Cryptography by Cellular Automata

Luca Mariot

luca.mariot@disco.unimib.it

Zagreb – November 14, 2017
One-dimensional Cellular Automaton (CA): a discrete parallel computation model composed of a finite array of $n$ cells

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 to its right.

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

Truth table: $\Omega(f) = 01101001 \rightarrow $ Rule 150

No Boundary CA – NBCA

Periodic Boundary CA – PBCA
Basic Goal of Cryptography: Enable two parties (Alice and Bob, A and B) to securely communicate over an insecure channel, even in presence of an opponent (Oscar, O)

- **Alice** → Encryption → Channel → Decryption → **Bob**

- **PT**: plaintext
- **CT**: ciphertext
- **$K_E$**: encryption key
- **$K_D$**: decryption key

Luca Mariot

Cryptography by Cellular Automata
CA-based Crypto History: Wolfram’s PRNG

- **General Idea**: exploit the emergent complexity of CA to design cryptosystems satisfying confusion and diffusion criteria [Shannon49]

- **CA-based Pseudorandom Generator (PRG)** [Wolfram86]: central cell of rule 30 CA used as a stream cipher keystream

![Diagram of CA-based PRNG]

- This CA-based PRNG was later shown to be vulnerable [Meier91]
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 [Daemen94]

Used as a PBCA with $n = 5$ in the Keccak specification of SHA-3 standard [Keccak11]
Research Goal: investigate the cryptographic properties and the combinatorial designs induced by CA to realize significant cryptographic schemes.

What do we mean by “significant”?

1. **Secure**: Satisfying strong security properties
2. **Efficient**: Leveraging CA parallelism for efficient hardware-oriented cryptography

Main focus: Security aspect
Summary of Contributions

Research lines investigated up to now:

- **Line 1**: CA cryptographic properties
  - Bounds on the nonlinearity and differential uniformity of CA-based S-boxes
  - CA Cryptographic properties optimization through Genetic Programming (GP)

- **Line 2**: Secret sharing schemes based on CA
  - Orthogonal Latin Squares (OLS) from linear CA
  - Evolutionary search of nonlinear CA generating OLS
Research Line 1: CA cryptographic properties
CA-based cipher design

**Design principle**: the CA used in cryptographic primitives must satisfy certain properties, to thwart particular attacks

State of the art, up to now:

\[
\begin{array}{cccccccc}
\cdots & 0 & 1 & 1 & 0 & 0 & \cdots \\
\downarrow & f : \{0,1\}^d & \rightarrow & \{0,1\} \\
0 & & & & & & & \\
\end{array}
\]

- Focus on CA local rules, viewed as **Boolean functions**
- Rationale: choose rule \( f \) with best crypto properties

Our approach:

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
\downarrow & F : \{0,1\}^n & \rightarrow & \{0,1\}^m \\
1 & 0 & 0 & 1 & 1 & 0 & & \\
\end{array}
\]

- Some attacks cannot be formalized in a local way
- **Idea**: Analyze the CA global rule as a S-box
Research Line 1: CA cryptographic properties

Contribution 1: Bounds on the nonlinearity and differential uniformity of CA-based S-boxes
Nonlinearity of Boolean Functions

- **Linear Boolean function** $L_\omega : \{0, 1\}^n \rightarrow \{0, 1\}$:

  $$L_\omega(x) = \omega \cdot x = \omega_1 x_1 \oplus \cdots \oplus \omega_n x_n$$

- **Nonlinearity** of $f : \{0, 1\}^n \rightarrow \{0, 1\}$: minimum Hamming distance of $f$ from the set of all linear functions:

  $$N_f = 2^{n-1} - \frac{1}{2} \left( |W_{\text{max}}(f)| \right)$$

  where $W_{\text{max}}(f)$ is the maximum absolute value of the Walsh transform of $f$:

  $$W_f(\omega) = \sum_{x \in \{0, 1\}^n} (-1)^{f(x) \oplus \omega \cdot x}$$
Nonlinearity of S-boxes

- A Substitution Box (S-box) is a mapping $F : \{0, 1\}^n \rightarrow \{0, 1\}^m$ defined by $m$ coordinate functions $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$

- The component functions $\nu \cdot F : \{0, 1\}^n \rightarrow \{0, 1\}$ for $\nu \in \{0, 1\}^m$ of $F$ are the linear combinations of the $f_i$

\[
\begin{array}{cccccccc}
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 \\
\downarrow & & & & & & & \\
F : \{0, 1\}^n \rightarrow \{0, 1\}^m
\end{array}
\]

\[
\begin{array}{cccccccc}
f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\
\end{array}
\]

\[
f_1 \oplus f_3 \oplus f_5
\]

- The nonlinearity of a S-box $F$ is defined as the minimum nonlinearity among all its component functions

- S-boxes with high nonlinearity allow to resist to linear cryptanalysis attacks
Differential Uniformity of S-boxes

- **delta difference table of** $F$ wrt $a, b$:

$$D_F(a, b) = \left\{ x \in \mathbb{F}_2^n : F(x) \oplus F(x \oplus a) = b \right\}.$$

- Given $\delta_F(a, b) = |D_F(a, b)|$, the **differential uniformity** of $F$ is:

$$\delta_F = \max_{a \in \{0, 1\}^n} \max_{b \in \{0, 1\}^m} \delta_F(a, b).$$

- S-boxes with low differential uniformity are able to resist differential cryptanalysis attacks.
We proved the following upper bounds for NBCA and PBCA:

**Theorem**

The nonlinearity and differential uniformity of the S-box $F$ of an $n$-cell NBCA or PBCA with local rule $f : \{0, 1\}^d \to \{0, 1\}$ satisfy

$$N_F \leq 2^{n-d} \cdot N_f$$

$$\delta_F \leq 2^{n-d} \cdot \delta_f$$

**Remark**: This explains why adding cells to a CA makes the cryptographic properties of the S-box worse (see e.g. **Keccak**).
Research Line 1: CA cryptographic properties

Contribution 2: CA Cryptographic properties optimization through \textbf{Genetic Programming} (GP)

(Joint work with Stjepan Picek and Domagoj Jakobovic)
Goal: Find PBCA of length $n$ and diameter $d = n$ having cryptographic properties equal to or better than those of other real-world S-boxes (e.g. Keccak, ...)

Considered S-boxes sizes: from $n = 4$ to $n = 8$

Using tree encoding, exhaustive search is already unfeasible for $n = 4$

We adopted an evolutionary heuristic – Genetic Programming
Genetic Programming (GP)

- Optimization method inspired by evolutionary principles, introduced by Koza [Koza93]
- Each candidate solution (individual) is represented by a tree
  - Terminal nodes: input variables
  - Internal nodes: Boolean operators (AND, OR, NOT, XOR, ...)
- New solutions are created through genetic operators like tree crossover and subtree mutation applied to a population of candidate solutions
- Optimization is performed by evaluating the new candidate solutions wrt a fitness function
GP Tree Encoding – Example

\[ f(x_1, x_2, x_3, x_4) = (x_1 \text{ OR } x_2) \text{ OR } (x_3 \text{ XOR } x_4) \]
Considered cryptographic properties:

- balancedness/invertibility ($BAL = 0$ if $F$ is balanced, $-1$ otherwise)
- nonlinearity $N_F$
- differential uniformity $\delta_F$

Fitness function maximized:

$$fitness = BAL + \Delta_{BAL,0} \left( N_F + \left( 1 - \frac{n\text{Min}N_F}{2^n} \right) + (2^n - \delta_F) \right).$$

where $\Delta_{BAL,0} = 1$ if $F$ is balanced and 0 otherwise, and $n\text{Min}N_F$ is the number of occurrences of the current value of nonlinearity.
Experimental Setup

- Problem instance / CA size: $n = 4$ up to $n = 8$
- Maximum tree depth: equal to $n$
- Genetic operators: simple tree crossover, subtree mutation
- Population size: 2000
- Stopping criterion: 2000000 fitness evaluations
- Parameters determined by initial tuning phase on $n = 6$ case
Results – Crypto Properties

Table: Statistical results and comparison.

<table>
<thead>
<tr>
<th>S-box size</th>
<th>$T_{max}$</th>
<th>GP</th>
<th>$N_F$</th>
<th>$\delta_F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4 \times 4$</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>$5 \times 5$</td>
<td>42</td>
<td>42</td>
<td>41.73</td>
<td>1.01</td>
</tr>
<tr>
<td>$6 \times 6$</td>
<td>86</td>
<td>84</td>
<td>80.47</td>
<td>4.72</td>
</tr>
<tr>
<td>$7 \times 7$</td>
<td>182</td>
<td>182</td>
<td>155.07</td>
<td>8.86</td>
</tr>
<tr>
<td>$8 \times 8$</td>
<td>364</td>
<td>318</td>
<td>281.87</td>
<td>13.86</td>
</tr>
</tbody>
</table>

- From $n = 4$ to $n = 7$, we obtained CA rules inducing S-boxes with optimal crypto properties.
- Only for $n = 8$ the performances of GP are consistently worse wrt to the theoretical optimum.
Research Line 2: CA-based secret sharing schemes
Secret Sharing Schemes

- Secret sharing scheme (SSS): a procedure enabling a dealer to share a secret $S$ among a set $\mathcal{P}$ of $n$ players
- $(k, n)$ threshold SSS: at least $k$ players to recover $S$

Example: $(2, 3)$–scheme
State of the art CA-based SSS

- All CA-based SSS (e.g. [Mariot14]) have a sequential threshold, where shares must be adjacent.

(a) Sequential threshold CA SSS

(b) Period of spatially periodic preimage

Question: Is it possible to design a CA-based threshold SSS without adjacency constraint?
Summary of Contributions

Research Line 2: CA-based secret sharing schemes

Contribution 1: Generating Orthogonal Latin Squares (OLS) through Linear CA
A Latin square (LS) is a $N \times N$ matrix where each row and each column permutes $[N] = \{1, \cdots, N\}$

$L_1, \cdots, L_n$ are mutually orthogonal ($n$-MOLS) if their pairwise superposition yields all the pairs $(x, y) \in [N] \times [N]$.

\[
\begin{array}{cccc}
1 & 3 & 4 & 2 \\
4 & 2 & 1 & 3 \\
2 & 4 & 3 & 1 \\
3 & 1 & 2 & 4 \\
\end{array}
\]

(a) $L_1$

\[
\begin{array}{cccc}
1 & 4 & 2 & 3 \\
3 & 2 & 4 & 1 \\
4 & 1 & 3 & 2 \\
2 & 3 & 4 & 1 \\
\end{array}
\]

(b) $L_2$

\[
\begin{array}{cccc}
1,1 & 3,4 & 4,2 & 2,3 \\
4,3 & 2,2 & 1,4 & 3,1 \\
2,4 & 4,1 & 3,3 & 1,2 \\
3,2 & 1,3 & 2,1 & 4,4 \\
\end{array}
\]

(c) $(L_1, L_2)$

Remark: $n$-MOLS $\Leftrightarrow (2, n)$ threshold SSS
**Latin Squares through Bipermutive CA (1/2)**

- **Idea:** determine which CA induce orthogonal Latin squares
- **Bipermutive CA:** local rule $f$ is defined as
  
  $$f(x_1, \cdots, x_{2r+1}) = x_1 \oplus g(x_2, \cdots, x_{2r}) \oplus x_{2r+1}$$

**Lemma**

Let $F$ be a $m$-cell bipermutive NBCA with diameter $d$ s.t. $(d - 1)|m$. Then, the CA generates a Latin square of order $N = 2^m$
Example: \( \langle \mathbb{F}_2, 4, 1, f \rangle \), \( f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \) (Rule 150)

Encoding: 
- 00 \( \mapsto \) 1
- 10 \( \mapsto \) 2
- 01 \( \mapsto \) 3
- 11 \( \mapsto \) 4

### (a) Rule 150 on 4 bits

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>01</td>
<td>11</td>
</tr>
<tr>
<td>01</td>
<td>11</td>
<td>00</td>
<td>10</td>
</tr>
</tbody>
</table>

### (b) Latin square \( L_{150} \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Local rule: *linear combination* of the neighborhood cells

\[ f(x_1, \cdots, x_d) = a_1 x_1 \oplus \cdots \oplus a_d x_d , \ a_i \in \mathbb{F}_2 \]

Associated polynomial:

\[ f \mapsto \varphi(X) = a_1 + a_2 X + \cdots + a_d X^{d-1} \]

Global rule: \( m \times (m + d - 1) \) \((d - 1)\)-diagonal *transition matrix*

\[
M_F = \begin{pmatrix}
        a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
        0 & a_1 & \cdots & a_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
        \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
        0 & \cdots & \cdots & \cdots & \cdots & 0 & a_1 & \cdots & a_d
\end{pmatrix}
\]

\[ x = (x_1, \cdots, x_n) \mapsto M_F x^\top \]
Orthogonal Latin Squares by Linear CA

Theorem

Let $F, G$ be linear bipermutive NBCA. The Latin squares induced by $F$ and $G$ are orthogonal if and only if $P_f(X)$ and $P_g(X)$ are coprime.

Figure:

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

Luca Mariot
Cryptography by Cellular Automata
Number of coprime polynomial pairs of degree $n$ and nonzero constant term:

$$a(n) = 4^{n-1} + a(n-1) = 4^{n-1} - 1 = \frac{4^{n-1} - 1}{3} = 0, 1, 5, 21, 85, ...$$

This sequence corresponds to OEIS A002450, which has several other interpretations (e.g. Collatz conjecture, ...)

https://xkcd.com/710/
Summary of Contributions

Research Line 2: CA-based secret sharing schemes

Contribution 2: Evolutionary search of nonlinear CA generating OLS

(Joint work with Stjepan Picek and Domagoj Jakobovic)
Motivations and Goals

- Construction of OLS solved for linear CA [Mariot16]
- MOLS arising from nonlinear constructions have relevance in cheater-immune Secret Sharing Schemes [Tompa88]

**Goal:** Design OLS based on CA by evolving pairs of nonlinear bipermutive local rules through GA and GP

Twofold motivation:

- **Theoretical:** Understand the mathematical structure of the space of nonlinear CA-based OLS
- **EC perspective:** Source of new problems for evolutionary algorithms
Search Space Size

- Number of Boolean functions of \( n \) variables: \( \mathcal{F}_n = 2^{2^n} \)
- Bipermutive rules of size \( n \) ⇔ Generating functions of size \( n - 2 \) (which are \( \mathcal{F}_{n-2} = 2^{2^{n-2}} \))
- Pairs of bipermutive rules of size \( n \): \( \mathcal{B}_n = 2^{2^{n-1}} = \mathcal{F}_{n-1} \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{B}_n )</td>
<td>16</td>
<td>256</td>
<td>65536</td>
<td>( \approx 4.3 \times 10^9 )</td>
<td>( \approx 1.8 \cdot 10^{19} )</td>
<td>( \approx 3.4 \cdot 10^{38} )</td>
</tr>
<tr>
<td>( N \times N )</td>
<td>4 × 4</td>
<td>8 × 8</td>
<td>16 × 16</td>
<td>32 × 32</td>
<td>64 × 64</td>
<td>128 × 128</td>
</tr>
<tr>
<td>#OLS</td>
<td>8</td>
<td>72</td>
<td>1704</td>
<td>533480</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

**Remark:** Exhaustive enumeration possible up to \( n = 6 \)
Fitness Functions (1/2)

- \#rep(L_1, L_2): Number of occurrences of each pair (except the first one) in the superposition of Latin squares \(L_1\) and \(L_2\)

\[
\begin{array}{cccc}
1 & 3 & 4 & 2 \\
4 & 2 & 1 & 3 \\
2 & 4 & 3 & 1 \\
2 & 3 & 4 & 1 \\
\end{array}
\quad
\begin{array}{cccc}
1 & 4 & 3 & 2 \\
2 & 3 & 4 & 1 \\
4 & 1 & 2 & 3 \\
3 & 2 & 1 & 4 \\
\end{array}
\quad
\begin{array}{cccc}
4,1 & 1,4 & 2,3 & 3,2 \\
3,2 & 2,3 & 1,4 & 4,1 \\
1,4 & 4,1 & 3,2 & 2,3 \\
2,3 & 3,2 & 4,1 & 1,4 \\
\end{array}
\]

(a) \(L_1\)  (b) \(L_2\)  (c) \#rep(L_1, L_2) = 12

- Let \(\varphi, \gamma\) be the generating functions of two bipermutive CA, and let \(L_\varphi, L_\gamma\) be the associated Latin squares.

**First fitness function**: minimize \(fit_1(\varphi, \gamma) = \#rep(L_\varphi, L_\gamma)\)
Remark: $fit_1$ does not consider the nonlinearity of $\varphi$ and $\gamma$!

Nonlinearity penalty factor:

$$NlPen(\varphi, \gamma) = \begin{cases} 
0, & \text{if } Nl(\varphi) > 0 \text{ AND } Nl(\gamma) > 0 \\
1, & \text{if } Nl(\varphi) = 0 \text{ XOR } Nl(\gamma) = 0 \\
2, & \text{if } Nl(\varphi) = 0 \text{ AND } Nl(\gamma) = 0 
\end{cases}$$

Second fitness function: minimize

$$fit_2(\varphi, \gamma) = \#\text{rep}(L_\varphi, L_\gamma) + NlPen(\varphi, \gamma) \cdot N^2$$

The $N^2$ scaling factor balances the range of $\#\text{rep}(L_\varphi, L_\gamma)$, which is $\{0, \cdots, N^2\}$
Let $\phi, \gamma : \{0, 1\}^{n-2} \rightarrow \{0, 1\}$ be a pair of generating functions, with $2^{n-2}$-bit truth tables $\Omega(\phi), \Omega(\gamma)$, and let $||$ denote concatenation.

**First GA encoding:** $enc_1(\phi, \gamma) = \Omega(\phi)||\Omega(\gamma)$

Example:

$\phi(x_1, x_2, x_3) = x_1 \oplus x_3 \Rightarrow \Omega(\phi) = (0, 1, 0, 1, 1, 0, 1, 0)$

$\gamma(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3 \Rightarrow \Omega(\gamma) = (0, 1, 1, 0, 1, 0, 0, 1)$

$enc_1(\phi, \gamma) = (0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1)$

Classic GA variation operators like one-point crossover and bit-flip mutation are applied in this case.
GA & GP Encodings: Double Bitstring/Double Tree

- **Idea:** Keep the generating functions separated and evolve them independently

  **Second GA encoding:** $\text{enc}_2(\varphi, \gamma) = (\Omega(\varphi), \Omega(\gamma))$

- We use the same idea for GP: the genotype is composed of the two trees $T(\varphi)$ and $T(\gamma)$ representing $\varphi$ and $\gamma$

  **GP encoding:** $\text{enc}_{GP}(\varphi, \gamma) = (T(\varphi), T(\gamma))$

- Classic GA and GP variations operators are applied independently on each of the two components
Definition

\(f, g : \{0, 1\}^n \to \{0, 1\}\) are **pairwise balanced** (PWB) if

\[
\left| (f, g)^{-1}(0, 0) \right| = \left| (f, g)^{-1}(1, 0) \right| = \\
\left| (f, g)^{-1}(0, 1) \right| = \left| (f, g)^{-1}(1, 1) \right| = 2^{n-2}
\]

Example:

- \(f(x_1, x_2, x_3) = x_1 \oplus x_3\) (Rule 90)
- \(f(x_1, x_2, x_3) = x_1 \oplus x_2 \oplus x_3\) (Rule 150)

\[
\Omega(f) = (0, 1, 0, 1, 1, 0, 1, 0) \\
\Omega(g) = (0, 1, 1, 0, 1, 0, 0, 1)
\]

Each of the pairs \((0, 0), (1, 0), (0, 1), (1, 1)\) occurs \(2^{3-2} = 2\) times
Experimental observations on exhaustive search:
- Two bipermutive CA generate OLS $\Rightarrow$ the local rules are PWB
- Generating functions are PWB $\Rightarrow$ the local rules are PWB

**Third GA encoding:** $enc_3(\varphi, \gamma)$ is a quaternary string of length $2^{n-2}$ where each number from 1 to 4 occurs $2^{n-4}$ times

Example: $n = 5, (0, 0) \mapsto 1, (1, 0) \mapsto 2, (0, 1) \mapsto 3, (1, 1) \mapsto 4$

$$\Omega(\varphi) = (0, 1, 0, 1, 1, 0, 1, 0)$$
$$\Omega(\gamma) = (0, 1, 1, 0, 1, 0, 0, 1)$$
$$enc_3(\varphi, \gamma) = (1, 4, 3, 2, 4, 1, 2, 3)$$

**Balancedness-preserving variation operators for GA:**
- **Crossover:** use counters to keep track of the multiplicities of the 4 values in the offspring
- **Mutation:** use a swap-based operator
Experimental settings

Common Parameters:
- Problem instances: rules of $n = 7$ and $n = 8$ variables
- Termination condition: 300000 fitness evaluations
- Each experiment is repeated over 50 independent runs
- Selection operator: steady-state with 3-tournament operator

GA Parameters:
- Population size: 30 individuals
- Crossover and mutation probabilities: $p_c = 0.95$, $p_m = 0.2$

GP Parameters:
- Boolean operators: AND, OR, XOR, XNOR, NOT, IF
- Population size: 500 individuals
- Mutation probability: $p_m = 0.5$
Results

- \((GA, n, enc_i)\): GA experiment with CA rules of \(n\) variables and encoding \(enc_i\), fitness function \(fit_1\)
- \((GP, n, fit_i)\): GP experiment with CA rules of \(n\) variables and encoding \(enc_{GP}\), fitness function \(fit_i\)

<table>
<thead>
<tr>
<th>Exp.</th>
<th>avg fit</th>
<th>std fit</th>
<th>#opt</th>
<th>#lin</th>
<th>#nlin</th>
</tr>
</thead>
<tbody>
<tr>
<td>((GA, 7, enc_1))</td>
<td>520.32</td>
<td>360.16</td>
<td>12/50</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>((GA, 7, enc_2))</td>
<td>565.44</td>
<td>389.03</td>
<td>15/50</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>((GA, 7, enc_3))</td>
<td>392.64</td>
<td>328.47</td>
<td>18/50</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>((GA, 8, enc_1))</td>
<td>4165.44</td>
<td>604</td>
<td>1/50</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>((GA, 8, enc_2))</td>
<td>4222.16</td>
<td>125.03</td>
<td>0/50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>((GA, 8, enc_3))</td>
<td>4696.48</td>
<td>135.51</td>
<td>0/50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>((GP, 7, fit_1))</td>
<td>0</td>
<td>0</td>
<td>50/50</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td>((GP, 7, fit_2))</td>
<td>0</td>
<td>0</td>
<td>50/50</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td>((GP, 8, fit_1))</td>
<td>0</td>
<td>0</td>
<td>50/50</td>
<td>47</td>
<td>3</td>
</tr>
<tr>
<td>((GP, 8, fit_2))</td>
<td>0</td>
<td>0</td>
<td>50/50</td>
<td>0</td>
<td>50</td>
</tr>
</tbody>
</table>
Discussion

For GP:

▶ GP always manages to converge to an optimal solution
▶ ... but under $fit_1$, all solutions found are linear!
▶ Possible explanation: GP first converges to linear pairs (since it has the XOR operator), then OLS are easily found

On the other hand, for GA:

▶ GA converged just once for $n = 8$ and the performances for $n = 7$ are worse than GP
▶ ... but all solutions found are nonlinear, even under $fit_1$
Conclusions
We investigated two applications of CA to cryptography, namely:

▶ Design of CA-based S-boxes:
  ▶ Study of the bounds on nonlinearity and differential uniformity of S-boxes generated through CA
  ▶ Evolutionary search of CA-based S-boxes with good crypto properties through GP

▶ Design of CA-based Secret Sharing Schemes:
  ▶ Characterization of OLS generated by linear CA
  ▶ Evolutionary search of nonlinear CA generating OLS
Future developments

Research Line 1:
- Consider CA with respect to cryptographic properties related to other kinds of attacks (algebraic attacks, ...)
- Prove lower bounds on the nonlinearity of CA induced by specific classes of rules (bipermutable rules, plateaued functions, ...)

Research Line 2:
- Investigate the behavior of GP in evolving CA generating OLS
- Generalize to higher thresholds (via orthogonal arrays)


