On a Family of Minimal Candidate One-way Functions and One-way Permutations

D. Gligoroski
Department of Telematics, Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology, O.S. Bragstads plass 2E, N-7491 Trondheim, Norway
(Email: danilo.gligoroski@gmail.com)

(Received Mar. 6, 2006; revised and accepted May 7, 2006)

Abstract

In order to achieve computational workload equivalent to the exhaustive key search of an n-bit key for inversion of RSA or Diffie-Hellman one-way candidate functions the length of their arguments have to have from $10n$ to $60n$ bits. One-way functions based on Elliptic Curves in this moment are holding the record, demanding only $2n$ bits for their arguments. In this paper we propose a definition and construction of a new family of one-way candidate functions $\mathcal{R}_N : Q^N \to Q^N$, where $Q = \{0, 1, \ldots, s-1\}$ is an alphabet with s elements. Special instances of these functions can be permutations (i.e. one-way permutations). These one-way functions have the property that for achieving the security level equivalent of exhaustive key search of n-bit key, only n bits of input are needed.

Keywords: One-way functions, one-way permutations, quasigroup string transformations

1 Introduction

The concept of one-way function is the fundamental concept in the modern cryptography and was first introduced by Diffie and Hellman in their seminal paper [2]. Since then, many designed cryptographic primitives that claim that are cryptographically strong, actually suppose that they have implemented the concept of one-wayness in some proper manner. Although the current level of our mathematical knowledge does not offer an answer whether the one-way functions exist or not, their existence is conjectured and we have several well defined families of one-way candidate functions. The other way around is also true. Namely, one-way candidate functions are applied in the design of various cryptographic primitives or protocols such as: cryptographic hash functions, cryptographic random number generators, stream ciphers, signature schemas, authentication problems or key exchange protocols.

Almost all known and well established one-way functions and one-way permutations in contemporary cryptology are based on intractable problems from number theory or closely related mathematical fields such as theory of finite fields, sphere packing or coding theory. For example, the discrete logarithm problem modulo a large randomly generated prime number is the Diffie-Hellman proposal (DH) in [2] for one-way permutations, quadratic residuosity is Goldwasser and Micali proposal in [10] and RSA is an one-way permutation candidate based on the difficulty of factoring a number that is a product of two large prime numbers proposed by RSA [26]. Another popular one-way candidate function is based on the difficulty of finding discrete-logarithm on elliptic curves (ECC) proposed by Koblitz [14] and Miller [20].

There are also some one-way functions candidates based on sphere-packing problems and coding theory such as the proposals from Goldreich, Krawczyk and Luby in [5]. Constructing one-way functions based on the subset sum problem have been proposed by Impagliazzo and Naor in [11]. As far as we know, the only attempt to construct a one-way function that is completely defined by combinatorial elements is the proposal of Goldreich in [6]. The proposal is based on the combinatorial field of Expander Graphs.

In this paper we construct a new family of one-way functions and one-way permutations defined on a finite set $Q = \{0, 1, \ldots, s-1\}$ with s elements. The construction is based on the theory of quasigroups, and quasigroup string transformations. Our approach in opposite to other approaches, with an exception of [6] is completely based on a mathematical field not closely related to the field of number theory. By some of their properties, such as speed of computation and security level of inversion, quasigroup one-way functions outperform all currently known one-way candidate functions.

2 Preliminaries

Here we give only a brief overview of quasigroups and quasigroup string transformations and more detailed explanation the reader can find in [1] and [15].

Definition 1. *A quasigroup $(Q, *)$ is a groupoid, i.e. a*

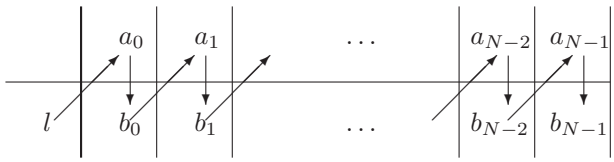


Figure 1: Graphical representation of the e–transformation of a string $A = (a_0, a_1, \ldots, a_{N-1})$

Table 1: Quasigroup $(Q, *)$

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

set Q with a binary operation $* : Q \times Q \to Q$, satisfying the law

$$(\forall u, v \in Q)(\exists!\ x, y \in Q) \quad u * x = v\ \&\ y * u = v.$$

If Q is a finite set then the main body of the multiplication table of the quasigroup is a Latin Square over the set Q. A Latin Square over Q is a $|Q| \times |Q|$-matrix such that each row and column is a permutation of Q (see for example [1]).

Next we define the basic quasigroup string transformation called e–transformation.

Definition 2. *A quasigroup e–transformation of a string $A = (a_0, a_1, \ldots, a_{N-1}) \in Q^N$ with a leader $l \in Q$ is the function $e_l : Q \times Q^N \to Q^N$ defined as $B = e_l(A)$ where $A = (a_0, a_1, \ldots, a_{N-1})$, $B = (b_0, b_1, \ldots, b_{N-1})$, $l \in Q$ and*

$$b_i := \begin{cases} l * a_0, & i = 0 \\ b_{i-1} * a_i, & 1 \le i \le N-1. \end{cases}$$

For better understanding the graphical representation of the e–transformation is shown on Figure 1.

Example 1. *Let $Q = \{0, 1, 2, 3\}$ and let the quasigroup $(Q, *)$ be given by the multiplication scheme in Table 1.*

Consider the string $A =$ 1021000 0000001 1210220 1010300 and let us choose the leader $l = 0$. Then by the e–transformation $e_0(A)$ we will obtain the following transformed string:

$$e_0(A) = 1322130\ 2130210\ 1121113\ 3013130.$$

The four consecutive applications of the e-transformation e_0 on A are represented in Table 2.

If we have a string of leaders, we can apply consecutive e–transformations on a given string, as a composition of e–transformations. That is defined by the following definition:

Definition 3. *A quasigroup E–transformation of a string $A = (a_0, a_1, \ldots, a_{N-1}) \in Q^N$ with a string of K leaders $\mathbf{L} = (l_0, l_1, \ldots, l_{K-1}) \in Q^K$ is the function $E_{\mathbf{L},K} : Q^K \times Q^N \to Q^N$ defined as $B = E_{\mathbf{L},K}(A)$ where $A = (a_0, a_1, \ldots, a_{N-1})$, $B = (b_0, b_1, \ldots, b_{N-1})$ and*

$$B = e_{l_{K-1}}(\ e_{l_{K-2}}(\ \ldots e_{l_1}(\ e_{l_0}(A)\ )\ldots\ )\ ).$$

Definition 4. *Quasigroup single reverse string transformation is the function $\mathcal{R}_1 : Q^N \to Q^N$ defined as*

$$B = \mathcal{R}_1(A) = E_{\overline{A},N}(A) = e_{a_{N-1}}(\ldots(e_{a_1}(e_{a_0}(A))))$$

where $A = (a_0, a_1, \ldots, a_{N-1})$ and $B = (b_0, b_1, \ldots, b_{N-1})$.

Definition 5. *Quasigroup double reverse transformation is the function $\mathcal{R}_2 : Q^N \to Q^N$ defined as*

$$\begin{aligned} B &= \mathcal{R}_2(A) = E_{\overline{AA},2N}(A) \\ &= e_{a_{N-1}}(\ldots(e_{a_1}(e_{a_0}(e_{a_{N-1}}(\ldots(e_{a_1}(e_{a_0}(A))))), \end{aligned}$$

where $A = (a_0, a_1, \ldots, a_{N-1})$ and $B = (b_0, b_1, \ldots, b_{N-1})$.

Example 2. *Let quasigroup $(Q, *)$ be given by the multiplication scheme in Table 1. Consider the string $A =$ 0 1 2 3 0. Then by the transformation $\mathcal{R}_1(A) = E_{\overline{A},5}(A)$ we will obtain the following transformed string: $\mathcal{R}_1(A) =$ 0 0 1 0 3 and by the transformation $\mathcal{R}_2(A) = E_{\overline{AA},10}(A)$ we will obtain the following transformed string: $\mathcal{R}_2(A) =$ 0 3 2 0 2. The calculation's steps are given in Table 3.*

3 Security Analysis and One-wayness from the Lookup Table Point of View

Both $\mathcal{R}_1$ and $\mathcal{R}_2$ are serious candidates for one-way functions, with the difference that the conjectured number of computations to invert $\mathcal{R}_1$ is $O(s^{\lfloor \frac{N}{3} \rfloor})$ and to invert $\mathcal{R}_2$ it is $O(s^N)$. In what follows we will support these claims from two perspectives: 1) From the (non)linearity of the expression by which the quasigroup $(Q, *)$ is defined, and 2) From the lookup table (Latin Square) that defines the used quasigroup $(Q, *)$. We will discuss later in this section the reasons for this approach.

First we will prove a Lemma that treats the case when the quasigroup $(Q, *)$ of order s is defined by a linear expression in the ring $\mathbb{Z}_s(+, \cdot)$. In the following two lemas we will abuse the notation for the quasigroup $(Q, *)$ and for the set $Q = \{0, 1, \ldots, s-1\}$ and instead of operation $x * y$ we will use the notation $Q(x, y)$.

Table 2: Four consecutive e-transformations of A with leader 0

	$1\ 0\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 2\ 1\ 0\ 2\ 2\ 0\ 1\ 0\ 1\ 0\ 3\ 0\ 0 = A$
0	$1\ 3\ 2\ 2\ 1\ 3\ 0\ 2\ 1\ 3\ 0\ 2\ 1\ 0\ 1\ 1\ 2\ 1\ 1\ 1\ 3\ 3\ 0\ 1\ 3\ 1\ 3\ 0 = e_0(A)$
0	$1\ 2\ 3\ 2\ 2\ 0\ 2\ 3\ 3\ 1\ 3\ 2\ 2\ 1\ 0\ 1\ 1\ 2\ 2\ 2\ 0\ 3\ 0\ 1\ 2\ 2\ 0\ 2 = e_0(e_0(A))$
0	$1\ 1\ 2\ 3\ 2\ 1\ 1\ 2\ 0\ 1\ 2\ 3\ 2\ 2\ 1\ 0\ 1\ 1\ 1\ 1\ 3\ 1\ 3\ 3\ 2\ 3\ 0\ 0 = e_0(e_0(e_0(A)))$
0	$1\ 0\ 0\ 3\ 2\ 2\ 2\ 3\ 0\ 1\ 1\ 2\ 3\ 2\ 2\ 1\ 0\ 1\ 0\ 1\ 2\ 2\ 0\ 3\ 2\ 0\ 2\ 1 = e_0(e_0(e_0(e_0(A))))$

Table 3: $\mathcal{R}_1(A)$ and $\mathcal{R}_2(A)$ transformation of the string $A = 0\ 1\ 2\ 3\ 0$

	0	1	2	3	0	$= A$
0	2	2	3	1	3	
3	2	3	1	0	3	
2	3	1	0	2	0	
1	2	2	1	1	3	
0	**0**	**0**	**1**	**0**	**3**	$= \mathcal{R}_1(A)$

	0	1	2	3	0	$= A$
0	2	2	3	1	3	
3	2	3	1	0	3	
2	3	1	0	2	0	
1	2	2	1	1	3	
0	0	0	1	0	3	
0	2	1	0	2	0	
3	2	2	1	1	3	
2	3	2	2	2	0	
1	2	3	2	3	0	
0	**0**	**3**	**2**	**0**	**2**	$= \mathcal{R}_2(A)$

Lemma 1. *If the quasigroup $(Q, *)$ of order s is defined by a linear expression in the ring $\mathbb{Z}_s(+, \cdot)$ i.e. if the operation $*$ can be expressed as $x * y \equiv Q(x, y) = \alpha x + \beta y + \gamma \bmod s, (\alpha, \beta, \gamma \in Q)$, then the problem of inverting $\mathcal{R}_1, \mathcal{R}_2 : Q^N \to Q^N$ is equivalent to a problem of solving a system of N linear equations with N unknown variables in the ring $\mathbb{Z}_s(+, \cdot)$.*

Proof. The main observation that leads to the proof of this lemma is the fact that any regular expression $Ex(a_1, a_2, \ldots, a_N)$ with N variables $a_1, a_2, \ldots, a_N$ and the operation $*$, gives a linear expression of the form: $P_1(\alpha, \beta, \gamma)a_1 + P_2(\alpha, \beta, \gamma)a_2 + \cdots + P_N(\alpha, \beta, \gamma)a_N + P_{N+1}(\alpha, \beta, \gamma)$ in the ring $\mathbb{Z}_s(+, \cdot)$ where $P_i(\alpha, \beta, \gamma)$ are polynomial expressions on α, β and γ, but since α, β and γ are predefined constants from the set $Q = \{0, 1, \ldots, s-1\}$, and the operations are performed in the ring $\mathbb{Z}_s(+, \cdot)$, the values of $P_i(\alpha, \beta, \gamma)$ are some constants from the set $Q = \{0, 1, \ldots, s-1\}$. Actually, this part can be easily proved by mathematical induction on the number N of different variables involved in the regular expression $Ex(a_1, a_2, \ldots, a_N)$, and we will omit it in this proof.

Now, the proof of the lemma follows directly from the definitions 5 and 6 if we interpret the definitions of $\mathcal{R}_1, \mathcal{R}_2$ as regular expressions with N variables $a_1, a_2, \ldots, a_N$. □

The situation becomes much different if the quasigroup $(Q, *)$ is defined by some nonlinear expression in the ring $\mathbb{Z}_s(+, \cdot)$. Namely, in that situation the following is true:

Lemma 2. *If the quasigroup $(Q, *)$ of order s is defined by a nonlinear expression in the ring $\mathbb{Z}_s(+, \cdot)$ i.e. if the operation $*$ can be expressed as $x * y \equiv Q(x, y) = P(x, y) \bmod s$, where the degree of the polynomial $P(x, y)$ is at least 2, then the problem of inverting $\mathcal{R}_1, \mathcal{R}_2 : Q^N \to Q^N$ reduces to a problem of solving a system of N multivariate polynomial equations with N unknown variables in the ring $\mathbb{Z}_s(+, \cdot)$.*

We will omit the proof of the last lemma since the proof technique is similar as the proof of the previous lemma with the difference that the obtained expressions will be multivariate polynomials in $\mathbb{Z}_s[a_1, a_2, \ldots, a_N]$.

Solving a system of N multivariate polynomials with N unknown variables is NP-complete problem (see for example [25]). From that perspective we can say that if the quasigroup is defined by a nonlinear expression in the ring $\mathbb{Z}_s(+, \cdot)$ then by Lemma 2 we have certain security guarantees that inverting the functions $\mathcal{R}_1$ and $\mathcal{R}_2$ would be similar as solving an NP-complete problem.

On the other hand, if the quasigroup is just given by its corresponding lookup table (Latin Square) and it is not possible to represent it by some linear expression, then in the next two theorems we will show that inversion of the functions $\mathcal{R}_1$ and $\mathcal{R}_2$ would require exponential number of addressing to that lookup table.

Theorem 1. *If the quasigroup $(Q, *)$ of order s is non-associative, non-commutative, and if it can not be represented by a linear expression in the ring $\mathbb{Z}_s$, then the number of computations based only on the lookup table that defines the quasigroup $(Q, *)$ in order to find the preimage for the function $\mathcal{R}_1 : Q^N \to Q^N$ is $O(s^{\lfloor \frac{N}{3} \rfloor})$.*

Proof. Let $B = (b_0, b_1, \ldots, b_{N-1})$ be given. The goal is to find a string $A = (a_0, a_1, \ldots, a_{N-1})$ that satisfies

the equality $B = E_{\overline{A},N}(A) = E_{(a_{N-1}, a_{N-2}, \ldots, a_1, a_0),N}(A)$. Further, because the final values of the string B are obtained after N consecutive operations e_{a_j} we will use the following notation: $B^{(i)} = e_{a_{N-i}}(B^{(i-1)}) = (b_0^{(i)}, b_1^{(i)}, \ldots, b_{N-2}^{(i)}, b_{N-1}^{(i)})$ for $i = \{1, \ldots, N-1\}$, and $B^{(0)} = A$, $B^{(N)} \equiv B$.

Table 4: Initial table obtained from the values of $B = (b_0, b_1, \ldots, b_{N-1})$ before making any guess for the values of $A = (a_0, a_1, \ldots, a_{N-1})$

	?	?	...	...	?
?	?	?	...	...	$b_{N-1}^{(1)}$
?	?	?	...	...	$b_{N-1}^{(2)}$
⋮	⋮	⋮	⋱	.	⋮
?	?	?	$b_2^{(N-2)}$	...	$b_{N-1}^{(N-2)}$
?	?	$b_1^{(N-1)}$	...	...	$b_{N-1}^{(N-1)}$
?	$b_0^{(N)}$	$b_1^{(N)}$	...	...	$b_{N-1}^{(N)}$

Since the quasigroup $(Q, *)$ is non-associative and non-commutative, the composition of e–transformations is fixed and it can not be changed, which is not the case if the quasigroup is commutative or associative. To solve the inverse task only by using the lookup table of the given quasigroup, is in fact a task to fill in the scheme in the Table 4, from bottom up using the properties of the quasigroup operation $*$. As a matter of fact due to the properties of quasigroup operation $*$ this scheme can be partially completed without guessing any value of A. Namely, from the N-th row we have the equations $b_i^{(N)} * x = b_{i+1}^{(N)}$, and since they are equations in a quasigroup, they have unique solutions $x = b_{i+1}^{(N-1)}$ for all $0 \leq i \leq N-1$. Then, similarly from the $N-1$-th row we have the equations $b_i^{(N-1)} * y = b_{i+1}^{(N-1)}$ that have unique solutions $y = b_{i+1}^{(N-2)}$ for all $1 \leq i \leq N-1$, and so on up to the first row of the table, where we can calculate the value of $b_{N-1}^{(1)}$.

Now, by knowing or by guessing the value of a_0 (that have range among s possible values) we can find value $b_0^{(N-1)}$, from which we can find the other values in the scheme of Table 5a, together with the value of a_{N-1}.

If we continue with choosing a_1 from all possible s values we will obtain a new value for a_{N-2}. Next, with every choice of a_i, $2 \leq i \leq \frac{N}{2}$ we will obtain also the values for a_{N-i-1}, and by knowing that, we will be in a position to complete the upper left corner of the scheme (see Table 5b). The intersection of the lower completed and the upper completed part is for $\lfloor \frac{N}{3} \rfloor$. So by choosing $\lfloor \frac{N}{3} \rfloor$ values we will obtain other values of the string A. Now, we can check whether we have made the right choice for $a_0, a_1, \ldots, a_{\lfloor \frac{N}{3} \rfloor}$ or not. Therefore, the complexity of inversion of $\mathcal{R}_1$ only by using the lookup definition of the quasigroup $(Q, *)$ is $O(s^{\lfloor \frac{N}{3} \rfloor})$. □

Theorem 2. *If the quasigroup $(Q, *)$ of order s is non-associative, non-commutative, and if it can not be represented by a linear expression in the ring $\mathbb{Z}_s$, then the number of computations based only on the lookup table that defines the quasigroup $(Q, *)$ in order to find the preimage for the function $\mathcal{R}_2 : Q^N \to Q^N$ is $O(s^N)$.*

Proof. The proof is similar to the proof for the function $\mathcal{R}_1$ except that now there is no intersection in the process of completing the scheme until the last guess for a_{N-1} is made. Therefore we have to make a guess for all N values $a_0, a_1, \ldots, a_{N-1}$ and thus the complexity of inverting the function $\mathcal{R}_2$ only by using the lookup definition of the quasigroup $(Q, *)$ is $O(s^N)$. □

As a consequence of Theorem 2 if the size of the alphabet Q is a power of 2 i.e. if $s = 2^k$, then we have the following Corollary.

Corollary 1. *Let the size of the alphabet Q be a power of 2 i.e. let $s = 2^k$. For any natural number N let define $n = N \times k$. If the quasigroup $(Q, *)$ is non-associative, non-commutative, and if it can not be represented by a linear expression in the ring $\mathbb{Z}_s$, then the number of computations based only on the lookup table that defines the quasigroup $(Q, *)$ in order to find the preimage for the function $\mathcal{R}_2 : Q^N \to Q^N$ i.e. $\mathcal{R}_2 : \{0,1\}^n \to \{0,1\}^n$ is $O(2^n)$.*

Based on claims in Lemma 2, Theorem 1, and Theorem 2, we can make the following conjectures.

Conjecture 1. *If the quasigroup $(Q, *)$ is non-associative, non-commutative, and if it can not be represented by a linear expression in the ring $\mathbb{Z}_s$, then functions $\mathcal{R}_1$ and $\mathcal{R}_2$ are one-way functions.*

To support Conjecture 1 we would like to stress that a random quasigroup $(Q, *)$ of order s, in general will not have any algebraic property such as commutativity, associativity, neutral elements etc, and either it will be represented by some polynomial of high degree or even it will be not possible to represent it by any polynomial (for such types of quasigroups see the recent work of Grosek at al. in [7]). Thus, one possible way to deal with the problem of inversion of the functions $\mathcal{R}_1$ and $\mathcal{R}_2$ is to look at the lookup table (or Latin Square) that defines the quasigroup $(Q, *)$. In fact if there is no other representation of the quasigroup except by its lookup table, then it will be the only way. In such a case the inversion problem will have to solve a complex system of quasigroup equations. In this moment there is not enough mathematical knowledge for solving the systems of quasigroup equations in order to prove or disprove the above conjecture. However, there are several remarkable scientific works that support our conjecture. First, we would like to mention

Table 5: Step by step process of completing the multiplication table

	a_0	?	...	?	a_{N-1}
a_{N-1}	$b_0^{(1)}$	?	...	$b_{N-2}^{(1)}$	$b_{N-1}^{(1)}$
?	?	?	...	$b_{N-2}^{(2)}$	$b_{N-1}^{(2)}$
⋮	⋮	⋮	.	.	⋮
?	?	$b_1^{(N-2)}$	$b_2^{(N-2)}$	...	$b_{N-1}^{(N-2)}$
?	$b_0^{(N-1)}$	$b_1^{(N-1)}$	...	...	$b_{N-1}^{(N-1)}$
a_0	$b_0^{(N)}$	$b_1^{(N)}$	...	...	$b_{N-1}^{(N)}$

a. Completing the table when the value of a_0 is guessed.

	a_0	a_1	...	a_{N-2}	a_{N-1}
a_{N-1}	$b_0^{(1)}$	$b_1^{(1)}$	...	$b_{N-2}^{(1)}$	$b_{N-1}^{(1)}$
a_{N-2}	$b_0^{(2)}$	$b_1^{(2)}$	...	$b_{N-2}^{(2)}$	$b_{N-1}^{(2)}$
⋮	⋮	⋮	.	.	⋮
?	?	$b_1^{(N-3)}$	$b_2^{(N-3)}$	...	$b_{N-1}^{(N-3)}$
?	$b_0^{(N-2)}$	$b_1^{(N-2)}$	$b_2^{(N-2)}$	...	$b_{N-1}^{(N-2)}$
a_1	$b_0^{(N-1)}$	$b_1^{(N-1)}$	...	...	$b_{N-1}^{(N-1)}$
a_0	$b_0^{(N)}$	$b_1^{(N)}$	...	...	$b_{N-1}^{(N)}$

b. Completing the table when the values of a_0 and a_1 are guessed.

the works of Moore at al. [21, 23] in period from 1997 to 2001 where predictability of cellular automata was investigated, but in cases when the obtained quasigroups have some structural properties (commutativity, associativity, ...). We want to stress that our quasigroup string transformations can be seen as a special type of cellular automata operations. Then, in 1999 Goldmann and Russell [4] have shown that solving system of equations in non-abelian groups is NP-complete. Moore, Tesson and Thérien in 2001 [22] have shown NP-completeness for even more general algebraic structures i.e. structures that are monoids that are not product of Abelian group and commutative idempotent monoid.

Table 6: Schematic representation of the process of computation of the function $\mathcal{R}_N$

$\mathcal{R}_N(A)$		a_0	a_1	...	a_{N-2}	a_{N-1}
L	l_0	.	.		.	.
	l_1	.	.		.	.
	⋮	⋮	⋮	...	⋮	⋮
	$l_{P(N)}$	.	.		.	.
$\overline{A}$	a_{N-1}	.	.		.	.
	a_{N-2}	.	.		.	.
	⋮	⋮	⋮	...	⋮	⋮
	a_0	.	.		.	.
$\overline{A}$	a_{N-1}	.	.	...	.	.
	a_{N-2}	.	.	...	.	.
	⋮	⋮	⋮	⋮	⋮	⋮
	a_0	b_0	b_1	...	b_{N-2}	b_{N-1}

Next we will use the function $\mathcal{R}_2$ as a core for defining a family of one-way function candidates. The idea is that before applying the function $\mathcal{R}_2$ on some string A of length N, we would like to apply a certain number (polynomial on N) of e–transformations with leaders that are some constants from Q or they are fixed indexes that address certain letters of the string A. For that purpose we will need the following definition.

Definition 6. *Preprocessing string of leaders* $\mathbf{L} = \mathbf{L}_{Q,I_N,P(N)} = (l_0, l_1, \ldots, l_{P(N)})$ *is a string of length that is polynomial of* N *and where* $l_i \in Q \cup I_N$, $Q = \{0, 1, \ldots, s-1\}$ *and* $I_N = \{i_0, i_1, \ldots, i_{N-1}\}$ *is an index set. By convention,* $\mathbf{L}$ *can be also an empty string.*

Definition 7. *The family* $\mathcal{Q}_N$ *of quasigroup one-way functions of strings of length* N *consists of functions* $\mathcal{R}_N : Q^N \to Q^N$ *such that*

$$B = \mathcal{R}_N(A) = E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$$

where $\mathbf{L}$ *is defined in Definition 6, and* $A, B \in Q^N$. *By convention, when applying the e–transformations with index leader i.e.* $l_j \in I_N$, *then e–transformation have to be applied with the leader* a_{l_j}.

For better understanding, a schematic representation of the process of computation of the function $\mathcal{R}_N$ is given in Table 6.

Conjecture 2. *The family* $\mathcal{Q}_N$ *is a family of one-way functions.*

Example 3. *Let chose* $N = 2$ *and* $(Q, *)$ *be as in Table 1. If we interpret the elements of* $Q = \{0, 1, 2, 3\}$ *as two-bit letters* $\{00, 01, 10, 11\}$ *then by having* $N = 2$ *we will define*

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

a. $(3, 3, i_1, i_0)$ b. $(3, 3, i_0, i_1)$

Figure 2: Functions obtained by $\mathbf{L}$ being a. $(3, 3, i_1, i_0)$ and b. $(3, 3, i_0, i_1)$

function $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ *from* $\{0, 1, \ldots, 15\}$ *into itself. If we chose* $\mathbf{L} = (3, 3, i_1, i_0)$*, then* $E_{(3,3,i_1,i_0)\overline{AA},8}(A)$ *is represented in Figure 2a. Notice that the function is permutation. On the other hand if we choose* $\mathbf{L} = (3, 3, i_0, i_1)$ *then we will get a function that is not a permutation. That is represented in Figure 2b. Particular computations for the string* $01 \equiv 1$ *in both cases are shown in Table 7.*

4 One-way Functions v.s. One-way Permutations – Non-fractal v.s. Fractal Quasigroups

Having defined families of one-way candidate functions, in this section we will examine in which case functions $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ are permutations, and when they are not. Actually, we will describe our experimental findings that give some directions for possible mathematical answers to these questions.

There are a lot of classifications of quasigroups of a specific order. Two main classifications are obtained by using the algebraic properties of the quasigroups: (1) classes of isotopic quasigroups, which are known only for quasigroups of orders up to 10 [19] and (2) classes of isomorphic quasigroups [1]. The importance of quasigroup classification is noted in many papers that deal with these algebraic structures (for example see [18], [16]).

From the point of view of this paper, classification of quasigroups can be done according to the nature of the one-way functions obtained by each quasigroup.

Since the number of quasigroups increases exponentially by the order of the quasigroup, we have made our experiments mainly for order 4 and some of our conjectures we have tested also on quasigroups of order 5. The total number of quasigroups of order 4 is 576. Our experiments have shown that the set of all 576 quasigroups of order 4 can be divided into two classes. One class $\mathcal{F}$ contains 192 quasigroups and the other class $\mathcal{NF}$ contains 384 quasigroups. If we order all quasigroups lexicographically from 1 to 576, then the class $\mathcal{F}$ contains the following quasigroups: $\mathcal{F}$ ={ 1, 2, 3, 4, 5, 7, 9, 11, 14, 18, 21, 24, 25, 26, 27, 28, 37, 40, 43, 46, 49, 51, 54, 57, 60, 63, 70, 71, 77, 80, 82, 83, 92, 93, 100, 101, 110, 111, 113, 116, 121, 126, 127, 132, 133, 138, 139, 144, 145, 146, 147, 148, 157, 160, 163, 166, 169, 170, 171, 172, 174, 176, 178, 179, 182, 185, 189, 192, 196, 197, 203, 206, 212, 213, 218, 222, 223, 228, 229, 232, 234, 235, 242, 243, 246, 252, 253, 259, 262, 263, 269, 272, 274, 275, 284, 285, 292, 293, 302, 303, 305, 308, 314, 315, 318, 324, 325, 331, 334, 335, 342, 343, 345, 348, 349, 354, 355, 359, 364, 365, 371, 374, 380, 381, 385, 388, 392, 395, 398, 399, 401, 403, 405, 406, 407, 408, 411, 414, 417, 420, 429, 430, 431, 432, 433, 438, 439, 444, 445, 450, 451, 456, 461, 464, 466, 467, 476, 477, 484, 485, 494, 495, 497, 500, 506, 507, 514, 517, 520, 523, 526, 528, 531, 534, 537, 540, 549, 550, 551, 552, 553, 556, 559, 563, 566, 568, 570, 572, 573, 574, 575, 576}. (By the way, the quasigroup defined in Table 1 by which we have performed examples in this paper has the lexicographic number 355.)

From numerous experiments that we have performed, we can post the following conjectures.

Conjecture 3. *For any quasigroup* $(Q, *) \in \mathcal{F}$ *and for every natural number* N *there exists at least one string* $\mathbf{L}$ *such that the function* $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ *is a permutation in the set* $\{0, 1, \ldots, 2^{2N} - 1\}$.

Conjecture 4. *For any quasigroup* $(Q, *) \in \mathcal{NF}$ *and for every natural number* N *there is no string* $\mathbf{L}$ *such that the function* $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ *is a permutation in the set* $\{0, 1, \ldots, 2^{2N} - 1\}$.

The classes $\mathcal{F}$ and $\mathcal{NF}$ have another interesting "graphical" property. Namely, if we take the periodic string 01230123..., and treat every letter as a pixel with the corresponding color, then by consecutive application of e–transformations with any constant leader l the set of

Table 7: Transformation of the string $A = 0\ 1$ when $\mathbf{L} = (3, 3, i_1, i_0)$ (on the left) and $\mathbf{L} = (3, 3, i_0, i_1)$ (on the right)

	0	1	$\equiv 0001 \equiv 1$
3	0	1	
3	0	1	
1	3	3	
0	3	1	
1	2	2	
0	0	0	
1	3	0	
0	**3**	**0**	$\equiv 1100 \equiv 12$

	0	1	$\equiv 0001 \equiv 1$
3	0	1	
3	0	1	
0	2	2	
1	1	1	
1	0	1	
0	2	2	
1	1	1	
0	**1**	**0**	$\equiv 0100 \equiv 4$

576 quasigroups can be divided into two classes: A class of quasigroups that give self-similar i.e. fractal images, and the class of quasigroups that give non self-similar images. As an example on Figure 3a we show the image obtained by the quasigroup number 46, and on Figure 3b the image obtained by the quasigroup number 47.

In [17] one can find the same classification, but instead of terms "fractal" and "non-fractal" the classes are named by an other property of them - a class of linear and a class of exponential quasigroups. In the same paper it is mentioned that when the order of quasigroup increases, the number of fractal (linear) quasigroups decreases exponentially compared to the number of non-fractal quasigroups. An additional classification that is close to the fractal – non-fractal classification can be found in [16].

It is really amazing how our experimental findings about the fractal – non-fractal classification of quasigroups comply with the classification of quasigroups that give one-way permutations and one-way functions. An open problem is to investigate the relation between these two classifications. Here even without precise definition of what "fractal" quasigroup would mean, we just give the following conjecture.

Conjecture 5. *The classes of fractal quasigroups and quasigroups for which there is a permutation* $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ *coincide.*

5 Some Properties of the Quasigroup One-way Functions

In this section we would like to set the following convention: For a random oracle in the sense of Rudich and Impagliazzo works on one-way functions ([27], [12]), we will take any quasigroup $(Q, *)$ of order s together with the family $\mathcal{Q}_N$ of one-way functions that can be defined by that quasigroup.

Rudich in his PhD thesis [27], based on a combinatorial conjecture (which was proved in 2000 by Kahn, Saks and Smith in [13]) concluded that there exist oracles for which there exist one-way functions, but there are no one-way permutations. That is in perfect compliance with our case of quasigroup one-way functions. If the oracle (quasigroup) is non-fractal, our Conjecture 4 says that there are no strings $\mathbf{L}_{Q,I_N,P(N)}$ such that the function $E_{\mathbf{L}\overline{AA},P(N)+2N}(A)$ is a permutation.

Impaliazzo and Rudich in [12] showed that "There exist an oracle relative to which a strongly one-way permutation exists, but secure secret-key agreement is impossible." That is again in compliance with quasigroup one-way functions. Namely, since quasigroup one-way functions rely on combinatorial characteristics of the quasigroups, in general there are no evident "shortcuts" and properties that will define a trapdoor function, that will enable secure secret-key agreement.

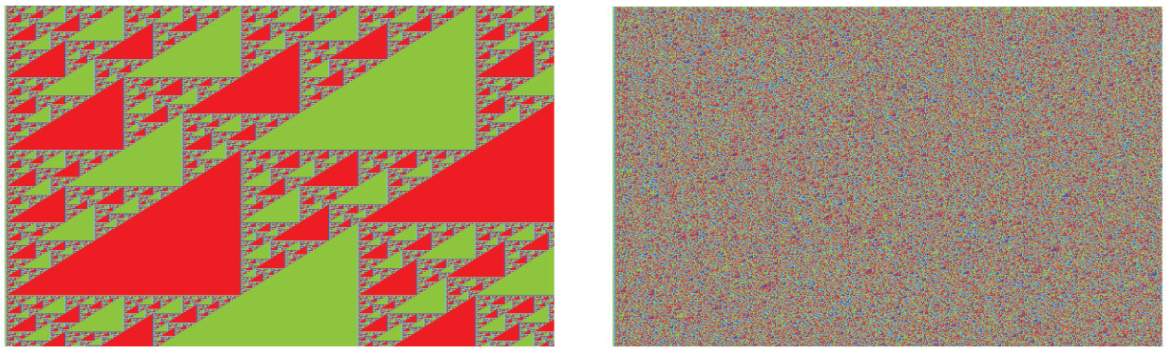
Quasigroup one-way functions are strong one-way functions i.e. there is only a small set of values on which they can be inverted in polynomial time. Thus, security amplification of a weak one-way function by an iterative process, that was established as a very useful technique in the work of Yao in 1982 [28] is not necessary for quasigroup one-way functions. This means that the speed of computation of quasigroup one-way functions can be very high. Some initial applications of quasigroup one-way functions and their properties to be easily parallelized are already done in definition of the stream cipher Edon80 [3]. In that stream cipher the IVSetup procedure is in fact a sort of quasigroup one-way function.

Speaking about the computational complexity of the quasigroup one-way functions, we can say that directly from the definition of $\mathcal{R}_1$ and $\mathcal{R}_2$ it is straightforward to prove the following Corollary.

Corollary 2. *The complexity of computing* $\mathcal{R}_1$ *is* N^2 *quasigroup operations, and the complexity of computing* $\mathcal{R}_2$ *is* $2N^2$ *quasigroup operations.*

However, in a similar manner as it was done in the design of the stream cipher Edon80 [3] there is a possibility to perform the quasigroup operations in parallel. In such a case, with a pipeline of N elements the function $\mathcal{R}_1$ can be computed in $2N$ cycles, and with a pipeline of $2N$ elements the function $\mathcal{R}_2$ can be computed in $4N$ cycles.

Moreover, instead of speaking in terms of N if we repeat the computational complexity analysis for $\mathcal{R}_1$ and $\mathcal{R}_2$ in terms of the bit length n of the strings that are mapped by those functions, then the complexity of the computations depends also on the order of the quasigroup



a. b.

Figure 3: The images obtained by consecutive e–transformations with the quasigroups of order 4 with lexicographic numbers 46 and 47. The transformations are done on a periodic string $01230123\ldots0123$ with the length 600 and with the leader 0.

that is used. If the order of the quasigroup is $2^k, k \geq 2$ (as it was considered in Corollary 1) then the number of quasigroup operations for computing $\mathcal{R}_1$ is $(\frac{n}{k})^2$ and the number of quasigroup operations for computing $\mathcal{R}_2$ is $2(\frac{n}{k})^2$. For implementing the parallel pipelined computation of $\mathcal{R}_1$ the number of computing elements have to be $\frac{n}{k}$ and then the computation of $\mathcal{R}_1$ can finish in $2\frac{n}{k}$ time units, while for $\mathcal{R}_2$ the number of computing elements have to be $2\frac{n}{k}$ and then the computation will finish in $4\frac{n}{k}$ time units.

Very similar analysis holds for the computing complexity of the family $\mathcal{Q}_N$. Namely, the number of quasigroup operations for computing any $\mathcal{R}_N$ is $P(\frac{n}{k})\frac{n}{k} + 2(\frac{n}{k})^2$ and its parallel pipelined implementation can do the computation in $2P(\frac{n}{k})+4\frac{n}{k}$ time units by using $P(\frac{n}{k})+2\frac{n}{k}$ computing elements.

From the above analysis it is clear that significant speedup in computing $\mathcal{R}_1$, $\mathcal{R}_2$ or $\mathcal{R}_N$ can be done if the order of the quasigroup is bigger. However, if the whole quasigroup is kept as a lookup table in the memory, then the price for that speedup is payed (or heavily overpaid) in the increased amount of memory needed for that lookup table. As an illustration in Table 8 we give the amount of memory needed to store some quasigroups of order $2^k, k \geq 2$.

Thus, in order to further speedup the computation of one-way quasigroup functions $\mathcal{R}_1$, $\mathcal{R}_2$ or $\mathcal{R}_N$ it would be a challenging research task to find ways to define quasigroups of huge order (order 2^k) that will be non-associative, non-commutative and nonlinear on the ring $\mathbb{Z}_{2^k}$, but without the need to keep their whole lookup table in memory.

From Corollary 1 we have that quasigroup one-way functions can achieve the security level of 2^n computations for their inversion with the length of their input being n bits. That is the most efficient construction as far as we know compared to other candidate one-way functions that require from $2n$ to $60n$ input bits to reach the security level of 2^n. In the Table 9 we show the values that NIST considers as equivalence table [24], with one additional raw for the Quasigroup one-way candidate function.

Table 8: Memory requirements for storage of quasigroups of different order

Order of quasigroup	Memory	Order of quasigroup	Memory
2^2	4 Bytes	2^{12}	24 MB
2^3	24 Bytes	2^{13}	104 MB
2^4	128 Bytes	2^{14}	448 MB
2^5	640 Bytes	2^{15}	1.875 GB
2^6	3 KB	2^{16}	8 GB
2^7	14 KB	2^{20}	2.2 TB
2^8	64 KB	2^{24}	768 TB
2^9	288 KB	2^{28}	$2^{17.8}$ TB
2^{10}	1.25 MB	2^{32}	2^{26} TB
2^{11}	5.5 MB	2^{48}	$2^{58.585}$ TB

The last property of quasigroup one-way functions that we want to mention, and that is similar to the properties that have been already found in other one-way functions is the property of *regularity* i.e. the property of having equal number of inversions on every point of their codomain. Namely, in [8] and [9] techniques for obtaining 1–1 one-way functions are proposed if the one-way function is regular. In our numerous experiments, every time when we have used fractal quasigroup, the obtained one-way functions were either permutations or regular ones. The example that we show on Figure 2b. is an example of a regular function, where every point of its codomain has exactly two inversions. It would be a challenging task to apply the same techniques to quasigroup one-way functions.

Table 9: Comparison for the equivalent length of the key for inverting one-way candidate functions

Equivalent symmetric key size	80	112	128	192	256
RSA/DH	1024	2048	3072	7680	15360
ECC	160	224	256	384	512
Quasigroup one-way	80	112	128	192	256

6 Conclusions and Further Directions

In this paper we have given a formal definition and construction of a new family of one-way functions and one-way permutations. They are based on quasigroup string transformations, and have numerous interesting properties. By some of those properties (such as speed of computation, security level of inversion) they outperform all currently known candidate one-way functions.

Some of our results concerning these functions are experimentally obtained, and we have set up several conjectures about them as well as we suggested several research directions. First research direction is the need to achieve much deeper theoretical understanding of these one-way functions. Another important research direction closely connected with findings and claims in this paper is to develop a fast heuristic or deterministic algorithm that will generate quasigroup one-way permutations. Then, it would be very important to find ways to define quasigroups of huge order (order 2^k) that will be non-associative, non-commutative and nonlinear on the ring $\mathbb{Z}_{2^k}$, but without the need to keep their whole lookup table in memory.

From the applicability point of view, we see a broad field of application of quasigroup one-way functions in the design of new cryptographic hash functions.

References

[1] J. Dénes and A.D. Keedwell, *Latin Squares and Their Applications*, English Univer, Press Ltd., 1974.

[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions Information Theory*, vol. 22, pp. 644-654, 1976.

[3] D. Gligoroski, S. Markovski, L. Kocarev, and M. Gušev, "Edon80 - Hardware synchoronous stream cipher," in *Proceedings Symmetric Key Encryption Workshop*, Århus, Denmark, May 2005.

[4] M. Goldmann and A. Russell, "The complexity of solving equations over finite groups," in *Proceedings of the 14th Annual IEEE Conference on Computational Complexity (CCC'99)*, pp. 80-86, 1999.

[5] O. Goldreich, H. Krawczyk, and M. Luby, "On the existance of pseudorandom generators," *SIAM Journal on Computing*, vol. 22, no. 6, pp. 1163-1175, 1993.

[6] O. Goldreich, "Candidate one-way functions based on expander graphs," Manuscript, 2000. (http://www.wisdom.weizmann.ac.il/∼oded/ow-candid.html)

[7] O. Grosek, P. Horák, T. van Trung, "On non-polynomial Latin squares," *Designs, Codes, and Cryptography*, vol. 32, no. 1-3, pp. 217-226, 2004.

[8] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan, and D. Zuckerman, "Security preserving amplification of hardness," in *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pp. 318- 326. IEEE, 1990.

[9] O. Goldreich, L. A. Levin, and N. Nisan, "On constructing 1-1 one-way functions", *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 2, no. 029, pp. 1-10, 1995.

[10] S. Goldwasser and S. Micali, "Probabilistic encryption", *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.

[11] R. Impagliazzo and M. Naor, "Efficient cryptographic schemes provably as secure as subset sum," in *Proceedings* 30^{th} *Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 236-241, 1989.

[12] R. Impagliazzo and S. Rudich, "Limits on the provable consequences of one-way permutations," in *Proceedings of the 21st ACM Symposium on Theory of Computing*, pp. 44-61, 1989.

[13] J. Kahn, M. Saks and C. Smyth, "A dual version of Reimer's inequality adn a proof of Rudich's conjecture" in *Proceedings of the 15th IEEE Conference on Computational Complexity*, pp. 98-103, 2000.

[14] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation,* vol. 48, pp. 203-209, 1987.

[15] S. Markovski, D. Gligoroski, and V. Bakeva, "Quasigroup string processing: Part 1," *Contributions, Sec. Math. Tech. Sci., MANU*, vol. XX 1-2, pp. 13-28, 1999.

[16] S. Markovski, D. Gligoroski, and J. Markovski, "Classification of quasigroups by random walk on torus," *Journal of Applied Mathematics & Computing*, vol. 19, no. 1-2, pp. 57-75, 2005.

[17] S. Markovski, D. Gligoroski, and L. Kocarev, "Unbiased random sequences from quasigroup string transformations," in *Proceedings of 12th International Workshop on Fast Software Encryption (FSE'05)*, LNCS 3557, pp. 163, 2005.

[18] R. L. McCasland and V. Sorge, "Automating algebra's tedious tasks: Computerised classification," in *Proceedings First Workshop on Challenges and Novel Applications for Automated Reasoning*, pp. 37-40, 2003.

[19] B. D. McKay and E. Rogoyski, "Latin squares of order 10," *Electronic Journal of Combinatorics* vol. 2, pp. 1-4, 1995. (http://ejc.math.gatech.edu:8080/Journal/journalhome.html)

[20] V. Miller, "Use of Elliptic curves in cryptography," Lecture Notes in Computer Sciences, vol. 218, Advances in cryptology – CRYPTO 85, pp. 417-426, 1985.

[21] C. Moore, "Predicting non-linear cellular automata quickly by decomposing them into linear ones," *Physica (D)*, vol. 111, pp. 27–41, 1997.

[22] C. Moore, P. Tesson, and D. Therien, "Satisfiability of systems of equations over finite monoids," in *Proceedings Mathematical Foundations of Computer Science (MFCS'01)*, Lecture Notes in Computer Science, vol. 2136, pp. 537-550, 2001.

[23] C. Moore, D. Therien, F. Lemieux, J. Berman, and A Drisko, "Circuits and expressions with nonassociative gates," *Journal of Computer and System Sciences*, vol. 60, no. 2, pp. 368-394, 2000.

[24] NIST, *Key Management Guideline - Workshop Document*, Draft, Oct. 2001. (csrc.nist.gov/encryption/kms/key-management-guideline-(workshop).pdf)

[25] J. Patarin and L. Goubin, "Trapdoor one-way permutations and multivariate polynomials," in *Proceedings 1st International Information and Communications Security Conference*, pp. 356-368, 1997.

[26] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[27] S. Rudich, *Limits on the provable consequences of one-way functions*, Ph. D Thesis, University of California at Berkeley, 1988. (ttp://www.cs.cmu.edu/∼rudich/)

[28] A. C. Yao, "Theory and application of trapdoor functions," in *Proceedings of 23th IEEE Symposium on Foundations of Computer Science*, pp. 80-91, 1982.

Danilo Gligorski received his PhD at "Ss Ciryl and Methodius" University of Skopje - Macedonia in 1997 in the field of Computer Science. His research interest are Cryptography, Computer Security, Discrete algorithms, Information Theory and Coding Theory. Now, he is a professor of Information Security, at the Department of Telematics, Faculty of Information Technology, Mathematics and Electrical Engineering, The Norwegian University of Science and Technology (NTNU), Trondheim, Norway.