# Constructing Cryptographic Primitives with Evolutionary Computation and Cellular Automata

Rocco Ascone[1], Firas Ben Ramdhane[2], Luca Manzoni[1,*], Giuliamaria Menara[2] and Gloria Pietropolli[1]

[1]*Dipartimento di Matematica, Informatica e Geoscienze, Università degli Studi di Trieste, Via Alfonso Valerio 12/1, 34127 Trieste, Italy*

[2]*Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy*

### Abstract

Finding Boolean functions with the necessary properties to be used as building blocks in cryptographic structures, like stream and block ciphers, is a complex combinatorial optimization problem. As such, there cannot be a single "silver bullet", but an investigation from multiple point of views, combining theoretical investigations, indirect construction, and search using metaheuristics, like evolutionary method, is the most promising approach. In this work we present different approaches currently being developed to tackle this complex optimization problem.

### Keywords

Evolutionary Computation, Reaction Systems, Cellular Automata, Combinatorial Designs, Boolean Functions, Cryptography

## 1. Introduction

The design of Boolean function with strong cryptographic properties is a complex task, since they must satisfy multiple possibly conflicting requirements, like nonlinearity, balancedness, and correlation immunity [1, 2]. These function are essential inside both block and stream ciphers in order to make them resistant to various kind of attack. However, there is no way to synthesize a function with all the desired properties in a deterministic and fast way: the search for such functions is a combinatorial optimization problem that, due to the size of the search space (which grows super-exponentially with the number of input bits) is still an active area of research [3, 4, 5, 6, 7].

For this reason, we will briefly present two different avenues of research currently being pursued in order to improve our ability to generate cryptographically robust Boolean functions:

1. The use of evolutionary methods based on Reaction Systems (RS), a bio-inspired computational model, to directly generate Boolean Functions;
2. A theoretical approach to the study of non-linearity and composition in CA.

The description of these investigations is necessarily limited in space, but we consider important to present both approaches, since we plan to integrate the results from the theoretical side as heuristics to guide the evolutionary search. For example, if a theorem proves that under certain conditions a property is not possible, then we can reduce the search space, simplifying the problem.

The paper is organized as follows: In Section 2 we introduce evolutionary reaction systems for the construction of Boolean functions, while in Section 3 we introduce the current theoretical results on CA. Finally, in Section 4 we provide the directions to unify the two research lines and provide a theoretically guided evolutionary method for cryptographic primitives via CA.

*Ital-IA 2025: 5th National Conference on Artificial Intelligence, organized by CINI, June 23-24, 2025, Trieste, Italy*

*Corresponding author.

rocco.ascone@phd.units.it (R. Ascone); firas.benramdhane@unimib.it (F. Ben Ramdhane); lmanzoni@units.it (L. Manzoni); giuliamaria.menara@unimib.it (G. Menara); gloria.pietropolli@units.it (G. Pietropolli)

0000-0001-6312-7728 (L. Manzoni)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2025-12-07

## 2. Evolution of Boolean Functions with Reaction Systems

In a recent work [8] we introduced the use of Evolutionary Reaction System [9, 10] for the construction of Boolean functions with good cryptographic properties. In this section we will introduce the problem, the evolutionary approach, and the obtained results.

### 2.1. Boolean Functions and Cryptographic Properties

Here we introduce some essential properties for cryptographically strong Boolean functions.

Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field of two elements, where addition corresponds to XOR and multiplication to AND. A Boolean function is a function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$, which can naturally be represented as its truth table, i.e., a Boolean vector of $2^n$ elements which we denote by $\Omega_f$ where the entries encodes the outputs for each possible input $x \in \mathbb{F}_2^n$ in lexicographic order. The Hamming weight of $w_H(f)$ is the number of ones in $\Omega_f$.

A first important cryptographic properties is *balancedness*, i.e., the function has the same number of ones and zeros as output, which can be formalized as requiring $w_H(f) = 2^{n-1}$.

Another important property for cryptographic applications is *nonlinearity*. This can be codified as considering the minimum Hamming distance between $\Omega(f)$ and the set of all vectors corresponding to linear functions. Higher nonlinearity is preferred for cryptographic applications [2], and the function with the maximum possible amount of nonlinearity are called *bent* functions [11].

### 2.2. Evolutionary Reaction Systems

Reaction Systems are a computational model taking inspiration by the biochemical reactions happening inside the living cell [12, 13]. Reaction systems have been studied in depth from a formal point of view (see for instance, [14, 15, 16, 17]). They consist of a set of *reactions* and each reaction is a triple $(R, I, P)$ where $R$, $I$, and $P$ are finite set of symbols called reactants, inhibitors, and products, respectively. The state $X$ of the system is a set of symbols, that is transformed by applying in parallel all reaction that are enabled (i.e., where all the reactants $R$ are in $X$ and none of the inhibitors $I$ are inside $X$) and where each of them produces the products in $P$. As such, reaction systems provide a compact way of representing function and, in particular, Boolean functions.

This ability of Reaction Systems to represent Boolean functions make possible to use them inside an evolutionary algorithm, as shown in [9, 10]. Here we provide a brief description of the main structure of the algorithm:

1 Generate $P$ reaction systems randomly. The collection of all of them will be called *population* of size $P$;

2 Evaluate the quality of each reaction system in solving the task at hand (in this case, the nonlinearity of the function represented);

3 Select the systems with the best nonlinearity (this step can be stochastic);

4 Exchange parts of the selected systems between them (*crossover*) and modify them stochastically (*mutation*);

5 Repeat the steps (2)-(5) (usually referred to as a *generation*) until a termination criteria is met.

The structure is the standard one of classical evolutionary algorithms like Genetic Algorithms (GA), but with specific mutation and crossover.

### 2.3. Results

Here we present, for the sake of compactness, only one result as taken from [8]. The tested problem is the search for bent functions on $n = 6$ inputs, thus maximizing nonlinearity. The proposed Evolutionary

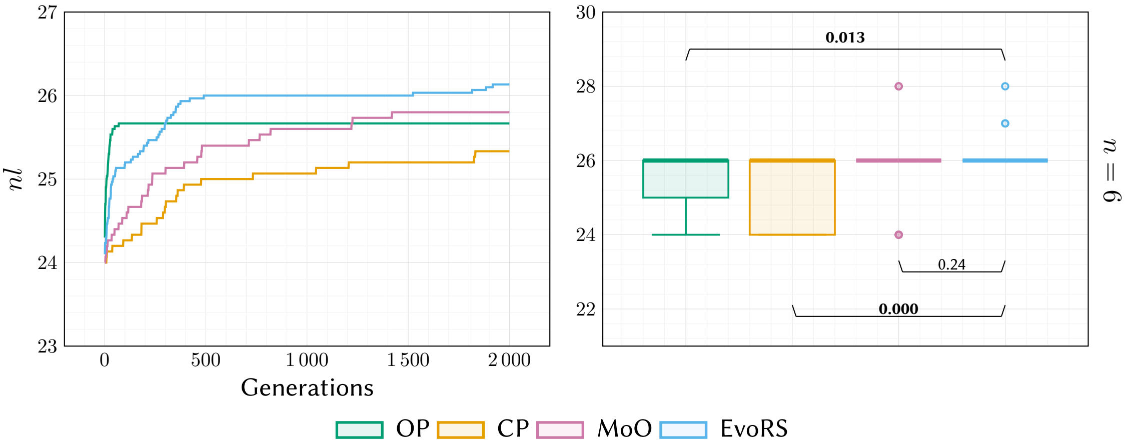

**Figure 1:** Average nonlinearity of the best individual over 30 runs (left) and final distribution of the nonlinearity (right). The p values of the Wilcoxon rank-sum test are shown in the same plot.

Reaction System (EvoRS) is compared with the variants of GA that are currently the state of the art, as proposed in [18]. For fairness in the comparison, the algorithm are compared with equal population sizes $P = 100$ and the same number of generations (2000).

The results are shown in Figure 1, where it is possible to observe that on average EvoRS are able to obtain higher nonlinearity and be statistically better than two of the three methods give the same computational budget. In particular, EvoRS are statistically better than OP and CP, but not MoO.

While the results are promising, it is important to notice that EvoRS are quite flexible providing, for example, a way to also evolve CA, as shown in the next section.

## 3. Non-linear CA and Composition of CA

Cellular automata (CA) are one of the oldest bio-inspired computational models in computer science, being introduced by Ulam [19] and Von Neumann [20] in the 50s. For reasons of space, we refer the reader to Kari's paper [21] for a formal and exhaustive introduction on the topic, while here we only provide a short overview.

A CA can be described as a set of cells in $d$-dimensional grid, each of them having a state from a fixed alphabet $\Sigma$. Each cell updates its state according to a local rule that depends only on the neighboring cells (i.e., the one at a distance below a certain radius). For example, in one dimension with a radius of $r$ and an alphabet $\Sigma = \{0, 1\}$, the local rule will be a function $f\colon \Sigma^{2r+1} \to \Sigma$. That is, each cell updates its state considering its cells, the $r$ cells to its right, and the $r$ cells to its left. All states are updates synchronously, giving rise to a global rule that can have a complex behavior even when the local rule is reasonably simple. For the purpose of this work we limit ourself to classical (synchronous) CA, even if asynchronous [22] and non-uniform CA [23] are also being explored.

For this reason, CA are models of distributed collective intelligence and they are used in many scientific fields for different purposes [24, 25, 26, 27, 28, 29]. CA has also been used for the design of several cryptographic primitives [30], like in Keccak [31].

To define the study of non-linear CA, it is essential to introduce the notion of *Linear CA (LCA)*, where the alphabet and the local rule are restricted to being the set $(\mathbb{Z}/m\mathbb{Z})^n$ and a linear combination defined by $n \times n$ matrices over $(\mathbb{Z}/m\mathbb{Z})^n$, respectively.

The main advantage of working with LCA is the easy-to-check characterizations of many set-theoretic and dynamical properties that have been provided for both LCA over $\mathbb{Z}/m\mathbb{Z}$ [32, 33, 34, 35] and more recently LCA over $(\mathbb{Z}/m\mathbb{Z})^n$ [36]. One big disadvantage is that their linearity is also what make them unsuitable for many cryptographic applications, thus prompting the study non-linear CA.

First of all, studying non-linear CA is important since the non-linearity introduces a level of complexity which is not present in LCA. This complexity may open new scenarios regarding CA classification and reveal behaviors that are different from those observed in well-studied classes. As to the applicative point of view, this complexity along with the non-linearity make nonlinear CA promising candidates for applications where such features are desirable, especially in cryptography where the potential of nonlinear CA is still largely unexploited. Therefore, the exploration of non-linear CA has been started, in particular, regarding their structural properties such as permutativity, surjectivity, and reversibility (see [21, 37, 38, 39] for a description of these properties).

Since characterizing local rules which make a general CA injective or surjective is arduous [40], the investigation started [41] on a subclass of non-linear CA over $\mathbb{Z}/m\mathbb{Z}$, namely, non-linear $j$-separated CA, and, among the first results, the main one concerns some structural properties of non-linear CA which are both $L$ and $R$ separated, i.e., CA with radius $r$ and local rule $f$ defined as:

$$f(x_{-r}, \ldots, x_r) = a_L x_L^{q_L} + \pi(x_{L+1}, \ldots, x_{R-1}) + a_R x_R^{q_R}$$

for some $0 \neq a_L, a_R \in \mathbb{Z}/m\mathbb{Z}$, some $q_L, q_R \in \mathbb{N}$, and where $\pi : (\mathbb{Z}/m\mathbb{Z})^{R-L-1} \to \mathbb{Z}/m\mathbb{Z}$ is any map.

For them, the following theorem has been proved [41]:

**Theorem 1.** *Let $\mathcal{F}$ be any $r$-radius non-linear CA over $\mathbb{Z}/m\mathbb{Z}$ with local rule $f$ (as above), where $m \geq 3$, and which is both $L$ and $R$ separated. If either $\gcd(q_L, \varphi(m)) = 1$ or $\gcd(q_R, \varphi(m)) = 1$ then $\mathcal{F}$ is surjective, while $\mathcal{F}$ is injective iff $L = R$ and $\gcd(q_L, \varphi(m)) = 1$, where $\varphi$ is the Euler's totient function.*

The first result obtained on the composition of CA is positive regarding structural properties, as shown in the following proposition [42]:

**Theorem 2.** *Let $(\Sigma^{\mathbb{Z}}, \mathcal{F})$ and $(\Sigma^{\mathbb{Z}}, \mathcal{G})$ be any two CA and let $f$ and $g$ their corresponding local rules. Let $h$ be the local rule of $\mathcal{F} \circ \mathcal{G}$. For each property $P$ among the following ones: injectivity, surjectivity, permutativity, closingness, and openness, it holds that: $\mathcal{F} \circ \mathcal{G}$ has property $P$ iff both $\mathcal{F}$ and $\mathcal{G}$ also have $P$.*

Unlike the structural CA properties, we have a negative result for the dynamical ones, as shown in the following proposition [42]:

**Proposition 1.** *Let $P$ be any dynamical property among the following ones: sensitivity, transitivity, mixing, chaos, positive expansivity, equicontinuity, and almost equicontinuity. There exist two CA such that the equivalence from Theorem 2 does not hold. In particular, both the directions of the equivalence certainly do not hold for transitivity, mixing, and chaos. Moreover, if $P$ is not sensitivity, certainly there exist two CA such that their composition has the property $P$, but none of them has $P$, while if $P \in \{$transitivity, mixing, chaos, sensitivity$\}$, there exist two CA with property $P$, but their composition does not have $P$.*

These results highlight how it can be possible to guide the search for CA rules in a way that reduces the search space: for preserved properties it is easy to generate new solutions by composition, while the search can be more difficult for non-preserved ones.

## 4. Conclusions

In this work we presented two lines of research for the construction of cryptographic primitives. In particular, we focused on an evolutionary approach and a theoretical study. In the future we plan to integrate the theoretical results as a factor inside the fitness function of evolutionary methods, thus gaining the ability to generate CA rules in an evolutionary way, possibly in a more advance way with respect to other metaheuristic approaches [43] The theoretical guarantees on CA structure can thus act as filters or biases in the evolutionary search, improving convergence and avoiding provably unfit candidates.

## Acknowledgments

This research was partially supported by the PRIN 2022 PNRR project entitled “Cellular Automata Synthesis for Cryptography Applications (CASCA)" (P2022MPFRT) funded by the European Union – Next Generation EU and by the Horizon 2020 MSCA project entitled “Application-driven Challenges for Automata Networks and Complex Systems (ACANCOS)” (MSCA-SE-101131549) funded by the European Union under the HORIZON-TMA-MSCA-SE action.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

[1] R. O’Donnell, Analysis of boolean functions, Cambridge University Press, 2014.

[2] C. Carlet, Boolean functions for cryptography and coding theory, CUP, 2021.

[3] S. Kavut, S. Maitra, M. D. Yucel, Search for boolean functions with excellent profiles in the rotation symmetric class, IEEE Transactions on Information Theory 53 (2007) 1743–1751.

[4] Z. Zhang, L. Wu, A. Wang, Z. Mu, X. Zhang, A novel bit scalable leakage model based on genetic algorithm, Security and Communication Networks 8 (2015) 3896–3905.

[5] G. T. Becker, The gap between promise and reality: On the insecurity of XOR arbiter pufs, volume 9293 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 535–555.

[6] S. Saha, R. S. Chakraborty, S. S. Nuthakki, Anshul, D. Mukhopadhyay, Improved test pattern generation for hardware trojan detection using genetic algorithm and boolean satisfiability, volume 9293 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 577–596.

[7] M. Djurasevic, D. Jakobovic, L. Mariot, S. Picek, A survey of metaheuristic algorithms for the design of cryptographic boolean functions, Cryptography and Communications 15 (2023) 1171–1197.

[8] R. A. annd Luca Mario, L. Manzoni, G. Pietropolli, Evolving cryptographic boolean functions with reaction systems, in: GECCO ’25 Companion, 2025. To appear.

[9] L. Manzoni, M. Castelli, L. Vanneschi, Evolutionary reaction systems, in: European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Springer, 2012, pp. 13–25.

[10] L. Manzoni, M. Castelli, L. Vanneschi, A new genetic programming framework based on reaction systems, Genetic Programming and Evolvable Machines 14 (2013) 457–471.

[11] M. Gadouleau, L. Mariot, S. Picek, Bent functions in the partial spread class generated by linear recurring sequences, Designs, codes and cryptography 91 (2023) 63–82.

[12] A. Ehrenfeucht, G. Rozenberg, Basic notions of reaction systems, volume 3340 of *Lecture Notes in Computer Science*, Springer, 2004, pp. 27–29.

[13] A. Ehrenfeucht, G. Rozenberg, Reaction systems, Fundamenta informaticae 75 (2007) 263–280.

[14] A. Dennunzio, E. Formenti, L. Manzoni, A. E. Porreca, Reachability in resource-bounded reaction systems, volume 9618 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 592–602.

[15] A. Dennunzio, E. Formenti, L. Manzoni, A. E. Porreca, Ancestors, descendants, and gardens of eden in reaction systems, Theor. Comput. Sci. 608 (2015) 16–26.

[16] R. Ascone, G. Bernardini, L. Manzoni, Fixed points and attractors of reactantless and inhibitorless reaction systems, Theoretical Computer Science 984 (2024) 114322.

[17] R. Brijder, A. Ehrenfeucht, M. Main, G. Rozenberg, A tour of reaction systems, International Journal of Foundations of Computer Science 22 (2011) 1499–1517.

[18] L. Manzoni, L. Mariot, E. Tuba, Balanced crossover operators in genetic algorithms, Swarm and Evolutionary Computation 54 (2020) 100646.

[19] S. Ulam, Random processes and transformations, in: Proceedings of the International Congress on Mathematics, volume 2, 1952, pp. 264–275.

[20] J. Von Neumann, Theory of self-reproducing automata, University of Illinois Press, 1966.

[21] J. Kari, Basic concepts of cellular automata, in: G. Rozenberg, T. Bäck, J. N. Kok (Eds.), Handbook of Natural Computing, Springer, 2012, pp. 3–24.

[22] A. Dennunzio, E. Formenti, L. Manzoni, Computing issues of asynchronous CA, Fundamenta Informaticae 120 (2012) 165–180.

[23] G. Cattaneo, A. Dennunzio, E. Formenti, J. Provillard, Non-uniform cellular automata, volume 5457 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 302–313.

[24] B. Chopard, A. Dupuis, A. Masselot, P. O. Luthi, Cellular automata and lattice boltzmann techniques: an approach to model and simulate complex systems, Adv. Complex Syst. 5 (2002) 103–246.

[25] S. Nandi, B. K. Kar, P. P. Chaudhuri, Theory and applications of cellular automata in cryptography, IEEE Trans. Computers 43 (1994) 1346–1357.

[26] Á. M. del Rey, J. P. Mateus, G. R. Sánchez, A secret sharing scheme based on cellular automata, Appl. Math. Comput. 170 (2005) 1356–1364.

[27] F. Farina, A. Dennunzio, A predator-prey cellular automaton with parasitic interactions and environmental effects, Fundamenta Informaticae 83 (2008) 337–353.

[28] G. Cattaneo, A. Dennunzio, F. Farina, A full cellular automaton to simulate predator-prey systems, volume 4173 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 446–451.

[29] A. Dennunzio, P. Guillon, B. Masson, Sand automata as cellular automata, Theor. Comput. Sci. 410 (2009) 3962–3974.

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

[31] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, The keccak reference, january 2011, 2011.

[32] M. Ito, N. Osato, M. Nasu, Linear cellular automata over $\mathbb{Z}_m$, Journal of Computer and System Sciences 27 (1983) 125–140.

[33] G. Manzini, L. Margara, A complete and efficiently computable topological classification of d-dimensional linear cellular automata over $\mathbb{Z}_m$, Theoretical Computer Science 221 (1999) 157–177.

[34] G. Cattaneo, A. Dennunzio, L. Margara, Solution of some conjectures about topological properties of linear cellular automata, Theoretical Computer Science 325 (2004) 249–271.

[35] A. Dennunzio, P. di Lena, E. Formenti, L. Margara, On the directional dynamics of additive cellular automata, Theor. Comput. Sci. 410 (2009) 4823–4833.

[36] A. Dennunzio, E. Formenti, D. Grinberg, L. Margara, Chaos and ergodicity are decidable for linear cellular automata over $(\mathbb{Z}/m\mathbb{Z})^n$, Information Sciences 539 (2020) 136–144.

[37] G. Cattaneo, A. Dennunzio, L. Margara, Chaotic subshifts and related languages applications to one-dimensional cellular automata, Fundam. Informaticae 52 (2002) 39–80.

[38] A. Dennunzio, From one-dimensional to two-dimensional cellular automata, Fundamenta Informaticae 115 (2012) 87–105.

[39] A. Dennunzio, P. Di Lena, E. Formenti, L. Margara, Periodic orbits and dynamical complexity in cellular automata, Fundam. Informaticae 126 (2013) 183–199.

[40] J. Kari, Linear cellular automata with multiple state variables, in: H. Reichel, S. Tison (Eds.), STACS 2000, volume 1770 of *LNCS*, Springer-Verlag, 2000, pp. 110–121.

[41] F. B. Ramdhane, A. Dennunzio, L. Margara, G. Menara, Structural properties of non-linear cellular automata: Permutivity, surjectivity and reversibility (2025). doi:10.48550/ARXIV.2504.15949.

[42] F. B. Ramdhane, G. Menara, On the composition of cellular automata, Preprint (2025).

[43] L. Mariot, M. Saletta, A. Leporati, L. Manzoni, Heuristic search of (semi-) bent functions based on cellular automata, Natural Computing 21 (2022) 377–391.