Thomas Prévost[1], Bruno Martin[2]

Université Côte d'Azur, France, I3S

[1]Ph.D. student,

[2]Ph.D., full professor

A 10-bit S-box generated by Feistel construction from cellular automata

Abstract. In this paper, we propose a new 10-bit S-box generated from a Feistel construction. The subpermutations are generated by a 5-cell cellular automaton based on a unique well-chosen local transition rule and bijective affine transformations. In particular, the cellular automaton rule is chosen based on empirical tests of its ability to generate good pseudorandom output on a ring cellular automaton. Similarly, Feistel network layout is based on empirical data regarding the quality of the output S-box.

Introduction. Symmetric cryptography is central to securing modern communication systems, enabling two parties that share a secret key to exchange information that appears indistinguishable from random data to any outside observer. Among symmetric techniques, block ciphers dominate practical applications, with algorithms such as AES and Blowfish setting the standard. A critical element in these designs is the substitution box (S-box), which introduces the nonlinearity necessary to resist a wide range of cryptanalytic attacks. Because attacks like linear, differential, and boomerang cryptanalysis typically exploit weaknesses in S-boxes, their careful design is essential.

The challenge lies in the immense design space: an n-bit S-box corresponds to a permutation of 2^n elements, leading to a factorial explosion in the number of possible candidates. For commonly used sizes, such as 8-bit S-boxes in AES or the 10-bit S-box introduced in this work, the search space is far too large for exhaustive analysis. Consequently, most existing constructions rely on algebraic structures over finite fields to produce S-boxes with desirable security properties.

In contrast, this paper explores a combinatorial approach to S-box design. We construct a 10-bit S-box using an 11-round Feistel network composed of three affine bijective layers

This work has been supported by a government grant managed by the Agence Nationale de la Recherche under the Investissement d'avenir program, reference ANR-17-EURE-004

and eight permutation layers derived from a one-dimensional uniform binary cellular automaton. A carefully selected local Boolean function drives the cellular automaton, providing pseudo-random permutation behavior within the Feistel framework.

Uniform cellular automaton. Cellular automata (CA) are discrete, parallel computational models composed of cells that update their states synchronously according to a local rule based on their own state and those of their neighbors. A well-known example is Conway's Game of Life.

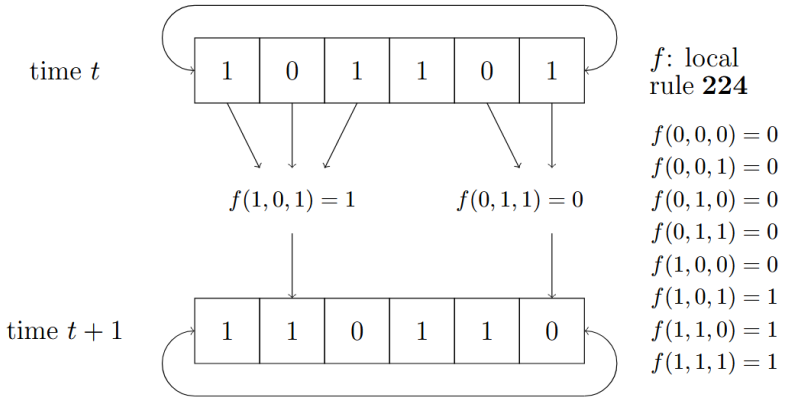

Figure 1: Example of a 1-dimensional uniform cellular automaton with the single 3-bit local rule 224. To compute the next value of a cell on the border, we consider the cell on the other edge as neighboring this one.

A 1-dimensional CA is defined by a triple *(Q, δ, N)* where *Q* is the set of states (here *{0,1}*), $\delta: Q^n \rightarrow Q$ is the local transition rule (a Boolean function of arity *n*), and $N \subseteq \mathbb{Z}$ is the neighborhood. We consider finite rings of Boolean cells, where edge cells take neighbors from the opposite side.

A CA is *uniform* if the same local rule is applied to every cell. With a suitable rule, uniform CAs can produce sequences with pseudo-random or chaotic behavior, as demonstrated for example by Wolfram's rule 30 [3]. They have been studied as potential pseudo-random bit generators, though LFSRs remain more common.

Uniform CAs can also exhibit short, non-chaotic cycles for certain inputs, such as uniform configurations, so care is required when using them in cryptographic settings.

Feistel networks.

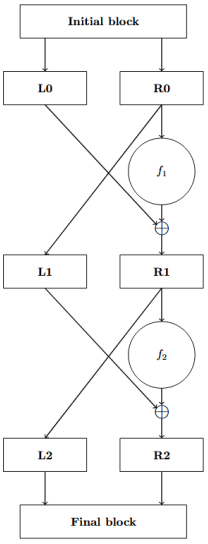

Figure 2: Example of Feistel construction of depth 2. f_1 and f_2 are pseudo-random permutations.

The Feistel network [1] provides a way to build pseudo-random bijective permutations from pseudo-random functions. When stacked in multiple rounds, the resulting network yields permutations that are computationally indistinguishable from random ones [2].

As illustrated in Fig. 2, an n-bit Feistel network splits the input into two halves and applies a sequence of round functions f_i of size $\frac{n}{2}$. The number of such functions defines the network's depth.

Our 10-bit S-Box construction. To generate our 10-bit S-box, we evaluate a Feistel permutation on all 1024 inputs, ensuring bijectivity so the S-box can be inverted. Each round function f_i is either a 5-cell uniform cellular automaton (used as a pseudo-random permutation) or an affine bijection $f_{a,b}(x)=ax+b \bmod 2^{10}$ with $gcd\,(a,2^{10})=1$.

Our Feistel network has 11 layers: an affine layer $f_{5,3}=5x+3 \bmod 1024$; four CA-based layers; an affine layer $f_{7,11}$; three CA layers; an affine layer $f_{13,17}$; one final CA layer.

To build the local CA permutation, we use a 5-cell ring updated once by a single 5-variable Boolean rule. We reduce the 2^{32} possible rules by requiring balance, first-order correlation immunity, nonlinearity, and the Strict Avalanche Criterion, leaving 7,080 candidates. These are tested as pseudo-random generators using NIST FIPS 140-2; 53 rules pass. Enforcing bijectivity leaves a single valid rule, whose ANF is

$$x_0x_3+x_1x_3+x_2x_3+x_3x_4+x_1+x_2+x_3+1.$$**Results.**

Property	Our 10-bit S-Box	AES S-Box	Comparison
Min/Max algebraic degree	8-9	7-7	Better
Algebraic complexity	1023	255	Equivalent
Nonlinearity	434 (=108.5*4)	112	Worse
Strict Avalanche Criterion	0.44-0.5-0.57	0.45-0.5-0.56	Worse
Bit Independence Criterion	0.124	0.134	Better
Lin. Approx. Probability	9.28%	6.25%	Worse
Diff. Approx. Probability	1.37%	1.56%	Better
Differential Uniformity	14	4	Worse
Boomerang Uniformity	24	6	Worse

Figure 3: Comparison of the security properties of our 10-bit S-Box and the 8-bit S-Box from AES (normalized results).

The code we used to generate our 10-bit S-Box is available at https://github.com/thomasarmel/luby_rackoff_sbox_finder .

References:

1. Feistel, H.: Cryptography and computer privacy. Scientific american (1973)
2. Luby, M., Rackoff, C.: How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal on Computing (1988).
3. Wolfram, S.: Statistical mechanics of cellular automata. Reviews of modern physics (1983).

Ключові слова: блоковий шифр на основі S-блоків, клітинні автомати, мережа Фейстеля, булеві функції.

Keywords: S-box Block cipher, Cellular automata, Feistel permutation, Boolean functions.