Construction of Optimal 4-bit S-boxes by Quasigroups of Order 4

Hristina Mihajloska
Faculty of Computer Science and Engineering
Ss Cyril and Methodius University
Skopje, Macedonia
email: hristina.mihajloska@finki.ukim.mk

Danilo Gligoroski
Department of Telematics
Norwegian University of Science and Technology
Trondheim, Norway
email: danilog@item.ntnu.no

***Abstract*—We present a new method for constructing cryptographically strong 4×4-bit S-boxes with the help of quasigroups of order 4. So far, cryptographers were constructing 4×4-bit S-boxes used in cryptographic primitives suitable for lightweight cryptography, only by exhaustive search of permutations of order 16. Our construction of 4×4-bit Quasigroup-S-boxes (Q-S-boxes) uses quasigroup string transformations. This methodology enables someone to work basically with several different strong S-boxes iteratively reusing only one hardware circuit and just changing a few parameters (called leaders in our method).**

Keywords-lightweight cryptography; quasigroups; quasigroup string transformations; S-boxes.

I. Introduction

In this paper, we focus on the Symmetric Lightweight Cryptography for cryptographic components that can be efficiently implemented into block ciphers. Although the Advanced Encryption Standard (AES) [1] block cipher is the most used cryptographic component, it was mainly designed to be efficient in software. For many constrained environments, using AES as a block cipher is either too expensive or there is no need for such a high level of security that it offers. Thus, it is not a surprise that in the last several years we see a dynamic development in the area of Lightweight Cryptography especially in the lightweight block ciphers such as PRESENT [2][3].

The main point of security in symmetric cryptography in almost all modern block ciphers is the substitution boxes also known as S-boxes. S-boxes work with a small unit of data, so they have to be distinguished with highly non-linear properties if they want to confuse the input data into the cipher.

PRESENT is an ultra-lightweight block cipher proposed by Bogdanov et al. [2]. It has been designed for extremely resource-constrained environments such as RFID tags. PRESENT is an SP-Network block cipher which consists of 31 rounds and operates on 64-bit block sizes. It supports two lengths of key, 80 or 128 bits, where 80-bit key is recommended to be used. Each of the 31 rounds is applied on three stages. The first stage is AddRoundKey, the second is SBoxLayer and the third stage is the bit permutation pLayer. The most interesting for us, is the second stage where the starring role belongs to the S-boxes.

The non-linear layer (SBoxLayer) uses a single 4-bit input and 4-bit output (4×4-bit) S-box. Also the choice of 4×4-bit S-box is a direct consequence of authors' pursuit for hardware efficiency, where implementation of such an S-box typically being much more compact and requires less resources than that of an 8×8-bit S-box. A 4-bit S-box requires less than a quarter of the hardware area (expressed in GEs - gate equivalences) of an 8-bit S-box. From cryptographic point of view, 4-bit S-boxes have to be selected very carefully because they are weaker than 8-bit S-boxes.

PRESENT S-boxes are derived as a result of an exhaustive search of all 16! bijective 4-bit S-boxes. All S-boxes found in this way that fulfilled additional criteria for optimality have been analyzed in relation to linear equivalence. So, there are only 16 different non-equivalent classes [4]. All the S-box members in these classes are optimal S-boxes with respect to linear and differential properties. Also the authors notified that these S-boxes are also optimal with respect to the algebraic degree or resistance against algebraic attacks. A slightly more general classification of all 4-bit S-boxes was given by Saarinen in [5].

Instead of an exhaustive search of all 16! bijections of 16 elements as it was done for the design of PRESENT, in this work we offer a compact, fast and elegant methodology for construction of cryptographically strong S-boxes by using quasigroups of order 4. Our goal is to give cryptographers an iterative tool for designing cryptographically strong S-boxes (in this paper, we denote them as Q-S-boxes since their construction is done by quasigroups) for future designs in the symmetric lightweight cryptography. Our methodology enables someone to work basically with several different strong S-boxes iteratively reusing only one hardware circuit and just changing a few parameters.

The structure of the paper is the following. In Section II, we give a brief mathematical description of the quasigroups and quasigroup string transformations. In Section III, we present the linear and differential characteristics of S-boxes, and conditions of one S-box to be optimal. We give the representation of quasigroups as vector valued Boolean functions in the Section IV. In Section V, we show a method for construction of cryptographic 4×4-bit Q-S-boxes, and, in Section VI, we give a conclusion and future work.

II. Preliminaries - Quasigroups and Quasigroup String Transformations

In this section, we give a brief mathematical introduction in the area of quasigroups and quasigroup string transformations. A more detailed explanation about quasigroups and their applications can be found in [6][7][8][9][10].

Let $(Q, *)$ be a finite binary groupoid, i.e., an algebra with one binary operation $*$ on the non-empty set Q and $a, b \in Q$.

Definition 1: A finite binary groupoid $(Q, *)$ is called a quasigroup if for all ordered pairs $(a, b) \in Q^2$ there exist unique solutions $x, y \in Q$ to the equations $x * a = b$ and $a * y = b$.

This implies the cancelation laws for quasigroup i.e., $x * a = x' * a \Longrightarrow x = x'$ and $a * y = a * y' \Longrightarrow y = y'$.

Any quasigroup is possible to be presented as a multiplication table known as Cayley table. Quasigroups are closely related to Latin squares. Removing the topmost row and the leftmost column of the Cayley table of a quasigroup, results in a Latin square. A Latin square is an arrangement of n symbols in a $n \times n$ matrix such that no row and no column contains any of the symbols twice.

The order of a quasigroup $(Q, *)$ is the cardinality $|Q|$ of the non-empty set Q. The set of all quasigroups of order n is denoted by $\mathbb{Q}_n$.

In what follows, we will work with finite quasigroups of order 4 only. That means that our design of S-boxes uses $|Q|^2 = 4^2$, 2-bit words of internal memory for storing the quasigroup. We will need 4 bytes (4B) of internal memory for storing the quasigroup, which is acceptable amount if we want to implement it in some lightweight designs.

Example 1: Let $Q = \{0, 1, 2, 3\}$. A quasigroup $(Q, *)$ of order 4 has the following Cayley table:

$*$	0	1	2	3
0	0	1	3	2
1	1	0	2	3
2	2	3	0	1
3	3	2	1	0

For our method for construction of optimal S-boxes described in Section V we will use the notion of quasigroup string transformation "e-transformation" as defined in [11].

Let Q be a set of elements ($|Q| \geq 2$) and let we denote by $Q^r = \{a_0, a_1, \ldots, a_{r-1} | a_i \in Q, r \geq 2\}$ the set of all finite strings with elements of Q.

Assuming that $(Q, *)$ is a given quasigroup, for a fixed element $l \in Q$, called leader, the transformation $e_l : Q^r \to Q^r$ is as follow:

$$e_l(a_0, a_1, \ldots, a_{r-1}) = (b_0, b_1, \ldots, b_{r-1}) \Leftrightarrow$$

$$\begin{cases} b_0 &= l * a_0 \\ b_i &= b_{i-1} * a_i, \quad 1 \leq i \leq r-1 \end{cases} \tag{1}$$

This e-transformation is called elementary quasigroup string transformation [12]. It transforms bijectively a given string with length r to other resulting string with the same length r.

Graphical representation of the transformation is shown in Figure 1.

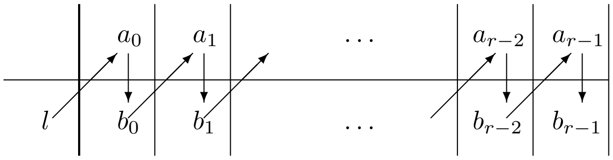

Figure 1. Graphical representation of e-transformation.

If we have some initial sequence of leaders $l_0, l_1, \ldots, l_{k-1}$, then we can make a composition of transformations by applying consecutive e-transformations.

Composite transformation obtained as a composition of e-transformations only, is defined by

$$E(l_0, l_1, \ldots, l_{k-1}) := e_{l_0}(e_{l_1} \ldots (e_{l_{k-1}}(a_0, a_1, \ldots, a_{r-1}))). \tag{2}$$

III. S-boxes and Their Properties

S-boxes have a fundamental role for the security of almost all modern block ciphers because they are usually the only non-linear part in the block ciphers. They have to be selected very carefully to make the cipher resistant against various kinds of attacks.

There is no formal definition for S-boxes. In general, they are defined as a lookup tables or vector valued Boolean functions or Boolean maps.

A Boolean function of n variables is a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, where $\mathbb{F}_2$ is a Galois field with two elements. A Boolean map (or vector valued Boolean function) is a map $f : \mathbb{F}_2^n \to \mathbb{F}_2^q$.

For two vectors $u, v \in \mathbb{F}_2^n$, where $u = (u_0, u_1, \ldots, u_{n-1})$ and $v = (v_0, v_1, \ldots, v_{n-1})$ the *canonical dot product* can be written as

$$u \cdot v = \sum_{i=0}^{n-1} u_i v_i. \tag{3}$$

Given an S-box mapping n bits to q bits, we present it as a Boolean map $S : \mathbb{F}_2^n \to \mathbb{F}_2^q$.

Linearity of an S-box represents a measure for the resistance against linear cryptanalysis. Therefore, the smaller the linearity of an S-box is, the more secure the S-box is against linear cryptanalysis. According to all of the mathematical results given in [13] about linearity of Boolean functions and Boolean maps, we can define *linearity of an S-box*, S as:

$$Lin(S) = max\{\frac{1}{2^{2n}} S^W(u, v)^2 \mid u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^q, (u, v) \neq 0\} \tag{4}$$

where u is the part of input, and v is the part of output values of the S-box.

The Walsh spectrum S^W for this S-box is calculated by:

$$S^W(u,v) = \sum_{x \in \mathbb{F}_2^n} (-1)^{u \cdot x + v \cdot S(x)} \quad (5)$$

Theoretically, it is proven that $Lin(S) \geq \frac{1}{2^n}$ [14].

One of the most important properties for S-boxes is so-called *differential potential* of an S-box. It is used in measuring the resistance of the cryptographic primitives that use that S-box against differential cryptanalysis. The differential potential of an S-box S is defined in [13] as:

$$Diff(S) = max\{\frac{1}{2^n}\Delta_S(u,v) \mid u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^q, \quad \text{and} \quad (u,v) \neq 0\} \quad (6)$$

where

$$\Delta_S(u,v) = |\{x \in \mathbb{F}_2^n \mid S(x \oplus u) = S(x) \oplus v\}| \quad (7)$$

Clearly, it holds for any S-box that Diff(S) $\geq \frac{1}{2^q}$.

A. Optimal 4-bit S-boxes in PRESENT

As we mentioned in the introduction, the used S-boxes in the block cipher PRESENT have been obtained by an exhaustive search of all 16! permutations by checking some optimality criteria for their linearity and their differential potentials. Namely, all generated S-boxes were first presented as a Boolean map $S : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$. Then, using the above formulations about $Lin(S)$ and $Diff(S)$, the optimal set of PRESENT S-boxes was formed by S-boxes that have $Lin(S) = \frac{1}{4}$ and $Diff(S) = \frac{1}{4}$ [4].

More formally, as it is given in [4], the definition of *an optimal S-box* is the following:

Definition 2: Let S be an 4×4-bit S-box with 2^4 input values. If S fulfills the following conditions we call S an optimal S-box:

1) S is a bijection;
2) $Lin(S) = \frac{1}{4}$;
3) $Diff(S) = \frac{1}{4}$.

IV. Quasigroups as Vector Valued Boolean Functions

We will use the representation of finite quasigroups $(Q, *)$, of order n, where $n \geq 2$ and $n = 2^d$ as vector valued Boolean functions. That means that the quasigroup can be presented as a Boolean map: $f : \mathbb{F}_2^{2d} \rightarrow \mathbb{F}_2^d$. For each elements $x, y, z \in Q$ the operation $x * y = z$ is represented by

$$f(x_0, x_1, \ldots, x_{d-1}, y_0, y_1, \ldots, y_{d-1}) = \\ (f_0(x_0, \ldots, x_{d-1}, y_0, \ldots, y_{d-1}), \ldots, f_{d-1}(x_0, \ldots, x_{d-1}, y_0, \ldots, y_{d-1})) \quad (8)$$

where $(x_0, x_1, \ldots, x_{d-1})$ and $(y_0, y_1, \ldots, y_{d-1})$ are the binary representations of x and y respectively, and $f_i : \mathbb{F}_2^{2d} \rightarrow \mathbb{F}_2$, $0 \leq i \leq d-1$ are the corresponding components of f (binary representation of z).

Every Boolean function $f : \mathbb{F}_2^m \rightarrow \mathbb{F}_2$, can be uniquely written in its Algebraic Normal Form (ANF), as a polynomial in m variables over the field $\mathbb{F}_2$ that has degree ≤ 1 in each single variable:

$$f(x_0, x_1, \ldots, x_{m-1}) = \sum_{I \subseteq \{0, \ldots, m-1\}} a_I x^I, \quad (9)$$

where the monomial x^I is the product

$$x^I = \prod_{i \in I} x_i, \quad (10)$$

and $a_I \in \{0, 1\}$.

The ANF has the advantage that we can immediately read off the algebraic degree. Algebraic degree of a Boolean function is a degree of a polynomial obtained with its ANF presentation. Algebraic degree of a Boolean map is a maximal algebraic degree of its component functions. So, the ANFs of the Boolean functions f_i give us information about their algebraic degree and much better about algebraic degree or complexity of the quasigroup $(Q, *)$.

Example 2: Let us take the quasigroup given in Example 1. This quasigroup can be presented as a vector valued Boolean function $f : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^2$ by:

$$f(x_0, x_1, y_0, y_1) = (x_0 + y_0, x_1 + y_0 + x_0 * y_0 + y_1)$$

We see that the algebraic degree of this quasigroup is 2.

According to their algebraic degree quasigroups can be divided in two classes, class of linear quasigroups and class of non-linear quasigroups. The class of linear quasigroups has a maximal algebraic degree 1, and all other quasigroups (which maximal algebraic degree is bigger than 1) belong to the class of non-linear.

Considering the class of quasigroups of order 4, it can be checked that there are 144 linear and 432 non-linear quasigroups, i.e., there are three times more non-linear quasigroups of order 4 [15].

V. Construction of Optimal 4-bit Q-S-boxes

Our goal is to generate 4×4-bit cryptographically strong S-boxes by using quasigroups of order 4. Quasigroups of order 4 themselves are 4×2-bit S-boxes. It is theoretically proven that any inversion mapping for even dimension n in $GF(2^n)$ must has algebraic degree smaller than $n-1$ [16]. It should be noted that criterion for good S-box is to have highest possible algebraic degree. From this perspective, we can conclude that we would search for 4×4-bit S-boxes that have algebraic degree 3 for all output bits.

We will use quasigroup string transformations that transform a given string with length 2 to a resulting string with the same length 2, i.e., that maps 4 bits bijectively to 4 bits (Figure 2).

As it is a case with any iterative application of non-linear Boolean transformations, by consecutive application

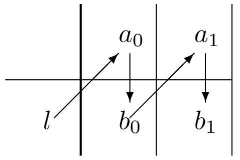

Figure 2. One e-transformation that bijectively transforms 4 bits into 4 bits by a quasigroup of order 4. Here, l, a_0, a_1, b_0 and $b_1 \in \{0, 1, 2, 3\}$.

of e-transformation we will raise the algebraic degree of the produced final bijections. More concretely, as it is shown in Figure 3, we will use one non-linear quasigroup of order 4 and at least 4 e-transformations to reach the desired degree of 3 for all the bits in final output block. Note that every row in Figure 3 starting with a leader l_i is a bijective e-transformation of the pairs of bits from previous row. In such a way, we have a composition of non-linear Boolean bijections producing the final bijection.

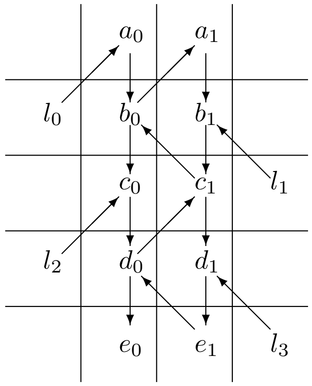

Figure 3. Four e-transformations that bijectively transforms 4 bits into 4 bits by a quasigroup of order 4.

Having one condition satisfied (the algebraic degree is maximal), we have to check further the other conditions from Section III in order to obtain optimal S-boxes, (i.e., optimal Q-S-boxes). The whole algorithm for our methodology is given in Table I.

This algorithm is for generating one Q-S-box from one chosen quasigroup of order 4 from the class of non-linear quasigroups and one combination of input leaders for the e-transformation. We already mentioned that the minimum number of rounds (iterations) for this methodology is 4. Using the described methodology we can generate Q-S-boxes in different ways depending on the number of rounds and the number of leaders that we can choose. In our investigation we choose to work with 2, 4 and 8 different leaders and 4 and 8 rounds, respectively. We found all the Q-S-boxes that fulfill the predetermined criteria to be optimal.

Experiments that are made with 2 leaders and 4 rounds as in the Algorithm 1 showed that there exist optimal Q-S-boxes. There are exactly 6,912 different Q-S-boxes (2^4 possibilities for the leaders $*$ 432 different non-linear quasigroups of order 4) that can be generated in this way, but 1,152 of them belong to the class of optimal. In Table II we give the distribution of differential and linear properties among the 6,912 examined Q-S-boxes.

Table I
CONSTRUCTION OF ONE Q-S-BOX

Algorithm 1. An iterative method for construction of Q-S-boxes

Step 1	Take one quasigroup of order 4 from the class of non-linear;
Step 2	Input the number of rounds;
Step 3	Input the leaders. Usually, their number is the same as the number of rounds;
Step 4	Generate all possible input blocks of 4 bits in the lexicographic ordering (they are 2^4);
Step 5	Take input blocks one by one, and for each of them:
Step 5.1	Apply e-transformation with leader l on the input block;
Step 5.2	Reverse the result from above and apply e-transformation with other leader l again;
Step 5.3	Continue this routine as many times as there is a number of rounds;
Step 5.4	Save the 4-bit result from the last round;
Step 6	At the end concatenate all saved results which generate permutation of order 16 or 4 × 4-bit Q-S-box;
Step 7	Investigate predetermined criteria;
Step 7.1	If the Q-S-box satisfies criteria, put it in the set of optimal S-boxes;
Step 7.2	If not, go to Step 3;
Step 8	Analyze the optimal set of newly obtained Q-S-boxes;

Table II
DISTRIBUTION OF THE 6,912 Q-S-BOXES IN RELATION TO DC AND LC WHERE THE ITERATIVE METHOD WITH 2 LEADERS IS USED

LC →	Lin(S)=1/4		Lin(S)=9/16		Lin(S)=1	
DC ↓	n	%	n	%	n	%
Diff(S)=1/4	1,152	16.7	0	0.00	0	0.00
Diff(S)=3/8	0	0.00	768	11.1	384	5.6
Diff(S)=1/2	0	0.00	2,304	33.3	768	11.1
Diff(S)=5/8	0	0.00	0	0.00	0	0.00
Diff(S)=3/4	0	0.00	0	0.00	0	0.00
Diff(S)=1	0	0.00	0	0.00	1,536	22.2

From the Table II can be seen that in total 1,152 Q-S-boxes have $Diff(S)$ =1/4 and $Lin(S)$ =1/4. They are 16.7% of all Q-S-boxes, that have a differential bound 1/4 and linear bound 1/4 and belong to the class of optimal S-boxes. All of these Q-S-boxes have maximal algebraic degree of all output bits 3, but some of the output bits may still have one non-linear monomial of degree 2, and therefore, this output bit depends only linearly on 2 input bits. This can be crucial when determining the number of secure rounds; final rounds can be peeled off using such properties. So, the number of Q-S-boxes that satisfy all of the output bits to have algebraic degree 3 is 128. One representative of them is given in Table III.

Table III
ONE OF THE 128 Q-S-BOXES GIVEN IN ITS HEXADECIMAL NOTATION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	C	1	2	E	F	9	3	4	8	0	A	B	7	D	6	5

Table IV
DISTRIBUTION OF THE Q-S-BOXES IN RELATION TO DC AND LC WHERE THE ITERATIVE METHOD WITH 4 LEADERS IS USED

LC →	Lin(S)=1/4		Lin(S)=9/16		Lin(S)=1	
DC ↓	n	%	n	%	n	%
Diff(S)=1/4	9,216	8.33	0	0.00	0	0.00
Diff(S)=3/8	3,072	2.78	12,288	11.11	6,144	5.56
Diff(S)=1/2	3,072	2.78	36,864	33.33	15,360	13.89
Diff(S)=5/8	0	0.00	0	0.00	0	0.00
Diff(S)=3/4	0	0.00	0	0.00	0	0.00
Diff(S)=1	0	0.00	0	0.00	24,576	22.22

We made experiments with 4 leaders and with the same number of rounds (one leader in each round). We produced 110,592 different Q-S-boxes (2^8 possibilities for the leaders $*$ 432 non-linear quasigroups of order 4), from which 9,216 fulfilled the criteria for optimality. In Table IV, we give the distribution of differential and linear properties among the 110,592 examined Q-S-boxes. There 8.3% of all Q-S-boxes have a differential bound 1/4 and linear bound 1/4 and belong to the class of optimal S-boxes. All of these Q-S-boxes have maximal algebraic degree of all output bits 3, but some of them still have one output bit of degree 2. The number of Q-S-boxes that satisfy, all of the output bits to have algebraic degree 3 in this case is 1,024. One representative of them is given in Table V.

Table V
ONE OF THE 1,024 Q-S-BOXES GIVEN IN ITS HEXADECIMAL NOTATION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	D	9	F	C	B	5	7	6	3	8	E	2	0	1	4	A

Table VI
DISTRIBUTION OF THE Q-S-BOXES IN RELATION TO DC AND LC WHERE THE ITERATIVE METHOD WITH 8 LEADERS IS USED

LC →	Lin(S)=1/4		Lin(S)=9/16		Lin(S)=1	
DC ↓	n	%	n	%	n	%
Diff(S)=1/4	756,480	2.67	280,320	0.99	0	0.00
Diff(S)=3/8	1,084,416	3.83	9,273,666	32.75	121,278	0.43
Diff(S)=1/2	63,744	0.23	8,394,186	29.65	2,590,518	9.15
Diff(S)=5/8	0	0.00	468,480	1.65	254,208	0.90
Diff(S)=3/4	0	0.00	224,244	0.79	87,564	0.31
Diff(S)=1	0	0.00	0	0.00	4,712,448	16.65

We made also experiments with 8 leaders and 8 rounds. In this case the number of generated Q-S-boxes significantly increased. We produced 28,311,552 different Q-S-boxes, from which 756,480 fulfilled the criteria for optimality. Distribution of these examined Q-S-boxes in relation to Differential Cryptanalysis (rows) and Linear Cryptanalysis (columns) is given in the Table VI.

The number of Q-S-boxes that satisfy, all of the output bits to have algebraic degree 3 is 331,264. One representative of them is given in Table VII.

Table VII
ONE OF THE 331,264 Q-S-BOXES GIVEN IN ITS HEXADECIMAL NOTATION

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	5	E	6	D	7	4	2	A	8	C	0	9	1	B	F	3

Apparently, by increasing the number of leaders and rounds, the number of optimal Q-S-boxes also increases. With this methodology we can generate all of the optimal S-boxes, which are already found for PRESENT. The concrete values for the leaders, the number of used leaders, and which non-linear quasigroup of order 4 to be used, in order to produce a PRESENT S-box, can be found by using some of the modern symbolic algebra systems such as Magma [17] or SAGE [18].

At the end of this section, we want to note that since we use non-linear quasigroups of order 4, the iterative procedure in Algorithm 1 has much bigger probability to produce S-boxes with optimal criteria than a random search through the set of all 16! permutations of order 16.

VI. CONCLUSION AND FUTURE WORK

In this paper, we gave a simple iterative method for producing cryptographically optimal 4×4-bit S-boxes by quasigroups of order 4, using the concept of quasigroup string transformations. We have given also the summary of our extensive experimental results. With this method and right choice of input parameters, we can generate the same optimal S-boxes like one in the lightweight block cipher PRESENT.

As a future work we emphasize the generality of our approach, and its extensibility to permutations of higher order. Thus a natural extension of our work would be to produce cryptographically strong 6×4-bit, 8×8-bit and other types of S-boxes using again iteratively just quasigroups of order 4. First of all, we should obtain how many rounds and leaders are necessary to produce Q-S-boxes with the same quality like known one, and then to see which of them belong to the class of optimal ones regarding to linear and differential characteristics of S-boxes.

REFERENCES

[1] J. Daemen and V. Rijmen. AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999. [retrieved: July, 2012]. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/index.html

[2] A. Bogdanov, L. R. Knudsen, G. Le, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," in *The Proceedings of CHES 2007*. Springer-Verlag, 2007, pp. 450–466.

[3] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, “A Survey of Lightweight-Cryptography Implementations,” in *IEEE Design and Test*, vol. 24, no. 6. IEEE Computer Society Press, November, 2007, pp. 522–533.

[4] G. Leander and A. Poschmann, “On the Classification of 4 Bit S-Boxes,” in *Proceedings of the 1st International Workshop on Arithmetic of Finite Fields*. Springer-Verlag, 2007, pp. 159–176.

[5] M.-J. O. Saarinen, “Cryptographic Analysis of all 4x4-bit S-boxes,” in *Proceedings of the 18th International Conference on Selected Areas in Cryptography*, ser. SAC’11. Springer-Verlag, 2012, pp. 118–133.

[6] V. D. Belousov, *Osnovi teorii kvazigrupp i lupp*. Nauka, Moskva, 1967.

[7] J. Denes and A. D. Keedwell, *Latin squares: New developments in the theory and applications*. Elsevier science publisher, 1991.

[8] ——, *Latin squares and their applications*. Academic Press Inc, December 1974.

[9] S. Markovski, D. Gligoroski, and L. Kocarev, “Unbiased Random Sequences from Quasigroup String Transformations,” in *Lecture Notes in Computer Science*, ser. FSE, vol. 3557. Springer-Verlag, 2005, pp. 163–180.

[10] J. D. H. Smith, *An Introduction to Quasigroups and Their Representations*. Chapman and Hall/CRC, 2007.

[11] S. Markovski, “Quasigroup String Processing and Applications in Cryptography,” in *The Proceedings of the 1st MII Conference*, 2003, pp. 278–290.

[12] A. Mileva, “Cryptographic Primitives with Quasigroup Transformations,” Ph.D. dissertation, University Ss. Cyril and Methodius, Skopje, Macedonia, 2010.

[13] K. Pommering, *Fourier Analysis of Boolean Maps*. A Tutorial, Mainz, 2005.

[14] F. Chabaud and S. Vaudenay, “Links Between Differential and Linear Cryptanalysis,” in *Advances in Cryptology - EUROCRYPT’94, Workshop on the Theory and Application of Cryptographic Techniques*, vol. 950. Springer-Verlag, 1995, pp. 356–365.

[15] D. Gligoroski, V. Dimitrova, and S. Markovski, “Quasigroups as Boolean Functions, Their Equation Systems and Gröbner Bases,” in *Grobner Bases, Coding, and Cryptography*. Springer-Verlag, 2009, pp. 415–420.

[16] K. Nyberg, “Differentially Uniform Mappings for Cryptography,” in *Advances in Cryptology - EUROCRYPT’93, Workshop on the Theory and Application of Cryptographic Techniques*, vol. 765. Springer-Verlag, 1994, pp. 55–64.

[17] W. Bosma, J. Cannon, and C. Playoust, “The Magma algebra system. I. The user language,” in *Journal of Symbolic Computation*, vol. 24, 1997, pp. 235–265.

[18] W. Stein, *Sage Mathematics Software (Version 5.0.1)*, The Sage Development Team, [retrieved: July, 2012][Online]. Available: http://www.sagemath.org.