Cryptanalysis of Full Sprout

CRYPTO 2015

Virginie Lallemand and María Naya-Plasencia

Inria, France

August 20th, 2015

New Design Principle

Frederik Armknecht, Vasily Mikhalev
On Lightweight Stream Ciphers with Shorter Internal States,
FSE 2015

Background: Lightweight Cryptography

Important metrics:

- area size
- throughput
- memory
- power consumption

Background: Lightweight Cryptography

Important metrics:

- area size
- throughput
- memory
- power consumption

Block or Stream ciphers?

Throughput generally better for stream ciphers, area size generally worse

Background: Sprout Motivation

Common Design Rule of Stream Ciphers

To avoid time-memory-data trade-off attacks, the internal state size of a stream cipher should be at least twice the security parameter

$\rightarrow$ Larger area size, bigger power consumption

Objective of the new stream cipher design:

Reduce the size of the internal state used in stream ciphers while offering resistance to time-memory-data trade-off attacks

New Design Principle

- Sufficiently large internal state
- Part of this state is the secret key itself
- Involve the secret key not only in the initialization process but also in the keystream generation phase

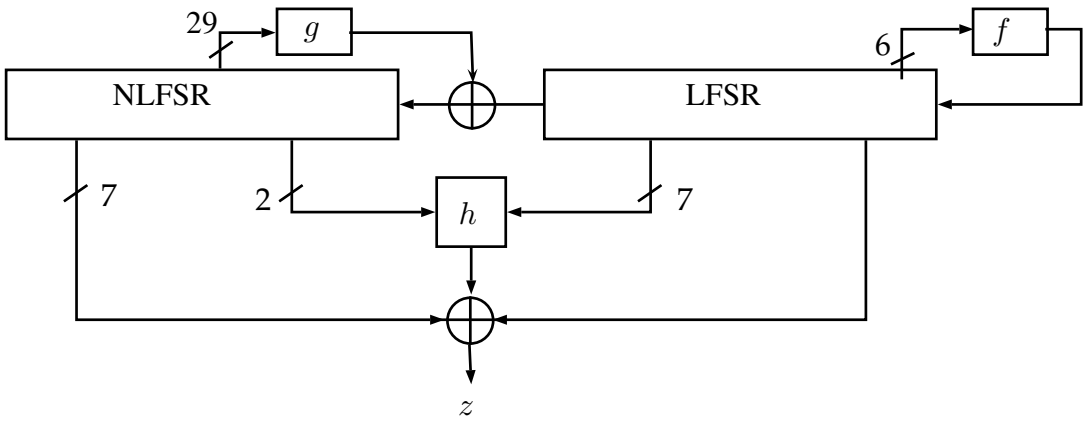
Remark

Storing a fixed key is significantly less area consuming than deploying a register of the same length

Concrete instantiation of this new design strategy: **Sprout**

Description of Sprout

Grain Family

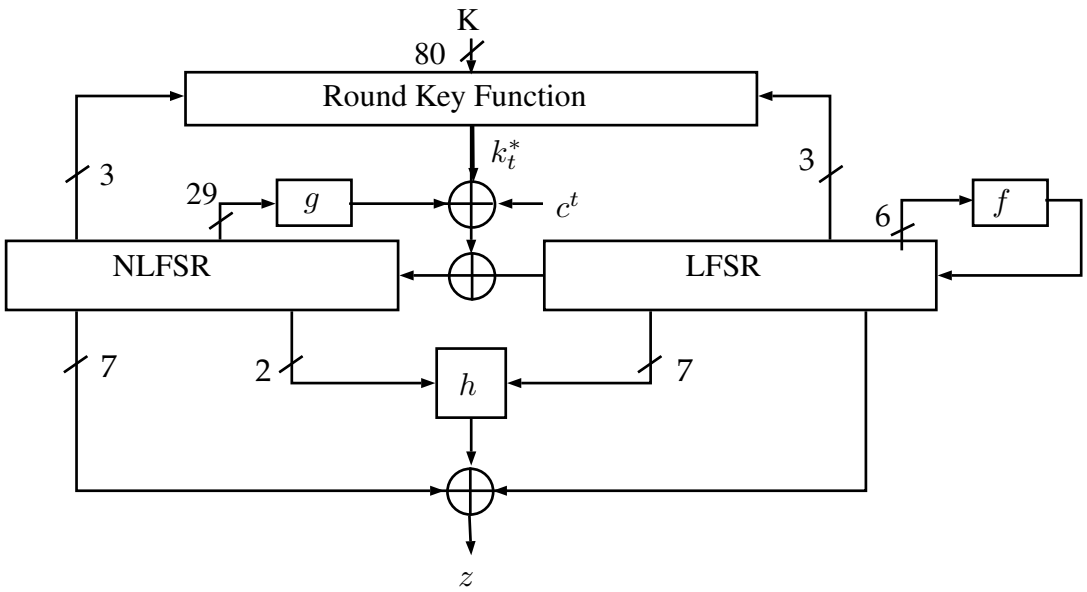


Keystream Generation Phase of Grain-128

Grain v1
80-bit LFSR + 80-bit NLFSR
80-bit key

Grain 128 / Grain 128a
128-bit LFSR + 128-bit NLFSR
128-bit key

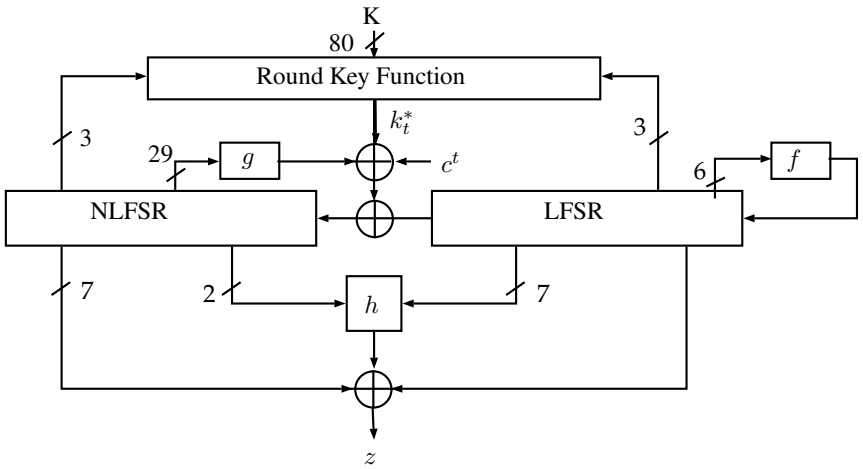
Sprout



Keystream Generation Phase of Sprout

40-bit LFSR + 40-bit NLFSR
80-bit key

Sprout - Keystream Generation Phase



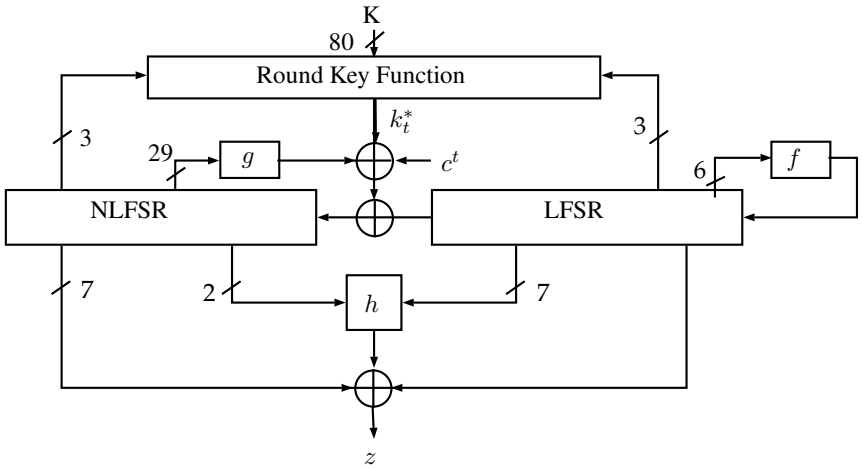
Feedback function of the **LFSR** (maximal period)

$$l_{39}^{t+1} = f(L^t) = l_0^t + l_5^t + l_{15}^t + l_{20}^t + l_{25}^t + l_{34}^t$$

Feedback function of the **NLFSR**

$$\begin{aligned} n_{39}^{t+1} &= n_0^t + n_{13}^t + n_{19}^t + n_{35}^t + n_{39}^t + n_2^t n_{25}^t + n_3^t n_5^t + n_7^t n_8^t + n_{14}^t n_{21}^t \\ &\quad + n_{16}^t n_{18}^t + n_{22}^t n_{24}^t + n_{26}^t n_{32}^t + n_{33}^t n_{36}^t n_{37}^t n_{38}^t + n_{10}^t n_{11}^t n_{12}^t + n_{27}^t n_{30}^t n_{31}^t + c^t + l_0^t + k_t^* \\ &= g(N^t) + c^t + l_0^t + k_t^* \end{aligned}$$

Sprout - Keystream Generation Phase



Feedback function of the **NLFSR**

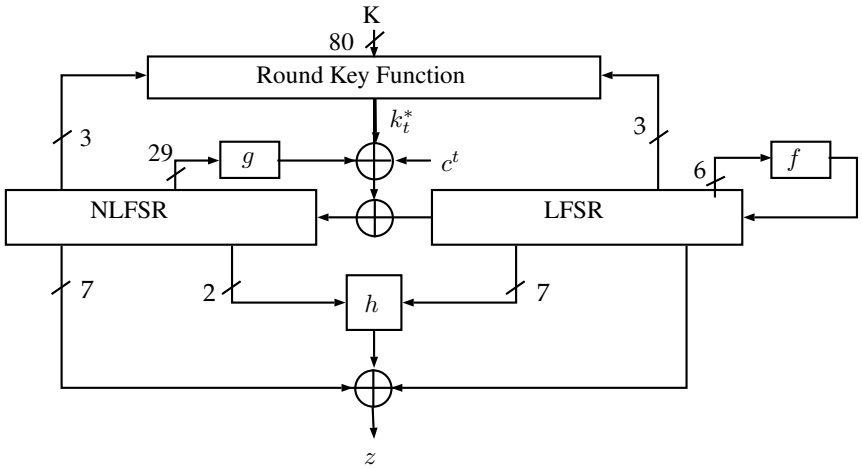
$$n_{39}^{t+1} = g(N^t) + c^t + l_0^t + k_t^*$$

with:

$$k_t^* = k_t, 0 \leq t \leq 79$$

$$k_t^* = (k_{t \bmod 80}) \times (l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t), t \geq 80$$

Sprout - Keystream Generation Phase



Feedback function of the **NLFSR**

$$n_{39}^{t+1} = g(N^t) + c^t + l_0^t + k_t^*$$

with:

$$k_t^* = k_t, 0 \leq t \leq 79$$

$$k_t^* = (k_{t \bmod 80}) \times (l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t), t \geq 80$$

Generation of the keystream bit:

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

Our Attack

Overview

Model:

Known Plaintext Scenario: plaintext and ciphertext are known, so we know the value of some keystream bits

We exploit:

- the short size of the LFSR and NLFSR register w.r.t the key
- the little dependencies between the LFSR and NLFSR registers
- the non linear influence of the key bits in the update function

Overview

3 main steps:

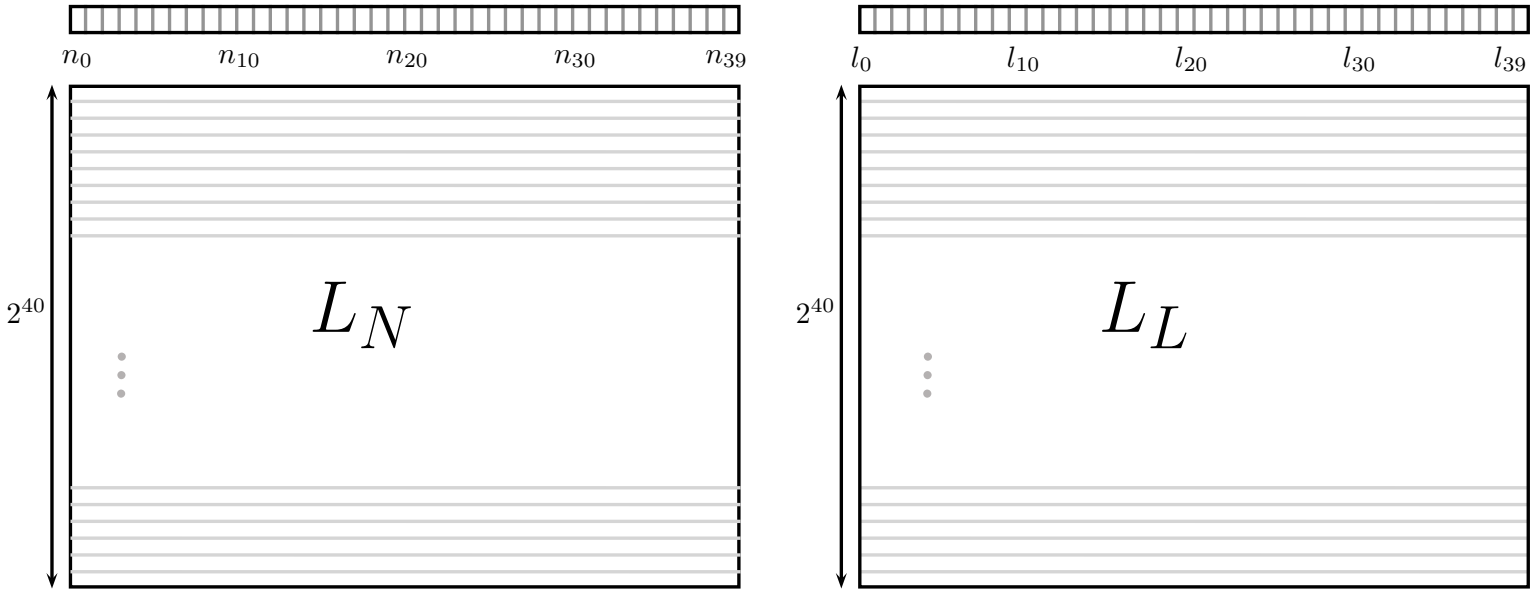
1. Build and arrange two independent lists :
   - the possible internal states for the LFSR at time t
   - the possible internal states for the NLFSR at time t
2. Merge them with the help of some keystream bits
3. Recover the whole key

$\rightarrow 2^{10}$ times faster than exhaustive search

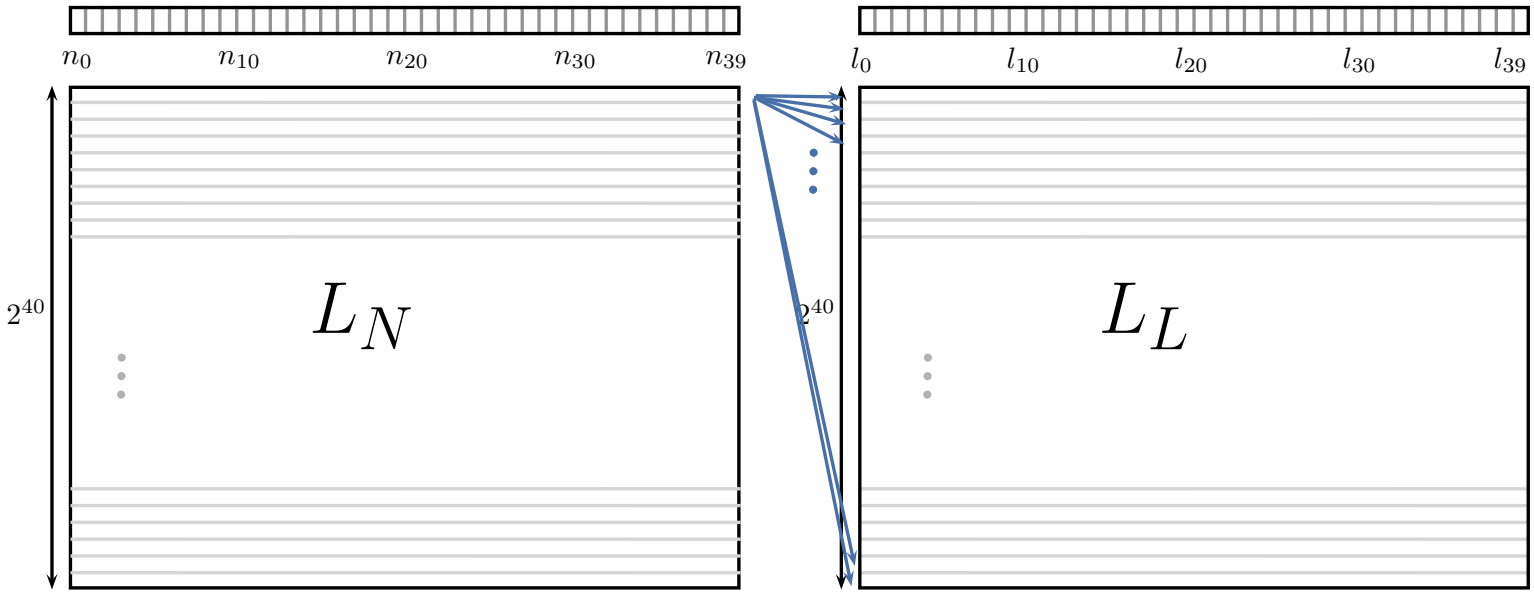
Overview

1. Build and arrange two independent lists :
   - the possible internal states for the LFSR at time t
   - the possible internal states for the NLFSR at time t
2. Merge them with the help of some keystream bits

Step 1 and 2



Step 1 and 2



Question: How to perform an efficient merge that provides a limited set of candidates for both states?

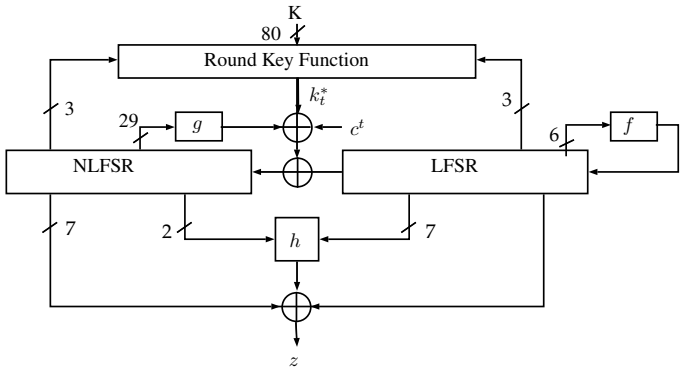
How to Merge

Remark

Property of the linear register

The linear register state is totally independent from the rest during the keystream generation phase so its value can be determined at any time

$$I_{39}^{t+1} = I_0^t + I_5^t + I_{15}^t + I_{20}^t + I_{25}^t + I_{34}^t$$



Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

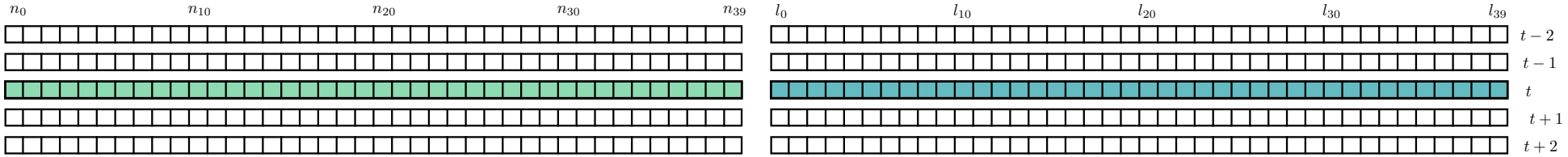
$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

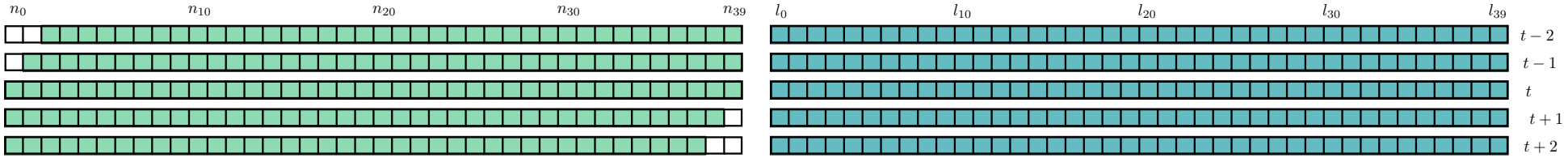


Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

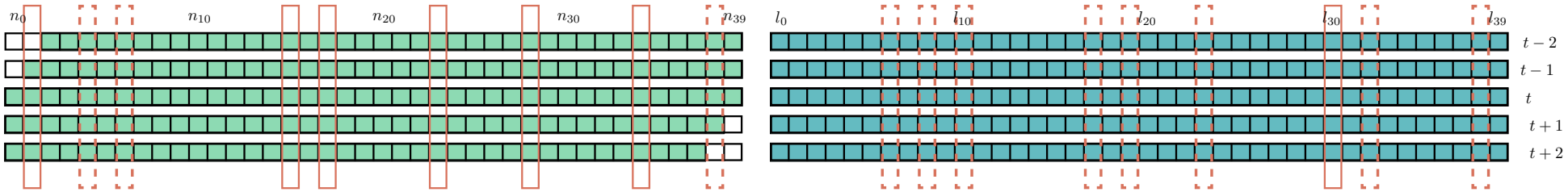


Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

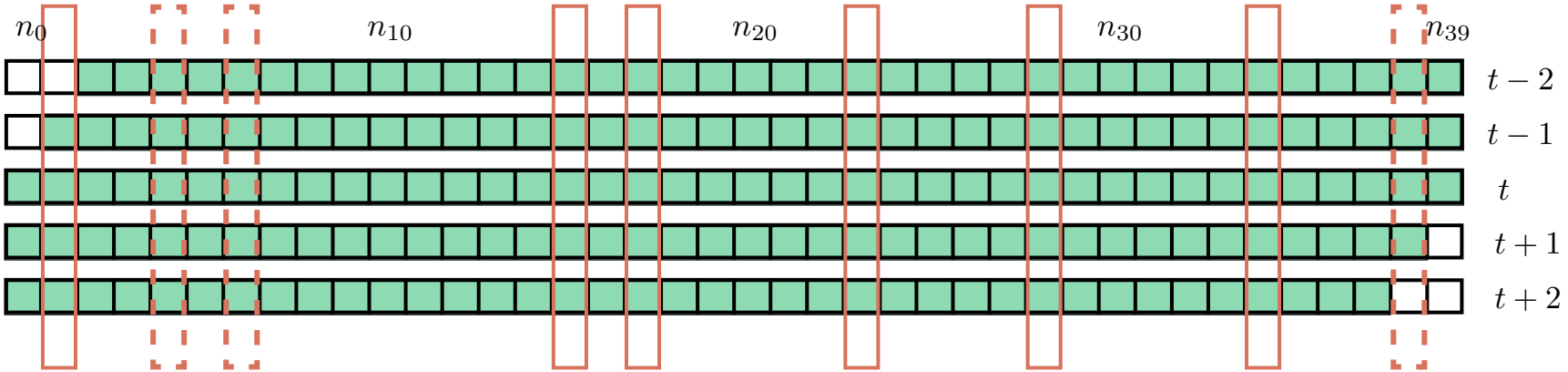


Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

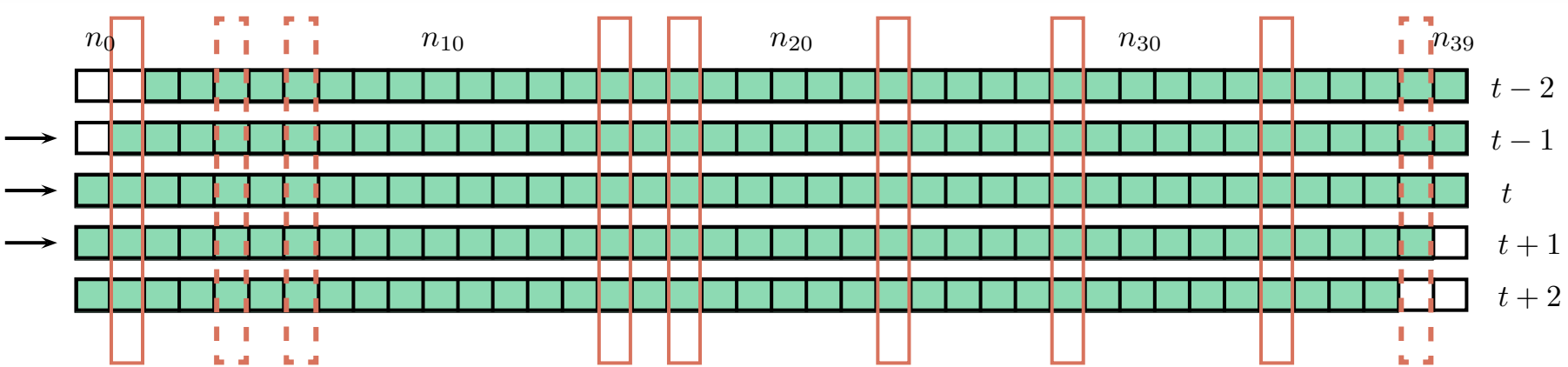


Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

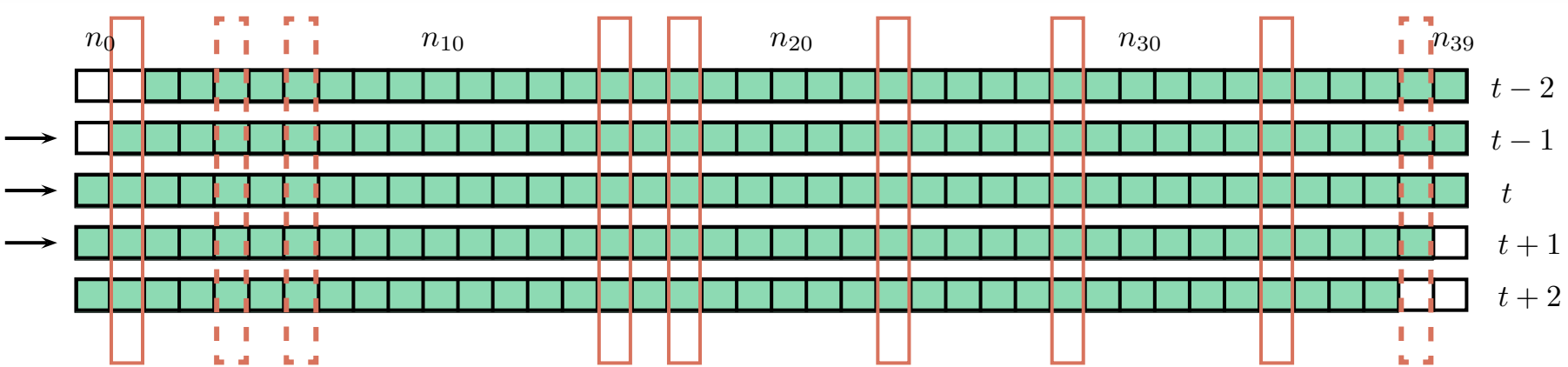


Merging Conditions - Type I and Type II filters

Given an internal state candidate for the NLFSR and for the LFSR at time t

Main idea: Compute z from the candidate and compare it with the actual keystream bit

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$



$\rightarrow$ check equality at time t, $t+1$ and $t-1$

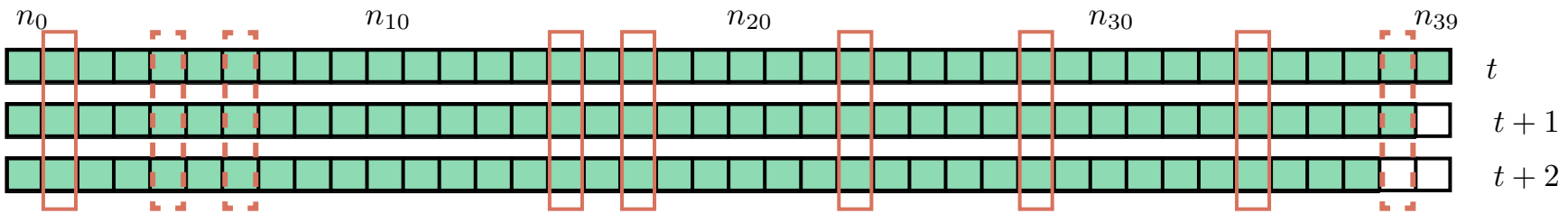
$\rightarrow$ **probability of 2^{-3} that a candidate verifies this**

Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$z_{t+2} = n_4^{t+2} l_6^{t+2} + l_8^{t+2} l_{10}^{t+2} + l_{32}^{t+2} l_{17}^{t+2} + l_{19}^{t+2} l_{23}^{t+2} + n_4^{t+2} n_{38}^{t+2} l_{32}^{t+2} + l_{30}^{t+2} + n_1^{t+2} + n_6^{t+2} + n_{15}^{t+2} + n_{17}^{t+2} + n_{23}^{t+2} + n_{28}^{t+2} + n_{34}^{t+2}$$

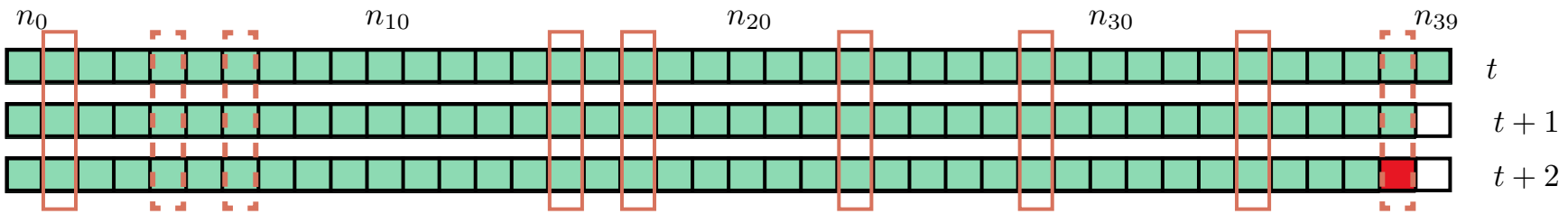


Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$z_{t+2} = n_4^{t+2} l_6^{t+2} + l_8^{t+2} l_{10}^{t+2} + l_{32}^{t+2} l_{17}^{t+2} + l_{19}^{t+2} l_{23}^{t+2} + n_4^{t+2} n_{38}^{t+2} l_{32}^{t+2} + l_{30}^{t+2} + n_1^{t+2} + n_6^{t+2} + n_{15}^{t+2} + n_{17}^{t+2} + n_{23}^{t+2} + n_{28}^{t+2} + n_{34}^{t+2}$$



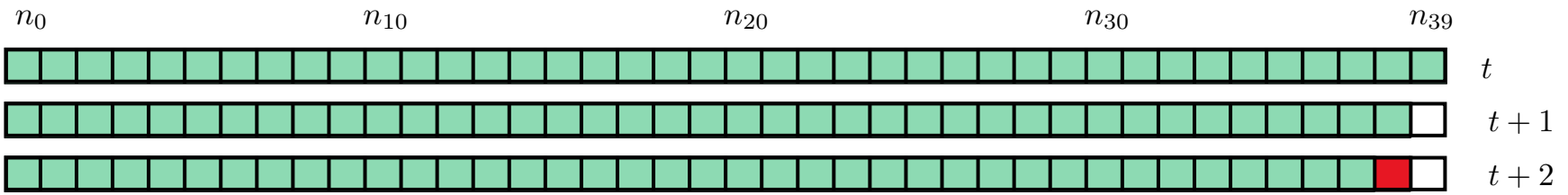
Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$z_{t+2} = n_4^{t+2} l_6^{t+2} + l_8^{t+2} l_{10}^{t+2} + l_{32}^{t+2} l_{17}^{t+2} + l_{19}^{t+2} l_{23}^{t+2} + n_4^{t+2} n_{38}^{t+2} l_{32}^{t+2} + l_{30}^{t+2} + n_1^{t+2} + n_6^{t+2}$$
$$+ n_{15}^{t+2} + n_{17}^{t+2} + n_{23}^{t+2} + n_{28}^{t+2} + n_{34}^{t+2}$$

$$n_{38}^{t+2} = n_{39}^{t+1} = g(N^t) + k_t^* + l_0^t + c^t$$



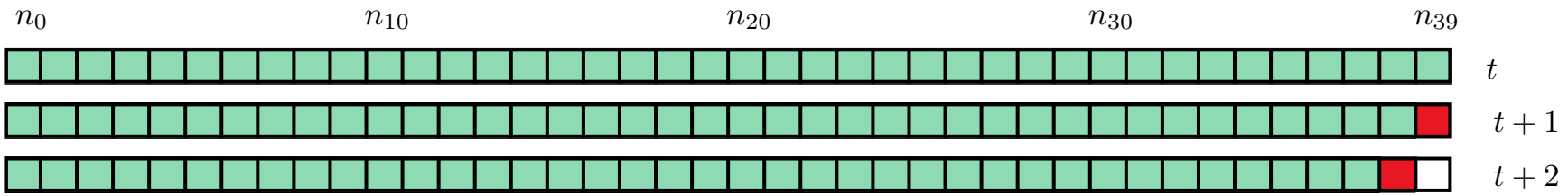
Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$z_{t+2} = n_4^{t+2} l_6^{t+2} + l_8^{t+2} l_{10}^{t+2} + l_{32}^{t+2} l_{17}^{t+2} + l_{19}^{t+2} l_{23}^{t+2} + n_4^{t+2} n_{38}^{t+2} l_{32}^{t+2} + l_{30}^{t+2} + n_1^{t+2} + n_6^{t+2}$$
$$+ n_{15}^{t+2} + n_{17}^{t+2} + n_{23}^{t+2} + n_{28}^{t+2} + n_{34}^{t+2}$$

$$n_{38}^{t+2} = n_{39}^{t+1} = g(N^t) + k_t^* + l_0^t + c^t$$



Merging Conditions - Type III filter

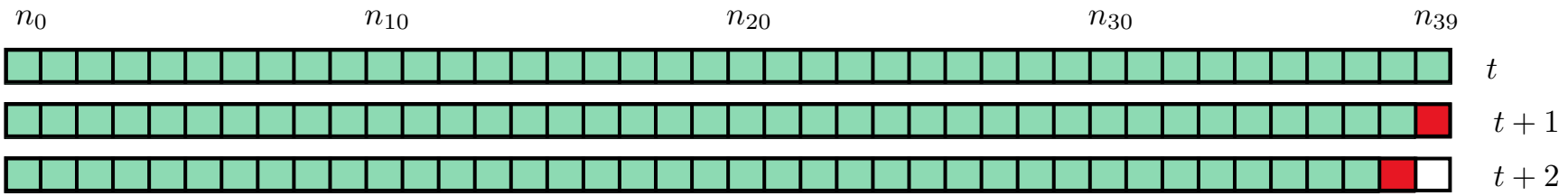
Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$z_{t+2} = n_4^{t+2} l_6^{t+2} + l_8^{t+2} l_{10}^{t+2} + l_{32}^{t+2} l_{17}^{t+2} + l_{19}^{t+2} l_{23}^{t+2} + n_4^{t+2} n_{38}^{t+2} l_{32}^{t+2} + l_{30}^{t+2} + n_1^{t+2} + n_6^{t+2} + n_{15}^{t+2} + n_{17}^{t+2} + n_{23}^{t+2} + n_{28}^{t+2} + n_{34}^{t+2}$$

$$n_{38}^{t+2} = n_{39}^{t+1} = g(N^t) + k_t^* + l_0^t + c^t$$

$$n_{39}^{t+1} = g(N^t) + (k_{t \bmod 80}) \times (l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t) + l_0^t + c^t$$



Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$n_{39}^{t+1} = g(N^t) + (k_{t \bmod 80}) \times (l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t) + l_0^t + c^t$$

$l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t = 0$	$l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t = 1$
no need of $k_{t \bmod 80}$	**guess** $k_{t \bmod 80}$
compute n_{39}^{t+1}	compute n_{39}^{t+1}
deduce z_{t+2}	deduce z_{t+2}
compare with the actual one	compare with the actual one
probability of being kept:	probability of being kept:
$\mathbf{1/2}$	$2 \times 1/2 = \mathbf{1}$

Merging Conditions - Type III filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

How can we use more keystream bits to filter out?

$$n_{39}^{t+1} = g(N^t) + (k_{t \bmod 80}) \times (l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t) + l_0^t + c^t$$

$l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t = 0$	$l_4^t + l_{21}^t + l_{37}^t + n_9^t + n_{20}^t + n_{29}^t = 1$
no need of $k_{t \bmod 80}$	**guess** $k_{t \bmod 80}$
compute n_{39}^{t+1}	compute n_{39}^{t+1}
deduce z_{t+2}	deduce z_{t+2}
compare with the actual one	compare with the actual one
probability of being kept:	probability of being kept:
$\mathbf{1/2}$	$2 \times 1/2 = \mathbf{1}$

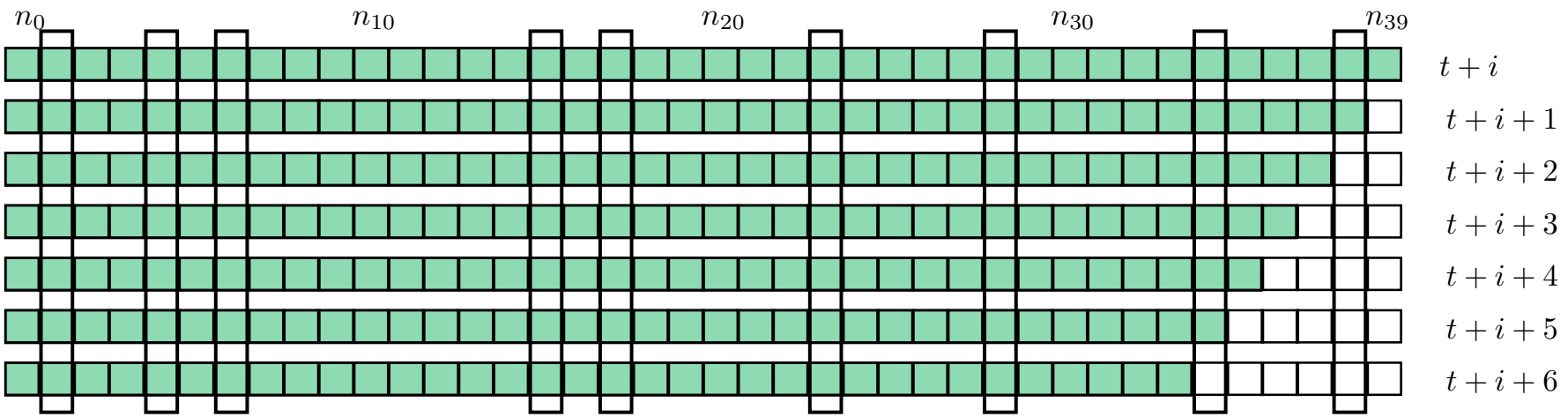
Total probability that a candidate state is kept: $1/2 \times \mathbf{1/2} + 1/2 \times \mathbf{1} = 3/4 = 2^{-0.415}$

Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

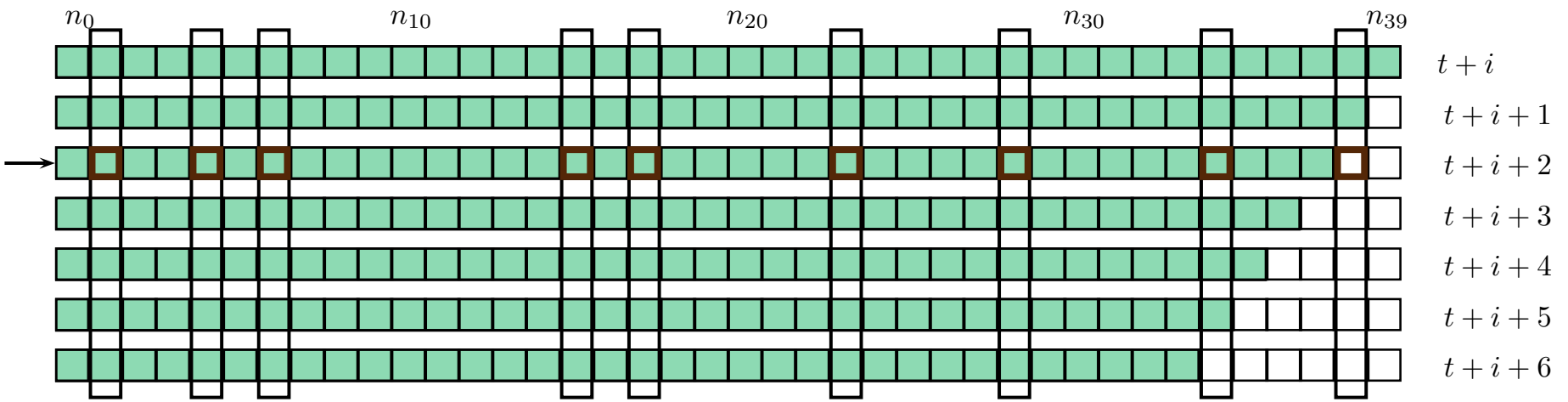


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

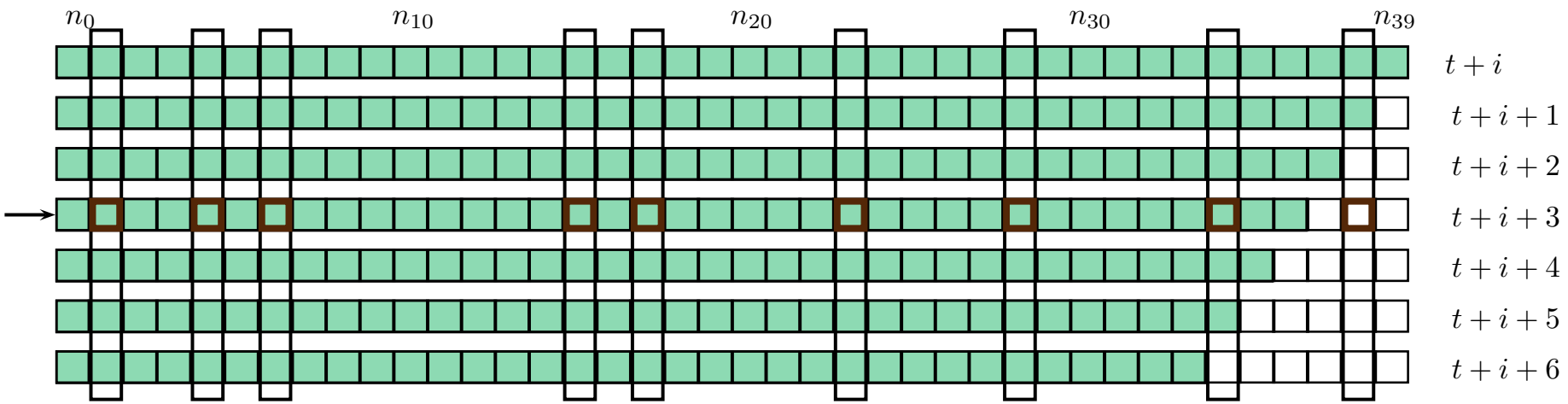


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

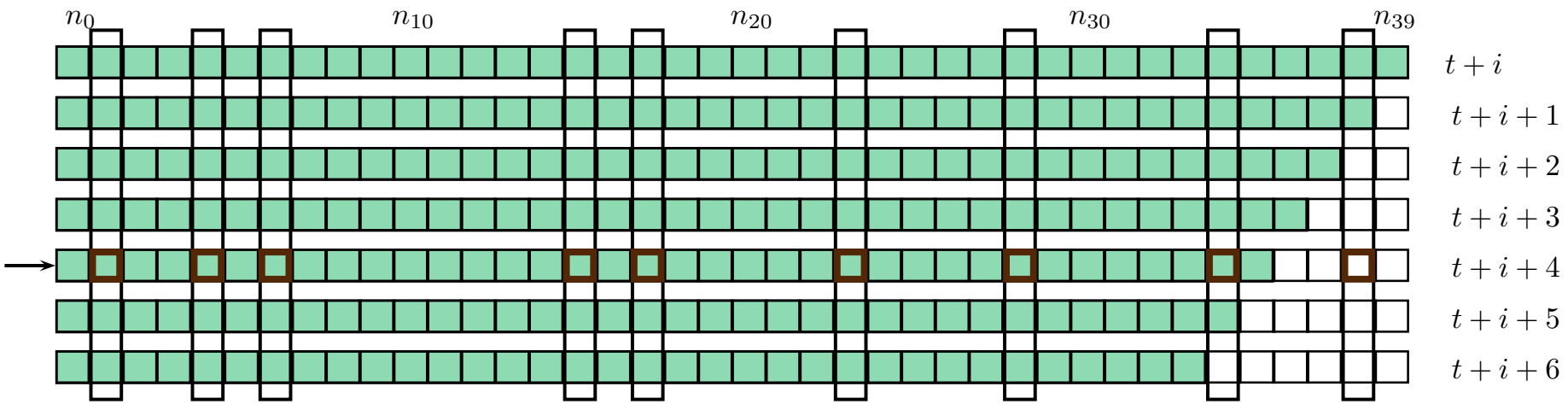


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

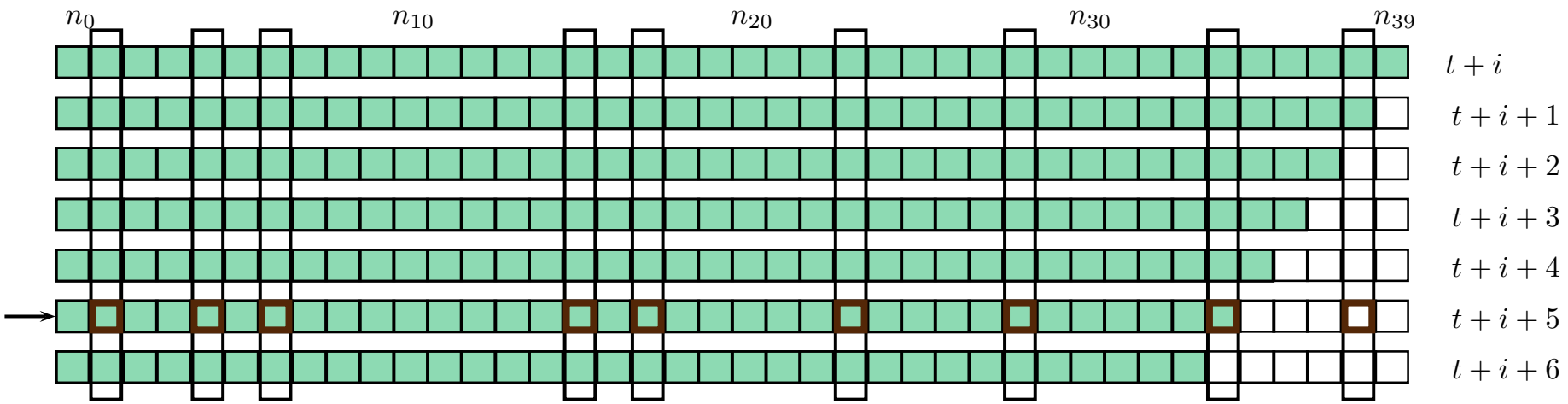


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

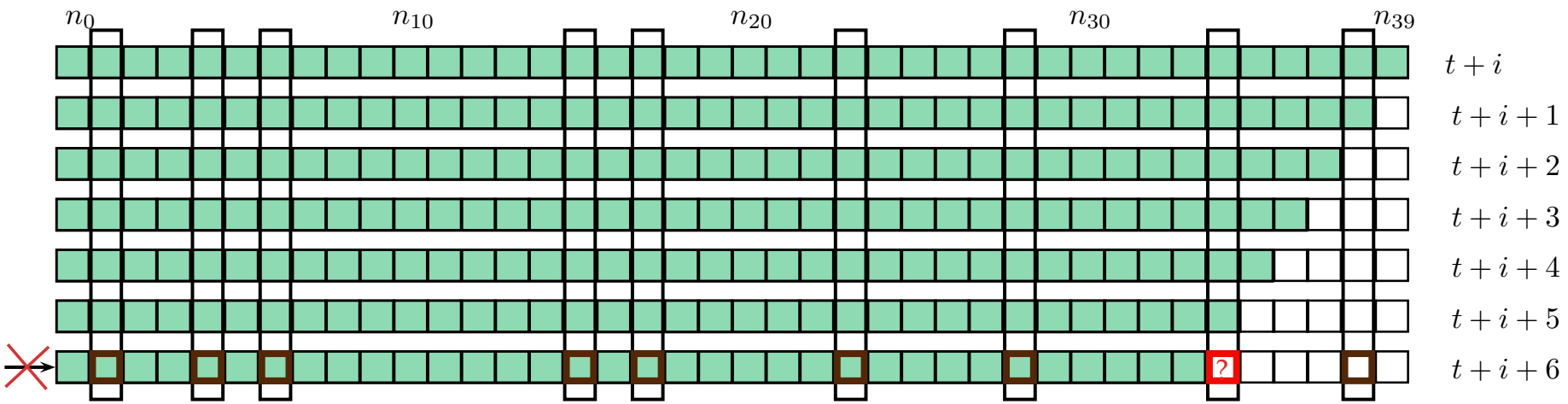


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

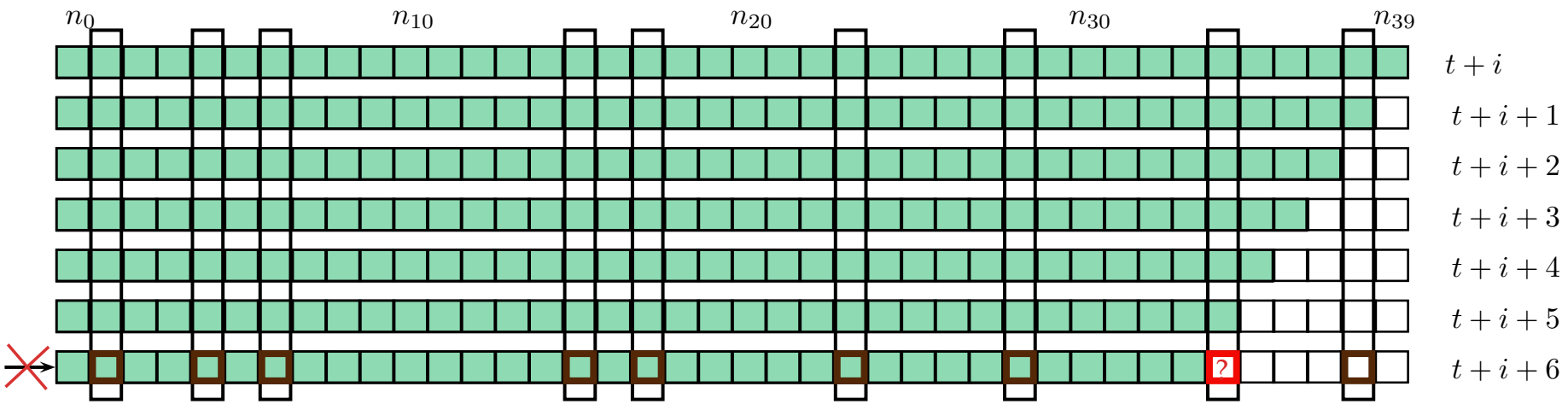


Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$



$\rightarrow$ As long as the computation of z_t doesn't imply another unknown state bit

$\rightarrow$ 4 rounds

Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

$n_4^t l_{32}^t = 0$	$n_4^t l_{32}^t = 1$
no need of n_{38}	$\times$
compute z_t	$\times$
compare with the actual one	$\times$
probability of being kept: **1/2**	probability of being kept: **1**

Merging Conditions - Type IV filter

Given an internal state candidate for the NLFSR and for the LFSR at time t

$$z_t = n_4^t l_6^t + l_8^t l_{10}^t + l_{32}^t l_{17}^t + l_{19}^t l_{23}^t + n_4^t n_{38}^t l_{32}^t + l_{30}^t + n_1^t + n_6^t + n_{15}^t + n_{17}^t + n_{23}^t + n_{28}^t + n_{34}^t$$

$n_4^t l_{32}^t = 0$ with probability $3/4$

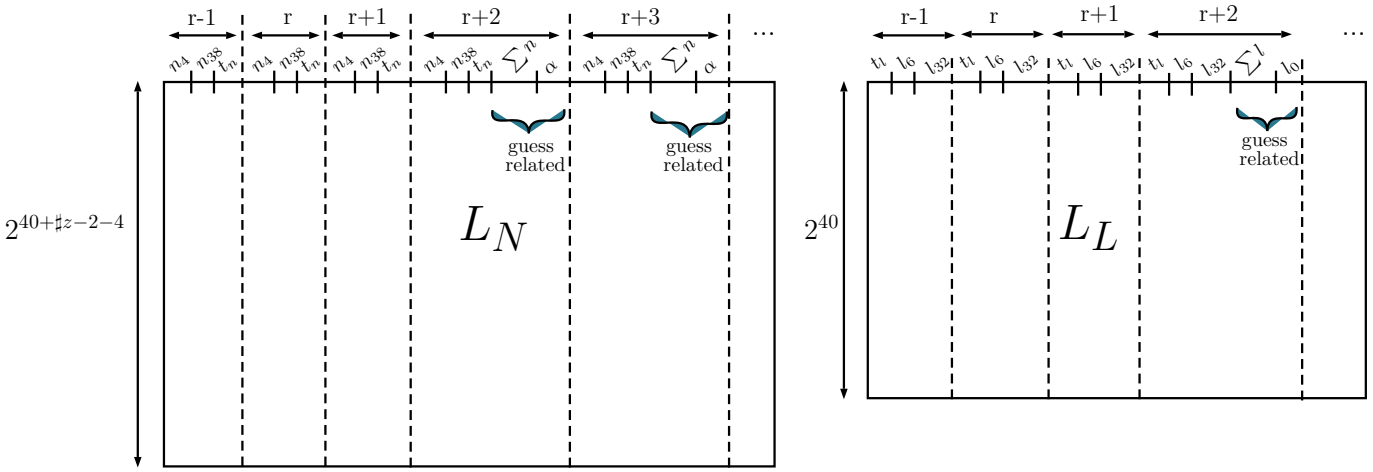
$n_4^t l_{32}^t = 0$	$n_4^t l_{32}^t = 1$
no need of n_{38}	$\times$
compute z_t	$\times$
compare with the actual one	$\times$
probability of being kept: **1/2**	probability of being kept: **1**

Total probability that a candidate state is kept:
$3/4 \times \mathbf{1/2} + 1/4 \times \mathbf{1} = 5/8 = 2^{-0.678}$

$\rightarrow$ **for 4 rounds total filter of** $2^{-2.71}$

Building the lists L_L and L_N

To make this step easier, we sort both lists according to the values of the bits that appear in the sieving relations

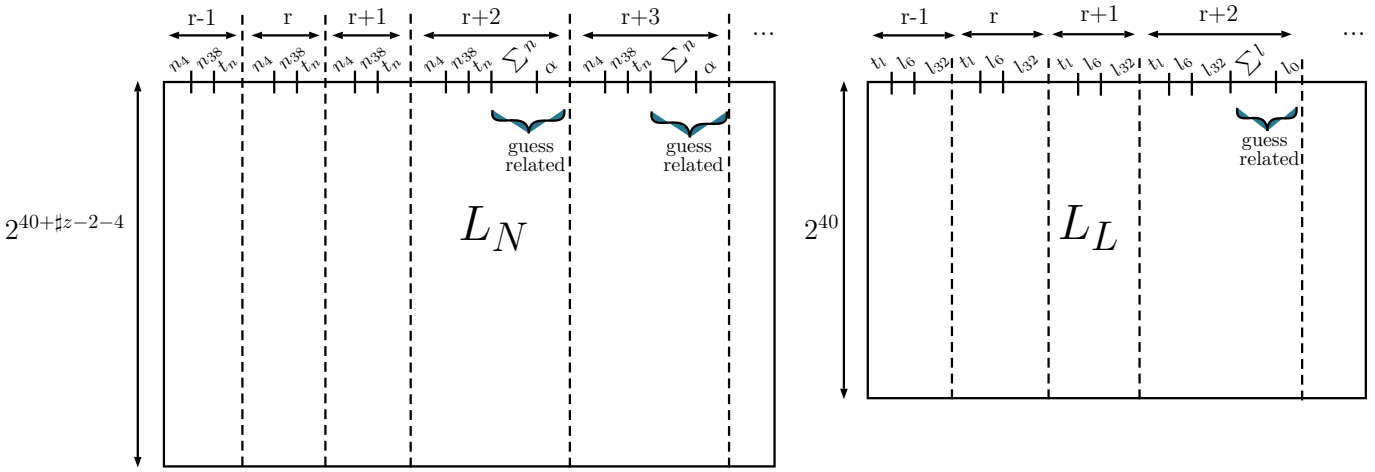


The merge is performed with the Gradual Matching technique

María Naya-Plasencia
How to Improve Rebound Attacks,
CRYPTO 2011.

Building the lists L_L and L_N

To make this step easier, we sort both lists according to the values of the bits that appear in the sieving relations



The merge is performed with the Gradual Matching technique

María Naya-Plasencia
How to Improve Rebound Attacks,
CRYPTO 2011.

→ **We end up with $\simeq 2^{72}$ candidate states with a time equivalent to $\simeq 2^{69}$ encryptions and 13 keystream bits**

Overview

3 main steps:

1. Build and arrange two independent lists :
   - the possible internal states for the LFSR at time t
   - the possible internal states for the NLFSR at time t
2. Merge them with the help of some keystream bits
3. Recover the whole key

Overview

3. Recover the whole key

Key Recovery

Key Recovery

Main Idea: Start from an internal state candidate and go back in time

k_τ^* is given by 2 formulas:

$$k_\tau^* = n_{39}^{\tau+1} + l_0^\tau + c^\tau + g(N^\tau)$$

$$k_\tau^* = (k_{\tau \bmod 80}) \times (l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau)$$

Key Recovery

Main Idea: Start from an internal state candidate and go back in time

k_τ^* is given by 2 formulas:

$$k_\tau^* = n_{39}^{\tau+1} + l_0^\tau + c^\tau + g(N^\tau)$$

$$k_\tau^* = (k_{\tau \bmod 80}) \times (l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau)$$

- $k_\tau^* = 1$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 0 \rightarrow$ discard (p=1/4)
- $k_\tau^* = 1$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 1 \rightarrow k_{\tau \bmod 80} = 1$ (p=1/4)
- $k_\tau^* = 0$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 1 \rightarrow k_{\tau \bmod 80} = 0$ (p=1/4)

Key Recovery

Main Idea: Start from an internal state candidate and go back in time

k_τ^* is given by 2 formulas:

$$k_\tau^* = n_{39}^{\tau+1} + l_0^\tau + c^\tau + g(N^\tau)$$

$$k_\tau^* = (k_{\tau \bmod 80}) \times (l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau)$$

- $k_\tau^* = 1$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 0 \rightarrow$ discard (p=1/4)
- $k_\tau^* = 1$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 1 \rightarrow k_{\tau \bmod 80} = 1$ (p=1/4)
- $k_\tau^* = 0$ and $(l_4^\tau + l_{21}^\tau + l_{37}^\tau + n_9^\tau + n_{20}^\tau + n_{29}^\tau) = 1 \rightarrow k_{\tau \bmod 80} = 0$ (p=1/4)

Repeat until the unique correct key and internal states are found
Note: If this step is done more than 80 times, master key bit involved start repeating

$\rightarrow$ In many cases we can check if the 2 results are consistent

Our Attack: Parameters

- Data complexity $= 112$ bits
  - 13 bits in the merging part
  - 99 bits in the key recovery part
- Memory complexity: 2^{46} elements of the NLFSR list
- Time complexity $< 2^{70}$

Conclusion

- We exhibit a key recovery attack on Sprout 2^{10} times faster than exhaustive search
- Experimentally verified on a toy version of Sprout
- Open question: Instantiate this design paradigm in a safe way (taking also into account recent independent analyses that showed other weaknesses on Sprout)

Thank you for your attention