Attacking the combination generator

Yann Laigle-Chapuy

INRIA - Équipe SECRET / Université Pierre et Marie Curie

Yann.Laigle-Chapuy@inria.fr

VERIMAG– Séminaire de cryptologie, codage et infrastructures sécurisées

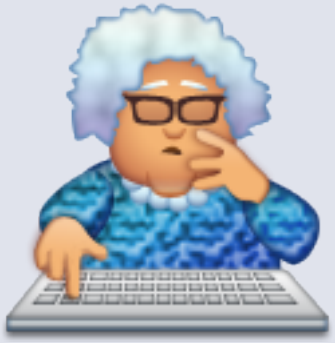
joint work with Frédéric Didier

June 16th 2009

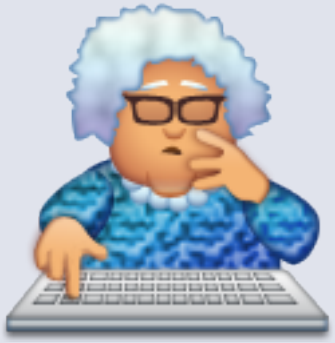
INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Stream ciphers

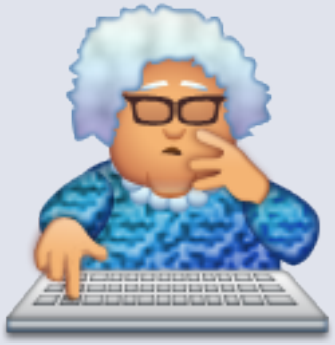


Stream ciphers



. . . 0100101001111010010010101111 0 . . .

Stream ciphers



. . . 01001010011110100100101011110 . . .

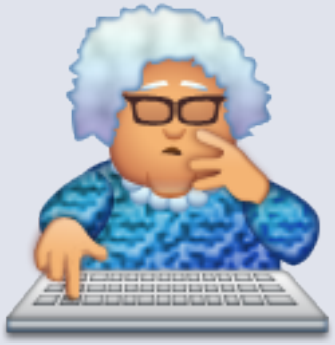
. . . 11001000100011111110001001011 . . .

. . . 10000010111101011010100010101 . . .

INRIA ROCQUENCOURT

UPMC PARISUNIVERSITAS

Stream ciphers



. . . 01001010011110100100101011110 . . .

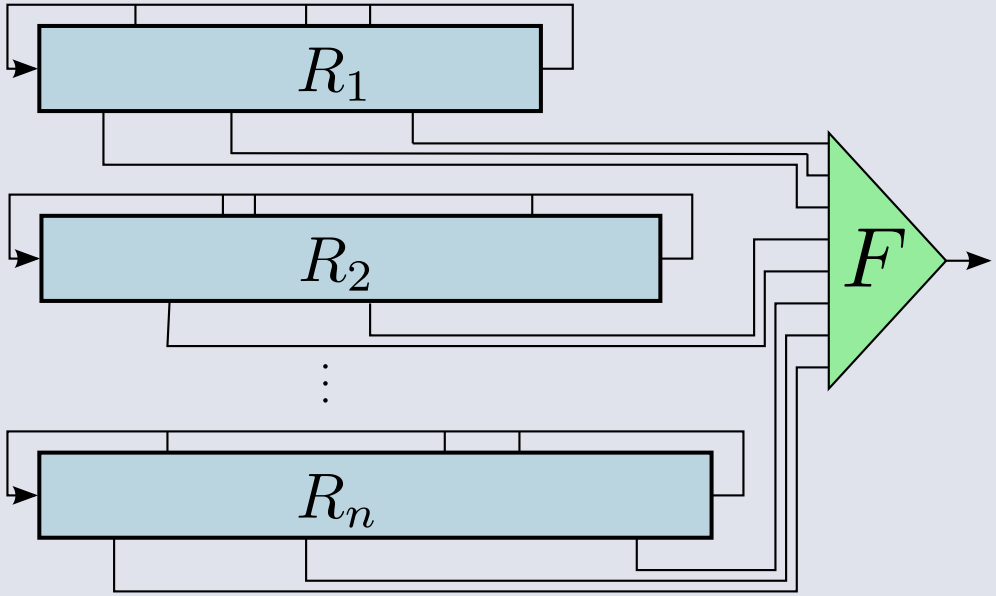
0x171980 . . . 11001000100011111110001001011 . . . 0x171980

. . . 10000010111101011010100010101 . . .

INRIA ROCQUENCOURT

UPMC PARISUNIVERSITAS

The combination generator



- LFSRs combined by a Boolean fonction

$$F : \{0,1\}^n \to \{0,1\}$$

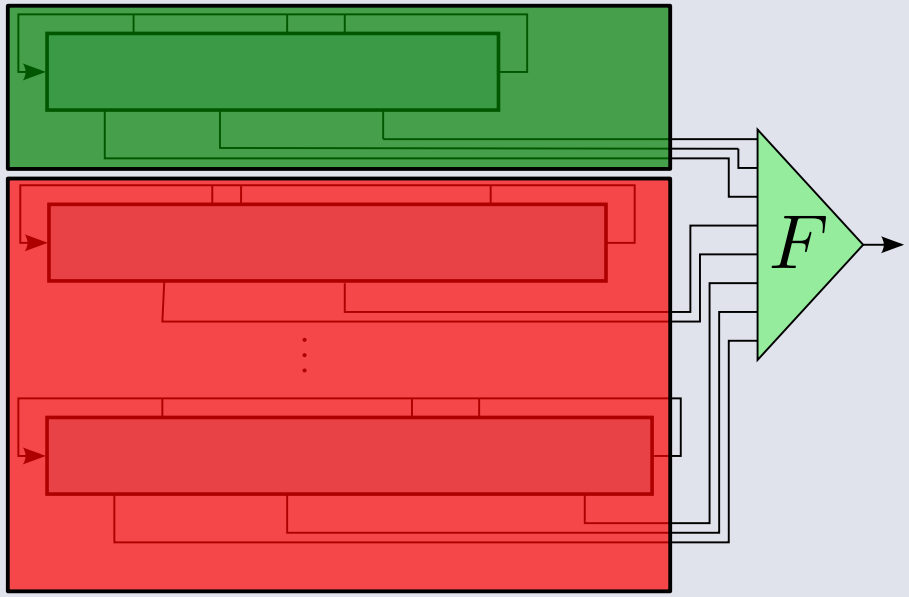
Outline

- Correlation attacks
  - correlation and fast correlation attacks
  - Our new attack
- Computing low weight multiples
- Experimental results

Correlation attacks

The combination generator

Attack Model



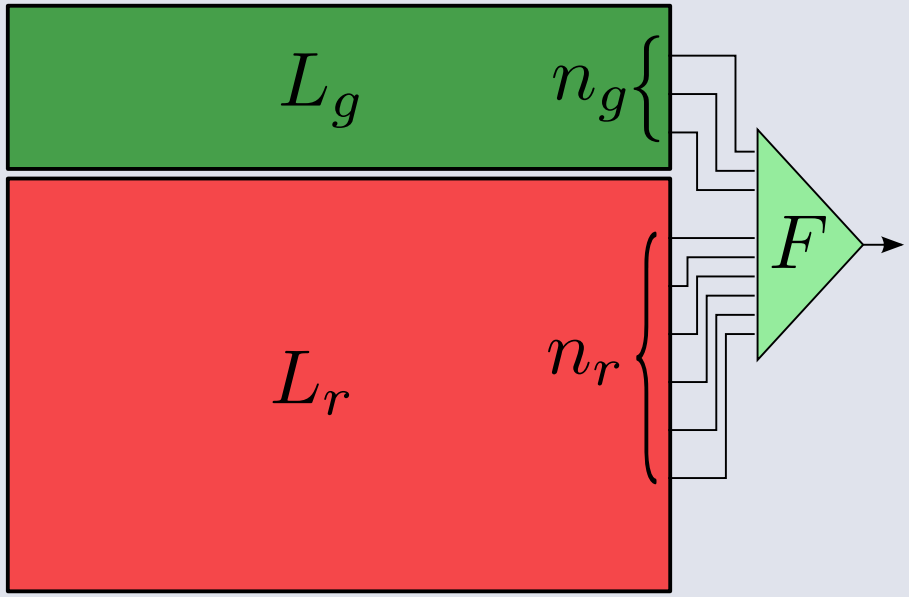
- LFSRs combined by a Boolean fonction

$$F : \{0,1\}^n \to \{0,1\}$$

Goal : find the initial state of the registers

The combination generator

Attack Model



- LFSRs combined by a Boolean fonction

$$F : \{0,1\}^{n_g+n_r} \rightarrow \{0,1\}$$

Goal : find the initial state of the green register

Correlation attacks

Siegenthaler 1985

Assume

$$Pr[F(x_1, \ldots, x_n) = x_1] \neq \frac{1}{2}$$

Attack :

Repeat

- guess the initial state of the first LFSR
- generate the sequence $(x_1^t)_{t \in [1..N]}$
- compute the probability to decide if our guess was good

until success...

Correlation attacks

Siegenthaler 1985

If F is not g-th order correlation-immune

$$Pr[F(x_1, \ldots, x_n) = x_1 + \cdots + x_{n_g}] \neq \frac{1}{2}$$

(see [CT00])

Attack :

Repeat

- guess the initial state of the green LFSRs
- generate the sequences $(x_1^t)_{t\in[1..N]}, \ldots, \left(x_{n_g}^t\right)_{t\in[1..N]}$
- compute the probability to decide if our guess was good

until success...

Fast correlation attacks

Meier Staffelbach 1988

Use the linearity of the LFSRs :

$u = x_1 + \cdots + x_{n_g}$ depends linearly on the initial state of the registers

Idea :

- Consider the keystream as a word in the corresponding linear code, plus some (randomish) error
- Use fast algorithm to decode

Fast correlation attacks

Meier Staffelbach 1988

Use the linearity of the LFSRs :

$u = x_1 + \cdots + x_{n_g}$ depends linearly on the initial state of the registers

Idea :

- Consider the keystream as a word in the corresponding linear code, plus some (randomish) error
- Use fast algorithm to decode $\longrightarrow$ e.g. LDPC

We need Low Density Parity Check equations
$\longrightarrow$this mean finding low weight multiples of the feedback polynomial

Our new attack

The idea

- Precomputation : Find weight 4 multiples of the `lcm` of the "red" feedback polynomials
- Try to guess the initial state of the green part
- Select instants where the inputs sum up to 0 i.e.

$$\sum \boldsymbol{x}^{t_i} = \left(\frac{0}{0}\right)$$

- Compute for those tuples

$$\Pr\left[\sum F(\boldsymbol{x}^{t_i}) = 0\right]$$

- If our guess is good, we should observe a bias

Our new attack

the explanation

If the guess is wrong we compute

$$\Pr\left[\sum F(\boldsymbol{x}^{t_i}) = 0 \middle| \sum \boldsymbol{x}^{t_i} = \begin{pmatrix} \star \\ 0 \end{pmatrix}\right]$$

If the guess is good we compute

$$\Pr\left[\sum F(\boldsymbol{x}^{t_i}) = 0 \middle| \sum \boldsymbol{x}^{t_i} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right]$$

Our new attack

the explanation

Theorem 1 (Leveiller - Didier).

Let F be a balanced n-variable Boolean function and

$$P_{\boldsymbol{u}} = \Pr[F(\boldsymbol{u}_1)+F(\boldsymbol{u}_2)+F(\boldsymbol{u}_3)+F(\boldsymbol{u}_4) = 0 | \boldsymbol{u}_1+\boldsymbol{u}_2+\boldsymbol{u}_3+\boldsymbol{u}_4 = \boldsymbol{u}]$$

$$P_{\vec{0}} \geq \frac{1}{2} + \frac{1}{2^{n+1}}$$

and

$$\min_{\boldsymbol{u}\neq\vec{0}} \left(P_{\vec{0}} - P_{\boldsymbol{u}}\right) \geq \frac{1}{2^{n+1}} \left(1 - \frac{\Delta_f}{2^n}\right)^2$$

where Δ_f is the maximum of the autocorrelation coefficients of f and is usually small compared to 2^n.

INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Our new attack

- **Big improvement** : using a Walsh transform, we can compute the propabilities obtained for each initial state with complexity $\mathcal{O}(N + L_g 2^{L_g})$ instead of $\mathcal{O}(N 2^{L_g})$ where N is the keystream needed
  - one input : classical result for maximal likelihood decoding long linear codes of small dimension
  - many inputs : use the linearity of the Walsh transform and

$$\#\{i, \boldsymbol{x}^{t_i} = \vec{0}\} = \frac{\sum_{y \in \mathbb{F}_2^{n_g}} \#\{i, \boldsymbol{x}^{t_i} \cdot y = 0\} - 2^{n_g - 1}}{2^{n_g - 1}}$$

Walsh transform

$$(u|v) \rightarrow (u+v|u-v)$$

x	000	001	010	011	100	101	110	111
$f(x)$	0	1	1	0	1	0	0	0
$(-1)^{f(x)}$	1	-1	-1	1	-1	1	1	1
	0	2	0	-2	0	-2	2	0
	0	0	0	4	2	-2	-2	-2
$walsh(f)(x)$	2	-2	-2	2	-2	2	2	6
$x \cdot 111$	0	1	1	0	1	0	0	1

Our new attack

Summary

- **Complexity**
  - keystream needed $\mathcal{O}\left((L_g 2^{L_r+2n+n_g})^{1/4}\right)$
  - total online time $\mathcal{O}\left(L_g 2^{L_g} + 2^{2(n+n_g)}\right)$

- **Key feature** No restriction on the correlation immunity of the function. It can even be unknown to the attacker.

Finding low weight multiple

Finding low weight multiple

Survey

Almost the same as finding a low weight word of a linear code of parity check matrix

$$H = \left(X^0 \pmod P, \ldots, X^D \pmod P\right)$$

$degree(P)$ rows, D columns

Relevant parameters :

- the weight we look for
- the maximum degree D
- the number of distincts multiples we need

Finding low weight multiple

Survey

Many different algorithms depending on the constraints :

- Canteaut Chabaud
- Chose Joux Mitton
- Brouwer Zimmermann
- Wagner

Finding low weight multiples

Finding low weight multiple

Survey

Many different algorithms depending on the constraints :

- Canteaut Chabaud
- Chose Joux Mitton
- Brouwer Zimmermann
- Wagner

Fast correlation attack :

- very low fixed weight (4 or 5)
- high degree (typically 2^{30})
- all the available multiples (to minimize the keystream needed)

Chose Joux Mitton algorithm

$$Q(X) = 1 + \left(X^{i_1} + \cdots + X^{i_{\lfloor (w-1)/2 \rfloor}}\right) + \left(X^{i_{\lceil (w-1)/2 \rceil}} + \cdots + X^{i_{w-1}}\right)$$

- Store $\alpha_k = \sum_{i=1}^{\lfloor (w-1)/2 \rfloor} X^{k_i} \mod P$
- Find $\beta_\ell = \sum_{i=1}^{\lceil (w-1)/2 \rceil} X^{\ell_i} \mod P$ such that $\beta_\ell = \alpha_k + 1$

Complexity : memory $D^{\lfloor (w-1)/2 \rfloor}$, time $D^{\lceil (w-1)/2 \rceil}$

INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Chose Joux Mitton algorithm

$$Q(X) = 1 + \left(X^{i_1} + \cdots + X^{i_{\lfloor (w-1)/2 \rfloor}}\right) + \left(X^{i_{\lceil (w-1)/2 \rceil}} + \cdots + X^{i_{w-1}}\right)$$

- Store $\alpha_k = \sum_{i=1}^{\lfloor (w-1)/2 \rfloor} X^{k_i} \mod P$
- Find $\beta_\ell = \sum_{i=1}^{\lceil (w-1)/2 \rceil} X^{\ell_i} \mod P$ such that $\beta_\ell = \alpha_k + 1$

Additional trick : enumerate the α_k and β_ℓ wisely

Complexity : memory $D^{\lceil (w-1)/4 \rceil}$, time $D^{\lceil (w-1)/2 \rceil}$

But, CJM does not use the multiplicative structure of our problem

INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Using discrete logarithm

[ISIT07]

$$Q(X) = \left(1 + X^{i_1} + \cdots + X^{i_{\lfloor (w-2)/2 \rfloor}}\right) + X^L \left(1 + X^{i_{\lceil (w-2)/2 \rceil}} + \cdots + X^{i_{w-2}}\right)$$

- Store *logarithms* of $\alpha_k = 1 + \sum_{i=1}^{\lfloor (w-2)/2 \rfloor} X^{k_i}$
- Find $\beta_\ell = 1 + \sum_{i=1}^{\lceil (w-2)/2 \rceil} X^{\ell_i}$

  such that

$$k_{\lfloor (w-1)/2 \rfloor} - D \leq \overbrace{\log(\beta_\ell) - \log(\alpha_k)}^{L} \leq D - \ell_{\lceil (w-1)/2 \rceil}$$

Complexity

General comparison

	$w = 2p$		$w = 2p+1$	
Algorithm	Time	Memory	Time	Memory
LogTMTO	$D^{p-1}L$	$D^{p-1}L$	D^pL	$D^{p-1}L$
CJM	D^p	$D^{\lceil p/2 \rceil}$	D^p	$D^{\lceil p/2 \rceil}$

Complexity

Small weights

	$w=4$		$w=5$		$w=6$	
Algorithm	Time	Memory	Time	Memory	Time	Memory
LogTMTO	DL	DL	D^2L	DL	D^2L	D^2L
CJM	D^2	D	D^2	D	D^3	D

In order to find weight 4 multiples, we need $D \sim 2^{d/3}$ (with $d = degree(P)$)

There exists sub-exponential algorithms for the discrete logarithm over finite fields, therefore it is asymptoticaly faster

Experimental results

Experimental results

- 3 registers of size 29, 31 and 37
- 9 variable 3 resilient Boolean function with 3 inputs in each register

Attack 1 : 29,31,37 Attack 2 : 37,31,29

	Precomputation	Total online time	Keystream used
Attack 1	12min27s	7min01s	3.06MB
Attack 2	3min02s	6h15min	985KB

	1st LFSR	2nd LFSR	3rd LFSR
Attack 1	51s	6min10s	0s
Attack 2	6h17min	1min27s	0s

Bibliography

- [Sieg85] Thomas Siegenthaler, *Decrypting a Class of Stream Ciphers Using Ciphertext Only*, IEEE Trans. Computers, 1985
- [MS88] Willi Meier and Othmar Staffelbach, *Fast Correltaion Attacks on Stream Ciphers (Extended Abstract)*, EUROCRYPT, 1988
- [CT00] Anne Canteaut and Michaël Trabbia, *Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5*, EUROCRYPT, 2000
- [CJM02] Philippe Chose and Antoine Joux and Michel Mitton, *Fast Correlation Attacks : An Algorithmic Point of View*, EUROCRYPT, 2002
- [Lev04] Sabine Leveiller, *Quelques algorithmes de cryptanalyse du registre filtré*, PhD, 2004

— The end —

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$\boxed{0\ 1\ 0\ 1\ 0\ 0\ 1}\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

α

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ \underbrace{1\ 0\ 1\ 0\ 0\ 1\ 0}_{\alpha x}\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ \underbrace{0\ 1\ 0\ 0\ 1\ 0\ 1}_{\alpha x^2}\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ \underbrace{1\ 0\ 0\ 1\ 0\ 1\ 0}_{\alpha x^3}\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ 1\ \underbrace{0\ 0\ 1\ 0\ 1\ 0\ 1}_{\alpha x^4}\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ 1\ 0\ \underbrace{0\ 1\ 0\ 1\ 0\ 1\ 0}_{\alpha x^5}\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ 1\ 0\ 0\ \underbrace{1\ 0\ 1\ 0\ 1\ 0\ 1}_{\alpha x^6}\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ 1\ 0\ 0\ 1\ \underbrace{0\ 1\ 0\ 1\ 0\ 1\ 1}_{\alpha x^7}\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

$$0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ \underbrace{1\ 0\ 1\ 0\ 1\ 1\ 1}_{\alpha x^8}\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

INRIA ROCQUENCOURT

UPMC PARIS UNIVERSITAS

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$

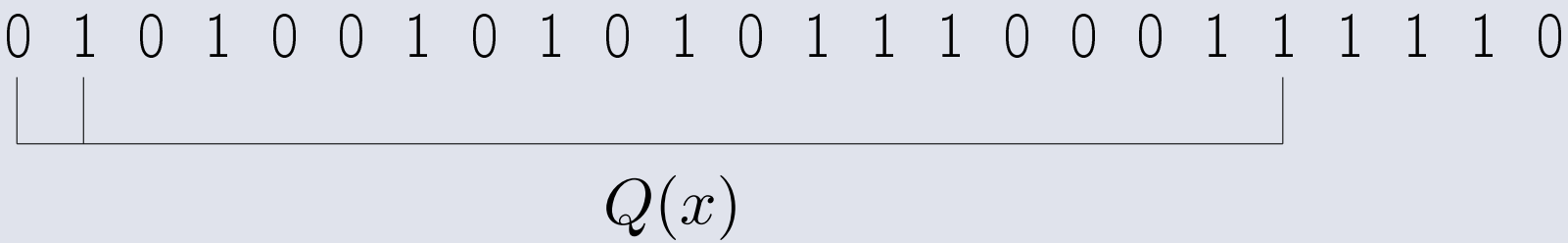
$$0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ \underbrace{0\ 1\ 0\ 1\ 1\ 1\ 0}_{\alpha x^9}\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$



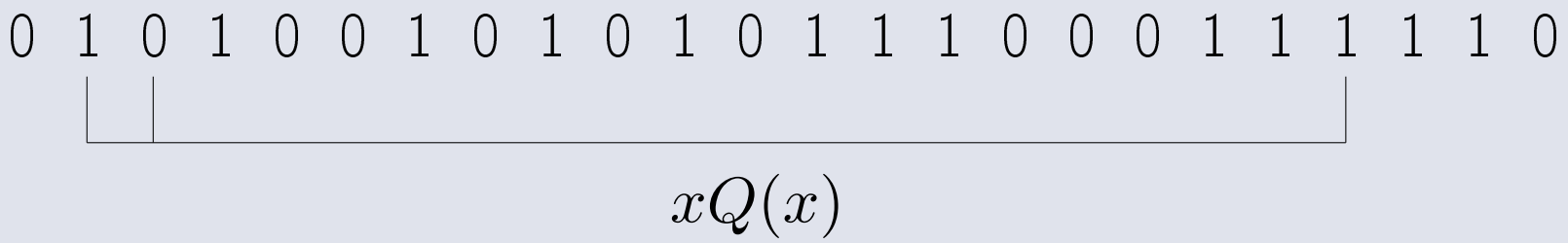
with $Q(x) = x^{19} + x + 1$ a multiple of $P(x)$.

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$



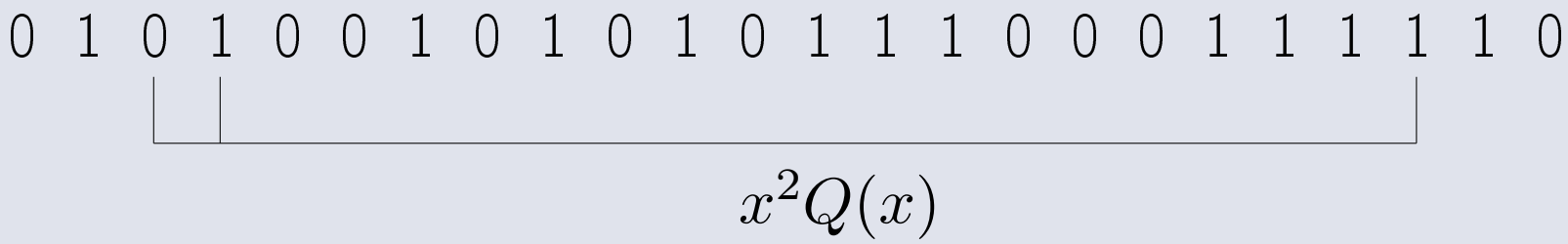
with $Q(x) = x^{19} + x + 1$ a multiple of $P(x)$.

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$



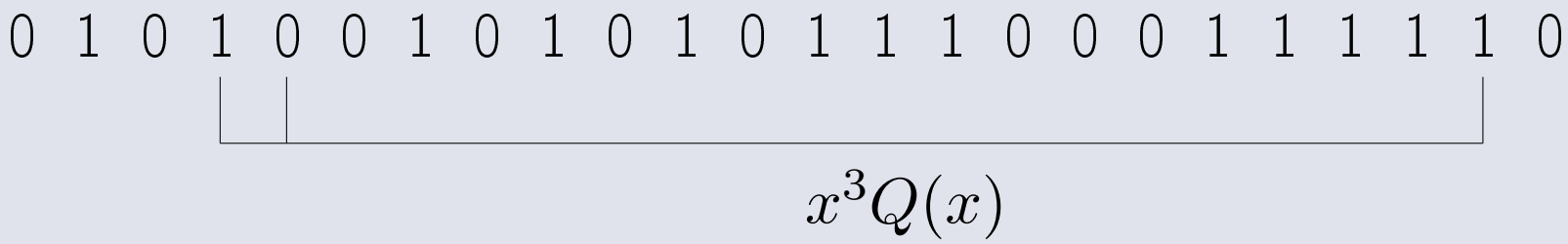
with $Q(x) = x^{19} + x + 1$ a multiple of $P(x)$.

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$



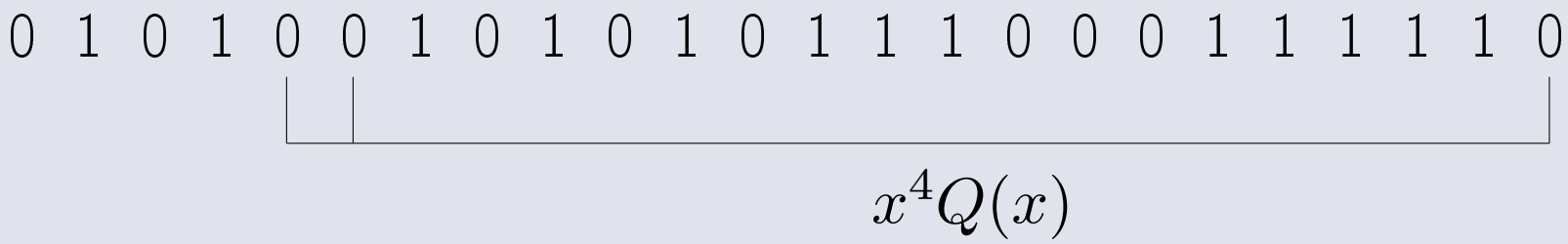
with $Q(x) = x^{19} + x + 1$ a multiple of $P(x)$.

Low weight multiples

What for ?

LFSR with feedback polynomial

$$P(x) = x^7 + x^5 + x^4 + x^3 + x^2 + x + 1$$



with $Q(x) = x^{19} + x + 1$ a multiple of $P(x)$.