Are Cellular Automata of any use to Cryptography?

Luca Mariot

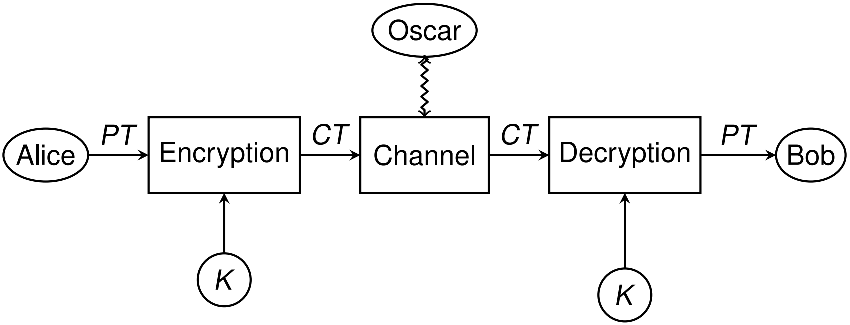
l.mariot@utwente.nl

Milano – December 4, 2024

Background on Cryptographic Primitives and Cellular Automata

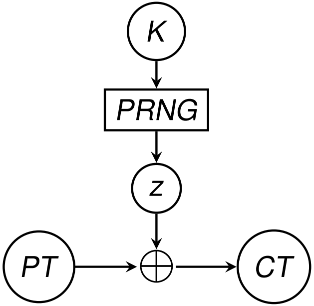
Symmetric Cryptography

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

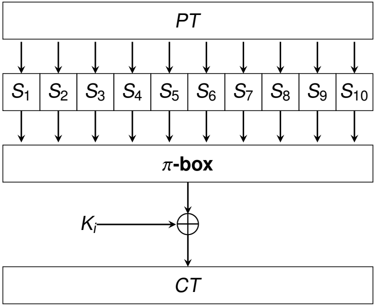


- *PT*: plaintext
- *CT*: ciphertext
- *K*: encryption/decryption key

Primitives in symmetric crypto



(a) Stream cipher



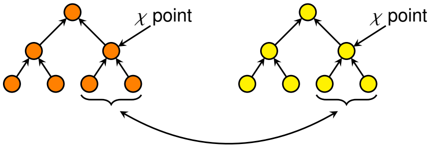
(b) Block cipher

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

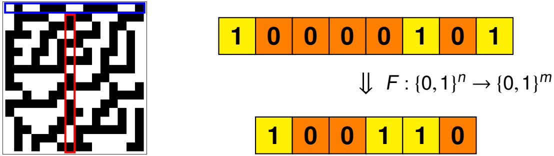
- Pseudorandom number generators (PRNG)
- Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and S-boxes
- Permutation (diffusion) layers, ...

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...)



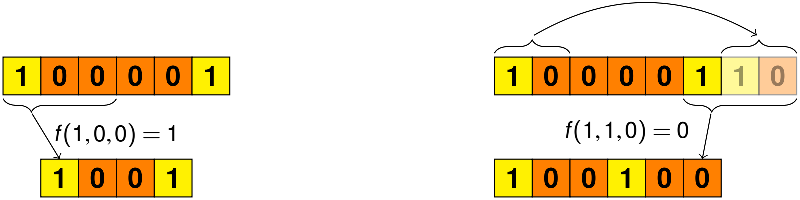
  - Computational models (cellular automata, neural networks...)



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)



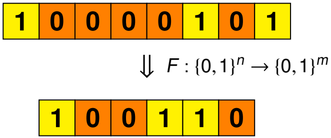
No Boundary CA – NBCA

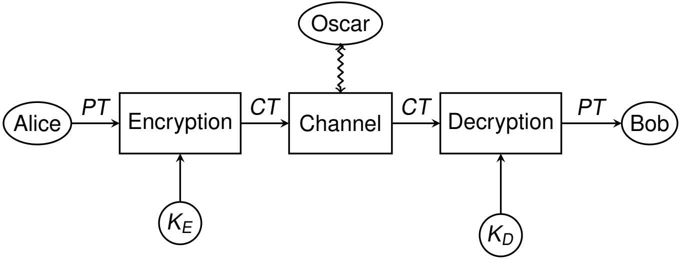
Periodic Boundary CA – PBCA

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

Motivations

General Research Goal: Investigate cryptographic primitives defined by CA



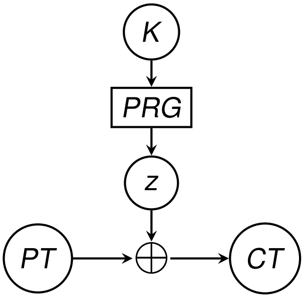


Why CA, anyway?

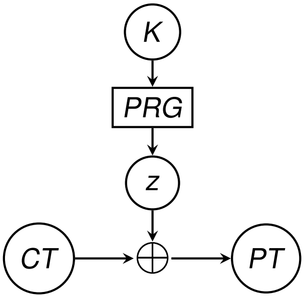
1. **Security from Complexity**: CA can yield very complex dynamical behaviors
2. **Efficient implementation**: Leverage CA parallelism and locality

Stream Ciphers based on CA

Vernam Stream Cipher



(a) Encryption

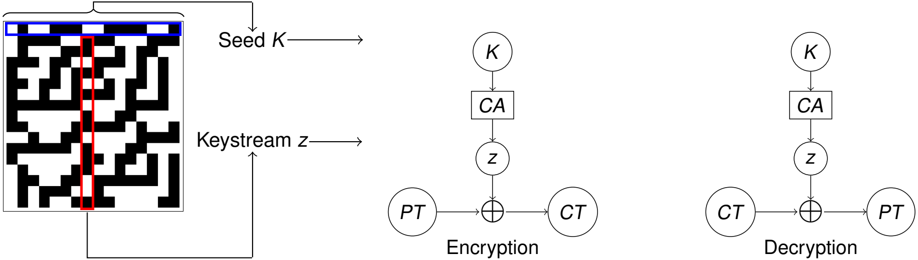


(b) Decryption

- 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
- Paper published in CRYPTO'85

Attacks on Wolfram's PRNG [M91, K97]

Exploiting the specific form of Rule 30: $f(x_1, x_2, x_3) = x_1 XOR(x_2 OR X_3)$

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.

Inversion of Cellular Automata Iterations

Ç. K. Koç *
Electrical and Computer Engineering
Oregon State University, ECE 220
Corvallis, Oregon 97331, USA

A. M. Apohan
TUBITAK MAM Research Centre
Department of Electronics, PK 21
Gebze, Kocaeli 41470, TURKEY

Abstract

We describe an algorithm for inverting an iteration of the one-dimensional cellular automaton. The algorithm is based on the linear approximation of the updating function, and requires less than exponential time for particular classes of updating functions and seed values. For example, an n-cell cellular automaton based on the updating function CA30 can be inverted in $O(n)$ time for certain seed values, and at most $2^{n/2}$ trials are required for arbitrary seed values. The inversion algorithm requires at most $2^{(q-1)(1-\alpha)n}$ trials for arbitrary nonlinear functions and seed values, where q is the number of variables of the updating function, and α is the probability of agreement between the function and its best affine approximation. The inversion algorithm coupled with the method of Meier and Staffelbach [6] becomes a powerful tool to cryptanalyze the random number generators based on one-dimensional cellular automata, showing that these random number generators provide less amount of security than their state size would imply.

Key Words: Random number generation, best affine approximation, cryptanalysis.

Consequences: Wolfram's PRNG is basically useless when instantiated with rule 30

Shortcoming 1 in CA Crypto

- Wolfram used only empirical and statistical tests for security analysis
- But statistical tests can be used as *necessary conditions*, so:

Shortcoming

Grounding security of CA-based primitives on statistical or empirical tests or criteria unrelated to cryptography (e.g., chaos-based properties) can be misleading.

Insight

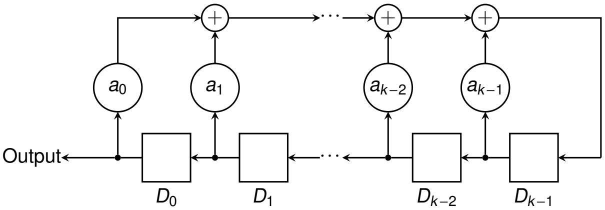
Statistical tests are fine only to filter out bad CA-based cryptographic primitives.
At least, the cryptographic properties of the local rules should be carefully investigated.

- How can we fix Wolfram's PRNG?

Linear Feedback Shift Registers (LFSR)

- Device computing the binary linear recurring sequence

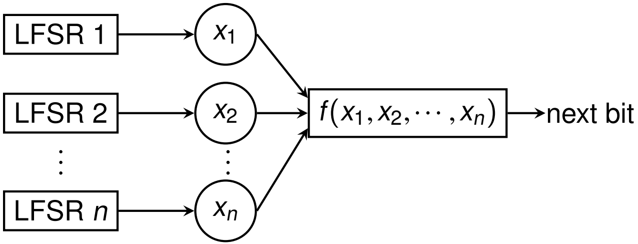
$$s_{n+k} = a + a_0 s_n + a_1 s_{n+1} + \cdots + a_{k-1} s_{n+k-1}$$



- Too weak as a PRG: $2k$ consecutive bits of keystream are enough to recover the LFSR initialization via the Berlekamp-Massey algorithm

The Combiner Generator

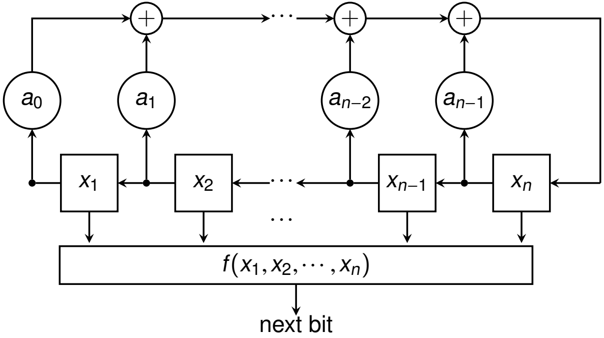
- Idea: use n LFSR in parallel, and combine their outputs with a *Boolean function* $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ of n variables [C21]



- The period of the combiner is at most the lcm of the periods of the n LFSR
- The function f must satisfy several **properties** to resist different attacks

The Filter Generator

- Idea: single LFSR of order n, but use the values of *all* flip-flops as inputs to a Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$| [C21]



- Equivalent to the combiner model with n copies of the same LFSR, but attacks work differently on the filter generator

Cryptographic Properties of Boolean Functions

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

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

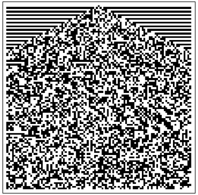
(x_1, x_2, x_3)	000	001	010	011	100	101	110	111
Ω_f	0	1	1	0	1	0	1	0
$W_f(a)$	0	-4	0	4	0	4	0	4

A **Boolean function** used in the combiner model should:

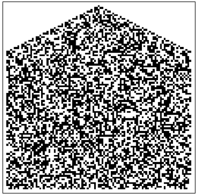
- be balanced
- have high nonlinearity $nl(F)$
- be correlation immune of high order t

Salvaging Wolfram's PRNG

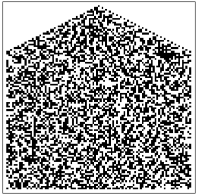
- **Problem** of rule 30: too small to give any meaningful cryptographic property [M08]
- Later works considered rules of larger diameters [L13, F14, L14]



(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.

Shortcoming 2 in CA Crypto

- Cryptographic properties are tailored for some PRNG models (combiner, filter, ...)
- But Wolfram's PRNG is not among them! So:

Shortcoming

Security claims for Wolfram-like PRNGs based on the cryptographic properties of the local rule are not enough:

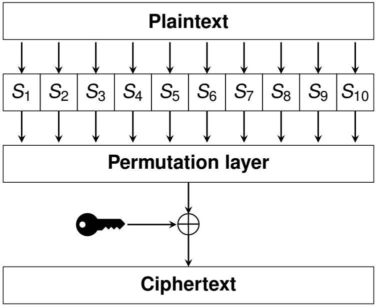
- *Attacks on the combiner or filter model might not be relevant in the CA setting*
- *Cryptographic properties might not capture other attacks unique to the CA model*

Insight

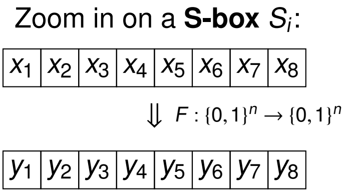
Consistently link the CA model with the security properties and the related attacks

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)

CA-based symmetric ciphers

The "**dynamical system**" way:

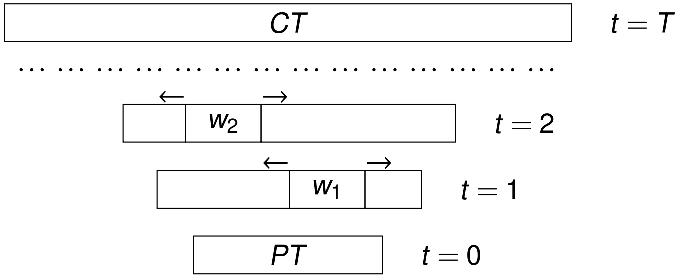
- Iterate a CA for **several** time steps to encrypt the **whole** plaintext
- Typically seen in CA venues [S04, M06, S08]
- Several weaknesses (low diffusion, ...)

The "**reductionist**" way:

- Iterate a CA for a **single** time step to encrypt a **part** of plaintext
- More common in crypto venues [P17a, G18, M19]
- In line with current state of the art

"Dynamical System" CA-Based Crypto: Gutowitz's Block Cipher [G93]

- **Diffusion**: iterative preimage computation of a permutive CA



- **Confusion:** Iteration of a partitioned (reversible) CA
- No formal security analysis so far

Shortcoming 3 in CA Crypto

- Gutowitz's cipher does not follow the SPN paradigm

Shortcoming

Using non-standard paradigms to design block ciphers hinders the security analysis. A general appeal to the confusion and diffusion principles is not a sound approach.

- But CA are simply vectorial Boolean functions, hence:

Insight

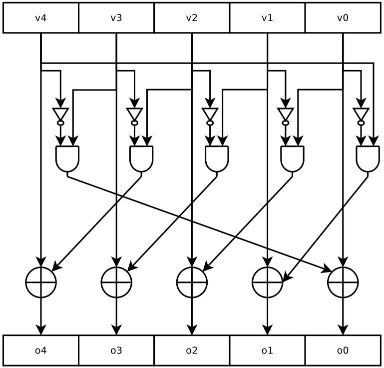
Prefer to work with well-established paradigms (e.g., SPN ciphers and sponges) and insert CA as building blocks inside them (e.g., as S-boxes)

"Reductionist" CA-Based Crypto: KECCAK χ 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



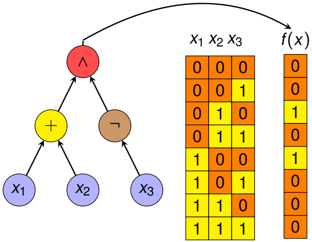
CA-based S-boxes

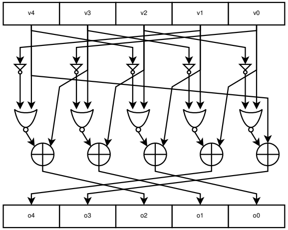
Algebraic approach:

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

Heuristic approach:

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





Can we use CA for everything?

- 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`

- But diffusion requires quick propagation!

Shortcoming 4 in CA Crypto

Shortcoming

CA are simply bad for diffusion

- Further motivation to work with established block cipher paradigms:

Insight

For certain components of a block cipher, it is better to abandon the CA approach. Non-local transformations are usually better, especially for the diffusion phase.

- Diffusion layers with CA: how many iterations do we need to reach good difference propagation?

Conclusions

Future Outlook

To sum up:

- CA are definitely useful for cryptography, but...
- ... need 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 4×4 S-Boxes from Cellular Automata Rules. IACR Trans. Symmetric Cryptol. 2018(3): 311-334 (2018)

[G93] H. Gutowitz: Cryptography with dynamical systems. In: Cellular Automata and Cooperative Systems, pp. 237—274 (1993)

[K97] C. Koc, A. Apohan: Inversion of cellular automata iterations. IEE Proc. Comput. Digit. Techniq. 144(5):279–284 (1997)

[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)

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

References (cont.)

- [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)
- [M08] B. Martin: A Walsh exploration of elementary CA rules. J. Cell. Autom. 3(2):145–156 (2008)
- [M91] W. Meier, O. Staffelbach: Analysis of pseudo random sequence generated by cellular automata. In: Proceedings 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)
- [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)
- [W86] S. Wolfram. Cryptography with cellular automata. In CRYPTO '85, pp. 429–432 (1986)