# Cellular Automata in Cryptography – A Survey of Past and Current Results, and Future Directions of Research

Luca Mariot

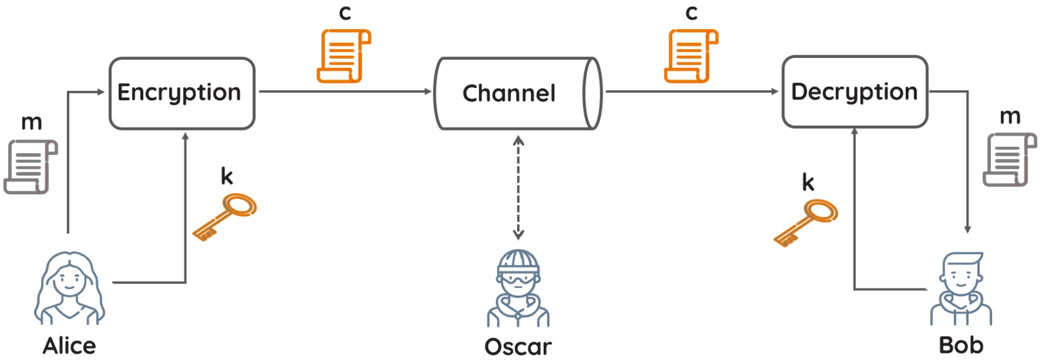
l.mariot@utwente.nl

January 6, 2026

# Introduction to Cryptography

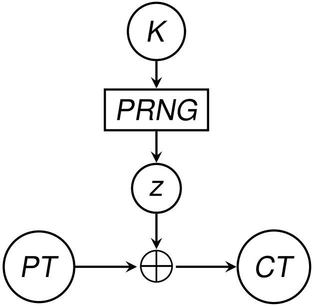
# Symmetric Cryptography

**Basic Goal**: enable *confidentiality* in communication using a shared symmetric key

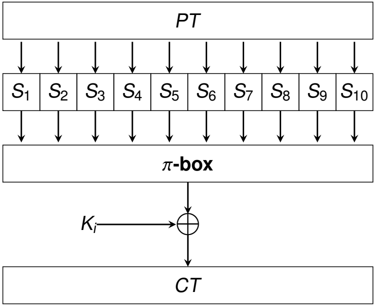



- **m**: plaintext
- **c**: ciphertext
- **k**: encryption/decryption key

# Primitives in symmetric crypto




(a) Stream cipher



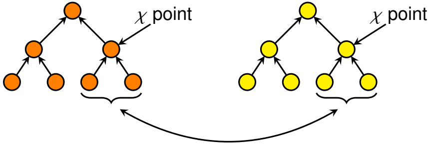

(b) Block cipher

Symmetric ciphers require several **low-level primitives**, such as:

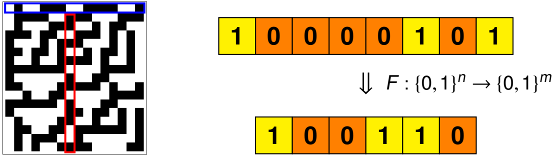
- Pseudorandom number generators (PRNG)
- Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and S-boxes
- Permutation (diffusion) layers, ...

# Design Approaches

- "Traditional" approach: ad-hoc algebraic constructions to choose primitives with specific security properties
- "AI" approach: support the designer in choosing the primitives using AI methods/models from the following domains:
  - Optimization (Evolutionary algorithms, swarm intelligence...)



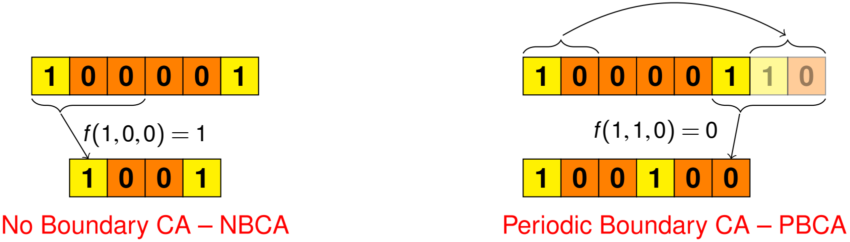

  - Computational models (cellular automata, neural networks...)



# Cellular Automata

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

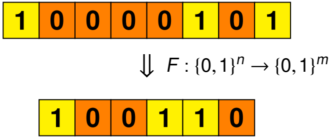
Example: $n = 6$, $d = 3$, $\omega = 0$, $f(s_i, s_{i+1}, s_{i+2}) = s_i \oplus s_{i+1} \oplus s_{i+2}$ (rule 150)

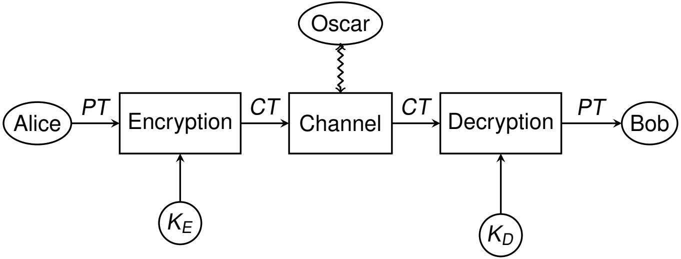



- Each cell updates its state $s \in \{0,1\}$ by applying a local rule $f : \{0,1\}^d \to \{0,1\}$ to itself, the $\omega$ cells on its left and the $d - 1 - \omega$ cells on its right

# Motivations

**General Research Goal**: Investigate cryptographic primitives defined by CA




Why CA, anyway?

1. **Security from Complexity**: CA can yield very complex dynamical behaviors
2. **Efficient implementation**: Leverage CA parallelism and locality

# Summary

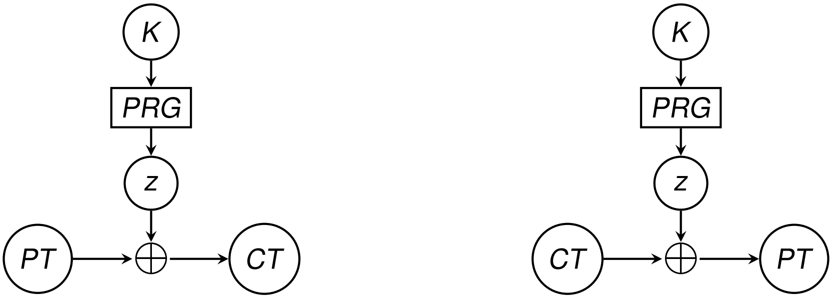
We are going to review three use cases of cyptographic primitives based on CA:

- **Stream Ciphers and Pseudorandom Generators**
- **Block Ciphers and S-boxes**

# Summary

## Stream Ciphers based on CA

# Vernam Stream Cipher

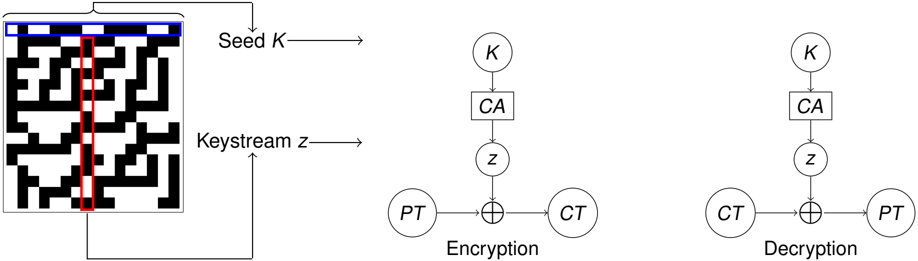



(a) Encryption (b) Decryption

- $K$: secret key
- $PRG$: Pseudorandom Generator
- $z$: keystream
- $\bigoplus$: bitwise XOR
- $PT$: Plaintext
- $CT$: Ciphertext

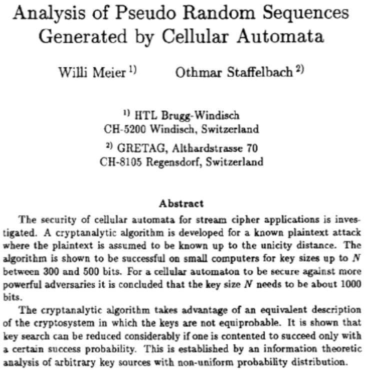
# CA-based Crypto History: Wolfram's PRNG

- CA-based Pseudorandom Generator (PRG) [W86]: central cell of rule 30 CA used as a stream cipher keystream




- Secret key: (random) initial condition of the CA

# Attacks on Wolfram's PRNG [M91]



Analysis of Pseudo Random Sequences Generated by Cellular Automata

Willi Meier [1] Othmar Staffelbach [2]

[1] HTL Brugg-Windisch
CH-5200 Windisch, Switzerland

[2] GRETAG, Althardstrasse 70
CH-8105 Regensdorf, Switzerland

**Abstract**

The security of cellular automata for stream cipher applications is investigated. A cryptanalytic algorithm is developed for a known plaintext attack where the plaintext is assumed to be known up to the unicity distance. The algorithm is shown to be successful on small computers for key sizes up to $N$ between 300 and 500 bits. For a cellular automaton to be secure against more powerful adversaries it is concluded that the key size $N$ needs to be about 1000 bits.

The cryptanalytic algorithm takes advantage of an equivalent description of the cryptosystem in which the keys are not equiprobable. It is shown that key search can be reduced considerably if one is contented to succeed only with a certain success probability. This is established by an information theoretic analysis of arbitrary key sources with non-uniform probability distribution.

- Wolfram used only *empirical* and *statistical* tests for security analysis
- Meier and Staffelbach [M91] showed a correlation attack exploiting the *quasi-linearity* of rule 30:

$$f(x_1, x_2, x_3) = x_1 XOR (x_2 OR X_3)$$

**Consequence:** Wolfram's PRNG is useless when equipped with rule 30

**Question:** Can we fix Wolfram's PRNG?

# Cryptographic Properties of Boolean Functions

- A mapping $f : \mathbb{F}_2^n \to \mathbb{F}_2$, most commonly represented by its *Truth Table* (TT) $\Omega_f$
- *Walsh Transform* (WT): represents $f$ as *correlations* with *linear* functions $a \cdot x$

$$W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus a \cdot x}$$

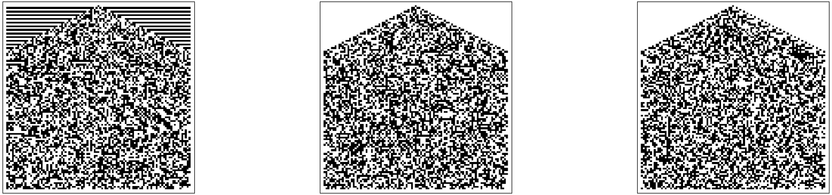
| $(x_1, x_2, x_3)$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $\Omega_f$ | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| $W_f(a)$ | 0 | $-4$ | 0 | 4 | 0 | 4 | 0 | 4 |

A **Boolean function** used in stream ciphers should be

- balanced: $W_f(\underline{0}) = 0$
- highly nonlinear $nl(f) = 2^{n-1} - \frac{1}{2} \max_{a \in \mathbb{F}_2^n} \{|W_f(a)|\}$
- correlation immune of high order $t$ (min value s.t. $W_f(a) = 0$ for all $a$ with at most $t$ ones)

# Salvaging Wolfram's PRNG

- **Problem** of rule 30: too small to give any meaningful cryptographic property
- Later works considered rules of larger diameters [L13, F14, L14]

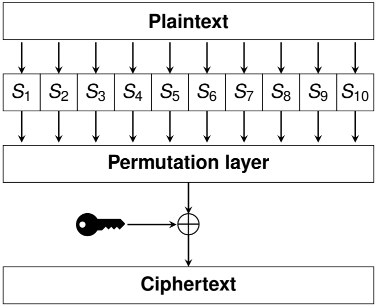


(a) Rule 1452976485 (b) Rule 1520018790 (c) Rule 2778290790

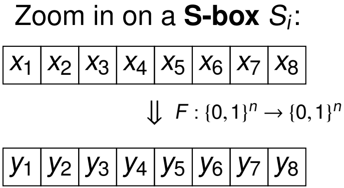
- Example: bipermutive rules [L13] satisfy 1st-order correlation immunity, $d = 5$ is the minimum to find also nonlinear rules.

# Block Ciphers based on CA

# Zoom on SPN Block Ciphers




(a) Substitution-Permutation Network (SPN)




(b) S-box $S_i$

S-boxes in SPN ciphers must satisfy several properties, mainly [C21]:

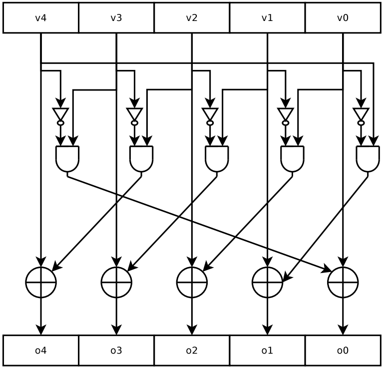
- **invertibility** (for decryption)
- High **nonlinearity** (for linear cryptanalysis)
- Low **differential uniformity** (for differential cryptanalysis)

# "Reductionist" CA-Based Crypto: KECCAK $\chi$ S-box

- Local rule (rule 210):

$$\chi(x_1, x_2, x_3) = x_1 \oplus (1 \oplus (x_2 \cdot x_3))$$

- Invertible for every odd CA size [D95]
- Used as a PBCA with $n = 5$ in the KECCAK specification of SHA-3 standard [B11]
- CA iterated for a *single* step, and interleaved with other (non-local) operations



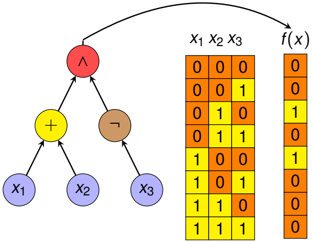
# CA-based S-boxes

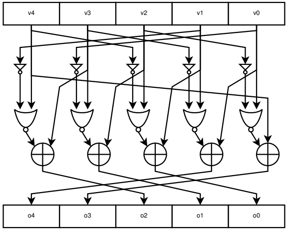
**Algebraic** approach:

- Theoretical analysis of specific CA rules as S-boxes
- Examples: $\chi$ in Keccak [D95, B11]

**Heuristic** approach:

- Use of heuristic algorithms to optimize the crypto properties of CA rules [P17a, P17b, M19, M21, D23]
- More flexibility wrt other properties (e.g. implementation cost)



# CA also for Diffusion Layers?

- The propagation of differences is bounded by the CA "speed of light" (diameter)

Image credits: J. Daemen, *On Keccak and SHA-3*,
`http://ice.mat.dtu.dk/slides/KeccakIcebreak-slides.pdf`

- **Consequence:** better to avoid CA in diffusion layers

# Conclusions

# Future Outlook

**To sum up:**

- CA have their place in cryptography
- But one needs to link them consistently to security models and properties of ciphers

**Directions for future research:**

- For stream ciphers: closely analyze Wolfram's PRNG, find new attacks [M17]
- For block ciphers: study CA-based permutation layers [M20, G23], and compare them with traditional ones

# References


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

[C21] C. Carlet: Boolean functions for cryptography and coding theory. Cambridge University Press (2021)

[D95] J. Daemen: Cipher and hash function design strategies based on linear and differential cryptanalysis. Ph.D. thesis, KU Leuven (1995)

[D23] M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek, S.: A survey of metaheuristic algorithms for the design of cryptographic Boolean functions. Cryptogr. Commun. 15(6):1171–1197 (2023)

[F14] E. Formenti, K. Imai, B. Martin, J. Yunès: Advances on random sequence generation by uniform cellular automata. In: Computing with New Resources, LNCS vol. 8808, pp. 56–70 (2014)

[G23] M. Gadouleau, L. Mariot, S. Picek: Bent functions in the partial spread class generated by linear recurring sequences. Des. Codes Cryptogr. 91(1):63–82 (2023)

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

[L14] A. Leporati and L. Mariot: Cryptographic properties of bipermutive cellular automata rules. J. Cell. Autom. 9(5-6):437–475 (2014)

[L13] A. Leporati and L. Mariot: 1-Resiliency of bipermutive cellular automata rules. In: Proceedings of AUTOMATA 2013, pp. 110–123 (2013)

[M21] L. Mariot, S. Picek, D. Jakobovic, A. Leporati: Evolutionary algorithms for designing reversible cellular automata. Genet. Prog. Evolvable Mach. 22(4):429–461 (2021)

[M20] L. Mariot, M. Gadouleau, M., 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, A. Leporati, A. Dennunzio, E. Formenti: Computing the periods of preimages in surjective cellular automata. Nat. Comput. 16(3):367–381 (2017)

[M91] W. Meier, O. Staffelbach: Analysis of pseudo random sequence generated by cellular automata. In: Proc. of EUROCRYPT'91, pp. 186–199 (1991)

[P17a] S. Picek, L. Mariot, B. Yang, D. Jakobovic, N. Mentens: Design of S-boxes defined with cellular automata rules. In: Proceedings of Conf. Computing Frontiers 2017, pp. 409–414 (2017)

[P17b] S. Picek, L. Mariot, A. Leporati, D. Jakobovic: Evolving s-boxes based on cellular automata with genetic programming. In: Proceedings of GECCO 2017 (Companion), pp. 251–252 (2017)

[W86] S. Wolfram. Cryptography with cellular automata. In CRYPTO '85, pp. 429–432 (1986)