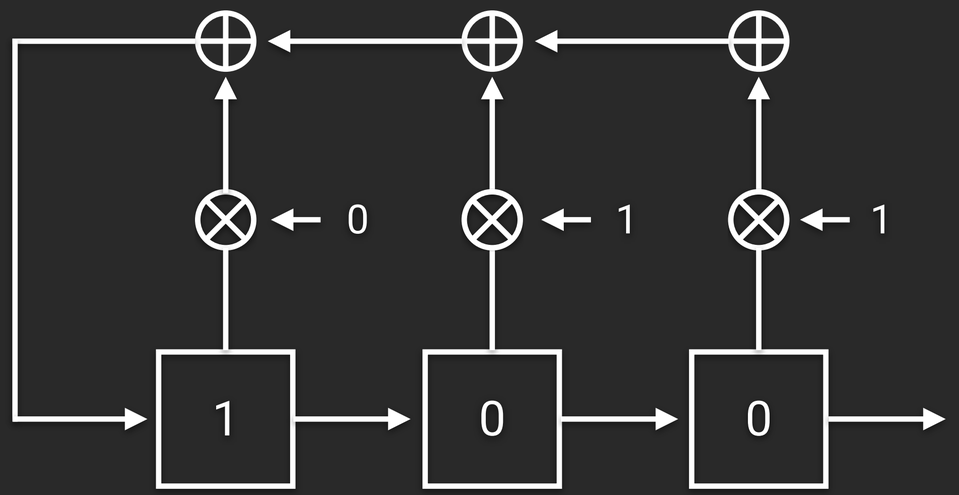
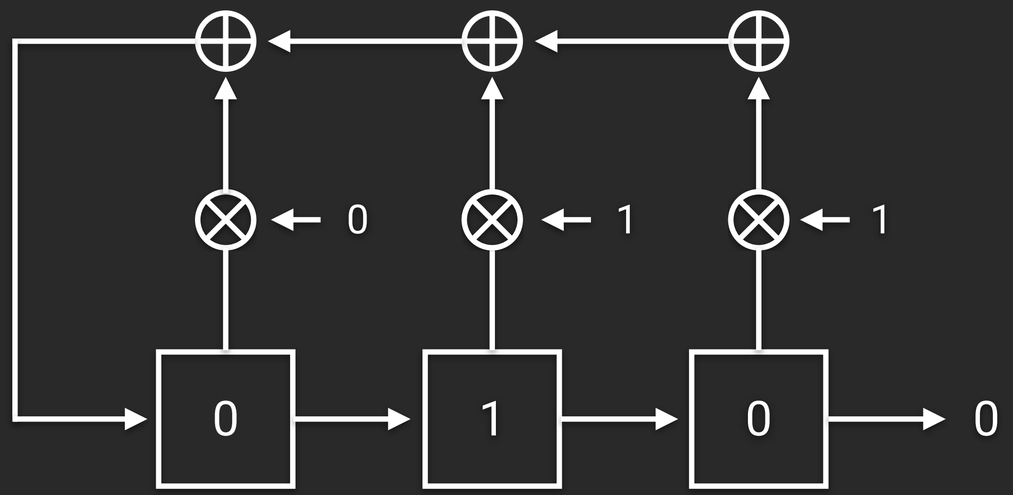
LFSR

oalieno

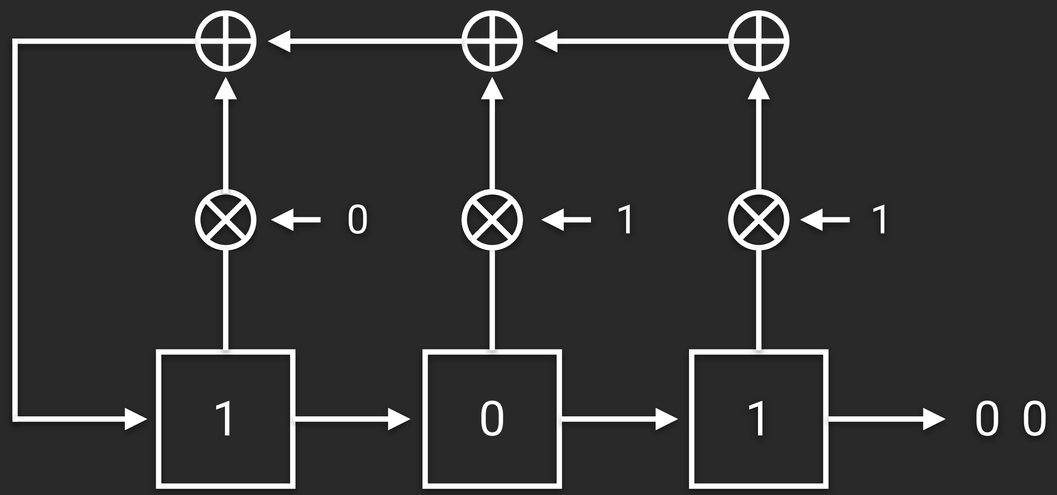
小例子



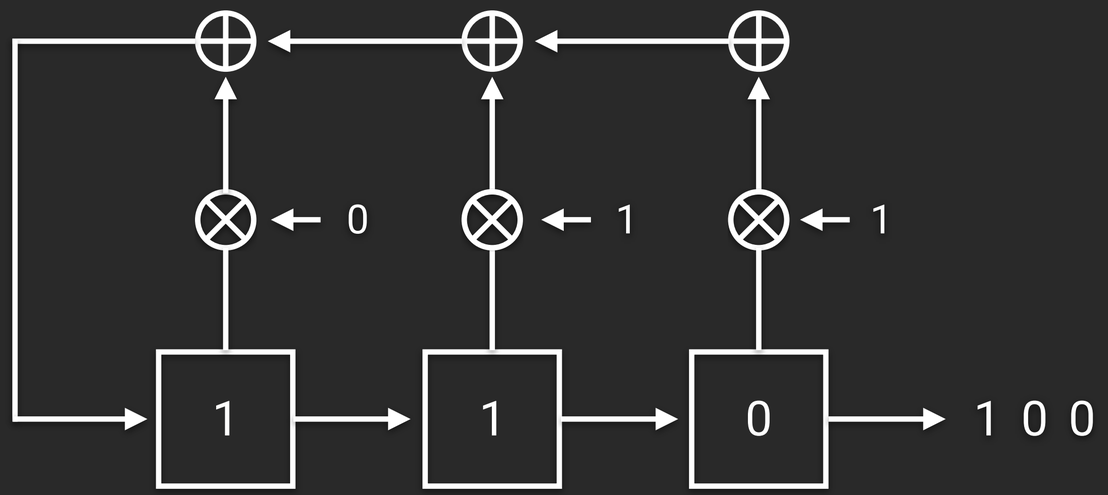
小例子



小例子

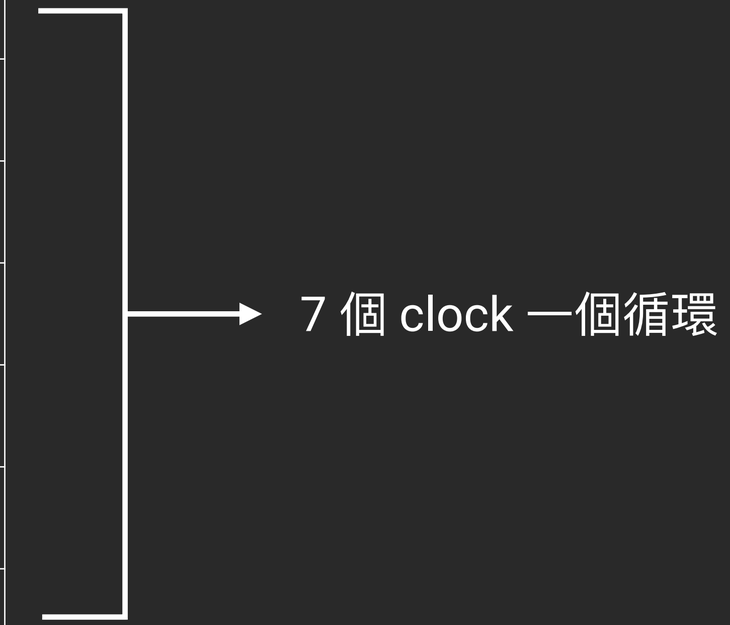


小例子

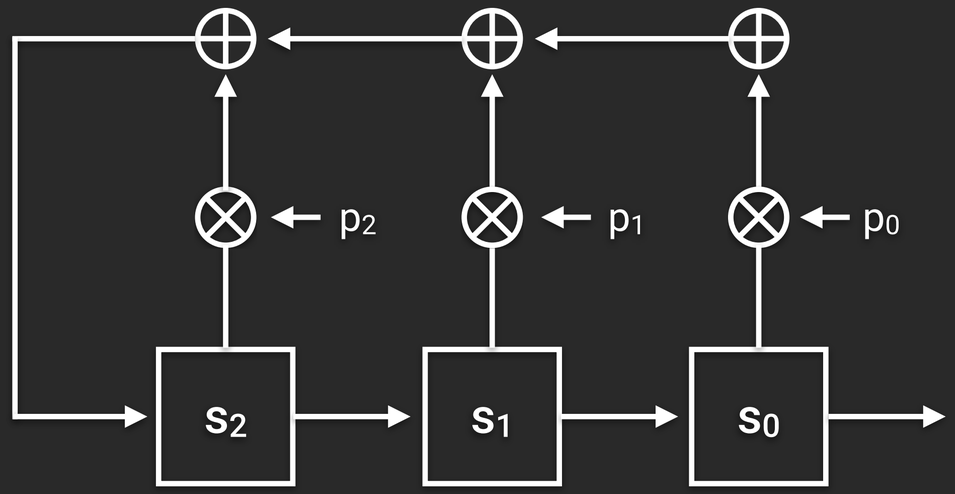


小例子

clk	FF2	FF1	FF0
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0



從數學的觀點



- 初始值 s_0, s_1, s_2
- 回饋係數 p_0, p_1, p_2
- 轉移方程 $s_i \equiv p_{i-1}s_{i-1} + p_{i-2}s_{i-2} + p_{i-3}s_{i-3} \mod 2$

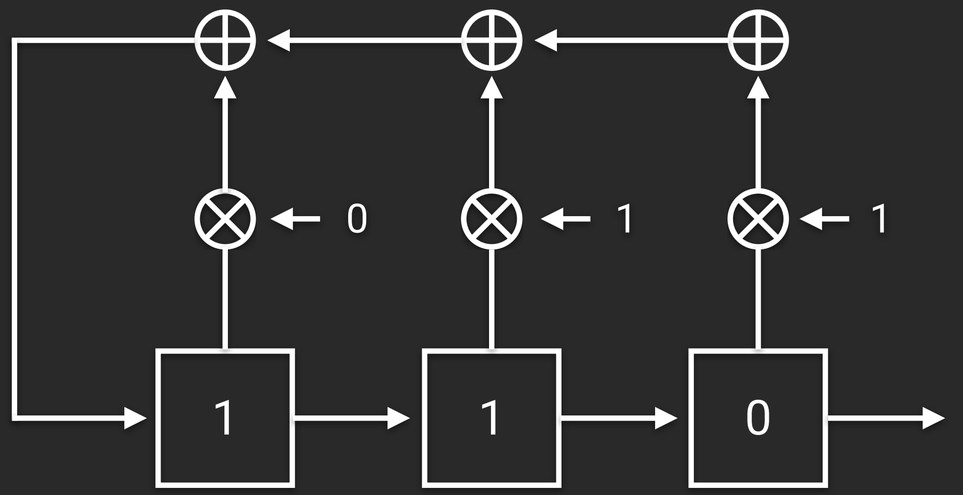
從數學的觀點

- 初始值 $s_0, s_1, \cdots, s_{m-1}$
- 回饋係數 $p_0, p_1, \cdots, p_{m-1}$
- 轉移方程 $s_i \equiv p_{i-1}s_{i-1} + p_{i-2}s_{i-2} + \cdots + p_{i-m}s_{i-m} \mod 2$

$$
\begin{aligned}
s_m &\equiv p_{m-1}s_{m-1} + p_{m-2}s_{m-2} + \cdots + p_0 s_0 \mod 2 \\
s_{m+1} &\equiv p_{m-1}s_m + p_{m-2}s_{m-1} + \cdots + p_0 s_1 \mod 2 \\
&\vdots
\end{aligned}
$$

使用 LFSR 作為 Stream Cipher

- 把 LFSR 產生的輸出當作 key，拿去做 xor cipher



1 0 0　密鑰

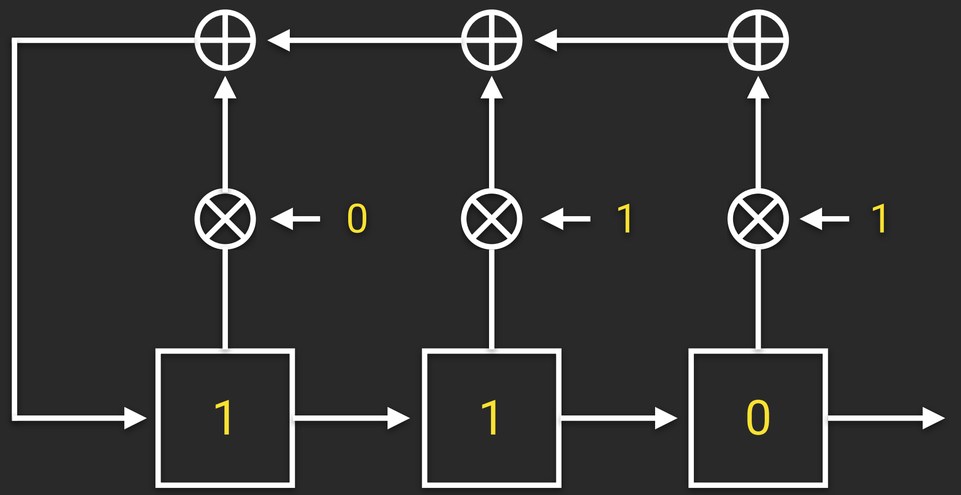
⊕

0 1 0　明文

0 1 0　密文

Known Plaintext Attack

- 攻擊者不知道黃色的部分
- 攻擊者知道了一小部分明文以及對應的密文，可推出一些 LFSR 的輸出



1 0 0 密鑰

⊕

0 1 0 明文

0 1 0 密文

解聯立方程式

- 只要知道 2n 個 bits 的輸出，攻擊者就可以算出回饋係數
- 比如知道 $s_0, s_1, \cdots, s_5$，那下面式子只會有 p_0, p_1, p_2 三個未知數
- 簡單的高斯消去法即可求解 (不一定有唯一解，也不一定最短)

$$
\begin{aligned}
s_3 &\equiv p_2 s_2 + p_1 s_1 + p_0 s_0 \mod 2 \\
s_4 &\equiv p_2 s_3 + p_1 s_2 + p_0 s_1 \mod 2 \\
s_5 &\equiv p_2 s_4 + p_1 s_3 + p_0 s_2 \mod 2
\end{aligned}
$$

Berlekamp Massey Algorithm

- 先介紹 Linear Recurrence
- 在 mod 13 下，[1, 2, 3, 2, 12] 符合 linear recurrence relation [7, 3, 1]
  - $1 \cdot 1 + 2 \cdot 3 + 3 \cdot 7 \equiv 2 \mod 13$
  - $2 \cdot 1 + 3 \cdot 3 + 2 \cdot 7 \equiv 12 \mod 13$

Sequence $a_0, a_1, \cdots$ satisfy a linear recurrence relation $p_1, p_2, \cdots, p_m$

iff $\forall i \geq m, a_i = \sum_{j=1}^{m} a_{i-j} p_j$

Berlekamp Massey Algorithm

- 這個演算法可以找到最短的 Linear Recurrence Relation
- 也可以用 Polynomial 來表示這個 Relation
- Relation [7, 3, 1] 就會是 $x^3 - 7x^2 - 3x - 1$

Berlekamp Massey Algorithm

sagemath

```
from sage.matrix.berlekamp_massey import berlekamp_massey

berlekamp_massey([GF(7)(1), 5, 1, 5])
```

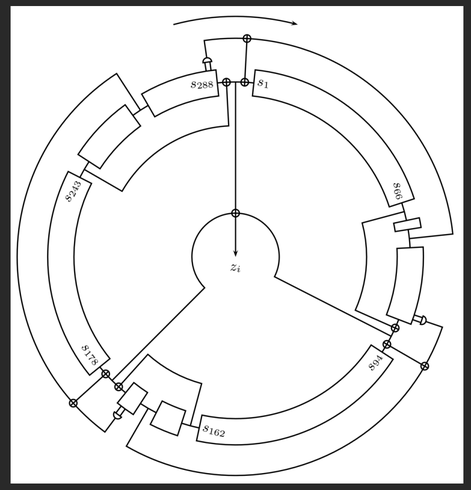
output

```
x^2 + 6
```

Mixed LFSR

https://en.wikipedia.org/wiki/Trivium_(cipher)

- 既然一個 LFSR 很容易被預測，那就兩個 LFSR
- 兩個不行，就三個，於是就有了 Trivium



Correlation Attack

- 那自己來簡單的組合一組 LFSR 來試試

```
class MYLFSR:
    def getbit(self):
        x1 = LFSR1.getbit()
        x2 = LFSR2.getbit()
        x3 = LFSR3.getbit()
        return (x1 & x2) ^ ((not x1) & x3)
```

Correlation Attack

x_1	x_2	x_3	輸出
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

75% of x_3 = 輸出

Correlation Attack

x_1	x_2	x_3	輸出
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

75% of x_2 = 輸出

Correlation Attack

- 假設回饋係數是已知的
- 要找回三個 LFSR 的初始值最簡單的做法就是暴搜全部可能
- 假設一個 LFSR 有的初始值有 32 bits 那就要爆搜 96 bits
- 其實可以單獨暴搜 LFSR3，根據暴搜的初始值產出的 x_3 去跟輸出比對，相同的比例有大約 75% 的話，就很有可能是真正的初始值
- 同理 LFSR2 也可以這樣做，最後只剩下 LFSR3 就直接暴
- 從要暴搜 2^{96} 變成暴搜 3×2^{32}

Fast Correlation Attack

- 有沒有比暴搜更好的做法，有
- Fast Correlation Attacks: Methods and Countermeasures
- A Fast Correlation Attack Implementation