Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 57 (2017) 67–72

www.elsevier.com/locate/endm

S-Boxes from Binary Quasi-Cyclic Codes ⋆

Iliya Bouyukliev [1]

Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Veliko Tarnovo, Bulgaria

Dusan Bikov and Stefka Bouyuklieva [2,3]

Faculty of Mathematics and Informatics, Veliko Tarnovo University, Bulgaria

Abstract

In this paper we present a construction for S-boxes using quasi-cyclic codes. We obtain S-boxes with good nonlinearity.

Keywords: Vectorial Boolean function, S-box, quasi-cyclic code, simplex code.

1 Introduction

S-boxes are key building blocks in the design of the block ciphers. They have to be chosen carefully to make the cipher resistant against all kinds of attacks. In particular, there are well studied criteria that a good S-box has to fulfill to make the cipher resistant against differential and linear cryptanalyses.

⋆ This research is supported by Bulgarian Science Fund under Contract DH 02/2,13.12.2016

[1] Email: iliyab@math.bas.bg

[2] Email: dule.juve@gmail.com

[3] Email: stefka@uni-vt.bg

http://dx.doi.org/10.1016/j.endm.2017.02.012

1571-0653/© 2017 Elsevier B.V. All rights reserved.

To construct good S-boxes, we use quasi-cyclic codes. A code is said to be quasi-cyclic if every cyclic shift of a codeword by s positions results in another codeword ($s \geq 1$). There are many construction methods for good QC codes. Generally, a QC code of length lm and index l may be represented as the row space of a block matrix, each row of which has the form $(G_1, \ldots, G_l)$, where G_i is an $m \times m$ circulant. These rows, or the equivalent polynomial vectors, are conventionally called "generators". This form helps to connect quasi-cyclic codes with S-boxes. More precisely, we consider different (but equivalent) binary simplex codes of length $2^k - 1$ and dimension k as quasi-cyclic codes.

2 Vectorial Boolean Functions (S-Boxes)

A vectorial Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$ (also called (n, m) S-box or shortly S-box) can be represented by the vector $(f_1, f_2, \ldots, f_m)$, where f_i are Boolean functions in n variables, $i = 1, 2, \ldots, m$. The functions f_i are called the coordinate functions of the S-box. Then the $m \times 2^n$ matrix

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

represents the considered S-box, where $TT(f_i)$ is the Truth Table of the Boolean function f_i, $i = 1, \ldots, m$ [4]. An S-box is invertible, if $n = m$ and S is an invertible function. The following lemma is very important in our research.

Lemma 2.1 *An S-box is invertible if and only if $n = m$ and the matrix G_S generates a $[2^n, n]$ code equivalent to the extended simplex code $\overline{S}_n$ (extended with a zero coordinate).*

Recall that $\overline{S}_n = \langle TT(x_1), \ldots, TT(x_n) \rangle$. 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 the coordinates of the S-box, denoted by $S_b = b \cdot S = b_1 f_1 \oplus \cdots \oplus b_m f_m$, where $b = (b_1, \ldots, b_m) \in \mathbb{F}_2^m$. These are the component functions of the S-box.

To define linearity of an S-box, we need the Walsh coefficients of a Boolean function f. Let $a = (a_1, \ldots, a_n) \in \mathbb{F}_2^n$ and $f_a(x) = a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n$. The Walsh coefficient $f^W(a)$ is defined by

$$f^W(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f_a(x)} = 2^n - 2d_H(f, f_a).$$

Linearity of a Boolean function is the maximum absolute value of an Walsh coefficient of f: $Lin(f) = \max\{|f^W(a)| \mid a \in \mathbb{F}_2^n\}$. The Parseval's Equality $\sum_{a\in\mathbb{F}_2^n}(f^W(a))^2 = 2^{2n}$ gives that $Lin(f) \geq 2^{n/2}$ [3]. Functions attaining this lower bound are called bent functions.

Another important parameter which is closely connected with the linearity is the nonlinearity.

Definition 2.2 *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 of the Boolean function f is given by the equality $Lin(f) = 2^n - 2nl(f)$ [3]. Obviously, the minimum linearity corresponds to maximum nonlinearity.

The linearity and nonlinearity of the S-box 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. Actually, the set of the Truth Tables of all affine Boolean functions coincides with the set of codewords of the Reed-Muller code of first order $RM(1, n)$, which is a linear $[2^n, n+1, 2^{n-1}]$ code with a generator matrix

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

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

The nonlinearity of the Boolean function f is $nl(f) = d_H(TT(f), RM(1, n))$. 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. We compute the nonlinearity of the Boolean function f (which is not affine) using that $nl(f)$ is equal to the minimum distance of the linear code with

a generator matrix $G_f = \begin{pmatrix} G(RM(1,n)) \\ TT(f) \end{pmatrix}$. This helps us to calculate the nonlinearity of an S-box as the minimum distance of the linear code generated by the matrix $\overline{G}_S = \begin{pmatrix} G(RM(1,n)) \\ G_S \end{pmatrix}$. We have in mind that if there is a coordinate function S_b which is affine then $nl(S) = 0$.

The differential uniformity of an $(n \times m)$ S-box S with $n \geq m$, denoted by δ, is defined as the largest value in its difference distribution table (DDT) not counting the first entry in the first row. Differential uniformity is define by:

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

S should have a differential uniformity as low as is possible. It is well known that δ takes always only even values in the interval $[2^{n-m}, 2^n]$. The smallest possible value of δ in the case of bijective S-boxes ($n = m$) is 2. Summarized results for good S-boxes are presented in [5] and [6].

3 Quasi-Cyclic Codes

Let $K = \mathbb{F}_{q^n}$ be a finite field, α be its primitive element, $q^n - 1 = m \cdot r$, 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 $a \in \mathbb{Z}_r$ we define the circulant $m \times m$ matrix $C_a = (Tr(\alpha^{ma} \beta^{i+j}))_{0 \leq i,j \leq m-1}$. When m and r are coprime, the matrices C_a correspond to the different cosets of G in K^*. The next theorem has been proven in [2] as Lemmas 1 and 2.

Theorem 3.1 *If m and r are coprime, the code $C(0)$ whose nonzero codewords are the rows of the matrix $(C_0 \ C_1 \ \cdots \ C_{r-1})^T$ is an irreducible cyclic code of length m and dimension $ord_m(q)$. Moreover, the code whose nonzero codewords are the rows of the matrix*

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

is equivalent to the simplex $[2^n - 1 = mr, n, 2^{n-1}]$ code S_n.

Let $\overline{M}$ be the matrix M extended with one zero column in the beginning, and $C(\overline{M})$ be the code whose codewords are the rows of $\overline{M}$, where $q = 2$. Then any generator matrix of $C(\overline{M})$ can be considered as an invertible S-box. Since all these S-boxes generate the same code $C(\overline{M})$, they have the same linearity and nonlinearity.

We consider two constructions for S-boxes. For the first construction we take the first ml rows of the matrix M such that the obtained matrix G_m has rank n. Then we investigate all S-boxes $G_m\pi$ where $\pi \in S_r$ is a permutation of the circulants $C_0, C_1, \ldots, C_{r-1}$. Unfortunately, these S-boxes do not have good nonlinearity. Therefore we decided to check another construction which we describe in the next section.

4 A new Construction using Quasi-Cyclic Codes

Take again the matrix G_m. For this construction, we consider the code with a generator matrix

$$MR = \left(\begin{array}{c|c} 1 & 11\ldots1 \\ \hline 0 & G_m \end{array}\right) \tag{2}$$

This matrix generates a code which is equivalent to $RM(1, n)$ but has the structure of a quasi-cyclic code. We again use the matrices $G_m\pi$ but now we compute the minimum distance d of the code generated by the matrix

$$\left(\begin{array}{c|c} 1 & 11\ldots1 \\ \hline 0 & G_m \\ 0 & G_m\pi \end{array}\right).$$

If σ is a permutation which maps the Reed-Muller code $RM(1, n)$ to the code with a generator matrix MR then d is the nonlinearity of the S-box represented by the matrix $\sigma^{-1}(G_m\pi)$.

Using such constructions, we can compute the distance distribution of the codes easier. Moreover, we can check most S-boxes of this type for $n \leq 8$ (all S-boxes for $r \leq 15$) and to take only those which have good (for cryptography) parameters. The algorithms are implemented in CUDA C and realized in parallel using NVIDIA GPUs with compute capability 3.0 and higher [7]. We have described several algorithms for computing the Walsh spectrum implemented in CUDA for parallel execution on GPU in the manuscript [1].

We investigate the S-boxes, constructed in the above method, for $n = 4$ and 8. In these cases $2^4 - 1 = 3 \cdot 5$, $2^8 - 1 = 15 \cdot 17 = 5 \cdot 51$ $(= 3 \cdot 85)$. For $n = 4$ we obtain three optimal S-boxes ($Lin = 8$, $\delta = 4$). We have done the exhaustive search for $m = 17$, $r = 15$, and have concluded that there are 15 S-boxes with nonlinearity 112, and 601 S-boxes with nonlinearity 108. The most important parameters (see [6]) of these 15 S-boxes are presented in the following table. Note that the computed parameters coincide with the parameters of the S-box used in the block cipher AES. The calculations for the case $m = 15, r = 17$ are still in progress but we have already obtained one S-box with good nonlinearity. Its parameters are listed in the table.

QC S-box for $n = 8$	Lin	nl	δ - uniformity	$deg(S)$	$AC(S)_{max}$
$m = 17, r = 15$	32	112	4	7	32
$m = 15, r = 17$	32	112	16	5	48

References

[1] Bikov D., and I. Bouyukliev, *Parallel Fast Walsh Transform Algorithm and its Implementation with CUDA on GPUs*, preprint.

[2] Bouyuklieva S., and I. Bouyukliev, *On the binary quasi-cyclic codes*, Proceedings of the Intern. Workshop OCRT, Albena, Bulgaria, (2013), 59–64.

[3] Carlet C., "Boolean Functions for Cryptography and Error Correcting Codes", in *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, Crama, Hammer, (Eds.), Cambridge University Press, 2010.

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

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

[6] Ivanov G., N. Nikolov, and S. Nikova, *Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties*, Cryptography and Communications **8** (2016), 247–276.

[7] CUDA C Programming Guide, Availaible on:
`https://docs.nvidia.com/cuda/cuda-c-programming-guide/`