

### Generation of quasigroup for cryptographic application

Monisha Sharma<sup>1</sup> and M.K.Kowar<sup>2</sup>

<sup>1</sup>Shri Shankaracharya College of Engineering & Technology, Bhilai, (CG), India

<sup>2</sup>Bhilai Institute of Technology, Durg(CG), India

<sup>1</sup> monisha\_sharma10@rediffmail.com <sup>2</sup> kowar\_bit@rediffmail.com

**Abstract:** A method of generating a practically unlimited number of quasigroups of a (theoretically) arbitrary order using the computer algebra system Maple 7 is presented. This problem is crucial to cryptography and its solution permits to implement practical quasigroup-based endomorphic cryptosystems. The order of a quasigroup usually equals the number of characters of the alphabet used for recording both the plaintext and the cipher text. Moreover, it can be used for varied information *viz* text, image, etc. Many of the on going algorithms uses NLFSSR to generate pseudo random sequence and thus the suggested method can be integrated in any of the existing pseudo random sequence to further enhance their complexity. The implementation of PRSG using quasi group processing is highly scalable and fairly unpredictable. It has passed all publicly available random sequence generator tests. That is exactly what this paper provides: fast and easy ways of generating quasigroups of order up to 256 and a little more.

**Keywords:** Quasigroup, cryptography, pseudo random sequence generator (PRSG's), QPRSG, Non Linear Feedback Shift Register (NLFSSR).

#### Introduction

Random numbers have applications in many areas: simulation, game playing, cryptography, statistical sampling, and evaluation of multiple integrals, particle transport calculations, and computations in statistical physics. Out of which a quasi-random sequence is a series of numbers that makes no pretense at being random but that has important predefine statistical properties shared with random sequences.

According to Dénes and Keedwell (1999), the earliest use of quasigroups in cryptography is attributed to the German mathematician R. Shauffler. The application of quasigroups for construction stream and block ciphers, are rather rare serious contemporary efforts to utilize the simplest non-associative algebraic systems in cryptography (Ko'scielny, 1995; 1996). In this context, the work of Ritter (1998) is by all means worth noticing (Menezes *et al.*, 2001).

We present a method of generating a practically unlimited number of quasigroups of an arbitrary order (practically < 256) by means of the computer algebra system Maple 7, for applications to cryptography. The mathematical background knowledge, mandatory for understanding this article, is

Fig.1. Quasigroup of order 2

	0	1
0	0	1
1	1	0

Fig.2. Quasigroup of order 4

	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

rather extensive. It is assumed that the reader is familiar with combinatorial structures, the basic properties of groups and Galois fields and acquainted with the basics of number theory. The only essential topics concerning quasigroups and various methods of constructing quasigroup are provided in the following sections.

#### Quasigroup

There are two big families of pseudo random sequence generators (PRSGs): (1) linear PRSGs, which rely on linear congruence functions and (2) nonlinear PRSGs (Tony Warnock, 1987; Guttmann, 2001). The best-known and most widely available implementations of linear PRSGs are linear congruence functions and linear feedback shift registers. They have relatively small periods, but are also highly predictable which makes them inappropriate for cryptography and authentication. However, the production of the pseudo random sequences is relatively fast and hence they have extended use in scientific experiments. The Quasi Pseudo Random sequence generator (QPRSG) is a nonlinear PRSG with arbitrary large period (Brotherton-Ratcliffe, 1995).

A quasigroup  $(Q, *)$  is a groupoid (i.e. algebra with one binary operation  $*$  on the set  $Q$ ) satisfying the law:  $(\forall u, v \in Q)(\exists! x, y \in Q) (x * u = v \ \& \ u * y = v)$  in  $(V, \cdot)$  (Dimitrova et al, 2003). In other words the equations  $x * u = v, u * y = v$  for each given  $u, v \in Q$  have unique solutions  $x, y$

The cancellation laws hold:

$$x * y = x * z \Rightarrow y = z, \quad x * y = z * y \Rightarrow x = z$$

Define  $\backslash *$  by:  $x \backslash * y = z \Leftrightarrow y = x * z$ .

Then  $(Q, \backslash)$  is a quasigroup too.

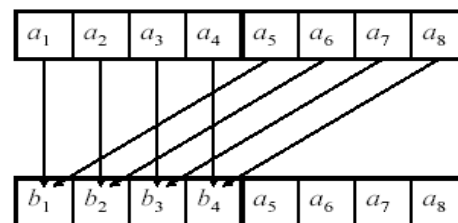
Thus we need an algebraic structure that is non-commutative, non-associative and non-idempotent. Quasigroup is one such algebraic structure.

#### Construction of quasigroup by direct product

Given effectively  $(Q, *)$ , its direct product  $(Q^n, *^n)$  can be effectively computed.  $|Q^n|$  can be huge for small values of  $|Q|$  and  $n$  in Fig.1 and Fig.2. Only  $(Q, *)$  should be stored. The operations  $*^n$  and  $\backslash^* n$  are effectively defined component wise:

$$(x_1, \dots, x_n) *^n (y_1, \dots, y_n) = (x_1 * y_1, \dots, x_n * y_n)$$

Fig.3. Quasigroup fold



$$b_1 = a_1 * a_5$$

$$b_2 = a_6 * a_2$$

$$b_3 = a_3 * a_7$$

$$b_4 = a_8 * a_4$$

$(x_1, \dots, x_n) \setminus^{*n} (y_1, \dots, y_n) = (x_1 \setminus y_1, \dots, x_n \setminus y_n)$   
 Fig.1 and Fig.2 when XOR is used

**Using fold operation**

Every 32-bit register is seen as a concatenation of 8, 4-bit variables  $a_1, a_8$  (Gligoroski *et al.*, 2005; Gligoroski *et al.*, 2009) as shown in Fig.3 and Fig.4.

**Construction of quasi group of higher order**

By the above discussed methods the construction of quasi groups of order  $n$  of complexity  $O(n^3)$  in (Smile Markovski, 2007). But it is not efficient.

For example for  $n = 218$ , 192 GB of RAM is needed and modern 3GHz computer (standard software-compiler optimizations) will need approximately a year to generate such a quasi group, that is why it is necessary to construct the quasi group of higher order. One example is shown below, where Fig.5 indicates quasi group of low order and Fig.6 shows huge order quasi group.

Fig.5. Quasigroup of order 3

*	2	1	0
2	0	2	1
1	2	1	0
0	1	0	2

$0=(0,0), 1=(0,1), 2=(0,2), 3=(1,0), 4=(1,1), 5=(1,2), 6=(2,0), 7=(2,1), 8=(2,2).$

**Methods of constructing a quasi group from other PRSG's**

Many scientific experiments require large amounts of random input data in order to simulate some process. Random sequences are inevitable in many fields like cryptography, authentication, cryptanalysis etc. For this reason the field of pseudorandom generators is widely exploited in (Smile Markovski, 2003).

The usefulness of the sequences such as derived either from linear or nonlinear equations depends in large part on there having nearly randomness properties. Therefore such sequences are termed as pseudorandom binary sequences.

The balance, run and correlation properties of these sequences make them more useful in the selection of secret keys. The Non Linear Forward Feedback Shift Register (Navin Rajpal *et al.*, 2002) generated sequences are of great importance in many fields of engineering and sciences.

A new quasi group pseudo random sequence can be obtained from NLFFSR technique and fold technique of quasi group. With this the randomness and key space of Pseudo random sequence can be improved.

**Algorithm for constructing a quasi group from NLFFSR**

1. Consider 4 bit shift register of any value except (0000)
2. Consider a feedback function to be  $f = 1+x+x^4$  and nonlinear function to be  $a_{n-1} \cdot a_{n-3} (+) a_{n-2} \cdot a_{n-4}$
3. Convert the binary sequence into integer value
  - a- Divide 16-bit sequence into four equal parts of 4 bits each.
  - b- Compare all the four parts and find it out that whether any part (4 bit each) corresponds to the number range from 1 to 4.

Fig.4. Quasigroup of order 5

	2	3	1	4	0
2	1	2	4	0	3
3	2	3	0	1	4
1	0	1	3	4	2
4	3	4	1	2	0
0	4	0	2	3	1

c- If yes, assigns that decimal number to that binary part If no then go to next step.

d- And then, considers the next part of 4bits and assigns next number and so on.

4. Use row and column of quasi group as NLFFSR sequence of order 4 and use the fold technique to

generate the group.

**Conclusion**

In this paper a new technique is developed to generate huge order QPRSG using fold technique and also QPRSG is generated using another random sequence (NLFSR) and more can be implemented for higher order QPRSG. We gave the required background and one possible implementation of a QPRSG. The implementation of PRSG using quasigroup processing is highly scalable and fairly

Fig.6. Quasigroup of order 9

	8	7	6	5	4	3	2	1	0
8	0	2	1	6	8	7	3	5	4
7	2	1	0	8	7	6	5	4	3
6	1	0	2	7	6	8	4	3	5
5	6	8	7	3	5	4	0	2	1
4	8	7	6	5	4	3	2	1	0
3	7	6	8	4	3	5	1	0	2
2	3	5	4	0	2	1	6	8	7
1	5	4	3	2	1	0	8	7	6
0	4	3	5	1	0	2	7	6	8

unpredictable. It has passed all publicly available random sequence generator tests.

**References**

1. Brotherton-Ratcliffe R (April 1995, December 1994) Using quasi-random sequences in Monte-Carlo valuation of path-dependent options. *Canadian Treasurer*. 11 (2), 36-38.
2. Dimitrova V and Markovski J (2003) On quasigroup pseudo random sequence generators, Instt. of Info. Faculty of Natural Sci. & Maths, Ss. Cyril & Methodius Univ., Arhimedova 5, Macedonia.
3. Gligoroski D, Markovski S and Knapskog L (2005) A fix of the MD4 family of Hash functions - quasigroup fold. *NIST Cryptographic Hash Workshop*. 31 Oct-1 Nov.
4. Gligoroski D, Markovski S, Kocarev L and Edon-R (2009) An infinite family of cryptographic Hash functions. *Intl. J. Network Security*. 8 (3), 293-300.
5. Gutmann P (2001) Random number generation. Available at [http://www.cypherpunks.to/~peter/06\\_random.pdf](http://www.cypherpunks.to/~peter/06_random.pdf)
6. Markovski S and Gligoroski D (2003) Differential cryptanalysis of the quasigroup cipher. Univ. of Kuopio, Finland, Ss Cyril & Methodius Univ., Republic of Macedonia.
7. Markovski S and Gligoroski D (2007) Construction of quasi groups of huge order. Ss Cyril & Methodius University Skopje, Macedonia, ICDMA7. June-17-20.
8. Menezes AJ, van Oorschot PC and Vanstone SA (2001) Handbook of applied cryptography. *CRC Press*, Canada.
9. Navin Rajpal, Anil Kumar, Sureka Dudhani and Pravesh Raja Jindal (2002) Copyright protection using non-linear forward feedback shift register and error-correction technique. In: Map India Conference, Taj Palace New Delhi, India, Feb 6-8
10. Ritter Terry (1991) The efficient generation of cryptographic confusion sequences. *Cryptologia*. I XV (2), 81-139.
11. Tony Warnock (1987) Random-number generators. *Los Alamos Sci. Special issue*, Vol 1, issue 1.