$F = F(x_1 \dots x_n)$ Boolean function in $n$ Boolean variables

ANF

$$F = \bigoplus_{\{i_1 i_2 \dots i_t\} \subseteq \{1 2 \dots n\}} C_{\{i_1 i_2 \dots i_t\}} x_{i_1} x_{i_2} \dots x_{i_t}$$

sum is XOR, including $\phi$

$C_\phi \cdot 1$

in the example

$$p(x_1 x_2 x_3) = \underbrace{x_1 x_2 x_3}_{C_{\{1,2,3\}}=1} + x_1 x_2 + x_1 x_3 + x_2 x_3 + x_1 + (C_\phi = 0)$$

---

## How to compute ANF.

$F = (F(0), F(1), \dots, F(2^n - 1))$ truth table

$F(0\dots0)$ $F(0\dots01)$ $\dots$ $F(11\dots1)$

$C = (C_0, C_1, \dots, C_{2^n-1})$ ANF coefficients

$$C_{\{i_1 i_2 \dots i_t\}} = C_{(0\dots1\dots1\dots10\dots0)} = C_{2^{n-i_1} + 2^{n-i_2} + \dots + 2^{n-i_t}}$$

($1$ at positions $i_1, i_2, \dots, i_t$)

lexicographic ordering on $n$-bit strings.

in the example.

| 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|
| $C_0$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_5$ | $C_6$ | $C_7$ |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

**TH.** $F \cdot A_n = C$ all arithmetic operations are mod 2

where $A_n = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{[n]} = \begin{pmatrix} A_{n-1} & A_{n-1} \\ 0 & A_{n-1} \end{pmatrix}$

$A_n$ is $2^n \times 2^n$-matrix with entries $0, 1$.

multiplication $F \cdot A_n$ takes $2^{2n}$ operations.

in $n \cdot 2^{n-1}$ XORs

---

## Fast Algorithm to compute ANF

input : truth table of $F(x_1 \dots x_n)$, $F = F(0), F(1), \dots, F(2^n-1)$

output : ANF coeff. $C$.

notation : $f_{k,a}$ of length $2^k$, $0 \le a \le 2^{n-k}-1$

$0 \le k \le n$.

$f_k = (\underbrace{f_{k,0}}_{2^n}, \underbrace{f_{k,1}}_{2^k}, \dots, \underbrace{f_{k,2^{n-k}-1}}_{2^k})$ - $2^n$-bit string.

1. initialization. Set

$$f_{0,a} = F(a) \qquad 0 \le a \le 2^n - 1.$$

this defines $f_0$.

2. loop $k = 0, 1, \dots, n-1$ do

$$f_{k+1,a} = (\overbrace{\underbrace{f_{k,2a}}_{2^k}, \underbrace{f_{k,2a} + f_{k,2a+1}}_{2^k}}^{2^{k+1}})$$

↑ vector XOR

$0 \le a \le 2^{n-k-1} - 1$.

3. Put $C = f_{n,0}$

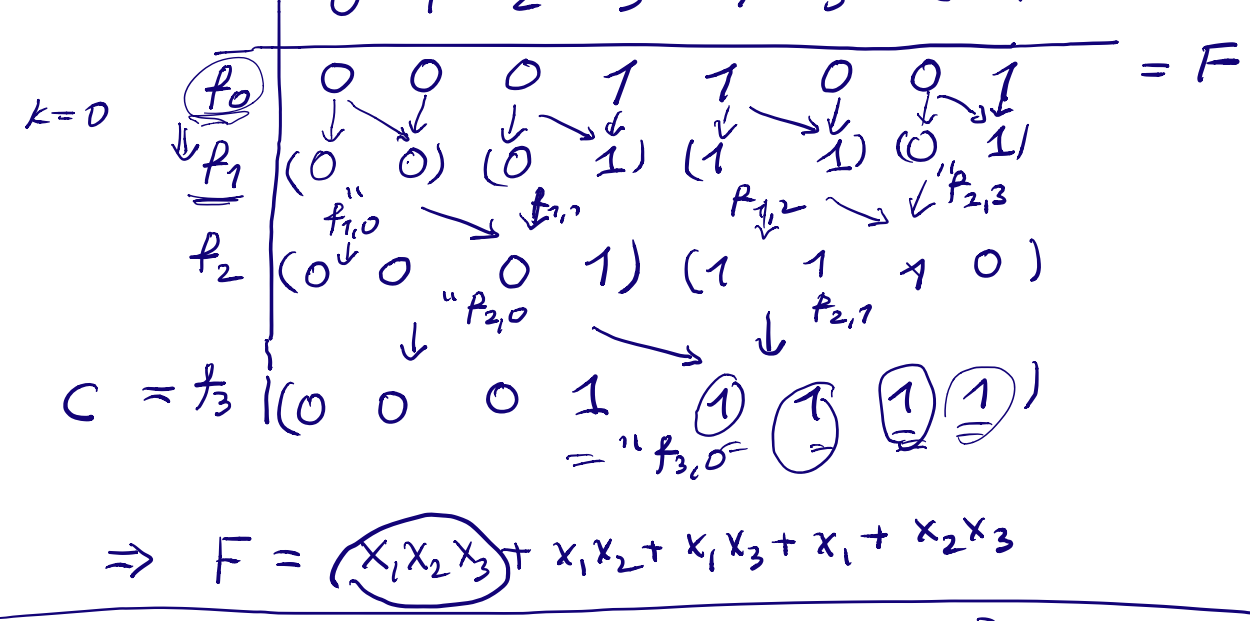
**Complexity.** at stage $k$ $2^k \cdot 2^{n-k-1} = 2^{n-1}$

# stages is $n$

overall # XORs is $n \cdot 2^{n-1}$.

**Example.**

| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |




$$\Rightarrow F = x_1x_2x_3 + x_1x_2 + x_1x_3 + x_1 + x_2x_3$$

---

definitions. Algebraic degree of a Boolean function = deg polynomial

e.g. $\deg F = 3$

if $\deg F = 1 \Rightarrow F$ is called affine

$F(0\ldots0) = 0 \Rightarrow F$ linear.

e.g. $x_1 + x_2 + x_3 + 1$ affine

$x_1 + x_2 + x_3$ linear.

$\deg F = 2$ quadratic

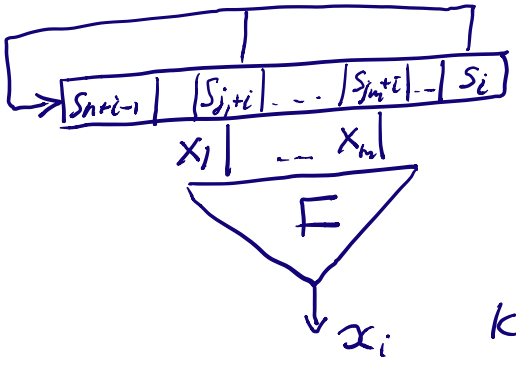
$3$ cubic

---

# Filter Generator

1. LFSR defined by gen. polynomial
$f(x) = x^n + c_1 x^{n-1} + \ldots + c_n$

2. Boolean function $F = F(x_1 x_2 \ldots x_m)$, $m \le n$.



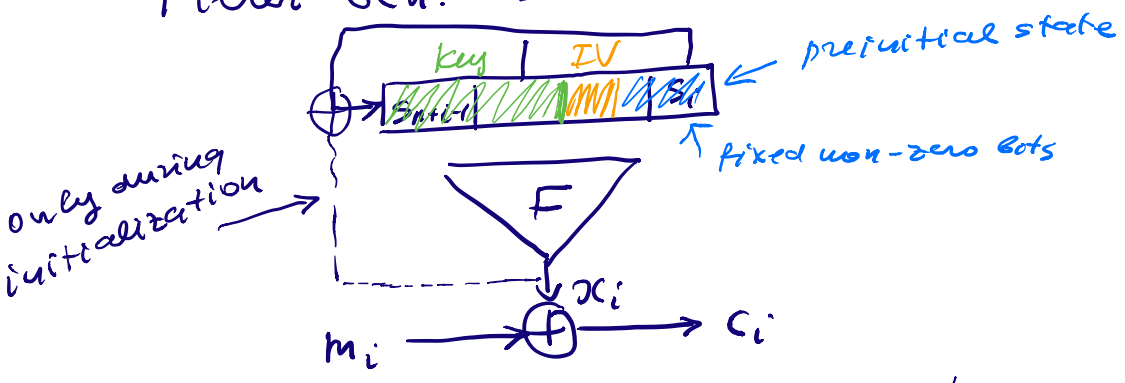

key-stream bit at time $i$

3. output taps $j_1 \ldots j_m$

Boolean f. F is called filtering function.

---

Filter Gen. based stream cipher.




initialization: introduce cipher key and IV into the register

that defines a preinitial state of the LFSR.

initialization works $2 \cdot n\,(3 \cdot n)$ clocks without excrypting data.

register is in initial state

encryption starts.

---

IV sent in clear (without encryption) before cipher-text

---

Cryptanalysis.

known pl-text attack.

$m_i$ and $c_i$ deduce $x_i = m_i \oplus c_i$

1. find cipher key.
2. find initial state
3. model the device and predict the rest of the key-stream.

---

Given $x^N = x_0 x_1 \ldots x_{N-1}$

Recover initial state of the LFSR. $= (s_{n-1} \ldots s_0)$

Brute force try all non-zero initial states (possible!)

generate key-stream

put against available key-stream

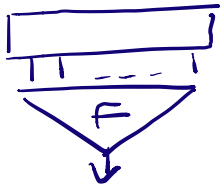
if match ⇒ current state is the cipher initial state.

# trials is $2^n - 1$

---

## Affine Approximation Attack.

$F = F(X_1 \cdots X_n)$




I find affine Boolean function

$g(X_1 \cdots X_n) = a_1 X_1 + \cdots + a_n X_n + b \pmod 2$

s.t. $P_2(F = g) = p$ is max. (relatively high)

$$p = \frac{\#(x_1 \cdots x_n) \text{ s.t. } F(x_1 \cdots x_n) = g(x_1 \cdots x_n)}{2^n}$$

there $2^{n+1}$ affine functions.

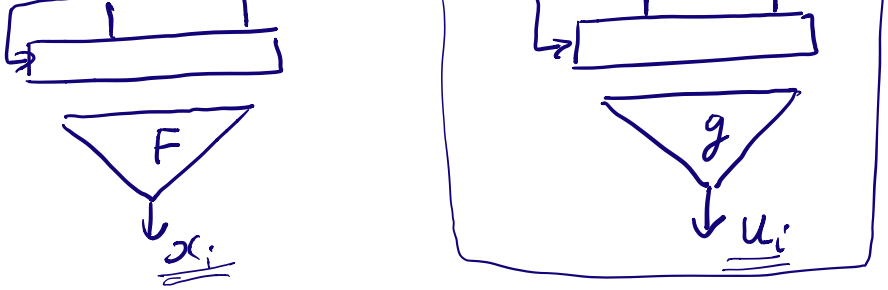
Examples.

| $x_1$ | $x_2$ | $x_3$ | $F$ | $x_1$ | $x_2$ | $x_2+x_3+1$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 ✓ | 0 ✓ | 1 |
| 0 | 0 | 1 | 0 | 0 ✓ | 0 ✓ | 0 ✓ |
| 0 | 1 | 0 | 1 | 0 | 1 ✓ | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 ✓ | 1 ✓ |
| 1 | 0 | 0 | 1 | 1 ✓ | 0 | 1 ✓ |
| 1 | 0 | 1 | 0 | 1 | 0 ✓ | 0 ✓ |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 ✓ |
| 1 | 1 | 1 | 1 | 1 ✓ | 1 ✓ | 1 ✓ |
| | | | $P_2$ | $\frac{4}{8}=\frac{1}{2}$ | $\frac{6}{8}=\frac{3}{4}$ | $\frac{6}{8}=\frac{3}{4}$ |

$2^{3+1} = 16$ affine function

↑ probability of approximation

II




$Pr(x_i = u_i) = Pr(F = g) = p$ relatively high

$u_i = u_i(s_{n-1} \dots s_1 s_0) = a_1 \cdot s_{n+i-1} + a_2 \cdot s_{n+i-2} + \dots + a_n s_i + B =$

↑ linear (affine) function

$= (s_{n+i-1} \dots s_i) \begin{pmatrix} a_1 \\ \dots \\ a_n \end{pmatrix} + B$

state at time $i$

$= \underbrace{(s_{n-1} \dots s_0)}_{\text{unknowns}} A^i \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} + B$ where $A = \begin{pmatrix} c_1 & 1 & 0 \\ \vdots & & \ddots \\ & & 1 \\ c_n & 0 & 0 \end{pmatrix}$

affine (lin.) function of $i$, initial state bits

companion matrix to LFSR gen. polyn. $X^n + c_1 X^{n-1} + \dots + c_n$.

III introduce new variables

$$(*) \begin{cases} v_0 = u_0 \overset{+}{\oplus} x_0 \\ v_1 = u_1 + x_1 \\ \dots \\ v_{N-1} = u_{N-1} + x_{N-1} \end{cases}$$

$Pr(v_i = 0) = Pr(u_i = x_i) = p$ rel. high.

$Pr(v_i = 1) = Pr(u_i \neq x_i) = q = 1 - p$. rel. low

use a subsystem of $(*)$ fix some $n_1 \gg n$ $(n_1 = n+2)$

$$(**) \begin{cases} v_0 = u_0 + x_0 \\ \dots \\ v_{n_1 - 1} = u_{n_1 - 1} + x_{n_1 - 1} \end{cases}$$

$v_{n_1+i} = u_{n_2+i} + x_{u_i+1}$

we want to solve (**)

# equations is $n_1$

# var. is $n_1+n$

$\Rightarrow$ # solutions $2^{(n_1+n)-n_1} = 2^n$

we distribution of $v_i$

$v = (v_0 v_1 \dots v_{n_1-1})$

$z = \text{weight}(v) = v_0 + v_1 + \dots + v_{n_1-1}$ (+ of integers)

# 1 in $v$.

$P_z(v_i = 1) = q \Rightarrow z \approx q n_1$

math. expectation (mean) of $z$

$Ez = E(v_0) + E(v_1) + \dots + E(v_{n_1-1}) = n_1 q.$

$E(v_i) = 1 \cdot q + 0 \cdot p = q$

Since $q$ is rel. small $\Rightarrow z \approx q n_1$ rel. small.

1) try all $v$ of weight $\approx q \cdot n_1$
(of weight $z$ s.t. $|z - q n_1| \le d$ parameter)

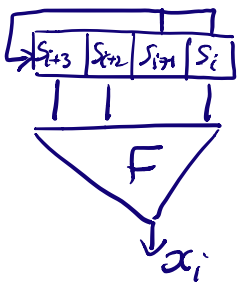
2) solve system of linear equations (**)

3) solution is an possible initial state check that with the rest of the key-stream.

---

Example.

$x^4 + x + 1$

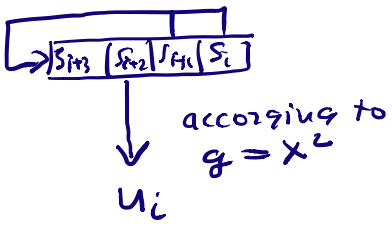



| $x_1 x_2 x_3$ | F |
|---|---|
| ... | 0 |
| | 0 |
| | 1 |
| | 1 |
| | 1 |
| | 0 |
| | 0 |
| | 1 |

given key-stream $x^{12} = 011000101110$

find initial state $(s_3 s_2 s_1 s_0)$

$P_2(F = x_2) = \frac{3}{4} \Rightarrow$ set $g = x_2 \Rightarrow$




$n_1 = 5 > n = 4$

$u_0 = s_2$
$u_1 = s_3$
$u_2 = s_0 + s_1$
$u_3 = s_1 + s_2$
$u_4 = s_2 + s_3$

new variables

(**)

$$\begin{cases} v_0 = s_2 \\ v_1 = s_3 + 1 \\ v_2 = s_0 + s_1 + 1 \\ v_3 = s_1 + s_2 \\ v_4 = s_2 + s_3 \end{cases}$$

weight $(v_0 v_1 \ldots v_4) \approx \frac{1}{4} \cdot 5 = \frac{5}{4}$

$v'$

try all vectors $v$ of weight $0, 1, 2, \ldots$

1) $v = (00\ldots0)$

$$\begin{cases} 0 = s_2 \\ 0 = s_3 + 1 \\ 0 = s_0 + s_1 + 1 \\ 0 = s_1 + s_2 \\ 0 = s_2 + s_3 \end{cases} \Rightarrow \text{no solutions}$$

$0 = 1$

$0 = s_2 + s_3 \Rightarrow s_3 = 0$

2) $v = (10000)$

$$\begin{cases} 1 = s_2 \\ 0 = s_3 + 1 \\ 0 = s_0 + s_1 + 1 \\ 0 = s_1 + s_2 \\ 0 = s_2 + s_3 \end{cases} \Rightarrow (s_3 s_2 s_1 s_0) = (1110)$$

use as initial state generate key-stream:

$0110011000, \ldots$

$01100\,\overset{\#}{0}$ available.

guess wrong.

3) $v = (01000)$

$$\begin{cases} 0 = s_2 \\ 1 = s_3 + 1 \\ 0 = s_0 + s_1 + 1 \\ 0 = s_1 + s_2 \\ 0 = s_2 + s_3 \end{cases} \Rightarrow (s_3 s_2 s_1 s_0) = (0001)$$

generate key-stream $\equiv$ given

$\Rightarrow$ 0001 is correct LFSR initial state.