vars. One output bit can generate 2 equations by 2 annihilators. So 8256 bit is enough to finish the attack. Finally, we list all the equations and use Gauss elimination to restore the initial state of the register.

WCTF 2019 Challenge Toy Solution

Challenge Analysis

Gaussian elimination results out the first 128 bits is the flag!



Why we put out this problem?

There are many crypto problems in former CTFs. However, algebraic attack is so important in cryptography, so we think we need put out a problem about algebraic attack to help the competitors to learn this attack method.

By work out this problem, what can you get?

Knowing the standard algebraic attack process. Knowing the way to work out annihilator. Knowing the way to simplify the attack process for algebraic attack by XL (eXtended Linearizations) method.

The End!
Thanks!

The All Download Address -> https://github.com/r3kapig/WCTF-2019