Cryptographic Primitives with Quasigroup Transformations

by Aleksandra Mileva

Dissertation
Typeset using LaTeX2e
Printed by
Cover design by Aleksandra Mileva

A catalogue record is available
from the Faculty of Natural Science Library
ISBN:

УНИВЕРЗИТЕТ “СВ. КИРИЛ И МЕТОДИЈ” СКОПЈЕ
ПМФ

Preface

This thesis is the final result of three years of research done in the Institute of Informatics at "Ss Cyril and Methodius" University in Republic of Macedonia. First of all, I would like to thank my supervisor, professor Smile Markovski, for his support, help, tolerance, understanding, and flexibility as much as a supervisor can give. I am grateful for the trust that was given to me in the beginning of my research and the freedom to follow my own path at the later stages. I gained expertise in writing papers under his guidance and he was always willing to share his experiences and teach me the tricks of the trade.

I owe a lot to professor Danilo Gligoroski, in whom I always had a interesting and inspirational interlocutor with cryptographic and quasigroup expertise. His remarks always hit the target and motivated me to put additional effort and finish this thesis in the right way. The quality of my research and my expertise as a researcher would never be on this level if it were not for him.

Many results presented in this thesis are a product of joint work. Vesna Dimitrova was involved in most of it as one of my closest co-workers and a great friend. In the past three years, she always shared her ideas, open to criticism, and promptly sharing her own considering my inventions as well.

One year ago, Simona Samardziski joined our team. Almost immediately we began collaborating, which resulted in some interesting research. She did a lots of programming for our hash function NaSHA.

I would like to thank professor Jasen Markovski, in helping this thesis to be finished in English language.

Endless amount of gratitude goes to Ile, for his love, understanding, encouraging and steadfast support. My two children, Iva and Nikola, make my life interesting, eventful, warm, and full of love and sunshine, even on the cloudiest of days.

Finally, I want to express my deepest gratitude and appreciation for my parents Violeta and Nikola, and my brother Kiril, who have always been there to support me. Without their love, support, compassion, selfless sacrifice, and vision I would have never become the person that I am.

Aleksandra Mileva Štip, June 30th, 2009

Summary

Cryptographic Primitives with Quasigroup Transformations

Cryptology is the science of secret communication, which consists of two complementary disciplines: cryptography and cryptanalysis. Cryptography is dealing with design and development of new primitives, algorithms and schemas for data enciphering and deciphering. For many centuries cryptographic technics have been applied in protection of secrecy and authentication in diplomatic, political and military correspondences and communications. Cryptanalysis is dealing with different attacks on cryptographic schemas and algorithms, with purpose to retrieve the hidden information and the same later to use, modify, forge etc. There is a big interconnection between these two disciplines. Cryptographer who design a new algorithm, must evaluate its security for all known cryptanalytic attacks and technics, if he wants its algorithm to be practical and useful. For future users to have confidence in a new algorithm and to use it, a long-time analysis and evaluation of its security from bigger group of cryptanalysts is needed, without any resulting weakness.

Quasigroups are very suitable for application in cryptography, because of their structure, features and big number. One of the problems is which quasigroup is suitable to choose for using, concerning what preconditions quasigroup must fulfill. Several classification and separations of quasigroups are made for that purpose, with possibility for more. Quasigroups are used for definition of a quasigroup transformations. Sequences produced by quasigroup transformations are also examined and their analysis shows that they can be used as building elements of different cryptographic primitives.

Cryptology as a science is developing with huge speed, because a new cryptographic schemas and algorithms, a new design strategies, a new fields of application, a new requirements and a new attacks are appearing, continuously. Appearance of new successful attacks and discovering weaknesses in declared standards, as well as requirements for augmented key and blocks

lengths, induce the necessity of a new approaches in design and security evaluation, deployment of new building elements, modification of existing algorithms and schemas etc.

The thesis investigates several issues: (1) What properties should have some quasigroup, so it can be used as non-linear building block in cryptographic primitives and it can contributed to the defence of linear and differential attacks? (2) How to generate and how to compute fast operation of huge quasigroups? (3) What kind of features have huge quasigroups obtained by new construction method? (4) In which way to use huge quasigroups as building blocks of cryptographic primitives?

The contents of the thesis is as follows. First, we introduce the theory of quasigroups and quasigroup transformations. We introduce a new way of computing the number of n-ary quasigroups, with which we obtained the number of ternary quasigroups of order 4 divided in 12 isotopy classes. We introduce some new kind of quasigroup transformations and we represent a prop ratio tables and correlation matrices of quasigroups of small order and some quasigroup transformations. This induce new classification of quasigroups according to their prop ratio tables and correlation matrices. We use the notation of the shapeless quasigroup and we introduce a notation of a perfect quasigroup. Then, we investigate different ways of producing huge quasigroups and suggest a new way of computing a huge quasigroup operation with applying Extended Feistel networks. This approach deploy Feistel network with special preconditions as an orthomorphism of a group. We analyze quasigroups obtained by Extended Feistel networks and show in which cases they are suitable for cryptographic needs. Next, we give a survey of quasigroup based hash functions, stream and block ciphers, public-key algorithms etc. We design two new cryptographic primitives which are using huge quasigroups as building blocks. We introduce NaSHA family of hash functions, with our implementation that is a candidate for NIST competition for SHA-3 standard and we show how by using Extended Feistel network we can apply different huge quasigroups for processing single message block and even how used quasigroups can depend of processed block. This features make harder the cryptanalyst job. We introduce Alexsmile family of block ciphers and give one implementation for 128-bit block size and key size of 128, 192 and 256 bits.

Contents

Chapter 1

Quasigroups and quasigroup transformations

In this chapter first we present a mathematical background, terminology and notation of n-ary quasigroups and quasigroup transformations. We introduce new types of quasigroup transformations, witch are used later for building cryptographic primitives. Also, we present a new method for computing the number of n-ary quasigroups of small order. We give analysis of prop ratio tables and correlation matrices of quasigroups of order 4 and same for several of their quasigroup transformations on strings of length 2. This analysis have produced some additional and confirmed existing partitioning of quasigroups of order 4.

We will examine the problem of which quasigroup is suitable to be chosen for using in cryptographic primitives, concerning what preconditions the quasigroup must fulfill. We will show that even quasigroups with low order are very suitable for application in cryptography. This is specially true for huge quasigroups because of their structure, features and big number. Good mathematical background for quasigroups you can find in [3, 19, 20, 63, 129].

1.1 Quasigroups - mathematical background

Definition 1 A **quasigroup** $(Q, \circ)$ is a set Q of elements with a binary operation $\circ$ with the following properties:
1. For all $a, b \in Q$, $a \circ b \in Q$ (that is, Q is a groupoid)
2. For all $a, b \in Q$, there exist unique $x, y \in Q$, so that $a \circ x = b$ and $y \circ a = b$. □

In other words, the equations $a \circ x = b$ and $y \circ x = b$ for any given $a, b \in Q$ have unique solutions x, y. So, each element will appear exactly once in each row and exactly once in each column of the multiplication table of $(Q, \circ)$. This means that every row and every column is a permutation of Q. To every finite quasigroup with n elements $(Q, \circ)$, given by its Cayley table, an equivalent combinatorial structure n by n Latin square can be associated, consisting of the matrix formed by the interior of the table (an n by n Latin square is made up of n distinct elements, each of which appears exactly once in each row and exactly once in each column). Examples of quasigroups are: $(\mathbb{Z}, -)$, $(\mathbb{Q}\backslash\{0\}, \div)$, $(\mathbb{R}\backslash\{0\}, \div)$ etc.

For all $a \in Q$ we can define two mappings R_a and L_a of Q into itself by

$$R_a(x) = x \circ a$$

$$L_a(x) = a \circ x$$

Then $(Q, \circ)$ is a quasigroup if and only if R_a and L_a are bijections for each $a \in Q$. The mapping R_a is known as *right multiplication* by a and the mapping L_a is known as *left multiplication* by a.

Definition 2 A groupoid $(G, \circ)$ is a *cancellative groupoid*, if for every $c, x, y \in G$ hold

$$c \circ x = c \circ y \Rightarrow x = y \quad \text{and} \quad x \circ c = y \circ c \Rightarrow x = y$$

Definition 3 A groupoid $(G, \circ)$ is a *solvable groupoid*, if for every $a, b \in G$ the equations $a \circ x = b$ and $y \circ a = b$ have solutions $x, y \in Q$. □

Proposition 1 *The following statements for a finite groupoid $(Q, \circ)$ are equivalent:*
(a) $(Q, \circ)$ is a quasigroup.
(b) $(Q, \circ)$ is a cancellative groupoid.
(c) $(Q, \circ)$ is a solvable groupoid. □

Proof The proof follows from the Proposition 2. ■

From the Definition 1 follows that every group is a quasigroup. Quasigroups differ from groups, mainly, in which they don't need to be associative, so they are sometimes considered to be "non-associative groups". A quasigroups with identity element are called *loops*.

Definition 4 A subset P of a quasigroup $(Q, \circ)$ is a *subquasigroup* of Q, if it is closed under operation $\circ$. □

Given a quasigroup $(Q, \circ)$, five operations $/, \backslash, \cdot, //, \backslash\backslash$ on the set Q can be derived by:

$$\begin{gathered} x/y = z \Longleftrightarrow x = z \circ y, \text{ right division} \\ x\backslash y = z \Longleftrightarrow x \circ z = y, \text{ left division} \\ x \cdot y = z \Longleftrightarrow y \circ x = z, \text{ opposite multiplication} \\ x//y = z \Longleftrightarrow y/x = z \Longleftrightarrow y = z \circ x, \text{ opposite right division} \\ x\backslash\backslash y = z \Longleftrightarrow y\backslash x = z \Longleftrightarrow y \circ z = x, \text{ opposite left division} \end{gathered}$$

The set $Par(\circ) = \{\circ, /, \backslash, \cdot, //, \backslash\backslash\}$ is said to be the set of *parastrophes* of quasigroup operation $\circ$. $|Par(\circ)| \leqslant 6$, i.e. some of the parastrophes may coincide between themselves. For each $g \in Par(f)$, (Q, g) is a quasigroup too, known as the *conjugate* of Q and $Par(f) = Par(g)$ (see [131], [121]). Now we can give another definition of quasigroup.

Definition 5 An **algebraic quasigroup** $(Q, \circ, \backslash, /)$ is a type $(2, 2, 2)$ algebra satisfying the identities:

$$\begin{aligned} y &= x \circ (x\backslash y) \\ y &= x\backslash(x \circ y) \\ y &= (y/x) \circ x \\ y &= (y \circ x)/x \end{aligned}$$

□

Since there is no any difference between quasigroups $(Q, \circ)$ and algebraic quasigroups $(Q, \circ, \backslash, /)$ when Q is finite, and we are dealing mainly with finite sets, we will use the name quasigroups for both of them.

Example 1 Let $Q = \mathbb{Z}_4 = \{0, 1, 2, 3\}$ and let $\circ$ be as shown in Table 1.1.

$\circ$	0	1	2	3
0	2	1	0	3
1	1	2	3	0
2	3	0	2	1
3	0	3	1	2

$/$	0	1	2	3
0	3	2	0	1
1	1	0	3	2
2	0	1	2	3
3	2	3	1	0

$\backslash$	0	1	2	3
0	2	1	0	3
1	3	0	1	2
2	1	3	2	0
3	0	2	3	1

$\cdot$	0	1	2	3
0	2	1	3	0
1	1	2	0	3
2	0	3	2	1
3	3	0	1	2

$//$	0	1	2	3
0	3	1	0	2
1	2	0	1	3
2	0	3	2	1
3	1	2	3	0

$\backslash\backslash$	0	1	2	3
0	2	3	1	0
1	1	0	3	2
2	0	1	2	3
3	3	2	0	1

Table 1.1: Example of quasigroup of order 4 and its conjugates

Then $(Q, \circ)$ is a quasigroup because the interior of its Cayley table is a Latin square. Notice that $(Q, \circ)$ is non-associative, non-commutative, non-idempotent and without left nor right identity. Also conjugates of Q are given. □

In the following sequel, we will first explain the terminology which will be used in this thesis.

A quasigroup $(Q, \circ)$ is said to be *idempotent* if it satisfies the identity

$$x \circ x = x.$$

A quasigroup $(Q, \circ)$ is said to be a *Schroeder quasigroup* (see [65]) if it satisfies the identity

$$(x \circ y) \circ (y \circ x) = x.$$

A quasigroup $(Q, \circ)$ is said to be a *Stein quasigroup* (see [65]) if it satisfies the identity

$$x \circ (x \circ y) = y \circ x.$$

A quasigroup $(Q, \circ)$ is said to be a *semisymmetric quasigroup* if it satisfies the identity

$$(x \circ y) \circ x = y.$$

Commutative and semisymmetric quasigroup is said to be *totally symmetric* and for it $x \circ y = x \backslash y = y/x$ for all $x, y \in Q$. An idempotent totally symmetric quasigroups are also referred as a *Steiner quasigroups*, since each such quasigroup gives rise to a Steiner triple system and conversely.

A quasigroup $(Q, \circ)$ is said to be a *totally anti-symmetric quasigroup* if for all $x, y, c \in Q$ the following two equations are true:

$$(c \circ x) \circ y = (c \circ y) \circ x \Rightarrow x = y$$

$$x \circ y = y \circ x \Rightarrow x = y$$

A quasigroup $(Q, \circ)$ is said to be a *(r, s, t)-inverse quasigroup* if there exists a permutation J on Q and integers r, s, and t such that, for all $x, y \in Q$, the following equation is true:

$$J^r(x \circ y) \circ J^s x = J^t y.$$

In the special case when $r = t = 0$ and $s = 1$ the quasigroup is *crossed inverse* or *CI-quasigroup.*

A *transversal* of a Latin square of order n is a set of n cells, one in each row, one in each column and such that no two of the cells contain the same symbol.

Definition 6 Two quasigroups $(Q, \circ)$ and $(Q, \cdot)$ on the same set Q are said to be **orthogonal** if for any u and v in Q, there exist a unique pair of elements x and y of Q such that $x \circ y = u$ and $x \cdot y = v$. □

In particular, if $(Q, \circ)$ and $(Q, \cdot)$ are orthogonal and x and y run through all elements of Q, the ordered pairs $(x \circ y, x \cdot y)$ run through all ordered pairs of elements of Q. Moreover a set $\{(Q, \circ_i) \mid i = 1 \dots t,\ t \geqslant 2\}$ of quasigroups of order n is orthogonal if any two distinct quasigroups are orthogonal. Such a set of pairwise orthogonal quasigroups is said to be a set of *mutually orthogonal quasigroups*, or more familiar when we speak about Latin squares - a set of *mutually orthogonal Latin squares (MOLS).* The maximum possible number of elements of these sets is $n-1$ and if we have a set of $n-1$ MOLS of order n, the set is said to be *complete.* Good background for MOLS, with their theory, application and construction, is given in [63].

From a given quasigroup $(Q, \circ)$ with transpose of its multiplication table, one can form a new quasigroup $(Q, \cdot)$, called the *transpose* of $(Q, \circ)$ $(x \cdot y = y \circ x)$. If a quasigroup $(Q, \circ)$ is orthogonal to its transpose, than $(Q, \circ)$ is said to be *self orthogonal.* Clearly, it is not possible for a commutative quasigroup to be self orthogonal and for two commutative quasigroups to be orthogonal. For commutative quasigroups of order n, there are at most $n(n+1)/2$ different ordered pairs, and if we have exactly $n(n+1)/2$ different ordered pairs, commutative quasigroups $(Q, \circ)$ and $(Q, \cdot)$ are said to be *perpendicular.*

1.1.1 Quasigroup isotopism, paratopism and isomorphism

Definition 7 Let $(Q_1, \circ)$ and $(Q_2, *)$ be two quasigroups. Q_1 *is homotopic to* Q_2 if there are maps $\alpha, \beta, \gamma : Q_1 \to Q_2$ so that $\alpha(x \circ y) = \beta(x) * \gamma(y)$ for all $x, y \in Q_1$. The ordered triple (α, β, γ) is called an *homotopism* or *homotopy.* □

The homotopy (α, α, α) is called a *homomorphism.*

Definition 8 Let $(Q_1, \circ)$ and $(Q_2, *)$ be two quasigroups. Q_1 *is isotopic to* Q_2 if there are bijections $\alpha, \beta, \gamma : Q_1 \to Q_2$ so that $\alpha(x \circ y) = \beta(x) * \gamma(y)$ for all $x, y \in Q_1$. The ordered triple (α, β, γ) is called an *isotopism* or *isotopy.*□

The isotopy (α, α, α) is called an *isomorphism.* An isotopy (α, β, γ) with equal domain and codomain Q is called an *autotopy.* An autotopy (α, β, γ) is said to be *principal* if its first component α is the identity map or id_Q on Q. Each isotopy (α, β, γ) factorizes as the product $(\alpha, \beta, \gamma) = (id_Q, \beta\alpha^{-1}, \gamma\alpha^{-1})(\alpha, \alpha, \alpha)$ of a principal isotopy and an isomorphism. An autotopy (α, α, α) is called an *automorphism.*

Example 2 We examine quasigroup $(Q, \circ)$ from Example 1. Let $\alpha, \beta, \gamma : Q \to Q$ be bijection defined by:

$$\alpha : \begin{pmatrix} 0123 \\ 3210 \end{pmatrix}, \ \beta : \begin{pmatrix} 0123 \\ 2301 \end{pmatrix}, \ \gamma : \begin{pmatrix} 0123 \\ 1023 \end{pmatrix}. \tag{1.1}$$

□

Then the quasigroup $(Q, *)$ defined by $x * y = \alpha^{-1}(\beta(x) \circ \gamma(y))$ is isotopic to $(Q, \circ)$ (Table 1.2).

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

Table 1.2: One isotopic quasigroup to $(Q, \circ)$ with isotopy (α, β, γ)

The relation “is isotopic to” is an equivalence relation in the set of all quasigroups of order r. The equivalence classes are called classes of isotopism or isotopy classes.

A combination of a conjugacy and an isotopism is called a *paratopism* or paratopy. The relation “is paratopic to” is also an equivalence relation, and the equivalence classes are called *paratopy classes, main classes* or *species.*

1.1.2 n-ary quasigroups

An n-ary groupoid $(n \geqslant 1)$ is an algebra (Q, f) on a nonempty set Q as its universe and with one n-ary operation $f : Q^n \to Q$. We use the definition for n-ary quasigroup from Belousov [4].

Definition 9 An n-ary groupoid (Q, f) is said to be an *n-ary quasigroup* (of order $|Q|$) if any n elements of the $a_1, a_2, \ldots, a_{n+1} \in Q$, satisfying the equality

$$f(a_1, a_2, \ldots, a_n) = a_{n+1},$$

uniquely specifies the remaining one. □

2-ary quasigroups, 3-ary quasigroups and 4-ary quasigroups are also known as binary, ternary and quaternary quasigroups, respectively. When we say only quasigroups, we mean binary quasigroups.

Definition 10 An n-ary groupoid is said to be a *cancellative n-ary groupoid* if it satisfies the cancellation law

$$f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)=f(a_1,\dots,a_i,y,a_{i+2},\dots,a_n)\Rightarrow x=y$$

for each $i=0,\dots,n-1$ and every $a_j\in Q$. □

Definition 11 An n-ary groupoid is said to be a *solvable n-ary groupoid* if the equation $f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)=a_{n+1}$ has solution x for each $i=0,\dots,n-1$ and every $a_j\in Q$. □

The definition of an n-ary quasigroup immediately implies the following.

Lemma 1 *Let (Q,f) be a finite n-ary quasigroup and let the mapping $\varphi:Q\to Q$ be defined by $\varphi(x)=f(a_1,\dots,a_i,x,a_{i+2},\ \dots,a_n)$. Then φ is a permutation on Q.* □

Here we consider only finite n-ary quasigroups (Q,f), i.e. Q are a finite sets, and in this case we have the next property.

Proposition 2 *The following statements for a finite n-ary groupoid (Q,f) are equivalent:*
(a) (Q,f) is an n-ary quasigroup.
(b) (Q,f) is a cancellative n-ary groupoid.
(c) (Q,f) is a solvable n-ary groupoid.

Proof $(a)\Rightarrow(b)$ follows immediately by the definitions.

$(a)\Rightarrow(c)$ follows by Lemma 1.

Clearly, (b) and (c) imply (a).

$(b)\Rightarrow(c)$: Let (Q,f) be cancellative n-ary groupoid. Then

$$\{f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)|\ x\in Q\}=Q$$

for any fixed $a_j\in Q$.

$(c)\Rightarrow(b)$: If the groupoid (Q,f) is not cancellative then, for some $a_j\in Q$ and $i\in\{0,\dots,n-1\}$, the equation $f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)=a_{n+1}$ has two different solutions $x_1\neq x_2$. Then there is an element $b\in Q$ such that $b\notin\{f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)|\ x\in Q\}$. Hence, the equation $f(a_1,\dots,a_i,x,a_{i+2},\dots,a_n)=b$ has no solution on x. ■

Let $Q = \{q_1, q_2, \dots, q_r\},\ r \geqslant 1$, and let (Q, f) be a n-ary quasigroup of order r. If we fix $a \in Q$, we define an *(a, i)-projected $(n-1)$-ary quasigroup* $(Q, f_{a,i})$ for each $i = 1, 2, \dots, n$ by

$$f_{a,i}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = f(x_1, \dots, x_{i-1}, a, x_{i+1}, \dots, x_n).$$

To every finite n-ary quasigroup of order r, an equivalent combinatorial structure n-dimensional Latin hypercubes of order r can be associated. Let Q be the set of r different elements. By *n-dimensional Latin hypercubes of order r H* we mean a n-dimensional array of r^n cells, where the cell contains an element of Q and where every set of r cells which coordinates match between themselves except in one coordinate, contains each of the elements of Q. Latin hypercubes of dimension 1, 2 and 3 are commonly called permutations, Latin squares, and Latin cubes, respectively. A *hyperplane* is the set of r^{n-1} cells of H, with one common coordinate. Any hyperplane in a n-dimensional Latin hypercubes can be considered to be a $(n-1)$-dimensional Latin hypercubes, by dropping the common coordinate. Our definition of Latin hypercubes is much broader then the one given in [95]. If we introduce an ordering in $Q = \{q_1, q_2, \dots, q_r\}$, then n-dimensional Latin hypercubes of order r, is *reduced* if in every dimension k the first elements (elements with all coordinates, except k coordinate, are 1) keep the ordering from Q. We define $\mathcal{H}_r^n$ to be the set of all n-dimensional Latin hypercubes of order r and $\mathcal{R}_r^n$ to be the set of all reduced n-dimensional Latin hypercubes of order r.

The number of n-dimensional Latin hypercubes of order r and the number of reduced n-dimensional Latin hypercubes of order r are connected with the following formula

$$\mid \mathcal{H}_r^n \mid = r!(r-1)!^{n-1} \mid \mathcal{R}_r^n \mid .$$

The usual notations of homotopism, isotopism, paratopism and isomorphism generalize naturally from binary quasigroups to n-ary quasigroups. Given n-ary quasigroups (Q, f) and (Q, h), we say that (Q, f) *is isotopic to* (Q, h) if there are permutations $\alpha_1, \alpha_2, \dots, \alpha_{n+1}$ on Q such that for every $a_j \in Q$

$$\alpha_{n+1}(f(a_1, \dots, a_n)) = h(\alpha_1(a_1), \dots, \alpha_n(a_n)).$$

The relation "is isotopic to" is an equivalence relation in the set of all n-ary quasigroups of order r. The equivalence classes are called the classes of isotopism or isotopy classes. The equivalence classes for equivalence relation "is isomorphic to" are called the classes of isomorphism.

The notation of orthogonality generalize naturally also from binary quasigroups to n-ary quasigroups. Two n-ary quasigroups (Q, f) and (Q, h) of order r are said to be *orthogonal* if for any u and v in Q, there exist a unique n tuple of elements $x_1, \ldots, x_n$ of Q such that $f(x_1, \ldots, x_n) = u$ and $h(x_1, \ldots, x_n) = v$. A set of pairwise orthogonal n-ary quasigroups is said to be a set of *mutually orthogonal n-ary quasigroups*, or in combinatorial language a set of *mutually orthogonal hypercubes (MOHC).*

One of the main objective of this section is finding a new method for enumeration of n-ary quasigroups. The enumeration of binary quasigroups has a long and fruitful history, that can be found in [94]. During our research, there were a few research in this field for higher dimensions. Mullen and Weber [105] counted the numbers of reduced Latin cubes of order 1 to 5 and their numbers of isomorphism classes. They reported the numbers of isotopy classes of Latin cubes of order 1 to 4 to be 1, 1, 1, 12. But two decades later, Jia and Qin [52] reported the same numbers for reduced Latin cubes, but gave wrong numbers 15 and 479 for the numbers of isotopy classes of Latin cubes of order 4 and 5, respectively. Our method confirm the results of Mullen and Weber for the numbers of isotopy classes of Latin cubes of order 1 to 4 [75]. See Table 1.3 for the number of isotopy classes and representative of each class for Latin cubes of order 4.

Theorem 1 *Let $Q = \{q_1, q_2, \ldots, q_r\}$, $r \geqslant 1$, and let (Q, g) and (Q, h) be two $(n-1)$-ary quasigroups from the same isotopy class. Fix a number $i \in \{1, 2, \ldots, n\}$. Then the number of n-ary quasigroups having (Q, g) as its (q_1, i)-projected $(n-1)$-ary quasigroup is equal to the number of n-ary quasigroups having (Q, h) as its (q_1, i)-projected $(n-1)$-ary quasigroup.*

In 2006 Potapov and Krotov [115] proved the following asymptotic for the $|\mathcal{H}_4^n|$:

$$3^{n+1}2^{2^n+1} \leqslant |\mathcal{H}_4^n| \leqslant (3^{n+1}+1)2^{2^n+1}$$

Our new method is based on Theorem 1.3, which allows the numbers of n-ary quasigroups (of small orders) to be computed, if the isotopy classes of $(n-1)$-ary quasigroups of given order are known. Formula for their computation is given in Corollary 1 and the proof is given in [75].

Corollary 1 *Let $Q = \{q_1, q_2, \ldots, q_r\}$, $r \geqslant 1$, and let the isotopy classes of the n-ary quasigroups on Q are $C_1, C_2, \ldots, C_k$. Then the number of n-ary quasigroups on Q is equal to*

$$b_1|C_1| + b_2|C_2| + \cdots + b_k|C_k| \tag{1.2}$$

where b_i denotes the number of n-ary quasigroups having as its $(q_1, 1)$-projected n-ary quasigroup an $(n-1)$-ary quasigroup from the class C_i. □

Isotopy class	Represent of C_i	$\lvert C_i\rvert$	b_i	$b_i\lvert C_i\rvert$
C_1	1234|2143|3412|4321|| 2143|1234|4321|3412|| 3412|4321|1234|2143|| 4321|3412|2143|1234	**864**	2292	1980288
C_2	1234|2143|3421|4312|| 2143|1234|4312|3421|| 3421|4312|2143|1234|| 4312|3421|1234|2143	**2592**	852	2208384
C_3	1234|2143|3412|4321|| 2143|1234|4321|3412|| 3412|4321|2143|1234|| 4321|3412|1234|2143	**2592**	876	2270592
C_4	1234|2143|3412|4321|| 2143|1234|4321|3412|| 3421|4312|1243|2134|| 4312|3421|2134|1243	**2592**	876	2270592
C_5	1234|2143|3412|4321|| 2143|1234|4321|3412|| 3421|4312|2134|1243|| 4312|3421|1243|2134	**2592**	876	2270592
C_6	1432|3241|4123|2314|| 4123|2314|1432|3241|| 3214|4132|2341|1423|| 2341|1423|3214|4132	**2592**	876	2270592
C_7	1432|3241|4123|2314|| 4123|2314|1432|3241|| 3241|1432|2314|4123|| 2314|4123|3241|1432	**2592**	876	2270592
C_8	1432|3241|4123|2314|| 4123|2314|1432|3241|| 3214|1423|2341|4132|| 2341|4132|3214|1423	**2592**	876	2270592
C_9	1234|2341|3412|4123|| 4123|3412|2341|1234|| 3412|1234|4123|2341|| 2341|4123|1234|3412	**5184**	144	746496
C_{10}	1234|2341|3412|4123|| 4321|1432|2143|3214|| 2413|3124|4231|1342|| 3142|4213|1324|2431	**5184**	144	746496
C_{11}	1243|2431|3124|4312|| 3421|4213|1342|2134|| 2314|3142|4231|1423|| 4132|1324|2413|3241	**5184**	144	746496
C_{12}	1234|2143|3412|4321|| 2143|1234|4321|3412|| 3412|4321|1243|2134|| 4321|3412|2134|1243	**20736**	816	16920576

Table 1.3: Isotopy classes of ternary quasigroups of order 4

By using this Corollary we calculated the cardinalities of $\mathcal{H}_4^n$ for $n \leqslant 4$, and they are 24, 576, 55 296, 36 972 288, respectively, and the cardinalities of $\mathcal{H}_5^n$ for $n \leqslant 3$, and they are 120, 576, 161 280, 2 781 803 520, respectively (see [75]). We remark that our result is the same as the result obtained by Ito [50] and the results obtained by Mullen and Weber [105]. Also in this paper we have that $|\mathcal{H}_3^n| = 3 \cdot 2^n$.

Recently there was a big progress in this field with results of McKay and Wanless [95]. Some of the main results here are the cardinalities of $\mathcal{H}_r^n$ for $r \leqslant 5$ and $n \leqslant 5$ and of the $\mathcal{H}_6^3$. The most important results in this field are represented in Table 1.4.

1.2 Quasigroup transformations

With the quasigroups one can define different quasigroup transformations.

1.2.1 Existing quasigroup transformations

$G = \mathbb{Z}_{2^n}$ be an alphabet. Let a quasigroup operation $*$ on G be chosen randomly and let $\backslash$ be left division and $/$ be the right division of $*$. Let denote by $G^+ = \{x_1x_2\dots x_t \mid x_i \in G, t \geqslant 1\}$ the set of all finite string over G. For fixed letter $l \in G$ the transformations $e_l : G^+ \to G^+$ and $d_l : G^+ \to G^+$ are defined in Markovski et al. [76], and $e'_l : G^+ \to G^+$ and $d'_l : G^+ \to G^+$ are defined in Markovski et al. [77].

$$e_l(x_1\dots x_t) = (z_1\dots z_t) \Leftrightarrow z_j = \begin{cases} l * x_1, \ j = 1 \\ z_{j-1} * x_j, \ 2 \leqslant j \leqslant t \end{cases} \tag{1.3}$$

$$d_l(z_1\dots z_t) = (x_1\dots x_t) \Leftrightarrow x_j = \begin{cases} l\backslash z_1, \ j = 1 \\ z_{j-1}\backslash z_j, \ 2 \leqslant j \leqslant t \end{cases} \tag{1.4}$$

$$e'_l(x_1\dots x_t) = (z_1\dots z_t) \Leftrightarrow z_j = \begin{cases} x_1 * l, \ j = 1 \\ x_j * z_{j-1}, \ 2 \leqslant j \leqslant t \end{cases} \tag{1.5}$$

$$d'_l(z_1\dots z_t) = (x_1\dots x_t) \Leftrightarrow x_j = \begin{cases} z_1/l, \ j = 1 \\ z_j/z_{j-1}, \ 2 \leqslant j \leqslant t \end{cases} \tag{1.6}$$

Every quasigroup transformation that apply on the given string in one pass we will call *elementary quasigroup transformation.* e_l, d_l, e'_l and d'_l are elementary quasigroup transformations. Composition of elementary quasigroup transformations we will call *composite quasigroup transformation.* For that purpose, let $*_1, *_2, \dots, *_s$ be quasigroup operations on G. Let

n	r	$\lvert\mathcal{R}_r^n\rvert$	$\lvert\mathcal{H}_r^n\rvert$	No. of isotopy classes	No. of isomorphism classes	No. of paratopy classes
2	2	1	2	1	1	1
2	3	1	12	1	5	1
2	4	4	576	2	35	2
2	5	56	161280	2	1411	2
2	6	9408	812851200	22	1130531	12
2	7	16942080	61479419904000	564	12198455835	147
2	8	535281401856	108776032459082956800	1676267	2697818331680661	283657
2	9	377597570964258816	5524751496156892842531225600	115618721533	15224734061438247321497	19270853541
2	10	7580721483160132811489280	9982437658213039871725064756920320000	208904371354363006	2750892211809150446995735533513	34817397894749939
2	11	5363937773277371298119673540771840	776966836171770144107444346734230682311065600000	-	-	-
3	2	1	2	1	1	1
3	3	1	24	1	11	1
3	4	64	55296	12	2589	5
3	5	40246	2781803520	59	23192922	15
3	6	95909896152	994393803303936000	5678334	1381105636226980	264248
4	2	1	2	1	1	1
4	3	1	48	1	21	1
4	4	7132	36972288	328	1565243	26
4	5	31503556	52260618977280	5466	435509352937	86
5	2	1	2	1	1	1
5	3	1	96	1	43	1
5	4	201538000	6268637952000	2133586	263347981121	4785
5	5	50490811256	2010196727432478720	1501786	16751644838639300	3102
6	2	1	2	1	1	1
6	3	1	192	1	85	1

Table 1.4: Number of reduced Latin hypercubes, Latin hypercubes, isotopy classes of Latin hypercubes, paratopy classes of Latin hypercubes for small order r and dimension $d \leqslant 6$

$e_{l_i},\ d_{l_i},\ e'_{l_i},\ d'_{l_i}$ $(i = 1, \ldots s)$ be transformations defined as in (1.3, 1.4, 2.1, 1.6) by choosing fixed elements $l_1,\ l_2, \ldots, l_s \in G$. Let t_{l_i} be any of previous $e_{l_i},\ d_{l_i},\ e'_{l_i},\ d'_{l_i}$ transformations. The following quasigroup $E,\ D,\ E',\ D'$ and T transformations can be defined [77]:

$$E = E^{(s)}_{l_s,\ldots,l_1} = e_{l_s} \circ e_{l_{s-1}} \circ \cdots \circ e_{l_1} \tag{1.7}$$

$$D = D^{(s)}_{l_s,\ldots,l_1} = d_{l_s} \circ d_{l_{s-1}} \circ \cdots \circ d_{l_1} \tag{1.8}$$

$$E' = E'^{(s)}_{l_s,\ldots,l_1} = e'_{l_s} \circ e'_{l_{s-1}} \circ \cdots \circ e'_{l_1} \tag{1.9}$$

$$D' = D'^{(s)}_{l_s,\ldots,l_1} = d'_{l_s} \circ d'_{l_{s-1}} \circ \cdots \circ d'_{l_1} \tag{1.10}$$

$$T = T^{(s)}_{l_s,\ldots,l_1} = t_{l_s} \circ t_{l_{s-1}} \circ \cdots \circ t_{l_1} \tag{1.11}$$

Theorem 2 *[77] The transformations $E,\ D,\ E',\ D'$ and T are permutations on G^+.*

Special kind of E transformation is the quasigroup reverse string transformation $\mathcal{R}$, first introduced in [35], where the leaders are the elements of the string, taken in reverse order.

Definition 12 Let s be a positive integer, let $(Q, *)$ be a quasigroup and $a_j \in Q,\ 1 \leqslant j \leqslant s$. **Quasigroup reverse string transformation** $\mathcal{R}$: $Q^s \to Q^s$ is defined as composition of e-transformations in following way

$$\mathcal{R}(a_1 a_2 \ldots a_s) = (e_{a_1} \circ e_{a_2} \circ \cdots \circ e_{a_s})(a_1 a_2 \ldots a_s) \tag{1.12}$$

□

Another special kind of D transformation is so called Quasigroup method 1 - QM1, defined in [78], and only special in this transformation are the special defined leaders for internal e_l-transformations.

All defined quasigroup transformations till now, transform string in other string with equal length s. The following transformation, presented in [78], transforms strings of length s into strings of length $2s$.

Definition 13 Let s be a positive integer, let $(Q, *)$ be a quasigroup and $a_j \in Q,\ 1 \leqslant j \leqslant s$. Let $(a'_1 a'_2 \ldots a'_s) = d_l(a_1 a_2 \ldots a_s)$, where $l = a_1 + a_2 + \ldots + a_s$ ($+$ is addition modulo 256). We define the mapping $\varphi : Q^s \to Q^{2s}$ by

$$\varphi(a_1 a_2 \ldots a_s) = (a_1 a'_1 a_2 a'_2 \ldots a_s a'_s).$$

Quasigroup method 2 $QM2 : Q^s \to Q^{2s}$ is defined as

$$QM2 \circ \varphi(a_1 \dots a_s) = QM2(x_1 \dots x_{2s}) = (z_1 \dots z_{2s}) \Leftrightarrow$$

$$z_j = \begin{cases} x_1 + (l * x_1), \ j = 1 \\ x_j + (x_{j-1} * x_j), \ 2 \leqslant j \leqslant 2s \end{cases} \tag{1.13}$$

□

1.2.2 Properties of sequences produced by quasigroup transformations

There are extensive theoretical studies and numerical experiments of the sequences produced by quasigroup transformations E, E', D and D' [77, 84, 85]. We present some of the most important.

Theorem 3 *Consider an arbitrary string $\beta = b_1 b_2 \dots b_t \in G^+$, where $b_i \in G$, and let $\gamma = E^{(s)}(\beta)$ and $\gamma' = E'^{(s)}(\beta)$. If n is sufficiently large integer then, for each l: $1 \leqslant l \leqslant s$ the distribution of substrings of γ and γ' of length l is uniform. (We note that for $l > k$ the distribution of substrings of γ and γ' of length l may not be uniform.)*

The Theorem 3 means, that if we apply once E or E' transformations on long enough string from alphabet G, every letter from G is appearing almost equally in the produced string. Generally, if we apply E or E' transformations l times, for $l \leqslant k$, then every substring with length l is appearing almost equally in the produced string.

Another important properties of obtained sequences by quasigroup string transformations are concerning their period.

Definition 14 The string $\beta = b_1 b_2 \dots b_s \in G^+$, where $b_i \in G$, has a *period p*, if p is the smallest integer, for which the following equality is true $a_{i+1}a_{i+2} \dots a_{i+p} = a_{i+p+1}a_{i+p+2} \dots a_{i+2p}$ for every $i \geqslant 0$. □

Theorem 4 *[74] Let α be a sequence of k elements. If the period of $E_l^{(1)}(\alpha)$ is p_0, then the sequences $E_l^{(t)}(\alpha)$ are periodical with periods p_{t-1} correspondingly, all of which are multiples of p_0. The periods satisfy the law $p_{p_{t-1}} > p_{t-1}$ for each $t \geqslant 1$.* □

Theorem 4 means that the period of the sequences obtained with consecutive application of quasigroup transformations, grows at least linearly. Let $\alpha = q_0 q_1 \dots q_{p-1} q_0 q_1 \dots q_{p-1} \dots$ be an enough long string of period p

over G and let $\alpha_k = E^{(n)}(\alpha)$. The following classification can be made on quasigroups [80]. If the period of the string α_k is a linear function of k, then the quasigroup $(G, *)$ is said to be *linear.* If the period of the string α_k is an exponential function 2^{ck} (where c is some constant), then the quasigroup $(G, *)$ is said to be *exponential.* The number c is called the *period growth* of the exponential quasigroup $(G, *)$ and represents how many times the period has grown (in average) after one application of the quasigroup transformation. It is obvious that the ideal period growth is at most the order n of the quasigroup. Thus, ideally, if we apply k times the quasigroup transformation, the period of obtained sequence will be n^k. From numerical experiments in [25] the percentage of linear quasigroups decreases when the order of the quasigroup increases and the percentage of the linear quasigroups and exponential quasigroup with period growth less then 2, is decreasing exponentially by the order of the quasigroups.

1.2.3 Left and right quasigroups

Definition 15 A groupoid $(G, \cdot)$ is said to be a **left quasigroup** (a **right quasigroup**) if the equation $xa = b$ $(ay = b)$ have a unique solution x (y) in G for every $a, b \in G$. □

In this subsection we define two special kinds of left and right quasigroups. They are going to be used for definition of quasigroup transformations.

Proposition 3 *Let $(G, +)$ be a group and let $(G, *)$ be a quasigroup. Then the operation $\bullet$ defined by $x \bullet y = (x + y) * y$ defines a left quasigroup $(G, \bullet)$.* □

Proof The solution $x = (b/a) - a$ of the equation $x \bullet a = b$ is unique, since $x \bullet y = x' \bullet y \Longrightarrow x = x'$. ■

Proposition 4 *Let $(G, +)$ be a group and let $(G, *)$ be a quasigroup. Then the operation $\diamond$ defined by $x \diamond y = x * (x + y)$ defines a right quasigroup $(G, \diamond)$.* □

Proof The solution $y = -a + (a\backslash b)$ of the equation $a \diamond y = b$ is unique, since $x \diamond y = x \diamond y' \Longrightarrow y = y'$. ■

Given a groupoid $(G, \cdot)$, for each $a \in G$ the left and the right translations L_a and R_a are defined by $L_a(x) = xa$ and $R_a(x) = ax$ respectfully. If $(G, \cdot)$ is a left (right) quasigroup then its left (right) translation is a permutation, while the right (left) translation can be arbitrary mapping.

Considering the left and the right quasigroups defined as in Proposition 3 and Proposition 4, the situation is quite different in the case when $G = \mathbb{Z}_{2^n}$ and the group operation is addition modulo 2^n. Namely, the right translation of $(G, \bullet)$ and the left translation of $(G, \diamond)$ may not be permutations in that case either. However, the probability of that event is quite small, roughly speaking, around $2/|G|$. To show the last statement we consider the problem of finding solutions of the equation $x \diamond a = b$, i.e.,

$$x * (x + a) = b \tag{1.14}$$

where $a, b \in G$ are given, and x is unknown.

Proposition 5 *Let $G = \mathbb{Z}_{2^n}$ be with group operation addition modulo 2^n. Let a quasigroup operation $*$ on G be chosen randomly. Then the probability the right quasigroup $(G, \diamond)$ to have two different solutions $x_1 \neq x_2$ of the equation (1.14) is less or equal to $\dfrac{2}{2^n - 1}$.* □

Proof Let x_1 and x_2 be two different solutions of the equation $x*(x+a) = b$. Then

$$\begin{cases} x_1 * (x_1 + a) = b \\ x_2 * (x_2 + a) = b \end{cases} \Rightarrow \begin{cases} x_1 \setminus b - x_1 = a \\ x_2 \setminus b - x_2 = a \end{cases} \Rightarrow x_1 \setminus b - x_2 \setminus b = x_1 - x_2 \neq 0.$$

At first, we find the probability a random quasigroup to satisfy the event $x_1 \setminus b - x_2 \setminus b = x_1 - x_2 \neq 0$.

The difference $x_1 - x_2$ can take any value $r \in G$, where $r \neq 0$. Fix an $r \neq 0$. Then there are $\binom{2^n}{2}$ pairs of different elements of G, and exactly 2^n of them satisfy the equation $x_1 - x_2 = r$. Hence, we have this probability for any fixed $r \neq 0$: $P_r\{x_1, x_2 \in G, x_1 - x_2 = r\} = \frac{2}{2^n - 1}$.

Consider now the equation $x_1 \backslash b - x_2 \backslash b = s$, where $s \neq 0 \in G$ is given. Denote by K the set of all quasigroups on G and let fix a solution (x_1, x_2) of $x_1 \backslash b - x_2 \backslash b = s$. Denote by $K_s = K_s(x_1, x_2)$ the set of all quasigroups on G with the property $x_1 \backslash b - x_2 \backslash b = s$. Then $|K_s| = |K_t|$ for each s and t. Namely, if $(G, \backslash_1) \in K_s$, then we can construct a quasigroup $(G, \backslash_2) \in K_t$ as follows. At first choose $x_1 \backslash_2 b$ and $x_2 \backslash_2 b$ such that $x_1 \backslash_2 b - x_2 \backslash_2 b = t$ and let π be the permutation generated by the two transpositions $(x_1 \backslash_1 b, x_1 \backslash_2 b), (x_2 \backslash_1 b, x_2 \backslash_2 b)$. Then define the operation $\backslash_2$ for each $u, v \in G$ by $u \backslash_2 v = \pi(u \backslash_1 v)$. (Note that we have obtained $(G, \backslash_2)$ from $(G, \backslash_1)$ in such a way that we have only replaced in the multiplication table of $(G, \backslash_1)$ all appearances of $x_1 \backslash_1 b$ $(x_2 \backslash_1 b)$ by $x_1 \backslash_2 b$ $(x_2 \backslash_2 b)$.) Now, for given $x_1, x_2 \in G$ and randomly chosen quasigroup $(Q, \backslash)$, we have the probability $P_s\{Q \in K, x_1 \backslash b - x_2 \backslash b = s \text{ is true in Q}\} = \frac{|K_s|}{|K|} = \frac{1}{2^n - 1}$.

Consequently, the probability a random quasigroup $(G,*)$ to satisfy the event $x_1 \setminus b - x_2 \setminus b = x_1 - x_2 \neq 0$ is

$$P\{x_1 - x_2 = r, x_1\backslash b - x_2\backslash b = r, r > 0\} =$$
$$\sum_{r=1}^{q-1} P\{x_1 - x_2 = r, x_1\backslash b - x_2\backslash b = r\} =$$
$$\sum_{r=1}^{2^n-1} P\{x_1\backslash b - x_2\backslash b = r |\ x_1 - x_2 = r\} P\{x_1 - x_2 = r\} =$$
$$\sum_{r=1}^{2^n-1} P_s\{Q \in K, x_1\backslash b - x_2\backslash b = r\} P_r\{x_1, x_2 \in G, x_1 - x_2 = r\} = \frac{2}{2^n-1}.$$

Finally, if we additionally take the condition $x_1\backslash b - x_1 = a$, we conclude that the probability a right quasigroup $(G,\diamond)$ to have two different solutions $x_1 \neq x_2$ of the equation (1.14) is less or equal than $\frac{2}{2^n-1}$. ■

In similar way one can prove the same property for left quasigroup $(G,\bullet)$.

Proposition 6 *Let $G = \mathbb{Z}_{2^n}$ be with group operation addition modulo 2^n. Let a quasigroup operation $*$ on G be chosen randomly. Then the probability the left quasigroup $(G,\bullet)$ to have two different solutions $x_1 \neq x_2$ of the equation*

$$(a + x) * x = b \tag{1.16}$$

□

is less or equal to $\dfrac{2}{2^n-1}$. □

Remark 1 In the set of all 576 quasigroups of order 4, each equation of kind $x*(x+a) = b$ (or $(a+x)*x = b$) has two (or more) solutions in exactly 168 quasigroups.

1.2.4 Some new quasigroup transformations

If we allow $G = \mathbb{Z}_{2^n}$ to be with group operation addition modulo 2^n, with previous defined left and right quasigroups we can define several new quasigroup transformations.

Definition 16 **Quasigroup additive string transformation** $\mathcal{A}_l : G^+ \to G^+$ **with leader** l is the transformation defined by

$$\mathcal{A}_l(x_1 \dots x_t) = (z_1 \dots z_t) \Leftrightarrow z_j = \begin{cases} (l + x_1) * x_1, \ j = 1 \\ (z_{j-1} + x_j) * x_j, \ 2 \leqslant j \leqslant t \end{cases} \tag{1.17}$$

where $x_i, z_i \in G, t \geqslant 1$. □

Definition 17 Quasigroup reverse additive string transformation $\mathcal{RA}_l : G^+ \to G^+$ **with leader** l is the transformation defined by

$$\mathcal{RA}_l(x_1 \dots x_t) = (z_1 \dots z_t) \Leftrightarrow z_j = \begin{cases} x_j * (x_j + z_{j+1}), \ 1 \leqslant j \leqslant t-1 \\ x_t * (x_t + l), \ j = t \end{cases} \tag{1.18}$$

where $x_i, z_i \in G, t \geqslant 1$. □

These transformations are not bijective mappings. Let $\mathcal{A}_{l_i}$ and $\mathcal{RA}_{l_i} (i = 1, \dots s)$ be transformations defined by choosing fixed elements $l_1, l_2, \dots, l_s \in G$. Let m_{l_i} be any of previous $\mathcal{A}_{l_i}, \mathcal{RA}_{l_i}$ transformations. We can define M transformations as

$$M = m_{l_1} \circ m_{l_2} \circ \dots \circ m_{l_n} \tag{1.19}$$

For an element $z \in G = \mathbb{Z}_{2^n}$ denote by $\rho(z, \lfloor \frac{n}{2} \rfloor)$ the element in G obtained by rotating left for $\lfloor \frac{n}{2} \rfloor$ bits the n-bit representation of z. Given a string $Z = (z_1 \dots z_t) \in G^t$, we denote by $\rho(Z)$ the string

$$\rho(Z) = \big(\rho(z_1, \lfloor \frac{n}{2} \rfloor) \dots \rho(z_t, \lfloor \frac{n}{2} \rfloor)\big) \in (\mathbb{Z}_{2^n})^t.$$

For a function $f = f(Z)$ we define a new function $\rho(f) = \rho(f)(Z)$ by $\rho(f)(Z) = f(\rho(Z))$.

Definition 18 Quasigroup main transformation $\mathcal{MT} : G^+ \to G^+$ **with complexity** k is defined as composition of transformations of kind $\mathcal{A}_{l_i}$ followed by $\rho(\mathcal{RA}_{l_j})$, for suitable choices of the leaders l_i and l_j as functions depending on variables $x_1, x_2, \dots, x_t$, as follows. For every $x_\lambda \in G$

$$\mathcal{MT}(x_1 \dots x_t) = \rho(\mathcal{RA}_{l_1})(\mathcal{A}_{l_2}(\dots(\rho(\mathcal{RA}_{l_{k-1}})(\mathcal{A}_{l_k}(x_1 \dots x_t)))\dots)), \tag{1.20}$$

i.e., $\mathcal{MT} = \rho(\mathcal{RA}_{l_1}) \circ \mathcal{A}_{l_2} \circ \dots \circ \rho(\mathcal{RA}_{l_{k-1}}) \circ \mathcal{A}_{l_k}$, where $\circ$ denotes a composition of functions. □

The main transformation is special kind of M transformation, which will be used later for cryptographic purposes.

In [101], another new quasigroup transformation is given. Let Q be endowed with two orthogonal quasigroup operations $*_1$ and $*_2$. Then we define so called **orthogonal quasigroup string transformation** $OT : Q^+ \to Q^+$ by the following iterative procedure.

$OT(x_1) = x_1$, $OT(x_1, x_2) = (x_1 *_1 x_2, x_1 *_2 x_2)$, and if $OT(x_1, x_2, \dots, x_{t-2}, x_{t-1}) = (z_1, z_2, \dots, z_{t-1})$ is defined for $t > 2$, then

$$OT(x_1, x_2, \dots, x_{t-1}, x_t) = (z_1, z_2, \dots, z_{t-1} *_1 x_t, z_{t-1} *_2 x_t), \tag{1.21}$$

where $x_i \in Q$.

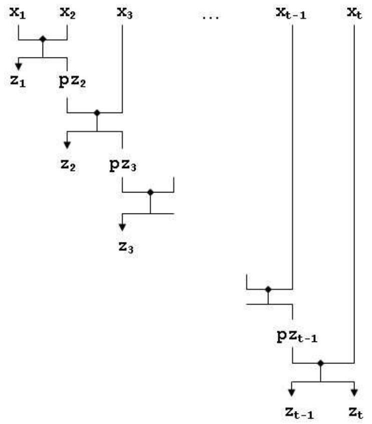

Figure 1: Schematic representation of the orthogonal quasigroup string transformation OT

Schematic representation of OT is given on Fig. 1. Note that the restriction OT_n of OT on the set Q^n is a mapping $OT_n : Q^n \to Q^n$ and so $OT = OT_1 \cup OT_2 \cup OT_3 \cup \ldots$, i.e., OT is a disjoint union of the mappings OT_n. OT_1 is the identity mapping on Q, so it is a permutation. OT_2 is a permutation of Q_2 since $(x_1 *_1 x_2, x_1 *_2 x_2) = (y_1 *_1 y_2, y_1 *_2 y_2)$ implies $(x_1, x_2) = (y_1, y_2)$ by the orthogonality of the quasigroup operations $*_1$ and $*_2$. Suppose that OT_{t-1} is a permutation for $t > 2$, and let $OT_t(x_1, x_2, \ldots, x_t) = OT_t(y_1, y_2, \ldots, y_t) = (z_1, z_2, \ldots, z_t)$. Let $OT_{t-1}(x_1, x_2, \ldots, x_{t-1}) = (u_1, u_2, \ldots, u_{t-1})$ and $OT_{t-1}(y_1, y_2, \ldots, y_{t-1}) = (v_1, v_2, \ldots, v_{t-1})$. Then $z_1 = u_1 = v_1$, $z_2 = u_2 = v_2$, $\ldots$, $z_{t-2} = u_{t-2} = v_{t-2}$ and $(z_{t-1}, z_t) = (u_{t-1} *_1 x_t, u_{t-1} *_2 x_t) = (v_{t-1} *_1 y_t, v_{t-1} *_2 y_t)$, that implies $(u_{t-1}, x_t) = (v_{t-1}, y_t)$ by orthogonality of $*_1$ and $*_2$. We have $x_t = y_t$ and $OT_{t-1}(x_1, x_2, \ldots, x_{t-1}) = OT_{t-1}(y_1, y_2, \ldots, y_{t-1}) = (z_1, z_2, \ldots, z_{t-2}, u_{t-1} = v_{t-1})$. Thus we have proved the following.

Theorem 5 *The orthogonal quasigroup string transformation OT is a permutation on Q^+, and its restriction OT_n is a permutation on Q_n for each positive integer n.* □

Note that if quasigroup operations are not orthogonal, the transformation defined by 1.21 is not necessarily a permutation.

1.3 How to choose a quasigroup

Quasigroups and quasigroup transformations have many applications in cryptography, coding theory, design theory and others. Our interest is specially application of quasigroups in cryptography. Quasigroups are very suitable for that purpose, because of their structure, features and big number. Effects of quasigroup transformations depend at most from the choice of a quasigroup. So, one of the problems is which quasigroup is suitable to be chosen for using, concerning what preconditions the quasigroup must fulfill. Several classifications are existing today and helping us in our choices.

Three main classifications are obtained by using the algebraic properties of the quasigroups to classes of isotopic quasigroups, classes of isomorphic quasigroups and classes of paratopic quasigroups. Quasigroups are classified on varieties according to identities they satisfy, like Schroeder quasigroups, totally anti-symmetric quasigroups, Stein quasigroups, Moufang quasigroups etc. There are some special classifications on quasigroups of small orders such as by random walk on torus (Markovski et al. [81]), by period of produced sequences (Markovski et al. [80], dividing to linear and exponential quasigroups) or by graphical presentation of sequences obtained by quasigroup transformations (Dimitrova [24], dividing to fractal and non-fractal quasigroups on quasigroups of order 4).

Specially, we are interesting in classification obtained from Gligoroski et al. [36], by examining the quasigroups as vector valued Boolean functions.

1.3.1 Quasigroups as vector valued Boolean functions

We denote by $\mathbb{F}_2$ the Galois field with two elements. A *Boolean function* of s variables or *$s-$ary Boolean function* is a function

$$b : \mathbb{F}_2^s \to \mathbb{F}_2.$$

A *vector valued Boolean function* is a map

$$B : \mathbb{F}_2^s \to \mathbb{F}_2^t,\ (t \geqslant 1)$$

Every vector valued Boolean function B can be represented by t $s-$ary Boolean functions $b_i : {\mathbb{F}_2}^s \to \mathbb{F}_2$ as follows:

$$B(x_1,\dots,x_s) = (b_1(x_1,\dots,x_s), b_2(x_1,\dots x_s),\dots,b_t(x_1,\dots,x_s)),$$

where

$$b_1(x_1,\dots,x_s) = y_1,\dots,b_t(x_1,\dots,x_s) = y_t \iff B(x_1,\dots,x_s) = (y_1,\dots,y_t).$$

Each $s-$ary Boolean function b_i can be represented in Algebraic Normal Form as

$$b_i(x_1,x_2,\dots,x_s) = \sum_{I\subseteq\{1,2,\dots,s\}} \alpha_I(\prod_{i\in I} x_i) \tag{1.22}$$

where $\alpha_I \in \mathbb{F}_2$, the sum is for the Boolean function XOR and the product is for the Boolean function conjunction. The right-hand side of (1.22) can be interpreted as a polynomial in the field $(\mathbb{F}_2, +, \cdot)$ and the degree of b_i is taken to be the degree of the polynomial. The algebraic degree of a vector valued Boolean function B is defined as the maximum of the degrees of its component polynomials $(b_1, b_2, \ldots, b_s)$:

$$deg(B) = \mathtt{max}\{deg(b_i) \mid i \in \{1, 2, \ldots, s\}\}.$$

If $deg(B) = 1$, then B is said to be linear. In the sequel, another definition of linear and affine function is given.

Definition 19 Let $(G, +)$ be a group and let $f : G \to G$ be a function. f is an **affine function** if $f(x+y) = f(x) + f(y) - f(0)$ for each $x, y \in G$, where $0 \in G$ is the identity element. A **linear function** is an affine function f with $f(0) = 0$. □

Now, every quasigroup $(Q, \circ)$ of order 2^n can be represented as vector valued Boolean function $B : \{0,1\}^{2n} \to \{0,1\}^n$ and every $x \in Q$, can be represented as n-dimensional binary vector $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$. We have:

$$x \circ y = (x_1, x_2, \ldots, x_n) \circ (y_1, y_2, \ldots, y_n) = B(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) =$$

$$(b_1(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n), \ldots, b_n(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n))$$

where b_i are $2n-$ary Boolean functions of B. We can represent B by true table or by ANF. In the second case we say that B is represented by n-tuple of polynomials $(b_1, \ldots, b_n)$ and algebraic degree of B is the maximum of the degrees of its component polynomials.

In [41, 47] one can find definition of the so called multivariate quadratic quasigroups.

Definition 20 A quasigroup $(Q, *)$ of order 2^n is called *Multivariate Quadratic Quasigroup (MQQ) of type* $Quad_{n-k}Lin_k$ if exactly $n-k$ of the polynomials b_i are of degree 2 (i.e., are quadratic) and k of them are of degree 1 (i.e., are linear), where $0 \leqslant k < n$. □

It can be observed that the degrees of the $2n-$ary Boolean functions rise with the order of the quasigroup. About the opposite problem, if the given family of $2n-$ary Boolean functions $b_1, \ldots, b_n$ determines a quasigroup, the following Theorem is true.

Theorem 6 *[107] A family of n Boolean functions $b_1, \dots, b_n$ in $2n$ variables determines a quasigroup iff the following holds:*

- *if one takes any product $b_{i_1}, \dots, b_{i_k}$ $1 \leqslant i_1 < \dots < i_k \leqslant n$, then its algebraic normal form does not contain terms including either $x_1 x_2 \dots x_n$ or $y_1 y_2 \dots y_n$.*
- *the product $b_1 \dots b_n$ contains both these terms and no other term containing either of them.* □

By classification in [36], quasigroups are divided in linear and non-linear quasigroups. *Linear quasigroups* are the quasigroups with all linear component Boolean functions. If one component Boolean function is non-linear, than the appropriate quasigroup is *non-linear.* But for building non-linear cryptographic primitives it is not good to have any linear component Boolean function. So, we introduce an augmentation to this classification, by dividing non-linear quasigroups to weak non-linear and pure non-linear quasigroups:
1. *linear quasigroups* - when all component Boolean functions are linear
2. *weak non-linear quasigroups* - when there exist one component Boolean function that is linear and one component Boolean function that is non-linear
3. *pure non-linear quasigroups* - when all component Boolean functions are non-linear.

Remark 2 From 576 quasigroup of order 4, 144 are linear (G_0), 288 are weak non-linear (G_1) and 144 are pure non-linear quasigroups (G_2).

$G_0 = \{1, 4, 11, 14, 21, 24, 26, 27, 37, 40, 43, 46, 51, 54, 57, 60, 70, 71, 77, 80, 82, 83, 92, 93, 100, 101, 110, 111, 113, 116, 126, 127, 132, 133, 138, 139, 146, 147, 157, 160, 163, 166, 169, 172, 179, 182, 189, 192, 196, 197, 203, 206, 212, 213, 222, 223, 228, 229, 234, 235, 243, 246, 252, 253, 259, 262, 269, 272, 274, 275, 284, 285, 292, 293, 302, 303, 305, 308, 315, 318, 324, 325, 331, 334, 342, 343, 348, 349, 354, 355, 364, 365, 371, 374, 380, 381, 385, 388, 395, 398, 405, 408, 411, 414, 417, 420, 430, 431, 438, 439, 444, 445, 450, 451, 461, 464, 466, 467, 476, 477, 484, 485, 494, 495, 497, 500, 506, 507, 517, 520, 523, 526, 531, 534, 537, 540, 550, 551, 553, 556, 563, 566, 573, 576\}$

$G_1 = \{2, 3, 5, 6, 12, 13, 15, 16, 17, 18, 19, 20, 25, 28, 29, 30, 35, 36, 38, 39, 41, 42, 47, 48, 52, 53, 55, 56, 58, 59, 61, 62, 65, 66, 67, 68, 75, 76, 78, 79, 81, 84, 85, 86, 89, 90, 95, 96, 97, 98, 99, 102, 105, 106, 109, 112, 117, 118, 119, 120, 121, 122, 125, 128, 129, 130, 131, 134, 141, 142, 143, 144, 145, 148, 151, 152, 153, 154, 155, 156, 164, 165, 167, 168, 170, 171, 175, 176, 177, 178, 183, 184, 187, 188, 190, 191, 193, 194, 195, 198, 201, 202, 207, 208, 211, 214, 215, 216, 217, 218, 221, 224, 225, 226, 231, 232, 233, 236, 239, 240, 244, 245, 247, 248, 249, 250, 255, 256, 257, 258, 260, 261, 267, 268, 270, 271, 277, 278, 279, 280, 281, 282, 283, 286, 291, 294, 295, 296, 297, 298, 299, 300, 306, 307, 309, 310, 316, 317, 319, 320, 321, 322, 327, 328, 329, 330, 332, 333, 337, 338, 341, 344, 345, 346, 351, 352, 353, 356, 359, 360, 361, 362, 363, 366, 369, 370, 375, 376, 379, 382, 383, 384, 386, 387, 389, 390, 393, 394, 399, 400, 401, 402, 406, 407, 409, 410, 412, 413, 421, 422, 423, 424, 425, 426, 429, 432, 433, 434, 435, 436, 443, 446, 447, 448, 449, 452, 455, 456, 457, 458, 459, 460, 465, 468, 471, 472, 475, 478, 479, 480, 481, 482, 487, 488, 491, 492, 493, 496, 498, 499, 501, 502, 509, 510, 511, 512, 515, 516, 518, 519, 521, 522, 524, 525, 529, 530, 535, 536, 538, 539, 541, 542, 547, 548, 549, 552, 557, 558, 559, 560, 561, 562, 564, 565, 571, 572, 574, 575\}$

$G_2 = \{7, 8, 9, 10, 22, 23, 31, 32, 33, 34, 44, 45, 49, 50, 63, 64, 69, 72, 73, 74, 87, 88, 91, 94, 103, 104, 107, 108, 114, 115, 123, 124, 135, 136, 137, 140, 149, 150, 158, 159, 161, 162, 173, 174, 180, 181, 185, 186, 199,$

$200, 204, 205, 209, 210, 219, 220, 227, 230, 237, 238, 241, 242, 251, 254, 263, 264, 265, 266, 273, 276, 287, 288, 289, 290, 301, 304, 311, 312, 313, 314, 323, 326, 335, 336, 339, 340, 347, 350, 357, 358, 367, 368, 372, 373, 377, 378, 391, 392, 396, 397, 403, 404, 415, 416, 418, 419, 427, 428, 437, 440, 441, 442, 453, 454, 462, 463, 469, 470, 473, 474, 483, 486, 489, 490, 503, 504, 505, 508, 513, 514, 527, 528, 532, 533, 543, 544, 545, 546, 554, 555, 567, 568, 569, 570\}$ □

1.3.2 Quasigroup transformations as vector valued Boolean functions

Let $QT_{L,s,t} : Q^t \to Q^t$ be a family of quasigroup transformations defined by the quasigroup $(Q, *)$, $|Q| = 2^n$, that are composition of s elementary quasigroup transformations, with leader string L of length s, $s \geqslant 1$. The transformation $QT_{L,s,t}$ can be represented as vector valued Boolean function $BQT_{L,s,t} : \{0,1\}^{tn} \to \{0,1\}^{tn}$.

Example 3 For the quasigroup of order 4 with lexicographic order 231 (given in Table 1.5), the elementary quasigroup transformations $e_1, d_1, \mathcal{A}_1$ and $\mathcal{RA}_1$ ($s = 1$ and $L = 1$) of strings of length $t = 2$, can be represented as vector valued Boolean functions $\{0,1\}^4 \to \{0,1\}^4$ (see Table 1.6), using integer representation.

$\circ$	0	1	2	3
0	1	2	3	0
1	2	3	0	1
2	0	1	2	3
3	3	0	1	2

Table 1.5: Quasigroup 231

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$e_1(x)$	8	9	10	11	15	12	13	14	1	2	3	0	6	7	8	9
$d_1(x)$	9	10	11	8	14	15	12	13	0	1	2	3	7	4	5	6
$\mathcal{A}_1(x)$	8	8	11	9	6	5	5	4	6	5	5	4	1	3	2	2
$\mathcal{RA}_1(x)$	14	4	3	3	6	12	11	11	2	8	7	7	2	8	7	7

Table 1.6: Transformations $e_1, d_1, \mathcal{A}_1$ and $\mathcal{RA}_1$ represented as vector valued Boolean functions

We can take the leader string L to be consider as a string of variables and in such a way we obtain a family of transformations $QT_{s,t} : Q^s \times Q^t \to Q^t$, where the elements of Q^s are considered as leaders. Then, the transformation $QT_{s,t}$ can be represented as vector valued Boolean functions $BQT_{s,t} : \{0,1\}^{sn} \times \{0,1\}^{tn} \to \{0,1\}^{tn}$. For example, the transformation $E = e_{l_1} \circ e_{l_2} \circ e_{l_3}$ obtained by a quasigroup of order 2^3 ($n = 3$), for strings with length $t = 4$, can be represented as $BE : \{0,1\}^9 \times \{0,1\}^{12} \to \{0,1\}^{12}$.

Example 4 For the same quasigroup 231, the elementary quasigroup transformation e_l of strings of length 2, can be represented as vector valued Boolean functions $\{0,1\}^2 \times \{0,1\}^4 \rightarrow \{0,1\}^4$ (see Table 1.7), using integer representation.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$l=0$	6	7	4	5	8	9	10	11	15	12	13	14	1	2	3	0
$l=1$	8	9	10	11	15	12	13	14	1	2	3	0	6	7	4	5
$l=2$	1	2	3	0	6	7	4	5	8	9	10	11	15	12	13	14
$l=3$	15	12	13	14	1	2	3	0	6	7	4	5	8	9	10	11

Table 1.7: The transformation e_l as vector valued Boolean function

1.3.3 Quasigroups correlation matrices and prop ratio tables

The correlation matrix of vector valued Boolean functions is an useful concept, introduced by Daemen et al. [14], in demonstrating and proving their properties. This is useful because most components of cryptographic primitives are vector valued Boolean functions. The elements of the correlation matrices consist of the correlation coefficients associated with linear combinations of input bits and linear combinations of output bits. Linear cryptanalysis (introduced by Matsui [91]) can be seen as the exploitation of correlations between linear combinations of bits of different intermediate encryption values in a block cipher calculation, so correlation matrices are therefore the natural representation for the description and understanding of the mechanisms of the linear cryptanalysis.

Definition 21 The **correlation coefficient** associated with a pair of Boolean functions $f(a)$ and $g(a)$ is denoted by $C(f,g)$ and is given by

$$C(f,g) = 2P[f(a) = g(a)] - 1$$

The correlation coefficient ranges between -1 and 1 and if it is different from 0, the functions are said to be *correlated.*

A *selection vector* w is a binary vector that selects all components i of a vector that have $w_i = 1$. By $w^T a$ can be represented the linear combination of the components of a vector a selected by w.

Let $\hat{f}(a)$ be a real-valued function defined by $\hat{f}(a) = (-1)^{f(a)}$, so in regards of a linear Boolean function, $w^T a$ becomes $(-1)^{w^T a}$. The bitwise sum of two Boolean functions corresponds to the bitwise product of their real-valued counterparts, i.e., $f(a)\hat{+}g(a) = \hat{f}(a)\hat{g}(a)$.

The *inner product* of real-valued functions is defined by,

$$\langle \hat{f}(a), \hat{g}(a) \rangle = \sum_a \hat{f}(a)\hat{g}(a)$$

It is shown in [14] that

$$C(f,g) = 2^{-n}\langle \hat{f}(a), \hat{g}(a) \rangle = 2^{-n}\sum_a (-1)^{f(a)}(-1)^{g(a)}.$$

If $C(f,g) = 1$, then $f(a) = g(a) = 0$ for every a. If $C(f,g) = -1$, then $f(a) \oplus g(a) = 1$ for every a.

The real-valued functions corresponding to the linear Boolean functions form an orthogonal basis with respect to the defined inner product:

$$\langle (-1)^{u^T a}, (-1)^{v^T a} \rangle = 2^n \delta(u+v)$$

where $\delta(w)$ is the real-valued function equal to 1 if w is the zero vector and 0 otherwise.

All correlation coefficients between linear combinations of input bits and that of output bits of the mapping h can be arranged in a correlation $2^m \times 2^n$-matrix C^h. The element C_{uw} in the row u and the column w is equal to $C(u^T h(a), w^T a)$. The rows in this matrix can be interpreted as

$$(-1)^{u^T h(a)} = \sum_w C^h_{uw} (-1)^{w^T a}.$$

In words, this means that the real-valued function corresponding to a linear combination of output bits can be written as a linear combination of the real-valued functions corresponding to a linear combination of input bits. One can see that if the correlation coefficient $C_{uw} = 1$ ($C_{uw} = -1$), then linear (affine) combination of output bits selected by u can be written as linear (affine) combination of input bits selected by w. This means that if $u = 2^i,\ i = 0, \ldots n-1$ and $C_{uw} = 1$ ($C_{uw} = -1$), component polynomial for $(n-i)-$th bit is linear (affine) function and can be read from its correlation matrix.

Correlation matrices can be applied to express correlations in iterated transformations, such as most block ciphers, hash functions etc. Linear cryptanalysis are possible if there are predictable input-output correlations over all but a few rounds significantly larger than $2^{n/2}$, where n is the block length of the block ciphers (see Daemen [13]). An input-output correlation is composed of linear trails and, in order a cryptographic primitive to be resistant against this attack, a necessary condition is that there are no linear trails with correlation coefficients higher than $2^{n/2}$.

Differential cryptanalysis (introduced by Biham and Shamir [7]) exploits difference propagation and so, as a tool for its examination, one can uses $2^m \times 2^n$ prop ratio tables (see Daemen [13]).

Let a and a^* be n-dimensional vectors with bitwise difference $a \oplus a^* = a'$. Let $b = h(a), b^* = h(a^*)$ and $b' = b \oplus b^*$. Hence, the difference a' propagates to the difference b' through mapping h and this can be represented by $(a' \dashv h \vdash b')$.

Definition 22 The **prop ratio** R_p of a difference propagation $(a' \dashv h \vdash b')$ is given by

$$R_p(a' \dashv h \vdash b') = 2^{-n} \sum_a \delta(b' \oplus h(a \oplus a') \oplus h(a)).$$

The prop ratio ranges between 0 and 1 and if a pair is chosen uniformly from the set of all pairs $(a, a*)$ with $a \oplus a* = a'$, the equality $h(a) \oplus h(a*) = b'$ is true with some probability. It can be easily seen that $\sum_b R_p(a' \dashv h \vdash b') = 1$. If $R_p(a' \dashv h \vdash b') = 0$, the difference propagation $(a' \dashv h \vdash b')$ is called *invalid*. The input difference a' and the output difference b' are said to be *incompatible* through h. Difference propagation is composed of differential trails.

Definition 23 The **restriction weight** of a valid difference propagation $(a' \dashv h \vdash b')$ is the negative of the binary logarithm of the prop ratio, i.e.,

$$w_r(a' \dashv h \vdash b') = -log_2 R_p(a' \dashv h \vdash b')$$

The restriction weight ranges between 0 and $n-1$ and can be seen as the amount of information (in bits) that is restricted by $(a' \dashv h \vdash b')$ on a. If h is linear, $w_r(a' \dashv h \vdash b') = 0$, so it can be seen that this difference propagation does not restrict or gives away information on a.

The correlation matrix and the prop ratio table of a mapping h are connected through the following Theorem from Daemen [13].

Theorem 7 *The table of prop ratios and the table containing the squared elements of the correlation matrix of a vector valued Boolean function h are linked by,*

$$R_p(a' \dashv h \vdash b') = 2^{-m} \sum_{u,w} (-1)^{w^T a' + u^T b'} C_{uw}^2$$

and, dually, by

$$C_{uw}^2 = 2^{-n} \sum_{a',b'} (-1)^{w^T a' + u^T b'} R_p(a' \dashv h \vdash b')$$

Differential cryptanalysis attacks are possible if there are predictable difference propagations over all but a few rounds that have prop ratio significantly larger than 2^{1-n}, where n is the block length in the block ciphers [13]. To be resistant against this attack, necessary condition is that there are no differential trails with predicted prop ratio higher than 2^{1-n}.

Example 5 The quasigroup of order 4 with lexicographic order 211 (given in Table 1.8) can be represented as a vector valued Boolean function $h : \{0,1\}^4 \to \{0,1\}^2$ and $h(x_0, x_1, x_2, x_3) = (y_0, y_1)$, where $y_0 = y_0(x_0, x_1, x_2, x_3)$ and $y_1 = y_1(x_0, x_1, x_2, x_3)$. Below we will show that the functions y_0 and y_1 can be found from the correlation matrix, in the case when they are linear.

$\circ$	0	1	2	3
0	1	2	0	3
1	3	0	2	1
2	0	1	3	2
3	2	3	1	0

The correlation matrix and prop ratio table of h are given in Table 1.8 and Table 1.9.

h	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
10	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	$\frac{1}{2}$	$-\frac{1}{2}$	0
11	0	0	0	0	0	$\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0

Table 1.8: Correlation matrix of quasigroup with lexicographic order 211

h	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1	0	0	0	0	0	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
01	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$
10	0	0	0	1	1	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
11	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$

Table 1.9: Prop ratio table of quasigroup with lexicographic order 211

One can see that there exists a nonzero output selection vector (01) that is correlated only to one input selection vector (1011) with correlation -1. This means that the second bit y_1 of the output can be represented by affine function from the input bits, i.e., $y_1 = 1 \oplus x_0 \oplus x_2 \oplus x_3$. So, every correlation of 1 or -1 give us immediately the appropriate component Boolean function of quasigroup, if appropriate output selection vector selects only one bit.

In the prop ratio table there are 3 nontrivial difference propagations with prop ratio 1 and restriction weight of 0. The input difference 0011 always propagates to output difference 10, 0100 always propagates to output difference 10 and the input difference 0111 always propagates to output difference 00. For example, the input difference 0011 is for the pairs: 0*0 = 1 and 0*3 = 3; 0*1 = 2 and 0*2 = 0; 1*0 = 3 and 1*3 = 1; 1*1 = 0 and 1*2 = 2; 2*0 = 0 and 2*3 = 2; 2*1 = 1 and 2*2 = 3; 3*0 = 2 and 3*3 = 0; and 3*1 = 3 and 3*2 = 1. Their output difference is 10. □

We examined correlation matrices and prop ratio tables of quasigroup of order 4 in Mileva and Markovski [100]. There are 144 out of 576 quasigroups of order 4 that have a prop ratio table with all nontrivial difference propagations with prop ratio 1 and restriction weight of 0, and correlation matrix with every nonzero output selection vector correlated only to one input selection vector with correlation 1. Clearly, they correspond to the set of linear quasigroups from classification of [36].

According to obtained correlation matrices, quasigroups can be divided to:
1. *totally correlated quasigroups* - when every nonzero output selection vector is correlated to only one input selection vector with correlation coefficient 1 or -1
2. *correlated quasigroups* - when at least one nonzero output selection vector is correlated to only one input selection vector with correlation coefficient 1 or -1
3. *non-correlated quasigroups* - when every nonzero output selection vector is correlated to more than one input selection vector.

Remark 3 From 576 quasigroup of order 4, 144 are totally correlated quasigroups (the same as linear quasigroups) and 432 are correlated quasigroups. □

According to obtained prop ratio tables, quasigroups can be divided to:
1. *non-restricted quasigroups* - when all nontrivial difference propagations are of prop ratio 1
2. *weak restricted quasigroups* - when at least one nontrivial difference propagation is of prop ratio 1

3. *restricted quasigroups* - when there is no nontrivial difference propagations of prop ratio 1.

Remark 4 From 576 quasigroup of order 4, 144 are non-restricted quasigroups (the same as linear quasigroups) and 432 are weak restricted quasigroups. □

From the previous considerations, it follows that the linear quasigroups are totally correlated and non-restricted quasigroups and vice versa.

1.3.4 Correlation matrices and prop ratio tables of quasigroup transformations

First, we proof the following proposition.

Proposition 7 *The transformations e_l, d_l, e'_l and d'_l produced by a linear quasigroup are linear functions.* □

Proof Let $(Q, \circ)$ be a linear quasigroup [36] of order $r = 2^n$. Then for all $x, y, z \in Q$, with binary representations $(x_1, \dots, x_n)$ of x and $(y_1, \dots y_n)$ of y we have

$$z = x \circ y = (\sum \alpha_i^{(1)} x_i + \sum \beta_i^{(1)} y_i, \dots, \sum \alpha_i^{(n)} x_i + \sum \beta_i^{(n)} y_i)$$

where $\alpha_i^{(k)}$ and $\beta_i^{(k)}$ are 1 or 0 for each $i, k \in \{1, 2, \dots, n\}$. For $l, a^1, \dots, a^s \in Q$ we have

$$e_l(a^1 \dots a^s) = z^1 \dots z^s, \quad d_l(a^1 \dots a^s) = u^1 \dots u^s.$$

Let $\alpha_{ri}^{(k)}$ $\beta_{ri}^{(k)}$, $\delta_{ri}^{(k)}$ and $\lambda_{ri}^{(k)}$ be 1 or 0 for each $i, k \in \{1, 2, \dots, n\}$ and each $r \in \{1, 2, \dots, s\}$. For each $j \in \{2, \dots s\}$ we have

$$z^1 = l \circ a^1 = (\sum \alpha_{1i}^{(1)} l_i + \sum \beta_{1i}^{(1)} a_i^1, \dots, \sum \alpha_{1i}^{(n)} l_i + \sum \beta_{1i}^{(n)} a_i^1) =$$

$$(z_1^1, \dots, z_n^1),$$

$$z^j = z^{j-1} \circ a^j = (\sum \alpha_{ji}^{(1)} z_i^{j-1} + \sum \beta_{ji}^{(1)} a_i^j, \dots, \sum \alpha_{ji}^{(n)} z_i^{j-1} + \sum \beta_{ji}^{(n)} a_i^j) =$$

$$(z_1^j, \dots, z_n^j),$$

$$u^1 = l \circ a^1 = (\sum \delta_{1i}^{(1)} l_i + \sum \lambda_{1i}^{(1)} a_i^1, \dots, \sum \delta_{1i}^{(n)} l_i + \sum \lambda_{1i}^{(n)} a_i^1) =$$

$$(u_1^1, \dots, u_n^1),$$

$$u^j = a^{j-1} \circ a^j = (\sum \delta_{ji}^{(1)} a_i^{j-1} + \sum \lambda_{ji}^{(1)} a_i^j, \dots, \sum \delta_{ji}^{(n)} a_i^{j-1} + \sum \lambda_{ji}^{(n)} a_i^j) =$$

$$(u_1^j, \dots, u_n^j)$$

So, inductively we have that every bit in $e_l(a^1 \dots a^s)$ and $d_l(a^1 \dots a^s)$ is obtained by linear Boolean function, therefore e_l and d_l are linear vector valued Boolean functions. Similarly, we can proof that e_l' and d_l' are linear vector valued Boolean functions. ■

Composition of linear functions is also a linear function, so the following corollary is true.

Corollary 2 *The transformations E, D, E', D' and T produced by a linear quasigroup are linear functions.* □

We investigate the behavior of transformations E, D, $\mathcal{A}_l$ and $\mathcal{RA}_l$ produced by all quasigroups of order 4, on strings of length $t = 2$ and $t = 3$. The transformations E and D are compositions of s elementary quasigroup transformations, where $1 \leqslant s \leqslant 100$. We use fixed leader l for all composite transformations, which is the worst case. All of these transformations can be represented as vector valued Boolean functions $\{0,1\}^4 \to \{0,1\}^4$ for $t = 2$ and $\{0,1\}^6 \to \{0,1\}^6$ for $t = 3$. As a tools we use the prop ratio tables and correlation matrices of quasigroup transformations. The results are summarized in [98].

Example 6 The representation of the transformation $E_{l=2,s=5,t=2}$, produced by quasigroup 231, as vector valued Boolean function is given in Table 1.10, where integer representation is used.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$BE_{l=2,s=5,t=2}(x)$	1	2	3	0	6	7	4	5	8	9	10	11	15	12	13	14

Table 1.10: Vector valued Boolean representation of $E_{l=2,s=5,t=2}$

The correlation matrix and prop ratio table for $E_{l=2,s=5,t=2}$ are given in Table 1.11 and Table 1.12, respectfully.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0
2	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
3	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0
6	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$
7	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
10	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
11	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0
15	0	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0

Table 1.11: Correlation matrix of transformation $E_{l=2,s=5,t=2}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0
5	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0
7	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0
9	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0
11	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$

Table 1.12: Prop ratio table of the transformation $E_{l=2,s=5,t=2}$

One can see from the correlation matrix that there exist 7 nonzero output selection vectors that are correlated only to one input selection vectors. Output selection vectors $0001 = 1,\ 0100 = 4$ and $1000 = 8$ are correlated with input selection vectors 1101, 0100 and 1000, respectfully, with correlation coefficient -1, 1 and 1. This means that this transformation has 2 linear and 1 affine component Boolean functions, i.e., $y_1 = x_1,\ y_0 = x_0$ and $y_3 = 1 \oplus x_0 \oplus x_1 \oplus x_3$. □

We obtain several interesting results from our numerical experiments. First, we can divide quasigroups of order 4 in 5 classes according to linearity of produced $E_{l,s,2}$ transformations on strings with length 2, $s \leqslant 100$. We already see that linear quasigroups produce linear E and D transformations, so the class G_0 consists of those quasigroups. Our experiments show us that all quasigroups of order 4 can produce linear $E_{l,s,2}$ transformations, but for some choices of the leader l. There are 48 quasigroups that form the class G_1, with property to produce linear $E_{l,s,2}$-transformations, independently from chosen leader and for every $s = 2k$ and another 16 quasigroups that form the class G_2, with same property but for $s = 4k$. Classes G_0 and G_1 together form the set of fractal quasigroups [24]. Class G_3 consist of 80 quasigroups, with property to produce linear $E_{l,s,2}$-transformations for at least one leader and for every $s = 2k$. The last class G_4 of 288 quasigroups, has the property to produce linear $E_{l,s,2}$ transformations, independently from chosen leader for some $s = \{6k, 8k, 9k, 12k, 24k\}$ and with only 3 nonzero output selection vectors that are correlated only to one input selection vectors. 240 quasigroups from this class produce $E_{l,s,2}$ transformations with maximal prop ratio of $\frac{1}{2}$ or $\frac{3}{4}$ for most of the leaders. Other quasigroups produce $E_{l,s,2}$ transformations with maximal prop ratio of 1.

$G_1 = \{2, 3, 5, 7, 9, 18, 25, 28, 49, 63, 121, 144, 145, 148, 170, 171, 174, 176, 178, 185, 218, 232, 242, 263, 314, 335, 345, 359, 392, 399, 401, 403, 406, 407, 429, 432, 433, 456, 514, 528, 549, 552, 559, 568, 570, 572, 574, 575\}$
$G_2 = \{8, 10, 15, 19, 173, 183, 186, 187, 390, 391, 394, 404, 558, 562, 567, 569\}$
$G_3 = \{6, 12, 13, 16, 17, 20, 22, 23, 35, 36, 47, 48, 50, 64, 69, 72, 122, 131, 134, 143, 155, 156, 167, 168, 175, 177, 180, 181, 184, 188, 190, 191, 217, 231, 233, 236, 241, 251, 254, 264, 313, 323, 326, 336, 341, 344, 346, 360, 386, 387, 389, 393, 396, 397, 400, 402, 409, 410, 421, 422, 434, 443, 446, 455, 505, 508, 513, 527, 529, 530, 541, 542, 554, 555, 557, 560, 561, 564, 565, 571\}$
$G_4 = \{29, 30, 31, 32, 33, 34, 38, 39, 41, 42, 44, 45, 52, 53, 55, 56, 58, 59, 61, 62, 65, 66, 67, 68, 73, 74, 75, 76, 78, 79, 81, 84, 85, 86, 87, 88, 89, 90, 91, 94, 95, 96, 97, 98, 99, 102, 103, 104, 105, 106, 107, 108, 109, 112, 114, 115, 117, 118, 119, 120, 123, 124, 125, 128, 129, 130, 135, 136, 137, 140, 141, 142, 149, 150, 151, 152, 153, 154, 158, 159, 161, 162, 164, 165, 193, 194, 195, 198, 199, 200, 201, 202, 204, 205, 207, 208, 209, 210, 211, 214, 215, 216, 219, 220, 221, 224, 225, 226, 227, 230, 237, 238, 239, 240, 244, 245, 247, 248, 249, 250, 255, 256, 257, 258, 260, 261, 265, 266, 267, 268, 270, 271, 273, 276, 277, 278, 279, 280, 281, 282, 283, 286, 287, 288, 289, 290, 291, 294, 295, 296, 297, 298, 299, 300, 301, 304, 306, 307, 309, 310, 311, 312, 316, 317, 319, 320, 321, 322, 327, 328, 329, 330, 332, 333, 337, 338, 339, 340, 347, 350, 351, 352, 353, 356, 357, 358, 361, 362, 363, 366, 367, 368, 369, 370, 372, 373, 375, 376, 377, 378, 379, 382, 383, 384, 412, 413, 415, 416, 418, 419, 423, 424, 425, 426, 427, 428, 435, 436, 437, 440, 441, 442, 447, 448, 449, 452, 453, 454, 457, 458, 459, 460, 462, 463, 465, 468, 469, 470, 471, 472, 473, 474, 475, 478, 479, 480, 481, 482, 483, 486, 487, 488, 489, 490, 491, 492, 493, 496, 498, 499, 501, 502, 503, 504, 509, 510, 511, 512, 515, 516, 518, 519, 521, 522, 524, 525, 532, 533, 535, 536, 538, 539, 543, 544, 545, 546, 547, 548\}$

Quasigroups of class G_1 produce linear $E_{l,s,3}$ transformations on strings with length 3, for every $s = 4k$, $s \leqslant 100$, independently from chosen leader. Quasigroups of classes G_2 and G_3 produce linear $E_{l,s,3}$ transformations for every $s = 8k$, independently from chosen leader. Quasigroups of class G_4 produce linear $E_{l,s,2}$ transformations for some $s = \{24k, 27k, 48k, 54k, 72k\}$, independently from chosen leader. This class is the only class that produce

$E_{l,s,3}$ transformations with maximal prop ratio not equal always to 1 (the least value is $\frac{3}{8}$).

For $D_{l,s,2}$ and $D_{l,s,3}$ transformations, $s \leqslant 100$, we do not obtain any linear transformation for any choice of the leader and any nonlinear quasigroups of order 4. They all produce correlation matrices with 7 ($t = 2$) and 15 ($t = 3$) nonzero output selection vectors that are correlated only to one input selection vectors and prop ratio tables with maximal prop ratio of 1.

All produced non-linear $D_{l,s,2}$, $D_{l,s,3}$, $E_{l,s,2}$ and $E_{l,s,3}$ transformations by quasigroups of order 4 have at least one linear component polynomial in their ANF.

These experiments and Proposition 1 are enough to conclude that E and D transformations preserve the linearity of used quasigroups. Even more, for small strings and for some choices of the leader string, the transformation E increases the linearity in the sense that beside the fact that used quasigroup is nonlinear, the produced transformation can be linear. This is not the case with D transformation. We can conclude also that non-linear E transformations have better propagation characteristics (smaller maximal prop ratio), with less correlation between their input and output, then D transformations from the same quasigroups. Note that we have investigated the worst case - when the leader is fixed for all composite quasigroup transformations.

We also take the shapeless quasigroup of order 8 from [100] and investigate $E_{l,s,2}$ and $D_{l,s,2}$ transformations, for $s \leqslant 100$, on strings with length 2, for different choices of the fixed leader. Obtained $E_{l,s,2}$ transformations have the least maximal absolute correlation coefficient of 0.5, and the least maximal prop ratio of $\frac{7}{32}$. All obtained $D_{l,s,2}$ transformations have maximal absolute correlation coefficient of 1 and the maximal prop ratio of $\frac{5}{16}$. In this set of $E_{l,s,2}$ and $D_{l,s,2}$ transformations, there are functions without any linear component polynomial in their ANF. Number of composite quasigroup transformations do not influences the correlation coefficients and the prop ratios in a sense that they do not decrease with it, but they vary in some range of values.

From this, one can see, that even for smaller strings, taking shapeless quasigroups with higher order decrease the maximal absolute correlation coefficient and maximal prop ratio table of produced E transformation, regardless the number of composite quasigroup transformations. Length of the string additionally put bigger confusion and diffusion property on the same transformations.

Example 7 The correlation matrix and prop ratio table for elementary

$\mathcal{A}_1$-transformation from Example 1 are given in Table 1.13 and Table 1.14. $\square$

$\mathcal{A}_1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$
2	0	$-\frac{1}{4}$	0	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$
3	0	$-\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$-\frac{1}{2}$
4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	0	0	0	$-\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0	0	$\frac{1}{2}$
6	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	0
7	0	$\frac{1}{4}$	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0
8	0	0	0	0	$-\frac{1}{2}$	0	0	0	$-\frac{1}{2}$	0	0	0	$-\frac{1}{2}$	0	0	0
9	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$
10	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
11	0	$-\frac{1}{2}$	$-\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$-\frac{1}{4}$
12	0	0	0	0	$-\frac{1}{2}$	0	0	0	$-\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0
13	0	0	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	0	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$
14	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$-\frac{1}{4}$
15	0	0	$\frac{1}{4}$	$-\frac{1}{4}$	0	0	$-\frac{1}{4}$	$\frac{1}{4}$	0	0	$-\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{2}$	$-\frac{1}{4}$	$\frac{1}{4}$

Table 1.13: Correlation matrix of $\mathcal{A}_1$ transformation for quasigroup 231

$\mathcal{A}_1$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0	0	0	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{4}$
1	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0
2	0	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$
3	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0
4	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0
5	0	0	0	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	0	0
6	0	0	0	0	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	0	0
7	0	0	0	0	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$
9	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{4}$
11	0	0	0	0	0	0	0	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0
12	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0
13	0	0	0	0	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0
14	0	0	0	0	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	0	0
15	0	0	0	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	0	0

Table 1.14: Prop ratio table of $\mathcal{A}_1$ transformation for quasigroup 231

With numerical experiments for $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations on strings of length 2, we obtain very interesting results. Because these transformations are not bijections, first, we investigate the case of producing constant functions. 24 quasigroups of order 4 produce constant functions with $\mathcal{A}_l$ and

$\mathcal{RA}_l$ transformations, independently from the chosen leader. These quasigroups have the structure - every next row is obtained from the previous one by rotating to the right by one position. In addition, it is not important the quasigroup to be linear, with or without some component linear polynomial in its ANF (8 quasigroups are without any linear component polynomial). We examine also $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations with this group of quasigroups on bigger strings, with length up to 10, and obtain again constant functions. We take several quasigroups of order 8 with this kind of structure and they produce constant $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations on strings of length 2 and 3.

Another 88 quasigroups produce constant functions for some choice of the leader.

24 non-linear quasigroups produce linear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations, independently from the chosen leader (again 8 quasigroups are without any linear component polynomial). They also have some structure - every next row is obtained from the previous one by rotating to the left by one position. We examine also $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations with this group of quasigroups on strings with length 3 and 4, and obtain again linear functions.

Another two sets of 86 quasigroups produce only linear $\mathcal{A}_l$ transformations or only linear $\mathcal{RA}_l$ transformations, independently from the chosen leader. Another 78 quasigroups produce linear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations for some choice of the leader.

At the end, 120 quasigroups produce nonlinear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations, independently from the chosen leader, and here structure of quasigroups is again different (7 are linear and 38 quasigroups are without any linear component polynomial). All these transformations have maximal absolute value of the correlation coefficient of 1 and dependently of the leader, maximal prop ratio is 1 for the linear, and $\frac{1}{2}$ for nonlinear quasigroups.

For the nonlinearity of $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations, nonlinearity of quasigroup is not important, but some other structural properties of quasigroups must be investigate. Linear quasigroups can produce nonlinear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations, and vice versa, linear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations can be produced by nonlinear quasigroups. Secondly, we can make a hypothesis that quasigroups with structure - next row to be the previous one, rotated to the right by one position, produce constant functions, independently of the choice of the leader, length of the string or order of the quasigroup. Also, quasigroups of order 4 with structure - next row to be the previous one, rotated to the left by one position produce linear $\mathcal{A}_l$ and $\mathcal{RA}_l$ transformations.

1.3.5 Perfect quasigroups

In a quasigroup based cryptography you can find that different authors seek quasigroups with different properties. One needs $CI-$quasigroups, other needs multivariate quadratic quasigroups, third needs quasigroups with less possible structure, fourth need exponential quasigroups, fifth need orthogonal quasigroups etc. There are special cryptosystems build on some particular subsets of quasigroups. Our interest is to find what properties should have a quasigroup, so that it can be used as non-linear building block in cryptographic primitives and it can contribute to the defence against linear and differential attacks. When we try to find quasigroups suitable for cryptography in this sense, we started from shapeless quasigroups, defined by Gligoroski et al. [43].

Definition 24 [43] A quasigroup $(Q, *)$ of order r is said to be **shapeless** iff it is non-idempotent, non-commutative, non-associative, it does not have neither left nor right unit, it does not contain proper sub-quasigroups, and there is no $k < 2r$ for which identities of the kinds are satisfied:

$$\underbrace{x(\ldots * (x}_{k} *y)) = y, \;\; y = ((y * \underbrace{x) * \ldots) * x}_{k} \tag{1.23}$$

□

Shapeless quasigroups are good choice, but sometimes even a quasigroup with some structure is preferable (when structure does not affect the security). In other cases quasigroups with additional restriction to the structure may be needed, for example, not to be either semisymmetric or Stein quasigroup or Schroeder quasigroup, etc. In the light of the recent linear and differential attacks we are going to extend the notation of shapeless quasigroups to perfect quasigroups.

Definition 25 A quasigroup $(Q, *)$ of order r is said to be **perfect** if it is pure non-linear, non-correlated and restricted shapeless. □

The quasigroup of Example 1 is shapeless, but is not perfect, because it is correlated, weak-restricted and weak non-linear.

Example 8 The quasigroup isotopic to the group $(\mathbb{Z}_8, +)$ with isotopism $(id_{\mathbb{Z}_8}, \beta, \gamma)$, where $\beta : \begin{pmatrix} 01234567 \\ 31652740 \end{pmatrix}$ and $\gamma : \begin{pmatrix} 01234567 \\ 03571624 \end{pmatrix}$ is a perfect quasigroup. □

About the question what kind of quasigroups to use for quasigroup transformations, it is very important how we want to apply them and where we want to apply them. Most often quasigroups are used for creating quasigroup transformations, and for them, usually it is enough quasigroup to be shapeless. Stronger requirement is the quasigroup to be perfect and this is needed especially in the cases when we use quasigroup alone (not for quasigroup transformation). Some quasigroup transformations, like $\mathcal{A}$ and $\mathcal{RA}$, even defined by linear quasigroups, can produce non-linear Boolean functions. Some quasigroup transformation, like E transformation, preserve linearity of used quasigroup. At the end, it is important quasigroup string transformations to be non-linear vector valued Boolean functions without any linear component Boolean function, without nontrivial difference propagations with prop ratio 1 and restriction weight of 0 and with every nonzero output selection vector correlated to more than one input selection vector. We showed by examples, that even the quasigroups of order 4 can produce this kind of quasigroup transformations.

Some cryptographic primitives need special kind of quasigroups. For example, when the period of produced sequences is important, like for PRNGs and stream ciphers, quasigroup must be exponential.

1.4 Summary

This chapter has been devoted to quasigroups and quasigroup transformations. Our own contributions in this chapter are:

– new method for enumeration of n-ary quasigroups of small order and revision of number of isotopy classes for ternary quasigroups of order 4

– examination of prop ratio tables and correlation matrices for quasigroups of order 4

– augmentation of existing classification of quasigroups as a vector valued Boolean functions

– new classifications of quasigroups according to their correlation matrices and prop ratio tables

– notation of elementary and composite quasigroup transformations

– new $\mathcal{A}, \mathcal{RA}, M, \mathcal{MT}$ and OT quasigroup transformations

– examination of prop ratio tables and correlation matrices of E, D, $\mathcal{A}$ and $\mathcal{RA}$ quasigroup transformations for quasigroups of order 4 and for small strings

– notation of perfect quasigroups.

There are several open problems, that remain to be solved, like, how to represent quasigroups of order $r \neq 2^n$ as vector valued Boolean functions, examination on their prop ratio tables and correlation matrices, etc.

Chapter 2

Generation of huge quasigroups

In this chapter, we examine several well-known ways and one new way of constructing quasigroups, specially huge quasigroups. First, we consider several methods of producing larger quasigroups from smaller ones. Than we consider several methods that incorporate permutations, polynomials, T-functions etc. to In what follows therefore we are going to introduce the so called extended Feistel networks, which are Feistel networks with additional properties, to define huge quasigroups. A Feistel network [32] takes any function and transforms it into a bijection, so it is commonly used technique for creating a non-linear cryptographic function [142], [69]. Using a Feistel network for creating a huge quasigroup is not a novel approach. Kristen [97] presents several different constructions using one or two Feistel networks and isotopies of quasigroups. Complete mappings, introduced by Mann [72] (the equivalent concept of orthomorphism was introduced explicitly in [27]), are also useful for creation of huge quasigroups. In [97] complete mappings with non-affine functions represented by Cayley tables or with affine functions represented by binary transformations, are used for that aim. The main disadvantages of the previously mentioned constructions are the lack of efficiency in one case and the lack of security in the other case. Namely, the Cayley table representations need a lot of memory, and also the affine functions don't have good cryptographic properties.

Our approach use the extended Feistel networks as orthomorphisms, to generate huge quasigroups of order $r = 2^{s2^t}$. We only need to store small permutations of order 2^s, $s = 4, 8, 16$. We show that the quasigroups obtained by our construction can have different properties, and on some of them we can influent by choosing bijection or parameters. We examine quasigroups obtained by this method on a group $(\mathbb{Z}_n, \oplus_n)$ and we prove that they can not be perfect quasigroups, but only shapeless. Quasigroups, produced by the extended Feistel networks $F_{A,B,C}$ defined on Abelian group $(\mathbb{Z}_n, \oplus_n)$, are weak-restricted, correlated and weak non-linear, but those produced by $F^2_{A,B,C}$ are much better. They are non-correlated and pure non-linear, but still weak-restricted.

2.1 Direct, semidirect and quasidirect product

One way of producing larger quasigroups from smaller ones, is the direct product of quasigroups. Let $(Q_1, \circ)$ and $(Q_2, \cdot)$ be two quasigroups of order r_1 and r_2, respectively. The *direct product* $(Q_1 \times Q_2, \otimes)$ of these quasigroups, defined by

$$(a_1, b_1) \otimes (a_2, b_2) = (a_1 \circ a_2, b_1 \cdot b_2)$$

where $a_1, a_2 \in Q_1$ and $b_1, b_2 \in Q_2$, is a quasigroup of order $r_1 r_2$. One way of representing the direct product is this - each element (a, b) of $Q_1 \times Q_2$ can be mapped with integer representation of concatenation of binary representations of a and b.

Example 1. Let $Q_1 = \{0, 1, 2, 3\}$ and $Q_2 = \{0, 1\}$. On the following Table one can see two quasigroups $(Q_1, \circ)$ and $(Q_2, \cdot)$ with order 4 and 2 respectively, and quasigroup of order 8, obtained from their direct product $(Q_1 \times Q_2, \otimes)$, with previous representation. This correlated and weak restricted quasigroup is not shapeless, because has left unit 0, also has a proper subquasigroup and the pair $(4, 12)$ satisfy 1.23.

$\circ$	0	1	2	3
0	0	1	2	3
1	1	3	0	2
2	3	2	1	0
3	2	0	3	1

$\cdot$	0	1
0	0	1
1	1	0

$\otimes$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	6	7	0	1	4	5
3	3	2	7	6	1	0	5	4
4	6	7	4	5	2	3	0	1
5	7	6	5	4	3	2	1	0
6	4	5	0	1	6	7	2	3
7	5	4	1	0	7	6	3	2

Table 2.1: The integer representation of direct product $(Q_1 \times Q_2, \otimes)$

There are several generalizations of this approach, as semidirect product and quasidirect product of quasigroups. The *semidirect product* $(Q_1 \times Q_2, \otimes)$

of two quasigroups $(Q_1, \circ)$ and $(Q_2, \cdot)$, is defined by

$$(a_1, b_1) \otimes (a_2, b_2) = (f_{b_1,b_2}(a_1 \circ a_2), b_1 \cdot b_2)$$

where $a_1, a_2 \in Q_1$, $b_1, b_2 \in Q_2$ and f_{b_1,b_2} are permutations on set Q_1.

Example 2. Let $(Q_1, \circ)$ and $(Q_2, \cdot)$ be quasigroups from previous example. Let $f_{b_1,b_2}(x) = (b_1 + 3b_2 + x)$ $(mod\ 4)$ be permutations on Q_1. On the Table 2.2 is given the semidirect product $(Q_1 \times Q_2, \otimes)$ (with a previous binary representation). This quasigroup is shapeless, but correlated and weak restricted.

$\otimes$	0	1	2	3	4	5	6	7
0	2	1	4	3	6	5	0	7
1	5	2	7	4	1	6	3	0
2	4	3	0	7	2	1	6	5
3	7	4	3	0	5	2	1	6
4	0	7	6	5	4	3	2	1
5	3	0	1	6	7	4	5	2
6	6	5	2	1	0	7	4	3
7	1	6	5	2	3	0	7	4

Table 2.2: The integer representation of semidirect product $(Q_1 \times Q_2, \otimes)$

A more general approach given by Bruck [8] and named by Wilson [141] as *quasidirect product* of quasigroups is defined as

$$(a_1, b_1) \otimes (a_2, b_2) = (a_1 \nabla_{b_1,b_2} a_2, b_1 \cdot b_2)$$

where (Q_1, ∇_{b_1,b_2}) are quasigroups for all $b_1, b_2 \in Q_1$.

Example 3. We use quasigroups $(Q_1, \circ)$ and $(Q_2, \cdot)$ from Example 1 again. Quasigroup operations are defined by $a_1 \nabla_{b_1,b_2} a_2 = (-a_1 + a_2 - b_1 + 3b_j)$ $(mod\ 4)$. The quasidirect product $(Q_1 \times Q_2, \otimes)$ is given on the Table 2.3 (with previous binary representation). This quasigroup is correlated and weak restricted, and it is not shapeless only because the pair $(8, 8)$ satisfy 1.23.

2.2 Generalized singular direct product

Let $(Q, \circ)$ be a quasigroup with a subquasigroup $(S, \circ)$ and let (I, ∇) be an idempotent quasigroup. Furthermore let $P = Q \backslash S$ and $(P, \otimes_{v,w})$ be

$\otimes$	0	1	2	3	4	5	6	7
0	0	7	2	1	4	3	6	5
1	7	4	1	6	3	0	5	2
2	6	5	0	7	2	1	4	3
3	5	2	7	4	1	6	3	0
4	4	3	6	5	0	7	2	1
5	3	0	5	2	7	4	1	6
6	2	1	4	3	6	5	0	7
7	1	6	3	0	5	2	7	4

Table 2.3: The integer representation of quasidirect product $(Q_1 \times Q_2, \otimes)$

quasigroups for all ordered pairs $(v, w) \in I \times I, v \neq w$. Sade [122] and Lindner [64] define *generalized singular direct product* $(S \cup (P \times I), \cdot)$ as:

$$x \cdot y = x \circ y$$

$$x \cdot (r, v) = (x \circ r, v)$$

$$(r, v) \cdot y = (r \circ y, v)$$

$$(r, v) \cdot (s, v) = r \circ s, \quad if\ r \circ s \in S$$

$$(r, v) \cdot (s, v) = (r \circ s, v), \quad if\ r \circ s \in P$$

$$(r, v) \cdot (s, w) = (r \otimes_{v,w} s, v \nabla w), \quad if\ v \neq w$$

where $x, y \in S$, $r, s \in P$ and $v, w \in I$. By this construction, new Steiner quasigroups are found which are self-orthogonal. If $|Q| = n, |I| = k$ and $|S| = m$, then $|S \cup (P \times I)| = k(n - m) + m$.

Example 4. Let $Q = \{0, 1, 2, 3\}$, $S = \{0\}$, $I = \{0, 1, 2\}$, $r \otimes_{v,w} s = (-r + s - i + 3j)\ (mod\ 3) + 1$ and $(Q, \circ)$ and (I, ∇) be defined by

$\circ$	0	1	2	3
0	0	1	2	3
1	1	3	0	2
2	3	2	1	0
3	2	0	3	1

∇	0	1	2
0	0	2	1
1	2	1	0
2	1	0	2

The obtained general singular direct product $(S \cup (P \times I), \cdot)$ is given on Table 2.4. This quasigroup is non-shapeless, because it has the left identity element 0, the proper subquasigroup $(Q, \circ)$ and also the pair $(8, 8)$ satisfy 1.23.

If $\otimes_{v,w}$ is the same operation for all $v, w \in I, v \neq w$, the operation $\cdot$ is the singular direct product, and if additionally $S = \emptyset$ and $\otimes_{v,w} = \circ$, we have the usual direct product.

$\cdot$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	3	0	2	7	8	9	4	5	6
2	3	2	1	0	9	7	8	6	4	5
3	2	0	3	1	8	9	7	5	6	4
4	4	9	7	8	6	0	5	3	1	2
5	6	8	9	7	5	4	0	2	3	1
6	5	7	8	9	0	6	4	1	2	3
7	7	5	6	4	2	3	1	9	0	8
8	9	4	5	6	1	2	3	8	7	0
9	8	6	4	5	3	1	2	0	9	7

Table 2.4: The general singular direct product $(S \cup (P \times I), \cdot)$

2.3 Prolongation

One can construct a quasigroup of order $n+1$ from existing quasigroup $(Q, \circ)$ of order n, where the multiplication table of $(Q, \circ)$ possesses a transversal. This method is known as *insertion construction* or *prolongation* (first construction is given by Bruck [9], who considered only the case of idempotent quasigroups).

The classical construction was given by Belousov [3] and is made by adding new element e to Q and adding additional row and column to the Cayley table of a given quasigroup. For each cell in the transversal, the element in the cell is moved in the new column and same row as the cell, and also placed in the new row and same column as the cell. The empty cells of the transversal as well as the empty cell in the right lower corner are filled with e.

Example 5. On Table 2.6 are given quasigroup $(Q, \circ)$ and one of its prolongation, where $Q = \{0, 1, 2\}$.

There is another construction of prolongation of admissible quasigroups given by Belyavskaya [5], and generally, these obtained prolongations are not

$\circ$	0	1	2
0	0	**1**	2
1	1	2	**0**
2	**2**	0	1

$\cdot$	0	1	2	3
0	0	**3**	2	**1**
1	1	2	**3**	**0**
2	**3**	0	1	**2**
3	**2**	**1**	**0**	**3**

Table 2.5: Prolongation of quasigroup $(Q, \circ)$ by classical construction

isotopic to prolongations from the previous method. Deriyenko and Dudek [23] gave another construction of prolongation for any quasigroups of order n with property that their multiplication tables have partial transversals of size $n-1$, which is generalisation of the previous two constructions. The Brualdi conjecture [19] says that each Latin square of order n has a partial transversal of size $n-1$. If this conjecture is true, then with this method, the prolongation can be constructed from every quasigroup. Let a be the element from partial transversal which occurs two times and let d be the missing element. This construction is made by adding new element e to Q and adding additional row and column to the Cayley table of a given quasigroup. For each cell in the partial transversal except the cell with first occurrence of a, the element in the cell is moved in new column and same row as the cell, and also placed in new row and same column as the cell. The empty cells of the partial transversal as well as the empty cell in the row of the first occurrence of a and the new column, and the empty cell in the new row and the column of the first occurrence of a, are filled with e. The empty cell in the right lower corner is filled with d.

Example 6. On Table 2.6 is given quasigroup $(Q, \circ)$ and one of its prolongation by Deriyenko and Dudek method, where $Q = \{0, 1, 2, 3, 4, 5\}$, $a = 1$ and $d = 4$.

2.4 Diagonal method and its modifications

Sade [120] proposed the following construction which is known as *diagonal method.* On $(\mathbb{Z}_n, +)$ let θ be a permutation of the set $\mathbb{Z}_n$, such that $\phi(x) = x - \theta(x)$ is also a permutation. Let $Q = \mathbb{Z}_n$. Define an operation $\circ$ on Q by:

$$x \circ y = \theta(x - y) + y \tag{2.1}$$

where $x, y \in Q$. Then $(Q, \circ)$ is a quasigroup. (Then we say that $(Q, \circ)$ is derived by θ).

Quasigroups which are constructed with the diagonal method possess a decomposition in the disjoint transversals and therefore an orthogonal mate.

$\circ$	0	1	2	3	4	5
0	0	1	5	**2**	4	3
1	1	2	**0**	3	5	4
2	**5**	0	4	1	3	2
3	2	**3**	1	4	0	5
4	4	5	3	0	2	**1**
5	3	4	2	5	**1**	0

$\cdot$	0	1	2	3	4	5	6
0	0	1	5	**6**	4	3	**2**
1	1	2	**6**	3	5	4	**0**
2	**6**	0	4	1	3	2	**5**
3	2	**6**	1	4	0	5	**3**
4	4	5	3	0	2	**1**	**6**
5	3	4	2	5	**6**	0	**1**
6	**5**	**3**	**0**	**2**	**1**	**6**	**4**

Table 2.6: Prolongation of quasigroup $(Q, \circ)$ by Deriyenko and Dudek method

Also for these quasigroups, every translation σ_h, given by $\sigma_h : x \rightarrow x + h$ is an automorphism. Note that if θ works for this method, than the mappings that map x in $x - \theta(x)$, θ^{-1}, $-\theta(-x)$, $x + \theta(-x)$, $\theta(x) + h$ and $\theta(x+h)$ for any h also works. Kristen [97] generalized this construction method for every group $(G, +)$. She incorrectly named those permutations as complete mappings which is different with generally accepted Definition 26 for complete mappings (Paige, Hall, Dénes, Keedwell). These permutations are complete mappings in some special cases, when group $(\mathbb{Z}_2^n, \oplus_n)$ is used. She uses these complete mappings with non-affine functions represented by Cayley tables or with affine functions represented by binary transformations for creating quasigroups. Correct definition of complete mappings follows. In some papers this definition of complete mappings is used for defining orthomorphisms (Johnson et al [53], Mittenthal [102]). In Mittenthal [102] you can find construction of such linear orthomorphisms of the group $(\mathbb{Z}_2^n, \oplus_n)$, and in [48] you can find construction of linear and non-linear orthomorphisms in the finite field $\mathbb{F}_{2^n}$.

Definition 26 [20] A **complete mapping** of a quasigroup (group) $(G, +)$ is a permutation $\phi : G \rightarrow G$ such that the mapping $\theta : G \rightarrow G$ defined by $\theta(x) = x + \phi(x)$ ($\theta = I + \phi$, where I is the identity mapping) is again a permutation of G. The mapping θ is said to be the **orthomorphism** associated to the complete mapping ϕ. A quasigroup (group) G is **admissible** if there is a complete mapping $\phi : G \rightarrow G$.

It is very easy to generalize this method to the complete mappings and the orthomorphisms. The following theorem is very easy to prove.

Theorem 8 *Let ϕ be a complete mapping of the admissible group $(G, +)$ and let θ be an orthomorphism associated to ϕ. Define an operations $\circ$ and $\bullet$ on G by:*

$$x \circ y = \phi(y - x) + y \tag{2.2}$$

$$x \bullet y = \theta(x - y) + y \tag{2.3}$$

where $x, y \in G$. Then $(G, \circ)$ and $(G, \bullet)$ are quasigroups. □

Question about whether or not a group G is admissible, is a subject that has been extensively studied [111, 112, 103]. It is well-known fact that inverse of the complete mapping (orthomorphism) is also a complete mapping (orthomorphism) of Abelian group $(G, +)$ [30].

With each orthomorphism θ one can associate a quasigroup $(G, \circ_\theta)$ defined as $x \circ_\theta y = x + \theta(y)$. Two orthomorphisms θ_1 and θ_2 are *orthogonal* if they produce orthogonal quasigroups $(G, \circ_{\theta_1})$ and $(G, \circ_{\theta_2})$. This is fulfilled if and only if the mapping $\alpha : x \to \theta_1(x) - \theta_2(x)$ is a permutation of G (see [31]). Orthogonality is a symmetric property. Mutually orthogonal orthomorphisms can be used to construct mutually orthogonal quasigroups (or MOLS) from groups. One can notice that, if θ is any orthomorphism then θ is orthogonal to I.

In the sequel, we will consider orthomorphisms (complete mappings) of the Abelian groups $(\mathbb{Z}_2^n, \oplus_n)$. The results of Paige [111] implies that the groups $(\mathbb{Z}_2^n, \oplus_n)$ are admissible. Then the equation (2.3) gets this form:

$$x \circ y = \theta(x \oplus_n y) \oplus_n y. \tag{2.4}$$

Example 7. Let $Q = \mathbb{Z}_2^2 = \{0, 1, 2, 3\}$, where we use the integer notation $0 \equiv \langle 0, 0\rangle, 1 \equiv \langle 0, 1\rangle, 2 \equiv \langle 1, 0\rangle, 3 \equiv \langle 1, 1\rangle$. Define $\theta : Q \to Q$ by $\theta(\langle x_0, x_1\rangle) = \langle x_0 \oplus x_1, x_0 \oplus 1\rangle$, where x_1, x_0 are bits. Table 2.7 demonstrates that both θ and $I \oplus_2 \theta$ are bijections, and the quasigroup $(Q, \circ)$ is defined by (2.4).

x	$\theta(x)$	$\phi(x) = x \oplus_2 \theta(x)$
$\langle 0, 0\rangle$	$\langle 0, 1\rangle$	$\langle 0, 1\rangle$
$\langle 0, 1\rangle$	$\langle 1, 1\rangle$	$\langle 1, 0\rangle$
$\langle 1, 0\rangle$	$\langle 1, 0\rangle$	$\langle 0, 0\rangle$
$\langle 1, 1\rangle$	$\langle 0, 0\rangle$	$\langle 1, 1\rangle$

$\circ$	0	1	2	3
0	1	2	0	3
1	3	0	2	1
2	2	1	3	0
3	0	3	1	2

Table 2.7: The complete mapping (orthomorphism) θ of the group $\mathbb{Z}_2^2$ and the derived quasigroup $(Q, \circ)$

The next theorem shows that if a quasigroup $(\mathbb{Z}_2^n, \circ)$ derives from diagonal method or its modifications, then all of its parastrophes can be derived by orthomorphisms (complete mappings) too. This fact can be especially useful for encoding and decoding purposes.

Theorem 9 *Let $\theta : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$ be an orthomorphism (complete mapping) of the group $(\mathbb{Z}_2^n, \oplus_n)$ and let $(\mathbb{Z}_2^n, \circ)$ be the quasigroup, which derives from $x \circ y = \theta(x \oplus_n y) \oplus_n y$. Then the following statements are true.*

a) The quasigroup $(Q, /)$ derives from the orthomorphism (complete mapping) $\delta = \theta^{-1}$.

b) The quasigroup $(Q, \backslash)$ derives from the orthomorphism (complete mapping) $\lambda = (I \oplus_n \theta^{-1})^{-1}$.

c) The quasigroup $(Q, //)$ derives from the orthomorphism (complete mapping) $\rho = I \oplus_n \theta^{-1}$.

d) The quasigroup $(Q, \backslash\backslash)$ derives from by the orthomorphism (complete mapping) $\tau = (I \oplus_n \theta)^{-1}$.

e) The quasigroup $(Q, \cdot)$ derives from the orthomorphism (complete mapping) $\varphi = I \oplus_n \theta$. □

Proof *a)* $x/y = z \Leftrightarrow z \circ y = x \Leftrightarrow$
$\theta(z \oplus_n y) \oplus_n y = x \Leftrightarrow z \oplus_n y = \theta^{-1}(x \oplus_n y) \Leftrightarrow$
$z = \theta^{-1}(x \oplus_n y) \oplus_n y,$
and that implies $x/y = \delta(x \oplus_n y) \oplus_n y$.

b) $x\backslash y = z \Leftrightarrow x \circ z = y \Leftrightarrow$
$\theta(x \oplus_n z) \oplus_n z = y \Leftrightarrow x \oplus_n z = \theta^{-1}(y \oplus_n z) \Leftrightarrow$
$x = \theta^{-1}(y \oplus_n z) \oplus_n z \oplus_n y \oplus_n y \Leftrightarrow x \oplus_n y = \theta^{-1}(y \oplus_n z) \oplus_n y \oplus_n z \Leftrightarrow$
$x \oplus_n y = (I \oplus \theta^{-1})(y \oplus_n z) \Leftrightarrow (I \oplus_n \theta^{-1})^{-1}(x \oplus_n y) = y \oplus_n z \Leftrightarrow$
$(I \oplus_n \theta^{-1})^{-1}(x \oplus_n y) \oplus_n y = z,$
and that implies $x\backslash y = \lambda(x \oplus_n y) \oplus_n y$.

c) $x//y = z \Leftrightarrow y/x = z \Leftrightarrow z \circ x = y \Leftrightarrow$
$\theta(z \oplus_n x) \oplus_n x = y \Leftrightarrow z \oplus_n x = \theta^{-1}(x \oplus_n y) \Leftrightarrow$
$z = \theta^{-1}(x \oplus_n y) \oplus_n x \oplus_n y \oplus_n y \Leftrightarrow$
$z = (I \oplus_n \theta^{-1})(x \oplus_n y) \oplus_n y,$
and that implies $x//y = \rho(x \oplus_n y) \oplus_n y$.

d) $x\backslash\backslash y = z \Leftrightarrow y\backslash x = z \Leftrightarrow y \circ z = x \Leftrightarrow$
$\theta(y \oplus_n z) \oplus_n z = x \Leftrightarrow z \oplus_n y \oplus_n \theta(z \oplus_n y) = x \oplus_n y \Leftrightarrow$
$(I \oplus_n \theta)(z \oplus_n y) = x \oplus_n y \Leftrightarrow z \oplus_n y = (I \oplus_n \theta)^{-1}(x \oplus_n y) \Leftrightarrow$
$z = (I \oplus_n \theta)^{-1}(x \oplus_n y) \oplus_n y,$
and that implies $x\backslash\backslash y = \tau(x \oplus_n y) \oplus_n y$.

e) $x \cdot y = z \Leftrightarrow y \circ x = z \Leftrightarrow$
$\theta(y \oplus_n x) \oplus_n x = z \Leftrightarrow \theta(x \oplus_n y) \oplus_n x \oplus_n y \oplus_n y = z \Leftrightarrow$

$(I \oplus_n \theta)(x \oplus_n y) \oplus_n y = z$,
and that implies $x \cdot y = \varphi(x \oplus_n y) \oplus_n y$. ■

2.5 T-functions

A **T-function** (T is short for triangular) is a mapping from n bit input to n bit output, in which the i^{th} bit of the output can depend only on bits $0, 1, \ldots, i$ of the input (Klimov and Shamir [59]). This definition can be naturally extended to functions that map several n-bit inputs to several n-bit outputs. All the boolean operations and most of the arithmetical operations available on modern processors are T-functions, and also their compositions are T-functions. Circular rotations and right shifts are not T-functions.

In [58], Klimov and Shamir noted that in order to use T-function f to define a quasigroup operation, f needs to be invertible. Also in [59], they showed that if f is a T-function, the mappings $v : x \rightarrow x + 2 \cdot f(x)\ mod\ 2^n$ and $u : x \rightarrow x + (x^2 \vee 1)\ mod\ 2^n$ are invertible T-functions.

One way of creating a quasigroup based on a T-function is given in [97] and quasigroups obtained by this way, have the structure such as entries in each row and each column alternate between even and odd numbers.

Proposition 8 *Let $Q = \mathbb{Z}_2^n$ and let $f : Q \times Q \rightarrow Q$ be a* T*-function. Define an operation $\circ$ on Q by:*

$$x \circ y = c + x + y + 2f(x, y)\ mod\ 2^n \tag{2.5}$$

□

where $c \in Q$. Then $(Q, \circ)$ is a quasigroup. □

Example 8. Let $Q = \mathbb{Z}_2^3$ and let $f : Q \times Q \rightarrow Q$ be given by

$$f(x, y) = 6(\rceil x \vee y) + y^2$$

where addition and multiplication are computed modulo 8, $\rceil$ is negation, $\vee$ is Boolean or. Let $c = 7$. We define quasigroup operation (see Table 2.8) as

$$x \circ y = 7 + x + y + 2(6(\rceil x \vee y) + y^2)\ mod\ 2^3.$$

This quasigroup is non-correlated and weak restricted, and it is not shapeless only because the pair $(4, 8)$ satisfy 1.23. But one can see that every row can be obtained by rotation of every other row.

$\circ$	0	1	2	3	4	5	6	7
0	3	2	5	4	7	6	1	0
1	4	7	6	1	0	3	2	5
2	5	4	7	6	1	0	3	2
7	6	1	0	3	2	5	4	7
4	7	6	1	0	3	2	5	4
5	0	3	2	5	4	7	6	1
6	1	0	3	2	5	4	7	6
7	2	5	4	7	6	1	0	3

Table 2.8: Quasigroup obtained by T-function

2.6 Isotopies

Isotopies are one common way of creating quasigroups, regardless the order of the quasigroup. You can find nice use of isotopies for creating a quasigroups with order 2^m, where $m \in \{224, 256, 384, 512\}$ in hash function Edon-R [46].

For creating huge quasigroups one can use non-linear functions, which are used in cryptography, such as the Feistel networks, the LFSRs and the previous T-functions. Kristen [97] presents several different constructions using two Feistel networks or one Feistel network and odd permutation. She proposes another way of creating odd non-linear permutation by modification of any linear feedback shift functions obtained from irreducible polynomial. Kristen proved also the following two propositions:

Proposition 9 *Let $(Q, \circ)$ be a quasigroup created from an abelian group $(Q, +)$ by*

$$x \circ y = f(x) + g(x)$$

for $x, y \in Q$, where $f, g : Q \to Q$ are bijections. Then

$$\left.\begin{array}{l} a \circ c = x \\ a \circ d = x + z \\ b \circ c = y \end{array}\right\} \Rightarrow b \circ d = y + z \qquad (2.6)$$

□

Proposition 10 *Let $(Q, \circ)$ be a quasigroup created from an abelian group $(\mathbb{Z}_2^k, \oplus)$ by*

$$x \circ y = f(x) \oplus g(x)$$

for $x, y \in Q$, where $f, g : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ are bijections. Then for $a, b, c, d \in Q$

$$a \circ c = b \circ d \Leftrightarrow a \circ d = b \circ c \tag{2.7}$$

□

There are some ”pairing” properties for quasigroup $(Q, \bullet)$ constructed by affine isotopies of a group $(Q, +)$, defined and proved in [97]. First ”pairing” property tells us that every row in multiplication table of $(Q, \bullet)$ is the reversal of another row, and every column of $(Q, \bullet)$ is the reversal of another column. Another ”pairing” property tells us that every element has its ”pair” element that appears next to it in every row and every column in the multiplication table of $(Q, \bullet)$.

Here, we are going to examine the use of one or two T-functions as isotopies for generating huge quasigroups.

If we use construction $v : x \to x + 2 \cdot f(x)\ mod\ 2^n$ for invertible T-functions for both isotopies, quasigroup operation can be defined by

$$x \circ y = v(x) + u(y) = x + 2 \cdot f(x) + y + 2 \cdot g(y)$$

Then it is easy to see that if $x \circ y$ is even, then $x \circ (y + 1)$ and $(x + 1) \circ y$ will be odd and vice versa.

$$x \circ (y + 1) = v(x) + u(y + 1) = x + 2 \cdot f(x) + y + 1 + 2 \cdot g(y + 1)$$

$$(x + 1) \circ y = v(x + 1) + u(y) = x + 1 + 2 \cdot f(x + 1) + y + 2 \cdot g(y)$$

Because $2 \cdot f(\cdot)$ and $2 \cdot g(\cdot)$ are always even, so the parity of $x \circ (y + 1)$ and $(x + 1) \circ y$ will be different than the parity of $x \circ y$.

Example 9. Let $Q = \mathbb{Z}_2^3$ and let quasigroup operation be addition modulo $2^3 = 8$. Let $f, g : Q \to Q$ be two invertible T-functions given by

$$f(x) = x + 2((2x + 3x^2) \vee x)$$

$$g(x) = x + 2(x \vee (3 + x^2))$$

where addition and multiplication are computed modulo 8 and $\vee$ is Boolean or. Let $c = 6$. We define quasigroup operation as (see Table 2.9)

$$x \circ y = f(x) + g(y).$$

This quasigroup is correlated and weak restricted, and is not shapeless only because the pair $(4, 4)$ satisfy the identity 1.23. One can see that each column in this quasigroup can be obtained by rotation of every other column.

$\circ$	0	1	2	3	4	5	6	7
0	6	3	0	1	2	7	4	5
1	1	6	3	4	5	2	7	0
2	4	1	6	7	0	5	2	3
3	7	4	1	2	3	0	5	6
4	2	7	4	5	6	3	0	1
5	5	2	7	0	1	6	3	4
6	0	5	2	3	4	1	6	7
7	3	0	5	6	7	4	1	2

Table 2.9: Quasigroup obtained by isotopies of two T-functions

2.7 Permutation polynomials

A polynomial $P(x) = a_0 + a_1x + \ldots + a_dx^d$ is said to be a **permutation polynomial** over a finite ring R if P permutes the elements of R. Rivest [119] gives the following two Theorems, important for the constructing quasigroups from permutation polynomials.

Theorem 10 *Let $P(x) = a_0 + a_1x + \ldots + a_dx^d$ be a polynomial with integral coefficients. Then $P(x)$ is a permutation polynomial modulo $n = 2^w$, $w \geqslant 2$, if and only if a_1 is odd, $(a_2 + a_4 + a_6 + \ldots)$ is even, and $(a_3 + a_5 + a_7 + \ldots)$ is even.* □

Theorem 11 *A bivariate polynomial $P(x, y) = \sum_{ij} a_{ij}x^iy^j$ represents a latin square modulo $n = 2^w$, $w \geqslant 2$, if and only if the four univariate polynomials $P(x, 0), P(x, 1), P(0, y)$, and $P(1, y)$ are all permutation polynomials modulo n.* □

Example 10. Here is the third-degree polynomial, representing a quasigroup modulo $n = 2^w$:

$$P(x, y) = 2x^2y + 2xy^2 + x + y$$

For $w = 3$ we obtain the following quasigroup, which is associative, commutative, correlated and weak restricted, with unit 0 and without proper subquasigroup.

Markovski et al [83] use polynomial functions of the set $Q_n = \{1, 3, \ldots, 2^n - 1\}$, which is group of units on $\mathbb{Z}_{2^n}$, for constructing huge n-ary quasigroups.

$\circ$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	6	7	4	5	2	3	0
2	2	7	4	1	6	3	0	5
3	3	4	1	2	7	0	5	6
4	4	5	6	7	0	1	2	3
5	5	2	3	0	1	6	7	4
6	6	3	0	5	2	7	4	1
7	7	0	5	6	3	4	1	2

Table 2.10: Quasigroup obtained by permutation polynomial modulo 8

Every polynomial $P(x)$ from the polynomial ring $\mathbb{Z}_{2^n}[x]$ induces a polynomial function $p : \mathbb{Z}_{2^n} \to \mathbb{Z}_{2^n}$ by the evaluation map. Denote by $\mathcal{P}_n$ the set of polynomials in $\mathbb{Z}_{2^n}[x]$ that induce polynomial function on Q_n, denote by $\mathcal{PF}_n$ the set of corresponding polynomial functions on Q_n, denote by $\mathcal{PPF}_n$ the set of permutational polynomial functions on Q_n and denote by $\mathcal{PP}_n$ the set of polynomials inducing such functions. Markovski et al [83] give the following propositions and theorems:

Proposition 11 *Let $P(x) = a_0 + a_1x + \ldots + a_dx^d$ be a polynomial in $\mathbb{Z}_{2^n}[x]$. Then $P(x)$ is in $\mathcal{P}_n$ if and only if the sum of the coefficients $a_0 + a_1 + \cdots + a_d$ is odd.* □

Proposition 12 *Let $P(x) = a_0 + a_1x + \ldots + a_dx^d$ be a polynomial in $\mathcal{P}_n$. Then $P(x)$ is in $\mathcal{PP}_n$ if and only if the sum of the odd indexed coefficients $a_1 + a_3 + a_5 \ldots$ is an odd number.* □

Theorem 12 *Let $p_1, p_2, \ldots p_k$ be permutations in $\mathcal{PPF}_n$. Define a k-ary operation f on Q_n by*

$$f(a1, a2, \ldots, a_k) = p_1(a_1)p_2(a_2)\ldots p_k(a_k)\ (mod\ 2^n) \tag{2.8}$$

Then the k-groupoid (Q_n, f) is a k-ary quasigroup. □

Theorem 13 *Let $p_1, p_2, \ldots p_k$ be permutations in $\mathcal{PPF}_n$. Define a k-ary operation f on $\mathbb{Z}_{2^n}$ by*

$$f(a1, a2, \ldots, a_k) = \hat{p}_1(a_1) + \hat{p}_2(a_2) + \ldots + \hat{p}_k(a_k)\ (mod\ 2^n) \tag{2.9}$$

where

$$\hat{p}_i(a) = \begin{cases} p_i(x), \ x \in Q_n \\ p_i(x+1) - 1, \ x \in \mathbb{Z}_{2^n} \backslash Q_n \end{cases} \tag{2.10}$$

Then the k-groupoid (Q_n, f) is a k-ary quasigroup. □

Theorem 14 *Let $p_1, p_2, \ldots p_k$ and $h_1, h_2, \ldots h_k$ be permutations in $\mathcal{PPF}_n$. Define a k-ary operation f on $\mathbb{Z}_{2^n}$ by*

$$f(a1, a2, \ldots, a_k) = f_{p_1,h_1}(a_1) + f_{p_2,h_2}(a_2) + \ldots + f_{p_k,h_k}(a_k) \ (mod\ 2^n) \tag{2.11}$$

where

$$f_{p_i,h_i}(a) = \begin{cases} p_{(}x), \ x \in Q_n \\ h_i(x+1) - 1, \ x \in \mathbb{Z}_{2^n} \backslash Q_n \end{cases} \tag{2.12}$$

Then the k-groupoid (Q_n, f) is a k-ary quasigroup. □

Example 11. Let $p_1(x) = x + 4x^2 + 12x^3$ and $p_2(x) = 11 + x + 3x^2$ be permutations in $\mathcal{PPF}_4$. Quasigroup defined by

$$f(x, y) = p_1(x)p_2(y) \ (mod\ 2^4)$$

and given on Table 2.11 is correlated and weak restricted, and not shapeless only because the pair $(8, 4)$ satisfy the identity 1.23.

$f(x, y)$	1	3	5	7	9	11	13	15
1	15	9	11	5	7	1	3	13
3	5	3	9	7	13	11	1	15
5	11	13	7	9	3	5	15	1
7	1	7	5	11	9	15	13	3
9	7	1	3	13	15	9	11	5
11	13	11	1	15	5	3	9	7
13	3	5	15	1	11	13	7	9
15	9	15	13	3	1	7	5	11

Table 2.11: Quasigroup obtained by Theorem 12

Quasigroup defined by

$$f(x, y) = \hat{p}_1(x) + \hat{p}_2(y) \ (mod\ 2^4)$$

and given on Table 2.12 is correlated and weak restricted, not shapeless quasigroup with the pair $(16, 16)$ satisfy the identity 1.23.

f	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	15	8	9	10	11	4	5	6	7	0	1	2	3	12	13
1	15	0	9	10	11	12	5	6	7	8	1	2	3	4	13	14
2	8	9	2	3	4	5	14	15	0	1	10	11	12	13	6	7
3	9	10	3	4	5	6	15	0	1	2	11	12	13	14	7	8
4	2	3	12	13	14	15	8	9	10	11	4	5	6	7	0	1
5	3	4	13	14	15	0	9	10	11	12	5	6	7	8	1	2
6	12	13	6	7	8	9	2	3	4	5	14	15	0	1	10	11
7	13	14	7	8	9	10	3	4	5	6	15	0	1	2	11	12
8	6	7	0	1	2	3	12	13	14	15	8	9	10	11	4	5
9	7	8	1	2	3	4	13	14	15	0	9	10	11	12	5	6
10	0	1	10	11	12	13	6	7	8	9	2	3	4	5	14	15
11	1	2	11	12	13	14	7	8	9	10	3	4	5	6	15	0
12	10	11	4	5	6	7	0	1	2	3	12	13	14	15	8	9
13	11	12	5	6	7	8	1	2	3	4	13	14	15	0	9	10
14	4	5	14	15	0	1	10	11	12	13	6	7	8	9	2	3
15	5	6	15	0	1	2	11	12	13	14	7	8	9	10	3	4

Table 2.12: Quasigroup obtained by Theorem 13

It is easy to see that for quasigroup operation defined with Theorem 13 if $f(a_1, a_2, \ldots, a_k)$ is even, then $f(a_1+1, a_2, \ldots, a_k), \ldots, f(a_1, a_2, \ldots, a_k+1)$ are odd and vise versa. The last values differ from $f(a_1, a_2, \ldots, a_k)$ in only one component $\hat{p}_i(a_i)$ and $\hat{p}_i(a_i + 1)$ which have different parity.

2.8 Quasigroups over Abelian groups

One way of constructing quasigroups is the method given by Nosov et al. [108], for construction of the parametric families of quasigroups (Latin squares) over the Abelian groups (this is general case, Nosov in [107] use similar method for constructing quasigroups over a set of Boolean n-tuples).

Let $(G, +)$ be a finite Abelian group and $Q = G^n$ be a direct product of n groups G. Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ be elements of

H. Define the $x \circ y = (z_1, z_2, \dots, z_n)$ by the formulas

$$\begin{array}{l} z_1 = x_1 + y_1 + f_1(p_1(x_1, y_1), \dots, p_n(x_n, y_n)) \\ z_2 = x_2 + y_2 + f_2(p_1(x_1, y_1), \dots, p_n(x_n, y_n)) \\ \vdots \\ z_n = x_n + y_n + f_n(p_1(x_1, y_1), \dots, p_n(x_n, y_n)) \end{array} \tag{2.13}$$

where $p_1, p_2, \dots, p_n$ are functions $G^2 \to G$ and $f_1, f_2, \dots, f_n$ are functions $G^n \to G$.

The functions $f_1, f_2, \dots, f_n$ of variables $p_1, p_2, \dots, p_n$ form a *proper family* if, for any distinct n-tuples $p' = (p'_1, p'_2, \dots, p'_n)$ and $p'' = (p''_1, p''_2, \dots, p''_n)$, there is an index α, $1 \leqslant \alpha \leqslant n$, such as $p'_\alpha \neq p'_\alpha$, while $f_\alpha(p') = f_\alpha(p'')$. Even for small dimensions, the number of proper families (up to permutation of indices) is unknown. In [108] are given some examinations of properness and some examples of proper families and the most important thing is that the following Theorem is proved.

Theorem 15 *Let $(G, +)$ be a finite Abelian group and $Q = G^n$ be a direct product of n groups G. Operation $\circ$ defined by formulas 2.13 is quasigroup operation on the set Q for any functions $p_1, p_2, \dots, p_n$ if and only if the family of functions $(f_1, f_2, \dots, f_n)$ is proper.* □

Example 12. Take $f_1 = const$, $f_2 = f_2(p_1)$, $f_3 = f_3(p_1, p_2), \dots, f_n = f_n(p_1, p_2, \dots, p_{n-1})$. Then these functions, being considered as functions of n variables $p_1, p_2, \dots, p_n$, form a proper family. Such families of functions are called *triangular families.* Let we use finite Abelian group $(\mathbb{Z}_2, \oplus)$, then $Q = \mathbb{Z}_2^n$. Let $p_i(x_i, y_i) = x_i \wedge y_i$ for $1 \leqslant i \leqslant n$. Let $f_1 = 1$ and $f_i = p_1(x_1, y_1) \oplus \dots \oplus p_{i-1}(x_{i-1}, y_{i-1})$ for $2 \leqslant i \leqslant n$. Because defined family $(f_1, f_2, \dots, f_n)$ is proper, operation $\circ$ defined by 2.14 is quasigroup operation.

$$\begin{array}{l} z_1 = x_1 \oplus y_1 \oplus 1 \\ z_2 = x_2 \oplus y_2 \oplus (x_1 \wedge y_1) \\ z_3 = x_3 \oplus y_3 \oplus (x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \\ \vdots \\ z_n = x_n \oplus y_n \oplus (x_1 \wedge y_1) \oplus (x_2 \wedge y_2) \oplus \dots \oplus (x_{n-1} \wedge y_{n-1}) \end{array} \tag{2.14}$$

For $n = 3$, quasigroup $(Q, \circ)$ is given on Table 2.13 and is non-correlated and weak restricted, commutative, with the pair $(4, 4)$ satisfy the identity 1.23.

$\circ$	0	1	2	3	4	5	6	7
0	4	5	6	7	0	1	2	3
1	5	4	7	6	1	0	3	2
2	6	7	5	4	2	3	1	0
3	7	6	4	5	3	2	0	1
4	0	1	2	3	7	6	5	4
5	1	0	3	2	6	7	4	5
6	2	3	1	0	5	4	6	7
7	3	2	0	1	4	5	7	6

Table 2.13: The integer representation of $(Q, \circ)$

2.9 Permutations in the set of $\mathbb{Z}_p^*$

Marnas et al [90] proposed a new way for generating a quasigroups of order $p-1$ where p is a prime, by knowing only the first row in the multiplication table of the quasigroup, which is permutation in the set of $\mathbb{Z}_p^* = \mathbb{Z}_p \backslash \{0\}$. Let the first row is $(a_1, \ldots, a_n)$. The quasigroup operation $\circ$ is defined as $i \circ j = i \cdot a_j \bmod p$, for $i \neq 1$, where $\cdot$ is multiplication modulo p.

Example 13. Let $p = 7$, $Q = \{1, 2, \ldots, 7\}$ and let first row of $(Q, \circ)$ is $(2, 4, 1, 5, 3, 6)$. $(Q, \circ)$ is given by following Table 2.14 and it has right identity 3.

$\circ$	1	2	3	4	5	6
1	2	4	1	5	3	6
2	$4(= 2 * 2 \text{ mod } 7)$	$1(= 2 * 4 \text{ mod } 7)$	2	3	6	5
3	$6(= 3 * 2 \text{ mod } 7)$	$5(= 3 * 4 \text{ mod } 7)$	3	1	2	4
4	$1(= 4 * 2 \text{ mod } 7)$	$2(= 4 * 4 \text{ mod } 7)$	4	6	5	3
5	$3(= 5 * 2 \text{ mod } 7)$	$6(= 5 * 4 \text{ mod } 7)$	5	4	1	2
6	$5(= 6 * 2 \text{ mod } 7)$	$3(= 6 * 4 \text{ mod } 7)$	6	2	4	1

Table 2.14: The integer representation of $(Q, \circ)$

Quasigroups generated in this way, have some structure, which can be presented with following proposition.

Proposition 13 *Let $(a_1, \ldots, a_n)$ be a permutation in the set $Q = \mathbb{Z}_p^*$ and let $\circ$ is quasigroup operation defined by* $i \circ j = \begin{cases} a_j, \ i = 1 \\ i \cdot a_j \ mod \ p, \ i \neq 1 \end{cases}$. *Then $(Q, \circ)$ has right unit.* □

PROOF Because $(a_1, \ldots, a_n)$ is permutation, $a_k = 1$, for some k. Because of the way how quasigroup is defined, k-th column in the multiplication table of $(Q, \circ)$ will be the same as the main column, so, k is the right unit. ■

2.10 Extended Feistel networks as orthomorphisms

Generally, a group with affine complete mapping or orthomorphism does not produce quasigroup that satisfies the needs of the cryptography. Non-affine orthomorphisms and complete mappings are more promising. It is very easy to create a table-driven non-affine orthomorphism or complete mappings as long as we don't care about the order of the quasigroup. Considering huge quasigroups, it is not practically possible to store table-driven bijections. It is much more difficult to create a non-affine bijection that is not table-driven and, additionally, that is an orthomorphism or complete mapping. One way of constructing orthomorphisms of the group $(\mathbb{Z}_2^n, \oplus_n)$ is given by Mittenthal [102]. In the paper [87], by using extended Feistel network, we create a huge non-affine complete mappings in the Kirsten sense, from a small table-driven non-affine bijections, but here we will create orthomorphisms in the sense of Definition 26. In the group $(\mathbb{Z}_2^n, \oplus_n)$ they are exactly the same.

Definition 27 Let $(G, +)$ be an Abelian group, let $f : G \to G$ be a mapping and let $a, b, c \in G$ are constants. The **extended Feistel network** $F_{a,b,c} : G^2 \to G^2$ created by f is defined for every $l, r \in G$ by

$$F_{a,b,c}(l, r) = (r + a, l + b + f(r + c)).$$

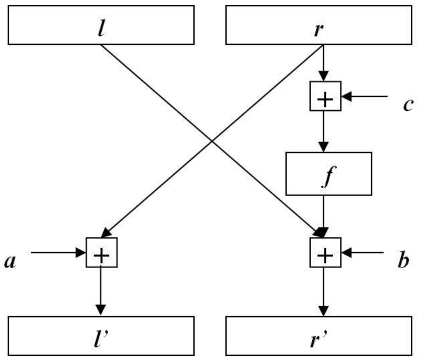

Figure 2: Extended Feistel network $F_{a,b,c}$

The extended Feistel network $F_{a,b,c}$ is a bijection with inverse

$$F^{-1}_{a,b,c}(l,r) = (r - b - f(l + c - a), l - a).$$

A Feistel network can be obtained from an extended Feistel network if we take constants $a = b = c = 0$.

One of the main results of the paper, that we will frequently use, is the following one.

Theorem 16 *Let $(G,+)$ be an Abelian group and $a,b,c \in G$. If $F_{a,b,c} : G^2 \to G^2$ is an extended Feistel network created by a bijection $f : G \to G$, then $F_{a,b,c}$ is an orthomorphism of the group $(G^2,+)$.* □

Proof Let $\Phi = F_{a,b,c} - I$, i.e.,

$$\Phi(l,r) = F(l,r) - (l,r) = (r - l + a, l - r + b + f(r + c))$$

for every $l, r \in G$. Define the function $\Omega : G^2 \to G^2$ by

$$\Omega(l,r) = (f^{-1}(l + r - a - b) - l + a - c, f^{-1}(l + r - a - b) - c).$$

We have $\Omega \circ \Phi = \Phi \circ \Omega = I$, i.e., Φ and $\Omega = \Phi^{-1}$ are bijections. ■

In the sequel we will consider only extended Feistel networks of the Abelian groups $(\mathbb{Z}_2^n, \oplus_n)$. One can notice that for those groups, every orthomorphism is complete mapping and vice versa.

Proposition 14 *Let $a,b,c \in \mathbb{Z}_2^k$ and let $F_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ be an extended Feistel network of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$ created by a mapping $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$. Then $F_{a,b,c}$ is affine iff f is affine.* □

Proof Let $l_1, l_2, r_1, r_2 \in \mathbb{Z}_2^k$ and let f be affine. Then, since $f(r_1 \oplus_k r_2 \oplus_k c) = f(r_1 \oplus_k c) \oplus_k f(r_2 \oplus_k c) \oplus_k f(c)$, we have that $F_{a,b,c}$ is affine as well:

$F_{a,b,c}((l_1,r_1) \oplus_{2k} (l_2,r_2))$

$= ((r_1 \oplus_k r_2 \oplus_k a), (l_1 \oplus_k l_2 \oplus_k b \oplus_k f(r_1 \oplus_k r_2 \oplus_k c)))$

$= [(r_1 \oplus_k a), (l_1 \oplus_k b \oplus_k f(r_1 \oplus_k c))] \oplus_{2k} [(r_2 \oplus_k a), (l_2 \oplus_k b \oplus_k f(r_2 \oplus_k c))] \oplus_{2k} [(0 \oplus_k a), (0 \oplus_k b \oplus_k f(0 \oplus_k c))]$

$= F_{a,b,c}(l_1,r_1) \oplus_{2k} F_{a,b,c}(l_2,r_2) \oplus_{2k} F_{a,b,c}(0,0)$,

Let now $F_{a,b,c}$ be an affine function. Then we have

$F_{a,b,c}((l_1,r_1) \oplus_{2k} (l_2,r_2)) = F_{a,b,c}(l_1,r_1) \oplus_{2k} F_{a,b,c}(l_2,r_2) \oplus_{2k} F_{a,b,c}(0,0)$

and that implies

$f(r_1 \oplus_k r_2 \oplus_k c) = f(r_1) \oplus_k f(r_2) \oplus_k f(c)$

for each $r_1, r_2 \in \mathbb{Z}_2^k$. We infer from the last equality that f is affine too:
$f(r_1 \oplus_k r_2) = f(r_1 \oplus_k (r_2 \oplus_k c) \oplus_k c) = f(r_1) \oplus_k f(r_2 \oplus_k c) \oplus_k f(c) = f(r_1) \oplus_k f(0 \oplus_k r_2 \oplus_k c) \oplus_k f(c) = f(r_1) \oplus_k f(0) \oplus_k f(r_2) \oplus_k f(c) \oplus_k f(c) = f(r_1) \oplus_k f(r_2) \oplus_k f(0)$. ■

So, if as orthomorphism a non-affine extended Feistel network $F_{a,b,c}$ created by f is needed, it is enough to take f to be a non-affine bijection.

Proposition 15 *Let $f, g : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ be bijections, $a, b, c, a', b', c' \in \mathbb{Z}_2^k$ and let $F_{a,b,c}, F_{a',b',c'} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ be extended Feistel networks of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by f and g respectfully. Then the composite function $F_{a,b,c} \circ F_{a',b',c'}$ is a complete mapping and orthomorphism on $\mathbb{Z}_2^{2k}$ too.* □

Proof Let $\Phi = I \oplus_{2k} F_{a,b,c} \circ F_{a',b',c'}$. Then, for every $l, r \in \mathbb{Z}_2^k$, we have

$$\Phi(l, r) = ((g(r \oplus_k c') \oplus_k a \oplus_k b'), (a' \oplus_k b \oplus_k f(l \oplus_k b' \oplus_k g(r \oplus_k c') \oplus_k c))).$$

Define the function $\Omega : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ by

$$\Omega(l, r) = ((f^{-1}(r \oplus_k a' \oplus_k b) \oplus_k l \oplus_k a \oplus_k c), (g^{-1}(l \oplus_k a \oplus_k b') \oplus_k c')).$$

It can be checked that $\Omega \circ \Phi = \Phi \circ \Omega = I$, i.e., Φ and $\Omega = \Phi^{-1}$ are bijections. ■

Corollary 3 *If $F_{a,b,c}$ is an extended Feistel network of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$ created by bijection f, then $F_{a,b,c}^2$ is a complete mapping and orthomorphism too.* □

In general, if θ is a orthomorphism on a group G, θ^2 may not be an orthomorphism on G, as Example 2 shows.

Example 14. We have in Table 2.15 an orthomorphism $\theta(x)$ on $(\mathbb{Z}_2^4, \oplus_4)$ (given in integer representation) such that $\theta^2(x)$ is not an orthomorphism, as it is shown in Table 2.16.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\theta(x)$	12	6	3	14	2	13	5	9	8	11	15	1	7	4	10	0
$x \oplus_4 \theta(x)$	12	7	1	13	6	8	3	14	0	2	5	10	11	9	4	15

Table 2.15: Integer representation of an orthomorphism $\theta(x)$

Example 15. In Table 2.17 we have an example of an extended Feistel network $F = F_{0,0,0}$ that is an orthomorphism created by a bijection f such

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\theta^2(x)$	7	5	14	10	3	4	13	11	8	1	0	6	9	2	15	12
$x \oplus_4 \theta^2(x)$	7	4	12	9	7	1	11	12	0	8	10	13	5	15	1	3

Table 2.16: Integer representation of a non-orthomorphism $\theta^2(x)$

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(x)$	3	2	1	0												
$F(x)$	3	6	9	12	2	7	8	13	1	4	11	14	0	5	10	15
$x \oplus_4 F(x)$	3	7	11	15	6	2	14	10	9	13	1	5	12	8	4	0

Table 2.17: Integer representation of an extended Feistel network $F(x)$

as F^3 is not an orthomorphism. Namely, F^3 is the identical mapping, so $I \oplus_4 F^3 = I \oplus_4 I$ is the constant zero mapping, that maps each $x \in \mathbb{Z}_2^4$ into 0.

Theorem 17 *Let $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ be a bijection of algebraic degree $deg(f) \geqslant 1$ and let $F_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ be an extended Feistel network of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by f. Then $deg(F_{a,b,c}) = deg(f)$.* □

Proof Let $(a_1, \dots, a_k)$, $(b_1, \dots, b_k)$ and $(c_1, \dots, c_k)$ be the binary representations of the constants $a, b, c \in \mathbb{Z}_2^k$. The mappings $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ and $F_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ are v.v.b.f. and so there are Boolean polynomials $q_1, q_2, \dots, q_k$ and $p_1, p_2, \dots, p_{2k}$ such that

$$f(x_1, \dots, x_k) = (q_1(x_1, \dots, x_k), q_2(x_1, \dots, x_k), \dots, q_k(x_1, \dots, x_k)),$$

$$F_{a,b,c}(x_1, \dots, x_{2k}) = (p_1(x_1, \dots, x_{2k}), p_2(x_1, \dots, x_{2k}), \dots, p_{2k}(x_1, \dots, x_{2k})).$$

Let $deg(f) = \mathtt{max}\{deg(q_i) \mid i \in \{1, 2, \dots, k\}\} \geqslant 1$. Then there is a $t \in \{1, 2, \dots, k\}$ such that $deg(f) = deg(q_t)$.

We have $F_{a,b,c}(x_1, \dots, x_{2k}) = (x_{k+1} \oplus a_1, \dots, x_{2k} \oplus a_k, x_1 \oplus b_1 \oplus q_1(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k), \dots, x_k \oplus b_k \oplus q_k(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k))$. This implies that $p_i(x_1, \dots, x_{2k}) = x_{i+k} \oplus a_i$ and $p_{i+k}(x_1, \dots, x_{2k}) = x_i \oplus b_i \oplus q_i(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k)$ for each $i \in \{1, 2, \dots, k\}$. Then, for each $i \in \{1, 2, \dots, k\}$, $deg(p_i) = 1$ and

$$deg(p_{i+k}) = \begin{cases} 0, & \text{when } q_i(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k) = x_i \oplus b_i \text{ for each } i \\ deg(q_i), & \text{otherwise.} \end{cases} \tag{2.15}$$

So, $deg(F_{a,b,c}) = deg(f)$. ■

Example 16. A bijection $f : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$ of $deg(f) = 3$ is given in Table 2.18. The representation of f as v.v.b.f. is $f(x_1, x_2, x_3, x_4) = (q_1, q_2, q_3, q_4)$, where

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$f(x)$	1	12	15	6	4	9	3	2	10	8	13	11	14	5	7	0

Table 2.18: A bijection f of $deg(f) = 3$

$q_1(x_1, x_2, x_3, x_4) = x_1 + x_3 + x_4 + x_1x_3 + x_1x_4 + x_2x_3 + x_1x_2x_4 + x_2x_3x_4$,
$q_2(x_1, x_2, x_3, x_4) = x_2 + x_3 + x_4 + x_1x_4 + x_3x_4 + x_1x_2x_3$,
$q_3(x_1, x_2, x_3, x_4) = x_1 + x_3 + x_1x_4 + x_1x_2x_3$,
$q_4(x_1, x_2, x_3, x_4) = 1 + x_1 + x_2 + x_4 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_1x_2x_3 + x_1x_2x_4$.

The Theorem 17 implies that we can make non-affine orthomorphisms $F_{a,b,c}$ of different non-linearity. Namely, it is enough to choose a non-affine bijection f of desired degree. An effective construction of bijection f of predefined higher degree is an open problem. Note that the maximum degree of a mapping $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ is less or equal than k.

The orthomorphism $F_{a,b,c}$ has the property that the first k polynomials are of degree 1. On the other side, the orthomorphism $F^2_{a,b,c}$ is with better performances, since $F^2_{a,b,c}(x_1, \dots, x_{2k}) = (A, B)$, where
$A = (x_1 \oplus b_1 \oplus q_1(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k), \dots, x_k \oplus b_k \oplus q_k(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k))$,
$B = (x_{k+1} \oplus a_1 \oplus q_1(x_1 \oplus b_1 \oplus q_1(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k), \dots, x_k \oplus b_k \oplus q_k(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k)), \dots, x_{2k} \oplus a_k \oplus q_k(x_1 \oplus b_1 \oplus q_1(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k), \dots, x_k \oplus b_k \oplus q_k(x_{k+1} \oplus c_1, \dots, x_{2k} \oplus c_k))$.

Theorem 18 *Let $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ be bijection, and let $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ be an extended Feistel network of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by f. For $0, x \in \mathbb{Z}_2^k$ we have $R_p(a' \dashv F_{A,B,C} \vdash b') = 1$ if and only if $a' = (x, 0)$ and $b' = (0, x)$.* □

Proof Let $b' = (b'_1, b'_2)$, $a' = (a'_1, a'_2)$, where $b'_1, b'_2, a'_1, a'_2 \in \mathbb{Z}_2^k$.
$R_p(a' \dashv F_{A,B,C} \vdash b') = 1 \Leftrightarrow$
$2^{-2k} \sum_a \delta(b' \oplus_{2k} F_{A,B,C}(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}(a)) = 1 \Leftrightarrow$
$\sum_a \delta(b' \oplus_{2k} F_{A,B,C}(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}(a)) = 2^{2k} \Leftrightarrow$
$\delta(b' \oplus_{2k} F_{A,B,C}(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}(a)) = 1 \ (\forall a \in \mathbb{Z}_2^{2k}) \Leftrightarrow$
$b' \oplus_{2k} F_{A,B,C}(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}(a) = 0 \ (\forall a \in \mathbb{Z}_2^{2k}) \Leftrightarrow$
$(b'_1, b'_2) \oplus_{2k} F_{A,B,C}((a_1, a_2) \oplus_{2k} (a'_1, a'_2)) \oplus_{2k} F_{A,B,C}(a_1, a_2) = 0 \ (\forall (a_1, a_2) \in \mathbb{Z}_2^{2k}) \Leftrightarrow$

$(b_1', b_2') \oplus_{2k} F_{A,B,C}(a_1 \oplus_k a_1', a_2 \oplus_k a_2') \oplus_{2k} F_{A,B,C}(a_1, a_2) = 0\ (\forall (a_1, a_2) \in \mathbb{Z}_2^{2k}) \Leftrightarrow b_1' \oplus_k a_2 \oplus_k a_2' \oplus_k A \oplus_k a_2 \oplus_k A = 0 \wedge b_2' \oplus_k a_1 \oplus_k a_1' \oplus_k B \oplus_k f(a_2' \oplus_k a_2 \oplus_k C) \oplus_k a_1 \oplus_k B \oplus_k f(a_2 \oplus_k C) = 0\ (\forall a_1, a_2 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b_1' = a_2' \wedge b_2' \oplus_k a_1' = f(a_2' \oplus_k a_2 \oplus_k C) \oplus_k f(a_2 \oplus_k C)\ (\forall a_2 \in \mathbb{Z}_2^k) \Leftrightarrow$ (f is bijection) $b_1' = a_2' = 0$ and $b_2' = a_1'$. ■

Corollary 4 *The prop ratio table of an extended Feistel network* $F_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ *of the group* $(\mathbb{Z}_2^{2k}, \oplus_{2k})$*, created by the bijection* $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ *on the group has exactly* 2^{k+1} *ones.* □

From the definition of the extended Feistel networks we have that at least first k component $2k$-ary Boolean functions are linear functions, so their correlation matrices have at least k values 1 or -1.

Theorem 19 *Let* $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ *be bijection, and let* $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ *be an extended Feistel network of the group* $(\mathbb{Z}_2^{2k}, \oplus_{2k})$*, created by* f*.* $R_p(a' \dashv F_{A,B,C}^2 \vdash b') = 1$ *if and only if* $a' = b' = (0, 0)$*.* □

Proof Let $b' = (b_1', b_2')$, $a' = (a_1', a_2')$, where $b_1', b_2', a_1', a_2' \in \mathbb{Z}_2^k$.
$R_p(a' \dashv F_{A,B,C}^2 \vdash b') = 1 \Leftrightarrow$
$2^{-2k} \sum_a \delta(b' \oplus_{2k} F_{A,B,C}^2(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}^2(a)) = 1 \Leftrightarrow$
$\sum_a \delta(b' \oplus_{2k} F_{A,B,C}^2(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}^2(a)) = 2^{2k} \Leftrightarrow$
$\delta(b' \oplus_{2k} F_{A,B,C}^2(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}^2(a)) = 1\ (\forall a \in \mathbb{Z}_2^{2k}) \Leftrightarrow$
$b' \oplus_{2k} F_{A,B,C}^2(a \oplus_{2k} a') \oplus_{2k} F_{A,B,C}^2(a) = 0\ (\forall a \in \mathbb{Z}_2^{2k}) \Leftrightarrow$
$(b_1', b_2') \oplus_{2k} F_{A,B,C}^2((a_1, a_2) \oplus_{2k} (a_1', a_2')) \oplus_{2k} F_{A,B,C}^2(a_1, a_2) = 0\ (\forall (a_1, a_2) \in \mathbb{Z}_2^{2k}) \Leftrightarrow (b_1', b_2') \oplus_{2k} F_{A,B,C}^2(a_1 \oplus_k a_1', a_2 \oplus_k a_2') \oplus_{2k} F_{A,B,C}^2(a_1, a_2) = 0\ (\forall (a_1, a_2) \in \mathbb{Z}_2^{2k}) \Leftrightarrow$
$b_1' \oplus_k a_1 \oplus_k a_1' \oplus_k A \oplus_k B \oplus_k f(a_2' \oplus_k a_2 \oplus_k C) \oplus_k a_1 \oplus_k A \oplus_k B \oplus_k f(a_2 \oplus_k C) = 0 \wedge b_2' \oplus_k a_2 \oplus_k a_2' \oplus_k A \oplus_k B \oplus_k f(a_1 \oplus_k a_1' \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a_2' \oplus_k C)) \oplus_k a_2 \oplus_k A \oplus_k B \oplus_k f(a_1 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k C)) = 0\ (\forall a_1, a_2 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b_1' \oplus_k a_1' = f(a_2' \oplus_k a_2 \oplus_k C) \oplus_k f(a_2 \oplus_k C) \wedge$
$b_2' \oplus_k a_2' = f(a_1 \oplus_k a_1' \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a_2' \oplus_k C)) \oplus_k f(a_1 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k C))\ (\forall a_1, a_2 \in \mathbb{Z}_2^k) \Leftrightarrow$
From first equality, because f is bijection, we have $a_2' = 0$ and $b_1' = a_1'$.
For second equality we have
$b_2' = f(a_1 \oplus_k a_1' \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k C)) \oplus_k f(a_1 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k C))\ (\forall a_1, a_2 \in \mathbb{Z}_2^k)$ and again because f is bijection, we have $a_1' = 0$ and $b_2' = 0$. ■

2.10.1 Orthogonal extended Feistel networks

The following propositions shows that the given extended Feistel network $F_{a,b,c}$ has at least two orthogonal orthomorphisms, its inverse $F^{-1}_{a,b,c}$ and $F^2_{a,b,c}$. In general, $F^{-1}_{a,b,c}$ and $F^2_{a,b,c}$ are not orthogonal.

Proposition 16 *Let $F_{a,b,c} : G^2 \to G^2$ be an extended Feistel network of Abelian group $(G^2, +)$ created by a bijection $f : G \to G$. $F_{a,b,c}$ and $F^{-1}_{a,b,c}$ are orthogonal orthomorphisms.* □

Proof Let conditions of the theorem be fulfilled. Let $\Phi = F_{a,b,c} - F^{-1}_{a,b,c}$. Then, for every $l, r \in G$, we have

$$\Phi(l, r) = (a + b + f(l + c - a), a + b + f(r + c)).$$

Define the function $\Omega : G^2 \to G^2$ by

$$\Omega(l, r) = (f^{-1}(l - a - b) - c + a, f^{-1}(r - a - b) - c).$$

It can be checked that $\Omega \circ \Phi = \Phi \circ \Omega = I$, i.e., Φ and $\Omega = \Phi^{-1}$ are bijections.■

Proposition 17 *Let $F_{a,b,c} : G^2 \to G^2$ be an extended Feistel network of Abelian group $(G^2, +)$ created by a bijection $f : G \to G$. $F_{a,b,c}$ and $F^2_{a,b,c}$ are orthogonal orthomorphisms.* □

Proof Let conditions of the theorem be fulfilled. Let $\Phi = F^2_{a,b,c} - F_{a,b,c}$. Then, for every $l, r \in G$, we have

$$\Phi(l, r) = (l - r + b + f(r + c), r - l + a + f(l + b + c + f(r + c)) - f(r + c)).$$

Define the function $\Omega : G^2 \to G^2$ by

$$\Omega(l, r) = (-f(f^{-1}(l+r-a-b)-l)+f^{-1}(l+r-a-b)-b-c, f^{-1}(l+r-a-b)-l-c).$$

It can be checked that $\Omega \circ \Phi = \Phi \circ \Omega = I$, i.e., Φ and $\Omega = \Phi^{-1}$ are bijections.■

Example 17. Let the group is $(\mathbb{Z}_2^2, \oplus_2)$. This is an example of extended Feistel network $F = F_{1,2,3} : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$, created by the bijection $f : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$ with two orthogonal mates, which are not orthogonal between themselves.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$F(x)$	5	0	14	11	4	1	15	10	7	2	12	9	6	3	13	8
$F^{-1}(x)$	1	5	9	13	4	0	12	8	15	11	7	3	10	14	2	6
$F(x) \oplus_4 F^{-1}(x)$	4	5	7	6	0	1	3	2	8	9	11	10	12	13	15	14
$F^2(x)$	1	5	13	9	4	0	8	12	10	14	6	2	15	11	3	7
$F(x) \oplus_4 F^2(x)$	4	5	3	2	0	1	7	6	13	12	10	11	9	8	14	15
$F^{-1}(x) \oplus_4 F^2(x)$	0	0	4	4	0	0	4	4	5	5	1	1	5	5	1	1

Table 2.19: $F = F_{1,2,3} : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$, $F^{-1}(x)$ and $F^2(x)$

2.10.2 Huge quasigroups generated by a chain of extended Feistel networks

Recall that an extended Feistel network $F_{a,b,c}$ $(a, b, c \in {\mathbb{Z}_2}^s)$ created by a bijection $f : {\mathbb{Z}_2}^s \to {\mathbb{Z}_2}^s$ is an orthomorphism, so $F_{a,b,c}$ is a bijection on ${\mathbb{Z}_2}^{2s}$ as well. Define $F^{(1)}_{a^{(1)},b^{(1)},c^{(1)}} = F_{a,b,c}$ and let $F^{(n)}{}_{a^{(n)},b^{(n)},c^{(n)}}$, $n \geqslant 1$, be defined. Then, for some $a^{(n+1)}, b^{(n+1)}, c^{(n+1)} \in {\mathbb{Z}_2}^{s2^{n+1}}$, define $F^{(n+1)}_{a^{(n+1)},b^{(n+1)},c^{(n+1)}}$ to be the extended Feistel network created by the bijection $F^{(n)}_{a^{(n)},b^{(n)},c^{(n)}}$. Note that $F^{(n)}_{a^{(n)},b^{(n)},c^{(n)}}$ is an orthomorphism of the group ${\mathbb{Z}_2}^{s2^n}$ for each $n \geqslant 1$, hence we have defined inductively a chain of orthomorphisms $\{F^{(n)}_{a^{(n)},b^{(n)},c^{(n)}} | \, n = 1, 2, 3, \ldots\}$ in the corresponding groups. Now, by using (1), one can define a quasigroup of order 2^{s2^n} on the set ${\mathbb{Z}_2}^{s2^n}$ for each $n \geqslant 1$.

In applications one needs effectively constructed quasigroups of order 2^{256}, 2^{512}, $2^{1024}, \ldots$. A huge quasigroup of order 2^{2^k} can now be designed as it follows. Take a suitable non-affine bijection of desired algebraic degree $f : {\mathbb{Z}_2}^{2^t} \to {\mathbb{Z}_2}^{2^t}$, where $t < k$ is a small positive integer $(t = 2, 3, 4)$. Choose suitable constants $a^{(i)}, b^{(i)}, c^{(i)} \in {\mathbb{Z}_2}^{2^{t+i}}$, $1 \leqslant i \leqslant k-t$, and construct iteratively the orthomorphisms $F = F^{(k-t)}_{a^{(k-t)},b^{(k-t)},c^{(k-t)}} : {\mathbb{Z}_2}^{2^k} \to {\mathbb{Z}_2}^{2^k}$. Define a quasigroup operation $\circ$ on the set ${\mathbb{Z}_2}^{2^k}$ by (1), i.e.,

$$x \circ y = F(x \oplus y) \oplus y, \text{ for every } x, y \in {\mathbb{Z}_2}^{2^k}.$$

Note that we need only $k - t$ iterations for getting F and a small amount of memory for storing the bijection f. Hence, the complexity of our algorithm for construction of quasigroups of order 2^{2^k} is $\mathcal{O}(\texttt{log}(\texttt{log}k))$.

Example 17. As starting bijection we can use the bijection $f : \mathbb{Z}_2^4 \to \mathbb{Z}_2^4$ from Example 16. So, $t = 2$. We choose constants $(a^{(i)}, b^{(i)}, c^{(i)}) = (i, 0, 0) \in$

${\mathbb{Z}_2}^{2^{t+i}}$, $i = 1, 2, \ldots, 7$. Now we can construct the following orthomorphisms, where $l_i, r_i \in \mathbb{Z}_2^i$, $i = 4, 8, 16, \ldots$:
$F^{(1)}_{1,0,0} : \mathbb{Z}_2^8 \to \mathbb{Z}_2^8$ as $F^{(1)}_{1,0,0}(l_4, r_4) = ((r_4 \oplus_4 1), (l_4 \oplus_4 f(r_4)))$,
$F^{(2)}_{2,0,0} : \mathbb{Z}_2^{16} \to \mathbb{Z}_2^{16}$ as $F^{(2)}_{2,0,0}(l_8, r_8) = ((r_8 \oplus_8 2), (l_8 \oplus_8 F^{(1)}_{1,0,0}(r_8)))$,
$F^{(3)}_{3,0,0} : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ as $F^{(3)}_{3,0,0}(l_{16}, r_{16}) = ((r_{16} \oplus_{16} 3), (l_{16} \oplus_{16} F^{(2)}_{2,0,0}(r_{16})))$,
$F^{(4)}_{4,0,0} : \mathbb{Z}_2^{64} \to \mathbb{Z}_2^{64}$ as $F^{(4)}_{4,0,0}(l_{32}, r_{32}) = ((r_{32} \oplus_{32} 4), (l_{32} \oplus_{32} F^{(3)}_{3,0,0}(r_{32})))$,
$F^{(5)}_{5,0,0} : \mathbb{Z}_2^{128} \to \mathbb{Z}_2^{128}$ as $F^{(5)}_{5,0,0}(l_{64}, r_{64}) = ((r_{64} \oplus_{64} 5), (l_{64} \oplus_{64} F^{(4)}_{4,0,0}(r_{64})))$,
$F^{(6)}_{6,0,0} : \mathbb{Z}_2^{256} \to \mathbb{Z}_2^{256}$ as $F^{(6)}_{6,0,0}(l_{128}, r_{128}) = ((r_{128} \oplus_{128} 6), (l_{128} \oplus_{128} F^{(5)}_{5,0,0}(r_{128})))$,
$F^{(7)}_{7,0,0} : \mathbb{Z}_2^{512} \to \mathbb{Z}_2^{512}$ as $F^{(7)}_{7,0,0}(l_{256}, r_{256}) = ((r_{256} \oplus_{256} 7), (l_{256} \oplus_{256} F^{(6)}_{6,0,0}(r_{256})))$.
So we need $7 = 9 - 2$ iterations for getting $F^{(7)}_{7,0,0} : \mathbb{Z}_2^{512} \to \mathbb{Z}_2^{512}$.

Further on in this section we consider the algebraic properties of the quasigroups obtained by the above mentioned algorithm. For that aim we take a somewhat simplified situation when $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ is a bijection and $F_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ is an extended Feistel network created by f. We denote by $(Q, \circ)$ the quasigroup on the set $Q = \mathbb{Z}_2^{2k}$ derived by the orthomorphism $F_{a,b,c}$.

Proposition 18 *The quasigroup $(Q, \circ)$ is non-idempotent iff $f(c) \neq b$ or $a \neq 0$.* □

Proof Let $(Q, \circ)$ be idempotent. Then for all $x \in Q$ we have

$$x \circ x = x \iff F_{a,b,c}(x \oplus_{2k} x) \oplus_{2k} x = x \iff F_{a,b,c}(0,0) = (0,0) \iff$$
$$F_{a,b,c}(a, b \oplus_k f(c)) = (0,0) \iff a = 0 \wedge f(c) = b.$$

Proposition 19 *The quasigroup $(Q, \circ)$ has neither left nor right unit.* □

Proof Let e be the right unit of $(Q, \circ)$. Then, for all $x \in Q$, we have

$$x \circ e = x \Longrightarrow F_{a,b,c}(x \oplus_{2k} e) \oplus_{2k} e = x \Longrightarrow F_{a,b,c}(x \oplus_{2k} e) = x \oplus_{2k} e.$$

This means that $F_{a,b,c} = I$ is the identity mapping. We have now, for every $l, r \in Q$, that $(r \oplus_k a, l \oplus_k b \oplus_k f(r \oplus_k c)) = (l, r)$ and this implies that $f(r \oplus_k c) = a \oplus_k b$ for each r. The last equality contradicts the bijectivity of f.

Let e be the left unit of $(Q, \circ)$. Then, for all $x \in Q$, we have

$$e \circ x = x \Longrightarrow F_{a,b,c}(e \oplus_{2k} x) \oplus_{2k} x = x \Longrightarrow F_{a,b,c}(e \oplus_{2k} x) = 0.$$

This contradicts the fact that $F_{a,b,c}$ is a bijection. ■

Proposition 20 *The equality*

$$(x \circ y) \circ (y \circ x) = x \tag{2.16}$$

is an identity in $(Q, \circ)$, i.e. $(Q, \circ)$ is a Schroeder quasigroup. □

Proof $(x \circ y) \circ (y \circ x) = F_{a,b,c}((x \circ y) \oplus_n (y \circ x)) \oplus_n (y \circ x)) =$
$F_{a,b,c}(F_{a,b,c}(x \oplus_n y) \oplus_n y \oplus_n F_{a,b,c}(y \oplus_n x) \oplus_n x) \oplus_n F_{a,b,c}(y \oplus_n x) \oplus_n x = x$

Corollary 5 *The quasigroup $(Q, \circ)$ is non-commutative and, much more, no different elements of Q commutes.* □

Proof Let $x, y \in Q$ and let $x \circ y = y \circ x$.
By (2.16), we have $x = (x \circ y) \circ (y \circ x) = (y \circ x) \circ (x \circ y) = y$. ■

Lemma 2 *Let $\phi = I \oplus_{2k} F_{a,b,c}$. Then $\phi \circ F_{a,b,c} = F_{a,b,c} \circ \phi$ iff $a = 0$ and $f(r \oplus_k c) \oplus_k f(l \oplus_k b \oplus_k c \oplus_k f(r \oplus_k c)) = b \oplus_k f(l \oplus_k r \oplus_k b \oplus_k c \oplus_k f(r \oplus_k c))$ for each $l, r \in Q$.* □

Proof Let $l, r \in Q$. Then
$\phi(l, r) = ((l \oplus_k r \oplus_k a), (l \oplus_k r \oplus_k b \oplus_k f(r \oplus_k c)))$,
$(\phi \circ F_{a,b,c})(l, r) = ((r \oplus_k l \oplus_k b \oplus_k f(r \oplus_k c)), (r \oplus_k a \oplus_k l \oplus_k f(r \oplus_k c) \oplus_k f(l \oplus_k b \oplus_k f(r \oplus_k c) \oplus_k c)))$,
$(F_{a,b,c} \circ \phi)(l, r) = ((l \oplus_k r \oplus_k b \oplus_k f(r \oplus_k c) \oplus_k a), (l \oplus_k r \oplus_k a \oplus_k b \oplus_k f(l \oplus_k r \oplus_k b \oplus_k f(r \oplus_k c) \oplus_k c)))$.
Hence, we have:
$(\phi \circ F_{a,b,c})(l, r) = (F_{a,b,c} \circ \phi)(l, r) \iff a = 0 \ \wedge \ f(r \oplus_k c) \oplus_k f(l \oplus_k b \oplus_k c \oplus_k f(r \oplus_k c)) = b \oplus_k f(l \oplus_k r \oplus_k b \oplus_k c \oplus_k f(r \oplus_k c))$. ■

Lemma 3 *For the quasigroup $(Q, \circ)$ we have*

$$x \circ (y \circ x) = (x \circ y) \circ x \iff (\phi \circ F_{a,b,c})(x \oplus_{2k} y) = (F_{a,b,c} \circ \phi)(x \oplus_{2k} y)$$

for any $x, y \in Q$, $x \neq y$, where $\phi = I \oplus_{2k} F_{a,b,c}$. □

Proof $x \circ (y \circ x) = (x \circ y) \circ x \iff$

$$\begin{aligned}
&F_{a,b,c}(x \oplus_{2k} F_{a,b,c}(y \oplus_{2k} x) \oplus_{2k} x) \oplus_{2k} F_{a,b,c}(y \oplus_{2k} x) \oplus_{2k} x = \\
&\qquad = F_{a,b,c}(F_{a,b,c}(x \oplus_{2k} y) \oplus_{2k} y \oplus_{2k} x) \oplus_{2k} x \iff \\
&F_{a,b,c}(F_{a,b,c}(y \oplus_{2k} x)) \oplus_{2k} F_{a,b,c}(y \oplus_{2k} x) = \\
&\qquad = F_{a,b,c}(F_{a,b,c}(x \oplus_{2k} y) \oplus_{2k} x \oplus_{2k} y) \iff \\
&\phi(F_{a,b,c}(x \oplus_{2k} y)) = F_{a,b,c}(\phi(x \oplus_{2k} y))
\end{aligned}$$

■

An immediate consequence of Lemma 2 and Lemma 3 is that

$$x \circ (x \circ x) = (x \circ x) \circ x \iff a = 0 \ \wedge \ f(c) = b.$$

Now we have the following sufficient conditions for non-associativity of the quasigroup $(Q, \circ)$.

Proposition 21 *If $a \neq 0$, or $f(c) \neq b$, or $\phi \circ F_{a,b,c}(x) \neq F_{a,b,c} \circ \phi(x)$ for some $x \neq 0 \in Q$, then the quasigroup $(Q, \circ)$ is non-associative.* □

It can be checked that the quasigroup $(Q, \circ)$ is associative iff the following equalities are identities in $(\mathbb{Z}_2^k, \oplus_k)$, where $t, x_l, x_r, y_l, y_r, z_l, z_r$ are variables:

$$\begin{array}{rcl} t & = & x_l \oplus_k x_r \oplus_k z_l \oplus_k z_r \oplus_k f(y_r \oplus_k z_r \oplus_k c), \\ t & = & a \oplus_k f(x_r \oplus_k y_r \oplus_k c), \\ t & = & b \oplus_k f(x_l \oplus_k y_l \oplus_k y_r \oplus_k z_r \oplus_k a \oplus_k b \oplus_k c \oplus_k t) \oplus_k \\ & & \quad \oplus_k f(x_l \oplus_k y_l \oplus_k b \oplus_k c \oplus_k t). \end{array} \tag{2.17}$$

Namely, we can represent $x, y, z \in Q$ by $x = (x_l, x_r)$, $y = (y_l, y_r)$, $z = (z_l, z_r)$, where $x_l, x_r, y_l, y_r, z_l, z_r \in \mathbb{Z}_2^k$, and then $(x \circ y) \circ z = x \circ (y \circ z)$ iff (2.17) holds true. This shows that the quasigroup $(Q, \circ)$ is highly non-associative, since a bijection f can hardly satisfies the equations (2.17) for the given elements $x, y, z \in Q$.

Note that if θ is an orthomorphism of a group $(\mathbb{Z}_2^n, \oplus_n)$, we have

$$y \circ x = \theta(y \oplus_n x) \oplus_n x$$

$$(y \circ x) \circ x = \theta(\theta(y \oplus_n x) \oplus_n x \oplus_n x) \oplus_n x = \theta^2(y \oplus_n x) \oplus_n x$$

and, by induction,

$$((y \circ \underbrace{x) \circ \dots) \circ x}_{l} = \theta^l(y \oplus_n x) \oplus_n x.$$

We have also

$$x \circ y = \theta(x \oplus_n y) \oplus_n y \oplus_n x \oplus_n x = \phi(x \oplus_n y) \oplus_n x,$$

$$x \circ (x \circ y) = \theta(x \oplus_n \phi(x \oplus_n y) \oplus_n x) \oplus_n \phi(x \oplus_n y) \oplus_n x = \phi^2(x \oplus_n y) \oplus_n x$$

and, by induction,

$$\underbrace{x \circ (\cdots \circ (x}_{l} \circ y)) = \phi^l(x \oplus_n y) \oplus_n x.$$

Proposition 22 *a) The identity*

$$y = ((y \circ \underbrace{x) \circ \dots) \circ x}_{l}$$

holds true in $(Q,\circ)$ *iff* $\theta^l = I$.

b) The identity

$$\underbrace{x \circ (\dots \circ (x}_{l} \circ y)) = y$$

holds true in $(Q,\circ)$ *iff* $\phi^l = I$*, where* $\phi = I \oplus_{2k} \theta$. □

Regarding the subquasigroups of the quasigroup $(Q,\circ)$, we notice the following property, where $< A >$ denotes the subquasigroup generated by the subset A of Q.

Proposition 23 $< 0 >=< \{\theta^i(0) |\ i = 1, 2, \dots\} >$. □

PROOF $0 \circ 0 = \theta(0),\ \theta(0) \circ 0 = \theta^2(0),\ \theta^2(0) \circ 0 = \theta^3(0), \dots$. ■

Theorem 20 *Let* $Q_{F_{A,B,C}} : \mathbb{Z}_2^{4k} \to \mathbb{Z}_2^{2k}$ *be a quasigroup generated by the extended Feistel network* $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ *of the group* $(\mathbb{Z}_2^{2k}, \oplus_{2k})$*, created by the bijection* $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$*. For* $x, y, z \in \mathbb{Z}_2^k$ *and we have* $R_p(a' \dashv Q_{F_{A,B,C}} \vdash b') = 1$ *if and only if* $a' = (x, y, z, y \oplus_k C)$ *and* $b' = (z \oplus_k C, x \oplus_k y \oplus_k z \oplus_k C)$. □

PROOF Let $b' = (b'_1, b'_2)$, $a' = (a'_1, a'_2, a'_3, a'_4)$, where $b'_1, b'_2, a'_1, a'_2, a'_3, a'_4 \in \mathbb{Z}_2^k$.
$R_p(a' \dashv Q_{F_{A,B,C}} \vdash b') = 1 \Leftrightarrow$
$2^{-4k} \sum_a \delta(b' \oplus_{2k} Q_{F_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F_{A,B,C}}(a)) = 1 \Leftrightarrow$
$\sum_a \delta(b' \oplus_{2k} Q_{F_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F_{A,B,C}}(a)) = 2^{4k} \Leftrightarrow$
$\delta(b' \oplus_{2k} Q_{F_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F_{A,B,C}}(a)) = 1\ (\forall a \in \mathbb{Z}_2^{4k}) \Leftrightarrow$
$b' \oplus_{2k} Q_{F_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F_{A,B,C}}(a) = 0\ (\forall a \in \mathbb{Z}_2^{4k}) \Leftrightarrow$
$(b'_1, b'_2) \oplus_{2k} Q_{F_{A,B,C}}(a_1 \oplus_k a'_1, a_2 \oplus_k a'_2, a_3 \oplus_k a'_3, a_4 \oplus_k a'_4) \oplus_{2k} Q_{F_{A,B,C}}(a_1, a_2, a_3, a_4) = 0\ (\forall a_1, a_2, a_3, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b'_1 \oplus_k a_2 \oplus_k a'_2 \oplus_k a_3 \oplus_k a'_3 \oplus_k a_4 \oplus_k a'_4 \oplus_k A \oplus_k a_2 \oplus_k a_3 \oplus_k a_4 \oplus_k A = 0 \wedge b'_2 \oplus_k a_1 \oplus_k a'_1 \oplus_k a_3 \oplus_k a'_3 \oplus_k a_4 \oplus_k a'_4 \oplus_k B \oplus_k f(a'_2 \oplus_k a_2 \oplus_k a'_4 \oplus_k a_4 \oplus_k C) \oplus_k a_1 \oplus_k a_3 \oplus_k a_4 \oplus_k B \oplus_k f(a_2 \oplus_k a_4 \oplus_k C) = 0\ (\forall a_1, a_2, a_3, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b'_1 = a'_2 \oplus_k a'_3 \oplus_k a'_4 \wedge$
$b'_2 \oplus_k a'_1 \oplus_k a'_3 \oplus_k a'_4 = f(a'_2 \oplus_k a_2 \oplus_k a'_4 \oplus_k a_4 \oplus_k C) \oplus_k f(a_2 \oplus_k a_4 \oplus_k C)\ (\forall a_2, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
From the second equation, because f is a bijection, we have that $a'_4 = a'_2 \oplus_k C$ ($b'_1 = a'_3 \oplus_k C$ for first equation) and $b'_2 \oplus_k a'_1 \oplus_k a'_3 \oplus_k a'_4 = 0$. The last equation can be written also as $b'_2 = a'_1 \oplus_k a'_2 \oplus_k a'_3 \oplus_k C$. This means that $a' = (x, y, z, y \oplus_k C)$ and $b' = (z \oplus_k C, x \oplus_k y \oplus_k z \oplus_k C)$ for some $x, y, z \in \mathbb{Z}_2^k$. ■

Corollary 6 *Extended Feistel network $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by the bijection $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ produces weak-restricted quasigroups and even more, its prop ratio table has 2^{3k} ones.* □

Remark 5 If we analyze quasigroup $Q_{F_{A,B,C}}(x_1, x_2, y_1, y_2) = (x_2 \oplus_k y_1 \oplus_k y_2 \oplus_k A, x_1 \oplus_k y_1 \oplus_k y_2 \oplus_k B \oplus_k f(x_2 \oplus_k y_2 \oplus_k C))$, where $x_1, x_2, y_1, y_2 \in \mathbb{Z}_2^k$, obtained by $F_{A,B,C}$, it is easy to see that first k component $4k$-ary Boolean functions are linear, so the following statement is true. *Extended Feistel network $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$ created by the bijection $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ produces weak non-linear and correlated quasigroups.* □

Theorem 21 *Let $Q_{F_{A,B,C}} : \mathbb{Z}_2^{4k} \to \mathbb{Z}_2^{2k}$ be a quasigroup generated by the extended Feistel network $F_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by the bijection $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$. For $x, y \in \mathbb{Z}_2^k$ we have $R_p(a' \dashv Q_{F^2_{A,B,C}} \vdash b') = 1$ if and only if $a' = (x, y, x, y \oplus_k C)$ and $b' = (x, y)$.* □

Proof Let $b' = (b'_1, b'_2)$, $a' = (a'_1, a'_2, a'_3, a'_4)$, where $b'_1, b'_2, a'_1, a'_2, a'_3, a'_4 \in \mathbb{Z}_2^k$.
$R_p(a' \dashv Q_{F^2_{A,B,C}} \vdash b') = 1 \Leftrightarrow$
$2^{-4k} \sum_a \delta(b' \oplus_{2k} Q_{F^2_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F^2_{A,B,C}}(a)) = 1 \Leftrightarrow$
$\sum_a \delta(b' \oplus_{2k} Q_{F^2_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F^2_{A,B,C}}(a)) = 2^{4k} \Leftrightarrow$
$\delta(b' \oplus_{2k} Q_{F^2_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F^2_{A,B,C}}(a)) = 1 \ (\forall a \in \mathbb{Z}_2^{4k}) \Leftrightarrow$
$b' \oplus_{2k} Q_{F^2_{A,B,C}}(a \oplus_{4k} a') \oplus_{2k} Q_{F^2_{A,B,C}}(a) = 0 \ (\forall a \in \mathbb{Z}_2^{4k}) \Leftrightarrow$
$(b'_1, b'_2) \oplus_{2k} Q_{F^2_{A,B,C}}(a_1 \oplus_k a'_1, a_2 \oplus_k a'_2, a_3 \oplus_k a'_3, a_4 \oplus_k a'_4) \oplus_{2k} Q_{F^2_{A,B,C}}(a_1, a_2, a_3, a_4) = 0 \ (\forall a_1, a_2, a_3, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b'_1 \oplus_k a_1 \oplus_k a'_1 \oplus_k A \oplus_k B \oplus_k f(a_2 \oplus_k a'_2 \oplus_k a_4 \oplus_k a'_4 \oplus_k C) \oplus_k a_1 \oplus_k A \oplus_k B \oplus_k f(a_2 \oplus_k a_4 \oplus_k C) = 0 \wedge b'_2 \oplus_k a_2 \oplus_k a'_2 \oplus_k A \oplus_k B \oplus_k f(a_1 \oplus_k a'_1 \oplus_k a_3 \oplus_k a'_3 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a'_2 \oplus_k a_4 \oplus_k a'_4 \oplus_k C)) \oplus_k a_2 \oplus_k A \oplus_k B \oplus_k f(a_1 \oplus_k a_3 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a_4 \oplus_k C)) = 0 \ (\forall a_1, a_2, a_3, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
$b'_1 \oplus_k a'_1 = f(a_2 \oplus_k a'_2 \oplus_k a_4 \oplus_k a'_4 \oplus_k C) \oplus_k f(a_2 \oplus_k a_4 \oplus_k C) \wedge b'_2 \oplus_k a'_2 = f(a_1 \oplus_k a'_1 \oplus_k a_3 \oplus_k a'_3 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a'_2 \oplus_k a_4 \oplus_k a'_4 \oplus_k C)) \oplus_k f(a_1 \oplus_k a_3 \oplus_k B \oplus_k C \oplus_k f(a_2 \oplus_k a_4 \oplus_k C)) \ (\forall a_1, a_2, a_3, a_4 \in \mathbb{Z}_2^k) \Leftrightarrow$
From the first equation, because f is a bijection, we have that $a'_4 = a'_2 \oplus_k C$ and $b'_1 = a'_1$ and from the second equation, because the same reason, $a'_3 = a'_1$ and $b'_2 = a'_2$. This means that $a' = (x, y, x, y \oplus_k C)$ and $b' = (x, y)$. ■

Corollary 7 *Extended Feistel network $F^2_{a,b,c} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by the bijection $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$ produces weak-restricted quasigroups and even more, its prop ratio table has 2^{2k} ones.* □

Remark 6 If we analyze the quasigroup $Q_{F^2_{A,B,C}}(x_1, x_2, y_1, y_2) = (x_1 \oplus_k A \oplus_k B \oplus_k f(x_2 \oplus_k y_2 \oplus_k C), x_2 \oplus_k A \oplus_k B \oplus_k f(x_1 \oplus_k y_1 \oplus_k B \oplus_k C \oplus_k f(x_2 \oplus_k y_2 \oplus_k C)))$, where $x_1, x_2, y_1, y_2 \in \mathbb{Z}_2^k$, obtained by $F^2_{A,B,C}$, it is easy to see that its linearity depends on linearity of f, so the following statement is true. *Extended Feistel network $F^2_{A,B,C} : \mathbb{Z}_2^{2k} \to \mathbb{Z}_2^{2k}$ of the group $(\mathbb{Z}_2^{2k}, \oplus_{2k})$, created by the bijection $f : \mathbb{Z}_2^k \to \mathbb{Z}_2^k$, where f as a vector valued Boolean function does not have any linear component Boolean function, produces pure non-linear and non-correlated quasigroups.* □

Proposition 24 *The quasigroup $(Q, \bullet)$, created by an affine complete mapping θ of a group $(\mathbb{Z}_2^n, \oplus_n)$ is totally anti-symmetric (TA-quasigroup).* □

Proof Let $\phi = I \oplus_n \theta$ is orthomorphism of affine complete mapping θ, so ϕ is affine bijection too.
(1) $x \bullet y = y \bullet x \Rightarrow x = y$ follows from Corollary 5.
(2) Let $x, y, c \in Q$ and let $(c \bullet x) \bullet y = (c \bullet y) \bullet x \Rightarrow$
$\theta(\theta(c \oplus_n x) \oplus_n x \oplus_n y) \oplus_n y = \theta(\theta(c \oplus_n y) \oplus_n y \oplus_n x) \oplus_n x \Rightarrow$
$\theta(\theta(c \oplus_n x)) \oplus_n \theta(x) \oplus_n \theta(y) \oplus_n y = \theta(\theta(c \oplus_n y)) \oplus_n \theta(y) \oplus_n \theta(x) \oplus_n x \Rightarrow$
$\theta(\theta(c \oplus_n x)) \oplus_n y = \theta(\theta(c \oplus_n y)) \oplus_n x \Rightarrow$
$\theta(\theta(c) \oplus_n \theta(x) \oplus_n \theta(0)) \oplus_n y = \theta(\theta(c) \oplus_n \theta(y) \oplus_n \theta(0)) \oplus_n x \Rightarrow$
$\theta(\theta(c)) \oplus_n \theta(\theta(x)) \oplus_n \theta(\theta(0)) \oplus_n y = \theta(\theta(c)) \oplus_n \theta(\theta(y)) \oplus_n \theta(\theta(0)) \oplus_n x \Rightarrow$
$\theta(\theta(x)) \oplus_n x = \theta(\theta(y)) \oplus_n y \Rightarrow$
$\theta(\theta(x)) \oplus_n \theta(x) \oplus_n \theta(x) \oplus_n x = \theta(\theta(y)) \oplus_n \theta(y) \oplus_n \theta(y) \oplus_n y \Rightarrow$
$\phi(\theta(x)) \oplus_n \phi(x) \oplus_n \phi(0) = \phi(\theta(y)) \oplus_n \phi(y) \oplus_n \phi(0) \Rightarrow$
$\phi(\theta(x) \oplus_n x) = \phi(\theta(y) \oplus_n y) \Rightarrow$ (ϕ is bijection)
$\theta(x) \oplus_n x = \theta(y) \oplus_n y \Rightarrow$
$\phi(x) = \phi(y) \Rightarrow$ (ϕ is bijection)
$x = y$
From (1) and (2) $\Rightarrow (Q, \bullet)$ is totally anti-symmetric quasigroup. ■

Affine extended Feistel network can find some application also, for example, for creating TA-quasigroups [16] that can be used for the definition of the check digit systems, where the early typing errors have to be recognized. Creating a quasigroup by using an affine complete map is simply a special case of creating a quasigroup by affine isotopies [97].

Remark 7 There are 384 complete mappings of the group $(\mathbb{Z}_2^3, \oplus_3)$ and they all are affine [99]. □

2.11 Summary

Our contributions in this chapter are:

- a survey of most common ways of constructing quasigroups
- new approach which connects the Feistel networks and the orthomorphisms as extended Feistel networks, for generating huge quasigroups
- examination of properties of quasigroups obtained by extended Feistel network.

As an open question remains the exploring of the extended Feistel networks from other groups than $(\mathbb{Z}_2^k, \oplus_k)$ and analyzing the produced quasigroups. Interesting will be the produce of extended Feistel network as orthomorphisms from dihedral group.

Chapter 3

Cryptographic primitives with quasigroup transformations

Most of the known constructions of cryptographic primitives, error detecting and error correcting codes use structures from the associative algebra as groups, rings and fields. Two eminent specialists on quasigroups, J. Dénes and A. D. Keedwell [22], once proclaimed the advent of a new era in cryptology, consisting in the application of non-associative algebraic systems as quasigroups and neo-fields. Quasigroups and their combinatorial equivalent Latin squares are very suitable for this aim, because of their structure, their features, their big number and because they lead to particular simple and yet efficient primitives. Nevertheless, at present, very few researchers use these tools and cryptographic community still hesitate about them.

First quasigroup-Latin square application in cryptography dated from 16 century. Johannes Trithemius (1462-1516) invented a progressive key polyalphabetic cipher called the Trithemius cipher, which switch alphabet for each letter in the message. This can be represented, for example for English alphabet, by 26 x 26 Latin square. Each next row is new alphabet shifted one letter to the left from the one above it. Another early application is in the Schaufler PhD dissertation [125] from 1948, where he reduced the problem of breaking the Vigènere cipher to minimum number of entries of a particular Latin square which would determine the square completely. Most of the results from application of quasigroups in cryptology to the end of eighties years of the 20 century are described in [19, 20]. Some newer results and topics are not covered in this thesis, like quasigroup based secret sharing schemas and zero knowledge protocols, generating the NLPN-sequences, application of critical sets and power sets of Latin square and row-Latin squares in cryptography. We refer [128] for those topics.

Application of quasigroups in cryptography is justified also by the concept of multipermutation, introduced by Schnorr and Vaudenay [127], which is pervasive in cryptography and correspond to pairs of orthogonal Latin squares. A permutation $f : Z^2 \to Z^2$, $f(a,b) = (f_1(a,b), f_2(a,b))$ is a multipermutation, if for every $a, b \in Z$ the mappings $f_1(a,*)$, $f_1(*,b)$, $f_2(a,*)$ and $f_2(*,b)$ are permutations on S. In the light of the latest linear and differential attacks to the cryptographic primitives, multipermutations are a basic cryptographic tool for a perfect generation of diffusion and confusion, because, intuitively, modifying one or several inputs of the multipermutation has the influence to modify a maximal number of outputs from the computation. Vaudenay [133] generalized the concept of multipermutation by following definition.

Definition 28 A $(r,n)-$ multipermutation over an alphabet Z is a function $f : Z^r \to Z^n$ such that two different $(r+n)$-tuples of the form $(x, f(x))$ cannot collide in any r positions. □

A $(2,1)$ multipermutation is equivalent to a Latin square. A $(2,n)$ multipermutation is equivalent to a set of n two wise orthogonal Latin squares.

In this chapter is given a survey of basic cryptographic primitives, like hash functions, block and stream ciphers, pseudo-random number generators and public key algorithms, build specifically with quasigroups and quasigroup transformations. In the earlier designs, security was based on secret quasigroup operations, big number of quasigroups of the same order, big number of isotopies for a given carrier, secret permutation J in $CI-$quasigroups, etc. The newer designs base their security mostly on difficulty to solve systems of quasigroup equations, but also you can find security based on secret order of elements in quasigroup operation, secret leaders and/or order of used elementary quasigroup transformations, secret order of used quasigroups from some predefined set of quasigroups, solving a system of multivariate quadratic functions, etc.

We introduce also a new family of cryptographic hash function NaSHA, which was one of the 1^{st} Round candidates to NIST SHA-3 competition. NaSHA has compression function based on the quasigroup string transformation $\mathcal{MT}$ and its implementation use novel design principle - *use of different quasigroups for every application of component quasigroup transformations in every iteration of the compression function and, much more, the used quasigroups are functions of the processed message block.* This can be achieved by using quasigroups generated by the extended Feistel networks with tunable parameters in them. NaSHA uses quasigroups of huge order 2^{64} and starting bijection of order 2^8. The name NaSHA in the macedonian language means "OURS".

We introduce a new family of tweakable block ciphers Alex'smile-(B, I, G) with 128-bit block size implementations ($B = 4$) for $G \in \{128, 192, 256\}$, $I = 2$. Encryption and decryption algorithms use quasigroup string transformations defined by the extended Feistel networks, three S-boxes chosen by the tweak and a fixed 4×4 maximum distance separable (MDS) matrix over $GF(2^8)$. Quasigroup operations are of order 2^{32} are defined only by xoring and table lookups.

3.1 Hash functions

Hash functions are functions that take a variable-size input messages and map them into fixed-size output, known as hash result, message digest, hashcode etc. They are considered as "Swiss army knife" because of their versatile application in checking data integrity, digital signature schemes, commitment schemes, password based identification systems, digital timestamping schemes, pseudo-random string generation, key derivation, one-time passwords etc. They are basic security mechanism for local or decentralized file systems, for P2P file-sharing, for decentralized revision control tools and for intrusion detection systems. They are also used in popular software package tools such as Microsoft CLR strong names, Python setuptools, Debian control files, Ubuntu system-integrity-check, etc. Hash functions can be divided in cryptographic hash functions (manipulation detection codes - MDCs) and keyed hash functions (message authentication codes - MACs). MACs use additional input of fixed length, known as a key and they are basic cryptographic tool for providing authentication in a wide range of applications. Further cryptographic hash functions can be divided into one way hash functions and collision-resistant hash functions. The following informal definitions are given by Preneel [116]. A **one-way hash function** is a function h satisfying the following conditions:

- The input X can be of arbitrary length and the result $h(X)$ has a fixed length of n bits.
- Given h and X, the computation of $h(X)$ must be "easy".
- The hash function must be one-way in the sense that given a Y in the image of h, it is "hard" to find a message X such that $h(X) = Y$ (*preimage-resistance*) and given X and $h(X)$ it is hard to find a message $X' \neq X$ such that $h(X') = h(X)$ (*second preimage-resistance*).

A **collision-resistant hash function** is a function h that satisfies the following conditions:

– The input X can be of arbitrary length and the result $h(X)$ has a fixed length of n bits.

– Given h and X, the computation of $h(X)$ must be "easy".

– The function must be preimage-resistant and second preimage-resistant.

– The hash function must be collision-resistant: this means that it is "hard" to find two distinct messages that hash to the same result (i.e., find X and X', $X' \neq X$, such that $h(X) = h(X')$).

A **message authentication code** or **MAC** is a function h that satisfies the following conditions:

– The input X can be of arbitrary length and the result $h(K, X)$ has a fixed length of n bits.

– Given h, K and X, the computation of $h(K, X)$ must be "easy".

– Given a message X, it must be "hard" to determine $h(K, X)$. Even when a large set of pairs $\{X_i, h(K, X_i)\}$ is known, it is "hard" to determine the key K or to compute $h(K, X')$ for any new message $X' \neq X$ (*adaptive chosen text attack*).

Almost every hash function consists of *compression function* C with fixed-size input and output, and *domain extender* that, from the given compression function, produces a function with a variable-size input. Often, the message M is divided in blocks $M_0, M_1, \ldots, M_n$ with fixed size of b bits, which then are processed iteratively by the compression function. Usually, some padding rule which often contains an encoding of the length of the message is used for the last message block. The compression function C takes two inputs: a chaining variable H_i and a message block M_i. The starting chaining value is fixed to initial vector IV. After processing the last message block, the output from C is send to the output transformation f which compute the hash result $h(M)$. This can be represented as

$$H_0 = IV$$

$$H_{i+1} = C(H_i, M_i),\ 0 \leqslant i \leqslant n$$

$$h(M) = f(H_{n+1})$$

The compression function for practical hash functions can be made from existing block ciphers or can be made specially, with optimized performance in mind. The simplest and most commonly used domain extender is the Merkle-Damgård construction, but recently many other are also used, like HAIFA, sponge construction, wide-pipe and double-pipe construction, enveloped MD construction, etc. The most often used constructions from block cipher are:

Davies-Meyer: $H_{i+1} = C(H_i, M_i) = E_{M_i}(H_i) \oplus H_i$

Miyaguchi-Preneel: $H_{i+1} = C(H_i, M_i) = E_{g(H_i)}(M_i) \oplus H_i \oplus M_i$

Matyas-Meyer-Oseas: $H_{i+1} = C(H_i, M_i) = E_{g(H_i)}(M_i) \oplus H_i$

The usual target of the attacks to hash functions is to find preimage, second preimage or collision. There is one group of attacks, known as *generic attacks*, that can be apply to any recent or future hash function. Generic attacks depend only of one generic parameter - the length of message digest and they provide the upper security bounds to the given hash function. Assume now that message digest from hash function is n-bit long. Time complexity of the generic random (second) preimage attack is $\mathcal{O}(2^n)$ operations, and the time complexity of the generic birthday attack is $\mathcal{O}(2^{n/2})$ operations, where the "operations" correspond to the computation of the hash result for a random input. Hash function is an *ideal secure* if the best attacks are the generic attacks. Second group of attacks are the *short-cut attacks*, in which for breaking the hash function, the attacker uses the flows in its design and internal structure. Hash function is said to be broken if there is a short-cut attack faster than the best generic attack.

The most often used and standardized cryptographic hash functions are MD4, MD5, SHA-0, SHA-1 and the family of SHA-2 hash functions, which are the last standard issued by NIST. In the light of recent differential attacks by Wang et al [139, 137, 136, 138], now is ongoing the NIST SHA-3 competition for new standard for cryptographic hash functions.

Usually MAC takes a secret key to generate a checksum (MAC-value, authentication tag) for a given message (*signing*) or to verify an existing checksum (*verifying*). The same iterated model as the one defined for cryptographic hash functions is used also for MAC constructions, and here one needs to consider forgery attacks based on internal collisions. The most common approach is to base the compression function on an existing cryptographic primitive, either a block cipher or a cryptographic hash function. One of the most popular construction from the hash function h is HMAC, suggested by Bellare et al. [2]. HMAC value is obtained by

$$HMAC(K_1||K_2, X) = h(K_2||h(K_1||X))$$

where the keys K_1 and K_2 are usually dependent on each other.

3.1.1 Cryptographic hash functions with quasigroups

First attempts for using quasigroups and quasigroup transformations for creating cryptographic hash functions do not have actual implementations. One of the earliest attempt is the work of Markovski et al [78]. They employ two previously defined quasigroup transformations QM1 and QM2 for obtaining hash functions, but they are not enough analyzed and elaborated. QM1 transforms string with length $2m$ in a string with same length, so the message M first is pad to be with the length $2mn$ - $a_1a_2\dots a_{2mn}$, and than is divide in n blocks B_i. We apply QM1 to everyone of the blocks B_i and $QM1(B_i) = g_1^i g_2^i \dots g_{2m}^i$. A hash function H can be defined by $H(M) = h_1h_2\dots h_{2m}$, where

$$h_i = \bigoplus_{j=1}^{n} g_i^j, \quad i = 1, 2, \dots 2m$$

QM2 transforms string with length m in a string with double length, so it can be used for hash results with length $2m$. Let the message $M = a_1 \dots a_r,\ r \geqslant m$ (if M has small length, padding rule can be employed), let $j = r - m$ and

$$QM2(a_1 \dots a_m) = g_1^1 \dots g_{2m}^1$$

$$QM2(g_{m+2}^1 \dots g_{2m}^1 a_{m+1}) = g_1^2 \dots g_{2m}^2$$

$$QM2(g_{m+2}^2 \dots g_{2m}^2 a_{m+2}) = g_1^3 \dots g_{2m}^3$$

$$\vdots$$

$$H(M) = QM2(g_{m+2}^j \dots g_{2m}^j a_r) = g_1^{j+1} \dots g_{2m}^{j+1}$$

This definition uses only one character of the message in every iterative step of compression function QM2, which is very impractical.

Another early attempt to use quasigroups for creating hash function is given by Dvorský et al [29], and preimage, second preimage and collision attacks against this hash function for some special quasigroups are given by Vojvoda [134]. Snášel et al [130] continue to develop this hash function. Let $(Q, \circ)$ be a quasigroup of order r and let a be a fixed element from Q. They define function $H_a(q_1q_2\dots q_n) = ((\dots((a * q_1) * q_2) * \dots) * q_n$ as hash function. Also, they proposed to use huge quasigroups obtained by isotopies from the quasigroup of modular substraction, given with

$$a \circ b = \pi^{-1}((\omega(a) + n - \rho(b)) \ \ mod\ n)$$

This quasigroup has a right unit 0 and is isotopic to the group $(\mathbb{Z}_n, +)$ (see [135]). If n is an even number, $(\mathbb{Z}_n, +)$ has a proper subgroup, the subset of even numbers. Some arguments why to use quasigroup of modular substraction as a carrier, are given in Ochodková et al [28]. They suggest that one can use also huge quasigroups isotopic to the following quasigroup

$$a \circ b = \overline{(h \cdot a + k \cdot b + l)} \ \ mod\ n$$

where h, k, l are integers and $GCD(h, n) = 1 = GCD(k, n)$ (the inverses h^{-1} and k^{-1} exist). But this quasigroup also is isotopic to the group $(\mathbb{Z}_n, +)$ with isotopes $\omega(x) = x \cdot h^{-1}$, $\rho(x) = x \cdot k^{-1}$ and $\pi(x) = \overline{(x + l)} \ \ mod\ n$, where $x \in \mathbb{Z}_n$. The authors suggest that for real usage of proposed hash function, arithmetic of long numbers (i.g. 512 bits) must be adopted.

Another generic quasigroup based hash function Edon-F without implementation, is given in [79]. Here we will explain only the used quasigroup string transformation $f : Q^n \to Q^n$, which is in fact one-way function. It uses two auxiliary vectors $U = (u_1, u_2, \ldots, u_n)$ and $V = (v_1, v_2, \ldots, v_n)$. Vector V at the beginning is fixed to some random values. Let $a_1, a_2, \ldots, a_n$ be a given message and let $(Q, *)$ be a quasigroup. At first, the values u_1 and u_2 are computed by

$$u_1 = ((a_1 * a_2) * (a_2 * a_1)) * v_1$$

$$u_2 = ((\ldots((a_1 * a_2) * a_3) * \ldots) * a_n) * v_2$$

Values u_i, for $3 \leqslant i \leqslant n$ are computed by

$$u_i = (a_i * u_{i-1}) * (a_i * v_i)$$

After that, the new values of V are computed by the rules of the same kind:

$$v_1 = ((c_1 * c_2) * (c_2 * c_1)) * u_1$$

$$v_2 = ((\ldots((c_1 * c_2) * c_3) * \ldots) * c_n) * u_2$$

$$v_i = (c_i * v_{i-1}) * (c_i * u_i), \quad 3 \leqslant i \leqslant n$$

where $c_i = u_i * a_i, 1 \leqslant i \leqslant n$. Then, $f(a_1, a_2, \ldots, a_n) = (v_1, v_2, \ldots, v_n)$.

A generic hash function with reverse quasigroup string transformation $\mathcal{R}$ (1.12) has been described in [43]. First implementation of this hash function with name: Edon-$R(256, 384, 512)$ has been described in [37]. But the most famous of its implementation is the Edon-$\mathcal{R}$, the fastest candidate of NIST SHA-3 competition, designed by Gligoroski et al [46]. This implementation is explained here.

The quasigroup reverse string transformation $\mathcal{R} : Q_q^4 \to Q_q^2$ is used for calculating new chaining value in following way

$$\mathcal{R}(H_i^1, H_i^2, M_i^1, M_i^2) = (H_{i+1}^1, H_{i+1}^2)$$

where

$$H_{i+1}^1 = \overline{M}_i^1 * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1)$$

$$H_{i+1}^2 = (\overline{M}_i^1 * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1)) * (((H_i^2 * (\overline{M}_i^2 * M_i^1))*$$

$$*((\overline{M}_i^2 * M_i^1) * M_i^2)) * ((H_i^2 * (\overline{M}_i^2 * M_i^1)) * H_i^1))$$

Edon-$\mathcal{R}$ is wide-pipe iterative hash function with standard MD-straitening. Its compression function $\mathcal{R}$ uses huge quasigroups of order 2^{256} and 2^{512} (the biggest so far) and their operations are defined by isotopies of Abelian groups $((\mathbb{Z}_2^w)^8, +_8)$, where $w = 32, 64$ and $+_8$ is componentwise addition on two 8-dimensional vectors in $(\mathbb{Z}_2^w)^8$. Definition of quasigroup operations uses only bitwise xoring, left rotations and addition modulo 2^{32} and 2^{64} and is given by

$$X * Y = \pi_1(\pi_2(X) +_8 \pi_3(Y))$$

where $X = (X_0, X_1, \ldots, X_7),\ Y = (Y_0, Y_1, \ldots, Y_7) \in (\mathbb{Z}_2^w)^8$ and $\pi_i : \mathbb{Z}_2^q \to \mathbb{Z}_2^q,\ 1 \leqslant i \leqslant 3,\ q = 256, 512$ are permutations. Authors have proofs that the used quasigroups are non-associative, non-commutative and without identity.

Let $Q_{256} = \{0,1\}^{256}$ and $Q_{512} = \{0,1\}^{512}$. Transformations $\pi_i : Q_q \to Q_2$ $(q = 256, 512)$ are defined as:

$$\pi_1(X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7) = (X_5, X_6, X_7, X_0, X_1, X_2, X_3, X_4)$$

$$\pi_2 \equiv \hat{\mathbb{A}}_1 \circ ROTL^{r_{1,q}} \circ \mathbb{A}_2$$

$$\pi_3 \equiv \hat{\mathbb{A}}_3 \circ ROTL^{r_{2,q}} \circ \mathbb{A}_4$$

where $ROTL^r(X)$ can be expressed as a linear matrix $-$ vector multiplication over the ring $(\mathbb{Z}_2, +, \times)$, $\hat{\mathbb{A}}_i = C_i + \mathbb{A}_i \cdot X,\ i = 1,\ 3.$ Invertible matrices $\mathbb{A}_i,\ 1 \leqslant i \leqslant 4$, rotation constants $r_{1,q},\ r_{2,q}$ and constant vectors C_1, C_3 are given in [46].

Because the used quasigroups are constructed by isotopes from the *Abelian group* $((\mathbb{Z}_{2^w})^8, +_8)$, every X has inverse $-X = (-X_0, -X_1, \ldots, -X_7)$, where $-X_i$ is inverse element of X_i in abelian group $(\mathbb{Z}_{2^w}, +)$ ($+$ is addition modulo 2^w, 0 is unit). We are interesting in those elements X in $(\mathbb{Z}_{2^w})^8$ for which

$X_0 = X_1 = \ldots = X_7 = x$. We will represent $(x, x, x, x, x, x, x, x) = \mathbf{X}$. For these elements $\pi_1(\mathbf{X}) = \mathbf{X}$.

Observation For quasigroups in Edon-R we have:

$$\left.\begin{array}{l} A * C = \mathbf{X} \\ A * D = \mathbf{X} +_8 \mathbf{Z} \\ B * C = \mathbf{Y} \end{array}\right\} \Rightarrow B * D = \mathbf{Y} +_8 \mathbf{Z} \tag{3.1}$$

Proof. We have

$$\pi_1(\pi_2(A) +_8 \pi_3(C)) = \mathbf{X} \Rightarrow \pi_2(A) +_8 \pi_3(C) = \mathbf{X}$$

$$\pi_1(\pi_2(A) +_8 \pi_3(D)) = \mathbf{X} +_8 \mathbf{Z} \Rightarrow \pi_2(A) +_8 \pi_3(D) = \mathbf{X} +_8 \mathbf{Z} \Rightarrow$$

$$\pi_3(D) = \mathbf{X} +_8 \mathbf{Z} - \pi_2(A)$$

$$\pi_1(\pi_2(B) +_8 \pi_3(C)) = \mathbf{Y} \Rightarrow \pi_2(B) +_8 \pi_3(C) = \mathbf{Y} \Rightarrow \pi_2(B) = \mathbf{Y} - \pi_3(C)$$

Therefore,

$$B * D = \pi_1(\pi_2(B) +_8 \pi_3(D)) = \pi_1(\mathbf{Y} - \pi_3(C) +_8 \mathbf{X} +_8 \mathbf{Z} - \pi_2(A)) =$$

$$\pi_1(\mathbf{Y} +_8 \mathbf{Z}) = \mathbf{Y} +_8 \mathbf{Z}$$

If we choose B in a way that $\mathbf{Y} = 0$, then by choosing D and $\mathbf{Z}$ we can obtain $B * D$ whatever we want.

Another interesting application of quasigroups is given by Gligoroski et al. [40] as security fix of the MD4 family of hash functions with so called *quasigroup folding*, that use shapeless randomly generated quasigroup $(Q, *)$ of order 16. This technique is applied at the end of every iterative step of hash function. Every 32-bit register is seen as a concatenation of 8, 4-bit variables $a_1, a_2, \ldots, a_8$. Variables a_1, a_2, a_3, a_4 are replaced with b_1, b_2, b_3, b_4, where $b_1 = a_1 * a_5$, $b_2 = a_6 * a_2$, $b_3 = a_3 * a_7$ and $b_4 = a_8 * a_4$. Obtained impact on the speed is 2 time slower hash function. The similar technique has been used in [39], where new hash function SHA-1Q2 has been constructed from SHA-1. The new hash function uses the message expansion part with quasigroup folding and has only 8 internal iterative steps (it is 3% faster that SHA-1).

3.1.2 MACs with quasigroups

First application of quasigroup for creating authentication scheme is explained by Dénes and Keedwell in [21]. Let $(Q, \circ)$ be a quasigroup and let

$M = m_1 \dots m_n$, $m_i \in Q$, be a message that need to be signed with authentication tag $b_0 \dots b_{s-1}$, $b_j \in Q$. Message M is divided into s mutually disjoint subsets S_j, $0 \leqslant j < s$, where $|S_j| = t = [\frac{n}{s}]$ and $S_j = \{m_{j_1}, \dots, m_{j_t}\}$. The last subset S_{s-1} can contain $r \leqslant t$ elements. Then b_j can be calculate with

$$b_j = (\dots((m_{j_1} \circ m_{j_2}) \circ m_{j_3}) \circ \dots) \circ m_{j_t}$$

with exception of the last value b_{s-1} for which only r elements are used for calculating. After that, the message and signature are concatenated and sent. The security of this authentication scheme lies in how the sets S_j are created, and for that aim authors suggest the use of the Latin square L with elements $\{0, 1, \dots, s-1\}$ as a secret key. Positions in L are numbered from 1 to s for the first row, $s+1$ to $2s$ for the second row and so on, $(s-1)s+1$ to s^2 for the last row. When set S_j is forming, positions of j in L are read as $j_1, \dots, j_t$ and proper elements $m_{j_1}, \dots, m_{j_t}$ from the message M are chosen. The authors also suggest the use of the same structure for $(Q, \circ)$ and L, for saving memory. The process can be made faster by precomputing of the sets S_j.

Security of this scheme is analyzed by Dawson et al [17]. One problem with this scheme is that it does not have an output with fixed sizes, it is not really a MAC. Also, properties of the quasigroup $(Q, \circ)$ are not being utilized and it will work even in the case of a group instead of quasigroup.

Meyer in [97] describes proper quasigroup based MAC algorithm, known as QMAC. In QMAC, $(Q, \circ)$ is public and the secret key is the order in which the message elements are multiplied together to create the MAC-value, i.e. the parentheses scheme. Also in key is incorporated one fixed element c which serves to hide the innermost multiplications. Without c, one can start an adaptive chosen-text attack, described in [97]. The authentication tag for a message $M = m_1 \dots, m_t$ is computed by multiplying the message elements together in the order specified by the key K, except that every innermost multiplication $(m_i \circ m_{i+1})$ is replaced by $((m_i \circ c) \circ m_{i+1})$. This can be represented as $h_K(m_1, \dots, m_t)$. Security of this scheme relies on the structure of used quasigroup. Huge "highly non-associative" quasigroup without any structure are wanted. The author gives 3 different methods for constructing MAC value for large messages and we are going to explain only one. Let every message block consists of t elements over Q and let $|M| = Nt$, with padding.

$$H_0 = IV \in Q$$

$$H_{i+1} = H_i \circ h_K(m_{it+1}, \dots, m_{(i+1)t}), \ 0 \leqslant i \leqslant N-1$$

$$QMAC_K(M) = H_N$$

The author also give nice representation of the key and show that the size of the keyspace increases exponentially in the length of the key.

Another quasigroup based MAC is defined by Bakhtiari et al [1]. They first define the family of hash functions $H = \{h : Q^{q^2} \to Q^q\}$ and then they use the Wegman-Carter universal-hash construction [140]. Let $(Q, *)$ is quasigroup of order $q = 2^t$ end let $b = q/2$ isotopies of $(Q, *)$ are given as $(Q, *_1), \dots, (Q, *_b)$. Let M be a message with q^2 elements arranged in $q \times q$ matrix. Define the sets $S_{r,c} = \{r *_1 c, \dots, r *_b c\}$, $1 \leqslant r, c \leqslant q$. Hash result D is represented as q-tuple $(d_1, \dots, d_q)$ and at the beginning all $d_k = 1$. The final output is calculated by

$$d_{i *_k j} = m_{i,j} * d_{i *_k j},\ 1 \leqslant k \leqslant b,\ 1 \leqslant i, j \leqslant q$$

Secret key is quasigroup $(Q, *)$ and its b isotopies. Authors suggest the key to be represented as (K_1, K_2), where K_1 is critical set of the correspondent Latin square to $(Q, *)$ and K_2 is information about the used permutations for obtaining the isotopies. The authors suggest that it is enough for security to take $q = 16$ and $b = 8$. One problem with this MAC is that the authors did not give any discussion about key space, and its relation with order of the chosen quasigroup.

3.1.3 Family of cryptographic hash functions NaSHA-(m, k, r)

We use the quasigroup transformation $\mathcal{MT}$ (Definition 18) for definition of a new family of hash functions NaSHA-(m, k, r). The parameters m, k and r denote the length of the output hash result (the message digest), the complexity of $\mathcal{MT}$ and the order 2^{2^r} of used quasigroup respectively, so k is a positive even integer and m and r are positive integers.

The main transformation $\mathcal{MT}$ as a one-way function

First, we will show that the transformation $\mathcal{MT} : Q^t \to Q^t$ can be considered as a one-way function when $Q = \mathbb{Z}_{2^n}$ is enough big.

Let us take $k = 2$ for simplicity, and let a quasigroup $(Q, *)$, leaders l_1, l_2 and elements $c_1, c_2, \dots, c_t \in Q$ be given. Suppose that for some unknown $x_1, x_2, \dots, x_t \in Q$ we have $(c_1, c_2, \dots, c_t) = \mathcal{MT}(x_1, x_2, \dots, x_t)$ $= \rho(\mathcal{RA}_{l_1})(\mathcal{A}_{l_2}(x_1, x_2, \dots, x_t))$. Then there are unknown $y_1, y_2, \dots, y_t \in Q$ such that

$$\mathcal{A}_{l_2}(x_1, x_2, \dots, x_t) = (y_1, y_2, \dots, y_t) \tag{3.2}$$

and

$$\mathcal{RA}_{l_1}(\rho(y_1, \lfloor\frac{n}{2}\rfloor), \rho(y_2, \lfloor\frac{n}{2}\rfloor), \dots, \rho(y_t, \lfloor\frac{n}{2}\rfloor)) = (c_1, c_2, \dots, c_t). \tag{3.3}$$

From the equations (3.2) and (3.3) we obtain the following system of $2t$ equations with $2t$ unknowns.

$$\begin{cases} (l_2+x_1)*x_1 = y_1 \\ (y_1+x_2)*x_2 = y_2 \\ \dots \\ (y_{t-1}+x_t)*x_t = y_t \end{cases} \tag{3.4}$$

$$\begin{cases} \rho(y_t,\lfloor\frac{n}{2}\rfloor)*(\rho(y_t,\lfloor\frac{n}{2}\rfloor)+l_1) = c_t \\ \rho(y_{t-1},\lfloor\frac{n}{2}\rfloor)*\rho((y_{t-1},\lfloor\frac{n}{2}\rfloor)+c_t) = c_{t-1} \\ \dots \\ \rho(y_1,\lfloor\frac{n}{2}\rfloor)*(\rho(y_1,\lfloor\frac{n}{2}\rfloor)+c_2) = c_1. \end{cases} \tag{3.5}$$

The subsystem (3.5) consists of t equations with t unknowns of kind $y*(y+a)=b$. As much as we know, there is no explicit formula to find the unknown y, so one has to check for each $y\in Q$ if the equation $y*(y+a)=b$ is satisfied. By Proposition 1.14 one has to make, roughly, $2^n - 1/2^n \approx 2^n$ checks, i.e., a solution can be found after 2^{n-1} checks on average. In the same way, by checking, solutions $x_1, x_2, \dots, x_t$ can be found. Altogether, for finding a solution of the system consisting of (3.4) and (3.5) one has to make, on average, $2t2^{n-1} = 2^n t$ checks. Thus, we have the following properties.

Proposition 25 *The system of equations (3.4) and (3.5) can be solved after $2^n t$ checks on average.* □

Proposition 26 *If Q is sufficiently large and $(Q,*)$ is an arbitrary quasigroup, chosen uniformly at random, the problem of finding a preimage of the transformation $\mathcal{MT}$ is computationally infeasible.* □

NaSHA-(m,k,r) hash algorithm

```
NaSHA-$(m,k,r)$ hash algorithm
Input: A positive even integer $k$ and positive integers $m$ and $r$
        such that $m > 2^r$, and an input message $M$.
Output: A hash value NaSHA-$(m,k,r)(M)$ of m bits.
  1. Denote by $n$ the smallest integer such that $m \leqslant 2^n$.
     (For example, $n$=8 for $m$=224 and $n$=9 for $m$=384.)
  2. Pad the message M, so that the length of the padded message $M'$ is
     a multiple of $2^{n+1}$, $|M'| = 2^{n+1}N$ for some $N$.
     Separate $M'$ in N $2^{n+1}$-bit blocks, $M' = M_1||M_2||\dots||M_N$, $|M_i| = 2^{n+1}$.
  3. Initialize the initial value $H_0$, which is a $2^{n+1}$-bit word.
  4. The first message block $M_1$ and the initial value $H_0$
     separate to $q = 2^{n-r+1}$ $2^r$-bits words:
     $M_1 = S_1||S_3||S_5||\dots||S_{2q-3}||S_{2q-1}$,
     $H_0 = S_2||S_4||S_6||\dots||S_{2q-2}||S_{2q}$, $(|S_i| = 2^r)$ and form the word
  $S^{(0)} = S_1||S_2||S_3||S_4||\dots||S_{2q-3}||S_{2q-2}||S_{2q-1}||S_{2q}$.
5. Choose leaders $l_i$ as functions that depend on $S_1, S_2, S_3, \dots, S_{2q}$
   and a suitable linear transformation $LinTr_{2^{n+2}}$.
6. Choose two quasigroups $(\{0,1\}^{2^r}, *_1)$ and $(\{0,1\}^{2^r}, *_2)$
   (one for $\mathcal{A}$ and one for $\mathcal{RA}$ transformation)
   and compute the string of bits $S^{(N-1)}$ as follows:
     for $i = 1$ to $N-1$ do
         $A_1||A_2||A_3||\dots||A_{2q} \leftarrow \mathcal{MT}(LinTr^{2q}_{2^{n+2}}(S^{(i-1)}))$
         $B_1||B_2||B_3||\dots||B_{q-1}||B_q \leftarrow M_{i+1}$,
         $S^{(i)} := B_1||A_2||B_2||A_4||\dots||B_{q-1}||A_{2q-2}||B_q||A_{2q}$,
     end
7. Choose two quasigroups $(\{0,1\}^{2^r}, *_1)$ and $(\{0,1\}^{2^r}, *_2)$ and compute
   $\mathcal{MT}(LinTr^{2q}_{2^{n+2}}(S^{(N-1)})) := A_1||A_2||A_3||\dots||A_{2q}$. Then
     NaSHA-$(m, k, r)(M) = A_4||A_8||\dots||A_{2q-4}||A_{2q}$ $(\mathtt{mod}\ 2^m)$.
```

We emphasize that some steps (e.g., Step 5) need more detailed elaborations in concrete implementations.

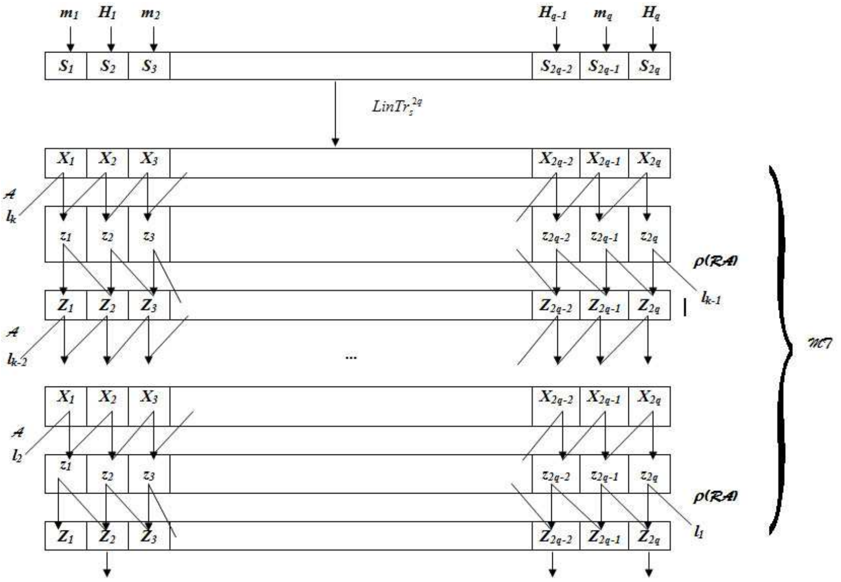

Figure 3: NaSHA-(m, k, r)

Implementation of NaSHA-$(m, 2, 6)$ hash functions for $m \in \{224, 256, 384, 512\}$

Here we give a complete implementation of NaSHA-$(m, 2, 6)$ algorithm where $m \in \{224, 256, 384, 512\}$. The used quasigroup of order $2^{2^6} = 2^{64}$ is constructed by extended Feistel networks. This implementation has been submitted as a candidate in the SHA-3 competition of The American National Institute of Standards and Technology, NIST. Now it is one of the 51 selected 1st round candidates [86, 88].

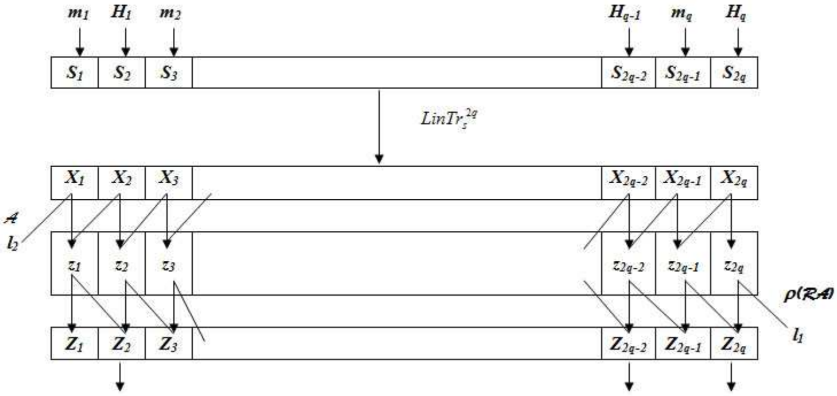

Figure 4: NaSHA-$(m, 2, 6)$

Padding

The padding consists of the standard Merkle-Damgård strengthening [96]. Denote by M the bit input message of length $s = |M| < 2^{128}$.

1. Denote by q the smallest nonnegative integer such that

$$s + q + 1 \equiv 384 \ (mod\ 512)$$

for $m = 224$ and $m = 256$, and

$$s + q + 1 \equiv 896 \ (mod\ 1024)$$

for $m = 384$ and $m = 512$.

2. Let 0_q denote the binary word consisting of q zeros, and let b_s be the binary presentation of s by 128 bits.

3. Append to the message M the words 1, 0_q and b_s.

The padding of M is the message $M' = M||1||0_q||b_s$ and for $m = 256$ is a multiple of 512 and for $m = 512$ is a multiple od 1024. This implementation of NaSHA hash algorithm accepts messages of length up to $2^{128} - 1$ bits.

Starting bijection

As starting bijection $f : \mathbb{Z}_2^8 \rightarrow \mathbb{Z}_2^8$ for creating extended Feistel network we use improved AES S-box with the APA structure from Cui and Cao [12], given on Table 3.1 in hexadecimal notation.

f	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	8c	90	d9	c1	46	63	53	f1	61	32	15	3e	26	9a	97	2e
1	d8	a0	99	9e	c0	95	67	b7	6d	e0	f3	28	20	86	b6	ef
2	4b	31	b5	d2	13	39	6c	a5	03	3f	4d	34	f9	ec	8e	17
3	c5	25	3c	89	c9	2b	3a	c2	6e	c6	aa	91	49	18	93	de
4	0d	6f	65	af	92	a7	f6	a6	40	b9	ed	b0	c3	d7	7d	7c
5	54	59	df	2f	da	a4	05	94	9b	72	01	74	a9	f7	81	e9
6	1f	b3	eb	cf	8	47	52	36	bc	16	29	76	12	fa	9c	8a
7	5b	a8	43	d1	79	85	42	82	c7	a1	78	4f	e2	35	ea	ad
8	dc	0e	d3	2d	6a	5a	44	ab	c8	e5	37	0a	6b	51	e3	14
9	cd	56	4a	d6	08	83	bb	33	e1	30	4e	24	5e	b4	00	48
a	5f	22	0b	50	3d	80	1a	bf	cc	ff	64	87	1b	c4	07	f8
b	0c	d4	ac	02	10	84	7e	69	70	60	55	2a	21	57	23	66
c	62	73	cb	41	58	71	77	1c	7b	8f	9f	9d	a3	b1	7f	5d
d	f4	06	ae	d5	e6	3b	ba	Fe	96	e7	0f	45	2c	f0	fc	bd
e	e4	98	fb	ca	11	f5	dd	7a	5c	fd	ce	88	d0	68	8d	4c
f	be	04	38	1d	1e	f2	27	19	b2	75	a2	ee	db	b8	09	8b

Table 3.1: The starting bijection $f = f(m||n)$

Linear transformation

The algorithm of NaSHA hash functions uses the following linear transformations.

Denote by $LinTr_{512}$ and by $LinTr_{256}$ the transformations of the sets $\{0,1\}^{2028}$ and $\{0,1\}^{1024}$ respectively, defined by

$$LinTr_{512}(S_1||S_2||\dots||S_{31}||S_{32}) = (S_7 \oplus S_{15} \oplus S_{25} \oplus S_{32})||S_1||S_2||\dots||S_{31},$$

$$LinTr_{256}(S_1||S_2||\dots||S_{15}||S_{16}) = (S_4 \oplus S_7 \oplus S_{10} \oplus S_{16})||S_1||S_2||\dots||S_{15},$$

where S_i are 64-bits words, $\oplus$ denotes the operation XOR on 64-bits words, and the operation $||$ denotes the concatenation of words.

Note that $LinTr_{512}$ is in fact the LFSR obtained from the primitive polynomial $x^{32} + x^{25} + x^{15} + x^7 + 1$ over the Galois field GF(2), applied in parallel 64 times, while $LinTr_{256}$ is obtained in the same way from the primitive polynomial $x^{16} + x^{10} + x^7 + x^4 + 1$. As a consequence we have the following.

Proposition 27 *$LinTr_{512}$ is a permutation of the set $\{0,1\}^{2028}$ and $LinTr_{256}$ is a permutation of the set $\{0,1\}^{1024}$.* □

Quasigroup operations via extended Feistel networks

From the starting bijection f we define three extended Feistel networks $F_{a_1,b_1,c_1}, F_{a_2,b_2,c_2}, F_{a_3,b_3,c_3} : \mathbb{Z}_2^{16} \to \mathbb{Z}_2^{16}$ by

$$F_{a_i,b_i,c_i}(l_8||r_8) = (r_8 \oplus a_i)||(l_8 \oplus b_i \oplus f(r_8 \oplus c_i)),$$

where l_8 and r_8 are 8-bit variables, and a_i, b_i, c_i are 8-bit words that are defined before each application of $\mathcal{MT}$. Denote by f' the bijection $F_{a_1,b_1,c_1} \circ F_{a_2,b_2,c_2} \circ F_{a_3,b_3,c_3} : \mathbb{Z}_2^{16} \to \mathbb{Z}_2^{16}$.

By using the bijection f' we define a quasigroup operation on $\mathbb{Z}_2^{64}$ which is going to be used for the additive string transformation $\mathcal{A}$ as follows. Create the Feistel networks $F_{\alpha_1,\beta_1,\gamma_1} : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ and $F_{A_1,B_1,C_1} : \mathbb{Z}_2^{64} \to \mathbb{Z}_2^{64}$ by

$$F_{\alpha_1,\beta_1,\gamma_1}(l_{16}||r_{16}) = (r_{16} \oplus \alpha_1)||(l_{16} \oplus \beta_1 \oplus f'(r_{16} \oplus \gamma_1)),$$

$$F_{A_1,B_1,C_1}(l_{32}||r_{32}) = (r_{32} \oplus A_1)||(l_{32} \oplus B_1 \oplus F_{\alpha_1,\beta_1,\gamma_1}(r_{32} \oplus C_1)),$$

where l_{16}, r_{16} are 16-bit variables, $\alpha_1, \beta_1, \gamma_1$ are 16-bit words, l_{32}, r_{32} are 32-bit variables and A_1, B_1, C_1 are 32-bit words. The constant words will be defined latter. The function F_{A_1,B_1,C_1} is a orthomorphism (complete mapping) in the group $(\mathbb{Z}_2^{64}, \oplus)$, and then the operation $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_1,\beta_1,\gamma_1,A_1,B_1,C_1}$ defined by

$$x *_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_1,\beta_1,\gamma_1,A_1,B_1,C_1} y = F_{A_1,B_1,C_1}(x \oplus y) \oplus y$$

is a quasigroup operation in $\mathbb{Z}_2^{64}$.

By using the bijection f' we define also a quasigroup operation in $\mathbb{Z}_2^{64}$ which is going to be used for the reverse additive string transformation $\mathcal{RA}$ as follows. Create the Feistel networks $F_{\alpha_2,\beta_2,\gamma_2} : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ and $F_{A_2,B_2,C_2} : \mathbb{Z}_2^{64} \to \mathbb{Z}_2^{64}$ by

$$F_{\alpha_2,\beta_2,\gamma_2}(l_{16}||r_{16}) = (r_{16} \oplus \alpha_2)||(l_{16} \oplus \beta_2 \oplus f'(r_{16} \oplus \gamma_2)),$$

$$F_{A_2,B_2,C_2}(l_{32}||r_{32}) = (r_{32} \oplus A_2)||(l_{32} \oplus B_2 \oplus F_{\alpha_2,\beta_2,\gamma_2}(r_{32} \oplus C_2)),$$

where l_{16}, r_{16} are 16-bit variables, $\alpha_2, \beta_2, \gamma_2$ are 16-bit words, l_{32}, r_{32} are 32-bit variables and A_2, B_2, C_2 are 32-bit words. The constant words will be defined latter. The function F_{A_2,B_2,C_2} is an orthomorphism (complete mapping) in the group $(\mathbb{Z}_2^{64}, \oplus)$, and then the operation $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_2,\beta_2,\gamma_2,A_2,B_2,C_2}$ defined by

$$x *_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_2,\beta_2,\gamma_2,A_2,B_2,C_2} y = F_{A_2,B_2,C_2}(x \oplus y) \oplus y$$

is a quasigroup operation in $\mathbb{Z}_2^{64}$.

In such a way we achieve for each application of $\mathcal{MT}$ to use different quasigroup operations $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_1,\beta_1,\gamma_1,A_1,B_1,C_1}$ for the transformation $\mathcal{A}$ and $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_2,\beta_2,\gamma_2,A_2,B_2,C_2}$ for the transformation $\mathcal{RA}$.

Chaining initial vectors

The definition of NaSHA-(m, k, r) hash function includes one initial string H_0. The initial strings that we are using are the following, represented in hexadecimal as concatenation of 64-bit chunks.

1. $m = 224, H_0 =$
6a09e667f3bcc908, cbbb9d5dc1059ed8, bb67ae8584caa73b, 629a292a367cd507, 3c6ef372fe94f82b, 9159015a3070dd17, a54ff53a5f1d36f1, 152fecd8f70e5939

2. $m = 256, H_0 =$
510e527fade682d1, 67332667ffc00b31, 9b05688c2b3e6c1f, 8eb44a8768581511, 1f83d9abfb41bd6b, db0c2e0d64f98fa7, 5be0cd19137e2179, 47b5481dbefa4fa4

3. $m = 384, H_0 =$
6a09e667f3bcc908, cbbb9d5dc1059ed8, bb67ae8584caa73b, 629a292a367cd507, 3c6ef372fe94f82b, 9159015a3070dd17, a54ff53a5f1d36f1, 152fecd8f70e5939, 510e527fade682d1, 67332667ffc00b31, 9b05688c2b3e6c1f, 8eb44a8768581511, 1f83d9abfb41bd6b, db0c2e0d64f98fa7, 5be0cd19137e2179, 47b5481dbefa4fa4

4. $m = 512, H_0 =$
2dd8a09a3c4e3efb, e07688dc6f166b73, 061a77a060948dcd, 0c34aa2a315e01d5, 8a47ea1880559ce6, c785f4364a0b98f4, 9f22535b264607a8, 53a8c8ca56e1288c, 2547d84e9ccde59d, 3c1563a9317c57a1, 9486eb50c7d8037f, 77341edad21e9a40, c0f905d741c9cb74, d648813e45121dbb, ad0d1e41a985e51e, 4cf768fc7df11b00

The initial values are randomly generated. If somebody has suspicions for NaSHA initial chaining values, at any time, they can be replaced by other, without changes in the security or in the performances.

Definition of the leaders and constants

Before every computation $\mathcal{MT}(S_1||S_2||S_3||\dots||S_{2q-1}||S_{2q})$, where S_i are 64-bit words, we define the 64-bit leaders l_1 of $\mathcal{RA}$ and l_2 of $\mathcal{A}$, the 8-bit words $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c_3$, the 16-bit words $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2$, γ_2 and the 32-bit words $A_1, B_1, C_1, A_2, B_2, C_2$.

For $m = 224$ and 256, necessary definitions are this ones:

$$l_1 = S_1 + S_2, \quad l_2 = S_3 + S_4,$$

$$a_1||b_1||c_1||a_2||b_2||c_2||a_3||b_3 = S_5 + S_6, \quad c_3 = a_1$$

$$\alpha_1||\beta_1||\gamma_1||\alpha_2 = S_7 + S_8,$$

$$\beta_2||\gamma_2 = (S_9 + S_{10})(\texttt{mod } 2^{32}),$$

$$A_1||B_1 = S_{11} + S_{12}, \quad C_1||A_2 = S_{13} + S_{14}, \quad B_2||C_2 = S_{15} + S_{16}.$$

For $m = 384$ and 512, necessary definitions are:

$$l_1 = S_1 + S_2 + S_{28} + S_{30}, \quad l_2 = S_3 + S_4 + S_{29} + S_{31},$$

$$a_1||b_1||c_1||a_2||b_2||c_2||a_3||b_3 = S_5 + S_6 + S_{17} + S_{18}, \quad c_3 = a_1$$

$$\alpha_1||\beta_1||\gamma_1||\alpha_2 = S_7 + S_8 + S_{19} + S_{20},$$

$$\beta_2||\gamma_2 = (S_9 + S_{10} + S_{21} + S_{22})(\texttt{mod } 2^{32}),$$

$$A_1||B_1 = S_{11} + S_{12} + S_{23} + S_{27}, \quad C_1||A_2 = S_{13} + S_{14} + S_{24} + S_{26},$$

$$B_2||C_2 = S_{15} + S_{16} + S_{25} + S_{32}.$$

Here, the addition + is modulo 2^{64}.

Design rationales

THE CHOICE OF THE STARTING BIJECTION. As NaSHA starting bijection we wanted to use some publicly known function in order to prevent the suspicious of possible "trap door" in the implementation. We considered several possibilities: the AES S-box [15], the improved AES S-box from Liu and all [68] and the improved AES S-box with the APA structure from Cui and Cao [12]. All three runners have some pros and cons. The AES S-box is the most famous and the most investigated S-box in cryptology, with good differential and linear resistance and high algebraic degree. But it has simple algebraic structure with only 9 terms. The improved AES S-boxes has also good differential resistance with differential 4-uniformity and good linear resistance. They have the same algebraic degree as AES S-box, but they have much bigger algebraic complexity of 255 terms for the first, and 253 terms for the second, S-box. Their inverse S-boxes have high algebraic complexity of 255 terms as AES inverse S-box. But both are not enough studied from other authors. Our winner f is the third solution, because of its algebraic complexity and because it is a little bit more studied than the second solution. The function f also satisfies the condition $f(0) \neq 0$ that is needed by our extended Feistel network to derive a non-idempotent and a non-associative quasigroup. In case of suspicion of a trapdoor being built into the hash, the current S-box might be replaced by other two candidates.

THE CHOICE OF THE LINEAR TRANSFORMATION. The linear transformation is used for obtaining suitable diffusion of the input of 64-bit words. We use LFSRs for obtaining linear transformation that is a bijection and that can be easily computed. For that aim we use primitive polynomials over the Galois field GF(2), from the Rajski's list [117]. The degree of the primitive polynomial for 224 and 256 hash needs to be 16, and 32 for 384 and 512 hash. Since the algorithm applies the linear transformation 16 (i.e, 32) times, we take the primitive polynomials with 5 terms. Any other polynomial that fulfils these requirements is a good choice too.

THE CHOICE OF THE QUASIGROUP TRANSFORMATIONS. By our experience and some theoretical results we found that the quasigroup transformations are good nonlinear building blocks for designing different cryptographic primitives. We use quasigroups of huge order 2^{64} and they are defined by extended Feistel networks, defined in [87]. Our algorithm can also be implemented by quasigroups of order 2^{32}, 2^{128}, 2^{256} etc, but we found that the choice of order 2^{64} is optimal for obtaining tradeoff between security and speed.

THE CHOICE OF THE EXTENDED FEISTEL NETWORKS. It is not easy to define a workable quasigroup of huge order, like 2^{64}, having good cryptographic properties. Our choice were the extended Feistel networks because they produce shapeless quasigroups, and they allow to insert tunable parameters in their definition. We used that feature to obtain different quasigroups for every application of component quasigroup transformations in every iteration of the compression function and, much more, the used quasigroups are functions of the processed message block.

We are using 9 8-bit words $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3, c3$, 6 16-bit words $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2$, γ_2 and 6 32-bit words $A_1, B_1, C_1, A_2, B_2, C_2$ in every iteration of the compression function and pass them to extended Feistel networks. The way of their definition was leaded by the idea all bits of the processed input block to be included.

If instead of extended Feistel network F, we were using extended Feistel network F^2, the obtained quasigroups will be also suitable for cryptographic purposes. Choice of F instead of F^2 and shapeless quasigroups instead of quasigroups suitable for cryptographic purposes, again was tradeoff between security and speed.

THE CHOICE OF THE COMPOSITE MAPPINGS IN THE MAIN TRANSFORMATION AND THE TUNABLE SECURITY PARAMETER k. In general, the main transformation $\mathcal{MT}$ can be defined as any composition of the transformations $\mathcal{A}$ and $\mathcal{RA}$. Having in mind the properties of the extended Feistel networks, where the starting bijection influences mostly the right half of the output result, we are going to use the transformation $\mathcal{RA}$ after rotating left for 32 bits the obtained 64-bit words from $\mathcal{A}$. In such a way, a homogeneous spreading of the starting bijection is obtained. Also, by the transformation $\mathcal{A}$ the influence of the input bits are spreading only in the right part of the output, which is why $\mathcal{RA}$ is defined as a reverse way of $\mathcal{A}$. In the end, we obtain every bit of an input block to influence almost all bits of the output blocks of $\mathcal{RA} \circ \mathcal{A}$.

The tunable security parameter of the NaSHA hash algorithm is the complexity k of the main transformation $\mathcal{MT}$, since we define $\mathcal{MT}$ as composition of k mappings of kind $\mathcal{RA}$ and $\mathcal{A}$, applied consecutively. The choice of higher values of k will give stronger security, but lower speed. Our choice, recommendation and low bound is $k = 2$ (there is no upper bound). We believe that the cryptanalysis will become practical if $k = 1$, that will happen if $\mathcal{MT} = \mathcal{A}$ or $\mathcal{MT} = \mathcal{RA}$.

Avalanche effect

We tested the avalanche propagation of one bit differences in the compression function of NaSHA-$(m, 2, 6)$, where $m \in \{224, 256, 384, 512\}$, in two cases: when the initial message consists of all zeros and when the initial message is randomly generated. We present in Tables 3.2 and 3.3 the obtained results for messages of length 8, 80, 800, 8000 and 80000 bits, where minimum, average and maximum different bits and standard deviation are given. Table 3.2 is for initial messages consisting of all zeros and Table 3.3 is for randomly generated initial message. One can see that in every case the Hamming distance is around $m/2$, or one bit difference of input bits produces about 50% different output bits, as it would be expected in theoretical models of ideal random functions.

Performances

Memory requirements for implementing NaSHA are quite small, only 0.625KB (0.25KB for starting bijection and 0.375 KB for 48 64-bit initial values). We tested the implementation on 32-bit and 64-bit architecture and for results of NaSHA on different configurations one can see EBASH project web site [6].

1.

a. Description of the platform: Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running

n	8 bits	80 bits	800 bits	8000 bits	80000 bits
224	$min = 42\%$ $avg = 50.06\%$ $max = 56\%$ $sd = 4.44$	$min = 41\%$ $avg = 49.86\%$ $max = 57\%$ $sd = 3.48$	$min = 41\%$ $avg = 50.21\%$ $max = 60\%$ $sd = 3.39$	$min = 38\%$ $avg = 49.97\%$ $max = 63\%$ $sd = 3.40$	$min = 35\%$ $avg = 50.02\%$ $max = 63\%$ $sd = 3.41$
256	$min = 45\%$ $avg = 49.12\%$ $max = 55\%$ $sd = 2.91$	$min = 43\%$ $avg = 50.88\%$ $max = 58\%$ $sd = 3.35$	$min = 40\%$ $avg = 50.11\%$ $max = 58\%$ $sd = 3.20$	$min = 37\%$ $avg = 49.96\%$ $max = 60\%$ $sd = 3.14$	$min = 35\%$ $avg = 50.00\%$ $max = 62\%$ $sd = 3.16$
384	$min = 46\%$ $avg = 49.32\%$ $max = 53\%$ $sd = 1.96$	$min = 45\%$ $avg = 49.86\%$ $max = 54\%$ $sd = 2.49$	$min = 40\%$ $avg = 50.10\%$ $max = 59\%$ $sd = 2.52$	$min = 40\%$ $avg = 50.04\%$ $max = 59\%$ $sd = 2.60$	$min = 39\%$ $avg = 50.00\%$ $max = 60\%$ $sd = 2.61$
512	$min = 47\%$ $avg = 50.12\%$ $max = 51\%$ $sd = 1.41$	$min = 45\%$ $avg = 50.01\%$ $max = 55\%$ $sd = 2.11$	$min = 42\%$ $avg = 50.04\%$ $max = 58\%$ $sd = 2.35$	$min = 41\%$ $avg = 49.99\%$ $max = 58\%$ $sd = 2.25$	$min = 41\%$ $avg = 50.00\%$ $max = 58\%$ $sd = 2.25$

Table 3.2: Avalanche effect of input message with all zeros

n	8 bits	80 bits	800 bits	8000 bits	80000 bits
224	$min = 49\%$ $avg = 52.68\%$ $max = 56\%$ $sd = 2.27$	$min = 41\%$ $avg = 50.38\%$ $max = 61\%$ $sd = 3.89$	$min = 41\%$ $avg = 50.14\%$ $max = 62\%$ $sd = 3.40$	$min = 37\%$ $avg = 49.99\%$ $max = 61\%$ $sd = 3.38$	$min = 35\%$ $avg = 50.00\%$ $max = 63\%$ $sd = 3.42$
256	$min = 42\%$ $avg = 48.73\%$ $max = 53\%$ $sd = 3.80$	$min = 41\%$ $avg = 50.72\%$ $max = 60\%$ $sd = 3.46$	$min = 41\%$ $avg = 50.06\%$ $max = 58\%$ $sd = 3.14$	$min = 38\%$ $avg = 50.01\%$ $max = 61\%$ $sd = 3.18$	$min = 36\%$ $avg = 50.01\%$ $max = 62\%$ $sd = 3.18$
384	$min = 47\%$ $avg = 50.29\%$ $max = 54\%$ $sd = 2.28$	$min = 43\%$ $avg = 49.95\%$ $max = 54\%$ $sd = 2.38$	$min = 42\%$ $avg = 49.87\%$ $max = 57\%$ $sd = 2.60$	$min = 40\%$ $avg = 49.98\%$ $max = 58\%$ $sd = 2.63$	$min = 39\%$ $avg = 50.00\%$ $max = 59\%$ $sd = 2.61$
512	$min = 49\%$ $avg = 51.20\%$ $max = 53\%$ $sd = 1.28$	$min = 47\%$ $avg = 50.32\%$ $max = 55\%$ $sd = 1.95$	$min = 43\%$ $avg = 50.00\%$ $max = 57\%$ $sd = 2.26$	$min = 41\%$ $avg = 50.05\%$ $max = 58\%$ $sd = 2.25$	$min = 40\%$ $avg = 50.02\%$ $max = 59\%$ $sd = 2.26$

Table 3.3: Avalanche effect of a randomly generated input message

Windows Vista Ultimate 32-bit (x86) Edition. **Compiler**: the ANSI C compiler in the Microsoft Visual Studio 2005 Professional Edition.

b. Speed estimate: Comparison of NaSHA-$(m, 2, 6)$ performance in Cycles/Byte Versus Message on 32-bit architecture, where $m \in \{224, 256, 384, 512\}$ is given in the Table 1.

c. Speed/memory tradeoffs: One way to change NaSHA performances is if as starting bijection we use function of order 2^{16}, instead of 2^8, paying with larger memory of 64KB instead of 0.25KB. In this way we will work with 16-bit words, instead of 8-bit words, increasing the performances by decreasing the number of operations. But searching the bigger Cayley

Length (bytes)	1	10	100	1000	10000	100000
NaSHA–$(224, 2, 6)$	2787.00	270.20	50.24	34.83	33.73	34.53
NaSHA-$(256, 2, 6)$	2797.00	279.70	51.37	37.68	36.43	34.56
NaSHA-$(384, 2, 6)$	5365.00	541.30	53.77	38.47	37.53	35.58
NaSHA-$(512, 2, 6)$	5485.00	548.50	55.21	38.68	37.57	37.16

Table 3.4: Performance in Cycles/Byte Versus Message of NaSHA-$(m, 2, 6)$, where $m \in \{224, 256, 384, 512\}$ on 32-bit architecture

table will decrease performances again. The examination of this option and the possible performance result is an open question. Another problem is the construction of suitable permutation of order 2^{16}.

Also we can speed NaSHA-$(m, k, 6)$, where $m \in \{224, 256, 384, 512\}$, by working with quasigroups of order 2^{128} or 2^{256} ($r = 7$ or $r = 8$). Our opinion is that in that case, the security will be somewhat weakened if the permutation of order 2^8 is used. We think that the same level of security as NaSHA-$(m, k, 6)$ can be obtained for NaSHA-$(m, k, 7)$ if we use a permutation of order 2^{16}.

If instead of $k = 2$ in NaSHA-$(m, k, 6)$, where $m \in \{224, 256, 384, 512\}$, we use $k = 4$, we obtain slowdowns by factor that ranges from 1.75 to 1.9 for NaSHA-$(224, k, 6)$ and NaSHA-$(256, k, 6)$ and from 1.78 to 2 for NaSHA-$(384, k, 6)$ and NaSHA-$(512, k, 6)$.

2.

a. Description of the platform: Wintel personal computer, with an Intel Core 2 Duo Processor, 2.4GHz clock speed, 2GB RAM, running Windows Vista Ultimate 64-bit (x64) Edition. **Compiler**: the ANSI C compiler in the Microsoft Visual Studio 2005 Professional Edition.

Length (bytes)	1	10	100	1000	10000	100000
NaSHA-$(224, 2, 6)$	1718.00	168.10	31.90	24.80	22.30	23.08
NaSHA-$(256, 2, 6)$	1729.00	174.90	32.77	24.94	22.32	23.06
NaSHA-$(384, 2, 6)$	3289.00	330.10	32.65	24.55	24.04	24.52
NaSHA-$(512, 2, 6)$	3361.00	336.10	36.25	24.64	24.04	24.55

Table 3.5: NaSHA Performance in Cycles/Byte Versus Message Length on 64-bit architecture

b. Speed estimate: Comparison of NaSHA-$(m, 2, 6)$ performance in Cycles/Byte Versus Message on 64-bit architecture, where $m \in \{224, 256, 384, 512\}$ is given in the Table 2.

c. Speed/memory tradeoffs: If instead of $k = 2$ in NaSHA-$(m, k, 6)$, where $m \in \{224, 256, 384, 512\}$, we use $k = 4$, we obtain slowdowns by factor almost 2.

Preliminary security analysis

NaSHA family of cryptographic hash function uses Merkle-Damgård domain extender with standard Merkle-Damgård strengthening. It has incorporated also the wide-pipe design of Lucks [70, 71] and Coron's [11] suggestions. In every iterative step of the compression function, we use $2n$-bit message blocks and $2n$-bit chaining variable, so the strings of length $4n$ bits are mapped to strings of length $4n$ bits and then only $2n$ bits are kept for the next iterative step. And, the most important, *the length of any chaining variable is at least two times wider than the final digest value.* For the same reasons D. Gligorovski [37] stated, by this kind of design we gain resistance to some generic attacks like: length extension attack, Joux's multicollision attack [54], length extension attack, Dean fixed point attack [18], Kelsey and Schneier's long message 2^{nd} preimage attack [57], Kelsey and Kohno's herding attack [56] and 2^{nd} collision attack.

Resistance to preimage and 2^{nd} preimage attacks

The quasigroup used for NaSHA-$(256, 2, 6)$ is of order $2^n = 2^{64}$ and $\mathcal{MT}$ is performed on $t = 16$ 64-bit words, so by Proposition 25 one can find a second preimage or collision after around 2^{68} checks, but *under condition that the quasigroup operations and the values of the leaders are known by the attacker.* The quasigroup operations $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_1,\beta_1,\gamma_1,A_1,B_1,C_1}$ and $*_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_2,\beta_2,\gamma_2,A_2,B_2,C_2}$ of NaSHA-$(256, 2, 6)$ hash function and the leaders l_1, l_2 depend on the input values of $\mathcal{MT}$.

Let $\mathcal{MT}(x_1||x_2||x_3||\dots||x_{16}) = (d_1, d_2, \dots, d_{16})$, where x_i are 64-bit unknowns and d_i are given 64-bit words. Let $\mathcal{A}_{l_2}(x_1||x_2||x_3||\dots||x_{16}) = z_1||z_2||z_3||\dots||z_{16}$ and put $y_i = \rho(z_i, 32)$ for $i = 1, 2, \dots, 16$. Then we obtain the following system of equations with unknowns x_i and y_i (i.e., z_i), unknown quasigroup operations $\bullet = *_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_1,\beta_1,\gamma_1,A_1,B_1,C_1}$ and

$\star = *_{a_1,b_1,c_1,a_2,b_2,c_2,a_3,b_3,c_3,\alpha_2,\beta_2,\gamma_2,A_2,B_2,C_2}$, and unknown leaders l_1 and l_2:

$$\begin{cases} (l_2 + x_1) \bullet x_1 = y_1 \\ (y_1 + x_2) \bullet x_2 = y_2 \\ \dots \\ (y_{15} + x_{16}) \bullet x_{16} = y_{16} \\ y_{16} \star (y_{16} + l_1) = d_{16} \\ y_{15} \star (y_{15} + d_{16}) = d_{15} \\ \dots \\ y_1 \star \ (y_1 + d_2) = d_1. \end{cases} \tag{3.6}$$

For solving the system (3.6) we need at first to define the quasigroup operation $\bullet$ and $\star$ and the leaders l_1 and l_2. So, we have to choose 8 bytes $a_1, b_1, c_1, a_2, b_2, c_2, a_3, b_3$ (note that $c_3 = a_1$), 6 16-bit words $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2$, 6 32-bit words $A_1, B_1, C_1, A_2, B_2, C_2$ and 2 64-bit words l_1, l_2, and that can be done in 2^{480} ways. Fix a choice of all of the constant words and then, after around 2^{68} checks, a solution $x_1, x_2, x_3, \dots, x_{16}$ of (3.6) can be found. Now, we have to see if the obtained solution satisfies the equalities

$$x_1 \oplus x_2 = l_1, \tag{3.7}$$

$$x_3 \oplus x_4 = l_2, \tag{3.8}$$

$$x_5 \oplus x_6 = a_1||b_1||c_1||a_2||b_2||c_2||a_3||b_3, \tag{3.9}$$

$$x_7 + x_8 = \alpha_1||\beta_1||\gamma_1||\alpha_2, \tag{3.10}$$

$$(x_9 + x_{10})(\texttt{mod } 2^{32}) = \beta_2||\gamma_2, \tag{3.11}$$

$$x_{11} + x_{12} = A_1||B_1, \tag{3.12}$$

$$x_{13} + x_{14} = C_1||A_2, \tag{3.13}$$

$$x_{15} + x_{16} = B_2||C_2. \tag{3.14}$$

For each of the equalities (3.7)–(3.10), (3.12)–(3.14), we have that the probability to be true is 2^{-64}, so these seven equalities will be true with probability 2^{-448}. The equality (3.11) will be true with a probability 2^{32}. So, all the equalities (3.7)–(3.14) will be true with probability 2^{-480}. (Namely, there are $(2^{64})^2$ pairs (x_1, x_2), and there are 2^{64} different solutions of (3.7) when (3.7) is considered as an equation with 2 unknowns x_5, x_6. The same discussion holds for the others equalities as well.)

So, after having around 2^{68} checks, we can find a solution of (3.6) with a probability 2^{-480}. The space of all possible values of $(a_1, b_1, c_1, a_2, b_2, c_2, a_3,$

$b_3, \alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, A_1, B_1, C_1, A_2, B_2, C_2, l_1, l_2)$ consists of 2^{480} elements. Then, after making $2^{68} \cdot 2^{480} = 2^{548}$ checks, a solution of (3.6) can be found with probability $1-(1-2^{-480})^{2^{480}} \approx 0.53$.

We conclude that NaSHA-$(256, 2, 6)$ is 2^{nd} preimage resistant. Consequently, it is preimage resistant with much higher complexity, since in this cases only d_4, d_8, d_{12} and d_{16} are known (the hash value of NaSHA hash is $d_4||d_8||d_{12}||d_{16}$). To discover the original image one has to choose $d_1, d_2, d_3, d_5, d_6, d_7, d_9, d_{10}, d_{11}, d_{13}, d_{14}, d_{15}$ in such a way the true values of $y_1, \dots, y_{16}$ of (3.6) have to be find, and that can be done with probability around $(2^{-64})^{12}$.

The analysis given above for NaSHA-$(256, 2, 6)$ holds true for NaSHA-$(224, 2, 6)$ too. The same analysis holds true for NaSHA-$(384, 2, 6)$ and NaSHA-$(512, 2, 6)$. In this case, a slightly better results are obtained since the value of t is 32.

Collision resistance

For the collision resistance we have to find $(x_1, \dots, x_{16}) \neq (x'_1, \dots, x'_{16})$ such that $\mathcal{MT}(x_1, x_2, \dots, x_{16}) = \mathcal{MT}(x'_1, x'_2, \dots, x'_{16})$. We infer equations of kind

$$\begin{cases} (l_2 + x_1) \bullet x_1 = y_1 \\ (y_1 + x_2) \bullet x_2 = y_2 \\ \dots \\ (y_{15} + x_{16}) \bullet x_{16} = y_{16} \end{cases} \tag{3.15}$$

$$\begin{cases} (l'_2 + x'_1) \bullet' x'_1 = y'_1 \\ (y'_1 + x'_2) \bullet' x'_2 = y'_2 \\ \dots \\ (y'_{15} + x'_{16}) \bullet' x'_{16} = y'_{16} \end{cases} \tag{3.16}$$

$$\begin{cases} y_{16} \bullet (y_{16} + l_1) = y'_{16} \bullet' (y'_{16} + l'_1) \\ y_{15} \bullet (y_{15} + (y_{16} \bullet (y_{16} + l_1))) = y'_{15} \bullet' (y'_{15} + (y'_{16} \bullet' (y'_{16} + l'_1))) \\ \dots \\ y_1 \bullet (y_1 + (y_2 \bullet \dots) \dots) = y'_1 \bullet' (y'_1 + (y'_2 \bullet' \dots) \dots). \end{cases} \tag{3.17}$$

Now, besides the equalities (3.7)–(3.14), we will have eight more

$$x'_1 \oplus x'_2 = l'_1, \tag{3.18}$$

$$x'_3 \oplus x'_4 = l'_2, \tag{3.19}$$

$$x'_5 \oplus x'_6 = a'_1||b'_1||c'_1||a'_2||b'_2||c'_2||a'_3||b'_3, \tag{3.20}$$

$$x'_7 + x'_8 = \alpha'_1||\beta'_1||\gamma'_1||\alpha'_2, \tag{3.21}$$

$$(x'_9 + x'_{10})(\texttt{mod } 2^{32}) = \beta'_2||\gamma'_2, \tag{3.22}$$

$$x'_{11} + x'_{12} = A'_1||B'_1, \tag{3.23}$$

$$x'_{13} + x'_{14} = C'_1||A'_2, \tag{3.24}$$

$$x'_{15} + x'_{16} = B'_2||C'_2. \tag{3.25}$$

Then, even we assume that we have a solution of the system of equations (3.17), after 2^{1028} checks we can find a solution of (3.15) and (3.16) with probability ≈ 0.5. So we have the following statement. Similar can be proved for NaSHA-$(384, 2, 6)$ and NaSHA-$(512, 2, 6)$.

Resistance to attacks that get all the additions to behave as XORs

Compression function of NaSHA-(m, k, r) use additions modulo 2^{32} and 2^{64}, XORs and left rotations, so we must to examine attacks that find values for which additions in NaSHA-(m, k, r) behave as XORs. It is important to mention the work of Lipmaa and Moriai [66], which constructed efficient algorithms for computing differential properties of addition modulo 2^n, work of Lipmaa et al [67], which constructed linear-time algorithm for computing the additive differential probability of XOR, and work of Paul and Preneel [113].

NaSHA-(m, k, r) is resistant to these kind of attacks, because it is using extended Feistel networks, which incorporate operations with 8, 16, 32 and 64-bits operations and table lookups, instead of using only combinations of 32 or 64-bits words. Additionally, having in mind that compression function of NaSHA-(m, k, r) is function from $\{0, 1\}^{4n}$ to $\{0, 1\}^{4n}$, at this moment it is impossible to find concrete values of arguments for this function, for which additions will behave as XORs.

Resistance to linear and differential attacks

Recent collision attacks on some hash functions [139, 137, 136, 138] are in fact differential attacks that involves modular integer subtraction or exclusive-or as a measure of difference and some kind of message modification techniques. There are several strategies which one might employ to attempt to prevent the success of these attacks. The first one is to attempt to prevent the existence of any "good" differential (a differential path that leads to (near) collisions and holds with probability greater than $2^{-n/2}$), like wide trail strategy for block ciphers. The second strategy would be to reduce the success probability of the attack with restraining the power of the message modification techniques. A third possibility is to consider situations in which single message bits are going to affect multiple blocks or maybe entire hash.

The NaSHA-(m,k,r) hash algorithm allows each bit of an input message M to influence almost all bits of the resulting hash value. To verify this let represent $S^{(i)}$ as

$$S^{(i)} = S_1^{(i)}||S_2^{(i)}||S_3^{(i)}||\dots||S_{2t-2}^{(i)}||S_{2t-1}^{(i)}||S_{2t}^{(i)}.$$

We have that every bit from the bit string $S^{(i)}$ influences all blocks $S_j^{(i+1)}$ with even subindexes ($j = 2, 4, 6, \dots, 2t$) of the bit string $S^{(i+1)}$. Namely, by Step 6 we apply the transformations $LinTr_{2^{n+2}}^{2t}$ and $\mathcal{MT}$ on $S^{(i)}$. The linear transformation besides diffusion spread out the influence of the bits. The $\mathcal{MT}$ transformation is composition of $\mathcal{A}_l$ and $\rho(\mathcal{RA}_l)$ transformations. Now, if b is a bit from a block $S_j^{(i)}$ of $S^{(i)}$, then all blocks of $\mathcal{A}_l(S^{(i)})$ from the $j+1$-th until $2t$-th are influenced by b. After that, all blocks of $\mathcal{MT}(\mathcal{A}_l(S^{(i)}))$ will be influenced by b. So we have the following theorem.

Theorem 22 *Every bit from the input message M influences all blocks of the hash value NaSHA-$(m,k,r)(M)$.* □

Proof By the above mentioned considerations we have that each bit of M influences all blocks with even subindexes of $S^{(N)}$. Since NaSHA-$(m,k,r)(M) = A_4||A_8||\dots||A_{2t-4}||A_{2t}$, where $A_1||A_2||A_3||\dots||A_{2t} = (LinTr_{2^{n+2}}^{2t}(S^{(N)}))$, all blocks of NaSHA-$(m,k,r)(M)$ are influenced by each bit of M. ■

Much more than Theorem 22 is stating, the internal structure of the quasigroup operation and the addition modulo 2^r allows us to conclude that almost all bits of the hash value are influenced by each bit of the input message.

Also we have to stress out that our starting bijection has also good resistance to differential attacks with differential 4-uniformity and good resistance to linear attack with nonlinearity of 112. All these together give a good resistance to any attack that will involve differential cryptanalysis.

Nonlinearity of 112 of starting function is inherited in constructed extended Feistel network in our implementation. From all this, we gain resistance of NaSHA-(m,k,r) to any attack that will involve linear cryptanalysis, but also we gain resistance to recent Cube attack of Dinur and Shamir [26], that can be applied to wide rang of cryptographic primitives which are provided as a black box (even when nothing is known about its internal structure) as long as at least one output bit can be represented by (an unknown) polynomial of relatively low degree in the secret and public variables.

Cryptanalysis

For the early version of our implementation of NaSHA-$(m, 2, 6)$ where definition of leaders and constants were same for all m, there have been some cryptanalysis for NaSHA-$(384, 2, 6)$ and NaSHA-$(512, 2, 6)$.

J. Li et al [51] have been given free start collisions for all versions of NaSHA with examples and really interesting truncated differential collision attack on NaSHA-512 with claimed complexity 2^{192} of the attack. They made a very interesting observation: that when a and x satisfy the conditions $(a)_{64...32} = \neg(x)_{64...32}$, $(a)_{32} = 1$ and $(a)_{31...1} = 0$, the input difference $\triangle x = 0x00000000FFFFFFFF$ always lead to the zero output difference for the calculation of $(a + x) * x$ ($(x)_i$ denotes the i-th bit of x). For example, given $x = 0xAAAAAAAA00000000$, $x_0 = 0xAAAAAAAAFFFFFFFF$ and $a = 0x5555555580000000$, $(a+x)*x = (a+x_0)*x_0$ always holds no matter what parameters are set for the quasigroup operation $*$. S. Markovski et al [89] confirmed that this attack has unknown probability, because attackers use a system of three quasigroup equations with five variables. Their claim will be true if this kind of systems always has a solution. But this is not true. There are examples of these kind of systems with no solutions for quasigroups of order 4.

I. Nikolić and D. Khovratovich [106] have been given free-start collision attacks on NaSHA with complexity of 2^{32} and free-start preimage attack on NaSHA-n with complexity of $2^{n/2}$. With recent changes only this attack will work and this will be only for 256 and 224 version of NaSHA.

3.2 Pseudo-random number generators

A *pseudo-random number generator (PRNG)* is an deterministic algorithm for generating a pseudo-random sequence of numbers that approximates the properties of random numbers. They are necessary in cryptography, stochastic simulations, search heuristics, game playing etc, for generation of keys, nonces, challenges etc. Pseudo-randomness comes from the fact that the sequence is completely determined by a relatively small set of initial values, called the PRNG's state, which is initialize by random seed. Random seeds are often generated from the state of the computer system (such as the time), a cryptographically secure pseudo-random number generator (CSPRNG) or from a hardware random number generator. PRNGs need to have very long periods and simple and fast software implementation. Common classes of these algorithms are the linear congruence functions and the linear feedback shift registers, which have relatively small periods and are highly predictable. Some newer PRNGs are Blum Blum Shub, Fortuna, and the Mersenne twister. PRNG can be made also from other cryptographic primitives as stream and block ciphers and hash functions. PRNG need to satisfy some requirements:

- PRNG needs to pass many statistical randomness tests.
- Produced pseudo-random sequences do not contain identical consecutive elements with a high probability.
- It should be impossible for any attacker to calculate, or otherwise guess, from any given sub-sequence, any previous or future values in the sequence, nor any inner state of the generator.
- It should be impossible, for an attacker to calculate, or guess from an inner state of the generator, any previous numbers in the sequence or any previous inner generator states.

Only PRNGs that meet the last requirement can apply in cryptography for key generation, generation of nonces, salts etc.

Dimitrova and Markovski [25] propose one quasigroup based PRNG - QPRSG with arbitrary large period and give analysis of which quasigroups are appropriate to use in PRNGs. Let $(Q, *)$ be a quasigroup and then choose $a \in Q$, so $a * a \neq a$. Then they apply k times the transformation E_a on string $aaa\ldots$ or

$$E_a^{(k)}(aaa\ldots) = a_1^{(k)} a_2^{(k)} a_3^{(k)} \ldots$$

The Theorem 4 provides that with the increasing of the k will also increase the period of QPRSG. Also obtained sequences pass all statistical randomness tests. From the numerical experiments made by authors, over 50% of the quasigroups have coefficient of period growth greater than half of their order. Fraction of quasigroup with almost ideal period growth is very small, but real. The QPRSG is CSPRNG if the quasigroup that was used to build the generator remains unknown. The QPRSG further can be make faster with parallelization and can be improved by using random starting sequence $b_1 b_2 b_3 \ldots$ instead of $aaa\ldots$ [73].

Markovski et al. [80] have been proposed a new method for simulating unbiased physical sources of randomness and improving the properties of existing PRNGs, which is based on the quasigroup string transformations. This method is flexible, highly parallel, with linear complexity and is capable of producing a random number sequence from a very biased stationary source. In fact, it comes in two variants based on $E-$ and $E'-$ quasigroup string transformations, represented below. The input of each algorithm consists of choosing quasigroup $(Q,*)$ of order s, fixed element l from Q as a leader, an integer k as the number of applied transformations and biased random string $b_0b_1b_2\dots b_j$.

E-algorithm	*E'-algorithm*
1. For $i = 1$ to k do $L_i \leftarrow l$;	1. For $i = 1$ to k do $L_i \leftarrow l$;
2. $j \leftarrow 0$;	2. $j \leftarrow 0$;
3. do	3. do
$b \leftarrow b_j$;	$b \leftarrow b_j$;
$L_1 \leftarrow L_1 * b$;	$L_1 \leftarrow b * L_1$;
For $i = 2$ to k do	For $i = 2$ to k do
$L_i \leftarrow L_i * L_{i-1}$;	$L_i \leftarrow L_{i-1} * L_i$;
Output: L_k;	Output: L_k;
$j \leftarrow j + 1$;	$j \leftarrow j + 1$;
loop;	loop;

The authors also made some recommendations about method's parameters. For simulating unbiased physical sources of randomness, order s can be arbitrary large and $4 \leqslant s \leqslant 256$. The number k should be chosen by the rule "for smaller s larger k" and its choice depends on the source. For highly biased sources recommendation are $ks \geqslant 512$ and $k > 8$. For improving the properties of existing PRNGs, the chosen quasigroup must be exponential.

3.3 Stream ciphers

Stream cipher is a symmetric key algorithm, which encrypt plaintext bits, usually individual bytes (or bit), one at a time, using an encryption transformation which varies with time. So, it gives different output for the same sequence of plaintext. Stream ciphers typically are faster than block ciphers and have lower hardware complexity. Because stream ciphers have limited or no error propagation, stream ciphers may be advantageous in situations where transmission errors are highly probable. They are mandatory when buffering is limited or when characters must be individually processed as they are received.

Usually stream ciphers generate the so called keystream which is than combined with plaintext stream by some combiner-type algorithms, which in most cases is simple bitwise xoring operation (*binary additive stream cipher*). Stream ciphers can be divided to synchronous and self-synchronous or asynchronous. *Synchronous stream ciphers* generate the keystream independently of the plaintext and ciphertext. They are having no error-propagation, which limits the opportunity to detect an error when decryption is performed, but more importantly an attacker is able to make controlled changes to parts of the ciphertext knowing induced changes on the corresponding plaintext. Also the sender and the receiver must be exactly in step for decryption to be successful and for that aim, restoration of synchronization is needed, usually by including "marker positions" in the transmission. Errors in the transmission results in incorrect decryption until one of the marker positions is received.

Self-synchronous or *asynchronous stream ciphers* use n bits of ciphertext to generate the keystream so it has limited error propagation - the one-bit error may produce incorrect decryption of the following n bits. They have ability to resume correct decryption if the decrypting keystream falls out of synchronization with the encrypting keystream. Big drawback of these ciphers is that the attacker knows some of the variables being used as input to the algorithm.

Some known stream ciphers are RC4, PANAMA, SEAL, Trivium etc. Stream ciphers must have long periods, must not produce related or weak keys and it must be impossible to recover the cipher's key or internal state from the keystream. Produced keystreams must not allow to attackers to distinguish them from random noise.

One of the earliest quasigroup based encryption method is given in [123], where a set of $\{L_1, \ldots, L_k\}$ MOLS of order n is used. The secret key is pair of different squares (L_c, L_d) and if the message is encoded as a pair (i, j), it can be encrypted in the pair (α, β), that occur at the intersection of row i and column j of the Latin squares L_c and L_d. Decryption is done by simple scanning of L_c and L_d and because of the orthogonality, unique pair of coordinates (i, j) will be obtained.

Another early attempt to use quasigroups for constructing stream cipher, which is synchronous, is made by Kościelny [60]. For that aim, he suggests to use quasigroup $(Q, \circ)$ obtained by isotopies from group, isomorphic to the additive group of $GF(q)$ or cyclic group of order q or Abelian loop of even order q (in [61] you can find several Maple 7 routines for generating quasigroups isomorphic to the interior of: cyclic group of order q, multiplicative group and additive group of a finite field $GF(p^m)$ and their isotopies). The quasigroup can also be represented as vector valued Boolean functions. For creating the stream cipher, he also uses the two conjugates of the given quasigroup, $(Q, \backslash)$ and $(Q, /)$. Let $m_1 m_2 m_3 \dots$ denote the stream of characters of the plaintext, $c_1 c_2 c_3 \dots$ denote the stream of characters of the ciphertext and $k_1 k_2 k_3 \dots$ denote the keystream. The author suggests 6 ways for enciphering and deciphering:

$$c_i = m_i \circ k_i, \quad m_i = c_i / k_i$$

$$c_i = k_i \circ m_i, \quad m_i = k_i \backslash c_i$$
$$c_i = k_i / m_i, \quad m_i = c_i \backslash k_i$$
$$c_i = m_i / k_i, \quad m_i = c_i \circ k_i$$
$$c_i = m_i \backslash k_i, \quad m_i = k_i / c_i$$
$$c_i = k_i \backslash m_i, \quad m_i = k_i \circ c_i$$

If $(Q, \circ)$ is not associative, than m_i can be mapped into 6 different characters, which is a progress in comparison with the stream ciphers built over $GF(2)$ with XOR operation. The secret key may have five components: the sequence of characters interacting with the stream of the characters of a plaintext, the quasigroup $(Q, \circ)$ and three permutations needed to form the conjugate quasigroups. One security argument is that the set of all isotopies of a quasigroup of order q forms a group of order $(q!)^3$.

Other early attempt to create quasigroup based asynchronous stream cipher is given in Markovski et al. [76]. Let $\circ$ be a quasigroup operation defined on alphabet Q and let $\backslash$ be its left division. The cipher stream is obtained by simple $e-$ transformation with fixed leader and decrypting is done by $d-$ transformation with the same leader. The secret key is the used quasigroup. Several years letter, Ochodkova and Snasel [109] use exactly the same method for encoding the file.

In Markovski et al [82] is given quasigroup based enciphering method, where encryption is done by $T^{(n)}_{l_n,\dots,l_1}-$ transformation and decryption with opposite transformation. This is in fact asynchronous stream cipher. The secret key are leaders $l_1 \dots l_n$ and the order of $e-$ or $d-$ transformations in encryption transformation, but quasigroup is publicly known. Interesting is that the authors implemented this in the so called Ytalk 3.0.2 software for on-line chat over Internet and for that aim they used quasigroup of order 128 with alphabet first 128 characters of ASCII table.

In [114] Petrescu gives an enciphering method using ternary quasigroups, which can be used as a asynchronous stream cipher. Let (Q, α) be publicly known quasigroup which will be used as a seed and as an isotope carrier. Every ternary quasigroup (Q, α) forms an algebra $(Q, \alpha, \alpha_1, \alpha_2, \alpha_3)$ with 4 ternary operation, satisfying the following identities

$$\begin{array}{ll} \alpha(\alpha_1(x_1), x_2, x_3) = x_1, & \alpha_1(\alpha(x_1), x_2, x_3) = x_1 \\ \alpha(x_1, \alpha_2(x_2), x_3) = x_2, & \alpha_2(x_1, \alpha(x_2), x_3) = x_2 \\ \alpha(x_1, x_2, \alpha_3(x_3)) = x_3, & \alpha_3(x_1, x_2, \alpha(x_3)) = x_3 \end{array}$$

Let $\mathcal{K} = Q^4 \times \{1, 2, 3\}$ be the key space. The key is represented as $k = a_1a_2a_3a_4i$ and it determines another isotopic quasigroup (Q, β) by

$$\beta(x_1, x_2, x_3) = f_4(\alpha(f_1^{-1}(x_1), f_2^{-1}(x_2), f_3^{-1}(x_3))),$$

where $f_j = f_{a_j}$ are permutations on Q. Every key uniquely determines a bijection $E_k(m_1m_2\ldots) = c_1c_2\ldots$, given below

$i = 1$	$i = 2$	$i = 3$
$c_1 = \beta(m_1, a_1, a_2)$ $c_2 = \beta(m_2, a_3, a_4)$ $j > 2$ $c_j = \beta(m_j, c_{j-2}, c_{j-1})$	$c_1 = \beta(a_1, m_1, a_2)$ $c_2 = \beta(a_3, m_2, a_4)$ $j > 2$ $c_j = \beta(c_{j-2}, m_j, c_{j-1})$	$c_1 = \beta(a_1, a_2, m_1)$ $c_2 = \beta(a_3, a_4, m_2)$ $j > 2$ $c_j = \beta(c_{j-2}, c_{j-1}, m_j)$

Decryption function $E_k(c_1c_2\ldots) = m_1m_2\ldots$ is defined as

$i = 1$	$i = 2$	$i = 3$
$m_1 = \beta_1(c_1, a_1, a_2)$ $m_2 = \beta_1(c_2, a_3, a_4)$ $j > 2$ $m_j = \beta_1(c_j, c_{j-2}, c_{j-1})$	$m_1 = \beta_2(a_1, c_1, a_2)$ $m_2 = \beta_2(a_3, c_2, a_4)$ $j > 2$ $m_j = \beta_2(c_{j-2}, c_j, c_{j-1})$	$m_1 = \beta_3(a_1, a_2, c_1)$ $m_2 = \beta_3(a_3, a_4, c_2)$ $j > 2$ $m_j = \beta_3(c_{j-2}, c_{j-1}, c_j)$

The author gave an assemble implementation also, where the seed quasigroup is defined as $\alpha(x_1, x_2, x_3) = (x_1 - x_2 + x_3) \bmod 256$ and $f_a(x) = x + a \bmod 256$.

Maybe the most famous quasigroup based stream cipher, which has been intrigue the cryptography community for a several years is Edon80, designed by Gligoroski et al. [42, 44]. It is one of the few left unbroken eSTREAM finalists. Especially interesting about this cryptographic primitive is that it uses 4 quasigroups of very small order, 4 actually, and it is still resisting to all attacks. The authors claimed that 64 out of 576 quasigroups of order 4 are very suitable for using in Edon80, and they have chose the quasigroups with the lexicographic order 61, 241, 350 and 564. Only quasigroup 350 is shapeless, 61 is commutative, 241 has left unit 1 and satisfy the identity $x \bullet_1 (x \bullet_1 (x \bullet_1 (x \bullet_1 y))) = y$ and 564 satisfy the identity $x \bullet_3 (x \bullet_3 y) = y$. Other algebraic properties of these quasigroups are examined in [135].

$\bullet_0$	0	1	2	3	$\bullet_1$	0	1	2	3	$\bullet_2$	0	1	2	3	$\bullet_3$	0	1	2	3
0	0	2	1	3	0	1	3	0	2	0	2	1	0	3	0	3	2	1	0
1	2	1	3	0	1	0	1	2	3	1	1	2	3	0	1	1	0	3	2
2	1	3	0	2	2	2	0	3	1	2	3	0	2	1	2	0	3	2	1
3	3	0	2	1	3	3	2	1	0	3	0	3	1	2	3	2	1	0	3

Table 3.6: Quasigroups used in Edon80

Edon80 is a binary additive stream cipher, with average period of 2^{91} and with three modes of operation: KeySetup, IVSetup and Keystream mode. First two modes serve for initialization of the key and the initial vector IV. The secret key is 80 bits long, and it is divided in 40 2-bits values, each of them selects one of four quasigroup operations. Obtained IV is consists also from 40 2-bits values $v_0 v_1 \dots v_{31} 32100123$ and it has the initial values of the internal states $a_0 \dots, a_{79}$. Encryption is done in Keystream mode and it starts with periodic string that has shape: $01230123 \dots 0123 \dots$. Encryption consists from 80 $e-$transformations, with initialized internal states $a_0 \dots, a_{79}$ as a leaders. The output of the stream cipher is every second value of the last $e-$transformation. In [38] one can find a proposal of adding MAC functionality to Edon80. A related key attack on Edon80 is suggested by Hell et al [49], with the complexity of 2^{69}, although this complexity has been disputed by the Edon80 authors.

Another eSTREAM unbroken phase 3 candidate that uses quasigroups is CryptMT v3 (Cryptographic Mersenne Twister), designed by Matsumoto et al [92, 93]. It is a binary additive stream cipher over the set $B = \mathbb{F}_2^8$, with period multiply of $2^{19937} - 1$. It uses combined generator, consisting of two parts. The first part is so called SFMT (SIMD-oriented Fast Mersenne Twister) generator, which generate 128-bit pseudo-number integer in one step, and the second part is an uniform quasigroup filter with memory of one wordsize. We are interesting in used quasigroup. Let Q be the ring $\mathbb{Z}/2^{32}$ of integers modulo 2^{32} and every $x \in Q$ corresponds to a 33-bit odd integer $2x + 1$ mod 2^{33}. Quasigroup operation $\circ$ is defined as

$$x \circ y = 2xy + x + y \ mod \ 2^{32}$$

which is essentially the multiplication of 33−bit odd integers. This quaisgroup definitely is too far from being shapeless. From the definition one can see that this quasigroup is associative, commutative, with unit 0 and has several proper quasigroups (Table 3.7).

$\circ$	0	$2^{31}-1$
0	0	$2^{31}-1$
2^{31-1}	$2^{31}-1$	0

$\circ$	0	$2^{32}-1$
0	0	$2^{32}-1$
2^{32-1}	$2^{32}-1$	0

$\circ$	0	2^{31}
0	0	2^{31}
2^{31}	2^{31}	0

$\circ$	0	$2^{31}-1$	2^{31}	$2^{32}-1$
0	0	$2^{31}-1$	2^{31}	$2^{32}-1$
$2^{31}-1$	$2^{31}-1$	0	$2^{32}-1$	2^{31}
2^{31}	2^{31}	$2^{32}-1$	0	$2^{31}-1$
$2^{32}-1$	$2^{32}-1$	2^{31}	$2^{31}-1$	0

Table 3.7: Some proper quasigroups used in CryptMT quasigroup

3.4 Block ciphers

Block cipher is a symmetric key algorithm, which encrypts plaintext in fixed-length groups of bits, termed blocks, with an unvarying transformation. Conventional block ciphers take two inputs: a key $K \in \{0,1\}^k$ and a plaintext $M \in \{0,1\}^n$ and produce a single output - a ciphertext $C \in \{0,1\}^n$. This can be represented as $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ or $E_K : \{0,1\}^n \to \{0,1\}^n$, when the key is fixed. Each key selects one bijection $E_K(\cdot)$ from the possible set of $(2^n)!$. Decryption can be done by inverse transformation E_K^{-1}, or for the message M, we have $E_K^{-1}(E_K(M)) = M$. There exist also "tweakable" block cyphers, which accept additional input called the tweak $T \in \{0,1\}^t$. The tweak, along with the key, selects the permutation computed by the cipher ($E_{K,T}(\cdot)$ is bijection). Role of the tweak is to provide variability, unlike the key which provides uncertainty to the adversary.

For encrypting messages larger than size of the block, a mode of operation and some padding rule are used. NIST [110] recommends the following modes of operation for use with an underlying symmetric key block cipher algorithm: *electronic codebook (ECB), cipher-block chaining (CBC), cipher feedback (CFB), output feedback (OFB)* and *counter (CTR)* mode. Some modes of operation, like OFB mode and CTR mode turn a block cipher to work as a stream cipher and for them, the plaintext does not need to be a multiple of the block size. For the ECB and CBC modes, the total number of bits in the plaintext must be a multiple of the block size and for CFB mode, the total number of bits in the plaintext must be a multiple of a parameter,that does not exceed the block size. CBC and CFB mode start with initialization vector IV, for which it is important to be unpredictable, and for OFB mode, IV can be nonce.

There is one interesting application of quasigroups, made by Gligoroski [45], who proved that CBC and OFB modes can be represented as quasigroup string transformations, and that OFB is special case of CBC mode of operation where the encryption of a string of all zeroes is performed. This implies that one can launch several attack scenarios against that interchanged use of CBC and OFB modes of operation.

Most block ciphers are constructed by repeatedly applying a simpler function, termed the round function. This approach is known as iterated block cipher where each iteration is termed a round. Most famous block ciphers are DES, IDEA, AES etc. Currently, there are 3 approved block ciphers: AES, Triple DES and Skipjack (EES).

One attempt to deploy quasigroups for block cipher is given by Carter et al [10]. They introduce DESV - a version of DES in which XOR is replaced by an arbitrary quasigroup operation defined by a Latin square, and also they claim that reduced numbers of rounds can be safely contemplated.

3.4.1 Block cipher Alex'smile-(B, I, G)

Here we define the family of tweakable block ciphers Alex'smile-(B, I, G) that works on 32-bit words. The parameters B, I, G denote block size in 32-bit words (even number), number of rounds and the length of the key size in bits (multiple of 32). Each round consists of two OT transformations going in different direction through the string, followed by fixed left rotations. Before the first round and after the last round there is classical whitening with XOR, which is applied in several designs, e. g. Khufu/Khafre, DES-X, Twofish, AES etc. Motivation for this is that every operations before the first and after the last key manipulation does not contribute to the security of the given cipher.

Generally, any Alex'smile-(B, I, G) block cipher transforms a plaintext message block of length B words into a ciphertext block of same length by using a secret key k of length G bits. It use one additional input, the tweak T of length 128 bits, which purpose is only to provide variability. Special algorithm for key expanding and key scheduling is used. It uses the secret key k and the tweak T as an input. Expanded key consists of $2B + 8I$ subkeys words, denoted by $K = K_1K_2 \dots K_{2B+8I}$, where $K_i, \in \mathbb{Z}_2^{32}$.

Given a plaintext $M = m_1m_2 \dots m_B \in (\mathbb{Z}_2^{32})^B$ we obtain the ciphertext $C = c_1c_2 \dots \dots c_B \in (\mathbb{Z}_2^{32})^B$ by using the following encryption algorithm.

```
Encryption algorithm of Alex'smile-$(B, I, G)$
Input: A plaintext $M = (m_1, \ldots, m_B)$, an expanded key $K = (K_1, \ldots, K_{2B+8I})$,
       fixed constants for left rotations $(l_1, \ldots, l_{2B-4})$ and
       constants $(RC_1, \ldots, RC_B)$ as 32-bit words.
Output: A ciphertext $C = (c_1, \ldots, c_B)$.
 1. for $i = 1$ to $B$ do $b_i \leftarrow (K_i \oplus m_i) + RC_i$;
 2. for $j = 1$ to $I$ do
            $*_1 = *_{K_{B+(j-1)I+1} \cdots K_{B+(j-1)I+4}}$ and $\circ_1 = \circ_{K_{B+(j-1)I+1} \cdots K_{B+(j-1)I+4}}$
            $OT_{*_1,\circ_1}(b_1, b_2 \ldots b_B) = a_1, a_2 \ldots a_B$
            $(lr_1, lr_2, lr_3, \ldots, lr_B) =$
                $(lr_1(K_{B+(j-1)I+3}), lr_2(K_{B+(j-1)I+3}), l_1, \ldots l_{B-2})$;
            for $i = 1$ to $B$ do $b_i \lll_{lr_i}$;
            $*_2 = *_{K_{B+(j-1)I+5} \cdots K_{B+(j-1)I+8}}$ and $\circ_2 = \circ_{K_{B+(j-1)I+5} \cdots K_{B+(j-1)I+8}}$
            $OT_{*_2,\circ_2}(a_B, a_{B-1} \ldots a_1) = b_B, b_{B-1} \ldots b_1$
            $(lr_{B+1}, lr_{B+2}, lr_{B+3}, \ldots, lr_{2B}) =$
                $(lr_{B+1}(K_{B+(j-1)I+7}), lr_{B+2}(K_{B+(j-1)I+7}), l_{B-1}, \ldots l_{2B-4})$;
            for $i = 1$ to $B$ do $b_i \lll_{lr_{B+i}}$;
 3. For $i = 1$ to $B$ do $c_i = K_{B+8I+i} \oplus b_i$
```

Decryption is done by the following algorithm.

```
Decryption algorithm of Alex'smile-$(B, I, G)$
Input: A ciphertext $C = (c_1, \ldots, c_B)$, an expanded key $K = (K_1, \ldots, K_{2B+8I})$,
       fixed constants for left rotations $(l_1, \ldots, l_{2B-4})$ and
       constants $(RC_1, \ldots, RC_B)$ as 32-bit words.
Output: A plaintext $M = (m_1, \ldots, m_B)$.
 1. For $i = 1$ to $B$ do $b_i \leftarrow K_{B+8I+i} \oplus c_i$;
 2. For $j = I$ down to 1 do
            $(lr_{B+1}, lr_{B+2}, lr_{B+3}, \ldots, lr_{2B}) =$
                $(lr_{B+1}(K_{B+(j-1)I+7}), lr_{B+2}(K_{B+(j-1)I+7}), l_{B-1}, \ldots l_{2B-4})$;
            for $i = 1$ to $B$ do $b_i \ggg_{lr_{B+i}}$;
            $*_2 = *_{K_{B+(j-1)I+5} \cdots K_{B+(j-1)I+8}}$ and $\circ_2 = \circ_{K_{B+(j-1)I+5} \cdots K_{B+(j-1)I+8}}$
            $OT^{-1}_{*_2,\circ_2}(b_B, b_{B-1} \ldots b_1) = a_B, a_{B-1} \ldots a_1$
            $(lr_1, lr_2, lr_3, \ldots, lr_B) =$
                $(lr_1(K_{B+(j-1)I+3}), lr_2(K_{B+(j-1)I+3}), l_1, \ldots l_{B-2})$;
            for $i = 1$ to $B$ do $b_i \ggg_{lr_i}$;
            $*_1 = *_{K_{B+(j-1)I+1} \cdots K_{B+(j-1)I+4}}$ and $\circ_1 = \circ_{K_{B+(j-1)I+1} \cdots K_{B+(j-1)I+4}}$
            $OT^{-1}_{*_1,\circ_1}(a_1, a_2 \ldots a_B) = b_1, b_2 \ldots b_B$
 3. For $i = 1$ to $B$ do $m_i = (b_i - RC_i) \oplus K_i$
```

Let denote the Encryption (Decryption) algorithm by EA_K (DA_K). The algorithms EA_K and DA_K for fixed K can be considered as transformations of the set Q^B and since

$$EA_K(DA_K(m_1m_2 \ldots m_B)) = m_1m_2 \ldots m_B$$

and

$$DA_K(EA_K(m_1m_2 \ldots m_B)) = m_1m_2 \ldots m_B,$$

we have

Theorem 23 *The transformations EA_K and DA_K are permutations of the set Q^B.* □

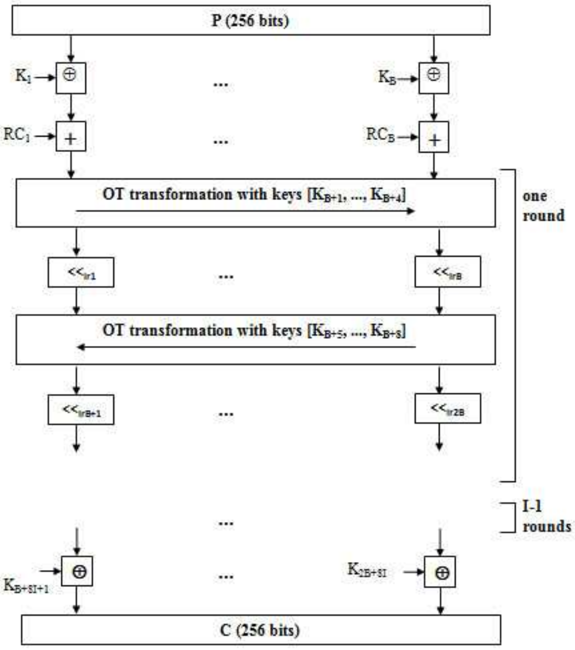

Figure 5: Alex'smile-(B, I, G)

Alex'smile has a special key expansion and key schedule algorithm, that needs to provide $2B + 8I$ words for expansion key, where $4I \geqslant G/32$.

```
Key expansion and key schedule algorithm of Alex'smile-$(B, I, G)$
Input: A key bytes $k = (k_0, \ldots, k_{G/8-1})$, a tweak words $(T_1, \ldots, T_4)$, $N = G/32$
        an $8 \times 8$ $S$-box and round constants $(RK_1, \ldots, RK_{2B+8I})$ as 32-bit words
Output: An expanded key $K = (K_1, \ldots, K_{2B+8I})$.
  1. $sum = 0$; For $i = 0$ to $N - 1$ do
         $KK_{i+1} = (k_{4i+3}||k_{4i}||k_{4i+1}||k_{4i+2}) \oplus RK_{i+1}$;
         $sum = sum + KK_{i+1}$;
  2. For $i = 5$ to $2B + 8I$ do $T_i = T_{(i \ mod \ 4)+1}$
  2. From $sum$ bytes $(s_1||s_2||s_3||s_4)$ we make $RS = (S(s_4)||S(s_3)||S(s_2)||S(s_1))$
  4. $KK_{N+1} = (RS + T_{N+1}) \oplus RK_{N+1}$;
     $KK_{N+2} = (KK_N)_{\ll 8} + (KK_{N-1})_{\gg 5}) \oplus RK_{N+2}$;
  5. For $i = N + 3$ to $2B + 8I - 1$ step 2 do
         $KK_i = ((KK_i - 2)_{\ll 7} + (KK_{i-1})_{\gg 4}) \oplus RK_i$;
         $KK_{i+1} = ((KK_i)_{\ll 8} + (KK_{i-1})_{\gg 5}) \oplus RK_{i+1}$;
  6. $*_1 = *_{0\ldots0}$ and $\circ_1 = \circ_{0\ldots0}$
               $OT_{*_1,\circ_1}(KK_1, KK_2 \ldots KK_{32}) = K_1, K_2 \ldots K_{32}$
```

It can be seen that the first N words are filled with bytes from the secret key, which are then xored with round constants. We find the sum of the first N words, and after that we produce the word RS from bytes of the sum, taken in reverse order and mapped with the given S box. Next word is obtained in special way, by the word RS, the tweak word and the round constant. In this way, we diffuse all secret key bits in all next calculated expanded key words with non-linearity of given S box. Every next word is sum of the previous two words, where one is rotated to the left and one is rotated to the right for fixed positions, xored with round constants. At the end, we apply the OT quasigroup transformation with all parameters zeros on obtained words, to produce the expanded key words. This transformation can be done "on the fly".

Implementation of Alex'smile-$(8, 2, G)$ for $G \in \{128, 192, 256\}$

We give the implementation of 256-bit Alex'smile-$(8, 2, G)$ block ciphers with key size of 128, 192 and 256 bits ($G \in \{128, 192, 256\}$). One can use shorter keys by padding them with zeros until the next larger defined key length. This implementation is very flexible, fast and simple.

Quasigroup operations via extended Feistel networks

In every round, we use two different pairs of orthogonal quasigroup operations $*_j$ and $\circ_j$ $(j = 1, 2)$. In our implementation, orthogonal quasigroups operations $*_j$ and $\circ_j$ are obtained from orthogonal orthomorphisms F_{A_j,B_j,C_j} and $F^2_{A_j,B_j,C_j}$ (extended Feistel networks, $j = 1, 2$) of the group $(\mathbb{Z}_2^{32}, \oplus_{32})$, by

$$x *_j y = x \oplus_{32} F_{A_j,B_j,C_j}(y)$$

$$x \circ_j y = x \oplus_{32} F^2_{A_j,B_j,C_j}(y)$$

We use the same S-box as NASHA (improved AES S-box with the APA structure from Cui and Cao [12], given on Table 3.1) as starting bijection and we define three extended Feistel networks $F_{a_1,b_1,c_1}, F_{a_2,b_2,c_2}, F_{a_3,b_3,c_3} : \mathbb{Z}_2^{16} \to \mathbb{Z}_2^{16}$ by

$$F_{a_i,b_i,c_i}(l_8||r_8) = (r_8 \oplus a_i)||(l_8 \oplus b_i \oplus S(r_8 \oplus c_i)),$$

where l_8 and r_8 are 8-bit variables, a_i, b_i, c_i are 8-bit words from the expanded key which are used as parameters for selecting the quasigroup operation. Denote by f' the bijection $F_{a_1,b_1,c_1} \circ F_{a_2,b_2,c_2} \circ F_{a_3,b_3,c_3} : \mathbb{Z}_2^{16} \to \mathbb{Z}_2^{16}$.

Create the extended Feistel networks $F_{A_j,B_j,C_j} : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ and $F^2_{A_j,B_j,C_j}$ $(j = 1, 2)$ by

$$F_{A_j,B_j,C_j}(l_{16}||r_{16}) = (r_{16} \oplus_{16} A_j)||(l_{16} \oplus_{16} B_j \oplus_{16} f'(r_{16} \oplus_{16} C_j)),$$

where l_{16}, r_{16} are 16-bit variables and A_j, B_j, C_j are 16-bit words, also part from the expanded key.

Definition of parameters to extended Feistel networks

The parameters of the used extended Feistel network $F_{A,B,C}$ needed for one pair of an orthogonal quasigroup operations $*$ and $\circ$ are obtained by 4 subkeys (SK_1, SK_2, SK_3, SK_4) from the extended key K. These dependencies can be written as $* = *_{SK_1,SK_2,SK_3,SK_4}$ and $\circ = \circ_{SK_1,SK_2,SK_3,SK_4}$. Every subkey SK_i can be represented as array of four bytes $(sk_{i_1}, sk_{i_2}, sk_{i_3}, sk_{i_4})$, where $i = 1 \ldots 4$. We have

$$a_1||b_1||c_1||a_2 = sk_{1_1}||sk_{1_2}||sk_{1_3}||sk_{1_4}$$
$$b_2||c_2||a_3||b_3 = sk_{2_1}||sk_{2_2}||sk_{2_3}||sk_{2_4}$$
$$c_3||d||A = sk_{3_1}||sk_{3_2}||(sk_{3_3}||sk_{3_4})$$
$$B||C = (sk_{4_1}||sk_{4_2})||(sk_{4_3}||sk_{4_4})$$

Parameter d is used for calculation of the first two rotation values as $[d/32]$ and $d\%32$, that are needed after every OT transformation.

Definition of constants

In Alex'smile-$(8, 2, G)$ we use several group of constants, all with purpose to make harder the attacker's job. 8 32-bits RC and 32 32-bits RK constants, are given in hexadecimal as:
$RC = 510e527f, ade682d1, 9b05688c, 2b3e6c1f, 1f83d9ab, fb41bd6b,$ $5be0cd19, 137e2179$

$RK = 2dd8a09a, 3c4e3efb, e07688dc, 6f166b73, 061a77a0, 60948dcd,$ $0c34aa2a, 315e01d5, 8a47ea18, 080559ce6, c785f436, 4a0b98f4, 9f22535b,$ $264607a8, 53a8c8ca, 56e1288c, 2547d84e, 9ccde59d, 3c1563a9, 317c57a1,$ $9486eb50, c7d8037f, 77341eda, d21e9a40, c0f905d7, 41c9cb74, d648813e,$ $45121dbb, 6a09e667, f3bcc908, cbbb9d5d, c1059ed8$

After every OT transformation, we use left rotation of every state word. First two rotation values are key dependent, but the other 6 are fixed. Fixed rotations are given in the following Table 3.8.

i	1	2	3	4	5	6	7	8	9	10	11	12
lr_i	3	4	5	6	7	8	12	11	2	1	23	13

Table 3.8: Fixed left rotations

Design rationales

THE CHOICE OF THE STARTING BIJECTION. When we discussed about NaSHA, we mentioned three S-boxes that where investigated: the AES S-box [15], the improved AES S-box from Liu and all [68] and the improved AES S-box with the APA structure from Cui and Cao [12]. For the same reasons as there, we chose the last one. In case of suspicion of a trapdoor being built into the block cipher, the current S-box might be replaced by other two candidates.

THE CHOICE OF THE QUASIGROUP TRANSFORMATION. We choose the orthogonal quasigroup string transformation mainly because of the following Theorem, given in [101].

Theorem 24 *Let OT be an orthogonal quasigroup string transformation defined by two orthogonal quasigroups $(Q, *_1)$ and $(Q, *_2)$. The restriction OT_t of the orthogonal quasigroup string transformation OT is a (t, t)-multipermutation, for each positive integer t.* □

PROOF OT_1 is an $(1, 1)-$ and OT_2 is a $(2, 2)$-multipermutation. We proceed by induction, and assume that OT_k are (k, k)-multipermutations for each $k < t$. Let $OT_t(x_1, x_2, \ldots, x_t) = (z_1, z_2, \ldots, z_t)$. We have $OT_{t-1}(x_1, x_2, \ldots, x_{t-1}) = (z_1, z_2, \ldots, z_{t-2}, u)$ and $(z_{t-1}, z_t) = (u{*_1}x_t, u{*_2}x_t)$. By the induction hypothesis, two different $2(t-1)$-tuples of the form $(x_1, x_2, \ldots, x_{t-1}, z_1, z_2, \ldots, z_{t-2}, u)$ cannot collide in any $t-1$ positions. Now, suppose that two different $2t$-tuples of the form $(x_1, x_2, \ldots, x_t, z_1, z_2, \ldots, z_t)$ collide in t positions. The collision cannot happen if $t - 1$ of the positions contains some elements of the set $\{x_1, x_2, \ldots, x_{t-1}, z_1, z_2, \ldots, z_{t-2}\}$. So, the collision happens at z_{t-1}, z_t and at some $t - 2$ elements of the set $\{x_1, x_2, \ldots, x_{t-1}, z_1, z_2, \ldots, z_{t-2}\}$. From $(z_{t-1}, z_t) = (u *_1 x_t, u *_2 x_t)$, since z_{t-1} and z_t collide, there are u' and x'_t such that $(z_{t-1}, z_t) = (u' *_1 x'_t, u' *_2 x'_t)$. But this is a contradiction with the orthogonality of 1 and 2. ■

In the light of the latest linear and differential attacks to the cryptographic primitives, the multipermutations are basic cryptographic tool for a perfect generation of diffusion, because, by changing i of the inputs at least $n-i+1$ of the outputs will be changed [127].

By changing the length of the string or the order of the used quasigroup we can influent the orthogonal quasigroup transformation as multipermutation. If we use larger string, we will obtain bigger multipermutation in the sense of the parameter t. If we use smaller quasigroups, we will obtain again bigger multipermutation, but also we will have the influence of the multipermutation on smaller group of bits.

Our OT quasigroup transformation is $(8,8)-$multipermutation in the encryption-decryption algorithm, and $(32,32)-$multipermutation in our key expansion and key schedule algorithm. Used quasigroups are of order 2^{32}.

THE CHOICE OF EXTENDED FEISTEL NETWORKS. Quasigroup operations in Alex'smile implementation are defined by extended Feistel networks of the groups $(\mathbb{Z}_n, \oplus_n)$, where $n = 16, 32$. There are several reasons for choosing them. First, for the OT transformation we needed two orthogonal quasigroups, and one way to obtain them is by orthogonal orthomorphisms. Extended Feistel network $F_{A,B,C}$ has at least two orthogonal orthomorphisms $F^{-1}_{A,B,C}$ and $F^2_{A,B,C}$. So, we choose the $F_{A,B,C}$ and $F^2_{A,B,C}$ for generating orthogonal quasigroups. Second, the extended Feistel network has parameters that can be changed. We made these parameters to be calculated from the expanded key, so, in that way we obtain different pair of orthogonal quasigroup operations for every quasigroup transformation. We already use this approach in NaSHA design. In this way we obtain keyed quasigroups.

THE CHOICE OF FIXED ROTATIONS AND CONSTANTS. For definition of quasigroup operations we use bitwise xoring and table lookups. So to avoid to have only xor operations and table lookups, we decide to use also left rotation of every state word after the OT transformation and addition modulo 2^{32} of the state words and the constants. In this way it is much harder for the attacker to analyze the cipher.

We use constants in key expansion and key schedule algorithm, also. In that way we remove the symmetry that exists between the rounds, because the round transformation is the same for all rounds.

To avoid suspicion of a trapdoor we reuse some of the constants from NaSHA, but any other constants can also be used.

THE CHOICE OF THE KEY EXPANSION AND KEY SCHEDULE ALGORITHM. Our implementation use much more key words (32) than it is provided by the actual key (4, 6 or 8). For that aim, we introduce a key expansion and key schedule algorithm. In our algorithm we reuse the S-box for high non-linearity and the OT matrix for high diffusion. This algorithm has also another input, the tweak of length 128 bits, which is used to obtain variability of encryption functions. With OT, we eliminate the possibility of existing a pair of different secret keys that produce the same expanded key. Key bits in every round are unique, so "slide" attacks are avoided. We decide not to use one-way function for generating subkeys, but instead we use previous generated subkeys for generating the next subkeys, together with predefined round constants. In this way we achieve fast key expansion and avoid symmetry, with minimal amount of storage for keeping the precomputed key material. Also we believe that possibility of existence of weak or related keys is very small.

The key expansion and key schedule algorithm has been chosen in that way that knowledge of a part of the secret key or round subkeys bits shall not allow determination of many other round subkeys bits. Also, important was not to allow full determination of round subkeys bits differences from the secret key differences.

length of the key	128 bits	192 bits	256 bits
$key = 0$ $tweak = 0$	$min = 39\%$ $avg = 46.22\%$ $max = 53\%$ $sd = 2.42$	$min = 39\%$ $avg = 44.63\%$ $max = 51\%$ $sd = 2.81$	$min = 35\%$ $avg = 42.50\%$ $max = 54\%$ $sd = 3.73$
$key = rand$ $tweak = 0$	$min = 41\%$ $avg = 46.56\%$ $max = 52\%$ $sd = 2.36$	$min = 38\%$ $avg = 44.39\%$ $max = 58\%$ $sd = 2.99$	$min = 35\%$ $avg = 42.31\%$ $max = 53\%$ $sd = 3.52$
$key = 0$ $tweak = rand$	$min = 41\%$ $avg = 46.29\%$ $max = 52\%$ $sd = 2.30$	$min = 38\%$ $avg = 44.46\%$ $max = 53\%$ $sd = 3.00$	$min = 35\%$ $avg = 42.34\%$ $max = 53\%$ $sd = 3.69$
$key = rand$ $tweak = rand$	$min = 41\%$ $avg = 46.39\%$ $max = 54\%$ $sd = 2.32$	$min = 35\%$ $avg = 44.60\%$ $max = 53\%$ $sd = 2.62$	$min = 34\%$ $avg = 42.70\%$ $max = 54\%$ $sd = 3.67$

Table 3.9: Avalanche effect of expanded key, when the secret key and the tweak are with all zeros or randomly generated

We reuse OT quasigroup transformation with all parameters zeros, to obtain high diffusion of the secret key bits in all expanded key bits. Our analysis shows us that by changing one bit in the secret key, we obtain more than changed 46% expanded key bits for 128 bit keys, 44% expanded key bits for 192 bit keys and 42% expanded key bits for 256 bit keys. All results are given in Table 3.9. This is close to ideal, but it is enough diffusion for protection from some slide and key-related attacks.

Avalanche effect

length of the key	128 bits	192 bits	256 bits
$key = 0$	$min = 42\%$	$min = 39\%$	$min = 39\%$
$message = 0$	$avg = 49.81\%$	$avg = 49.83\%$	$avg = 50.01\%$
$tweak = 0$	$max = 58\%$	$max = 57\%$	$max = 58\%$
	$sd = 3.39$	$sd = 3.20$	$sd = 3.37$
$key = 0$	$min = 40\%$	$min = 40\%$	$min = 38\%$
$message = 0$	$avg = 50.03\%$	$avg = 49.48\%$	$avg = 49.78\%$
$tweak = rand$	$max = 58\%$	$max = 61\%$	$max = 57\%$
	$sd = 3.24$	$sd = 3.15$	$sd = 3.35$
$key = rand$	$min = 40\%$	$min = 38\%$	$min = 41\%$
$message = 0$	$avg = 49.90\%$	$avg = 50.23\%$	$avg = 49.74\%$
$tweak = 0$	$max = 58\%$	$max = 57\%$	$max = 56\%$
	$sd = 3.40$	$sd = 3.36$	$sd = 3.23$
$key = rand$	$min = 42\%$	$min = 41\%$	$min = 41\%$
$message = 0$	$avg = 50.16\%$	$avg = 50.19\%$	$avg = 49.97\%$
$tweak = rand$	$max = 58\%$	$max = 60\%$	$max = 59\%$
	$sd = 3.25$	$sd = 2.95$	$sd = 3.22$
$key = 0$	$min = 41\%$	$min = 42\%$	$min = 41\%$
$message = rand$	$avg = 50.02\%$	$avg = 49.91\%$	$avg = 49.91\%$
$tweak = 0$	$max = 58\%$	$max = 59\%$	$max = 62\%$
	$sd = 3.17$	$sd = 3.13$	$sd = 3.34$
$key = 0$	$min = 42\%$	$min = 41\%$	$min = 42\%$
$message = rand$	$avg = 49.95\%$	$avg = 50.18\%$	$avg = 49.77\%$
$tweak = rand$	$max = 59\%$	$max = 60\%$	$max = 57\%$
	$sd = 3.18$	$sd = 3.18$	$sd = 3.21$
$key = rand$	$min = 40\%$	$min = 41\%$	$min = 41\%$
$message = rand$	$avg = 49.65\%$	$avg = 49.68\%$	$avg = 50.18\%$
$tweak = 0$	$max = 59\%$	$max = 58\%$	$max = 58\%$
	$sd = 2.98$	$sd = 3.24$	$sd = 3.11$
$key = rand$	$min = 40\%$	$min = 40\%$	$min = 41\%$
$message = rand$	$avg = 50.05\%$	$avg = 50.65\%$	$avg = 49.83\%$
$tweak = rand$	$max = 58\%$	$max = 60\%$	$max = 57\%$
	$sd = 3.36$	$sd = 3.46$	$sd = 3.00$

Table 3.10: Avalanche effect of 256-bit message block, when the message block, secret key and the tweak are with all zeros or randomly generated

We tested the avalanche propagation of one bit differences in the encryption function of Alex'smile-$(8, 2, G)$ for $G \in \{128, 192, 256\}$, in 8 cases: when the message, the key and the tweak consist of all zeros or are randomly generated. We present in Table 3.10 the obtained results for 256-bit message block, where minimum, average and maximum different bits and standard deviation are given. One can see that in every case the Hamming distance is around $m/2$, or one bit difference of input bits produces about 50% different output bits, as it would be expected in theoretical models of ideal random functions.

Resistance to slide and key-related attacks

To understand the resistance of the Alex'smile-$(8, 2, G)$ to many attacks, first, it is necessary to consider how key material is used in it. Beside the usual whitening, we use round subkeys for producing keyed orthogonal quasigroup operations for every OT transformation. To obtain one pair of keyed orthogonal quasigroup operations, we use 4 expanded key words. Only 120 out of the 128 bits, are used for quasigroups, and the rest 8 bits are used for definition of two rotation values.

Let $F_{A,B,C} : \mathbb{Z}_2^{2n} \to \mathbb{Z}_2^{2n}$ be an extended Feistel network created by a bijection $f : \mathbb{Z}_2^n \to \mathbb{Z}_2^n$. In general, let exist A, B, C and A', B', C' in $\mathbb{Z}_2^n$ so the following equation is true $F_{A,B,C}(l, r) = F_{A',B',C'}(l, r)$ for every $(l, r) \in (\mathbb{Z}_2^n)^2$. We have

$$(r \oplus A, l \oplus B \oplus f(r \oplus C)) = (r \oplus A', l \oplus B' \oplus f(r \oplus C')).$$

From here we have $A = A'$ and $B \oplus B' = f(r \oplus C) \oplus f(r \oplus C') = K$, where K is a constant. Let $C \oplus C' = R$, where R is a constant. If we write $r = t \oplus C$, we obtain $f(t) \oplus f(t \oplus R) = K$, for every t.

Delot sto sleduva so prasalnici ne mi e dokazan. Treba da najdam za edna OT transformacija kolku razlicni kvazigrupi moze da se generiraat, pa posle ke ja krenam verojatnosta na stepen 4. Za prethodnoto sakam da dokazam deka vazi samo koga K=0 i R=0. Ako toa vazi delot so verojatnosti 2^{48} ke bide tocen. Delot so 2^{24} e sigurno tocen, poradi toa sto proveriv so programa za nasiot Sbox. Isto taka ako se zeme i f da e druga startna biekcija g, se dobiva slicno deka $B \oplus B' = f(r \oplus C) \oplus g(r \oplus C') = K$. I ova mi e problem da go dokazam ili samo da go ogranicam.

???The final extended Feistel network $F_{A,B,C} : \mathbb{Z}_2^{32} \to \mathbb{Z}_2^{32}$ is unique determined by the three 16-bits words A, B and C and its starting bijection f'. For a given starting bijection f' there are 2^{48} different extended Feistel networks $F_{A,B,C}$ and the same number of different quasigroups of order 2^{32}. 8-bits parameters $a_i,\ b_i,\ c_i$, uniquely determine the extended Feistel network $F_{a_i,\ b_i,\ c_i}$ $(i = 1, 2, 3)$, so we have 2^{24} different extended Feistel networks of this kind. The composite bijection f' is not necessary unique, but because of the previous, there are at least 2^{24} different compositions f' (the number is much larger, but smaller than 2^{72}).??

One can notice that, because the key expansion and key schedule algorithm use the $(32, 32)-$multipermutation OT, there is no pair of keys k and k' that gives the same expanded key sequence. So it is very unlikely that a pair of equivalent keys (a pair of secret keys that encrypt all plaintexts into the same ciphertexts) exists. Pairs of inverse keys k and k', that have the property to give always the original message after two encryptions, first with k and then with k', are also unlikely to exist at all. The same is true also for self-inverse keys, which are keys for which encrypting a block of data twice with the same key gives back the original data. We also have not found simple relations between the key, message and cipher, and strongly doubt that they exist. So, we can give a conjecture, that Alex'smile-$(8, 2, G)$ does not have weak keys.

3.5 Public-key algorithms

Public key algorithms encrypt messages using a nonsecret key. They are much slower than symmetric key algorithms, so they are usually used for key agreement and key management between two communication parties, and then, the actual communication is continued by some symmetric fast block or stream cipher algorithm.

In a public key encryption scheme a pair of encryption key and decryption key (public and private key) is generated for each user, and all the encryption keys are made public (decryption key is private key for the user). When sending a secret message to a receiver, the sender encrypts the message with the receiver's public key. Receiver decrypt the message with his private key. So, a public key encryption scheme is comprised of three algorithms: a key generation algorithm, an encryption algorithm and a decryption algorithm.

The design of a public key cryptosystem can be based on a trapdoor one-way function. A *trapdoor one-way function* is a function f onto a set X that anyone can compute efficiently; however inverting f is hard unless one is also given some "trapdoor" information. Construction on trapdoor function can be based on the hardness of discrete logarithm problem, on the difficulty of integer factorization, on the discrete logarithm problem in an additive group of points defined by elliptic curves over finite fields, on error correcting codes, on multivariate quadratic polynomials, etc. Some examples of public key encryption schemes are: RSA public key encryption, ElGamal public key encryption, McEliece public key cryptosystem, Rabin's digital signature method, Goldwasser-Micali encryption scheme, Blum-Goldwasser probabilistic public-key encryption scheme, etc.

In practice, it is very important to have certificates for users public keys. In order to certify public keys, the notion of a public key infrastructure - PKI has been developed, which is usually based on some general standard, such as X.509 or EMV. When certificates are required, it is often necessary to provide means for verifying whether a certificate has not been revoked for some reason. This is handled by means of revocation lists or on-line inquiry protocols regarding the status of a certificate. You can find more for public-key cryptography in any cryptographic book, like [132, 33].

In 2000, NIST approved Digital Signature Standard (DSS), which specifies three FIPS-approved algorithms for generating and verifying digital signatures: Digital Signature Algorithm (DSA), RSA and Elliptic Curve DSA (ECDSA).

One of the early attempts to make quasigroup based public-key algorithm is made by Keedweel [55]. He uses $CI-$quasigroups $(Q, \circ)$ with long inverse cycles for that aim. A key distributing centre would be established and only it will have knowledge of the long inverse cycle and would use it to distribute a public key $c_i^u \in Q$ and a private key $c_i^{u+1} \in Q$ to each user U_i, where $Jc_i^u = c_i^{u+1}$. Every user can perform the needed quasigroup operation $\circ$. When user U_i wish to send a message m to user U_j, he would send $c_i^u \circ m$, and U_j with his private key c_j^{u+1} will decipher as $(c_i^u \circ m) \circ c_j^{u+1} = m$. The key exchange can be done without the key distributing centre also, if sender and receiver have both knowledge of J. Then sender will choose randomly $c^u \in Q$ and he will send it together with the ciphertext $c^u \circ m$ to the receiver. The receiver will use J to obtain the c^{u+1} and to decrypt the message. Big drawback of these methods is that if the attacker knows the permutation J, he can decipher any encrypted message.

Kościelny and Mullen [62] tried to built a quasigroup-based public key cryptosystem with help of its previous defined stream-cipher [60], but this is not public-key cryptosystem in a real sense. There is no public and private keys, but only encryption and decryption procedure in which random k_x bytes, as public portion of the key, are used for initial condition of used PRNG, for obtaining the keystream K. Used quasigroup is also part of the secret key. Everybody with knowledge of used quasigroup and k_x can obtain the secret key K and can do decryption or encryption. At the end, security of this cryptosystem reduce to secret quasigroup.

The public key stream cipher based on quasigroups is given by Gligoroski [34] and interesting, according to the author, its speed can be comparable with the fastest symmetric key stream ciphers. It uses the ElGamal algorithm in the initialization phase and $E-$transformations for encryption, with appropriate $D-$transformation for decryption. The cryptographical strength of the proposed stream cipher is based on the fact that breaking it would be at least as hard as solving systems of multivariate polynomial equations modulo big prime number p which is NP-hard problem and there aren't any fast randomized or deterministic algorithms for solving it.

The used quasigroup $(Q, \circ)$ is defined by permutation in the set of $\mathbb{Z}_p^*$, where p is a big prime number with more than 1024 bits. The permutation is produced by $f_K(j) = \frac{1}{1+(K+j) \ mod \ (p-1)} \bmod p$, where $1 \leqslant K \leqslant p-2$, and then the quasigroup operation is defined by $i \circ j = i \cdot f_K(j) \bmod p$. For decryption, we need the left parastrophe $(Q, \backslash)$, which is defined as

$$i \backslash j = \begin{cases} g_K(i,j), \ g_K(i,j) \neq 0 \\ p-1, \ g_K(i,j) = 0 \end{cases} \tag{3.26}$$

where $g_K(i,j) = ((i \cdot j^{-1} \bmod p) - 1 - K) \bmod (p-1)$. So, the session key consists of number K, which determine the quasigroup and k leaders. As a prime, one can use a prime numbers of the form $p_l = 2^{8l} + 3$ (for example p_{213} and p_{251} are prime numbers with 1704 and 2008 bits respectfully).

The first trapdoor one-way function that use quasigroup string transformations with multivariate quadratic quasigroups (MQQ) is given by Gligoroski et al [47, 41]. This is a new class of trapdoor functions for building public key cryptosystems by multivariate quadratic polynomials. Obtained public key algorithm is a bijective mapping, it does not perform message expansions and can be used both for encryption and signatures. The speed of encryption of this scheme is similar to other MQ schemes, and the speed of decryption is in the range of 500–1000 times faster than the most popular public key schemes. Unfortunately, this cryptosystem was successfully broken by Mohamed et al [104] by modified version of MutantXL algorithm.

Sufficient conditions some quasigroup $(Q, \circ)$ to be MQQ is given by the following Theorem.

Theorem 25 *[47] Let $\mathbf{A_1} = [f_{ij}]_{d\times d}$ and $\mathbf{A_2} = [g_{ij}]_{d\times d}$ be two $d\times d$ matrices of linear Boolean expressions, and let $\mathbf{b_1} = [u_i]_{d\times 1}$ and $\mathbf{b_2} = [v_i]_{d\times 1}$ be two $d \times 1$ vectors of linear or quadratic Boolean expressions. Let the functions f_{ij} and u_i depend only on variables $x_1, \dots, x_d$, and let the functions g_{ij} and v_i depend only on variables $x_{d+1}, \dots, x_{2d}$. If*

$$\mathbf{Det}(\mathbf{A_1}) = \mathbf{Det}(\mathbf{A_2}) = 1 \ \textit{in } GF(2) \tag{3.27}$$

and if

$$\mathbf{A_1} \cdot (x_{d+1}, \ldots, x_{2d})^T + \mathbf{b_1} \equiv \mathbf{A_2} \cdot (x_1, \ldots, x_d)^T + \mathbf{b_2} \tag{3.28}$$

□

then the vector valued operation $*_{vv}(x_1, \ldots, x_{2d}) = \mathbf{A_1} \cdot (x_{d+1}, \ldots, x_{2d})^T + \mathbf{b_1}$ *defines a quasigroup* $(Q, *)$ *of order* 2^d *that is MQQ.*

The authors give heuristic algorithm for finding MQQ of order 2^d and of type $Quad_{d-k}Lin_k$ and with it, they generate two sets of MQQ of type $Quad_4Lin_1$ and $Quad_5Lin_0$ are generated with more than 2^{20} elements each (preprocessing phase). A generic description for this scheme can be expressed as: $T \circ P' \circ S : \{0,1\}^n \to \{0,1\}^n$ where T and S are two nonsingular linear transformations, and P' is a bijective multivariate quadratic mapping on $\{0,1\}^n$. T and S together with 8 chosen MQQs $*_1, \ldots, *_8$ form the private key. The public key consist of set of n multivariate quadratic polynomials with n variables $\mathbf{P} = \{P_i(x_1, \ldots, x_n) \mid i = 1, \ldots, n\}$, where $n = 140, 160, \ldots$ and its size is $n \cdot (1 + \frac{n(n+1)}{2})$ bits. Generation of these polynomials is done by e-transformation with chosen quasigroups and bijection of Dobbertin, with requirement - minimal rank of quadratic polynomials when represented in matrix form to be at least 8. Encryption is done by direct applying of multivariate quadratic polynomials over a vector $x = (x_1, \ldots, x_n)$, i.e. $y = \mathbf{P}(x)$. Decryption is done by using of T^{-1}, S^{-1}, Dobbertin inverse and left parastrophes $\backslash_i$ of the quasigroups $*_i$, $i = 1, \ldots, 8$. In fact, the owner of the private key need to store left parastrophes of key's quasigroups.

3.6 Some other cryptographic primitives

Marnas et al [90] have been suggested a new quasigroup based transformation scheme for All-Or-Nothing encryption (Rivest [118]). AON transformation is used for pre-processing of the message into pseudo-message, before the encryption, achieving that it is computationally infeasible for the attacker to decrypt the message if any of the pseudo-message block is missing. Quasigroup modification uses a quasigroup $(Q, \circ)$ of order 256 represented as a permutation in the set of $Q = \mathbb{Z}_{257}^*$, with which they encoded ASCII table, with one difference, 256 stands for 0. The message is transformed in pseudo-message by one $e-$transformation using fixed leader l. The message needs to be encrypted is constructed as

$$\text{message to encrypt} = \text{leader } l + \text{1st row of the quasigroup} + \\ \text{pseudo-message}$$

and it is only $257B$ longer than original message. Then the actual encryption takes place with any known algorithm. On the other side, the actual decryption is done first to obtain the pseudo-message. After that, the quasigroup $(Q, \backslash)$ is formed first, and then decryption is done by using $d-$transformation with the same leader l.

The authors did not mention one thing, that with their modification, the basic idea of AONT is violated. The attacker can start with decrypting without knowing all pseudo-message blocks. For example, if he knows only those blocks that contains the quasigroup and the leader, he can starts decrypting character by character only if he obtains characters in right order.

In [124] Satti gives an quasigroup based cryptosystem, which can be used as stream or block cipher, that involves the Trusted Authority. This cryptosystem is not elaborated enough. Encryption use only $E^{(n)}_{h_1,\dots,h_n}-$ transformations. The main difference from previous designs is that it uses different quasigroup operations for every transformation. First half of $e-$ transformations are made by different isotopies of one smaller quasigroup, and the second half by different isotopies of one bigger quasigroup. Also he suggests one not very practical way of implementing the cipher. He suggests sender and receiver to have stored one smaller quasigroup and all their isotopies as an array, and the same for the bigger quasigroup. Even more quasigroups and their isotopies must be changed in regular intervals. The choice of the quasigroups and isotopies indexing is issued by the Trusted Authority in regular intervals. The Trusted Authority use some algorithm for generating order of quasigroups and indexes of isotopies. The secret key consists of the leaders (hidden keys) and is produced by some algorithm in both communication parties.

3.7 Summary

Our contributions in this chapter are:

- a survey of quasigroup based primitives like hash functions, block and stream ciphers, PRNGs, public-key cryptosystems etc;
- new quasigroup based family of hash functions NaSHA-(m, k, r);
- implementations of NaSHA-$(m, 2, 6)$ hash functions for $m \in \{224, 256, 384, 512\}$;
- a new quasigroup based family of tweakable block ciphers Alex'smile-(B, I, G);

– implementations of Alex'smile-$(4, 2, G)$ block ciphers for $G \in \{128, 192, 256\}$.

Chapter 4

Conclusions and Future Work

In the summary we answer the research questions posed in the introduction.

- What properties should have some quasigroup, so it can be used as non-linear building block in cryptographic primitives and it can contributed to the defence of linear and differential attacks?

When we try to find quasigroups suitable for cryptography in this sense, we started from shapeless quasigroups, defined by Gligoroski et al. [43]. Additionally we investigate the prop ratio tables and correlation matrices of quasigroups and some quasigroup transformations to answer this question and we introduce a new classification of quasigroups. In the light of the recent linear and differential attacks we extend the notation of shapeless quasigroups to perfect quasigroups. It is important used quasigroups to be non-linear vector valued Boolean functions without any linear component Boolean function, without nontrivial difference propagations with prop ratio 1 and restriction weight of 0 and with every nonzero output selection vector correlated to more than one input selection vector. This is the stronger requirement and this is needed especially in the cases when we use quasigroup without any quasigroup transformation. If we use quasigroups with quasigroup transformation usually it is enough quasigroup to be only shapeless, and still to have defence to differential and linear attacks. Sometimes even a quasigroup with some structure is preferable or structure does not affect the security. In other cases quasigroups with additional restriction to the structure maybe needed, as not to be semisymmetric or Stein quasigroup or Schroeder quasigroup, etc. Also, some cryptographic primitives need special kind of quasigroups. For example, when the period of produced sequences is important, like for PRNGs and stream ciphers, quasigroup must be exponential.

– How to generate and how to compute fast operation of huge quasigroups?

We suggest a new hybrid method for definition of huge quasigroups. It integrate the known cryptographic building block, a Feistel network with orthomorphisms and Sade's diagonal method for constructing quasigroups. The complexity of our algorithm for construction of quasigroups of order 2^{2^k} is $\mathcal{O}(\texttt{log}(\texttt{log}k))$. We use group $(\mathbb{Z}_n, \oplus_n)$ as an example. But extended Feistel networks from other group need to be investigate also.

– What kind of features have huge quasigroups obtained by new construction method?

We examined quasigroups obtained by the extended Feistel networks on a group $(\mathbb{Z}_n, \oplus_n)$ and proved that they can not be perfect quasigroups, but only shapeless. These quasigroups are anti-commutative, non-associative, without left or right unit, Shroeder quasigroups, and from the choice of starting bijection, we can influence on property quasigroup to be idempotent, or to satisfy the identities of the kinds $\underbrace{x(\ldots * (x}_{k} * y)) = y,\ y = ((y * \underbrace{x) * \ldots) * x}_{k}.$

Quasigroups produced by extended Feistel networks $F_{A,B,C}$ defined on Abelian group $(\mathbb{Z}_n, \oplus_n)$ are weak-restricted, correlated and weak non-linear, but $F^2_{A,B,C}$ produces much better quasigroups which are non-correlated and pure non-linear, but steel weak-restricted quasigroups.

– In which way to use huge quasigroups as building blocks of cryptographic primitives?

The best way to use quasigroups as building blocks for cryptographic primitives is as part of some quasigroup transformation. We showed this by designing NaSHA, a new family of cryptographic hash functions and Alexsmile, a new family of block ciphers.

As future work, it is interesting to analyze quasigroups obtained by extended Feistel networks from other groups, for example dihedral groups. Also it is interesting to find a way to generate and compute fast operation of huge n-ary quasigroups, with $n > 2$, but also to investigate n-ary quasigroups as vector valued Boolean functions, their prop ratio tables and correlation matrices etc. One can try to build quasigroup transformation with n-ary quasigroups. Finally, it is interesting to analyze security of cryptographic primitives obtained by n-ary quasigroups and quasigroup transformations.

Bibliography

[1] S. Bakhtiari, R. Safavi-Naini, and J.Pieprzyk. A message authentication code based on latin square. *LNCS*, 1270:194–203, 1997.

[2] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. *Advances in Cryptology, Crypto'96, LNCS*, 1109:1–15, 1996.

[3] V. D. Belousov. *Osnovi teorii kvazigrup i lup.* Nauka, Moskva, 1967.

[4] V. D. Belousov. *n-ary kvazigrup.* Shtiintsa, Kishinev, 1972.

[5] G. B. Belyavskaya. On generalized prolongation of quasigroups. *Math. Issled.*, 5(2):28–48, 1970.

[6] Daniel J. Bernstein and Tanja Lange (editors). ebacs: Ecrypt benchmarking of cryptographic systems. http://bench.cr.yp.to, accessed 6 April 2009.

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

[8] R. H. Bruck. Simple quasigroups. *Bull. Amer. Math. Soc.*, 50:769–781, 1944.

[9] R. H. Bruck. Some results in the theory of quasigroups. *Trans. Amer. Math. Soc.*, 55:19–52, 1944.

[10] G. Carter, E. Dawson, and L. Nielsen. Desv: A latin square variation of des. In *Proc. of the Workshop on Selected Areas in Cryptography*, pages 144–158. Ottawa, Canada, 1995.

[11] J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. *Advances in Cryptology - CRYPTO 2005, LNCS*, 3621:430–448, 2005.

[12] L. Cui and Y. Cao. A new s-box structure named affine-power-affine. *International Journal of Innovative Computing, Information and Control*, 3(3):751–759, 2007.

[13] J. Daemen. *Cipher and Hash Function Design. Strategies based on Linear and Differential Cryptanalysis.* PhD thesis, Katholieke Universiteit Leuven, 1995.

[14] J. Daemen, R. Govaerts, and J. Vandewalle. Correlation matrices. In *Fast Software Encryption 1994, LNCS 1008*, pages 275–285. Springer-Verlag, 1995.

[15] J. Daemen and V. Rijmen. *The Design of Rindael: AES - The Advanced Encryption Standard.* Springer-Verlag, 2002.

[16] H. M. Damm. Totally anti-symmetric quasigroups for all orders $n \neq 2, 6$. *Discrete Mathematics*, 307(6):715–729, 2007.

[17] E. Dawson, D. Donowan, and A. Offer. Ouasigroups, isotopisms and authentification schemes. *Australasian J. of Comb.*, 13:75–88, 1996.

[18] R. D. Dean. *Formal Aspects of Mobile Code Security.* PhD thesis, Princeton University, 1999.

[19] J. Denes and A. D. Keedwell. *Latin squares and their applications.* Academic Press, Inc., 1974.

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

[21] J. Dénes and A. D. Keedwell. A new authentification scheme based on latin squares. *Discrete Math.*, 106/107:157–161, 1992.

[22] J. Dénes and A. D. Keedwell. Some applications of non-associative algebraic systems in cryptology. *Pure Mathematics and Applications*, 12(2):147–195, 2001.

[23] I. I. Deriyenko and W. A. Dudek. On prolongations of quasigroups. *Quasigroups and related systems*, 16(2):187–198, 2008.

[24] V. Dimitrova. Quasigroup transformations and their applications. Master's thesis, Faculty of Natural Science, Skopje, 2005.

[25] V. Dimitrova and J. Markovski. On quasigroup pseudo random sequence generators. In *Proc. of the 1-st Balkan Conference in Informatics*, pages 393–401. Thessaloniki, 2004.

[26] I. Dinur and A. Shamir. Cube attacks on tweakable black box polynomials. *Advances in Cryptology - EUROCRYPT 2009, LNCS*, 5479:278–299, 2009.

[27] A. L. Dulmage, N. S. Mendelsohn, and D. M. Johnson. Orthomorphisms of groups and orthogonal latin squares i. *Canad. J. Math.*, 13:356–372, 1961.

[28] J. Dvorský, E. Ochodková, and V. Snášel. Generation of large quasigroups: an application in cryptography. In *Proc. of AAA64*, 2002.

[29] J. Dvorský, E. Ochodková, and V. Snášel. Hash function based on large quasigroups. In *Proc. of Velikonocni kriptologie, Brno.* 1-8, 2002.

[30] A. B. Evans. Orthomorphism graphs of groups. *Journal of Geometry*, 32 (No. 1–2):66–74, 1989.

[31] A. B. Evans. On orthogonal orthomorphisms of cyclic and non-abelian groups. *Discrete Mathematics*, 243:229–233, 2002.

[32] H. Feistel. Cryptography and computer privacy. *Scientific American*, 228 (No. 5):15–23, 1973.

[33] N. Ferguson and B. Schneier. *Practical Cryptography.* Wiley, 1st edition, 2003.

[34] D. Gligoroski. Stream cipher based on quasigroup string transformations in $\mathbb{Z}_p^*$. *Contributions, Sec. Math. Tech. Sci., MANU*, 2004.

[35] D. Gligoroski. Candidate one-way functions and one-way permutations based on quasigroup string transformations. *Cryptology ePrint Archive, Report*, 2005/352, 2005.

[36] D. Gligoroski, V. Dimitrova, and S. Markovski. Classification of quasigroups as boolean functions, their algebraic complexity and application of gröbner bases in solving systems of quasigroup equations. In *Groebner, Coding, and Cryptography, Ed. M. Sala.* Springer, 2007.

[37] D. Gligoroski and S. Knapskog. Edon-$\mathcal{R}(256, 384, 512)$-an efficient implementation of edon-∇ family of cryptographic hash functions. *ecrypt archive*, 2007/154.

[38] D. Gligoroski and S. J. Knapskog. Adding mac functionality to edon80. *International Journal of Computer Science and Network Security*, 7(1):194–204, 2007.

[39] D. Gligoroski, S. Markovski, and S. Knapskog. A secure hash algorithm with only 8 folded sha-1 steps. *International Journal of 194 Computer Science and Network Security*, 6(10):194–205, 2006.

[40] D. Gligoroski, S. Markovski, and S. J. Knapskog. A fix of the md4 family of hash functions - quasigroup fold. In *NIST Cryptographic Hash Workshop*. Gaithersburg, Maryland, USA, 2005.

[41] D. Gligoroski, S. Markovski, and S. J. Knapskog. A new class of mulrivariate quadratic trapdoor functions based on multivariate quadratic quasigrops. In *Proc. of MATH′*08, pages 44–49. Cambridge, Massachusetts, 2008.

[42] D. Gligoroski, S. Markovski, and S. J. Knapskog. The stream cipher edon80. In *New Stream Cipher Designs: The eSTREAM Finalists, 152–169.* Springer-Verlag, 2008.

[43] D. Gligoroski, S. Markovski, and L. Kocarev. Edon-$\mathcal{R}$, an infinite family of cryptographic hash functions. In *The Second NIST Cryptographic Hash Workshop, UCSB, 275–285.* Santa Barbara, CA, 2006.

[44] D. Gligoroski, S. Markovski, Lj. Kocarev, and M. Gusev. The stream cipher edon80. Submission to eSTREAM project, 2005, http://www.ecrypt.eu.org/stream/edon80p3.html.

[45] D. Gligorovski. On the insecurity of interchanged use of ofb and cbc modes of operation. http://eprint.iacr.org/2007/385.pdf.

[46] D. Gligorovski, R.S. Ødegård, M. Mihova, S.J. Knapskog, L. Kocarev, A. Drápal, and V. Klima. Cryptographic hash function edon-r. Submission to NIST, 2008.

[47] D. Gligorovski, S. Markovski, and S. J. Knapskog. A public key block cipher based on multivariate quadratic quasigrops. Cryptology ePrint Archive, Report 2008/320.

[48] S. W. Golomb, G. Gong, and L. Mittenthal. Constructions of orthomorphisms of F_2^n. In *The* 5^{th} *International Conference on Finite Fields and Applications,* F_q5*, Germany*, pages 178–195. Springer, 1999.

[49] M. Hell and T. Johansson. A key recovery attack on edon80. *Advances in Cryptology ASIACRYPT 2007, LNCS*, 4833:568–581, 2008.

[50] T. Ito. Creation method of table, creation apparatus, creation program and program storage medium. US Patent application 20040243621, Dec. 2, 2004.

[51] L. Ji, X. Liangyu, and G. Xu. Collision attack on $nasha - 512$. *Cryptology ePrint Archive, Report*, 2008/519.

[52] X. W. Jia and Z. P. Qia. The number of latin cubes and their isotopy classes. *J. Huazhong Univ. Sci. Tech.*, 11(27):104–106, 1999.

[53] D. M. Johnson, A. L. Dulmage, and N. S. Mendelsohn. Orthomorphisms of groups and orthogonal latin squares, i. *Canadian Journal of Mathematics*, 13(3):356–372, 1961.

[54] A. Joux. Multi-collisions in iterated hash functions. applications to cascades constructions. *Advances in Cryptology - CRYPTO 2004, LNCS*, 3152:306–316, 2004.

[55] A. D. Keedwell. Crossed inverse quasigroups with long inverse cycles and applications to cryptography. *Australasian J.of Comb.*, 20:241–250, 1999.

[56] J. Kelsey and T. Kohno. Herding hash functions and the nostradamus attack. *Advances in Cryptology - EUROCRYPT 2006, LNCS*, 4004:183–200, 2006.

[57] J. Kelsey and B. Schneier. Second preimages on n-bit hash functions for much less than 2n work. *Advances in Cryptology - CRYPTO 2005, LNCS*, 3494:474–490, 2005.

[58] A. Klimov and A. Shamir. Cryptographic applications of t-functions. *LNCS*, 3006:248–261, 2002.

[59] A. Klimov and A. Shamir. A new class of invertible mappings. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, 2002.

[60] C. Kościelny. A method of constructing quasigroup-based streamciphers. *Appl. Math. and Comp. Sci.*, 6:109–121, 1996.

[61] C. Kościelny. Generating quasigroups for cryptographic applications. *Int. J. Appl. Math. Comput. Sci.*, 12(4):559–569, 2002.

[62] C. Kościelny and G. L. Mullen. A quasigroup-based public-key cryptosystem. *Int. J. Appl. Math. Comput. Sci.*, 9(4):955–963, 1999.

[63] C. F. Laywine and G. L. Mullen. *Discrete Mathematics using Latin Squares.* John Wiley & Sons, Inc., 1998.

[64] C. C. Lindner. The generalized singular direct product for quasigroups. *Can. Math. Bull.*, 14:61–63, 1971.

[65] C. C. Lindner, N. S. Mendelsohn, and S. R. Sun. On the construction of schroeder quasigroups. *Discrete Mathematics*, 3(32):271–280, 1980.

[66] H. Lipmaa and S. Moriai. Efficient algorithms for computing differential properties of addition. *FSE 2001, LNCS*, 2355:336–350, 2002.

[67] H. Lipmaa, J. Wallen, and P. Dumas. On the additive differential probability of exclusive-or. *FSE 2004, LNCS*, 3017:317–331, 2004.

[68] J. Liu, B. Wei, X. Cheng, and X. Wang. Cryptanalysis of rijndael s-box and improvement. *Applied Mathematics and Computation*, 170(2):958–975, 2005.

[69] M. Luby and C. Rackoff. How to construct pseudorandom permutations and pseudorandom functions. *SIAM J. Comput.*, 17:373–386, 1988.

[70] S. Lucks. Design principles for iterated hash functions. *Cryptology ePrint Archive, Report*, 2004/253.

[71] S. Lucks. A failure-friendly design principle for hash functions. *ASIACRYPT 2005, LNCS*, 3788:474–494, 2005.

[72] H. B. Mann. The construction of orthogonal latin squares. *The Annals of Mathematical Statistics*, 13:418–423, 1942.

[73] J. Markovski and V. Dimitrova. Improving existing prsg using qsp. In *Proc. of the CIIT*, pages 380–386. Bitola, 2003.

[74] S. Markovski. Quasigroup string processing and applications in cryptography. In *1st Conference of Mathematics and Informatics for Industry*, pages 278–290. Thessaloniki, 2003.

[75] S. Markovski, V. Dimitrova, and A. Mileva. A new method for computing the number of $n-$quasigroups. *Buletinul Academiei De Ştiinte A Republicii Moldova, Matematica*, 3(52):57–64, 2006.

[76] S. Markovski, D. Gligoroski, and S. Andova. Using quasigroups for one-one secure encoding. In *Proc. VIII Conf. Logic and Computer Science LIRA97*, pages 157–162. Novi Sad, 1997.

[77] S. Markovski, D. Gligoroski, and V. Bakeva. Quasigroup string processing - part 1. *Contributions, Sec. Math. Tech. Sci., MANU*, XX, 1-2:13–28, 1999.

[78] S. Markovski, D. Gligoroski, and V. Bakeva. Quasigroups and hash functions. In *Proc. VI Int. Conf. on Discrete Mathematics and Applications*. Bansko, Bulgaria, 2001.

[79] S. Markovski, D. Gligoroski, and V. Bakeva. On infinite class of strongly collision resistant hash functions "edon-f" with variable length of output. In *Proc.* 1^{st} *Int. Conf. on Mathematics and Informatics for Industry*, pages 302–308. Thessaloniki, 2003.

[80] S. Markovski, D. Gligoroski, and Lj. Kocarev. Unbiased random sequences from quasigroup string transformations. *LNCS*, 3557:163–180, 2005.

[81] S. Markovski, D. Gligoroski, and J. Markovski. Classification of quasigroups by random walk on torus. *Journal of applied mathematics and computing*, 19, 1-2:57–75, 2005.

[82] S. Markovski, D. Gligoroski, and B. Stojčevska. Secure two-way on-line communications by using quasigroup enchipering with almost public key. *Novi Sad Journal of Mathematics*, 30(2):43–49, 2000.

[83] S. Markovski, D. Gligoroski, and Z. Šunić. Polinomial functions on the units of $\mathbb{Z}_{2^n}$. *Journal of applied mathematics and computing*, 19, 1-2:57–75, 2009.

[84] S. Markovski and V. Kusakatov. Quasigroup string processing - part 2. *Contributions, Sec. Math. Tech. Sci., MANU*, XXI, 1-2:15–32, 2000.

[85] S. Markovski and V. Kusakatov. Quasigroup string processing - part 3. *Contributions, Sec. Math. Tech. Sci., MANU*, XXIII-XXIV, 1-2:7–27, 2002-2003.

[86] S. Markovski and A. Mileva. Nasha. Submission to NIST, 2008.

[87] S. Markovski and A. Mileva. Generating huge quasigroups from small non-linear bijections via extended feistel function. *Quasigroups and Related Systems*, 17:91–106, 2009.

[88] S. Markovski and A. Mileva. Nasha - cryptographic hash functions. In *NIST The First SHA-3 Candidate Conference*. Leuven, Belgium, 25-28 February 2009.

[89] S. Markovski, A. Mileva, V. Dimitrova, and D. Gligoroski. On a conditional collision attack on nasha-512. *Cryptology ePrint Archive, Report*, 2009/034.

[90] S. I. Marnas, L. Angelis, and G. L. Bleris. All-or-nothing transform using quasigroups. In *Proc. 1st Balkan Conference in Informatics*, pages 183–191. Thessaloniki, 2004.

[91] M. Matsui. Linear cryptanalysis method for des cipher. In *Advances in Cryptology, EUROCRYPT 1993, LNCS 765, pp. 386–397*. Springer, 1993.

[92] M. Matsumoto, M. Saito, T. Nishimura, and M. Hagita. A fast stream cipher with huge state space and quasigroup filter for software. *Selected Area in Cryptography, LNCS*, 4876:246–263, 2007.

[93] M. Matsumoto, M. Saito, T. Nishimura, and M. Hagita. Cryptmt3 stream cipher. *New Stream Cipher Designs, LNCS*, 4986:7–19, 2008.

[94] B. D. McKay, A. Meynert, and W. Myrvold. Small latin squares, quasigroups and loops. *J. Combinatorial Designs*, 15:98–119, 2007.

[95] B. D. McKay and I. M. Wanless. A census of small latin hypercubes. *SIAM Journal on Discrete Mathematics*, 12:719–736, 2008.

[96] R. C. Merkle. One way hash functions and des. *Advances in Cryptology - CRYPTO 1989, LNCS*, 435:428–446, 1990.

[97] K. A. Meyer. *A new message authentication code based on the non-associativity of quasigroups*. PhD thesis, Iowa State University, 2006.

[98] A. Mileva. Analysis of some quasigroup transformations as boolean functions. In *MASSEE International Congress on Mathematics MICOM 2009, 16-20 September, Ohrid*, 2009.

[99] A. Mileva and V. Dimitrova. Quasigroups constructed from complete mappings of a group $(\mathbb{Z}_2^n, \oplus_n)$. *Contributions, Sec. Math. Tech. Sci., MANU*, 1:1, 2009.

[100] A. Mileva and S. Markovski. Correlation matrices and prop ratio tables for quasigroups of order 4. In *The* 6^{th} *International Conference for Informatics and Information Technology, CIIT*, pages 17–22, 2008.

[101] A. Mileva and S. Markovski. Quasigroups string transformations and hash function design. a case study: The nasha hash function. In *ICT Innovations conference 2009, Ohrid*, 2009.

[102] L. Mittenthal. Block substitutions using orthomorphic mappings. *Advances in Applied Mathematics*, 16:59–71, 1995.

[103] A.R. Moghaddamfar and A.R. Zokayi. On the admissibility of finite groups. *Southeast Asian Bulletin of Mathematics*, 33:485–489, 2009.

[104] M. S. E. Mohamed, J. Ding, and J. Buchmann. Algebraic cryptanalysis of mqq public key cryptosystem by mutantxl. Cryptology ePrint Archive, Report 2008/451.

[105] G. L. Mullen and R. E. Weber. Latin cubes of order $\leqslant 5$. *Discrete Mathematics*, 32:291–297, 1980.

[106] I. Nikolić and D. Knovratovich. Free-start attacks on nasha. $http : //ehash.iaik.tugraz.at/uploads/3/33/Free - start$ ${}_attacks_on_Nasha.pdf$.

[107] V. A. Nosov. Constructing families of latin squares over boolean domains. In *Boolean Functions in Cryptology and Information Security*, pages 200–207. IOS Press, 2008.

[108] V. A. Nosov and A. E. Pankratiev. Latin squares over abelian groups. *Fundamental and applied math.*, 12(3):65–71, 2006.

[109] E. Ochadková and V. Snášel. Using quasigroups for secure encoding of file system. Abstract of Talk on Conference Security and Protection of information, Brno, 2001.

[110] National Institute of Standards and Special Publication 800-38A 2001 Technology. Recommendation for block cipher modes of operation methods and techniques. December 2001.

[111] L. J. Paige. A note on finite abelian groups. *Bull. Amer. Math. Soc.*, 53:590–593, 1947.

[112] L. J. Paige. Complete mappings of finite groups. *Pacific Journal of Mathematics*, 1:111–116, 1951.

[113] S. Paul and B. Preneel. Near optimal algorithms for solving differential equations of addition with batch queries. *In Progress in Cryptology - INDOCRYPT 2005, LNCS*, 3797:90–103, 2005.

[114] A. Petrescu. Applications of quasigroups in cryptography. In *Proc of Inter Ing 2007*, 2007.

[115] V. N. Potapov and D. S. Krotov. Asymptotics for the number of $n-$quasigroups of order 4. *Siberian Math. J.*, 47:720–731, 2006.

[116] B. Preneel. *Analysis and Design of Cryptographic Hash Functions.* PhD thesis, Katholieke Universiteit Leuven, 1993.

[117] J. Rajski and J. Tyszer. Primitive polynomials over $gf(2)$ of degree up to 660 with uniformly distributed coefficients. *Journal of Electronic Testing: Theory and Applications*, 19(6):645 – 657, 2003.

[118] R. Rivest. All-or-nothing encryption and the package transform. *Fast Software Encryption '97, Springer LNCS*, 1267:210–218, 1997.

[119] R. Rivest. Permutation polynomials modulo 2^w. *Finite Fields and Their Applications*, 7:287–292, 2001.

[120] A. Sade. Groupoides automorphes par le groupe cyclique. *Canadian Journal of Mathematics*, 9(3):321–335, 1957.

[121] A. Sade. Quasigroupes parastrophiques. expressions et identites. *Math. Nachr.*, 20:73–106, 1959.

[122] A. Sade. Produit direct singulier de quasigroups orthogonaux et anti-abéliens. *Ann. Soc. Sci. Bruxelles Ser. I*, 74:91–99, 1960.

[123] D. G. Sarvate and J. Seberry. Encryption methods based on combinatorial designs. *Ars Combinatoria*, 21A:237–246, 1986.

[124] M. Satti. A quasigroup based cryptographic system. arXiv:cs/0610017v1 [cs.CR], 2006.

[125] R. Schaufler. *Eine Anwendung zyklischer Permutationen und ihre Theorie.* PhD thesis, Marburg University, 1948.

[126] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. Twofish: A 128-bit block cipher. Submission to NIST, 1998.

[127] C. P. Schnorr and S. Vaudenay. Black box cryptanalysis of hash networks based on multipermutations. *Advances in Cryptology - EUROCRYPT 94, LNCS*, 950:47–57, 1995.

[128] V. Shcherbacov. On some known possible applications of quasigroups in cryptology. *SMIK*, 2007.

[129] J. D. H. Smith. *An introduction to quasigroups and their representations.* Academic Press, Inc., 1974.

[130] V. Snášel, A. Abraham, J. Dvorský, P. Krömer, and J. Platoš. Hash functions based on large quasigroups. *Computational Science ICCS 2009, LNCS*, 5544:521–529, 2009.

[131] S. K. Stein. On the foundations of quasigroups. *Trans. Amer. Math. Soc.*, 85:228–256, 1957.

[132] D. R. Stinson. *Cryptography: Theory and Practice.* Chapman & Hall / CRC, 2nd edition, 2002.

[133] S. Vaudenay. On the need for multipermutations: Cryptanalysis of md4 and safer. *FSE 94, LNCS*, 1008:286–297, 1995.

[134] M. Vojvoda. Cryptanalysis of one hash function based on quasigroup. *Tatra Mt. Math. Publ.*, 29(3):173–181, 2004.

[135] M. Vojvoda, M. Sýs, and M. Jókay. A note on algebraic properties of quasigroups in edon80. In *SASC*. Bochum, Germany, 2007.

[136] X. Wang, X. Lai, D. Feng, H. Chen, and H. Yu. Cryptanalysis of the hash functions md4 and ripemd. *Advances in Cryptology - EUROCRYPT 2005, LNCS*, 3494:1–18, 2005.

[137] X. Wang, Y. L. Yin, and H. Yu. Finding collisions in the full sha-1. *Advances in Cryptology - CRYPTO 2005, LNCS*, 3621:17–36, 2005.

[138] X. Wang and H. Yu. How to break md5 and other hash functions. *Advances in Cryptology - EUROCRYPT 2005, LNCS*, 3494:19–35, 2005.

[139] X. Wang, H. Yu, and Y. L. Yin. Efficient collision search attacks on sha-0. *Advances in Cryptology - CRYPTO 2005, LNCS*, 3621:1–16, 2005.

[140] M. Wegman and J. Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Science*, 22:265–279, 1981.

[141] R. L. Wilson. Quasidirect products of quasigroups. *Commun. Algebra*, 3:835–850, 1975.

[142] Federal Information New York. Data encryption standard. Processing Standards Publication **No. 46** (1977), National Bureau of Standards.

Curriculum Vitae

Aleksandra Mileva was born on the 6th of April 1975 in Štip, Republic of Macedonia. She studied computer science at the Institute of Informatics, Faculty of Natural Sciences and Mathematics, University "Ss Cyril and Methodius" of Skopje, Macedonia, and obtained the degree of Graduated Engineer in Informatics in April 1998. In October 2004 she obtained a M.Sc. degree in Informatics from the same institution. She is working now as an Assistant on Faculty of Informatics, University "Goce Delčev" of Štip.