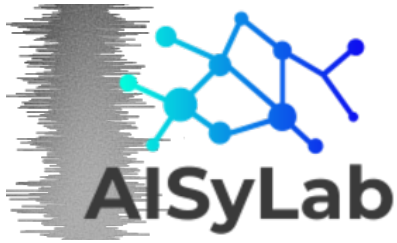
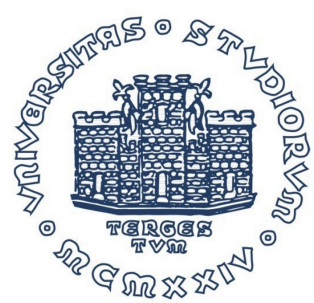




Radboud University

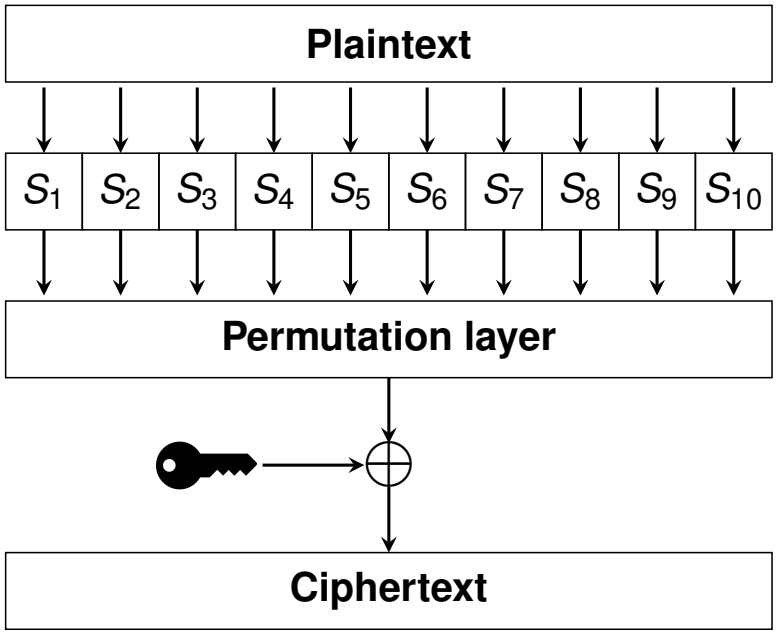
On the Linear Components Space of S-boxes Generated by Orthogonal CA

Luca Mariot, Luca Manzoni

luca.mariot@ru.nl

ACRI 2022 – Geneva, 13 September 2022

S-boxes in symmetric crypto



(a) Substitution-Permutation Network (SPN)

Zoom in on a **S-box** S_i:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8

$\Downarrow F : \{0,1\}^n \rightarrow \{0,1\}^n$

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8

(b) S-box S_i

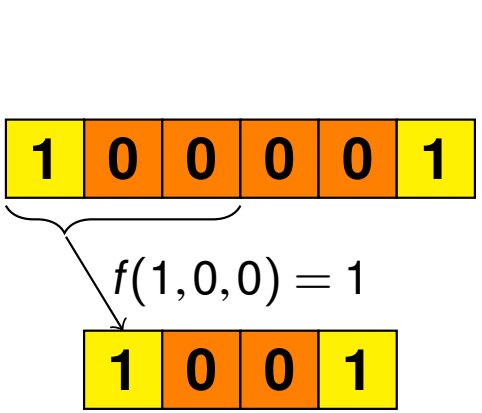
S-boxes in SPN ciphers must satisfy several properties, mainly:

- **invertibility** (for decryption)
- High **nonlinearity** (for resistance to linear cryptanalysis)

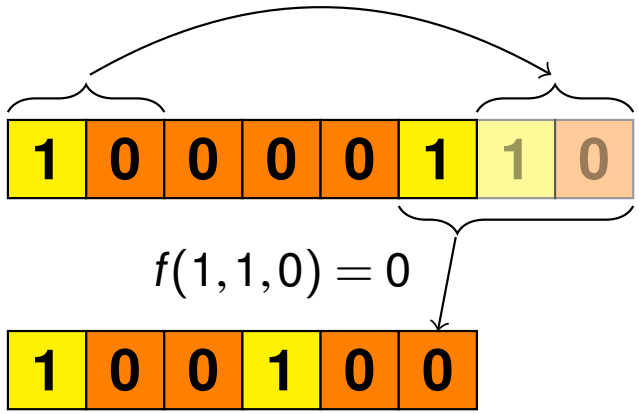
Cellular Automata

- One-dimensional Cellular Automaton (CA): a discrete parallel computation model composed of a finite array of n cells

Example: $n = 6$, $d = 3$, $f(s_i, s_{i+1}, s_{i+2}) = s_i \oplus s_{i+1} \oplus s_{i+2}$ (rule 150)



No Boundary CA – NBCA



Periodic Boundary CA – PBCA

- Each cell updates its state $s \in \{0,1\}$ by evaluating a local rule $f : \{0,1\}^d \to \{0,1\}$ on itself and the $d-1$ cells on its right

CA-based symmetric ciphers

The "**dynamical system**" way:

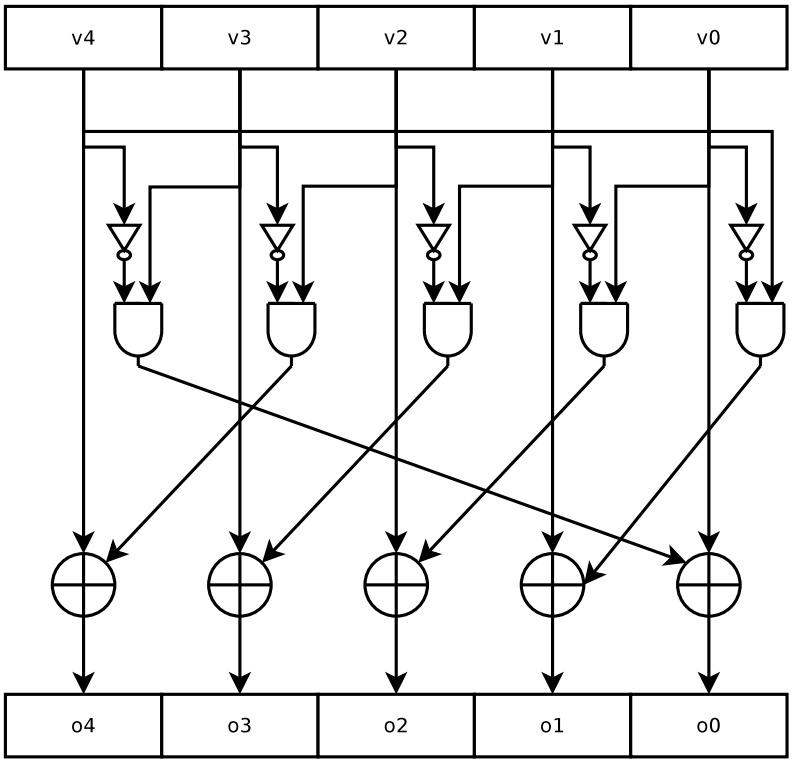
- Iterate a CA for **several** time steps to encrypt the **whole** plaintext
- Typically seen in CA venues [S04, M06, S08]
- Several weaknesses (low diffusion, ...)

The "**reductionist**" way:

- Iterate a CA for a **single** time stes to encrypt a **part** of plaintext
- More common in crypto venues [P17, G18, M19]
- In line with current state of the art

Real world CA-Based Crypto: KECCAK χ S-box

- Local rule: $\chi(x_1, x_2, x_3) = x_1 \oplus (1 \oplus (x_2 \cdot x_3))$ (rule 210)
- Invertible for every odd size n of the CA



- Used as an S-box with $n = 5$ in the KECCAK specification of the SHA-3 standard [B11]

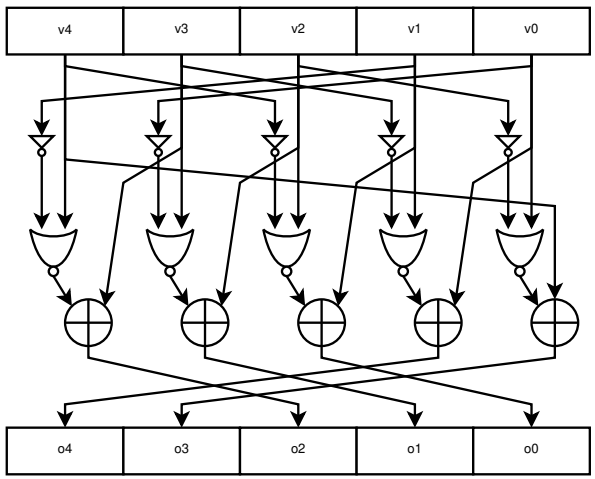
Periodic CA-based S-boxes

Algebraic approach:

- Theoretical analysis of specific CA rules as S-boxes
- Examples: χ in KECCAK [B11]

Heuristic approach:

- Use of heuristic algorithms (e.g. GP) to optimize the crypto properties of CA rules [P17, M19]
- More flexibility wrt other properties (e.g. implementation cost)



Another angle: Orthogonal Latin Squares (OLS)

Definition

A *Latin square* is a $n \times n$ matrix where all rows and columns are permutations of $[n] = \{1, \cdots, n\}$. Two Latin squares are *orthogonal* if their superposition yields all the pairs $(x, y) \in [n] \times [n]$.

1	3	4	2
4	2	1	3
2	4	3	1
3	1	2	4

1	4	2	3
3	2	4	1
4	1	3	2
2	3	1	4

1,1	3,4	4,2	2,3
4,3	2,2	1,4	3,1
2,4	4,1	3,3	1,2
3,2	1,3	2,1	4,4

Remark: two OLS define a **bijection** over pairs in $[n] \times [n]$

Latin Squares through Bipermutive CA (1/2)

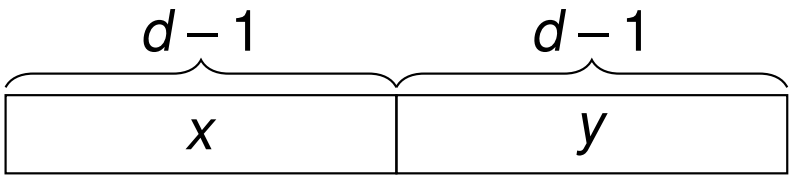
- Bipermutive CA: denoting $\mathbb{F}_2 = \{0,1\}$, local rule f is defined as

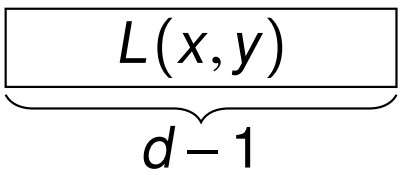
$$f(x_1, \cdots, x_d) = x_1 \oplus \varphi(x_2, \cdots, x_{d-1}) \oplus x_d$$

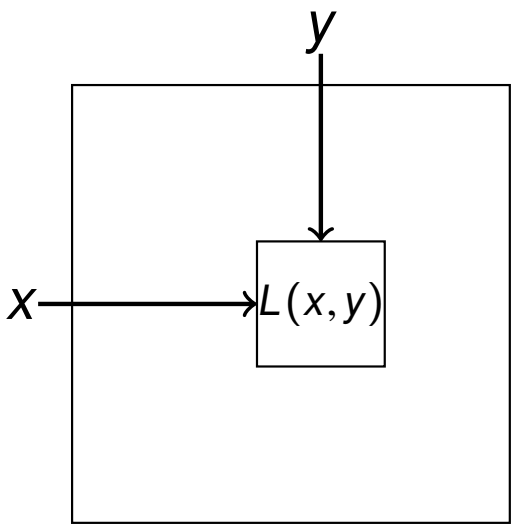
- $\varphi : \mathbb{F}_2^{d-2} \to \mathbb{F}_2$: generating function of f

Lemma ([M20])

A CA $F : \mathbb{F}_2^{2(d-1)} \to \mathbb{F}_2^d$ with bipermutive rule $f : \mathbb{F}_2^d \to \mathbb{F}_2$ generates a Latin square of order $N = 2^{d-1}$

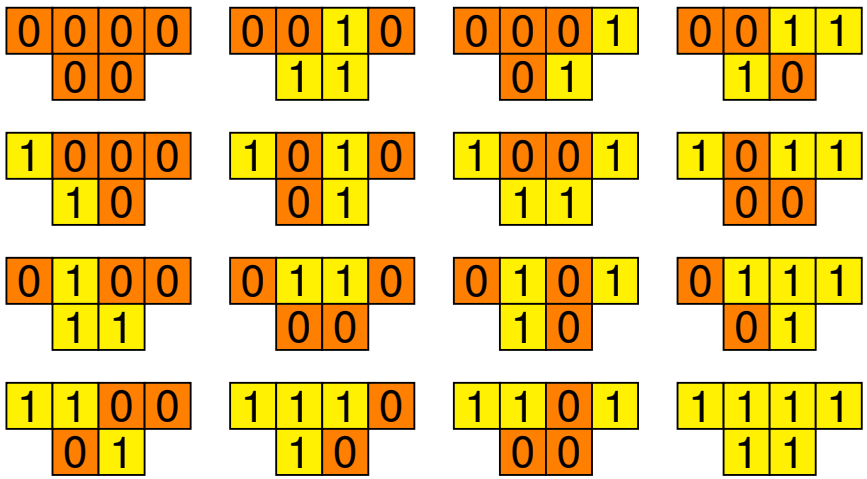






Latin Squares through Bipermutive CA (2/2)

- Example: CA $F : \mathbb{F}_2^4 \to \mathbb{F}_2^2$, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3$ (Rule 150)
- Encoding: $00 \mapsto 1, 10 \mapsto 2, 01 \mapsto 3, 11 \mapsto 4$



(a) Rule 150 on 4 bits

1	4	3	2
2	3	4	1
4	1	2	3
3	2	1	4

(b) Latin square L_{150}

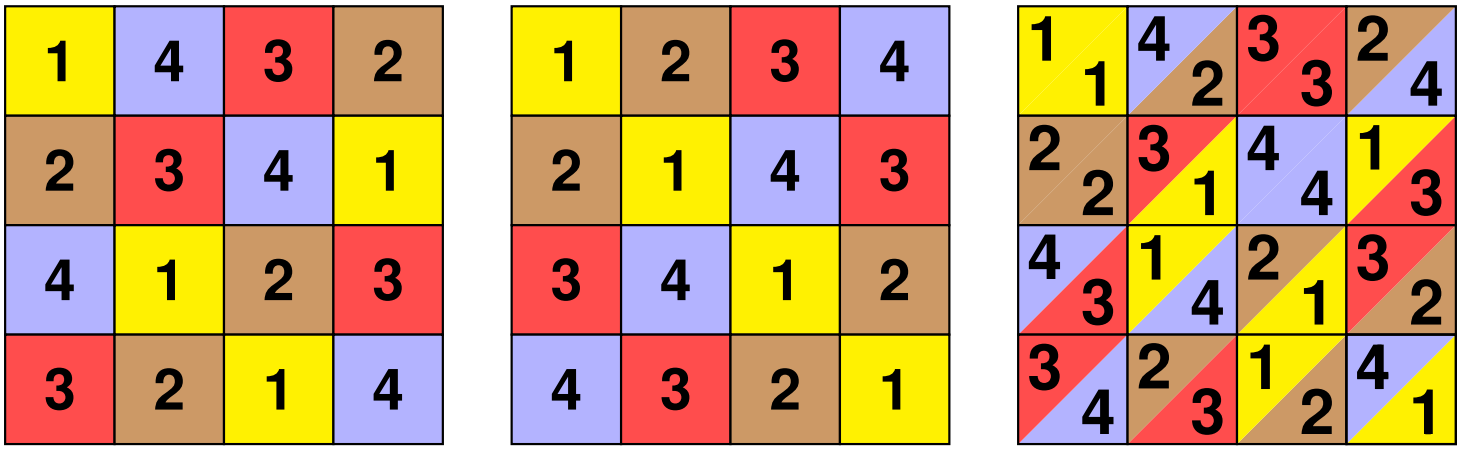
Orthogonal Cellular Automata (OCA): two bipermutive CA F, G generating a pair of OLS

Linear OCA

- Bipermutive Linear rule: $f(x) = x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_{d-1} x_{d-1} \oplus x_d$
- Polynomial rule: $P_f(X) = 1 + a_2 X + \cdots + a_{d-1} X^{d-2} + X^{d-1}$

Theorem ([M20])

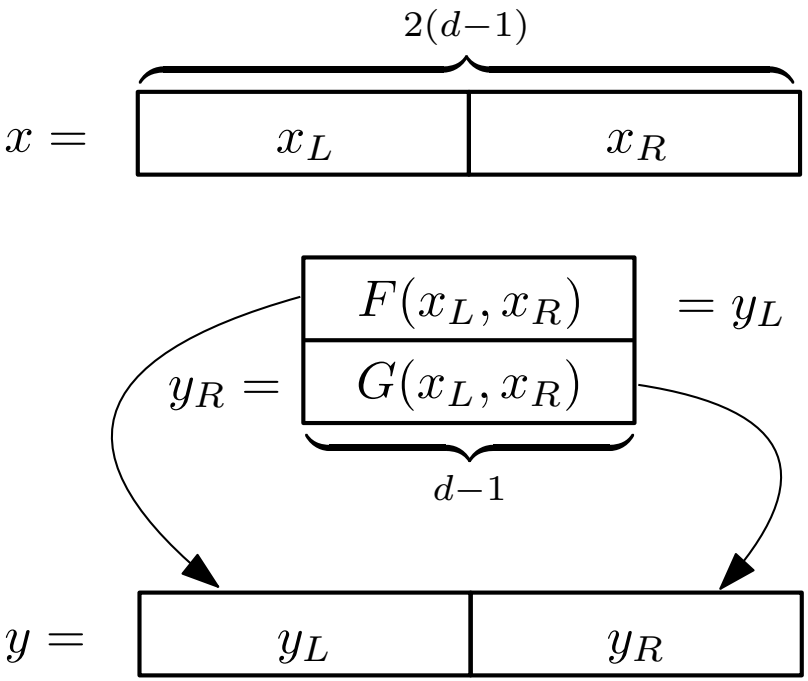
Two bipermutive linear rules generates OCA if and only if their associated polynomials are coprime



(a) Rule 150 (b) Rule 90 (c) Superposition

Figure: $P_{150}(X) = 1 + X + X^2$, $P_{90}(X) = 1 + X^2$ (coprime)

S-boxes based on OCA



Basic Idea:

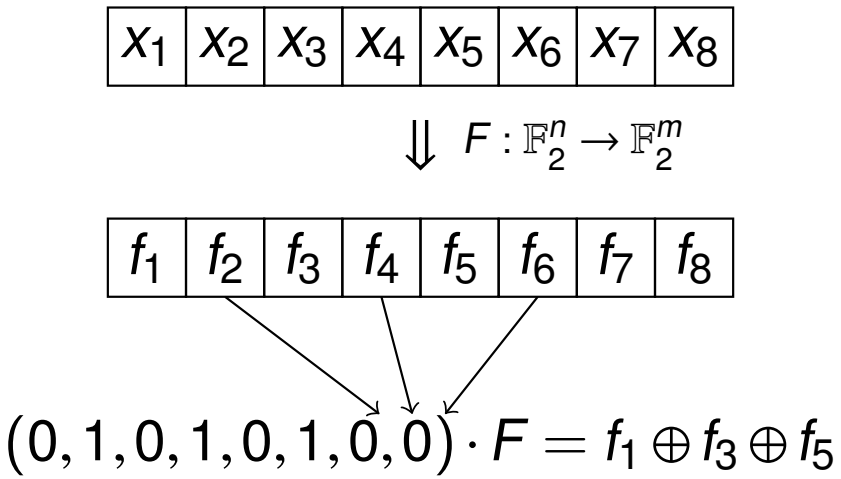
- Evaluate two OCA F, G over the input $x = (x_L, x_R)$
- Use the outputs $F(x_L, x_R)$ and $G(x_L, x_R)$ as left and right outputs of a S-box H

Motivation:

- The S-box is **invertible** (because of orthogonality)
- Latin squares ensure a minimum degree of **diffusion**
- Approach used iteratively in [M21] to generate pseudorandom sequences

Search for nonlinear OCA S-boxes

- **Method**: exhaustive search of OCA pairs of $d = 4, 5$, which generate S-boxes of $n = 6, 8$ bits
- Compute the **nonlinearity** of the S-boxes as the **minimum nonlinearity** of their component functions



- Nonlinearity of a component $v \cdot F$: computed with the **Walsh-Hadamard transform**

Result: All S-boxes are linear! :-(

- **First finding**: all tested S-boxes are linear
- Linear components form a **vector subspace** of $\{0,1\}^n$

Table: Classification of OCA-based S-boxes of diameter $d = 4$ and $d = 5$ in terms of the nonlinearity of their local rules and LCS dimensions.

d	$nl(f,g)$	$\#OCA$	dim	$\#dim$
4	$(4,4)$	32	3	32
5	$(4,4)$	768	4	768
	$(8,8)$	768	4	704
			3	64

- **Consequence:** useless as S-boxes :-(

The structure of LCS vector spaces

- Vector spaces over finite fields are **error-correcting codes**
- What about the **generating matrices** of LCS in OCA?

$$G = \begin{pmatrix} a_0 & \cdots & a_{t-1} & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & a_0 & \cdots & a_{t-1} & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 & a_0 & \cdots & a_{t-1} & 1 \end{pmatrix}.$$

- ... but this is the form of the transition matrix of a linear CA!
- So: *the subspaces of the linear components of OCA S-boxes are represented themselves by a linear CA*

Conclusions and Future Work

Summing up:

- OCA cannot be used to design useful S-boxes
- Surprisingly, the vector space of the linear components of an OCA S-box is *itself* a linear CA

Future directions:

- Check if there are nonlinear OCA S-boxes of higher diameter
- If not, can the CA structure of LCS be used to give a theoretical characterization of nonlinear OCA?

References

[B11] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche: The Keccak reference. (January 2011). http://keccak.noekeon.org/

[G18] A. Ghoshal, R. Sadhukhan, S. Patranabis, N. Datta, S. Picek, D. Mukhopadhyay: Lightweight and Side-channel Secure 4×4 S-Boxes from Cellular Automata Rules. IACR Trans. Symmetric Cryptol. 2018(3): 311-334 (2018)

[M06] S. Marconi, B. Chopard: Discrete physics, cellular automata and cryptography. In: Proceedings of ACRI 2006, pp. 617-626 (2006)

[M21] L. Mariot: Hip to Be (Latin) Square: Maximal Period Sequences from Orthogonal Cellular Automata. In: Proceedings of CANDAR 2021, pp. 29-37 (2021)

[M20] L. Mariot, M. Gadouleau, E. Formenti, A. Leporati: Mutually orthogonal latin squares based on cellular automata. Des. Codes Cryptogr. 88(2): 391-411 (2020)

[M19] L. Mariot, S. Picek, A. Leporati, and D. Jakobovic. Cellular automata based S-boxes. Cryptography and Communications 11(1): 41-62 (2019)

[M17] L. Mariot, E. Formenti, A. Leporati: Enumerating Orthogonal Latin Squares Generated by Bipermutive Cellular Automata. Proceedings of AUTOMATA 2017, pp. 151-164 (2017)

[P17] S. Picek, L. Mariot, B. Yang, D. Jakobovic, N. Mentens: Design of S-boxes defined with cellular automata rules. Conf. Computing Frontiers 2017: 409-414 (2017)

[S04] M. Seredynski, P. Bouvry: Block encryption using reversible cellular automata. In: Proceedings of ACRI 2004, pp. 785-792 (2004)

[S08] M. Szaban, F. Seredynski: Cryptographically strong S-boxes based on cellular automata. In: Proceedings of ACRI 2008, pp. 478-485 (2008)