J. Algebra Comb. Discrete Appl.
6(3) • 123–134

Received: 4 July 2018
Accepted: 23 April 2019

Journal of Algebra Combinatorics Discrete Structures and Applications

Bijective S-boxes of different sizes obtained from quasi-cyclic codes*

Research Article

Dusan Bikov, Iliya Bouyukliev, Stefka Bouyuklieva

Abstract: The aim of this paper is to construct S-boxes of different sizes with good cryptographic properties. An algebraic construction for bijective S-boxes is described. It uses quasi-cyclic representations of the binary simplex code. Good S-boxes of sizes 4, 6, 8, 9, 10, 11, 12, 14, 15, 16 and 18 are obtained.

2010 MSC: 94A60, 11T71, 06E30, 94B05

Keywords: S-box, Simplex code, Quasi-cyclic codes

1. Introduction

S-boxes are among the most common and essential components of the block ciphers. They provide block ciphers with resistance to known and potential cryptanalytic attacks. Therefore significant research effort has been made in developing methods for constructing S-boxes with optimal parameters and desirable cryptographic properties. There are well studied criteria that a good S-box has to fulfill to make the cipher resistant against differential and linear cryptanalyses. However, the construction of a cryptographically secure S-box is still a problem. For many years, properties as well as various techniques and methods for constructing good S-boxes have been investigated. The popular techniques for constructing S-boxes can be classified into three categories: algebraic structures, pseudo-random generation and different heuristic approaches.

The aim of this paper is the constructions of bijective S-boxes of different sizes with good cryptographic properties. To do this, we use binary quasi-cyclic codes. We need a method to construct

* *This work was supported by Bulgarian Science Fund under Contract DN-02-2/13.12.2016.*
Dusan Bikov; Faculty of Computer Science, Goce Delchev University, Shtip, Macedonia (email: dusan.bikov@ugd.edu.mk).
Iliya Bouyukliev; Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, P.O.Box 323, 5000 Veliko Tarnovo, Bulgaria (email: iliyab@math.bas.bg).
Stefka Bouyuklieva (Corresponding Author); Faculty of Mathematics and Informatics, St. Cyril and St. Methodius University of Veliko Tarnovo, Bulgaria (email: stefka@ts.uni-vt.bg).

the codes, then effective algorithms to compute the parameters of the corresponding S-boxes, and fast programs that implement these algorithms.

Increasing the size of input data leads to complicated computations that become more difficult and use too many objects. On the other hand, some of the algorithms are suitable for parallel implementation, which makes it possible to scan the intermediate objects at the same time, and thus allows the study of S-boxes of relatively large sizes. Therefore we have used the parallel library BOOLSPLG [1] to realize the presented constructions and to calculate the parameters of the obtained S-boxes. For the programs we have used GPU computing model with CUDA. This allows us to interact directly with the GPUs and run programs on them, thus effectively utilizing the advantages of parallelization (many details about the parallel algorithms and programs are given in [2]). For this research, we have used a graphic card NVIDIA GeForce Titan X Pascal which we have as a donation from NVIDIA Corporation.

The bijective S-boxes of size $n = 4$ have been extensively studied, classifications have been made, criteria of optimality are defined [12, 16, 17]. A general classification of all optimal S-boxes of size $n = 4$ is given in the work of Leander and Poschmann [12] in 2007, where the authors make a comprehensive analysis and find all classes of affine equivalent S-boxes for which they explore linearity, differential uniformity, and algebraic degree. Saarinen [16] in 2011 has extended the work of Leander and Poschmann, making a comprehensive analysis and finding all classes of linear equivalence. In the article [17] from 2015, a new classification of S-boxes of size $n = 4$ is made, dividing them into 183 different categories.

Almost everything about the S-boxes with size $n = 4$ is clear, but the situation for larger sizes is radically different. S-boxes with size $n = 8$ are of particular interest due to the fact that these cryptographic primitives are embedded in the widely used cryptographic standards. It is still unclear whether there are S-boxes for $n = 8$ with nonlinearity greater than 112 or other better cryptographic properties. Therefore, a construction of S-boxes with better "optimal" cryptographic properties is a very topical issue. We construct bijective S-boxes of sizes 4, 6, 8, 9, 10, 11, 12, 14, 15, 16 and 18.

The paper is organized as follows. Section 2 provides the necessary definitions and assertions required for our constructions and for the investigation of the obtained S-boxes. In Section 3 we describe the constructions that we use. The results are presented in Section 4.

2. Preliminaries

S-boxes are also called multi-output Boolean functions or *vectorial Boolean functions* because of their connection with Boolean functions. They are defined as functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ (also called (n, m)-functions or (n, m) S-boxes) where n and m are positive integers. An S-box can be represented by the vector $(f_1, f_2, \ldots, f_m)$, where f_i are Boolean functions of n variables, called its coordinate functions, $i = 1, 2, \ldots, m$. Then the $m \times 2^n$ matrix

$$B_S = \begin{pmatrix} TT(f_1) \\ \vdots \\ TT(f_m) \end{pmatrix}$$

also represents the considered S-box, where $TT(f_i)$ is the Truth Table of the Boolean function f_i, $i = 1, \ldots, m$ [5]. An S-box is *bijective (or invertible)*, if $n = m$ and S is an invertible function. If an S-box is represented by the matrix B_S, then it is bijective if and only if $n = m$ and the columns of B_S are all binary vectors of length n.

To connect the codes with S-boxes, we consider the binary simplex codes. An $[n, k]$ linear code C over the binary field $\mathbb{F}_2$ is a k-dimensional linear subspace of $\mathbb{F}_2^n$. The Hamming weight of a vector in $\mathbb{F}_2^n$ is equal to the number of its nonzero coordinates, and the Hamming distance between two vectors is the number of positions in which they differ. We call C an $[n, k, d]$ code if d is the minimum distance of the code, $d = \min\{d(x, y),\ x, y \in C, x \neq y\}$. Two binary codes of length n are equivalent if there is a permutation $\sigma \in S_n$ which maps one code to the other.

Let G_S be an $n \times (2^n - 1)$ binary matrix whose columns are all nonzero binary vectors of length n.

The code generated by G_S is called a *binary simplex code* and we denote it by $\mathscr{S}_n$. This is a constant weight code with nonzero weight 2^{n-1} and its dual code is the $[2^n-1, 2^n-1-n, 3]$ binary Hamming code (for more information see [14]). The simplex code $\mathscr{S}_n$ can be considered as an irreducible cyclic code. Moreover, if $h(x)$ is any primitive polynomial (a minimal polynomial of a primitive element α of the field $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$), the code with check polynomial $h(x)$ is equivalent to the simplex code $\mathscr{S}_n$ [14]. The columns of G_S can be considered as the binary representations of the integers $1, \ldots, 2^n-1$ which we denote by $\overline{1}, \ldots, \overline{2^n-1}$, respectively. Suppose that the columns of G_S are ordered as $G_S = (\overline{1}^T \ \cdots \ \overline{2^n-1}^T)$. Let $\overline{G_S} = (\overline{0}^T \ \overline{1}^T \ \cdots \ \overline{2^n-1}^T)$. Obviously,

$$\overline{G_S} = \begin{pmatrix} TT(x_1) \\ \vdots \\ TT(x_n) \end{pmatrix}$$

and so $\overline{\mathscr{S}}_n = \langle TT(x_1), \ldots, TT(x_n) \rangle$, where $\overline{\mathscr{S}}_n$ is the extended simplex code (extended with a zero coordinate). This proves the following theorem that is very important in our research.

Theorem 2.1. *An S-box is invertible if and only if $n = m$ and the matrix B_S generates a $[2^n, n, 2^{n-1}]$ code equivalent to the extended simplex code $\overline{\mathscr{S}}_n$.*

In order to study the cryptographic properties of an S-box related to the linearity, we need to consider all non-zero linear combinations of its coordinate functions, namely $S_b = b \cdot S = b_1 f_1 \oplus \cdots \oplus b_m f_m$, where $b = (b_1, \ldots, b_m) \in \mathbb{F}_2^m$, $b \neq 0$. These are the component functions of the considered S-box. The Truth Tables of the component functions are all nonzero linear combinations of the rows of matrix B_S and so they coincide with the nonzero codewords of the linear code generated by B_S. In the case of bijective S-box, instead of a generator matrix we can consider a $(2^n-1) \times 2^n$ matrix whose rows are all nonzero codewords of the given extended simplex code (which are the component functions of the corresponding S-box). There are a few different definitions for equivalence of S-boxes but for all of them equivalent but different linear codes can lead to nonequivalent S-boxes with different characteristics. Therefore we consider different codes all of which are equivalent to $\overline{\mathscr{S}}_n$, and these codes produce S-boxes with different cryptographic properties.

Since the building blocks of an S-box are Boolean functions, we define in the beginning some of their parameters which are important for cryptography. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a Boolean function of n variables. The functions of the form $f_a(x) = a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n = a \cdot x$ are linear, and $f_a(x) \oplus b = a_1x_1 \oplus a_2x_2 \oplus \cdots \oplus a_nx_n \oplus b$ are affine functions, $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$, $b \in \mathbb{F}_2$, $x = (x_1, x_2, \ldots, x_n)$. The Walsh coefficients of the Boolean function f are defined as

$$f^W(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f_a(x)}.$$

The 2^n-tuple $(f^W(\overline{0}), f^W(\overline{1}), \ldots, f^W(\overline{2^n-1}))$ is called the *Walsh spectrum* of the function f, the set of all Walsh coefficients is its *Walsh distribution*, and the maximum absolute value of an Walsh coefficient of f is its *linearity* $Lin(f) = \max\{|f^W(a)| \mid a \in \mathbb{F}_2^n\}$.

Another important parameter which is closely connected with the linearity is the nonlinearity. *Nonlinearity* $nl(f)$ of the Boolean function f is the minimum Hamming distance from f to the nearest affine function:

$$nl(f) = \min\{d_H(f, g) \mid g \ - \ \text{affine function}\}.$$

The relation between the linearity and nonlinearity is given by the equality $Lin(f) = 2^n - 2nl(f)$ [4]. Obviously, the minimum linearity corresponds to maximum nonlinearity. The Parseval's equality $\sum_{a \in \mathbb{F}_2^n} (f^W(a))^2 = 2^{2n}$ gives that $Lin(f) \geq 2^{n/2}$ [4]. Functions attaining this lower bound are called bent functions.

The Walsh spectrum of S is defined as the collection of all Walsh spectra of its component functions. The linearity and nonlinearity of S are defined as

$$Lin(S) = \max_{b \in \mathbb{F}_2^m \setminus \{0\}} Lin(b \cdot S), \qquad nl(S) = \min_{b \in \mathbb{F}_2^m \setminus \{0\}} nl(b \cdot S).$$

The nonlinearity and the Walsh spectrum of a Boolean function can be calculated using linear codes. It is well known that the set of Truth Tables of all affine Boolean functions coincides with the set of codewords of the Reed-Muller code of first order $\mathcal{RM}(1, n)$, which is a linear $[2^n, n+1, 2^{n-1}]$ code with a generator matrix

$$G(\mathcal{RM}(1,n)) = \begin{pmatrix} TT(1) \\ TT(x_1) \\ \vdots \\ TT(x_n) \end{pmatrix}.$$

The code $\mathcal{RM}(1, n)$ is obtained from the extended simplex code by adding the all ones vector $\mathbf{1}$ to its generator matrix. This means that $\mathcal{RM}(1, n)$ consist of the codewords of $\overline{\mathscr{S}}_n$ and their complements, or $\mathcal{RM}(1, n) = \overline{\mathscr{S}}_n \cup (\mathbf{1} + \overline{\mathscr{S}}_n)$.

The nonlinearity of the Boolean function f is the Hamming distance from $TT(f)$ to the Reed-Muller code $\mathcal{RM}(1, n)$, or $nl(f) = d_H(TT(f), \mathcal{RM}(1, n))$. This means that we can use algorithms for calculating the distance from a vector to a code (or for minimum distance of a linear code) to find the nonlinearity and linearity of a Boolean function without having the whole Walsh spectrum. If f is an affine function then $nl(f) = 0$, otherwise $nl(f)$ is equal to the minimum distance of the linear code with a generator matrix $G_f = \begin{pmatrix} G(\mathcal{RM}(1,n)) \\ TT(f) \end{pmatrix}$. This helps us to calculate the nonlinearity of an S-box as the minimum distance of the linear code generated by the matrix $\overline{B_S} = \begin{pmatrix} G(\mathcal{RM}(1,n)) \\ B_S \end{pmatrix}$. Let us recall that if there is a coordinate function S_b which is affine then $nl(S) = 0$.

Other important parameters of an S-box are the algebraic degree $\deg(S)$, the differential uniformity δ and the autocorrelation $AC(S)$. Any Boolean function f can be represented uniquely as a binary polynomial of n variables whose monomials have the form $x_{i_1} x_{i_2} \cdots x_{i_k}$, $1 \le i_1 < i_2 < \cdots < i_k \le n$, $0 \le k \le n$, which is called the *algebraic normal form* ANF of f. The degree of this polynomial is the algebraic degree of the Boolean function denoted by $\deg(f)$. The algebraic degree of an $m \times n$ S-box is equal to the minimum algebraic degree of the component functions of S, $deg(S) = \min\{\deg(b \cdot S), b \in \mathbb{F}_2^m \setminus \{0\}\}$.

Autocorrelation of the Boolean function f is defined by

$$AC(f) = \max\{|\sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(x \oplus w)}| \mid w \in \mathbb{F}_2^n\}.$$

The autocorrelation of an S-box is the maximal autocorrelation of its components functions, $AC(S) = \max\{AC(b \cdot S), b \in \mathbb{F}_2^m \setminus \{0\}\}$.

The differential uniformity of an $(n \times n)$ S-box S is defined by:

$$\delta = \max_{\alpha, \beta \in \mathbb{F}_2^n, \alpha \neq 0} |\{x \in \mathbb{F}_2^n | S(x) \oplus S(x \oplus \alpha) = \beta\}|.$$

An S-box should have a differential uniformity as low as possible. The smallest possible value of δ in the case of bijective S-boxes is 2. Summarized results for good S-boxes are presented in [9, 10].

3. The considered constructions

In this section we present the main definitions and properties of the linear codes that are important for our constructions. A linear code of length n is *quasi-cyclic (QC) code* if a cyclic shift of a codeword by m positions results in another codeword. Obviously, m must divide the length n of the code, and n/m is called *index* of the considered QC code. If $m = 1$ the code is cyclic so QC codes are a generalization of cyclic codes. Many quasi-cyclic codes have the largest minimum distance among the linear codes with given length and dimension. There are different methods to construct quasi-cyclic codes. A QC code of length lm and index l is generated by a block matrix such that each block is an $m \times m$ circulant. The structure of QC codes is studied in [6, 11, 13].

To connect the codes with S-boxes, we consider the binary simplex codes as quasi-cyclic codes. As we defined in Section 2, G_S is the $n \times (2^n - 1)$ generator matrix of $\mathscr{S}_n$ such that the columns of G_S are ordered as $G_S = (\overline{1} \ \cdots \ \overline{2^n - 1})$. We use two more constructions for the codes equivalent to $\mathscr{S}_n$.

Let $K = \mathbb{F}_{2^n}$ be a finite field, α be its primitive element, $2^n - 1 = mr$, $1 < m, r < 2^n - 1$, and $\beta = \alpha^r$. If $G = \langle\beta\rangle < K^*$ then G is a cyclic group of order m and $G, \alpha G, \alpha^2 G, \ldots, \alpha^{r-1} G$ are all different cosets of G in K^*. For our constructions, we use left circulant matrices and the trace map. The trace map $Tr : K \to \mathbb{F}_2$ is defined as $Tr(\xi) = \xi + \xi^2 + \xi^4 + \cdots + \xi^{2^{n-1}}$, $\xi \in K$.

We present two constructions of quasi-cyclic codes. For the first construction, we need the left circulant $m \times m$ matrices $C_a = (Tr(\alpha^{ma}\beta^{i+j}))_{0 \le i,j \le m-1}$, $a = 0, 1, \ldots, r-1$, and the left circulant block matrix

$$M_1 = \begin{pmatrix} C_0 & C_1 & \ldots & C_{r-1} \\ C_1 & C_2 & \ldots & C_0 \\ \vdots & \vdots & \ddots & \vdots \\ C_{r-1} & C_0 & \ldots & C_{r-2} \end{pmatrix}. \tag{1}$$

The second construction is similar to the first one but the constructed block matrix is not circulant. We use again left circulant matrices but these are $D_a = (Tr(\alpha^a\beta^{i+j}))_{0 \le i,j \le m-1}$, $a = 0, 1, \ldots, 2r-2$. We construct a block matrix M_2 in the following way:

$$M_2 = \begin{pmatrix} D_0 & D_1 & \ldots & D_{r-1} \\ D_1 & D_2 & \ldots & D_r \\ \vdots & \vdots & \ddots & \vdots \\ D_{r-1} & D_r & \ldots & D_{2r-2} \end{pmatrix}. \tag{2}$$

There are two main differences between these two constructions. First, in the circulants C_a and D_a we multiply the powers of β by α^{ma} and α^a, respectively. Second, in the block matrix M_1 we take the indexes of the circulants modulo r. The following theorem is important for our research. Both codes generated by M_1 and M_2 are quasi-cyclic.

Theorem 3.1. *The code whose nonzero codewords are the rows of the matrix M_2 is equivalent to the simplex $[2^n - 1 = mr, n, 2^{n-1}]$ code. If m and r are coprime, the same is true for the code whose nonzero codewords are the rows of the matrix M_1.*

Proof. The proof consists of two parts. First, we will prove that the Hamming weight of each row of both matrices is 2^{n-1}. Second, the sum of any two rows of M_1 (respectively M_2) is a row in the same matrix. This proves that the rows in any of both matrices are the nonzero codewords of a linear code of length $2^n - 1$ and dimension n. Moreover, all nonzero codewords of the corresponding code have the same weight 2^{n-1} and therefore it is a linear constant-weight $[2^n - 1, n, 2^{n-1}]$ code. Up to equivalence, the simplex code $\mathscr{S}_n$ is the unique code with these parameters for a given positive integer n.

Consider the elements $M_1[i, j]$ and $M_2[i, j]$, $0 \le i, j \le 2^n - 2$. If $i = mi_1 + i_2$, $j = mj_1 + j_2$,

$0 \le i_1, j_1 \le r-1$, $0 \le i_2, j_2 \le m-1$, then

$$\begin{aligned} M_2[i,j] &= D_{i_1+j_1}[i_2,j_2] = Tr(\alpha^{r(i_2+j_2)+i_1+j_1}) = Tr(\alpha^{(i_1+ri_2)+(j_1+rj_2)}), \\ M_1[i,j] &= C_{(i_1+j_1)_r}[i_2,j_2] = Tr(\alpha^{r(i_2+j_2)+m(i_1+j_1)}) \\ &= Tr(\alpha^{(mi_1+ri_2)+(mj_1+rj_2)}), \end{aligned}$$

where $(i_1+j_1)_r = (i_1+j_1) \mod r$.

If m and r are coprime, for a fixed i the exponents $(mi_1+ri_2)+(mj_1'+rj_2')$ and $(mi_1+ri_2)+(mj_1''+rj_2'')$ are different for $j' = mj_1' + j_2' \ne j'' = mj_1'' + j_2''$, where $j', j'' \in \{0,1,\dots,mr-1\}$. Hence the i-th row of M_1 consists of the traces of all nonzero elements of the field K, and this holds for all $i = 0,1,\dots,mr-1$. Since exactly half of the elements of the field have trace 1 and the other half (including 0) have trace 0, the Hamming weight of each row is 2^{n-1}.

For the matrix M_2 the integers m and r are not necessarily coprime. It is easy to see that if $j_1' + rj_2' = j_1'' + rj_2''$, $0 \le j_1', j_1'' \le r-1$, $0 \le j_2', j_2'' \le m-1$, then

$$j_1' - j_1'' = r(j_2'' - j_2') \ \Rightarrow r \mid j_1' - j_1'' \ \Rightarrow j_1' = j_1'' \ \Rightarrow j_2' = j_2''.$$

Hence the i-th row of M_2 consists of the traces of all nonzero elements of the field, and this holds for all $i = 0,1,\dots,mr-1$.

Take the s-th and the t-th rows of the matrix M_1 (respectively M_2), and $s = ms_1 + s_2$, $t = mt_1 + t_2$, $0 \le s_1, t_1 \le r-1$, $0 \le s_2, t_2 \le m-1$. Then

$$\begin{aligned} M_1[s] + M_1[t] &= (Tr(\alpha^{(ms_1+rs_2)+(mj_1+rj_2)}) + Tr(\alpha^{(mt_1+rt_2)+(mj_1+rj_2)}))_{j=0,\dots,mr-1} \\ &= (Tr(\alpha^{(ms_1+rs_2)+(mj_1+rj_2)} + \alpha^{(mt_1+rt_2)+(mj_1+rj_2)}))_{j=0,\dots,mr-1} \\ &= (Tr((\alpha^{(ms_1+rs_2)} + \alpha^{(mt_1+rt_2)})\alpha^{(mj_1+rj_2)}))_{j=0,\dots,mr-1}. \end{aligned}$$

Since $ms_1 + rs_2 \not\equiv mt_1 + rt_2 \pmod{mr}$, then $\alpha^{(ms_1+rs_2)} + \alpha^{(mt_1+rt_2)} \ne 0$ and so $\alpha^{(ms_1+rs_2)} + \alpha^{(mt_1+rt_2)} = \alpha^c$ for some $c \in \{0,1,\dots,mr-1\}$. If m and r are coprime, $c = mc_1 + rc_2$, $0 \le c_1 \le r-1$, $0 \le c_2 \le m-1$. It follows that

$$M_1[s] + M_1[t] = (Tr(\alpha^c \alpha^{mj_1+rj_2}))_{j=0,\dots,mr-1} = (Tr(\alpha^{mc_1+rc_2}\alpha^{mj_1+rj_2}))_{j=0,\dots,mr-1}$$

$$= (Tr(\alpha^{m(c_1+j_1)}\alpha^{r(c_2+j_2)}))_{j=0,\dots,mr-1} = (Tr(\alpha^{m(c_1+j_1)}\beta^{c_2+j_2}))_{j=0,\dots,mr-1}$$

$$\Rightarrow M_1[s] + M_1[t] = (C_{c_1}[c_2], C_{c_1+1}[c_2], \dots, C_{c_1-1}[c_2]).$$

Hence $M_1[s] + M_1[t]$ is equal to the c-th row of M_1.

In the similar way we obtain that $M_2[s] + M_2[t]$ is equal to the c-th row of M_2, where $\alpha^{s_1+rs_2} + \alpha^{t_1+rt_2} = \alpha^c$, $0 \le c \le mr-1$.

So we proved that the rows of the matrix M_i, $i = 1,2$, are all nonzero codewords in a linear code, equivalent to $\mathscr{S}_n$ (for M_1 we need m and r to be coprime). □

Denote by $C(M_1)$ and $C(M_2)$ the codes generated by M_1 and M_2, respectively. The above theorem says that $C(M_1) \cong C(M_2) \cong \mathscr{S}_n$. Let $\overline{M}_1$ be the matrix M_1 extended with one zero column in the beginning, and $C(\overline{M}_1)$ be the code whose codewords are the rows of $\overline{M}_1$ (the same for M_2). Then any generator matrix of $C(\overline{M}_1)$ can be considered as an invertible S-box. Since all these S-boxes generate the same code $C(\overline{M}_1)$ and have the same component Boolean functions, they have the same linearity, nonlinearity, degree, autocorrelation and differential uniformity. Therefore it doesn't matter which generator matrix of $C(\overline{M}_1)$ (or $C(\overline{M}_2)$) we consider, so we take the matrices $G(M_i)$ whose rows are the first n linearly independent rows of M_i, $i = 1,2$.

We use the described constructions of quasi-cyclic codes to obtain S-boxes in two different ways. The constructed S-boxes are called here QCS-boxes. Moreover, we take a permutation $\pi \in S_r$ which permutes the block-columns of M_1 (or M_2). Unfortunately, the QCS-boxes $G(M_1\pi)$ and $G(M_2\pi)$ do not have good nonlinearity. This construction is natural but looking for better results we transform the matrices M_1 and M_2.

(C1) First construction:

We describe this construction for the matrix M_1, but we use it in the same way for M_2. Let $\sigma \in S_{2^n}$ permutes the columns of the matrix $G(\overline{M_1})$ into the vectors $\overline{0}, \overline{1}, \ldots, \overline{2^n-1}$. Now we consider the QCS-box, represented by the matrix $\sigma(G(\overline{M_1\pi}))$.

The nonlinearity of this QCS-box is equal to the minimum distance d of the code generated by the matrix

$$G_1^{(\pi)} = \left(\begin{array}{c|c} 1 & 11\ldots1 \\ \hline 0 & G(M_1) \\ 0 & G(M_1\pi) \end{array}\right).$$

This follows from the fact that σ maps the above matrix into

$$\begin{pmatrix} 111\ldots1 \\ \sigma(G(\overline{M}_1)) \\ \sigma(G(\overline{M_1\pi})) \end{pmatrix} = \begin{pmatrix} G(\mathcal{RM}(1,n)) \\ \sigma(G(\overline{M_1\pi})) \end{pmatrix}.$$

The quasi-cyclic structure of the matrices provides a faster algorithm for calculating linearity of the obtained QCS-boxes. The code generated by $G_1^{(\pi)}$ is invariant under the action of the cyclic group $\langle\tau\rangle$ where $\tau \in S_{2^n}$ is presented as a product of independent cycles in the following way

$$\tau = (1, 2, \ldots, m)(m+1, \ldots, 2m)\ldots(mr-r+1, \ldots, mr).$$

The group $\langle\tau\rangle$ defines a relation of equivalence in the considered code (two codewords u and v are equivalent if $u = \tau^s(v)$, $0 \le s \le m-1$). To calculate the minimum distance d of the above code we need only one codeword from each equivalence class. We present this observation in the next proposition

Proposition 3.2. *Let $A = (A_0, A_1, \ldots, A_{r-1})$ and $B = (B_0, B_1, \ldots, B_{r-1})$ be block matrices, where A_i and B_i are $m \times m$ circulants, $i = 0, 1, \ldots, r-1$. If $a_0, a_1, \ldots, a_{m-1}$ are the rows of A, and $b_0, b_1, \ldots, b_{m-1}$ are the rows of B, then $d(a_i, b_j) = d(a_{i+1}, b_{j+1})$ for $0 \le i, j \le m-1$ ($i+1$ and $j+1$ are taken modulo m).*

Since $f^W(a) = 2^n - 2d(f, f_a)$ [4], Proposition 3.2 shows that the Walsh distributions of all Boolean function in one equivalence class are the same. This allows us to calculate the linearity (the same for the other parameters) listing only r of the component functions of the considered QCS-box.

(C2) Second construction:

For each of the circulants $C_0, C_1, \ldots, C_{r-1}$ we reorder the columns in the following way: first we take the last column, then the previous one, and in the end the first one. In this way we obtain the circulants C_0', C_1', …, C_{r-1}', which define the matrix $G(M_1')$. If

$$C_a = \begin{pmatrix} c_1^{(a)} & c_2^{(a)} & \cdots & c_{m-1}^{(a)} & c_m^{(a)} \\ c_2^{(a)} & c_3^{(a)} & \cdots & c_m^{(a)} & c_1^{(a)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m-1}^{(a)} & c_m^{(a)} & \cdots & c_{m-3}^{(a)} & c_{m-2}^{(a)} \\ c_m^{(a)} & c_1^{(a)} & \cdots & c_{m-2}^{(a)} & c_{m-1}^{(a)} \end{pmatrix}, \text{ then } C_a' = \begin{pmatrix} c_m^{(a)} & c_{m-1}^{(a)} & \cdots & c_2^{(a)} & c_1^{(a)} \\ c_1^{(a)} & c_m^{(a)} & \cdots & c_3^{(a)} & c_2^{(a)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{m-2}^{(a)} & c_{m-3}^{(a)} & \cdots & c_m^{(a)} & c_{m-1}^{(a)} \\ c_{m-1}^{(a)} & c_{m-2}^{(a)} & \cdots & c_1^{(a)} & c_m^{(a)} \end{pmatrix}.$$

Hence C'_a is a right circulant matrix, and since $C_a = (Tr(\alpha^{ma}\beta^{i+j}))_{0\le i,j\le m-1}$, then $C'_a = (Tr(\alpha^{ma}\beta^{i-j-1}))_{0\le i,j\le m-1}$, $a = 0, 1, \ldots, r-1$. So M'_1 is a left circulant block matrix whose blocks are right circulants.

We use $G(M'_1\pi)$, where $\pi \in S_r$ is a permutation on the block-columns of $G(M'_1)$, but now we compute the minimum distance d of the code generated by the matrix

$$G_2^{(\pi)} = \left(\begin{array}{c|c} 1 & 11\ldots1 \\ \hline 0 & G(M_1) \\ 0 & G(M'_1\pi) \end{array}\right).$$

If σ is defined as in (C1) then d is the nonlinearity of the QCS-box represented by the matrix $\sigma(\overline{G(M'_1\pi)})$. This construction extends the first construction C1.

Using such constructions we can easily compute the Walsh distributions of the considered QCS-boxes and take only those that have large nonlinearity. We apply these constructions to obtain $n \times n$ bijective S-boxes with sizes $4 \le n \le 18$ such that $2^n - 1$ is not prime.

Remark 3.3. *In our paper [3] we considered only the first construction technique and its application only in the case of 8×8 S-boxes. Now we extend our study to more construction methods with applications to bijective S-boxes of sizes 4, 6, 8, 9, 10, 11, 12, 14, 15, 16 and 18.*

4. Constructed QCS-boxes

In this section we present the constructed QCS-boxes, that have linearity close to the Parseval bound, small differential uniformity δ ($\delta \ge 2$), high algebraic degree and small autocorrelation $AC(S)$, and compare them with the best known S-boxes. We use a fixed finite field with 2^n elements, generated by a primitive binary polynomial $g(x)$ of degree n, and take $\alpha = x$. If we change the generator polynomial of the field, we can get different results, but we have not studied how the selected field affects the parameters of the constructed S-boxes. This is still an open problem as well as the equivalence of the constructed QCS-boxes with respect to an equivalence relation (affine, CCZ, or another equivalence relation).

The obtained results are presented in tables. The first column shows the used construction (C1 or C2), the used matrix (M_1 or M_2) and the integers m and r, the next columns contain the values of the computed cryptographic parameters, and the last column gives the number of the constructed QCS-boxes in each of the cases. For $r \le 15$ we study the S-boxes for all permutations $\pi \in S_r$ used as it is described in C1 and C2, otherwise we consider only a part of the permutations (in this cases the study of the S-boxes constructed from the matrices M_1 and M_2 using construction methods C1 and C2 is not completed - we put $*$ in the last column of the corresponding row in the table). In the tables we list only those S-boxes which have linearity $Lin \le 2^{n/2+1}$. For example, if $m = 5$, $r = 3$, we have $3! = 6$ permutations π but only three of the S-boxes constructed in the combination (C1, M_1) have linearity 8 (the other three S-boxes have bigger linearity).

All QCS-boxes with minimum linearity among the constructed S-boxes of even sizes have linearity $Lin(S) = 2^{n/2+1}$ (respectively nonlinearity $nl(S) = 2^{n-1} - 2^{n/2}$). This is the best known linearity for S-box of the corresponding sizes but it is far from the Parseval bound $Lin(S) \ge 2^{n/2}$.

1. **n=4:** A definition for optimal 4×4 S-boxes is given in [12]. Using our constructions and the field generated by the polynomial $1 + x + x^4$, we obtain many optimal S-boxes presented in Table 1. All of them have the same parameters $Lin(S) = 8$, $nl(S) = 4$, $deg(S) = 3$, $AC(S) = 8$, and $\delta = 4$.
2. **n=6:** The used generator polynomial is $g(x) = 1 + x + x^3 + x^4 + x^6$. The constructed bijective 6×6 QCS-boxes with best cryptographic properties are given in Table 2. They have linearity 16, most of them have algebraic degree 5, all but one have differential uniformity 4, and their autocorrelations are 16, 32, 64.

Table 1. Bijective 4×4 QCS-boxes

QCS-boxes	*Lin*	*nl*	δ	*deg*(S)	*AC*(S)	number
C1, M_1, $m = 5, r = 3$	8	4	4	3	8	3
C1, M_1, $m = 3, r = 5$	8	4	4	3	8	60
C1, M_2, $m = 5, r = 3$	8	4	4	3	8	3
C1, M_2, $m = 3, r = 5$	8	4	4	3	8	28
C2, M_1, $m = 5, r = 3$	8	4	4	3	8	3
C2, M_1, $m = 3, r = 5$	8	4	4	3	8	60
C2, M_2, $m = 5, r = 3$	8	4	4	3	8	6
C2, M_2, $m = 3, r = 5$	8	4	4	3	8	28

3. **n=8:** In this case $g(x) = 1+x+x^3+x^5+x^8$. Many of the constructed bijective 8×8 QCS-boxes for different values of m and r have parameters close or equal to the best known nonlinearity for this size, namely $nl(S) = 112$. This is the nonlinearity of the S-box of the most popular block cipher AES [8]. Our results are described in Table 3.

Table 2. Bijective 6×6 QCS-boxes

QCS-boxes	*Lin*	*nl*	δ	*deg*(S)	*AC*(S)	number
C1, M_1, $m = 9, r = 7$	16	24	4	5	16	7
	16	24	4	3	32	7
	16	24	4	2	64	7
C1, M_2, $m = 7, r = 9$	16	24	8	4	24	1
C2, M_1, $m = 9, r = 7$	16	24	4	5	16	7
C2, M_1, $m = 7, r = 9$	16	24	4	5	16	18
C2, M_2, $m = 21, r = 3$	16	24	4	5	16	1
C2, M_2, $m = 9, r = 7$	16	24	4	5	16	1
C2, M_2, $m = 7, r = 9$	16	24	4	5	16	1

4. **n $\geq$ 10:** We consider the sizes $n = 10$, $n = 12$, $n = 14$, $n = 16$ and $n = 18$. For these sizes we obtain QCS-boxes with nonlinearity $nl(S) = 2^{n-1} - 2^{n/2}$. The AES S-box have the same nonlinearity for $n = 8$ but these values are not so close to the Parseval bound $nl(S) =\leq 2^{n-1} - 2^{n/2-1}$. The used generator polynomials of the considered fields, as well as the cryptographic parameters of the constructed S-boxes in this case are presented in Table 4.

5. **n odd:** For the odd values of n, we apply all our constructions for sizes $n = 9$, $n = 11$ and $n = 15$ but only the second construction gives bijective S-boxes with good cryptographic properties. We list them in Table 5. The used generator polynomials are $1+x^5+x^9$, $1+x+x^2+x^3+x^6+x^7+x^9+x^{10}+x^{11}$ and $1 + x^2 + x^4 + x^5 + x^{15}$, respectively.

We use the procedures from the parallel library BoolSPLG [1] to design algorithms realizing the presented constructions. BoolSPLG is a CUDA library that includes algorithms for calculation of some cryptographic parameters and characteristics of Boolean and vectorial Boolean functions (S-boxes) (see

Table 3. Bijective 8×8 QCS-boxes

S-boxes	Lin	nl	δ	$deg(S)$	$AC(S)$	number
AES S-box [8]	32	112	4	7	32	/
C1, M_1, $m = 17, r = 15$	32	112	4	7	32	15
C1, M_1, $m = 15, r = 17$	32	112	4	5	48	4*
	32	112	4	5	56	4*
C2, M_1, $m = 85, r = 3$	32	112	4	7	32	3
C2, M_1, $m = 51, r = 5$	32	112	4	7	32	5
C2, M_1, $m = 17, r = 15$	32	112	4	7	32	15
C2, M_1, $m = 15, r = 17$	32	112	4	7	32	1*
C2, M_2, $m = 85, r = 3$	32	112	4	7	32	1
C2, M_2, $m = 51, r = 5$	32	112	4	7	32	1
C2, M_2, $m = 17, r = 15$	32	112	4	7	32	1

[7] for more information about CUDA parallel computing platform). The advantage of using parallel algorithms is essential for bigger values of n (especially for $n \geq 14$).

For our calculations, we used a server with Intel Xeon E5-2640 processor that contains two graphics cards. The first graphics card is NVIDIA GeForce GTX TITAN [15], which has 2688 cores running at 837 MHz and 288.4 GB/sec memory bandwidth. The second graphics card is NVIDIA TITAN X Pascal [15], which has 3584 cores running at 1.5 GHz and 549 GB/sec memory bandwidth. We have used CUDA Toolkit 8.0 and developed environment MS Visual Studio 2012.

All constructed QCS-boxes are available at the web page of the second author: `http://www.moi.math.bas.bg/~iliya/`. Each S-box is represented as a sequence of hexadecimal numbers, representing the corresponding columns in the matrix B_S. We give two examples.

1. The sequence $(0, b, 2, 7, 4, 5, f, 3, d, 9, a, 1, e, 8, c, 6)$ represents the S-box with

$$B_S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

   We give also the values of n, m and r, the permutation π in word representation, and the parameters $Lin(S)$, $nl(S)$, $AC(S)$ and δ. In this case $n = 4$, $m = 5$, $r = 3$, $\pi = (1,\ 0,\ 2)$, $Lin(S) = 8$, $nl(S) = 4$, $deg(S) = 3$, $AC(S) = 8$, and $\delta = 4$.

2. One of the obtained 6×6 S-boxes is
   $0, 2f, 17, 25, 2b, 1c, 32, f, 3a, 15, 21, 2e, 1b, 19, 34, 7, 3d, 33, 2a, 2c, 30, 9, 37, 2,$
   $29, d, 1d, c, 5, 1a, 20, 23, 1e, a, 39, 1f, 35, 3, 36, 28, 27, 18, 12, 4, 13, 3b, b, 1, 14,$
   $3f, 6, 11, e, 24, 26, 16, 3e, 22, 8, 2d, 3c, 10, 38, 31.$
   Its parameters are $n = 6$, $m = 9$, $r = 7$, $\pi = (6,\ 5,\ 4,\ 3,\ 2,\ 1,\ 0)$, $Lin(S) = 16$, $nl(S) = 24$, $deg(S) = 5$, $AC(S) = 16$, $\delta = 4$.

Acknowledgment: We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. We also thank the anonymous reviewer for his/her valuable comments.

Table 4. Bijective QCS-boxes for $n = 10$, $n = 12$, $n = 14$, $n = 16$ and $n = 18$

QCS-boxes	Lin	nl	δ	$deg(S)$	$AC(S)$	number
$n = 10$			$g(x) = 1 + x^3 + x^4 + x^8 + x^{10}$			
C1, M_1, $m = 33, r = 31$	64	480	4	9	64	1*
C2, M_1, $m = 341, r = 3$	64	480	4	9	64	3
C2, M_1, $m = 93, r = 11$	64	480	4	9	64	11
C2, M_1, $m = 33, r = 31$	64	480	4	9	64	1*
C2, M_2, $m = 341, r = 3$	64	480	4	9	64	1
C2, M_2, $m = 93, r = 11$	64	480	4	9	64	1
$n = 12$			$g(x) = 1 + x + x^5 + x^8 + x^{10} + x^{11} + x^{12}$			
C1, M_1, $m = 65, r = 63$	128	1984	4	11	128	1*
C2, M_1, $m = 819, r = 5$	128	1984	4	11	128	5
C2, M_1, $m = 585, r = 7$	128	1984	4	11	128	7
C2, M_1, $m = 455, r = 9$	128	1984	4	11	128	9
C2, M_1, $m = 315, r = 13$	128	1984	4	11	128	13
C2, M_2, $m = 1365, r = 3$	128	1984	4	11	128	1
C2, M_2, $m = 819, r = 5$	128	1984	4	11	128	1
C2, M_2, $m = 585, r = 7$	128	1984	4	11	128	1
C2, M_2, $m = 455, r = 9$	128	1984	4	11	128	1
C2, M_2, $m = 315, r = 13$	128	1984	4	11	128	1
$n = 14$			$g(x) = 1 + x + x^2 + x^3 + x^{10} + x^{12} + x^{14}$			
C1, M_1, $m = 129, r = 127$	256	8064	4	13	256	1*
C2, M_1, $m = 5461, r = 3$	256	8064	4	13	256	3
C2, M_2, $m = 5461, r = 3$	256	8064	4	13	256	1
$n = 16$			$g(x) = 1 + x + x^2 + x^4 + x^5 + x^9 + x^{10} + x^{12} + x^{16}$			
C1, M_1, $m = 257, r = 255$	512	32512	4	15	512	1*
C2, M_1, $m = 21845, r = 3$	512	32512	4	15	512	3
C2, M_1, $m = 13107, r = 5$	512	32512	4	15	512	5
C2, M_2, $m = 21845, r = 3$	512	32512	4	15	512	1
C2, M_2, $m = 13107, r = 5$	512	32512	4	15	512	1
$n = 18$	$g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^9 + x^{10} + x^{11} + x^{15} + x^{17} + x^{18}$					
C1, M_1, $m = 513, r = 511$	1024	130560	4	17	1024	1*

Table 5. Bijective QCS-boxes for $n = 9$, $n = 11$ and $n = 15$

QCS-boxes	Lin	nl	δ	$deg(S)$	$AC(S)$	number
C2, M_1, $n = 9$, $m = 73, r = 7$	44	234	2	8	48	7
C2, M_2, $n = 9$, $m = 73, r = 7$	44	234	2	8	48	1
C2, M_1, $n = 11$, $m = 89, r = 23$	88	980	2	10	88	1*
C2, M_1, $n = 15$, $m = 4681, r = 7$	360	16204	2	14	360	7
C2, M_2, $n = 15$, $m = 4681, r = 7$	360	16204	2	14	360	1

References

[1] D. Bikov, I. Bouyukliev, BoolSPLG: A library with parallel algorithms for Boolean functions and S-boxes for GPU.

[2] D. Bikov, I. Bouyukliev, Parallel Fast Walsh Transform Algorithm and its implementation with CUDA on GPUs, Cybernetics and Information Technologies, Cybernetics and Information Technologies 18(5) (2018) 21–43.

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

[4] C. Carlet, Boolean Functions for Cryptography and Error Correcting Codes, In: Boolean Models and Methods in Mathematics, Computer Science, and Engineering, Crama, Hammer, Cambridge University Press, 2010.

[5] C. Carlet, Vectorial Boolean Functions for Cryptography, In: Boolean Models and Methods in Mathematics, Computer Science, and Engineering, Crama, Hammer, (Eds.), Cambridge University Press, 2010.

[6] E. Z. Chen, New quasi-cyclic codes from simplex codes, IEEE Trans. Inform. Theory 53(3) (2007) 1193–1196.

[7] CUDA Zone.

[8] J.Daeman, V.Rijmen, The Design of Rijndael, AES–the advanced encryption standard, Springer-Verlag Berlin Heidelberg, 2002.

[9] I. Hussain, T. Shah, M. A. Gondal, W. A. Khan, Construction of Cryptographically Strong 8×8 S-boxes, World Applied Sciences Journal 13 (2011) 2389–2395.

[10] G. Ivanov, N. Nikolov, S. Nikova, Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties, Cryptogr. Commun. 8(2) (2016) 247–276.

[11] K. Lally, P. Fitzpatrick, Algebraic structure of quasi-cyclic codes, Discrete Applied Mathematics 111(1–2) (2001) 157–175.

[12] G. Leander, A. Poschmann, On the Classification of 4 Bit S-Boxes, In: Carlet C., Sunar B. (eds) Arithmetic of Finite Fields. WAIFI 2007. Lecture Notes in Computer Science, vol 4547. Springer, Berlin, Heidelberg (2007) 159–176.

[13] S. Ling, P. Solé, On the algebraic structure of quasi-cyclic codes I: finite fields, IEEE Trans. Inform. Theory 47(7) (2001) 2751–2760.

[14] F. J. MacWilliams, N. J. A. Sloane, The Theory of Error-Correcting Codes, North-Holland, Amsterdam 1977.

[15] NVIDIA Data Center.

[16] M. J. O. Saarinen, Cryptographic Analysis of all 4×4–bit S–boxes, In: Proceedings of the 18th International Conference on Selected Areas in Cryptography, ser. SAC 11. Springer-Verlag (2012) 118–133.

[17] W. Zhang, Z. Bao, V. Rijmen, M. Liu, A New Classification of 4-bit Optimal S-boxes and Its Application to PRESENT, RECTANGLE and SPONGENT. In: Leander G. (eds) Fast Software Encryption. Lecture Notes in Computer Science, vol 9054. Springer, Berlin, Heidelberg (2015) 494–515.