An Improved 3-Quasigroup based Encryption Scheme

Sucheta Chakrabarti[1], Saibal K. Pal[1], Sugata Gangopadhyay[2]
[1]SAG, DRDO, India
suchetadrdo@hotmail.com, skptech@yahoo.com
[2]Indian Statistical Institute, India
gsugata@gmail.com

Abstract. The Crypto-community is always in search of new strong crypto-primitives to handle the present security threats and for providing efficient secure digital communication. One of the main goals of the cryptographer is to make an encryption scheme computationally fast with optimized use of memory and high cryptographic complexity. In this direction n-quasigroups (n = 2, 3) are considered as a class of new strong crypto-primitives. In this paper we propose an improved version of 3-quasigroup based encryption scheme given by Petrescu. Here we only consider reducible 3-quasigroup as the seed. It is randomly generated based on a secret key. The process of deriving different 3-ary operations for construction of reducible 3-quasigroups is described together with some related issues. We also present experimental results on 4-order reducible 3-quasigroups which are generated to find the different cases suitable for cryptographic applications.

Keywords: Quasigroup, quasi algebra, reducible 3-quasigroup, isotopy, order of quasigroup, encryption, decryption, stream cipher

1. Introduction

Stream ciphers [7], [11], [12] play an important role in secure digital communication. This category of encryption schemes is fast in implementation and can be used in applications with less computational resources. Stream ciphers form a class of symmetric key cryptographic algorithms. Here the crypto primitives are mainly the Pseudo-Random Bit Generators (PRBG) based on nonlinear combinations of different Linear Feedback Shift Registers (LFSR). Sequences based on LFSRs [7] have been widely used in the design of stream ciphers in the past decades. Today, one of the main research areas is to explore and identify new crypto primitives which are optimised both in terms of security & efficiency. In this direction, present research shows that algebraic structures based on quasigroups or n-quasigroups [4], [5], [6], [8] are suitable crypto primitives for stream ciphers and other cryptographic applications. Quasigroups [1]

S. Markovski, M. Gusev (Editors): ICT Innovations 2012, Web Proceedings, ISSN 1857-7288

have applications in coding theory also. Different transformations of quasigroups play an important role in ensuring security of the crypto algorithms based on these structures. Isotopy is one of the widely used transformation which is used to generate large number of quasigroups [3] and increase the security of the scheme.

Quasigroup based encryption scheme working as a stream cipher was first proposed by Koscielny [2] in 1996. Design and analysis of different encryption schemes for stream ciphers based on quasigroups have motivated researchers to generalize it for 3-quasigroups. These schemes have been able to exponentially increase the number of 3-quasigroups due to isotopic operations. Petrescu [8] has given a 3-quasigroup based encryption scheme for stream ciphers. The author considers the 3-quasigroup which acts as publicly known seed. In this paper, we have improved 3-quasigroup based encryption scheme given in [7] by randomly generating the quasigroup based on a key and then deriving the reducible 3-quasigroup (which plays the role of the seed for the encryption scheme). Section 2 contains the definitions and brief descriptions of the fundamental concepts related to this paper and Petrescu's encryption scheme. In Section 3 the process to randomly generate a quasigroup based on the key [10] is given. The process to derive reducible 3-quasigroups is discussed in Section 4. In Section 5 description of the improved version of the 3-quasigroup based encryption scheme is given. We have reported observations and inferences based on experimental results in Section 6 and conclusions are drawn in Section 7.

2. Preliminaries

In this section we present fundamental definitions related to this work and provide brief description of Petrescu's encryption scheme [9].

Definition 1. A Quasigroup $\langle A,\circ \rangle$ is a groupoid consisting of elements of A w.r.t. a binary operation $'\circ'$ such that $\forall a,b \in A$ there exist unique $x,y \in A$ for which it satisfies the identities $a \circ x = b \ \& \ y \circ a = b$

This means that every row and every column of the Cayley's table is a permutation of A. The cardinality of A is called the order of the quasigroup. For every finite quasigroup of order m, given by the Cayley table, it can be equivalently associated with the combinatorial design viz. a $m \times m$ Latin square.

Definition 2. An algebraic quasigroup $(A,\circ,/,\backslash)$ is an algebra with 3 binary operations which satisfy the following identities:

$(x \ / \ y) \circ y = x = (x \circ y) \ / \ y \ \ \& \ \ x \circ (x \backslash y) = y = x \backslash (x \circ y)$

Definition 1 & 2 are same when A is finite and hence both of them are called quasigroup. By quasigroup we mean the binary quasigroup denoted by 2-quasigroup. An algebraic quasigroup can be represented in general as follows:
$\langle A, \alpha, \alpha_1, \alpha_2\rangle$ is an algebra where $\alpha, \alpha_1, \alpha_2$ are three binary operations satisfying the following identities:
$\alpha(\alpha_1(x_1, x_2), x_2) = x_1 = \alpha_1(\alpha(x_1, x_2), x_2)$ & $\alpha\left(x_1, \alpha_2(x_1, x_2)\right) = x_2 = \alpha_2\left(x_1, \alpha(x_1, x_2)\right)$

Definition 3. A Ternary quasigroup (3-quasigroup) is a finite algebra $\langle A, \alpha, \alpha_1, \alpha_2, \alpha_3\rangle$ consisting of the elements of A and 4 ternary operations satisfying the following identities:

$$\alpha(\alpha_1(x_1, x_2, x_3), x_2, x_3) = x_1 = \alpha_1\big(\alpha(x_1, x_2, x_3), x_2, x_3\big)$$
$$\alpha(x_1, \alpha_2(x_1, x_2, x_3), x_3) = x_2 = \alpha_2(x_1, \alpha(x_1, x_2, x_3), x_3)$$
$$\alpha\big(x_1, x_2, \alpha_3(x_1, x_2, x_3)\big) = x_3 = \alpha_3\big(x_1, x_2, \alpha(x_1, x_2, x_3)\big)$$

If the 3-quasigroup is derived from the quasigroup then it is called the **reducible 3-quasigrouop.** We can generalise this definition to n-quasigroup.

Definition 4. Isotopy of 3-quasigroup is defined as follows:
Let $\langle A_1, \alpha\rangle$ & $\langle A_2, \beta\rangle$ be two 3-quasigroups. A_1 is isotopic to A_2 if there are four bijections
$f, f_1, f_2, f_3 : A_1 \to A_2$ such that $f\big(\alpha(x_1, x_2, x_3)\big) = \beta\big(f_1(x_1), f_2(x_2), f_3(x_3)\big)$. The ordered quadruple (f, f_1, f_2, f_3) is called an isotopism or isotopy.

Next, we briefly describe Peterscu's [9] **3-quasigroup based encryption scheme** for stream ciphers. Let $\langle A, \alpha, \alpha_1, \alpha_2, \alpha_3\rangle$ be a publicly known 3-quasigroup which is used as seed and isotopic carrier. Let $K = A^8 \times \{1, 2, 3\}$ be the key space. The key is represented as $k = a_1 a_2 \ldots a_8 i$ and it determines another isotopic quasigroup operation β on A as follows: $\beta(x_1, x_2, x_3) = f_4\left(\alpha\big(f_1^{-1}(x_1), f_2^{-1}(x_2), f_3^{-1}(x_3)\big)\right)$, where $f_j = f_{a_j}$ are permutations on A based $a_1 a_2 a_3 a_4$ of k and $a_5 a_6 a_7 a_8$ of k are Initial Value (IV) for Encryption / Decryption (E_k/D_k). Every key k uniquely determines the E_k/D_k s.t. $D_k\big(E_k(m)\big) = m$.
Let $m = m_1 m_2 \ldots$ and the encryption function $E_k(m) = c_1 c_2 \ldots$ is defined as follows:

$i = 1$	$i = 2$	$i = 3$
$c_1 = \beta(m_1, a_5, a_6)$	$c_1 = \beta(a_5, m_1, a_6)$	$c_1 = \beta(a_5, a_6, m_1)$
$c_2 = \beta(m_2, a_7, a_8)$	$c_2 = \beta(a_7, m_2, a_8)$	$c_2 = \beta(a_7, a_8, m_2)$
$for\ j > 2$	$for\ j > 2$	$for\ j > 2$
$c_j = \beta(m_j, c_{j-2}, c_{j-1})$	$c_j = \beta(c_{j-2}, m_j, c_{j-1})$	$c_j = \beta(c_{j-2}, c_{j-1}, m_j)$

The Decryption function $D_k(c_1, c_2, \dots) = m_1 m_2 \dots$ is defined as follows:

$i = 1$	$i = 2$	$i = 3$
$m_1 = \beta_1(c_1, a_5, a_6)$	$m_1 = \beta_2(a_5, c_1, a_6)$	$m_1 = \beta_3(a_5, a_6, m_1)$
$m_2 = \beta_1(c_2, a_7, a_8)$	$m_2 = \beta_2(a_7, c_2, a_8)$	$m_2 = \beta_3(a_7, a_8, m_2)$
$for\ j > 2$	$for\ j > 2$	$for\ j > 2$
$m_j = \beta_1(c_j, c_{j-2}, c_{j-1})$	$m_j = \beta_2(c_{j-2}, c_j, c_{j-1})$	$m_j = \beta_3(c_{j-2}, c_{j-1}, m_j)$

In the next section, we describe the process to generate a quasigroup randomly based on a secret key [5].

3. Randomly Generated Quasigroup based on a Key

We present two methods to generate any m-order quasigroup based on a key.

Method I - Let $A = \{1,2, \dots, m\}$ be a set of m elements. To randomly generate the quasigroup of order m, the required key length is $2m$. So the key space is as follows: $K = \{a_1 \dots a_{2m} | \ a_j \in A, 1 \le j \le 2m \}$. Let the key be $k = a_1 \dots a_{2m}$. First, we take the basic quasigroup $\langle A, \alpha \rangle$ and represented by the matrix A_s whose 1st row is the identity permutation of A and other $m-1$ rows are derived by left shift of one character of the previous row

$$A_s = \begin{bmatrix} 1 & 2 & 3 & \cdots & m \\ 2 & 3 & \cdots & m & 1 \\ & & \vdots & & \\ m & 1 & \cdots & m-1 & \end{bmatrix}$$

Now, by applying the following row swapping process based on $a_1 \cdots a_m$ of the key k we derive the matrix A_{sr}. Here we take $x \bmod m \equiv m \ if \ m|x$

$temp = A_s$
$for\ \ j = 1{:}\,m$
  $r_1 = (j + a_j) mod\ m$
  $r_2 = \left(\left\lceil \frac{m}{2} \right\rceil * j + 1 + a_j\right) mod\ m$
$Swap\ \ r_1\ and\ r_2\ of\ temp$
$A_{sr} = temp$

Similarly, as above based on $a_{m+1} \cdots a_{2m}$ of the key k the computation is carried out on A_{sr} by swapping the column m times. This process will generate a quasigroup $\langle A, \gamma \rangle$ which we called the Initial Quasigroup and represented by the matrix A_f.

If the order of the quasigroup $m > 16$ then it requires a key of more than 128 bits in length. In this case for practical reasons we can follow the 2nd Method which is given below for any $m = 2^n > 16$.

Method II – Let $A = \{1,2, \dots, m\}$ *be t*he set of $m = 2^n > 16$ elements. The key space consists of
$K = \{a_1 \dots a_{16} | \, a_j \in A, 1 \leq j \leq 2m\}$. Let the key $= a_1 \dots a_{16}$. First, take $a_1 \cdots a_8$ of the key k as the seed of a PRBG to generate $2 * 8n$ bits. Then we take n bits at a time and convert it to the decimal value $a'_j \in A$. So we derive a new key $k' = a'_1 \cdots a'_{16}$ .. Based on the key k' we calculate two row values as follows:

$for \;\; j = 1{:}\,16$
$for \; i = 1{:}\,100$
$r_1 = (j + a_j) mod \; m$
$r_2 = \left(\left\lceil \frac{m}{2} \right\rceil * j + 1 + a_j\right) mod \; m$
$Swap \;\; r_1 \; and \; r_2 \; of \; A_s$ to derive A_{sr}

Similarly, based on $a_9 \cdots a_{16}$ of the key k we generate another key k'' and calculate two column values as above and swap the columns of A_{sr} to generate the Initial Quasigroup $\langle A, \gamma \rangle$. Example of generating initial quasigroup of order 4 based on a key by using Method I is given below:

Example: Let the order of $|A| = 4$ and let the key $k = a_1 \cdots a_8 = 4\,1\,3\,1\,2\,1\,4\,3$

Here , $\left\lceil \frac{m}{2} \right\rceil = 2$ and

$$A_s = \begin{bmatrix} 1\,2\,3\,4 \\ 2\;\;3\,4\,1 \\ 3\,4\,1\,2 \\ 4\,1\,2\,3 \end{bmatrix}$$

After swapping the rows for 4 times as described in Method I

$$A_{sr} = \begin{bmatrix} 1\,2\,3\,4 \\ 3\,4\,1\,2 \\ 2\,3\,4\,1 \\ 4\,1\,2\,3 \end{bmatrix}$$

Now we apply the column swapping as described in Method I and finally derive the Initial Quasigroup

$$A_f = \begin{bmatrix} 3\,1\,4\,2 \\ 1\,3\,2\,4 \\ 4\,2\,1\,3 \\ 2\,4\,3\,1 \end{bmatrix}$$

In the next section we explain the process to derive the 3-quasigroup from the Initial Quasigroup.

4. Generation of Reducible 3-quasigroup based on a Key

Based on a key we generate the Initial quasigroup $\langle A, \gamma \rangle \equiv \langle A, \gamma, \gamma_1 \gamma_2 \rangle$. Now by using these 2-ary operations $\gamma, \gamma_1 \gamma_2$ on A we define 3-ary operations $\alpha, \alpha_1 \alpha_2 \alpha_3$ on A. There are many ways we can define 3-ary operation α from $\gamma, \gamma_1 \; and \; \gamma_2$ by using their non-associative & non-commutative properties. So, for each initial quasigroup generated based on a key the following ternary operations are defined and are arranged in the lexicographical order as follows:

1. $\alpha(x_1, x_2, x_3) = \gamma(\gamma(x_1, x_2), x_3)$
2. $\alpha(x_1, x_2, x_3) = \gamma(x_1, \; \gamma(x_2, x_3))$
3. $\alpha(x_1, x_2, x_3) = \gamma_1(\gamma_1(x_1, x_2), x_3)$
4. $\alpha(x_1, x_2, x_3) = \gamma_1(x_1, \; \gamma_1(x_2, x_3))$
5. $\alpha(x_1, x_2, x_3) = \gamma_2(\gamma_2(x_1, x_2), x_3)$
6. $\alpha(x_1, x_2, x_3) = \gamma_2(x_1, \; \gamma_2(x_2, x_3))$
7. $\alpha(x_1, x_2, x_3) = \gamma(x_1, \; \gamma_1(x_2, x_3),)$
8. $\alpha(x_1, x_2, x_3) = \gamma(\gamma_2(x_1, x_2), x_3)$
9. $\alpha(x_1, x_2, x_3) = \gamma_2(\gamma(x_1, x_2), x_3)$
10. $\alpha(x_1, x_2, x_3) = \gamma_1(\gamma_2(x_1, x_2), x_3)$
11. $\alpha(x_1, x_2, x_3) = \gamma_2(\gamma_1(x_1, x_2), x_3)$
12. $\alpha(x_1, x_2, x_3) = \gamma_2(x_1, \; \gamma_1(x_2, x_3))$

We derive different 3-ary operations on the set A and then for each 3-ary operation α we derive $\alpha_1, \alpha_2, \alpha_3$ by using the identities of $\gamma, \; \gamma_1, \gamma_2$ so that $\langle A, \alpha, \alpha_1, \alpha_2, \alpha_3 \rangle$ is a 3-quasigroup. Hence from an Initial Quasigroup we can generate a large number of 3-quasigroups which plays the role of the seed for the encryption scheme. Here we present an example of generation of a reducible 3-quasigroup.

Example: Let the Initial Quasigroup generated from a key be represented by a matrix as follows:

$$P = \begin{bmatrix} 4\,1\,3\,2 \\ 3\,2\,1\,4 \\ 2\,3\,4\,1 \\ 1\,4\,2\,3 \end{bmatrix}$$

It represents a Latin square which is the inner body of the Caley's table and hence it defines γ. Now we define two matrices which represent γ_1, γ_2 of the Initial Quasigroup $\langle A, \gamma, \gamma_1 \gamma_2 \rangle$ as follows:

$$P_1 = \begin{bmatrix} 4\,1\,2\,3 \\ 3\,2\,4\,1 \\ 2\,3\,1\,4 \\ 1\,4\,3\,2 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 2\,4\,3\,1 \\ 3\,2\,1\,4 \\ 4\,1\,2\,3 \\ 1\,3\,4\,2 \end{bmatrix}$$

Define $\alpha = \gamma(\gamma(x_1, x_2), x_3)$ and consequently $\alpha_1 = \gamma_1(\gamma_1(x_1, x_3), x_2)$, $\alpha_2 = \gamma_2(x_1, \; \gamma_1(x_2, x_3),)$ and $\alpha_3 = \gamma_2(\gamma(x_1, x_2), x_3)$. Then $\langle A, \alpha, \alpha_1, \alpha_2, \alpha_3 \rangle$ is a reducible 3-quasigroup. In the next section we present our modified 3-quasigroup based encryption scheme for stream cipher.

5. Improved Version of 3-quasigroup based Encryption Scheme

The quasigroup based encryption scheme given by Peterscu [9] considered that the seed quasigroup is known to public. So it is to be sent to the receiver which increases the communication overhead. In this case only the key which control the isotopy, Initial Value (IV) and Encryption (E_k) / Decryption (D_k) functions are unknown. The key complexity in this case is $|A|^8 * \{1,2,3\}$. This is not sufficient for providing security under practical scenario with respect to the present computational power (assumed to be) available to the attacker. We have modified this scheme to improve its cryptographic strength.

Let $K = K_1 \times K_2$ where the first part of the key space K_1 is as described in Section 3 & $K_2 = \{b_1 \cdots b_8 i \mid b_j \in A\ ,\ i \in \{1,2,3\}\}$. So for $m \leq 16$, we take $k_1 = a_1 \cdots a_{2m}$ & $k_2 = b_1 \cdots b_8\, i$ and for $m > 16$ & $m = 2^n$ where $n \in Z_+$ we take $k_1 = a_1 \cdots a_{16}$ & $k_2 = b_1 \cdots b_8\, i$. So the key is represented as $k = k_1 k_2$.

Now based on the key k_1 the random initial quasigroup is generated as described in Section 3. Also, the list of 12 different 3-ary operation α is assumed to be known. Now, based on the key $k_1$we get order as $\sum_{i=1}^{2m} a_i \, mod\ 12$ or $\sum_{i=1}^{16} a_i \, mod\ 12$ where we denote $mod\ 12 \equiv 12\ if\ 12 | x$. So we fix α by the key k_1. By using the initial quasigroup and α we generate the reducible 3-quasigroup which is taken as the seed 3-quasigroup for the Encryption scheme. For each element $\in A$, let f_a denotes the permutation of A. We define the isotopy of $\langle A,\ \alpha, \alpha_1, \alpha_2\ , \alpha_3 \rangle$ as follows:

The permutation $f_j = f_{b_j}$, $1 \leq j \leq 4$ is defined by $b_1\, b_2\, b_3\, b_4$ of k_2 as given below:

$f_j = f_{b_j} = \alpha_j(b_j, b_{j+1}, x)$; $1 \leq x \leq |A|$ and $j + 1 = j + 1\ mod\ 4\ \ if\ j + 1 > 4$

So $(f_4\ ,\ f_1\ ,\ f_2\ ,\ f_3)$ is an isotopy of 3-quasigroup $\langle A,\ \alpha, \alpha_1, \alpha_2\ , \alpha_3 \rangle$. So, by applying the isotopy on the seed 3-quasigroup we generate the new isotopic 3-quasigroup $< A, \beta, \beta_1, \beta_2, \beta_3 >$ as follows:

$$\beta(x_1, x_2,\ x_3) = f_4\left(\alpha\left(f_1^{-1}(x_1), f_2^{-1}(x_2), f_3^{-1}(x_3)\right)\right)$$
$$\beta_1(x_1, x_2,\ x_3) = f_1\left(\alpha_1\left(f_4^{-1}(x_1), f_2^{-1}(x_2), f_3^{-1}(x_3)\right)\right)$$
$$\beta_2(x_1, x_2,\ x_3) = f_2\left(\alpha_2\left(f_1^{-1}(x_1), f_4^{-1}(x_2), f_3^{-1}(x_3)\right)\right)$$
$$\beta_3(x_1, x_2,\ x_3) = f_3\left(\alpha_3\left(f_1^{-1}(x_1), f_2^{-1}(x_2), f_4^{-1}(x_3)\right)\right)$$

Now $i = 1\ ,\ 2,\ or\ 3\ of\ k_2$ uniquely determines the (E_k, D_k) as $(\beta\ , \beta_1)$, $(\beta\ , \beta_2)$ or $(\beta\ , \beta_3)$ respectively (as defined in Section 2) where $b_5\, b_6\, b_7\, b_8$ of k_2 play the role of IV for the encryption scheme.

The key complexity of our scheme for 3-quasigroups of order 4 is $2^{32} \times \{1,2,3\}$ where as in earlier case it is only $2^{16} \times \{1,2,3\}$.However due to order 4 total number of

3-quasigroups is $\ll 2^{32}$ For practical purposes, 3-quasigroups of order $256 = 2^8$ are commonly used. In our proposed modified scheme the key complexity is $(2^8)^{16} * (2^8)^8 \times \{1, 2, 3\} \approx 2^{195}$ instead of 2^{65}.For such cases number of 3-quasigroups is $\gg 2^{195}$. This key complexity is considered secure even in today's computational scenario. The block diagram of the modified scheme is given in the Appendix.

In the following section we present observations and some inferences from the experimental results.

6. Observations & Experimental Results

We have carried out experiments on the algorithm based on different reducible 3-quasigroup seed and different type of inputs under different keys. Here we discuss some important observations and inference from these experiments. Experimental results of different cases of a seed are given below:

Experiment

We take the quasigroup $\langle A, \gamma \rangle \equiv \langle A, \gamma, \gamma_1, \gamma_2 \rangle$ of order 4 where $A = \{1,2,3,4\}$ and the binary operation γ represented by matrix $P = [4\ 1\ 3\ 2;\ 3\ 2\ 1\ 4; 2\ 3\ 4\ 1; 1\ 4\ 2\ 3]$ which is a Latin Square of order 4 and hence derive P_1, P_2 which represented γ_1, γ_2. We define 3-ary operation as $\alpha = \gamma(\gamma(x_1, x_2), x_3)$ and hence calculate $\alpha_1, \alpha_2, \alpha_3$ to derive the 3-quasigroup $\langle A, \alpha \rangle \equiv \langle A, \alpha, \alpha_1, \alpha_2, \alpha_3 \rangle$, which is a reduced 3-quasigroup to be considered as a seed for the algorithm.

Case 1

(a) **Binary operation representation** P: [4 1 3 2 ; 3 2 1 4 ; 2 3 4 1;1 4 2 3]

2nd part of the KEY (k_2): [2 4 1 3 3 4 2 1 1]

Message: [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

P_1 = [4 1 2 3 ; 3 2 4 1; 2 3 1 4 ; 1 4 3 2]

P_2 = [2 4 3 1 ; 3 2 1 4 ; 4 1 2 3 ; 1 3 4 2]

Cipher:

1 4 4 2 3 2 4 3 4 1 1 3 1 2 2 1 4

4 2 3 2 4 3 4 1 1 3 1 2 2

(b) **Binary Operation** – same P as in Case1 (a)

2nd part of the KEY (k_2): [1 1 1 1 2 2 2 2 1]

Message: message:[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Cipher:

4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4

(c) **Binary Operation –** same P as in Case1 (a)

2nd part of the KEY (k_2): [3 4 3 4 4 4 3 3 1]

Message:[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

Cipher:

3 2 3 3 2 3 3 2 3 3 2 3 3 2 3 3 2 3
3 2 3 3 2 3 3 2 3 3 2 3

(d) **Binary Operation –** same P as in Case1 (a)

2nd part of the KEY (k_2): [3 4 3 4 4 4 3 3 1]

Message: [3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]

Cipher:

1 3 2 1 2 3 1 4 4 1 3 2 1 2 3 1 4 4
1 3 2 1 2 3 1 4 4 1

From these experimental results it is inferred that there exists k_2 which can make the crypts almost flatten (statistically uniform frequency distribution) even if the message is a sequence of a single element of the set. It also shows that there exists k_2 which returns the output of a sequence of single element only (which is undesirable). These features are depicted in (a) & (b) respectively. In (a) the frequency of occurrence of the element '1' i.e. freq(1) = 8, freq(2) = 8, freq(3) = 6, freq(4) = 8 and in (b) freq(1) = freq(2) = freq(3) = 0 , freq(4) = 30.

It has been also inferred from observations of different experimental results that crypts of some messages are almost flat whereas for some messages crypts are not flat under the same key. In this case (c) & (d) support the fact. In (c) freq(1) = 0, freq(2) = 10, freq(3) = 20 & freq(4) = 0 and in (d) freq(1) = 10 , freq(2) = 6 , freq(3) = 6 , freq(4) = 6.

Case2

(a) **Binary Operation –** same P as in Case1 (a)

2nd part of the KEY (k_2): [1 1 1 1 2 2 2 2 1]

Message: [1 2 3 2 2 1 1 1 3 3 2 2 3 4 4 4 3 2 1 2 3 4 3 2 1 2 3 4]

Cipher:

4 2 2 2 2 4 2 1 4 1 2 4 3 1 1 4 1 2
2 2 1 2 3 3 4 3 3 3

(b) **Binary Operation –** same P as in Case1 (a)

2nd part of the KEY (k_2): [1 1 1 1 3 3 3 3 1]

Message: [1 2 3 2 2 1 1 1 3 3 2 2 3 4 4 4 3 2 1 2 3 4 3 2 1 2 3 4]

Cipher:

4 2 2 2 2 4 2 1 4 1 2 4 3 1 1 4 1 2
2 2 1 2 3 3 4 3 3 3

It has been inferred from the experiments that there are equivalent keys in the set of keys for this scheme. In this case (a) & (b) shows that crypts of any message under two different keys are same.
Sometimes it may happen that crypts of some particular message under two different keys are same. The presence of equivalent keys reduce the key space and hence the key complexity.

Case 3

Binary Operation – same P as in Case1 (a)

2nd part of the KEY (k_2): [1 4 3 2 2 2 2 2 1]

Message: [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]

Cipher:

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2

It shows that there exist keys which returns the message even by using sophisticated operations on a 3-quasigroup. This is a cryptographic weakness. These are weak keys which should not be considered as a part of the key space and should not be used for encryption.

7. Conclusions

In this paper we have proposed a modified 3-quasigroup based stream cipher. The scheme is designed to increase the key complexity exponentially so that it may be used for present day practical applications. The first part of the key randomly generates the initial quasigroup. It selects different parameters for generation of the seed 3-

quasigroup and in a way acts as a different algorithm for encryption. Since the choice of 3-quasigroups is large even for small order and it also increases exponentially, we include this structure and use it to customize the algorithm based on the key used for encryption. In this paper, we have considered only reducible 3-quasigroups. Extension of this scheme for selecting suitable randomly generated 3-quasigroup based on the key is in progress. This would help to improve the cryptographic strength of our scheme. We have carried out large number of experiments on different order quasigroups and specifically on 4 order reducible 3-quasigroups for ease of visualization. Theoretical research based on observed results is also in progress which would help us to derive different suitable cases for cryptographic applications. Our future work in this direction also includes automatic property testing of large quasigroups generated using this process and thorough security analysis of this scheme.

References

1. Bruck R. H.: Simple Quasigroup, Bull. Amer. Math. Soc., 50, pp.769-781, (1944).
2. Koscielny C.: A Method of Constructing Quasigroup-based Stream Ciphers, Int. Journal Applied Math and Computational Sciences, Vol. 6, No. 1, 1996, pp. 109-121, (1996).
3. Koscielny C.: Generating quasigroups for cryptographic applications , Int. Journal Applied Math and Computational Sciences, Vol 12, No. 4,pp 559-569, (2002).
4. Markovski S., Gligoroski D. & Andova S.: Using Quasigroups for One-one Secure Encoding, Proc. VIII Conf. Logic and Computer Science (LIRA), Novisad, (1997).
5. Markovski S., Dimitrova V. and Mileva A.: A new method for computing the number of n-quasigroups, Buletinul Academiei De Stiinte A Republicii Moldova, Matematica, Vol 52, No 3, pp. 57-64 , (2006).
6. Markovski S.: Quasigroup String Processing and Applications in Cryptography, In 1st conference of Mathematics and Informatics for Industry, pp. 278-290, Thessaloniki, (2003).
7. Menezes A.,V Oorschot P., and Vanstone S.: Handbook of Applied Cryptography, CRC Press,(1997).
8. Petrescu A.: Applications of Quasigroups in Cryptography, Proc. Inter-Eng 2007, Univ. Petru Maior of Tg. Mures, Romania, (2007).
9. Petrescu A.: n-quasigroup Cryptographic Primitives: Stream Ciphers, Studia Univ. Babes Bolyai, Informatica, Vol. LV, No 2, pp.27-34, (2010).
10. Pal S.K., Jaiswal A. & Chamoli V.: Quasigroup based Design of New Cryptographic Schemes, Aryabhatt Journal of Mathematics & Informatics, Vol.3, No.2, pp. 277-294, (2011).

11. Stinson D. R.: Cryptography: Theory and Practice, Chapman & Hall / CRC, 3rd Edition, (2006).
12. Stallings W.: Cryptography and Network Security, Fifth Edition, Pearson,(2011).

Appendix