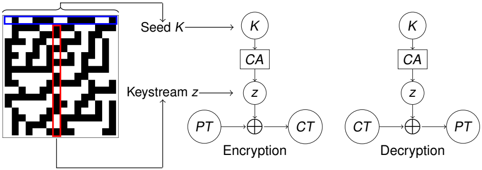
Radboud University

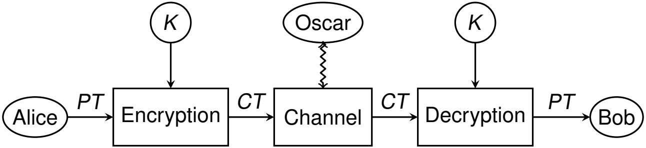
New Directions in AI-based Cryptography

Luca Mariot
https://lucamariot.org

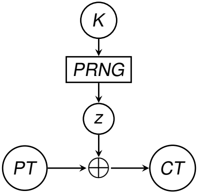
Dagstuhl Seminar – Intelligent Security
11 October 2022



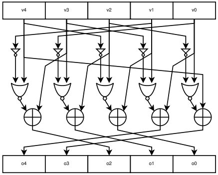
AI Methods for Symmetric Cryptography



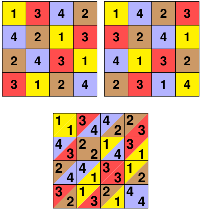
Symmetric ciphers require several low-level primitives, such as:



(a) Pseudorandom Generators



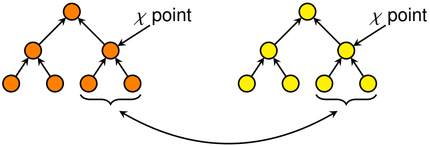
(b) Boolean functions and S-boxes



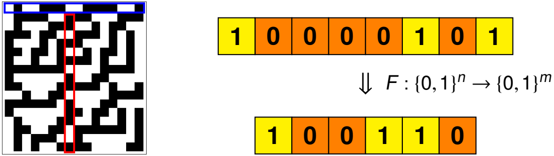
(c) Latin Squares and Orthogonal Arrays

AI approach for symmetric crypto

- "Traditional" approach: ad-hoc and algebraic constructions
- "AI" approach: support the designer using AI methods:
  - Optimization (Evolutionary algorithms, swarm intelligence...)



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



Genetic Algorithms (GA) & Genetic Programming (GP)

- **Black-box optimization** of a fitness function [L15]
- Work on a **coding** of the solutions

- **GA Encoding**: bitstrings

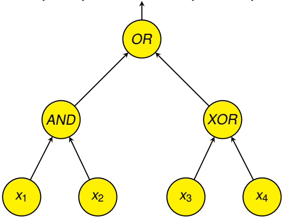
0	**1**	**1**	**1**	**1**	**0**	**0**	**0**

$$\Downarrow$$

$$f(x_1, x_2, x_3) = x_1 \cdot x_2 \oplus x_1 \oplus x_2 \oplus x_3$$

- **GP Encoding**: trees

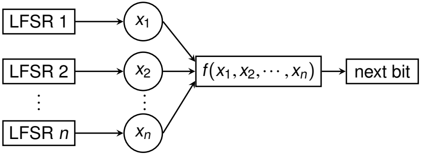
$$f(x_1, x_2, x_3, x_4) = (x_1\ AND\ x2)\ OR\ (x_3\ XOR\ x_4)$$



Use of EA in symmetric cryptography

Design of primitives as **combinatorial optimization problems**, examples [C21, M22]:

- Boolean functions $f : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ for stream ciphers



- S-Boxes $F : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ for block ciphers

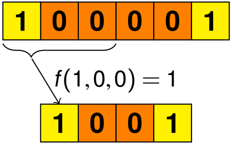
Possible advantages of using EA for this search [P16, M19b]:

- Diversity of solutions, due to the "blindness" of EA
- Flexibility of EA (optimizing several properties at once

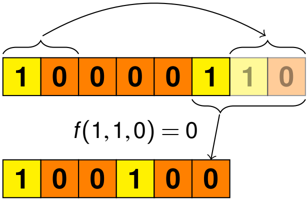
Cellular Automata

- One-dimensional Cellular AutomatA (CA):

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



No Boundary CA

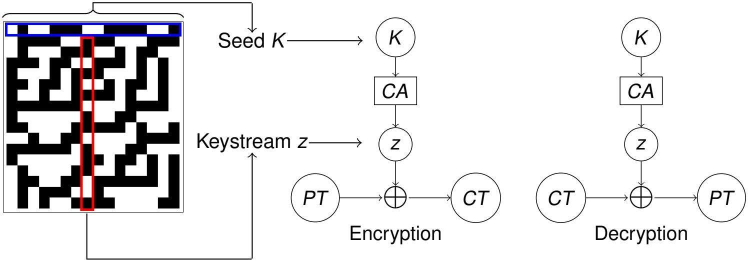


Periodic Boundary CA

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

Cellular Automata and Cryptography

Goal: investigate how CA can be used in the design of cryptographic primitives [W86, L13]

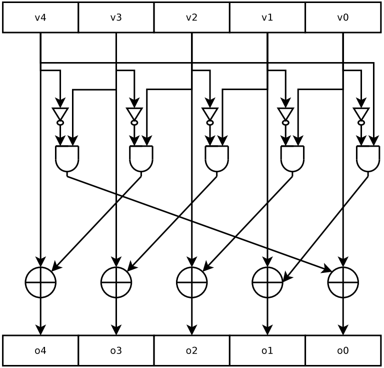


Why CA?

1. **Security from Complexity**
2. **Efficient Implementation**

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

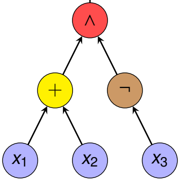


- Used as a PBCA with $n = 5$ in Keccak [B11]

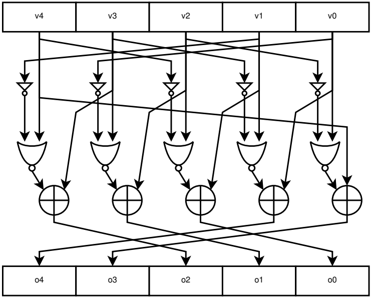
CA S-boxes found by GP

Idea: evolve a CA rule that defines an S-box, optimizing:

- **crypto** properties (nonlinearity, differential uniformity) [M19a]
- **implementation** properties (area, latency)



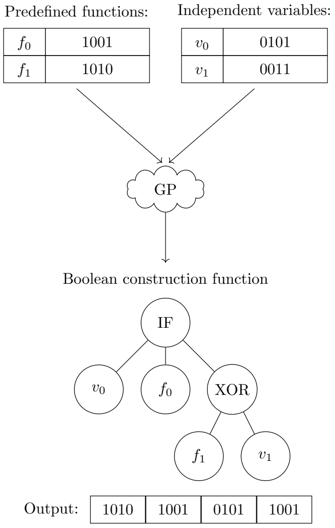
x_1	x_2	x_3	$f(x)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



- Up to size 7×7: results on par or slightly better than the state of the art (Keccak, PRESENT, Piccolo, ...) [P17]

New Direction 1:
Evolve constructions of crypto primitives

Evolving Constructions of Boolean functions with GP

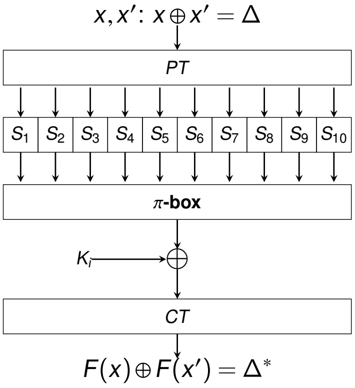


- **Idea:** Do not evolve primitives directly, but rather their mathematical constructions [C22]
- Use Boolean minimizers to interpret the constructions
- **Research Question**: Does GP obtain previously known constructions or new ones?

New Direction 2: Evolutionary-based distinguishers

Differential Cryptanalysis

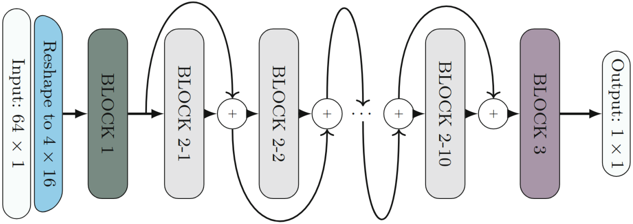
- **Idea**: chosen plaintext attack, see how differences propagate to the ciphertext



- **Goal**: Compute differential probability of $\Delta \to \Delta^*$
- **Distinguishing attack**: given (x, x'), classify if it is a *random* or *real* pair
- **Tool**: Difference Distribution Table (DDT)

Deep learning-based differential distinguishers

- A. Gohr (CRYPTO 2019): train a CNN as a differential distinguisher
- Better accuracy than pure distinguishers on SPECK32/64



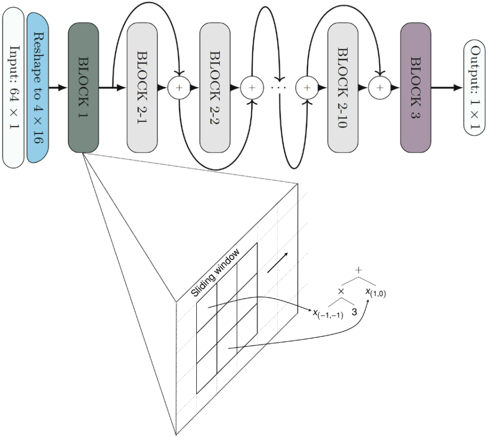
[1]

- **Problem**: learned models are hardly interpretable!

[1] Image credits: A. Benamira et al., *A Deeper Look at Machine Learning-Based Cryptanalysis*, EUROCRYPT 2021

New Direction 2: GP-based distinguishers

- **Idea**: Replace convolutional layers with convolutional GP [J21]



- **Research Question**: Is "convolutional" GP able to reach CNN performances, and yield models easier to interpret?

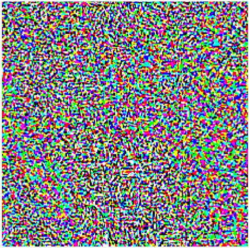
New Direction 3: Evolutionary approach to adversarial examples

Adversarial Examples in DNN

- DNN known to be vulnerable to **adversarial examples** (AE)
- **Idea**: perturb a valid example to mess the DNN's classification

Classification: Panda

+



Noise perturbation

$\Rightarrow$

Classification: Gibbon

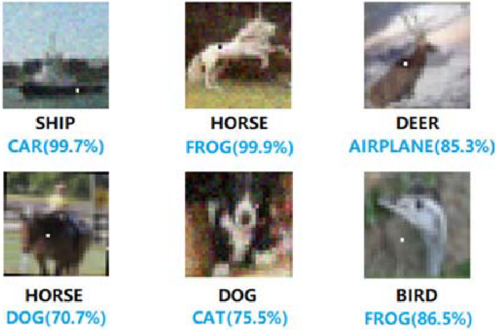
[2]

- Perturbation moves the example beyond the *decision boundary* of a DNN

[2] Example credits: I.J. Goodfellow, J. Shlens, C. Szegedy, *Explaining and Harnessing Adversarial Examples*, ICLR 2015

Evolutionary Construction of AE

- Perturbations for AE can be **minimal**
- **One-pixel attack**: Modify just one pixel in a valid example



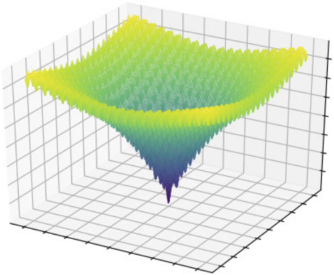
[3]

- Pixel selection done with **Evolutionary Algorithms**

[3] Image credit: J. Su et al., *One Pixel Attack for Fooling Deep Neural Networks*. IEEE Trans. Evol. Comput 23(5):828-840 (2019)

New Direction 2: LON Analysis of Loss Landscapes

- **Idea**: use fitness landscape analysis on the space of AE
- **Approach**: continuous variant of Local Optima Networks



$\Rightarrow$

[4]

Research Questions:

- Is it possible to improve EA-based one-pixel attacks?
- Gain insights to build more robust DNN?

---

[4] Image credit: J. Adair et al., *Local Optima Networks for Continuous Fitness Landscapes*. In: GECCO'21 (Companion), pp.1407-1414. ACM (2019)

Summary

Where we arrived so far:

- Evolutionary algorithms and CA give interesting alternatives for the design of symmetric primitives
- Flexibility of optimization objectives

Looking at the future:

- Plenty of open problems in the design research thread, **but**...
  ... mainly of mathematical interest
- Leverage on the interpretability of evolutionary models for cybersecurity applications

References

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

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

[C22] C. Carlet, M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek: Evolving constructions for balanced, highly nonlinear boolean functions. Proceedings of GECCO 2022, pp. 1147-1155 (2022)

[J21] D. Jakobovic, L. Manzoni, L. Mariot, S. Picek, M. Castelli: CoInGP: convolutional inpainting with genetic programming. Proceedings of GECCO 2021, pp. 795-803 (2021)

[L13] A. Leporati and L. Mariot: 1-Resiliency of Bipermutive Cellular Automata Rules. Proceedings of Automata 2013, pp. 110-123 (2013)

[L15] S. Luke. Essentials of Metaheuristics. Lulu, 2015. 2nd ed.

[M22] L. Mariot, D. Jakobovic, T. Bäck, J. Hernandez-Castro: Artificial Intelligence for the Design of Symmetric Cryptographic Primitives. Security and Artificial Intelligence 2022, pp. 3-24 (2022)

[M19a] L. Mariot, S. Picek, A. Leporati, and D. Jakobovic. Cellular automata based S-boxes. Cryptography and Communications 11(1):41–62 (2019)

[M19b] L. Mariot, D. Jakobovic, A. Leporati, S. Picek: Hyper-bent Boolean Functions and Evolutionary Algorithms. Proceedings of EuroGP 2019, pp. 262-277 (2019)

[P16] S. Picek, D. Jakobovic, J.F. Miller, L. Batina, M. Cupic: Cryptographic Boolean functions: One output, many design criteria. Appl. Soft Comput. 40: 635-653 (2016)

[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)

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