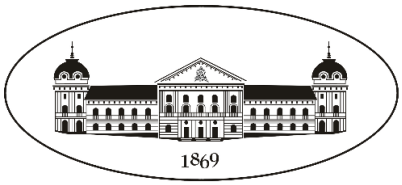
Designing Boolean Functions and Digital Sequences for Cryptology and Communications



Miroslav Marinov Dimitrov

Supervisor: Prof. Tsonka Baicheva

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Department "Mathematical Foundations of Informatics"

synopsis of dissertation
for the awarding of educational and scientific degree philosophiae doctor
in professional area 4.6
informatics and computer science

Sofia 2023

Acknowledgements

I would like to express my deepest appreciation to my supervisor prof. D.Sc. Tsonka Baicheva for the invaluable patience, feedback, and constructive criticism. I cannot begin to express my thanks for her editing help, late-night feedback sessions, and moral support. I would also like to extend my deepest gratitude to Ph.D. Nikolay Nikolov for the countless interesting conversations regarding the challenges addressed in this thesis and for bringing my attention to the PSL problem in the first place. I'm deeply indebted to prof. Bernhard Esslinger from the university of Siegen for the editing help, support, and trust he invested in me. I'm extremely grateful to the hardware parts he gratuitously sent me to build a mini GPU grid, which successfully solved some of the problems regarding the rotated binary sequences. I am also grateful to Ph.D. Georgi Ivanov who motivated me to start my Ph.D. journey through his recommendations to prof. D.Sc. Tsonka Baicheva. Special thanks to Ph.D. Violeta Andreeva for her unwavering support. Last but not least, I am thankful to my family, especially my parents, who sparked the love of mathematics in me, to my wife Geri, and our wonderful princess Mariah, for the inspiration and emotional support.

Preface

0.1 Scientific contributions

The main scientific contributions could be summarized as follows:

1. A rich collection of popular S-boxes is analyzed in great detail.

2. It is shown that the majority of chaos-based S-boxes are vulnerable to linear cryptanalysis. A simple and lightweight algorithm is proposed, which significantly outperforms all previously published chaos-based S-boxes, in those cryptographic terms, which they utilize for comparison.

3. By introducing some new definitions like couplings, coordinate decomposition, degree of descendibility, and CELAT, the S-box nonlinearity optimization problem is projected to a satisfiability problem, which could be attacked by using SAT solvers.

4. By applying the SAT solver it is shown that 8×8 bijective S-boxes with all eight coordinates having the maximal nonlinearity value of 116 do exist.

5. A strategy of analyzing various spectra channels to detect hidden patterns and anomalies in S-boxes is proposed.

6. A simple and efficient algorithm based on a heuristic search by shotgun hill climbing to construct binary sequences with small peak sidelobe levels (PSL) is proposed. The algorithm successfully revealed binary sequences of lengths between 106 and 300 with record-breaking PSL values.

7. By using some useful properties, the aforementioned algorithm time and memory complexities are reduced to $\mathcal{O}(n)$. This allowed us to reach record-breaking PSL values for less than a second. Moreover, the efficiency range of the algorithm is further extended to binary sequences of longer lengths.

8. A detailed comparison and fine-grain analysis of the proposed algorithms is performed. By using the insights of this analysis, a heuristic algorithm is proposed, which successfully reached all the optimal PSL values known in the literature, which was previously discovered by an exhaustive search. This was achieved by using a low-cost mid-range computer station, while the time required to reach the optimal PSL value for most of the lengths is less than a second.

9. A GPU efficient algorithm addressing the well-known computational problem of finding the lowest possible PSL among the set of a binary sequence B and all binary sequences generated by rotations of B is proposed. The problem is projected to a perfectly balanced parallelizable algorithm. By using the algorithm, the search space of all m-sequences with lengths $2^n - 1$, for $18 \leq n \leq 20$ is successfully exhausted. Furthermore, a complete list of all PSL-optimal Legendre sequences for lengths up to 432100 is revealed. A conjecture is made, that all PSL-optimal Legendre sequences, with or without rotations, and with lengths N greater than 235723, are strictly greater than $\sqrt{N}$.

10. Some useful mathematical properties related to the flip operation of the skew-symmetric binary sequences are discovered, which could be exploited to significantly reduce the memory complexity of state-of-the-art stochastic Merit Factor (MF) optimization algorithms from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, without degrading their time complexity. As a proof of concept, a lightweight algorithm was constructed, which could optimize pseudo-randomly generated skew-symmetric binary sequences with long lengths (up to $10^5 + 1$) to skew-symmetric binary sequences with a MF greater than 5. This contradicts the Bernasconi conjecture, that a stochastic search procedure will not yield MF higher than 5 for long binary sequences (sequences with lengths greater than 200).

11. A new class of finite binary sequences with even lengths with alternate autocorrelation absolute values equal to 1, called pseudo skew-symmetric class, is found. It is shown that the MF values of the new class are closely related to the MF values of adjacent classes of Golay's skew-symmetric sequences.

12. Sub-classes of sequences based on the partition number problem, as well as the notion of potentials, measured by helper ternary sequences, are proposed. Binary sequences with MF records for binary sequences with many lengths less than 225, and all lengths greater than 225, are revealed. Two extremely hard search spaces of lengths 573 and 1009 are also attacked. It was estimated that a state-of-the-art stochastic solver requires

respectively 32 and 46774481153 years to reach MF values of 6.34, while the required time from the proposed algorithm to reach such MF values is just several hours.

13. Using aperiodic autocorrelation functions for the S-box reverse engineering problem is proposed.

0.2 Publications related with the thesis

1. Dimitrov, Miroslav M. "On the design of chaos-based S-boxes." IEEE Access 8 (2020): 117173-117181, **IF:3.367, Q2**.

2. Dimitrov, Miroslav, Tsonka Baitcheva, and Nikolay Nikolov. "Efficient generation of low autocorrelation binary sequences." IEEE Signal Processing Letters 27 (2020): 341-345, **IF:3.109, Q2**.

3. Dimitrov, Miroslav, Tsonka Baitcheva, and Nikolay Nikolov. "On the generation of long binary sequences with record-breaking PSL values." IEEE Signal Processing Letters 27 (2020): 1904-1908, **IF:3.109, Q2**.

4. Dimitrov, Miroslav. "On the aperiodic autocorrelations of rotated binary sequences." IEEE Communications Letters 25.5 (2020): 1427-1430, **IF:3.436, Q2**.

5. Dimitrov, Miroslav, Tsonka Baicheva, and Nikolay Nikolov. "Hybrid Constructions of Binary Sequences With Low Autocorrelation Sideobes." IEEE Access 9 (2021): 112400-112410, **IF:3.476, Q2**.

6. Dimitrov, Miroslav M. "A Framework for Fine-Grained Nonlinearity Optimization of Boolean and Vectorial Boolean Functions." IEEE Access 9 (2021): 124910-124920, **IF:3.476, Q2**.

7. Iliev, M., Nikolov, N., Dimitrov, M. and Bedzhev, B. "Genetic algorithm for synthesis of binary signals with optimal autocorrelation." 2020 International Conference on Information Technologies (InfoTech). IEEE, 2020.

8. Dimitrov, Miroslav. "On the Skew-Symmetric Binary Sequences and the Merit Factor Problem." arXiv preprint arXiv:2106.03377 (2021).

9. Dimitrov, Miroslav. "New Classes of Binary Sequences with High Merit Factor." arXiv preprint arXiv:2206.12070 (2022).

Chapter 1

Introduction

Boolean functions, vector Boolean functions, or S-boxes, and digital sequences are widely used in various practical fields such as telecommunications, radar technology, navigation, cryptography, measurement sciences, biology, or industry.

S-boxes are one of the most important primitives to be found in modern block ciphers. A weak S-box, in a cryptographic perspective, can be exploited by various attacks like linear cryptanalysis [17], differential cryptanalysis [18] , boomerang attack [147], algebraic attacks [34] or others like in [59]. Arguably, one of the most important properties of a given S-box is its nonlinearity. An S-box with high nonlinearity can be achieved by using the finite field inversion method [113]. However, such S-box is closely related to various algebraic structures. As a proactive countermeasure to future algebraic attacks, new ways of generation or optimization of pseudo-random S-boxes are proposed. Some examples of the aforementioned algorithms are published in [32], [85], [107], [108], and [145]. However, heuristically optimization of a given S-boxes could be a resource-consuming task.

Given their significance and importance, the design principles of an S-box construction, especially when implemented in a widely used and critical cryptosystem, should be publicly available and reproducible. However, in some cases, a given S-box generation method is not announced, or worse, misleadingly announced as a pseudo-randomly generated one. The reasons for obfuscating the design of a given S-box are manifold. For example, the initial S-boxes used in the Data Encryption Standard (DES) [55] were originally modified by NSA. The reasons for applying those modifications were not known. However, in [33], D. Coppersmith announces the motivation behind the S-box modifications. It appears that the agency knew about the existence of differential attacks about 20 years before the academic world.

Hiding a given S-box design could be related to some hidden construction, the knowledge of which could be exploited to gain a significant advantage in terms of hardware implementa-

tion. For example, as discovered in [21], the S-boxes used in the hash function Streebog and the 128-bit block cipher Kuznyechik, standardized by the Russian Federation, are designed with such a hidden structure. A user knowing this decomposition could implement the given S-box with a significantly smaller hardware footprint, allowing him to reach an up to 8 times faster S-box look-up.

A practical reason for hiding the design of a given S-box could be related to an encapsulated trapdoor as discussed in [128]. Even though the aforementioned trapdoor can be easily detected, as shown in [151], the motivation for finding other trapdoor S-box techniques should not be underestimated. Moreover, the designers of a given S-box could unintentionally create it with a flaw, which further rises the academic attention to the S-box reverse engineering problem.

Finding binary sequences whose aperiodic autocorrelation characteristics are collectively small according to some pre-defined criteria is a famous and well-studied problem. Two such measures are the Peak Sidelobe Level (PSL) and the Merit Factor (MF) value, which was first introduced by Golay in 1972 [60]. However, before Golay's definition, Littlewood [98] studied the norms of polynomials with ± 1 coefficients on the unit circle of the complex plane.

One of the desirable characteristics a given binary sequence should possess is a low peak sidelobe level. Some well-known constructions of such sequences includes the Barker codes [9], Rudin-Shapiro sequences [129][136], m-sequences [67], Gold codes [66], Kasami codes [84], Weil sequences [130], Legendre sequences [124]. Nevertheless, none of the aforementioned constructions guarantees that the generated binary sequence will possess the lowest possible (optimal) PSL value. Thus, currently, initiating an exhaustive search is the only way to reveal an optimal PSL value for binary sequences of some fixed length. The PSL-optimal values of binary sequences with lengths n greater than 84 are still unknown. This is not surprising, since the cardinality of the search space comprised of all binary sequences with some fixed length rises exponentially.

Golay's publications reveal a dedication to the merit factor problem for nearly twenty years (surveyed in [80]). Since then, a significant number of possible constructions of binary sequences with high merit factors were published. Near-optimal and optimal candidates are found by using heuristic search methods for longer lengths or a more direct approach, like the exhaustive search method, for smaller problem spaces. In [65], the merit factor problem was referenced by Golay as ***...challenging and charming***.

The problem of minimizing the merit factor is also known as the "low autocorrelated binary string problem", or the LABS problem. It has been well studied in theoretical physics and chemistry. For example, the LABS problem is correlated with the quantum models of

magnetism. Bernasconi predicted that [14] ***... stochastic search procedures will not yield merit factors higher than about 5 for long sequences***. By long sequences, Bernasconi was referring to binary sequences with lengths greater than 200. Furthermore, in [41] the problem was described as ***... amongst the most difficult optimization problems***. Since the merit factor problem has resisted more than 50 years of theoretical attacks, a significant number of computational pieces of evidence were collected.

In this thesis, several design strategies for constructing and analyzing Boolean functions, S-boxes, and digital sequences are proposed. In Chapter 2 the preliminaries are provided. In Sections 2.1 and 2.2 some important definitions regarding Boolean and vector Boolean functions are given. Then, in Section 2.3 a rich collection of popular S-boxes is thoroughly analyzed. In general, the S-box construction methods could be divided into four categories as shown in Section 2.4. Then, S-boxes generated by using chaotic functions (CF) are analyzed to measure their actual resistance to linear cryptanalysis. The majority of the published papers using CFs emphasize the average nonlinearity of the S-box coordinates only, ignoring the rest of the S-box components in the process. Thus, integrating such S-boxes in a given cryptosystem should be done with considerable caution. Furthermore, it appears that in the context of the nonlinearity optimization problem the profit of using chaos structures is negligible. During our experiments, by using two heuristic methods and starting from pseudo-random S-boxes, we repeatedly reached S-boxes, which significantly outperform all previously published CF-based S-boxes, in those cryptographic terms, which the aforementioned papers utilize for comparison. Then, in Section 2.5, we project the S-box nonlinearity optimization problem to a satisfiability problem, which could be solved by using SAT solvers. This is achieved by introducing some new definitions like couplings, coordinate decomposition, degree of descendibility, S-box coordinate extended linear approximation table (CELAT), as well as some useful properties and inner connections. The SAT projection revealed that we could successfully construct bijective 8×8 S-boxes from 8 Boolean functions as components, each of which possesses the maximum nonlinearity value of 116. The provided toolbox could serve in cases, where the designer's goal is to increase (or intentionally decrease) the nonlinearity of a given S-box by applying as minimum changes as possible. For example, we demonstrate how the Skipjack S-box, developed by the U.S. National Security Agency (NSA), and the Kuznyechik S-box, developed by the Russian Federation's standardization agency, could be optimized to a higher nonlinearity by tweaking just 4 and 12 bits, respectively (out of 2048).

In Chapter 3, a strategy of analyzing various spectra channels to detect hidden patterns and anomalies in popular S-boxes is discussed. It could serve as a more fine-grained extension to the methods discussed in [119]. More specifically, by applying spectral analysis on various

S-box characteristics, as a linear approximation, difference distribution, and auto-correlation tables, we can detect visual symmetries or anomalies, which could not only serve as proof that the S-box was not generated pseudo-randomly but additionally provides some further information about the inner structure of the S-box, making the complete reverse-engineering of the hidden construction possible [1].

Chapter 4 addresses the PSL optimization problem. In Section 4.1, a simple and efficient algorithm based on a heuristic search by shotgun hill climbing to construct binary sequences with small peak sidelobe levels is suggested. The algorithm is applied for the generation of binary sequences of lengths between 106 and 300. Improvements are obtained in almost half of the considered lengths while for the rest of the lengths, binary sequences with the same PSL values as reported in the state-of-the-art publications are found. Then, in Section 4.2, a method to generate long binary sequences with low PSL value is proposed. Both the time and memory complexities of the proposed algorithm are reduced to $\mathcal{O}(n)$. During our experiments, we repeatedly reach better PSL values than the currently known state of art constructions, such as Legendre sequences, with or without rotations, Rudin-Shapiro sequences or m-sequences, with or without rotations, by always reaching record-breaking PSL values strictly less than $\sqrt{n}$. Furthermore, the efficiency and simplicity of the proposed method are particularly beneficial to the lightweightness of the implementation, which allowed us to reach record-breaking PSL values for less than a second. In Section 4.3 we continue our research with the exploration of hybrid algorithms for achieving binary sequences with arbitrary lengths and high PSL values. During our experiments, and by using the aforementioned algorithms, we were able to find PSL-optimal binary sequences for all those lengths, which were previously found during exhaustive searches by various papers. Then, by using a general-purpose computer, we further demonstrate the effectiveness of the proposed algorithms by revealing binary sequences with lengths between 106 and 300, the majority of which possess record-breaking PSL values. Then, by using some well-known algebraic constructions, we outline a few strategies for finding highly-competitive binary sequences, which could be efficiently optimized, in terms of PSL, by the proposed algorithms. Finally, in Section 4.3.3, a well-known computational problem is finding the lowest possible PSL among the set of a binary sequence B, and all binary sequences generated by rotations of B is discussed. Some useful properties of rotated binary sequences are discovered, which allowed us to project the aforementioned problem to a perfectly balanced parallelizable algorithm. The proposed algorithm, altogether with its graphics processing unit (GPU) implementation, is significantly faster than the existing instruments. We were able to exhaust

[1] Although the demonstrated anomalies are visible on paper, reading the electronic version is greatly encouraged.

the search space of all m-sequences with lengths $2^n - 1$, for $18 \leq n \leq 20$, and to reveal a complete list of all PSL-optimal Legendre sequences, with or without rotations, for lengths up to 432100 - out of computational reach until now. The numerical experiments suggest that the PSL value of all PSL-optimal Legendre sequences, with or without rotations, and with lengths N greater than 235723, are strictly greater than $\sqrt{N}$.

Chapter 5 deals with the Merit Factor (MF) problem. It was conjectured that stochastic search procedures will not yield merit factors higher than 5 for long binary sequences (sequences with lengths greater than 200). Some useful mathematical properties related to the flip operation of the skew-symmetric binary sequences are presented in Section 5.1. By exploiting those properties, the memory complexity of state-of-the-art stochastic MF optimization algorithms could be reduced from $O(n^2)$ to $O(n)$. As a proof of concept, a lightweight stochastic algorithm was constructed, which can optimize pseudo-randomly generated skew-symmetric binary sequences with long lengths (up to $10^5 + 1$) to skew-symmetric binary sequences with a merit factor greater than 5. An approximation of the required time is also provided. Golay introduced one beneficial class of sequences, called skew-symmetric sequences; finite binary sequences with odd lengths with alternate autocorrelation values equal to 0. Their special construction greatly reduces the computational efforts of finding binary sequences with odd lengths and high MF. Having this in mind, the majority of papers to be found in the literature are focused solely on this class, preferring them over binary sequences with even lengths. In Section 5.1.2, a new class of finite binary sequences with even lengths with alternate autocorrelation values equal to ± 1 is presented (see also [46]). We show that the MF values of the new class are closely related to the MF values of adjacent classes of skew-symmetric sequences. We further introduce new sub-classes of sequences using the partition number problem and the notion of potentials, measured by helper ternary sequences. Throughout our experiments, MF records for binary sequences with many lengths less than 225, and all lengths greater than 225, are discovered. Binary sequences of all lengths, odd or even, less than 2^8 and with MF > 8, and all lengths, odd or even, less than 2^9 and with MF > 7, are now revealed. We demonstrate the efficiency of the proposed algorithm by launching it on two extremely hard search spaces of binary sequences of lengths 573 and 1009. It was estimated that finding solutions with a merit factor of 6.34 for a binary sequence with length 573 requires around 32 years, while for binary sequences with length 1009, the average runtime prediction to reach the merit factor of 6.34 was 46774481153 years (see [24]). By using the proposed in Section 5.1.2 algorithm, we were able to reach such binary sequences within several hours. Finally, in Section 5.2, a method addressing the S-box reverse engineering problem using spectrography on aperiodic autocorrelation functions is presented.

Chapter 2

Vector Boolean Functions and Cryptography

2.1 Boolean Functions

Definition 2.1.1 (Boolean Function & Truth Tables)**.** Let us define the set $B = \{0,1\}$. A Boolean function $f(x)$ of n variables $x_1,...,x_n$ is a mapping $f : B^n \mapsto B$ from n binary inputs $x = (x_1,x_2,\cdots,x_n) \in B^n$ to one binary output $y = f(x) \in B$. The binary truth table (BTT) of an n-variable Boolean function $f(x)$ is the vector of all the consecutive outputs of the Boolean function:

$$[f(x)] = [f(0,0,\cdots,0), f(0,0,\cdots,1), \cdots, f(1,1,\cdots,1)]$$

The polarity truth table (PTT) of an n-variable Boolean function $f(x)$ is derived from the binary truth table. We define the PTT by $[\hat{f}(x)] = [1-2f(x)]$.

Definition 2.1.2 (Algebraic Normal Form)**.** The algebraic normal form of an n-variable Boolean function $f(x)$, denoted by ANF_f, is given by the following equation: $ANF_f = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n \oplus a_{1,2}x_1x_2 \oplus \cdots \oplus a_{1,2,\cdots,n}x_1x_2\cdots x_n$, where the coefficients a belongs to B.

Definition 2.1.3 (Algebraic Degree)**.** The algebraic degree of a Boolean function $f(x)$, denoted by $deg(f)$, is equal to the number of variables in the longest item of its ANF_f.

Definition 2.1.4 (Hamming Distance)**.** The Hamming distance between two n-variable Boolean functions $f(x)$ and $g(x)$, denoted by $d_H(f,g)$, represents the number of differing elements in the corresponding positions of their truth tables.

Definition 2.1.5 (Linear Boolean Function)**.** Any n-variable Boolean function of the form:

$$l_w(x) =< w,x >= w_1x_1 \oplus w_2x_2 \oplus \cdots \oplus w_nx_n,$$

where $w,x \in B^n$, is called a linear function.

Definition 2.1.6 (Affine Boolean Function)**.** Any n-variable Boolean function of the form:

$$l_w(x) =< w,x >= w_0 \oplus w_1x_1 \oplus w_2x_2 \oplus \cdots \oplus w_nx_n,$$

where $w_0 \in B$ and $w,x \in B^n$, is called an affine function.

Definition 2.1.7 (Walsh-Hadamard Transform)**.** For an n-variable Boolean function $f(x)$, represented by its polarity table $[\hat{f}(x)]$, the Walsh-Hadamard transform, or WHT, $\hat{F}_f : B^n \to Z$, is defined by:

$$\hat{F}_f(w) = \sum_{x \in B^n} \hat{f}(x)(-1)^{<w,x>}$$

Definition 2.1.8 (Absolute Indicator)**.** For an n-variable Boolean function $f(x)$, we denote the absolute indicator of f as Δ_f. For all $u \in F_2^n$, except the zero vector, write

$$\Delta_f(u) = \sum_x (-1)^{f(x)+f(x+u)}$$

The absolute indicator of f is calculated by

$$\Delta_f = \max_u \mid \Delta_f(u) \mid \tag{2.1}$$

2.2 Vector Boolean Functions (S-boxes)

Definition 2.2.1 (Vectorial Boolean Function – Substitution Table – S-box)**.** An n-binary input to m-binary output mapping $S : B^n \Longleftrightarrow B^m$, which assigns some $y = (y_1, y_2, \cdots, y_m) \in B^m$ by $S(x) = y$ to each $x = (x_1, x_2, \cdots, x_n) \in B^n$, is called an (n,m) substitution table (S-box) and is denoted by $S(n,m)$.

Definition 2.2.2 (Bijective S-box)**.** An S-box $S(n,m)$ is said to be bijective, if it maps each input $x \in B^n$ to a distinct output $y = S(x) \in B^m$ and all possible 2^m outputs are present.

Definition 2.2.3 (S-box Look-up Table – LUT)**.** The look-up table LUT of an S-box $S(n,m)$ is an (2^n x m) binary matrix S, which rows consist of all outputs of $S(n,m)$, corresponding to all possible 2^n inputs ordered lexicographically.

Definition 2.2.4 (S-box Coordinates)**.** We define each column of the $S(n,m)$ LUT as a coordinate of $S(n,m)$. Each column represents the truth table of some Boolean function f_i. If $S(n,m)$ is bijective vectorial Boolean function it follows that $n = m$ and we have exactly n coordinates.

Definition 2.2.5 (Polarity Look-up Table – PLUT)**.** The polarity look-up table PLUT of an S-box $S(n,m)$, denoted by S_{PLUT}, is an $(2^n,m)$ matrix with elements in $\{-1,1\}$, where each element on row j and column k, denoted by $S_{PLUT}[j][k]$, for $j = 1,2,\cdots,2^n$ and $k = 1,2,\cdots,m$, is derived from $S_{LUT}[j][k]$ by

$$S_{PLUT}[j][k] = (-1)^{S_{LUT}[j][k]} = 1 - 2S_{LUT}[j][k]$$

where $\hat{f}_i(\alpha) = (-1)^{f_i(\alpha)} = 1 - 2f_i(\alpha)$.

Definition 2.2.6 (S-box Extended WHT Spectrum Matrix – EWHTSM)**.** The extended Walsh-Hadamard transform spectrum matrix (EWHTSM) of an S-box $S(n,m)$ is a $(2^n,2^m)$ matrix $\hat{F}_{ExtS}$, which columns are represented by the Walsh-Hadamard transform spectra $[\hat{F}_{g_v}(w)]$ of the Boolean functions $g_v(x) = v_1 f_1(x) \oplus v_2 f_2(x) \oplus \cdots \oplus v_m f_m(x)$, where w and v are arranged lexicographically respectively in B^n and B^m.

$$\hat{F}_{ExtS} = \begin{bmatrix} \hat{F}_{g_0}(0,0,\cdots,0) & \cdots & \hat{F}_{g_{2^m-1}}(0,0,\cdots,0) \\ \hat{F}_{g_0}(0,0,\cdots,1) & \cdots & \hat{F}_{g_{2^m-1}}(0,0,\cdots,1) \\ \vdots & \ddots & \vdots \\ \hat{F}_{g_0}(1,1,\cdots,0) & \cdots & \hat{F}_{g_{2^m-1}}(1,1,\cdots,0) \\ \hat{F}_{g_0}(1,1,\cdots,1) & \cdots & \hat{F}_{g_{2^m-1}}(1,1,\cdots,1) \end{bmatrix} \tag{2.2}$$

The importance of the S-box extended Walsh-Hadamard transform matrix is to quantitatively describe the distance with a special measure, alike the Hamming distance, between each linear combination of coordinates in the given S-box and each possible linear function.

Definition 2.2.7 (Linear Approximation Table – LAT)**.** The linear approximation table of an S-box $S(n,m)$, denoted by LAT_S or S_{LAT}, is a table with 2^n rows and 2^m columns, which entries are given by:

$$S_{LAT}[X][Y] = LAT_S[X][Y] = 2^{n-1} - d_H(X,Y), \tag{2.3}$$

where Y is a consequent linear combination of coordinates of the current S-box and X is the consequent linear function with length n.

Definition 2.2.8 (S-box Nonlinearity)**.** The nonlinearity of an S-box $S(n,m)$, denoted by S_{NL}, is defined as:

$$S_{NL} = 2^{n-1} - \max(\{\mid w_i \mid\}), \tag{2.4}$$

where $\{\mid w_i \mid\}$ is the set of all absolute values of elements in LAT, except the uppermost left one.

Definition 2.2.9 (S-box ACNV)**.** The average coordinate nonlinearity value, or S_{ACNV}, of a given S-box S, is the average value of all nonlinearities of coordinates of S.

Definition 2.2.10 (S-box Decimal Look-up Table – DLUT)**.** Each S-box is uniquely defined by its LUT. Translating each row of the LUT as a decimal number uniquely defines the same S-box as a decimal look-up table (DLUT).

Definition 2.2.11 (XOR Table)**.** The XOR table of an S-box $S(n,m)$ is a $(2^n \times 2^m)$ binary matrix S_{XORT}, which columns consist of all linear combinations of S_{LUT} columns ordered lexicographically.

Definition 2.2.12 (S-box Minimal Algebraic Degree)**.** The minimal algebraic degree of an S-box $S(n,m)$ is the minimum algebraic degree among all component functions of S.

$$\begin{aligned} S_{DEG} = min_{(v \in B^m)} deg(g_v) = \\ = min_{((v_1,v_2,\cdots,v_m) \in B^m)} deg(v_1 f_1(x) \oplus v_2 f_2(x) \oplus \cdots \oplus v_m f_m(x)), \end{aligned} \tag{2.5}$$

where $f_1, f_2, \cdots, f_m$ are the coordinate Boolean functions of $S(n,m)$.

Definition 2.2.13 (S-box Absolute Indicator)**.** The absolute indicator of a given S-box S, denoted as S_{AC}, is equal to the maximal absolute indicator among all absolute indicators of component functions of S.

Definition 2.2.14 (S-box Differential Uniformity)**.** Differential uniformity, or δ-uniformity of a given S-box $S(n,m)$, denoted by S_δ, is defined by:

$$S_\delta = \max_{\alpha \in B^n \setminus \{0\}} \max_{\beta \in B^m} \left| \{ x \in B^n \mid S(x) \oplus S(x \oplus \alpha) = \beta) \} \right|$$

2.3 Cryptographic Properties of Some Popular S-boxes

The cryptographic properties of vector boolean functions are thoroughly examined by introducing a rich list of desirable parameters an S-box should have to guarantee an acceptable

resistance to sophisticated cryptographic attacks such as the linear cryptanalysis [103][17], the differential cryptanalysis [18], boomerang attack [147] or interpolation attack [79]. S-boxes are widely used in modern cryptographic algorithms like AES [40], Whirlpool [11], Camellia [7] and many others. For a given S-box S the goal of the designer is to achieve high values of S_{NL} and S_{DEG}, as well as small values of S_δ and S_{AC}.

The S-boxes, created with the Finite Field Inversion method [114], as the Rijndael S-box used in AES [40], have the best currently known cryptographic properties among all 8×8 S-boxes. However, some concerns about constructing S-boxes by using a purely algebraic approach can make them vulnerable to algebraic attacks [34]. Hence, in some applications, randomly or heuristically generated S-boxes are used. Throughout the dissertation, a collection of well-known and published S-boxes used in popular cryptographic algorithms are analyzed, and one can see that only 11 S-boxes, out of 47, are AES-alike. For a more detailed picture, the LAT Spectras of the S-boxes is also provided, i.e. the real-valued vector of all absolute values of LAT coefficients. The distribution of the S_{LAT} coefficients of a given S-box S could also provide some more insights into how S is constructed when the construction method is not announced (intentionally or not) by the designers of S.

2.4 Design Strategies for Constructing S-boxes

The rich variety of proposed S-boxes constructions can be classified into four categories. The first category $\boldsymbol{T_1}$ for finding S-boxes with good cryptographic properties uses the pseudo-random generation method. The highest reported nonlinearity (NL) of an $(8,8)$ S-box generated by this approach is 100 [110]. We generated over one billion S-boxes $(1,387,914,282)$ and, for example, find that the probability to randomly construct an $(8,8)$ S-box with NL 100 is $2^{-25.978}$. Thus, the probability to find an S-box of NL 100, or higher, at random is rather small.

The second category $\boldsymbol{T_2}$ uses a more straightforward (deterministic) approach, like an algebraic constructions like finite field inversion method, cellular automata based methods [16], quasi-cyclic codes methods [25][19], affine-power-affine methods [38] or using some other deterministic approach as Feistel and Misty constructions [29].

The third category $\boldsymbol{T_3}$ is about applying heuristic search methods to optimize pseudo-randomly generated S-boxes. Members of this category are methods like hill climbing [107], simulated annealing [32], genetic algorithms [108], special genetic algorithms combined with total tree searching [145], special immune algorithms [78], and others [142][121].

The fourth category $\boldsymbol{T_4}$ is using hybrid search, i.e starting from an S-box generated by some T_2 construction, and then obtaining a new one by using some T_3 algorithm. Such

methods are suggested in [85][31][76][101][42][77][4]. It should be noted that categories T_3 and T_4 looks similar. However, the comparison between T_3 and T_4 methods is not entirely fair, since the authors of the latest do not start from a pseudo-random state. Instead, they initialize their algorithm with some highly competitive candidate. The same observation is made in [121], p.9.

We should also address the S-box chaos-based constructions methods. They could belong to either of categories T_2, T_3 or T_4. However, in [50], S-boxes generated by using chaotic functions (CF) are analyzed to measure their actual resistance to linear cryptanalysis. It appears that most of the aforementioned papers emphasize the average nonlinearity of the S-box coordinates (ACNV) only, ignoring the rest of the S-box components in the process. Having this in mind, the majority of those studies should be re-evaluated. Integrating such S-boxes in a given cryptosystem should be done with considerable caution. Furthermore, we show that in the context of the nonlinearity optimization problem the profit of using chaos structures appears to be negligible. By using two heuristic methods and starting from pseudo-random S-boxes, we repeatedly reached S-boxes, that significantly outperform all previously published CF-based S-boxes, in those cryptographic terms, that the aforementioned papers utilize for comparison. Moreover, we have linked the multi-armed bandit problem to the problem of maximizing an S-box average coordinate nonlinearity value, which further allowed us to reach near-optimal average coordinate nonlinearity values significantly greater than those known in the literature.

The methods involved in CF S-box constructions are manifold (see the comparison table provided in [50]). As defined in Definition 2.2.8, we seek the maximum absolute value v of all the elements in S-box $S(n,n)$ LAT, to find the nonlinearity of S, i.e. $S_{NL} = 2^{n-1} - v$. In the context of block ciphers, a low nonlinearity S-box value is associated with the cipher linear cryptanalysis resistance [103][17][74]. As shown in [50], the average value of the nonlinearities of the coordinates of a given S-box S doesn't correspond to the actual nonlinearity of S. However, from the designer's perspective, when a higher value of ACNV is desirable, a simple heuristic construction could be used instead.

In general, if we want to improve the nonlinearity of a given bijective S-box $S(n,n)$, a strategy of lowering the absolute value of coefficients in S_{LAT} makes sense. Moreover, the elements of each column of S_{LAT} are entangled by Parceval's theorem [104]. Let's denote as C_i the array composed of the elements of $S_{LAT}[i]$. Since we want to lower the nonlinearities of coordinates of S only, an evaluating function $E(S)$ is created, s.t. $E(S) = \sum_{p=0}^{n-1} \sum_{x \in C_{2^p}} |x|^M$, where M denotes a magnitude of our choice. The restriction $x \in C_{2^p}$ narrows down the set of possible columns of S_{LAT} to be optimized, in terms of nonlinearity, to the set of coordinates of S.

By using stochastic[1] hill climbing as a heuristic function, starting from arbitrary pseudo-random S-box construction and by using $E(S)$, we could repeatedly optimize pseudo-randomly generated S-boxes to ACNV of 114.0, the highest reported in the literature. Moreover, by exploiting the techniques discussed in the multi-armed bandit problem [15], we were able to reach ACNV of 114.5 (see [50]). The algorithm was implemented with the built-in tools provided by the open-source mathematical software system SageMath [43].

2.5 Nonlinearity Optimization Using SAT Solvers

In this section, an interconnection between the S-box nonlinearity optimization problem and binary integer programming is shown. A lightweight optimization routine is proposed, which does not cause any significant computational burden. Moreover, the toolbox could be utilized as proof of infeasibility.

A major drawback of the state-of-the-art heuristic techniques is their aggressiveness on the initial S-box. Hence, in most cases, it is difficult to link the resulting S-box with the initial S-box. It is difficult to prove that such a link exists in the first place. The fine-grained optimization routine proposed in [51] allows us to optimize the nonlinearity value of a given S-box with as minimum changes as possible. From the designer's perspective, this property is particularly beneficial, since we could focus the optimization routine on the weak components of a given S-box, without degrading the remaining ones. The effectiveness of the proposed algorithm is further demonstrated by increasing the nonlinearity of the Skipjack S-box, developed by NSA, and Kuznyechik S-box, developed by the Russian Federation's standardization agency, by tweaking respectively 4 and 12 (out of 2048) bits only.

The currently known maximum nonlinearity value for 8-variable balanced Boolean functions is 116 [122]. Furthermore, as shown in [133], the nonlinearity value of 8-variable balanced Boolean functions is upper bounded by 120, which means that the maximum theoretical ACNV of (8,8) bijective S-boxes is less or equal to 118.0. If a bijective S-box with ACNV greater than 116.0 is found, at least one of its eight coordinates will possess a nonlinearity value of 118, which will finally answer the long-standing problem of the maximum possible nonlinearity value for 8-variable balanced Boolean functions. However, there is academic skepticism that 8-variable balanced Boolean functions with nonlinearity value 118 exist. Having this in mind, one open question to be answered is: *Does bijective (8,8) S-box with an ACNV value of 116 exist*? By using the SAT solving techniques, we showed that bijective (8,8) S-boxes with an ACNV value of 116.0 exist. However, despite our

[1] hill climbing without neighborhood search

attempts, we were not able to find an 8-variable balanced Boolean function with a nonlinearity of 118.

We first introduce the concept of couplings, coordinate decomposition, degree of descendibility, S-box coordinate extended linear approximation table (CELAT), as well as some useful properties and inner relationships. For convenience, let us denote as $f(n)^i$ the integer extracted from n, by flipping its i-th bit of its binary representation. Obviously, $f(f(n)^i)^i = n$.

Lemma 2.5.1 (The Parity Lemma)**.** Tweaking a bijective S-box S by flipping just one bit in its corresponding Look-up Table (LUT) will convert S to a non-bijective S-box.

Lemma 2.5.2 (Couplings Lemma)**.** The smallest nonzero number of bits from the LUT of a random bijective S-box that needed to be modified to obtain another bijective S-box is 2.

Definition 2.5.1 (Couplings)**.** Let us take a bijective S-box $S(n,n)$ and its corresponding DLUT

$$S_{DLUT} = [d_0, d_1, \cdots, d_i, \cdots, d_{2^n-1}].$$

We define as a **coupling** each set $\{d_s, f(d_s)^j\}$, while the set of all couplings in S as $\{S \updownarrow\}$.

Lemma 2.5.3 (Couplings Set Cardinality)**.** Given a bijective S-box $S(n,n)$:

$$|\{S \updownarrow\}| = n2^{n-1}.$$

Definition 2.5.2 (Couplings Pivot Set)**.** We define the set $\{S \updownarrow^i\}$ as the maximum subset of the coupling set of a bijective S-box S(n,n), which holds couplings operating only on column i of the S_{LUT}, i.e. couplings of the form $\{d_x, f(d_x)^i\}$. We call each such maximum subset $\{S \updownarrow^i\}$ a couplings pivot set operating on column i of S_{LUT}.

Corollary 2.5.1 (Properties of Couplings Pivot Sets)**.** Considering the definitions of the Couplings Pivot Sets on bijective S-box S(n,n), the following properties hold:

- $\forall i \neq j, \{S \updownarrow^i\} \cap \{S \updownarrow^j\} = \varnothing$
- $\forall i, |\{S \updownarrow^i\}| = 2^{n-1}$
- $|\bigcup_{i=1}^{n} \{S \updownarrow^i\}| = n2^{n-1}$

Definition 2.5.3 (Coordinate Decomposition)**.** Let S be an (n,n) bijective S-box. We take a random element with coordinates (x,y) of its corresponding linear approximation table S_{LAT}. We denote the binary representation of y as:

$$y_{(2)} = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \cdots + b_1 2^1 + b_0 2^0$$

The coordinate decomposition of an element with coordinates (x,y), denoted by $\triangle_{x,y}$, is the set:

$$\triangle_{x,y} = \bigcup_{i=0,b_i\neq 0}^{n-1} \{b_i(n-i-1)\}$$

Definition 2.5.4 (Nonlinearity Bottleneck Snapshot – NBS)**.** We define the nonlinearity bottleneck snapshot S_{NBS} of a bijective S-box $S(n,n)$ as a set of tuples holding all coordinates of the elements of S_{LAT}, which are holding down the nonlinearity value S_{NL} of S, i.e.

$$(x,y) \in S_{NBS} \Leftrightarrow |LAT_S[x][y]| = 2^{n-1} - S_{NL}$$

Definition 2.5.5 (NBS Coordinate Decomposition – NBSCD)**.** We define the nonlinearity bottleneck snapshot coordinate decomposition of a bijective S-box $S(n,n)$, denoted by Δ_S, as a set of all S_{NBS} coordinate decompositions, i.e.:

$$\Delta_S = \bigcup_{(x,y)\in S_{NBS}} \triangle_{x,y}$$

Definition 2.5.6 (Degree of Descendibility – Λ_S)**.** For a given bijective S-box $S(n,n)$, we define a family of sets Ψ_S, s.t.:

$$E \in \Psi_S \Leftrightarrow \forall Q \in \Delta_S \; \exists q \in Q : \; q \in E$$

The degree of descendibility of S is the minimum cardinality of a set in Ψ_S, i.e.:

$$\Lambda_S = \min_{\forall A\in\Psi_S} |A|$$

Corollary 2.5.2 (Basic properties of Λ_S)**.** For a given bijective S-box $S(n,n)$:

- $\Lambda_S \in \mathbb{N}$
- $\Lambda_S \in [1,n]$
- $\Lambda_S = 1 \Leftrightarrow |\bigcap_{s\in\Delta_S}| \geq 1$
- $\Lambda_S > 1 \Leftrightarrow \bigcap_{s\in\Delta_S} = \varnothing$

Definition 2.5.7 (Descendible Coordinate)**.** For a given bijective S-box $S(n,n)$, we say that coordinate j is descendible if the following properties hold:

- $\Lambda_S = 1$

- $j \in \bigcap_{s \in \Delta_S}$

Definition 2.5.8 (Couplings Transformation)**.** For a given bijective S-box $S(n,n)$ and some coupling c_i, we denote as S^{c_i} the S-box created by applying coupling c_i on S. We define this transform as coupling transform denoting it with the operator $\circ$, i.e.

$$S^{c_i} = S \circ c_i$$

When we have a list of couplings $\{c_1, c_2, \cdots, c_i\}$, which we want to use for transformation of S in this exact order, we will use the following expression:

$$S^{c_1, c_2, \cdots, c_i} = S \circ c_1 \circ c_2 \circ \cdots \circ c_i$$

Lemma 2.5.4 (Couplings Inverse)**.** For a given bijective S-box S and any coupling c, the following property holds:

$$S = S \circ c \circ c$$

Definition 2.5.9 (Coupling Transformation Matrix – CTM)**.** For a given bijective S-box $S(n,n)$ and some coupling c_i, we denote as ${S^{c_i}}_{LAT}$ the transformed LAT of S caused by c_i. We define the coupling transformation matrix of c_i on S, as:

$$S^{c_i}_{CTM} = {S^{c_i}}_{LAT} - S_{LAT}$$

Lemma 2.5.5 (Pivot Couplings Commutativity)**.** For a given bijective S-box $S(n,n)$, for any two couplings c_a and c_b, which belongs to the same couplings pivot set $\{S \updownarrow^i\}$, we have the following property:

$$S \circ c_a \circ c_b = S \circ c_b \circ c_a$$

Corollary 2.5.3. For a given bijective S-box $S(n,n)$, for any couplings c_j, which belongs to the same couplings pivot set $\{S \updownarrow^i\}$, we have the following properties:

$$S^{c_a, c_b}_{LAT} = S^{c_b, c_a}_{LAT} = S_{LAT} + S^{c_a}_{CTM} + S^{c_b}_{CTM}$$

$$S^{c_1, c_2, \cdots, c_k}_{LAT} = S_{LAT} + \sum_{i=1}^{k} S^{c_i}_{CTM}$$

Lemma 2.5.6 (CTM Values)**.** The value of each element in a CTM is -2, 0, or 2.

Corollary 2.5.4. For a given bijective S-box $S(n,n)$, let us apply transformations of couplings $c_1, c_2, \cdots, c_k$, which belongs to the same couplings pivot set $\{S \updownarrow^i\}$. The elements of the resulting CTM are numbers in the interval $[-2k, -2(k-1), \cdots, -2, 0, 2, \cdots, 2(k-1), 2k]$.

Definition 2.5.10 (S-box Coordinate Extended LAT – CELAT)**.** For a given bijective S-box $S(n,n)$, and a given coordinate i, we can define the one-dimensional linear approximation table of S as:

$$S_{LAT_{1D}}[x] = S_{LAT}[x \;/\; 2^n][x \;\%\; 2^n]$$

Furthermore, we denote all the couplings in the couplings pivot set $\{S \updownarrow^i\}$ as $c_1, c_2, \cdots, c_{2^n-1}$. We have:

$$\begin{aligned} S^{c_1}_{CTM} &= S^{c_1}_{LAT} - S_{LAT} \\ S^{c_2}_{CTM} &= S^{c_2}_{LAT} - S_{LAT} \\ &\cdots \\ S^{c_{2^n-1}}_{CTM} &= S^{c_{2^n-1}}_{LAT} - S_{LAT} \end{aligned} \tag{2.6}$$

Following the same concept used in the construction of one-dimensional LAT of S, we can define one-dimensional CTM, i.e.:

$$\begin{aligned} S^{c_1}_{CTM_{1D}} &= S^{c_1}_{LAT_{1D}} - S_{LAT_{1D}} \\ S^{c_2}_{CTM_{1D}} &= S^{c_2}_{LAT_{1D}} - S_{LAT_{1D}} \\ &\cdots \\ S^{c_{2^n-1}}_{CTM_{1D}} &= S^{c_{2^n-1}}_{LAT_{1D}} - S_{LAT_{1D}} \end{aligned} \tag{2.7}$$

Finally, we define S-box i-th Coordinate Extended LAT S^i_{CELAT} as the following table:

$$S^i_{CELAT} = \begin{bmatrix} S_{LAT_{1D}} \\ S^{c_1}_{CTM_{1D}} \\ S^{c_2}_{CTM_{1D}} \\ \cdots \\ S^{c_{2^n-1}}_{CTM_{1D}} \end{bmatrix}$$

S^i_{CELAT} has $2^{n-1}+1$ rows and 2^{2n} columns.

Definition 2.5.11 (Binary Integer Programming – Feasibility or SAT Problem)**.** A feasibility binary integer program is a problem of the form:

$$\begin{array}{lrcl} \text{subject to} & Ax & \leq & b \\ & x & \geq & 0 \quad \text{binary} \end{array}$$

where the data consists of (m,n)-matrix A and column vectors b and x with respective sizes of m and n. The column vector x contains the binary variables to be optimized. We say that the set S is the set of feasible solutions, i.e.:

$$S := \left\{x \in B^n : Ax \leq b\right\}$$

In the context of the feasibility problem we are looking for just one element in the set S, not the optimal one.

For an (n,n) S-box S, we denote 2^{n-1} by r and 2^{2n} by m. Let us construct its CELAT using coordinate i i.e:

$$S^i_{CELAT} = \begin{bmatrix} S_{LAT_{1D}} \\ S^{c_1}_{CTM_{1D}} \\ S^{c_2}_{CTM_{1D}} \\ \cdots \\ S^{c_{2^n-1}}_{CTM_{1D}} \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & \cdots & l_m \\ c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r1} & c_{r2} & \cdots & c_{rm} \end{bmatrix}$$

We want to apply some coupling transformations subset $P = p_1, p_2, \cdots, p_k$ which belongs to the pivot coupling set $\{S \updownarrow^i\}$. From corollary 2.5.3 it follows that:

$$S^{p_1,p_2,\cdots,p_k}_{LAT} = S_{LAT} + \sum_{i=1}^{k} S^{p_i}_{CTM}$$

We denote

$$S^{p_1,p_2,\cdots,p_k}_{LAT_{1D}} = [q_1, q_2, \cdots, q_m]$$

Then, we can construct the following system of equations:

$$\begin{aligned} q_1 &= l_1 + c_{11}x_1 + c_{21}x_2 + \cdots + c_{r1}x_r \\ q_2 &= l_2 + c_{12}x_1 + c_{22}x_2 + \cdots + c_{r2}x_r \\ &\cdots \\ q_m &= l_m + c_{1m}x_1 + c_{2m}x_2 + \cdots + c_{rm}x_r \end{aligned} \tag{2.8}$$

where $x = (x_1, x_2, \cdots, x_r) \in B^r$, and $x_t = 1$ iff $p_t \in P$. We have $S_{NL} = 2^{n-1} - \max^m_{j=1} abs(l_j)$. If coordinate i is descendable, we can construct the following binary integer programming

feasibility problem:

$$\begin{array}{lccll} \text{subject to} & < S^i_{CELAT}{}^T, x > & \leq & A & \\ \text{subject to} & < S^i_{CELAT}{}^T, x > & \geq & B & \\ & x & \geq & 0 & \text{binary} \end{array}$$

where A is a column vector with $2^{n-1}+1$ elements, each equal to $2^{n-1}-S_{NL}-2$, while B is a column vector with $2^{n-1}+1$ elements, each equal to $S_{NL}-2^{n-1}+2$. Let us denote the SAT problem descending on coordinate i in equation 2.5 as $\Omega_{S,i}$. This is NP-hard [2] problem with a total of 2^{n-1} binary variables and 2^n+2 restrictions. However, we can further divide the problem to an union of subproblems, i.e.:

$$\Omega_{S,i} = \bigcup_{d=1}^{n-1} \Omega^d_{S,i}$$

where each subproblem $\Omega^d_{S,i}$ is modelled using the following restrictions:

$$\begin{array}{lccll} \text{subject to} & < S^i_{CELAT}{}^T, x > & \leq & A & \\ \text{subject to} & < S^i_{CELAT}{}^T, x > & \geq & B & \\ \text{subject to} & \sum_{j=1}^{r} x_j & = & d & \\ & x & \geq & 0 & \text{binary} \end{array}$$

Solving any of the subproblems will yield a solution to the original problem.

For subproblems $\Omega^d_{S,i}$ of a binary integer programming feasibility problem $\Omega_{S,i}$, the following property holds:

$$\bigcap_{d=1}^{n-1} \Omega^d_{S,i} = \varnothing$$

It is easy to show that the search space of the subproblem $\Omega^d_{S,i}$ for the bijective S-box $S(n,n)$ is $\binom{2^{n-1}}{d}$.

Theorem 2.5.1. For a subproblem $\Omega^d_{S,i}$, all restrictions with the participation of some l_j for which the following inequalities hold:

$$\begin{aligned} l_j &\leq 2^{n-1} - S_{NL} - 2d - 2 \\ l_j &\geq S_{NL} - 2^{n-1} + 2d + 2 \end{aligned} \tag{2.9}$$

[2] The complexity class of decision problems that are intrinsically harder than those that can be solved by a nondeterministic Turing machine in polynomial time.

are always satisfied.

Definition 2.5.12 (CELAT with radius R)**.** For a given bijective S-box $S(n,n)$, and a given coordinate i, we have:

$$S^i_{CELAT} = \begin{bmatrix} l_1 & l_2 & \cdots & l_m \\ c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r1} & c_{r2} & \cdots & c_{rm} \end{bmatrix}$$

We define as $S^{i,R}_{CELAT}$ a matrix constructed of those columns of S^i_{CELAT} with first element ρ, for which the following inequalities hold:

$$\begin{aligned} \rho &> 2^{n-1} - S_{NL} - 2R - 2 \\ \rho &< S_{NL} - 2^{n-1} + 2R + 2 \end{aligned} \tag{2.10}$$

Hence, a given suproblem $\Omega^d_{S,i}$ could be further reduced and launched on $S^{i,d}_{CELAT}$, instead of its corresponding full (unreduced) version S^i_{CELAT}.

We have implemented the algorithm by using Python and the Gurobi SAT Solver [71]. We analyzed two famous S-boxes: the Skipjack S-box, developed by the U.S. National Security Agency (NSA) [138], which we will denote as S_k, and the Kuznyechik S-box, standardized by the Russian Federation's standardization agency [53], which we will denote as K_k. We have shown how S_k and K_k could be optimized to S-boxes with higher nonlinearities by tweaking respectively just 4 and 12 bits (out of 2048).

2.5.1 The ACNV problem

The ACNV optimization problem could be represented as a special, and significantly lighter, in terms of computational burden, case of $S^{i,R}_{CELAT}$, where S denotes the initial S-box and i denotes the coordinate of S to be optimized.

We have initiated the optimization routine on a bijective S-box from [50], possessing the highest, currently known, ACNV of 114.5. It is composed of 6 coordinates with a nonlinearity value of 114 and 2 coordinates with a nonlinearity value of 116. The algorithm was able to optimize it to an S-box with ACNV 116 with overall nonlinearity of 92. Significant efforts were made to reach higher ACNV - reaching higher ACNV would reveal a balanced Boolean function having a nonlinearity of 118. Unfortunately, all tried instances were proofed infeasible.

Chapter 3

On the S-box Reverse Engineering

3.1 Introduction and motivation

The reasons for obfuscating the design of a given S-box are manifold. For example, the initial S-boxes used in the Data Encryption Standard (DES) [55] were originally modified by NSA. The reasons for applying those modifications were not known. However, in [33], D. Coppersmith announces the motivation behind the S-box modifications. It appears that the agency knew about the existence of differential attacks about 20 years before the academic world. However, they kept that in secrecy. D. Coppersmith further commented on this secrecy decision by saying:

> *... that was because [differential cryptanalysis] can be a very powerful tool, used against many schemes, and there was concern that such information in the public domain could adversely affect national security.*

Another reason for hiding a given S-box design could be related to some hidden structure, the knowledge of which could be exploited to gain a significant advantage in terms of hardware implementation. For example, as discovered in [21], the S-boxes used in the hash function Streebog and the 128-bit block cipher Kuznyechik, standardized by the Russian Federation, are designed with such a hidden structure. A user knowing the not published decomposition could implement the given S-box with a significantly smaller hardware footprint, allowing him to reach an up to 8 times faster S-box look-up.

Another practical reason for hiding the design of a given S-box could be related to an encapsulated trapdoor as discussed in [128]. Although the aforementioned trapdoor can be easily detected, as shown in [151], the motivation for finding other trapdoor S-box techniques should not be underestimated.

There are various tools and techniques, which could help us to initiate some S-box reverse engineering (see [119][20][120]). In the next section, the concept of S-box spectrography is presented. A good example of using spectrography for S-box reverse engineering purposes is the Pollock representation (see [21]).

3.2 S-box spectrography

We can isolate the coordinates, in terms of row and column indexes, of those elements of the LAT of a given S-box $S(n,m)$, which are equal to some fixed value or, in the more unrestricted case, belong to some set of values of our choice. We define each distinct isolation as a **spectra channel**. For convenience, we denote as $\S_S^E$ the spectra channel isolated from an S-box S, by using restriction set E. We can further visualize the channel as a $(2^n \times 2^m)$ matrix plot – those elements, which belong to the restriction set are colored in red, while the remaining elements are left colorless.

During our experiments, we repeatedly generated random bijective $(8,8)$ S-boxes and thoroughly analyzed their spectra channels. However, we didn't find any anomalies, symmetries, or visual patterns. It is really difficult to distinguish visually their spectra channel plots from plots populated with randomly scattered points.

In [132] a rich database of popular S-boxes is published. The rest of this section presents our results in applying spectra channel analysis on the aforementioned S-box collection.

Anubis is a block cipher, which was submitted to the NESSIE project [127]. The Anubis S-box is constructed by using involutions. It appears that such constructions are easily detected by using some spectra channel plot of the form $\S^{-x,x}$.

CLEFIA is a 128-bit block cipher supporting key lengths of 128, 192 and 256 bits [137]. We analyzed its S-boxes to find anomalies in their plots. There are respectively vertical and horizontal red lines immediately next to the x and y axis, while a complete red square is visible in the upper-left of the matrix plot.

The Cellular Message Encryption Algorithm (CMEA) is a US block cipher that was used for securing mobile phone communications [126]. By analyzing the $\S^0_{CMEA}$ plot we found anomalies immediately next to the y-axis horizontal red lines.

Crypton is a new 128-bit block cipher algorithm proposal for AES. The S-box in the first version (S_0) [93] was further revised and replaced by four S-boxes (S_1, S_2, S_3 and S_4) [94]. Anomalies are found in S_0 by the restriction $\{-8,8\}$. All revised Crypton S-boxes possess anomalies in their spectra channel plots as well.

Another NESSIE project block cipher submission is the CS-cipher [144]. By using spectra channel $\S^0_{CS}$ a picturesque plot was discovered.

The content scrambling system (CSS) [13] is used to encode DVDs. We analyzed the implemented S-box to find sections with anomalies. We further analyzed the S-boxes published in Enocoro [148], Fantomas [69], FLY [83], Fox [82] and Iceberg [143]. Enocoro anomalies are clearly visible in spectra channel with restriction $E = \{0\}$. White rectangular covering the lower values on x-axis of Fantomas is detected on spectra channel $\S^{-4,4}_{Fantomas}$, while smaller almost perfect rectangulars are visible on the x-axis of FLY spectra channel $\S^{-8,8}_{FLY}$. Analyzing Fox by applying spectra channel $\S^{-4,4}_{Fox}$ reveals a grid-alike structure. The Iceberg S-box is involution.

Anomalies are found in Iraqi [150], iScream [70], Khazad [10], Lilliput [3] and Picaro [123]. The spectra channel for $\S^{-1,1}_{Iraqi}$ is distinguishable from pseudo-randomly generated S-box by the striped-alike structure. Furthermore, we can deduce from $\S^{-1,1}_{Iraqi}$ that the Iraqi S-box is not bijective. Fractal-alike structure is revealed in plot $\S^{-4,4}_{iScream}$, while involution is observed in $\S^{0}_{Khazad}$. Analyzing the Lilliput S-box a Tetris-alike structure is revealed on spectra channel $\S^{-4,4}_{Lilliput}$, while fence-alike structure is clearly visible in Picaro S-box on spectra channel $\S^{-8,8}_{Picaro}$.

By applying the same method we were able to detect anomalies in Safer [102], Scream [30], SKINNY [5], SNOW 3G [117] and Twofish [134]. $\S^{-4,4}_{Scream}$ revealed a very curious pattern in Scream S-box. $\S^{-4,4}_{SKINNY}$ is heavily partitioned, while $\S^{-2,2}_{SNOW3G}$ is completely blank, which, for example, is completely unusual for a pseudo-randomly generated S-box. In Twofish, two S-boxes π_0 and π_1 are used. Both of them are very similar in terms of their spectra channels. Furthermore, they are distinguishable from pseudo-randomly generated S-boxes as well (lines on the y-axis are visible).

Finally, we analyzed the S-boxes used in Whirlpool [12], Zorro [58] and ZUC [152]. All of them are easily distinguishable from pseudo-randomly generated S-boxes by using spectra channels $\S^{0}_{Whirlpool}$, $\S^{-2,2}_{Zorro}$ and $\S^{-8,8}_{ZUCS_0}$.

3.3 Automatic spectral analysis of S-box LAT, DDT, XORT, ACT spectras

We could automate the process of anomaly discovery in a given S-box S LAT spectra. Moreover, it could be easily generalized for other spectras of S like the DDT, ACT, and XORT.

S_{LAT} has 2^n columns. We denote as $S^T_{LAT}[i]$ the i-th column of S_{LAT}. We further denote as $\sigma(S,i,e)$ the total number of occurrences of e and $-e$ in $S^T_{LAT}[i]$, while $\sigma_{ind}(S,e)$ denotes

the set of indexes of columns of S_{LAT}, s.t:

$$\forall_{i_1 \neq i_2, i_1, i_2 \in \sigma_{ind}(S,e)} : \sigma(S, i_1, e) \equiv \sigma(S, i_2, e).$$

For some reasonable threshold value t and two different values e_1 and e_2, in respect of pseudo-randomly generated S-box, $\sigma_{ind}(S, e_1) \equiv \sigma_{ind}(S, e_2)$, where $\sigma(S, i, e_1) \geq t$ and $\sigma(S, i, e_2) \geq t$, is highly unlikely. During our experiments, we generated more than 10^5 pseudo-random S-boxes. Only in 0.3% of all generated S-boxes a collision was found and always with length 8. Let's denote such collision as $\Gamma(S, t, e_1, e_2, I)$, where I is a set of indexes of columns of S_{LAT}.

We have found a collision in the Russian Federation's standardization agency Kuznyechik S-box [68], which was not visible during our spectra channel analysis. The indexes of the collision confirm the observations made in [21]. We can apply the same strategy on LAT rows (instead of columns). Collisions are found in the state standard of Republic of Belarus (BelT) [131]. We further analyzed various S-box DDT spectra. We find a collision in π_3 S-box of the new encryption standard of Ukraine Kalyna [116]. By applying the same method to the transformed DDT, more collisions are found in Kalyna and BelT. We found collisions in various S-boxes by using the ACT spectra as well.

We further analyzed the XORT of various S-boxes. A visual interpretation of some XORT relies on the order in which the columns of XORT are populated. In the original definition, the columns are populated in lexicographical order. However, we can tweak that order and populate the XORT by first plotting the n coordinates of a given S-box $S(n, n)$, then all linear combinations of S coordinates with two terms, three terms, and so on, until the last column, which is the XOR of all n coordinates. Such rearrangement makes sense since we group the XORs of the main building blocks of the S-box (the coordinates) into the most significant columns of XORT (the left ones). We analyzed the XORT and rearranged XORT plot of BelT S-box. The lexicographically sorted XOR reveals some vertical lines, which is not unusual for XOR tables of pseudo-randomly generated S-boxes. However, the rearranged XORT reveals some interesting leafs-alike patterns in the upper left section. Furthermore, each consequent column is similar to the previous column when upward-slide.

Chapter 4

Binary Sequences and Their Autocorrelation

Sequences with low autocorrelation functions are necessary for a variety of signal and information-processing applications. For example, in pulse codes-based compression for radars and sonars, such sequences are used to obtain high resolution. The shifts of sequences with low autocorrelation can be also used for better synchronization purposes or to identify users in multi-user systems. Due to their big practical importance, these sequences have been widely studied and various methods for constructing sequences with small values of autocorrelation are developed.

Let $B = (b_0, b_1, \cdots, b_{n-1})$ be a binary sequence of length $n > 1$, where $b_i \in \{-1, 1\}, 0 \leq i \leq n-1$. The **aperiodic autocorrelation function** (AACF) of B is given by

$$C_u(B) = \sum_{j=0}^{n-u-1} b_j b_{j+u}, \quad for\ u \in \{0, 1, \cdots, n-1\}.$$

We will note that the AACF is originally defined in the interval

$$\{-n+1, -n+2, \cdots, -2, -1, 0, 1, 2, \cdots, n-1\}.$$

As the AACF is an even function with $C_u(B) = -C_u(B)$, we will consider it for the interval $\{0, 1, \cdots, n-1\}$ only. The $C_0(B)$ is called **mainlobe** and the rest $C_u(B)$ for $u \in \{1, \cdots, n-1\}$ are called **sidelobe** levels. We define the **peak sidelobe level (PSL)** [146] of B as

$$PSL(B) = \max_{0<u<n} |C_u(B)|.$$

The value of the PSL can also be represented in decibels

$$PSL_{db}(B) = 20\log\left(\frac{PSL(B)}{n}\right).$$

Another important measure of an AACF is the **merit factor** (MF), which gives the ratio of the energy of the mainlobe level to the energy of sidelobe levels, i.e.

$$MF(B) = \frac{C_0(B)}{2\sum_{u=1}^{n-1}|C_u(B)|^2}.$$

The binary sequences of low autocorrelation are of special interest and some of the well known such sequences are the Barker codes [9], M-sequences [67], Gold codes [66], Kasami codes [84], Weil sequences [130], Legendre sets [124] and others (see [92][139]). Barker sequences are known to have the best autocorrelation properties, but the longest such sequence is of length 13. M-sequences, Gold codes, and Kasami sequences have ideal periodic autocorrelation functions but have no constraints on the sidelobes of their aperiodic autocorrelation functions. As summarized in [111], during the years a variety of analytical constructions and computer search methods are developed to construct binary sequences with relatively minimal PSL. By an exhaustive search the minimum values of the PSL for $n \leq 40$[96], $n \leq 48$[8], $n = 64$[35], $n \leq 68$[88], $n \leq 74$[90], $n \leq 80$ [91], $n \leq 82$ [89] and $n \leq 84$ [87] are obtained. The best currently known values for PSL for $85 \leq n \leq 105$ are published in [112], and for $n \geq 106$ in [54].

4.1 Efficient Generation of Low Autocorrelation Binary Sequences

In this section an efficient and easy-to-implement heuristic algorithm is suggested and, as an illustration of its effectiveness, it was further utilized for the generation of binary sequences with lengths between 106 and 300. The generated sequences are better, in terms of PSL values than a significant part of those obtained in [54] ones. The algorithm can also be used for the generation of sequences with lengths greater than 300.

Since our goal is to lower the PSL of a given binary sequence, i.e. to lower the value of PSL(B), it makes sense to simultaneously lower the values of each $C_u(B)$, for $u \in$

$\{1, \cdots, n-1\}$. By making this observation, we define the following fitness function:

$$F(B) = \sum_{u=1}^{n-1} |C_u(B)|^P = \sum_{u=1}^{n-1} \left(|\sum_{j=0}^{n-u-1} b_j b_{j+u}| \right)^P ,$$

where P is the magnitude of the fitness function, i.e. the higher the magnitude is the higher the fitness function intolerance to large absolute values of $C_u(B)$'s will be. We made experiments with various values of P and the best results were obtained for values in the interval $[3,5]$. Lower values of P make the fitness function too tolerant to higher absolute values of the PSLs $C_u(B)$, while higher values of P are heavily populating the heuristic topology with local minimums. We have fixed the magnitude P of the fitness function to 4.

Let's denote the i-th position of a binary sequence B of length n as b_i. Flipping the i-th position of B is to interchange b_i with $-b_i$. By the neighborhood of the binary sequence B, denoted by $N(B)$, we define the set of all binary sequences constructed from B by making a single flip in B.

The optimization process takes as input the length of the binary sequence n, the fitness function F, the threshold value t, the two integers h_{min} and h_{max} defining the flipping allowance interval, and the goal G which is the desired final PSL value to be reached.

In the beginning, we generate a random binary sequence B of length n. Then, by searching the neighborhood of B, we look for a better binary sequence, i.e. a binary sequence with a smaller fitness value. If some X out of the neighbors of B has PSL equal to G we output X and quit. If during the search of the neighborhood no better binary sequence is found, we are stuck in some local minimum B'. In order to escape the local minimum we flip h randomly chosen elements of B', where $h \in [h_{min}, h_{max}]$. We will call such try a **quake**. In the case when t consecutive quakes are not sufficient to escape the local minimum, we start the process from the beginning by randomly generating a new binary sequence, i.e. the shotgun hill-climbing approach. The algorithm stops when a binary sequence with the searched value of the PSL is found or when the preliminary defined number of restarts is reached.

We improve the PSL values for 95 from the launched 195 lengths. The remaining 100 binary sequences have the same values of the PSL as the currently known best ones. Furthermore, all of them are unique and unpublished before.

The suggested in this section algorithm is highly parallelizable so that a multicore architecture can be fully utilized. It is implemented in Python on a single mid-range computer with an octa-core CPU. During our experiments, the time required to reach a given PSL goal was between a few minutes to several hours. Furthermore, with each instance of the

algorithm, we repeatedly reached binary sequences with lower or the same PSL than the state-of-the-art algorithms.

4.2 On the Generation of Long Binary Sequences with Record-Breaking PSL Values

M-sequences, Gold codes, and Kasami sequences have ideal periodic autocorrelation functions but have no constraints on the sidelobes of their aperiodic autocorrelation functions, i.e. their PSL value is not pre-determined. The same is true for Legendre sets and Rudin-Shapiro sequences. Furthermore, it is difficult to calculate the growth of the PSL of the aforementioned families of binary sequences. It is conjectured that the PSL values of m-sequences grow like $\mathcal{O}(\sqrt{n})$, making them one of the best methods to straightforwardly construct a binary sequence with near-optimal PSL value. However, as stated in [81]:

> *The claim that the PSL of m-sequences grows like $\mathcal{O}(\sqrt{n})$, which appears frequently in the radar literature, is concluded to be unproven and not currently supported by data.*

As summarized in [111], during the years a variety of analytical constructions and computer search methods are developed to construct binary sequences with relatively minimal PSL. It appears that the current state of art computer search methods, like CAN [73], ITROX [140], MWISL-Diag, MM-PSL [141] or DPM [86], could yield better, or at least not worse PSL values, than the algebraic constructions. However, when the length of the generated by a given heuristic algorithm binary sequences rises, so is the overall time and memory complexity of the routine. As concluded in [109]:

> *As an indication of the runtime complexity of our EA[1], the computing time is 58009 s or 16.1136 h for L=1019. For lengths up to 4096, the computing time required empirically shows a seemingly quadratic growth with L.*

Thus, the main motivation of this section is to create an efficient and lightweight algorithm, in terms of time and memory complexity, to address the heuristic generation of very long binary sequences with near-optimal PSL values.

Let us denote $C_{n-i-1}(B)$ by $\hat{C}_i(B)$. Since this is just a rearrangement of the sidelobes of B, it follows that:

$$B_{PSL} = \max_{0<u<n} |C_u(B)| = \max_{0\leq u<n-1} |\hat{C}_u(B)|.$$

[1] EA stands for Evolutionary Algorithm

We denote as Ω_Ψ the array of all the consequent sidelobes of Ψ. The calculation of Ω_Ψ, corresponding to some random binary sequence Ψ, is not linear. The time complexity of the trivial computational approach is $\mathcal{O}(n^2)$ (two nested **for** cycles). However, as shown in Wiener–Khinchin–Einstein theorem [149], the autocorrelation function of a wide-sense-stationary random process has a spectral decomposition given by the power spectrum of that process, we can use one regular and one inverse Fast Fourier Transform (FFT), to achieve a faster way of calculating Ω_Ψ. Despite its time complexity of $\mathcal{O}(n \log n)$, its memory requirement is significantly higher than the trivial computational approach.

By exploiting the observations made in this section, we present an algorithm that can calculate the array Ω_{Ψ_f} (Ψ_f corresponds to Ψ with flipped element f), if we hold the array Ω_Ψ in memory, with time and memory complexity of $\mathcal{O}(n)$.

We have implemented the algorithm by using the C language and a mid-range computer station. Given the linear time and memory complexity of the algorithm, we were able to repeatedly generate binary sequences with record-breaking PSL values for less than a second. As stated in [109], the time required to reach a PSL value 26, for a binary sequence with length 1019, is 58009 seconds or 16.1136 hours. For comparison, we reach this value for less than a second.

We present the results achieved by the algorithm, for binary sequences with lengths x^2 for $x \in [18, 44]$, compared with the currently known state of art algorithms found in the literature, like CAN [73], ITROX [140], MWISL-Diag, MM-PSL [141], DPM [86], 1bCAN [95]. We will refer to this collection of algorithms as collection **A**. We want to emphasize, that the differences between the proposed algorithm with algorithms from collection A are manifold. For example, we do not use converging functions, mini regular or quadratic optimization problems, or floating-based arithmetic. Furthermore, the provided algorithm does not suffer from a unique navigation trace through the sequence search space. The experiments were based on 12 instances of each algorithm (each ran to a distinct thread of the processor). Furthermore, the lifetime of our algorithm is restricted to 1 minute. We significantly outperform the best results achieved by state-of-the-art algorithms. In fact, for some of the lengths, less than a second was needed to reach a record-breaking PSL.

In contrast to some other state-of-the-art algorithms, the computing complexity of the algorithm presented in this work does not grow quadratically. Maybe this is the reason for the lack of published results for binary sequences of lengths greater than 2^{12}. Nevertheless, the results with which we can further compare are m-sequences. However, such sequences exists only for lengths $2^n - 1$, $n \geq 1, n \in N$. Our results significantly outperform the best results achieved by m-sequences.

4.3 Hybrid Constructions of Binary Sequences with Low Autocorrelation Sidelobes

An m-sequence $M = (x_0, x_1, \cdots, x_{2^m-2})$ of length $2^m - 1$ is defined by:

$$x_i = (-1)^{Tr(\beta\alpha^i)}, \text{ for } 0 \leq i < 2^m - 1,$$

where α is a primitive element of the field $\mathbb{F}_{2^m}$, $\beta \in \mathbb{F}_{2^m}$, and Tr is denoting the trace function from $\mathbb{F}_{2^m}$ to $\mathbb{F}_2$.

Given an odd prime p, a Legendre sequence L with length p is defined by:

$$L_i = \begin{cases} 1, \text{ if } i \text{ is a quadratic residue mod } p \\ -1, \text{ otherwise.} \end{cases}$$

We denote as $B \leftarrow \rho$ the binary sequence obtained from B, by left-rotating it ρ times. By definition, $B \leftarrow |B| \equiv B$. Furthermore, if b_i is the element of B on position i, we will denote as $b_i^{\leftarrow \rho}$ the element of $B \leftarrow \rho$ on position i.

A comparison, in terms of algorithm efficiency (the ratio of the beneficial work performed by the algorithm to the total energy invested) and actual effectiveness (the quality of the achieved results), was made. The best results were achieved by the SHC (Shotgun Hill Climbing) algorithm, regarding the binary sequences with lengths less than 300, and HC (Hill Climbing), for all the remaining lengths. However, the approximated binary sequence's length, from which HC starts outperforming SHC, is fuzzy.

As observed in Section 4.2 of this thesis, or [49], the PSL-optimization process of very long binary sequences is a time-consuming routine, despite the algorithm's linear time and memory complexities. Thus, HC avoids restarts, i.e. re-initializing the starting state with a pseudo-random binary sequence. However, re-initialization appears to be significantly beneficial when dealing with PSL optimization of binary sequences with relatively small lengths, such as the SHC algorithm.

By considering the observations made above, we have revisited the SHC algorithm.

Considering the significant changes made in the SHC algorithm, the fitness function parameters are carefully analyzed, re-evaluated, and updated. Given a binary sequence Ψ, both algorithms (SHC and HC) are sharing the same fitness function F, s.t:

$$F(\Psi) = \sum_{x \in \Omega_\Psi} |x|^4 = \sum_{x \in \Omega_\Psi} x^4$$

During our experiments, we used a single general-purpose computer with a 6-cored central processing unit architecture, capable of running 12 threads simultaneously. By using the SHC revisited kernel, we were able to reach binary sequences with optimal PSL values for each length in $[1,82]$. Given the linear time and memory complexities of the algorithm, for the majority of those lengths, the PSL-optimal binary sequences were reached for less than a minute.

We have further launched the algorithm on binary sequences with lengths up to 300. Almost all of the results known in the literature were improved. More precisely, we have improved 179 out of 195 cases. Curiously, for some lengths, we have even revealed binary sequences with record-breaking PSL values, having a distance of 2 to the previously known PSL record value.

In [36], the best results achieved by the D-Wave 2 quantum computer for binary sequences with length 128 is PSL 8, while our algorithm could reach PSL 6. For longer lengths, for example, binary sequences with lengths 256, the best PSL achieved by the D-Wave 2 quantum computer was 12, while during our experiments we reached PSL values of 10. We reached PSL values of 10 for binary sequences up to 271. For completeness, since the D-Wave 2 quantum computer is tested on binary sequences with length 426, we have further launched the algorithm on the same length. Surprisingly, the algorithm was able to find binary sequences with PSL values of 17 (the best value achieved by the quantum computer) for less than a second. It reached PSL values of 16, and even 15, for less than a second as well. However, PSL value of 14 was noticeable harder to reach (199 seconds).

Recently, in [37] a multi-thread evolutionary search algorithm was proposed. We were able to improve almost all of the best PSL values from the aforementioned paper - usually for less than a second. For example, the best PSL value for binary sequences with length 3000 achieved in [37] is 51. We have launched the algorithm on binary sequences with the same length. Record-breaking PSL values of 44 and 43 were reached for respectively 111 and 371 seconds.

The reasoning behind announcing one binary sequence as long, or short, is ambiguous. Measuring the largeness of a given binary sequence is probably more related to the capabilities of the used algorithm than the actual length itself. From a practical point of view, some algorithms, or their implementations, would not even start the optimization (or construction) process, since their computational capabilities (or hardware restrictions) would not be able to process the desired length. For example, as discussed in [36], the usage of a 512-qubit D-Wave 2 quantum computer limits the code length that can be handled, to at most 426, due to a combination of overhead operations and qubits unavailability. Moreover, it was estimated that a 2048-qubit D-Wave computer could handle binary sequences with lengths

up to 2000. Hence, the exact fixed value differentiating short from long binary sequences is still unclear.

From now on, we denote the algorithm in this section as $\mathscr{A}$ with fixed power of the fitness function to 4 if not specified otherwise.

4.3.1 Using $\mathscr{A}$ as an m-sequences extension

The following procedure is proposed:

- Choose a primitive polynomial f over F_{2^m}
- Fix an initial element a over F_{2^m}
- Convert f to a linear-feedback shift register $\mathscr{L}$
- Expand the $\mathscr{L}$ to a binary sequence L, $|L| = 2^m - 1$.
- Launch $\mathscr{A}$ with L as an input

The primitive polynomials over F_{2^m} could be calculated in advance. Furthermore, the PSL of L, where L is seeded by some initial element a over F_{2^m}, could be specially chosen to have the minimum possible value. This is easily achievable by using the theorems discussed later in this chapter (see Subsection 4.3.3):

We were able to repeatedly reach record-breaking binary sequences of length 131071 having PSL equal to 359. The time required was less than 2 minutes, which was a significant improvement over the time required for $\mathscr{A}$ (starting from pseudo-randomly generated sequences) to reach binary sequences with PSL close to 359: approximately 3 days. Leaving $\mathscr{A}$ to work for another 46 minutes it even reached binary sequences of length 131071 with PSL 356.

The proposed procedure, as demonstrated, is highly efficient and is capable to reach binary sequences with $\mathscr{A}$-long lengths and record-breaking PSL values for a few minutes. Unfortunately, it is applicable on binary sequences with lengths of the form $2^n - 1$ only. However, throughout the next section, we provide another procedure that can generate binary sequences with length p and record-breaking PSL values, where p is a prime number.

4.3.2 Using $\mathscr{A}$ as an Legendre-sequences extension

The following procedure is proposed:

- Choose a prime number p

- Generate the sequence $L = \left[t_1, t_2, \cdots, t_p\right]$
- For i, s.t. $i \in N$, $1 \leq i \leq p$, and in case i is a quadratic residue mod p, replace t_i with 1. Otherwise, replace t_i with -1.
- Launch $\mathscr{A}$ with L as an input

As the numerical experiments suggested in [44], it is highly unlikely that a Legendre sequence with length p, for $p > 235723$, or any rotation of it, would yield a PSL value less than $\sqrt{p}$. Having this in mind, experiments with initializing $\mathscr{A}$ (α=8) with a rotation of Legendre sequence with length 235747 were made (the next prime number after 235723). Again, by using SIMD-capable devices, we have extracted the PSL-optimal rotation among all possible rotations of a Legendre sequence with length 235747. More precisely, on rotation 60547, a binary sequence with PSL equal to 508 was yielded. $\mathscr{A}$ was able to significantly optimize this binary sequence. For less than 25 minutes, using only 1 thread of a Xeon-2640 CPU with a base frequency of 2.50 GHz, a binary sequence with PSL equal to 408 was found.

Since $\sqrt{235747} \approx 485.54$, it follows that 408 is significantly smaller than the expected value of 485.54. In fact, by leaving $\mathscr{A}$ for a total of 2.21 hours, a binary sequence with length 235747 and PSL 400 was reached. More details could be found in [47].

4.3.3 On the Aperiodic Autocorrelations of Rotated Binary Sequences

The maximal length shift register sequences, or m-sequences, is a well-known algebraic design [67]. Unfortunately, they are defined for lengths $2^n - 1$ only ($n \in N$). Nevertheless, as shown in [52], their extensive study could provide valuable insights into understanding the world of binary sequences possessing low aperiodic autocorrelation characteristics. However, finding the PSL-optimal m-sequences is a rigid and tedious task - during each iteration, the PSL value of a given binary sequence B, altogether with all possible rotations of B, should be calculated. In [81], an exhaustive search of PSL-optimal m-sequence with lengths up to $2^{15} - 1$ is given. Later, in [52], the exhaustive search study was extended with results regarding m-sequences with lengths $2^{16} - 1$ and $2^{17} - 1$. Since then, no progress was made.

Similar to the problem of finding PSL-optimal m-sequences, finding PSL-optimal Legendre sequences involves a significant computational burden - during each iteration, the PSL value of the binary sequence, altogether with all possible rotations of B, should be calculated. This explains why the numerical results regarding the PSL-optimal Legendre sequences are scarce. For example, in [135], Fig.4, a list of all PSL-optimal Legendre sequences, up to length 3500 only, is given.

The routine of finding the minimum PSL among all the possible rotations of a given binary sequence plays an important role in the overall computational burden. By making some observations of the behavior of the sidelobes array in a rotated sequence, we were able to project the routine to a perfectly balanced parallelizable algorithm. This allows us to efficiently utilize the computational possibilities of modern GPUs. Hence, we were able to exhaustive search all m-sequences with lengths $2^{18}-1$, $2^{19}-1$ and $2^{20}-1$, as well as finding all optimal Legendre sequences with lengths up to 432100 - something out of reasonable computational reach until now.

We denote as $B \leftarrow \rho$ the binary sequence obtained from B, by left-rotating it ρ times. By definition, $B \leftarrow |B| \equiv B$. Furthermore, if b_i is the element of B on position i, we will denote as $b_i^{\leftarrow \rho}$ the element of $B \leftarrow \rho$ on position i.

Theorem 4.3.1. Given a binary sequence $B = b_0 b_1 \cdots b_{n-1}$ with length n, the following property holds:

$$\hat{C}_i(B \leftarrow 1) - \hat{C}_i(B) = b_0 \left(b_{i+1} - b_{n-i-1}\right)$$

Theorem 4.3.2. Given a binary sequence $B = b_0 b_1 \cdots b_{n-1}$ with length n, the difference $\hat{C}_i(B \leftarrow \rho) - \hat{C}_i(B \leftarrow (\rho - 1))$ is equal to $b_{(\rho-1) \bmod n}(b_{(i+\rho) \bmod n} - b_{(n-i+\rho-2) \bmod n})$.

Let us denote as Ω_B the array of all the sidelobes of a some binary sequence B with length n, or more formally: $\Omega_B = \left[\hat{C}_0(B), \hat{C}_1(B), \cdots, \hat{C}_{n-2}(B)\right]$. By using Theorem 4.3.2 and the inherited relationship between elements of $\Omega_{B \leftarrow \rho}$ and $\Omega_{B \leftarrow (\rho-1)}$, we can calculate $\Omega_{B \leftarrow \rho}$, given $\Omega_{B \leftarrow (\rho-1)}$, by using $n-1$ distinct parallel threads. Two very beneficial properties should be emphasized:

- The threads are independent of each other.
- The pool of the threads is perfectly balanced in terms of synchronization, i.e. if we have two distinct threads t_i and t_j, the arithmetic operations involved throughout the calculation process of t_i and t_j are the same.

This scenario suits well in the context of the single instruction, multiple data (SIMD) model [56]. We could dedicate the calculation of $\hat{C}_i(B \leftarrow \rho)$ to a thread t_i only since the aforementioned calculation is independent of other threads' results. Moreover, to optimize the routine further, we could just in-memory replace the values of $\hat{C}_i(B)$, i.e. Ω_B, with the consequent values of $\hat{C}_i(B \leftarrow \rho)$, i.e. $\Omega_{B \leftarrow \rho}$, for $\rho \in [1, n-1]$.

The observations made in the previous section allow us to design a fast routine for finding the minimum PSL among all the possible rotations of a given binary sequence. Our first practical application was an exhaustive search of all m-sequences with fixed lengths.

Following the same approach, the proposed algorithm could be also successfully utilized in finding optimal Legendre sequences.

We have implemented the m-sequence exhaustive search algorithm by using an amalgam of programming languages[2] and GPUs as SIMD-capable devices.

To analyze the efficiency of our implementation, we have further compared it to the popular scientific computing library NumPy [115].

During the comparison, a mid-range GPU with approximately 1200 CUDA cores and a mid-range CPU with 6 cores (12 threads) were used. For example, by using a single mid-range GPU, altogether with the aforementioned algorithm, the time required to find the PSL-optimal binary sequence, among the set comprised of a binary sequence B of length $2^{20} - 1$ and all the possible rotations of B, would be 191 seconds. For completing the same calculation on a mid-range CPU, and by using a single thread, the required time would be approximately 36 years. This results in an approximate speed-up factor of $2^{22.5}$.

The proposed algorithm allowed us to successfully exhaust search all possible m-sequences with lengths $2^{18} - 1$, $2^{19} - 1$ and $2^{20} - 1$. We were aslo able to successfully reveal all the optimal PSL values for Legendre sequences up to length 432100. The numerical experiments suggest that all Legendre sequences, with or without rotation, and with lengths $n > 235723$, could not reach a PSL value less or equal to $\sqrt{n}$.

[2] C, Python, SageMath, CUDA

Chapter 5

Binary Sequences and the Merit Factor Problem

The merit factor problem is of practical importance to manifold domains, such as digital communications engineering, radars, system modulation, system testing, information theory, physics, and chemistry. However, the merit factor problem is referenced as one of the most difficult optimization problems and it was further conjectured that stochastic search procedures will not yield merit factors higher than 5 for long binary sequences (sequences with lengths greater than 200). Some useful mathematical properties related to the flip operation of the skew-symmetric binary sequences are presented in this chapter. By exploiting those properties, the memory requirement of state-of-the-art stochastic merit factor optimization algorithms could be reduced from $O(n^2)$ to $O(n)$. As a proof of concept, a lightweight stochastic algorithm was constructed, which can optimize pseudo-randomly generated skew-symmetric binary sequences with long lengths (up to $10^5 + 1$) to skew-symmetric binary sequences with a merit factor greater than 5. An approximation of the required time is also provided. The numerical experiments suggest that the algorithm is universal and could be applied to skew-symmetric binary sequences with arbitrary lengths.

5.1 On the Skew-Symmetric Binary Sequences and the Merit Factor Problem

If F_n denotes the optimal (greatest) value of the merit factor among all sequences of length n, then the merit factor problem could be described as finding the value of $\limsup_{n\to\infty} F_n$. Several conjectures regarding the $\limsup_{n\to\infty} F_n$ value should be mentioned. The first conjecture published in [75] assumes that $\limsup_{n\to\infty} F_n = 6$. A more extreme conjec-

ture that $\limsup_{n\to\infty} F_n = \infty$ is given by Littlewood [97]. In [28], it was conjectured that $\limsup_{n\to\infty} F_n = 5$. Golay [63] assumed that the expected value of $\limsup_{n\to\infty} F_n$ is very close to 12.32. However, in [64] he added that "...no systematic synthesis will ever be found which will yield higher merit factors [than 6]...". Nevertheless, in [22] it was conjectured that $\limsup_{n\to\infty} F_n > 6.34$. The latest assumption is based on the specially constructed infinite family of sequences.

Since the merit factor problem has resisted more than 50 years of theoretical attacks, a significant number of computational pieces of evidence were collected. They could be divided into exhaustive search methods and heuristic methods.

Regarding the exhaustive search methods, the optimal merit factors for all binary sequences with lengths $n \leq 60$ are given in [105]. Twenty years later, the list of optimal merit factors was extended to $n \leq 66$ [118]. The two largest known values of F_n are 14.1 and 12.1 for n equals respectively 13 and 11. It should be mentioned that both of those binary sequences are comprised of the Barker sequences [9]. In fact, in [80] the author published a personal selection of challenges concerning the merit factor problem, arranged in order of increasing significance. The first suggested challenge is to find a binary sequence X of length $n > 13$ for which $F(X) \geq 10$.

A reasonable strategy for finding binary sequences with near-optimal merit factors is to introduce some restriction on the sequences' structure. A well-studied restriction on the structure of the sequence has been defined by the skew-symmetric binary sequences, which were introduced by Golay [60]. Having a binary sequence $(b_0, b_1, \cdots, b_{2l})$ of odd length $n = 2l + 1$, the restriction is defined by $b_{l+i} = (-1)^i b_{l-i}$ for $i = 1, 2, \cdots, l$.

Golay observed that odd-length Barker sequences are skew-symmetric. Therefore, an idea of binary sequences' sieving was proposed [62]. Furthermore, as shown in [60], all aperiodic autocorrelations of a skew-symmetric sequence with even indexes are equal to 0.

The optimal merit factors for all skew-symmetric sequences of odd length $n \leq 59$ were given by Golay himself [62]. Later, the optimal merit factors for skew-symmetric sequences with lengths $n \leq 69$ and $n \leq 71$ were revealed respectively in [65] and [41], while the optimal skew-symmetric solutions for $n \leq 89$ and $n \leq 119$ were given in respectively [125] and [118].

It should be noted, that the problem of minimizing F_n is also known as the "low autocorrelated binary string problem", or the LABS problem. It has been well studied in theoretical physics and chemistry. For example, the LABS problem is correlated with the quantum models of magnetism. Having this in mind, the merit factor problem was attacked by various search algorithms, such as the branch and bound algorithm proposed in [118], as well as stochastic search algorithms like tabu search [72], memetic algorithm combined with tabu search [57], as well as evolutionary and genetic algorithms [41][106]. However, since the

search space grows like 2^n, the difficulty of finding long binary sequences with near-optimal F_n significantly increases. Bernasconi predicted that [14] " ... stochastic search procedures will not yield merit factors higher than about $F_n = 5$ for long sequences". By long sequences, Bernasconi was referring to binary sequences with lengths greater than 200. Furthermore, in [41] the problem was described as " ... amongst the most difficult optimization problems".

The principle behind basic search methods could be summarized as moving through the search space by doing tiny changes inside the current binary sequence. In the case of skew-symmetric binary sequences, Golay suggested [61] that only one or two elements should be changed at a given optimization step. In case the new candidate has a better merit factor, the search method accepts it as a new current state and continues the optimization process. Having this in mind, a strategy of how to choose a new sequence when no acceptable neighbor sequence exists should be considered as well.

The best results regarding skew-symmetric binary sequences with high merit factors are achieved by [57][24][26][27]. In [57], the authors introduced a memetic algorithm with an efficient method to recompute the characteristics of a given binary sequence L', such that L' is one flip away from L, and assuming that some products of elements from L have been already stored in memory. More precisely, a square $(n-1, n-1)$ tau table $\tau(S)$, such that $\tau(S)_{ij} = s_j s_{i+j}$ for $j \leq n-i$ was introduced. Later, in [24] the principle of self-avoiding walk [100] was considered. By using Hasse graphs the authors demonstrated that considering the LABS problem, a basic stochastic search method could be easily trapped in a cycle. To avoid this scenario, the authors suggested the usage of a self-avoiding walk strategy accompanied by a hash table for efficient memory storage of the pivot coordinates. Then, in [26] an algorithm called xLastovka was presented. The concept of a priority queue was introduced. In summary, during the optimization process, a queue of pivot coordinates altogether with their energy values is maintained. Recently, some skew-symmetric binary sequences with record-breaking merit factors for lengths from 301 to 401 were revealed [27].

The aforementioned state-of-the-art algorithms are benefiting from the tau table $\tau(S)$ previously discussed. It significantly increases the speed of evaluating a given one-flip-away neighbor, reaching a time complexity of $O(n)$. However, the memory requirement of maintaining $\tau(S)$ is $O(n^2)$. Having this in mind, the state-of-the-art algorithms could be inapplicable to very long binary sequences due to hardware restrictions.

In this section, by using some mathematical insights, an alternative to the $\tau(S)$ table is suggested, the usage of which significantly reduces the memory requirement of the discussed state-of-the-art algorithms from $O(n^2)$ to $O(n)$. This enhancement could be easily integrated. For example, in an online repository [23] a collection of currently known best merit factors for skew-symmetric sequences with lengths from 5 to 449 is given. The longest binary

sequence is of length 449, having a merit factor of 6.5218. As a proof of concept, by using just a single budget processor Xeon-2640 CPU with a base frequency of 2.50 GHz, the price of which at the time of writing this work is about 15 dollars, and our tweaked implementation of the lssOrel algorithm introduced in [23], we were able to find a skew-symmetric binary sequence with better merit factor of 6.5319. The time required was approximately one day. As a comparison, the currently known optimal results were acquired by using the Slovenian Initiative for National Grid (SLING) infrastructure (100 processors) and a 4-day threshold limitation per length.

It should be noted, that despite the significant memory complexity optimization introduced, the state-of-the-art algorithms could still suffer from memory and speed issues. As previously discussed, additional memory-requiring structures were needed, such as, for example, a set of all previously visited pivots [24] or a priority queue with 640 000 coordinates and a total size of 512MB [26].

Another issue is the "greedy" approach of collecting all the neighbors to determine the best one. This could dramatically decrease the optimization process, especially when very long binary sequences are involved. This side-effect was already discussed in Section 4.2.

Having those observations in mind, an almost memory-free optimization algorithm is suggested. More precisely, both the time and memory complexities of the algorithm are linear. This could be particularly beneficial for multi-thread architectures or graphical processing units. During our experiments, and by using the aforementioned algorithm, we were able to find skew-symmetric sequences with merit factors strictly greater than $F_n = 5$ for all the tested lengths up to $10^5 + 1$. Thus, Bernasconi's prediction that no stochastic search procedure will yield merit factors higher than $F_n = 5$ for binary sequences with lengths greater than 200 was very pessimistic.

Let us consider a skew-symmetric binary sequence defined by an array $L = [b_0, b_1, \cdots, b_{n-1}]$ with an odd length $n = 2l + 1$. If the corresponding to L sidelobes' array is denoted by an array W, we have:

$$W = [C_{n-1}(L), C_{n-2}(L), \cdots, C_1(L), C_0(L)],$$

where

$$C_u(L) = \sum_{j=0}^{n-u-1} b_j b_{j+u}, \quad for\ u \in \{0, 1, \cdots, n-1\}.$$

In this section, for convenience, we will use the reversed version of W, denoted by S, s.t:

$$S = \left[\hat{C}_0(L), \hat{C}_1(L), \cdots, \hat{C}_{n-2}(L), \hat{C}_{n-1}(L)\right],$$

where $\hat{C}_{n-i-1}(L) = C_i(L)$, for $i \in \{0, 1, \cdots, n-1\}$. Thus,

$$\hat{C}_i(L) = C_{n-i-1}(L) = \sum_{j=0}^{n-(n-i-1)-1} b_j b_{j+(n-i-1)}.$$

Hence,

$$\hat{C}_i(L) = \sum_{j=0}^{i} b_j b_{j+n-i-1}, \quad for\ i \in \{0, 1, \cdots, n-1\}.$$

Furthermore, we will denote the i-th element of a given array A as $A[i]$. It should be noted that the first index of an array is 0, not 1. For example,

$$W[n-1] = S[0] = \hat{C}_0[L] = C_{n-1}(L).$$

Since L is a skew-symmetric binary sequence, the following properties hold:

- $S[i] = 0$, for odd values of i.
- $L[l-i] = (-1)^i L[l+i]$.

Having this in mind, the array of sidelobes S could be represented as follows:

$$S = \left[\hat{C}_0(L), 0, \hat{C}_2(L), 0, \cdots, 0, \hat{C}_{n-3}(L), 0, \hat{C}_{n-1}(L)\right].$$

For convenience, we will use the notation S_i which represents the $(i-1)$-th element of a given array S, or more formally $S_i = S[i-1]$.

Thus, for every odd value r, we have

$$S_r = \hat{C}_{r-1}(L) = \sum_{j=0}^{r-1} b_j b_{j+n-r+1-1} = \sum_{j=0}^{r-1} b_j b_{j+n-r} = \sum_{j=1}^{r} b_{j-1} b_{j-1+n-r}.$$

In terms of L, the previous relationship could be written down as follows:

$$S_r = \sum_{j=1}^{r} b_{j-1} b_{j-1+n-r} = \sum_{i=1}^{r} L[i-1] L[n+i-r-1].$$

Given a skew-symmetric sequence L with length $n = 2l+1$, if we flip both the elements on positions q and $n-q-1$, for some fixed $q \in \{0, 1, \cdots, l\}$, the resulted binary sequence L^q will be skew-symmetric as well. Let's denote the array of sidelobes of L^q as S^q:

$$S^q = \left[\hat{C}_0(L^q), 0, \hat{C}_2(L^q), 0, \cdots, 0, \hat{C}_{n-3}(L^q), 0, \hat{C}_{n-1}(L^q)\right].$$

Theorem 5.1.1. Given two skew-symmetric sequences L and L^q with length $n = 2l + 1$, and with sidelobes arrays respectively S and S^q, where $q < l$, the following properties hold:

I For $\forall e$, s.t. e is an even number, $S^q_e - S_e = 0$.

II If r is an odd number and $r \leq q$, $S^q_r - S_r = 0$.

III If r is an odd number and $r > q$, and $r < n - q$, and $q \neq r - q - 1$, then:

$$S^q_r - S_r = -2\left(L[q]L[n + q - r] + L[r - q - 1]L[n - q - 1]\right).$$

IV If r is an odd number and $r > q$, and $r < n - q$, and $q = r - q - 1$, then $S^q_r - S_r = 0$.

V If r is an odd number and $r \geq n - q$, and $q \neq r - q - 1$, then:

$$S^q_r - S_r = -2L[n - q - 1]L[2n - q - r - 1] - 2L[q + r - n]L[q] -$$

$$-2L[q]L[n + q - r] - 2L[r - q - 1]L[n - q - 1].$$

VI If r is an odd number and $r \geq n - q$, and $q = r - q - 1$, then:

$$S^q_r - S_r = -2L[n - q - 1]L[2n - q - r - 1] - 2L[q + r - n]L[q].$$

We should emphasize, that Theorem 5.1.1 covers all the possible sidelobes positions and all the possible flip bit choices.

Theorem 5.1.2. Given two skew-symmetric sequences L and L^q with length $n = 2l + 1$, where L^q corresponds to L with q-th and $n - q - 1$-th bit flipped for some fixed $q < l$, and with sidelobes arrays denoted respectively as S and S^q, the following property holds:

$$\mathbb{E}(L^q) = \mathbb{E}(L) + \sum_{r=q+1, r\neq 2q+1}^{n-q-1} (16 + \sigma\kappa\varepsilon_1) + \sum_{r=n-q, r\neq 2q+1}^{n-1} (\kappa(\varepsilon_2 + \sigma\varepsilon_1) + 32 + 32\sigma\varepsilon_1\varepsilon_2) +$$
$$+ \sum_{r\geq n-q, r\leq n-1, r=2q+1} (16 + \kappa\varepsilon_2), \tag{5.1}$$

where $\sigma = (-1)^{l-q}$, $\kappa = -8S_rL[q]$, $\varepsilon_1(r) = L[r - q - 1]$, $\varepsilon_2(r) = L[q + r - n]$.

The last property allows us to reduce the memory requirement of some state-of-the-art algorithms from $O(n^2)$ to $O(n)$. For example, by using just one thread of the processors, the

tau table corresponding to binary sequences with length 5000 would require approximately 95.37 Megabytes to be allocated for the tau table expansion routine, while the sidelobe array presented in this work would require the allocation of approximately 19.53 Kilobytes. It should be emphasized, that interchanging the tau table used by the state-of-the-art algorithms with the proposed sidelobe array structure would not impact the time complexity of the tweaked algorithm. However, from a practical point of view, the significant memory reduction could greatly enhance the overall time performance of a tweaked algorithm, since the size of the sidelobe array could be usually saved inside the CPU cache layers, instead of saving it to the slower memory banks. Furthermore, interchanging the tau table with the proposed sidelobe array could allow the multithreading capabilities of modern CPUs, and even GPUs, to be fully utilized.

The algorithm was implemented (C++) on a general-purpose computer equipped with a budget processor Xeon-2640 CPU, having a base frequency of 2.50 GHz. A skew-symmetric binary sequence with length 449 and a record-breaking merit factor of 6.5319 was found after approximately one day. It should be noted that all 12 threads of the CPU were launched in parallel. As a comparison, the currently known optimal results (a merit factor of 6.5218) were acquired by using the Slovenian Initiative for National Grid (SLING) infrastructure (100 processors) and 4-day threshold limitation [23].

5.1.1 On the Bernasconi Conjecture

As previously discussed, in [14] Bernasconi conjectured that stochastic search procedures will not yield merit factors higher than 5 for long sequences (greater than 200). It should be mentioned that this prediction was made in 1987. Since then, many years have passed and pieces of evidence that stochastic search procedures could perform better than the prediction's expectations were found. Indeed, heuristic algorithms that could find odd binary sequences with lengths up to about 500 and merit factors greater than 5 were discovered. However, the Bernasconi conjecture appears valid when the threshold of the binary sequence's length is updated and lifted. Since during the last 35 years the computational capabilities of modern CPUs are rising almost exponentially such actualization would be fair. However, if a stochastic search procedure is found, a procedure that could reach extremely long binary sequences with merit factors greater than 5, by using a mid-range general-purpose computer, then the barriers predicted by Bernasconi could be very pessimistic.

During our experiments, by using a modification of the algorithm discussed in the previous section, we were able to reach skew-symmetric binary sequences with lengths up to 100 001 and merit factors greater than 5.

5.1.2 New Classes of Binary Sequences with High Merit Factor

Despite the rich results regarding the skew-symmetric binary sequences, the search for binary sequences with even lengths and high MF was scarcely researched. This is not surprising, since the sieving proposed by Golay applies to odd-length sequences only.

In this section, motivated by the absence of computationally efficient sieving for binary sequences with even lengths and high merit factor values, several new classes of binary sequences are proposed. We start with the definition of a class of finite binary sequences, called pseudo-skew-symmetric, with alternate auto-correlation absolute values equal to one. The class is defined by using sieving suitable for even-length binary sequences. Then, by using some mathematical observations, we show how state-of-the-art algorithms for searching skew-symmetric binary sequences with high merit factor and length $2n+1$ could be easily converted to algorithms searching pseudo-skew-symmetric binary sequences with high merit factor and lengths $2n$ or $2n+2$. More importantly, this conversion does not degrade the performance of the modified algorithm.

Then, by using number partitions [6], an additional sieving strategy for both skew-symmetric and pseudo-skew-symmetric sequences is proposed. A method of finding subclasses of binary sequences with high MF is further discussed. The experiments revealed that the classes defined in this section are highly promising. By using a single mid-range computer, we were able to improve all records for skew-symmetric binary sequences with lengths above 225, which were recently reached by various algorithms and a supercomputer grid. We further revealed that binary sequences with even or odd length n, for $n \leq 2^8$, and with merit factor strictly greater than 8, and binary sequences with even or odd length n, for $n \leq 2^9$ and with a merit factor strictly greater than 7 do exist.

Definition 5.1.1 (Pseudo-Skew-Symmetric Binary Sequence)**.** We call a given sequence $P = a||X = Y||b$ a pseudo-skew-symmetric binary sequence, if either X or Y are skew-symmetric binary sequences, for some $a \in \{-1, 1\}$ or $b \in \{-1, 1\}$.

Proposition 5.1.1. The sidelobes array of pseudo-skew-symmetric binary sequences consists of alternating $\pm$ ones.

Proposition 5.1.2. Given a skew-symmetric binary sequence $B = (b_0, b_1, \cdots, b_{n-1})$ with sidelobes array

$$S_B = \left[\hat{C}_0(B), \hat{C}_1(B), \cdots, \hat{C}_{n-2}(B), \hat{C}_{n-1}(B)\right],$$

the following property holds:

$$\mathbb{E}(P) = \mathbb{E}(B) + n + 2b_n\delta,$$

where P is the pseudo-skew-symmetric sequence $B||b_n$ and $\delta = \sum_{u=0,u_{even}}^{n-2} \hat{C}_u(B)b_{u+1}$.

The last property is of significant importance when converting an algorithm searching for skew-symmetric binary sequences, denoted as $\mathscr{A}$, to an algorithm searching for pseudo-skew-symmetric binary sequences $\mathscr{B}$ and a high merit factor. Indeed, despite the complexity of algorithm $\mathscr{A}$ we can decompose it to a tape $\cdots||\mathbb{L}_1||\cdots||\mathbb{L}_2||\cdots||\mathbb{L}_n||\cdots$, where $\mathbb{L}_i$ are stages of $\mathscr{A}$, where better candidates could be announced. They are known as local optimums in heuristic search literature. We could easily replace $\mathbb{L}_i$ with $\mathbb{L}_i||\mathbb{T}_i$, where $\mathbb{T}_i$ is a simple routine with memory and time complexity of $O(n)$, which calculates the pseudo-skew-symmetric sequences $L_i||1$ and $L_i||-1$ merit factors, where L_i is the current best candidate. It should be noted that $\mathscr{B} = \cdots||\mathbb{L}_1||\mathbb{T}_1||\cdots||\mathbb{L}_2||\mathbb{T}_2||\cdots||\mathbb{L}_n||\mathbb{T}_n||\cdots$ does not interfere with the normal work of $\mathscr{A}$ by design. Furthermore, since those linear time complexity checkups are initiated on local optimums only, the delay of $\mathscr{B}$ compared to $\mathscr{A}$ caused by the additional instructions $\mathbb{T}_i$ is negligible.

We could further extend the search of highly-competitive pseudo-skew-symmetric sequences by the following observations:

Proposition 5.1.3. Given a skew-symmetric binary sequence $B = b_0||B'||b_{n-1}$ both binary sequences $b_0||B'$ and $B'||b_{n-1}$ are pseudo-skew-symmetric.

Proposition 5.1.4. Given a skew-symmetric binary sequence $B = (b_0, b_1, \cdots, b_{n-1}) = b_0||B'||b_{n-1}$ with sidelobes array

$$S_B = \left[\hat{C}_0(B), \hat{C}_1(B), \cdots, \hat{C}_{n-2}(B), \hat{C}_{n-1}(B)\right],$$

the following property holds:

$$\mathbb{E}(P) = \mathbb{E}(B) + n - 3 + 2b_{n-1}\delta,$$

where P is the pseudo-skew-symmetric sequence $b_0||B'$ and $\delta = \sum_{u=1,u_{even}}^{n-2} -\hat{C}_u(B)b_u$.

The last property further enhances the power of the algorithm. Thus now we can modify each algorithm $\mathscr{A}$, searching for skew-symmetric binary sequences with odd length n and high merit factor, to an algorithm $\mathscr{B}$, searching simultaneously skew-symmetric binary sequences with odd length n and pseudo-skew-symmetric binary sequences with even lengths $n-1$ and $n+1$.

Definition 5.1.2 (Restriction Class of Binary Sequence)**.** We will call the class of binary sequences of length n, with the first k elements fixed, a restriction class of order k on

binary sequences with length n. We will denote this set as R_n^k. If the binary sequence is skew-symmetric we will use the notation $\mathscr{R}_n^k$.

A well-studied area in number theory and combinatorics is the number partition problem - distinct ways of writing a given integer number n as a sum of positive integers. We define the number of possible partitions of a non-negative integer n as the partition function $p(n)$. No closed-form expression for $p(n)$ is known. However, the partition functions for some different values of n could be found in the online encyclopedia of integer numbers (OEIS), sequence A000041 [1].

Theoretically, searching for skew-symmetric binary sequences of length n with high merit factors could be parallelized to $|\mathscr{R}_n^k|$ instances. To minimize the total number of instances needed, we should consider several actions to a given skew-symmetric binary sequence $B = (b_0, b_1, \cdots, b_{n-1})$:

- Reversing B defined as operator δ_1: $\delta_1(B) = (b_{n-1}, \cdots, b_1, b_0)$
- Complementing B defined as operator δ_2: $\delta_2(B) = (\overline{b_0}, \overline{b_1}, \cdots, \overline{b_{n-1}})$, where $\overline{b_i} = -b_i$
- Alt. complementing of B defined as operator δ_3 : $\delta_3(B) =$
  $= (\cdots, \overline{b_{i-2}}, b_{i-1}, \overline{b_i}, b_{i+1}, \overline{b_{i+2}}, \cdots)$

All three operators leave the energy of B intact. If we further add the identity operator δ_0 we construct a group G of order 8. By using some group theory [118], we could derive a closed formula of the exact number of symmetry classes with length k: $2^{k-3} + 2^{\lfloor \frac{k}{2} \rfloor - 2 + (k \bmod 2)}$. The same formula arises from the row sums of the Losanitsch's triangle (OEIS, sequence A005418 [2]) - named after the S. Lozanić, in his work related to the symmetries exhibited by rows of paraffins [99]. This fact could be used to partition the search space from $p(k)$ covering subsets to $2^{k-3} + 2^{\lfloor \frac{k}{2} \rfloor - 2 + (k \bmod 2)}$ non-covering subsets.

Definition 5.1.3 (Potential of a Restriction Subclass)**.** For a skew-symmetric binary sequence $B = (b_0, b_1, \cdots, b_{n-1})$, we fix a partitioning with length k: $t_0, t_1, \cdots, t_g$, s.t. $\sum_{i=0}^{g} t_i = k$. The partitioning could be projected to a skew-symmetric binary sequence with the following procedure:

$$R = \underbrace{a\cdots a}_{t_0}\underbrace{\overline{a}\cdots\overline{a}}_{t_1}\underbrace{a\cdots a}_{t_2}\underbrace{\overline{a}\cdots\overline{a}}_{t_3}\cdots\underbrace{(-1)^g a\cdots(-1)^g a}_{t_g}\underbrace{u_1u_2u_3\cdots u_{n-2k-2}u_{n-2k-1}u_{n-2k}}_{\text{non-fixed (free) elements}}\underbrace{f_1f_2f_3\cdots f_{k-2}f_{k-1}f_k}_{\text{last elements are fixed}}$$

The last k elements f_i are fixed due to the first k elements of the sequence and its skew-symmetric property. Please note that all elements $a, \overline{a}, (-1)^g a, u_i, f_i \in \{-1, 1\}$. We define

the potential of the binary skew-symmetric sequence R as the energy of the ternary sequence R^z, where:

$$R^z = \underbrace{a\cdots a}_{t_0}\underbrace{\bar{a}\cdots\bar{a}}_{t_1}\underbrace{a\cdots a}_{t_2}\underbrace{\bar{a}\cdots\bar{a}}_{t_3}\cdots\underbrace{(-1)^g a\cdots(-1)^g a}_{t_g}\ \underbrace{000\cdots000}_{n-2k\text{ zeroed elements}}\ \underbrace{f_1 f_2 f_3\cdots f_{k-2} f_{k-1} f_k}_{\text{last elements are fixed}}$$

5.1.3 Algorithm for Finding Binary Sequences with Arbitrary Length and High Merit Factor

The algorithm was implemented (C++) on a general-purpose computer equipped with a central processing unit with 8 cores and 16 threads. Despite using just a single low-budget personal computer, we were able to improve all the results, for all skew-symmetric lengths in the range 225-451, announced in literature and reached by using a supercomputer grid. Furthermore, by using classes of pseudo-skew-symmetric sequences, we were able to simultaneously reach binary sequences of even lengths between 225 and 512, and beyond, with merit factors greater than 7. We demonstrate the efficiency of our approach by publishing a complete list of binary sequences, for both even and odd lengths up to 2^8, and merit factors greater than 8. The list is further accompanied by a complete list of binary sequences, for both even and odd lengths up to 2^9, and merit factors greater than 7.

We further demonstrate the power and efficiency of the proposed algorithm by launching it on binary sequences of lengths 573 and 1009. As mentioned earlier, it was estimated that finding solutions with a merit factor of 6.34 for a binary sequence with length 573 requires around 32 years, while for binary sequences with length 1009, the average runtime prediction to reach the merit factor of 6.34 is 46774481153 years. By using the proposed algorithm, we were able to reach such candidates within several hours.

5.2 Using Aperiodic Autocorrelation functions for an S-box reverse engineering

We can treat all $\binom{n}{2}$ columns of two-term linear combinations of coordinates of an S-box $S(n,n)$ as binary sequences and analyze their sidelobe levels. Such a strategy makes sense since sidelobe levels can reveal hidden inner relationships between the coordinates of S.

Anomalies in S-boxes of BelT, CSS, Safer, and SKINNY are discovered.

References

[1] Oeis a000041. https://oeis.org/A000041. Accessed: 2022-05-30.

[2] Oeis a005418. https://oeis.org/A005418. Accessed: 2022-05-30.

[3] Adomnicai, A., Berger, T. P., Clavier, C., Francq, J., Huynh, P., Lallemand, V., Le Gouguec, K., Minier, M., Reynaud, L., and Thomas, G. (2019). Lilliput-ae: a new lightweight tweakable block cipher for authenticated encryption with associated data. *Submitted to NIST Lightweight Project*.

[4] Ahmad, M., Bhatia, D., and Hassan, Y. (2015). A novel ant colony optimization based scheme for substitution box design. *Procedia Computer Science*, 57:572–580.

[5] Andreeva, E., Lallemand, V., Purnal, A., Reyhanitabar, R., Roy, A., and Vizár, D. (2019). Forkae v. *Submission to NIST lightweight cryptography project*.

[6] Andrews, G. E. (1998). *The theory of partitions*. Number 2. Cambridge university press.

[7] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T. (2000). Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis. In *International Workshop on Selected Areas in Cryptography*, pages 39–56. Springer.

[8] Baden, J. and Cohen, M. (1990). Optimal peak sidelobe filters for biphase pulse compression. In *IEEE International Conference on Radar*, pages 249–252. IEEE.

[9] Barker, R. H. and Jackson, W. (1953). Group synchronization of binary digital systems in Communication Theory. *Academic Press, New York*, pages 273–287.

[10] Barreto, P. and Rijmen, V. (2000). The khazad legacy-level block cipher. *Primitive submitted to NESSIE*, 97:106.

[11] Barreto, P., Rijmen, V., et al. (2000a). The Whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*, volume 13, page 14. Citeseer.

[12] Barreto, P., Rijmen, V., et al. (2000b). The whirlpool hashing function. In *First open NESSIE Workshop, Leuven, Belgium*, volume 13, page 14. Citeseer.

[13] Becker, M. and Desoky, A. (2004). A study of the dvd content scrambling system (css) algorithm. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.*, pages 353–356. IEEE.

[14] Bernasconi, J. (1987). Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *Journal de Physique*, 48(4):559–567.

[15] Berry, D. A. and Fristedt, B. (1985). Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). *London: Chapman and Hall*, 5:71–87.

[16] Bhattacharya, D., Bansal, N., Banerjee, A., and RoyChowdhury, D. (2007). A near optimal S-box design. In *International Conference on Information Systems Security*, pages 77–90. Springer.

[17] Biham, E. (1994). On Matsui's linear cryptanalysis. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 341–355. Springer.

[18] Biham, E. and Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY*, 4(1):3–72.

[19] Bikov, D., Bouyukliev, I., and Bouyuklieva, S. (2019). Bijective S-boxes of different sizes obtained from quasi-cyclic codes. *Journal of Algebra Combinatorics Discrete Structures and Applications*, 6(3):123–134.

[20] Biryukov, A. and Perrin, L. (2015). On reverse-engineering s-boxes with hidden design criteria or structure. In *Annual Cryptology Conference*, pages 116–140. Springer.

[21] Biryukov, A., Perrin, L., and Udovenko, A. (2016). Reverse-engineering the s-box of streebog, kuznyechik and stribobr1. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 372–402. Springer.

[22] Borwein, P., Choi, K.-K., and Jedwab, J. (2004). Binary sequences with merit factor greater than 6.34. *IEEE transactions on information theory*, 50(12):3234–3249.

[23] Boškovic, B., Brglez, F., and Brest, J. (2016). A github archive for solvers and solutions of the labs problem. *For updates, see https://github. com/borkob/git_labs (January 2016)*.

[24] Bošković, B., Brglez, F., and Brest, J. (2017). Low-autocorrelation binary sequences: On improved merit factors and runtime predictions to achieve them. *Applied Soft Computing*, 56:262–285.

[25] Bouyukliev, I., Bikov, D., and Bouyuklieva, S. (2017). S-boxes from binary quasi-cyclic codes. *Electronic Notes in Discrete Mathematics*, 57:67–72.

[26] Brest, J. and Bošković, B. (2018). A heuristic algorithm for a low autocorrelation binary sequence problem with odd length and high merit factor. *IEEE Access*, 6:4127–4134.

[27] Brest, J. and Bošković, B. (2020). In searching of long skew-symmetric binary sequences with high merit factors. *arXiv preprint arXiv:2011.00068*.

[28] Byrnes, J. and Newman, D. J. (1990). The l4 norm of a polynomial with coefficients$\pm$1. *Amer. Math. Monthly*, 97:42–45.

[29] Canteaut, A., Duval, S., and Leurent, G. (2015a). Construction of lightweight S-boxes using Feistel and MISTY structures. In *International Conference on Selected Areas in Cryptography*, pages 373–393. Springer.

[30] Canteaut, A., Duval, S., and Leurent, G. (2015b). Construction of lightweight s-boxes using feistel and misty structures. In *International Conference on Selected Areas in Cryptography*, pages 373–393. Springer.

[31] Chen, G. (2008). A novel heuristic method for obtaining S-boxes. *Chaos, Solitons & Fractals*, 36(4):1028–1036.

[32] Clark, J. A., Jacob, J. L., and Stepney, S. (2005). The design of S-boxes by simulated annealing. *New Generation Computing*, 23(3):219–231.

[33] Coppersmith, D. (1994). The data encryption standard (des) and its strength against attacks. *IBM journal of research and development*, 38(3):243–250.

[34] Courtois, N. T. and Pieprzyk, J. (2002). Cryptanalysis of block ciphers with overdefined systems of equations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 267–287. Springer.

[35] Coxson, G. and Russo, J. (2005). Efficient exhaustive search for optimal-peak-sidelobe binary codes. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):302–308.

[36] Coxson, G. E., Hill, C. R., and Russo, J. C. (2014). Adiabatic quantum computing for finding low-peak-sidelobe codes. In *2014 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE.

[37] Coxson, G. E., Russo, J. C., and Luther, A. (2020). Long low-psl binary codes by multi-thread evolutionary search. In *2020 IEEE International Radar Conference (RADAR)*, pages 256–261. IEEE.

[38] Cui, L. and Cao, Y. (2007). A new S-box structure named Affine-Power-Affine. *International Journal of Innovative Computing, Information and Control*, 3(3):751–759.

[39] Daemen, J. and Rijmen, V. (2013a). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

[40] Daemen, J. and Rijmen, V. (2013b). *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media.

[41] De Groot, C., Würtz, D., and Hoffmann, K. H. (1992). Low autocorrelation binary sequences: Exact enumeration and optimization by evolutionary strategies. *Optimization*, 23(4):369–384.

[42] de la Cruz Jiménez, R. A. (2017). Generation of 8-Bit S-Boxes Having Almost Optimal Cryptographic Properties Using Smaller 4-Bit S-Boxes and Finite Field Multiplication. In *International Conference on Cryptology and Information Security in Latin America*, pages 191–206. Springer.

[43] Developers, T. S. (2016). Sagemath.

[44] Dimitrov, M. (2020a). On the aperiodic autocorrelations of rotated binary sequences. *IEEE Communications Letters*, 25(5):1427–1430.

[45] Dimitrov, M. (2021a). On the skew-symmetric binary sequences and the merit factor problem. *arXiv preprint arXiv:2106.03377*.

[46] Dimitrov, M. (2022). New classes of binary sequences with high merit factor. *arXiv preprint arXiv:2206.12070*.

[47] Dimitrov, M., Baicheva, T., and Nikolov, N. (2021). Hybrid constructions of binary sequences with low autocorrelation sideobes. *IEEE Access*, 9:112400–112410.

[48] Dimitrov, M., Baitcheva, T., and Nikolov, N. (2020a). Efficient generation of low autocorrelation binary sequences. *IEEE Signal Processing Letters*, 27:341–345.

[49] Dimitrov, M., Baitcheva, T., and Nikolov, N. (2020b). On the generation of long binary sequences with record-breaking psl values. *IEEE Signal Processing Letters*, 27:1904–1908.

[50] Dimitrov, M. M. (2020b). On the design of chaos-based s-boxes. *IEEE Access*, 8:117173–117181.

[51] Dimitrov, M. M. (2021b). A framework for fine-grained nonlinearity optimization of boolean and vectorial boolean functions. *IEEE Access*, 9:124910–124920.

[52] Dmitriev, D. and Jedwab, J. (2007). Bounds on the growth rate of the peak sidelobe level of binary sequences. *Advances in Mathematics of Communications*, 1(4):461.

[53] Dolmatov, V. (2016). Gost r 34.12-2015: Block cipher "kuznyechik". *Transformation*, 50:10.

[54] Du, K. L., Wu, W. H., and Mow, W. H. (2013). Determination of long binary sequences having low autocorrelation functions. US Patent 8,493,245.

[55] FIPS, P. (1999). 46-3. data encryption standard (des). *National Institute of Standards and Technology*, 25(10):1–22.

[56] Flynn, M. J. (1972). Some computer organizations and their effectiveness. *IEEE transactions on computers*, 100(9):948–960.

[57] Gallardo, J. E., Cotta, C., and Fernández, A. J. (2009). Finding low autocorrelation binary sequences with memetic algorithms. *Applied Soft Computing*, 9(4):1252–1262.

[58] Gérard, B., Grosso, V., Naya-Plasencia, M., and Standaert, F.-X. (2013). Block ciphers that are easier to mask: How far can we go? In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 383–399. Springer.

[59] Gilbert, H. and Peyrin, T. (2010). Super-Sbox cryptanalysis: improved attacks for AES-like permutations. In *International Workshop on Fast Software Encryption*, pages 365–383. Springer.

[60] Golay, M. (1972). A class of finite binary sequences with alternate auto-correlation values equal to zero (corresp.). *IEEE Transactions on Information Theory*, 18(3):449–450.

[61] Golay, M. (1975). Hybrid low autocorrelation sequences (corresp.). *IEEE Transactions on Information Theory*, 21(4):460–462.

[62] Golay, M. (1977). Sieves for low autocorrelation binary sequences. *IEEE Transactions on information theory*, 23(1):43–51.

[63] Golay, M. (1982). The merit factor of long low autocorrelation binary sequences (corresp.). *IEEE Transactions on Information Theory*, 28(3):543–549.

[64] Golay, M. (1983). The merit factor of legendre sequences (corresp.). *IEEE Transactions on Information Theory*, 29(6):934–936.

[65] Golay, M. J. and Harris, D. B. (1990). A new search for skewsymmetric binary sequences with optimal merit factors. *IEEE Transactions on Information Theory*, 36(5):1163–1166.

[66] Gold, R. (1967). Optimal binary sequences for spread spectrum multiplexing (Corresp.). *IEEE Transactions on Information Theory*, 13(4):619–621.

[67] Golomb, S. W. et al. (1967). *Shift register sequences*. Aegean Park Press.

[68] GOST, R. (2015). R 34.12-2015. *Information Technology. Cryptographic Protection of Information. Block Ciphers]. Moscow, Standartinform*.

[69] Grosso, V., Leurent, G., Standaert, F.-X., and Varıcı, K. (2014a). LS-designs: Bitslice encryption for efficient masked software implementations. In *International Workshop on Fast Software Encryption*, pages 18–37. Springer.

[70] Grosso, V., Leurent, G., Standaert, F.-X., Varici, K., Durvaux, F., Gaspar, L., and Kerckhof, S. (2014b). Scream & iscream side-channel resistant authenticated encryption with masking. *Submission to CAESAR*.

[71] Gurobi Optimization, I. (2018). Gurobi optimizer reference manual. *URL http://www. gurobi. com*.

[72] Halim, S., Yap, R. H., and Halim, F. (2008). Engineering stochastic local search for the low autocorrelation binary sequence problem. In *International Conference on Principles and Practice of Constraint Programming*, pages 640–645. Springer.

[73] He, H., Stoica, P., and Li, J. (2009). Designing unimodular sequence sets with good correlations—including an application to mimo radar. *IEEE Transactions on Signal Processing*, 57(11):4391–4405.

[74] Heys, H. M. (2002). A tutorial on linear and differential cryptanalysis. *Cryptologia*, 26(3):189–221.

[75] Hoholdt, T. and Jensen, H. E. (1988). Determination of the merit factor of legendre sequences. *IEEE Transactions on Information Theory*, 34(1):161–164.

[76] Isa, H., Jamil, N., and Z'aba, M. R. (2013). S-box construction from non-permutation power functions. In *Proceedings of the 6th International Conference on Security of Information and Networks*, pages 46–53. ACM.

[77] Isa, H., Jamil, N., and Z'aba, M. R. (2016). Construction of cryptographically strong S-Boxes inspired by bee waggle dance. *New generation computing*, 34(3):221–238.

[78] Ivanov, G., Nikolov, N., and Nikova, S. (2015). Cryptographically strong S-boxes generated by modified immune algorithm. In *International Conference on Cryptography and Information Security in the Balkans*, pages 31–42. Springer.

[79] Jakobsen, T. and Knudsen, L. R. (1997). The interpolation attack on block ciphers. In *International Workshop on Fast Software Encryption*, pages 28–40. Springer.

[80] Jedwab, J. (2004). A survey of the merit factor problem for binary sequences. In *International Conference on Sequences and Their Applications*, pages 30–55. Springer.

[81] Jedwab, J. and Yoshida, K. (2006). The peak sidelobe level of families of binary sequences. *IEEE transactions on information theory*, 52(5):2247–2254.

[82] Junod, P. and Vaudenay, S. (2004). Fox: a new family of block ciphers. In *International Workshop on Selected Areas in Cryptography*, pages 114–129. Springer.

[83] Karpman, P. and Grégoire, B. (2016). The littlun s-box and the fly block cipher. In *Lightweight Cryptography Workshop*, pages 17–18.

[84] Kasami, T. (1966). Weight distribution formula for some class of cyclic codes. *Coordinated Science Laboratory Report no. R-285*.

[85] Kazymyrov, O., Kazymyrova, V., and Oliynykov, R. (2013). A Method For Generation Of High-Nonlinear S-Boxes Based On Gradient Descent. *IACR Cryptology ePrint Archive*, 2013:578.

[86] Kerahroodi, M. A., Aubry, A., De Maio, A., Naghsh, M. M., and Modarres-Hashemi, M. (2017). A coordinate-descent framework to design low psl/isl sequences. *IEEE Transactions on Signal Processing*, 65(22):5942–5956.

[87] Leukhin, A., Parsaev, N., Bezrodnyi, V., and Kokovihina, N. (2017). The exhaustive search for optimum minimum peak sidelobe binary sequences. *Bulletin of the Russian Academy of Sciences: Physics*, 81(5):575–578.

[88] Leukhin, A. and Potehin, E. (2012). Binary sequences with minimum peak sidelobe level up to length 68. *arXiv preprint arXiv:1212.4930*.

[89] Leukhin, A. and Potekhin, E. (2015). A Bernasconi model for constructing ground-state spin systems and optimal binary sequences. In *Journal of Physics: Conference Series*, volume 613, page 012006. IOP Publishing.

[90] Leukhin, A. N. and Potekhin, E. N. (2013). Optimal peak sidelobe level sequences up to length 74. In *2013 European Radar Conference*, pages 495–498. IEEE.

[91] Leukhin, Anatolii N and Potekhin, Egor N (2014). Exhaustive search for optimal minimum peak sidelobe binary sequences up to length 80. In *International Conference on Sequences and Their Applications*, pages 157–169. Springer.

[92] Levanon, N. and Mozeson, E. (2004). *Radar signals*. John Wiley & Sons.

[93] Lim, C. H. (1998). Crypton: A new 128-bit block cipher. *NIsT AEs Proposal*.

[94] Lim, C. H. (1999). A revised version of crypton: Crypton v1. 0. In *International Workshop on Fast Software Encryption*, pages 31–45. Springer.

[95] Lin, R., Soltanalian, M., Tang, B., and Li, J. (2019). Efficient design of binary sequences with low autocorrelation sidelobes. *IEEE Transactions on Signal Processing*, 67(24):6397–6410.

[96] Lindner, J. (1975). Binary sequences up to length 40 with best possible autocorrelation function. *Electronics letters*, 11(21):507–507.

[97] Littlewood, J. (1966). On Polynomials $\sum^n \pm z^m$, $\sum^n e^{\alpha_m i} z^m$, $z = e^{\theta i}$. *Journal of the London Mathematical Society*, 1(1):367–376.

[98] Littlewood, J. E. (1968). *Some problems in real and complex analysis*. DC Heath.

[99] Losanitsch, S. (1897). Die isomerie-arten bei den homologen der paraffin-reihe. *Berichte der deutschen chemischen Gesellschaft*, 30(2):1917–1926.

[100] Madras, N. and Slade, G. (2013). *The self-avoiding walk*. Springer Science & Business Media.

[101] Mamadolimov, A., Isa, H., and Mohamad, M. S. (2013). Practical bijective S-box design. *arXiv preprint arXiv:1301.4723*.

[102] Massey, J. L. (1993). Safer k-64: A byte-oriented block-ciphering algorithm. In *International Workshop on Fast Software Encryption*, pages 1–17. Springer.

[103] Matsui, M. (1993). Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 386–397. Springer.

[104] Meier, W. and Staffelbach, O. (1989). Nonlinearity criteria for cryptographic functions. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 549–562. Springer.

[105] Mertens, S. (1996). Exhaustive search for low-autocorrelation binary sequences. *Journal of Physics A: Mathematical and General*, 29(18):L473.

[106] Militzer, B., Zamparelli, M., and Beule, D. (1998). Evolutionary search for low autocorrelated binary sequences. *IEEE Transactions on Evolutionary Computation*, 2(1):34–39.

[107] Millan, W. (1998). How to improve the nonlinearity of bijective S-boxes. In *Australasian Conference on Information Security and Privacy*, pages 181–192. Springer.

[108] Millan, W., Burnett, L., Carter, G., Clark, A., and Dawson, E. (1999). Evolutionary heuristics for finding cryptographically strong S-boxes. In *International Conference on Information and Communications Security*, pages 263–274. Springer.

[109] Mow, W. H., Du, K.-L., and Wu, W. H. (2015). New evolutionary search for long low autocorrelation binary sequences. *IEEE Transactions on aerospace and electronic systems*, 51(1):290–303.

[110] Mroczkowski, P. (2009). Generating Pseudorandom S-Boxes-a Method of Improving the Security of Cryptosystems Based on Block Ciphers. *Journal of Telecommunications and Information Technology*, pages 74–79.

[111] Nasrabadi, M. A. and Bastani, M. H. (2010). A survey on the design of binary pulse compression codes with low autocorrelation. In *Trends in Telecommunications Technologies*. IntechOpen.

[112] Nunn, C. J. and Coxson, G. E. (2008). Best-known autocorrelation peak sidelobe levels for binary codes of length 71 to 105. *IEEE transactions on Aerospace and Electronic Systems*, 44(1):392–395.

[113] Nyberg, K. (1991a). Perfect nonlinear S-boxes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 378–386. Springer.

[114] Nyberg, K. (1991b). Perfect nonlinear S-boxes. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 378–386. Springer.

[115] Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.

[116] Oliynykov, R., Gorbenko, I., Kazymyrov, O., Ruzhentsev, V., Kuznetsov, O., Gorbenko, Y., Dyrda, O., Dolgov, V., Pushkaryov, A., Mordvinov, R., et al. (2015). A new encryption standard of ukraine: The kalyna block cipher. *IACR Cryptology ePrint Archive*, 2015:650.

[117] Orhanou, G., El Hajji, S., and Bentaleb, Y. (2010). Snow 3g stream cipher operation and complexity study. *Contemporary Engineering Sciences-Hikari Ltd*, 3(3):97–111.

[118] Packebusch, T. and Mertens, S. (2016). Low autocorrelation binary sequences. *Journal of Physics A: Mathematical and Theoretical*, 49(16):165001.

[119] Perrin, L. P. (2017). *Cryptanalysis, reverse-engineering and design of symmetric cryptographic algorithms*. PhD thesis, University of Luxembourg, Luxembourg, Luxembourg.

[120] Perrin, L. P. and Udovenko, A. (2017). Exponential s-boxes: a link between the s-boxes of belt and kuznyechik/streebog. *IACR Transactions on Symmetric Cryptology*, 2016(2):99–124.

[121] Picek, S., Cupic, M., and Rotim, L. (2016a). A new cost function for evolution of s-boxes. *Evolutionary computation*, 24(4):695–718.

[122] Picek, S., Santana, R., and Jakobovic, D. (2016b). Maximal nonlinearity in balanced boolean functions with even number of inputs, revisited. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3222–3229. IEEE.

[123] Piret, G., Roche, T., and Carlet, C. (2012). Picaro–a block cipher allowing efficient higher-order side-channel resistance. In *International Conference on Applied Cryptography and Network Security*, pages 311–328. Springer.

[124] Pott, A. (2006). *Finite geometry and character theory*. Springer.

[125] Prestwich, S. D. (2013). Improved branch-and-bound for low autocorrelation binary sequences. *arXiv preprint arXiv:1305.6187*.

[126] Reeds III, J. A. (1992). Cryptosystem for cellular telephony. US Patent 5,159,634.

[127] Rijmen, V. and Barreto, P. (2000). The anubis block cipher. *Submission to NESSIE*.

[128] Rijmen, V. and Preneel, B. (1997). A family of trapdoor ciphers. In *International Workshop on Fast Software Encryption*, pages 139–148. Springer.

[129] Rudin, W. (1959). Some theorems on fourier coefficients. *Proceedings of the American Mathematical Society*, 10(6):855–859.

[130] Rushanan, J. J. (2006). Weil sequences: A family of binary sequences with good correlation properties. In *2006 IEEE International Symposium on Information Theory*, pages 1648–1652. IEEE.

[131] SageMath. Preliminary State Standard of Republic of Belarus (STB P 34.101.31–2007). http://apmi.bsu.by/assets/files/std/belt-spec27.pdf.

[132] SageMath. SageMath Sbox library. https://github.com/sagemath/sage/blob/master/src/sage/crypto/sboxes.py.

[133] Sarkar, P. and Maitra, S. (2000). Nonlinearity bounds and constructions of resilient boolean functions. In *Annual International Cryptology Conference*, pages 515–532. Springer.

[134] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., and Ferguson, N. (1998). Twofish: A 128-bit block cipher. aes submission, 15 june 1998.

[135] Schotten, H. D. and Lüke, H. D. (2005). On the search for low correlated binary sequences. *AEU-International Journal of Electronics and Communications*, 59(2):67–78.

[136] Shapiro, H. S. (1952). *Extremal problems for polynomials and power series*. PhD thesis, Massachusetts Institute of Technology.

[137] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T. (2007). The 128-bit blockcipher clefia. In *International workshop on fast software encryption*, pages 181–195. Springer.

[138] Skipjack, N. (1998). KEA algorithm specifications. *Online document: http://csrc. nist. org/encryption/skipjack/skipjack. pdf*.

[139] Skolnik, M. I. (1970). Radar handbook.

[140] Soltanalian, M. and Stoica, P. (2012). Computational design of sequences with good correlation properties. *IEEE Transactions on Signal processing*, 60(5):2180–2193.

[141] Song, J., Babu, P., and Palomar, D. P. (2015). Sequence design to minimize the weighted integrated and peak sidelobe levels. *IEEE Transactions on Signal Processing*, 64(8):2051–2064.

[142] Souravlias, D., Parsopoulos, K. E., and Meletiou, G. C. (2017). Designing bijective S-boxes using Algorithm Portfolios with limited time budgets. *Applied Soft Computing*, 59:475–486.

[143] Standaert, F.-X., Piret, G., Rouvroy, G., Quisquater, J.-J., and Legat, J.-D. (2004). Iceberg: An involutional cipher efficient for block encryption in reconfigurable hardware. In *International Workshop on Fast Software Encryption*, pages 279–298. Springer.

[144] Stern, J. and Vaudenay, S. (1998). Cs-cipher. In *International Workshop on Fast Software Encryption*, pages 189–204. Springer.

[145] Tesař, P. (2010). A new method for generating high non-linearity s-boxes. *Radioengineering*, 19(1):23–26.

[146] Turyn, R. et al. (1968). Sequences with small correlation. In *Error correcting codes*, pages 195–228. Wiley New York.

[147] Wagner, D. (1999). The boomerang attack. In *International Workshop on Fast Software Encryption*, pages 156–170. Springer.

[148] Watanabe, D., Furuya, S., Yoshida, H., Takaragi, K., and Preneel, B. (2002). A new keystream generator mugi. In *International Workshop on Fast Software Encryption*, pages 179–194. Springer.

[149] Wiener, N. (1964). *Extrapolation, interpolation, and smoothing of stationary time series*. The MIT press.

[150] Wikipedia source (1999). Wikipedia. https://en.wikipedia.org/wiki/Iraqi_block_cipher.

[151] Wu, H., Bao, F., Deng, R. H., and Ye, Q.-Z. (1998). Cryptanalysis of rijmen-preneel trapdoor ciphers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 126–132. Springer.

[152] Xiu-tao, F. (2011). Zuc algorithm: 3gpp lte international encryption standard [j]. *Information Security and Communications Privacy*, 12.