# Hardware Implementation of Quasigroup Based Encryption

Nikhil N A, D S Harish Ram

**Abstract**— this paper presents a Hardware implementation of Quasigroup based encryption. Unlike number system based approach it requires simple table lookup operations thereby enabling efficient hardware implementation. A hardware model of the encryption is proposed and synthesized to an FPGA library. Quasigroup based encryption is less affected by brute force attacks compared to the conventional methods this is because the number of quasigroups increases as the order increases in an astronomical fashion. The Verilog code is synthesized and implemented in Altera Cyclone II FPGA kit. The advantage of this method is that, it does not need to store the Quasigroup table and hence it require less memory.

**Index Terms**— Cryptography, Encryption, FPGA, Quasigroup

## 1 INTRODUCTION

Secret key encryption methods such as the Triple DES (Data Encryption Standard) or AES (Advanced Encryption Standard) have much lower computational requirements compared to public key encryption schemes. Most of these encryption schemes are vulnerable to brute force attacks. Almost all known cryptographic encryption methods use associative algebraic structures. Codes and ciphers based on non-associative structures provide better encryption compared to codes and ciphers based on associative structures. Quasigroups use a non associative structure. Quasigroup based encryption provides a new method of encrypting the data with less computation and memory requirement. Quasigroup based encryption requires simple lookup operations and does not need any complex mathematical computaion [3][4][6]. An algorithm for quasigroup generation is presented by Abraham and Ochodkova et, al [5]. Permutation can be done in the quasigroup to increase the security of the system. Pal and Sumitra present an algorithm for permutation of the quasigroup [6]. Multilevel quasigroup based encryption was proposed by Satti and Kak [8]. Several levels of permutation are done to enhance the security of the system. Quasigroup based encryption is less affected by brute force attacks compared to the conventional methods such as AES and DES. This is because the number of quasigroups increases as the order increases in an astronomical fashion [1]. Quasigroup based block cipher encryption for sensor network is proposed by Battey and Parakh [1]. The Quasigroup used for encryption should be of high cryptographic quality [5]. It should be non-associative, non-commutative and non-idempotent. The simplicity and power of quasigroup based encryption is presented by Gligoroski and Markovski [10]. They describe several algorithms that include block cipher, stream cipher and pseudo random number generator. The properties of quasiqroups are explained in [3][5][7][11]. Battey and Parakh[2] propose an efficient low memory quasigroup based random number generator for resource constrained environments. The quality of the random number generated is also tested. Dimitrova and Markovski suggest another method for random number generation [12] where the period of the random number generated depends on the property called coefficient of period growth

## 2 QUASIGROUP

Quasigroup is much similar to a Latin square. A quasgroup of order $n$ is a $n \times n$ matrix with $n$ different elements in such a way that each element occurs exactly once in each row and once in each column. Here $n$ is the order of the quasigroup. Quasigroup based encryption is defined by the operation '$\times$' such that for any two elements in the matrix there exist an inverse operation denoted by '\'. The '$\times$' and '\' are table lookup operations.
Encryption is given by

$$b_1 = k \times a_1 \qquad (1)$$

$$b_i = b_{i-1} \times a_i$$

Where a1, a2, a3……a4 is the input, k is the key and b1, b2, b3,….bi is the encrypted output. The encryption is illustrated by the following example

Example: Table 1 shows a quasigroup of order 5. Let {4,2,4,3,1,2,4} be the plain text and 2 be the key

**Table 1: Quasigroup of order 5**

| | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **0** | 4 | 3 | 2 | 1 | 0 |
| **1** | 3 | 2 | 1 | 0 | 4 |
| **2** | 2 | 1 | 0 | 4 | 3 |
| **3** | 1 | 0 | 4 | 3 | 2 |
| **4** | 0 | 4 | 3 | 2 | 1 |

$b_1 = 2 \times 4 = 3$
$b_2 = 3 \times 2 = 4$
$b_3 = 4 \times 4 = 1$

$b_4=1\times3=0$
$b_5=0\times1=3$
$b_6=3\times2=4$
$b_7=4\times4=1.$

This simple quasigroup based encryption is subjected to known plain text attack. To overcome this multiple level of encryption is done. The level of encryption in the proposed design depends on the number of Order and Key given by the user in each clock cycle. The proposed harware will store all the Order and its coresponding Key given by the user until the last key and order matches with the present key and order.

Example2: clock 1: key=2 order=5
Clock 2: key=5 order=7
Clock 3: key=10 order=20
The block of input or the plain text will be encrypted using the order 5 and key 2 first. The resultant output will be given as input to the encryptor and it will be again encrypted using the order 7 and 5 as the key.

The proposed hardware consist of a ALU unit, Register Array and a Counter. The ALU unit is used to calculate the encrypted data. Quasigroup is generated using the mathematical formula.

$$qg[\ ] = (key + order - input) \bmod order \quad (2)$$

A Register Array is used to shift the data and to store the result. The Counter is used to check for the end of encryption.The advantage of this method is that, it does not need to store the Quasigroup table. The inputs are directly given to calculate the encrypted data. So memory is not needed for the proposed method. The block diagram of the proposed method is shown in Fig1. The input to the system is the Key, Order and the Raw data.

Raw data= $a_1, a_2, a_3, \ldots\ldots a_n$

Key= $k_1, k_2, k_3, \ldots\ldots k_n$

Order= $o_1, o_2, o_3, \ldots\ldots o_n$

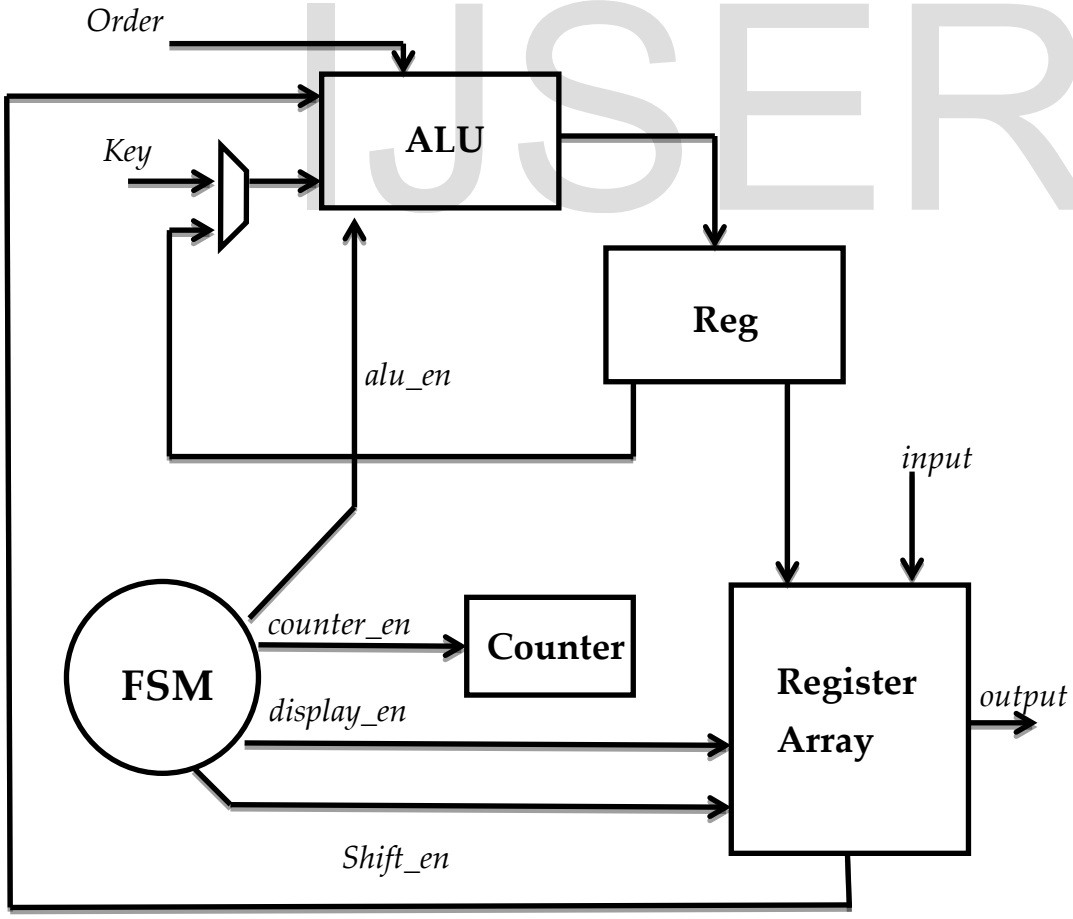


**Fig 1: Proposed architecture**

The hardware is able to encrypt N inputs at a time. The input signal will be mapped using the first order and the corresponding key. The resultant output is then mapped using the second Order and the second key. An FSM controls the operation of the system. Initial state is to input the data to the Register Array. When the N data are loaded into the Register Array, ALU wil start encrypting the data, The encrypted data is loaded into the shift register array. The register array will shift during each clock cycle. The output of the AL U is loaded into the LSB of the Register Array. The MSB of the Register Array is moved as the input to the ALU for calculating the next encrypted data. A counter is used to check the end of the encryption procedure and the data is taken out of the register array.

## 3 SECURITY

The level of encryption in this proposed method depends on the user. It is difficult to break this method of encryption without the prior idea of the total number of order and key used. Also it is difficult to predict the quasigroup used even if the order is known because of the higher robustness due to exponential relationship between the order and the number of quasigroup.

$$n!(n-1)!T(n) \qquad (3)$$

Where n is the order of the quasigroup and T(n) is the reduced Latin square. Figure 2 shows the number of reduced Latin square for different values of 'n'. The key should be a value less than or equal to the order used. The proposed architecture has the flexibilty to convert the key given by the user to a value between 0 and the order of the quasigroup. This enhances the security of the system.

| n | T(n) |
|---|---|
| **1** | 1 |
| **2** | 1 |
| **3** | 1 |
| **4** | 4 |
| **5** | 56 |
| **6** | 9,408 |
| **7** | 16,942,080 |
| **8** | 535,281,401,856 |

**Fig 2: Reduced Latin square**

## 4 RESULTS

The functionality of Quasigroup based encryption is verified using the Altera Cyclone II FPGA kit. The proposed hardware is compared with quasigroup based encryption which uses memory to store the quasigroup. It is proved that the proposed hardware uses less memory. The Altera Quartus II Synthesis report is shown in table 2:

**Table 2: Altera Quartus Synthesis Report**

| Parameters | With memory (4-bit input) | Without memory(prop -osed design) |
|---|---|---|
| Total logic elements | 41416 | 259 |
| Total registers | 164 | 87 |
| Total LAB | 2590/6839 | 21/6839 |
| I/O pins | 27/508 | 27/508 |

## 5 CONCLUSION

The proposed hardware model for the quasigroup based encryption is implemented in Verilog. The RTL consists of three modules for the FSM, ALU unit and Shift Register unit. The Verilog code is synthesized and implemented in Cyclone II Altera FPGA kit using the Quartus II tool. Quasigroup based block cipher encryption maximizes the entropy of the output. The level of encryption depends on the number of Order and key given by the user. The proposed design was implemented on a Cyclone FPGA kit and the functionality was verified. It is proposed to enhance the architecture to improve the throughput of the system and to incorporate reconfigurability to accommodate different quasigroup encryption schemes.

## REFERENCES

[1] Battey, Matthew, and AbhishekParakh. "Efficient quasigroup block cipher for sensor networks." In Computer Communications and Networks (ICCCN), 2012 21st International Conference on, pp. 1-5. IEEE, 2012.

[2] Battey, Matthew, and AbhishekParakh. "A Quasigroup Based Random Number Generator for Resource Constrained Environments." IACR Cryptology ePrint Archive 2012 (2012): 471.

[3] Dvorsky, J., EliškaOchodková, VáclavSnásel, and Ajith Abraham. "Large quasigroups in cryptography and their properties testing."In Nature & Biologically Inspired Computing, 2009.NaBIC 2009. World Congress on, pp. 965-971. IEEE, 2009.

[4] Koscielny, C. Z. E. S. Ł. A. W. "Generating quasigroups for cryptographic applications." International Journal of Applied Mathematics and Computer Science 12, no. 4 (2002): 559-570.

[5] Ochodkova, Eliska, J. Dvorsky, VáclavSnásel, and Ajith Abraham. "Testing the properties of large quasigroups."In Ultra Modern Telecommunications & Workshops, 2009.ICUMT'09. International Conference on, pp. 1-7. IEEE, 2009.

[6] Pal, S. K. "Development of Efficient Algorithms for Quasigroup Generation & Encryption." In Advance Computing Conference, 2009.IACC 2009. IEEE International, pp. 940-945. IEEE, 2009.

[7] Ritter, Terry. "Ritter's Crypto Glossary and Dictionary of Technical Cryptography." (2006).

[8] Satti, Maruti, and SubhashKak. "Multilevel indexed quasigroup encryption for data and speech." Broadcasting, IEEE Transactions on 55, no. 2 (2009): 270-281.

[9] shcherbacov, V. A. "Quasigroups in cryptology." Computer Science 17, no. 2 (2009): 50.

[10] Gligoroski, Danilo, Smile Markovski, and S. Markovski. "Cryptographic potentials of quasigroup transformations." (2003).

[11] Deriyenko, Andriy I., Ivan I. Deriyenko, and Wieslaw A. Dudek. "Rigid and super rigid quasigroups." Quasigroups Relat. systems 17, no. 1 (2009): 17-28.

[12] Dimitrova, V., and J. Markovski. "On quasigroup pseudo random sequence generators." In proceedings of the 1st Balkan Conference in Informatics,Thessaloniki, Greece, November, pp. 21-23