Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 27 (2024), No. 8, pp. 2337–2364
DOI : 10.47974/JDMSC-1917

Symmetric encryption scheme based on quasigroup using chained mode of operation

Satish Kumar *
Department of Mathematical Sciences
Indian Institute of Technology (BHU)
Varanasi 221005
Uttar Pradesh
India

Harshdeep Singh †
Indivar Gupta §
Defence R & D Organization
Metcalfe House
Delhi 110054
India

Ashok Ji Gupta ‡
Department of Mathematical Sciences
Indian Institute of Technology (BHU)
Varanasi 221005
Uttar Pradesh
India

Abstract

In this paper, we propose a novel construction for a symmetric encryption scheme, referred as SEBQ which is built on the structure of quasigroup. Utilizing the concepts of chaining like mode of operation, SEBQ is a block cipher with various in-built properties. This scheme is proven to be resistant against chosen plaintext attack (CPA) and on applying unbalanced Feistel transformation [18], it achieves security against chosen ciphertext attacks (CCA). Subsequently, assessment has been conducted for the randomness of this scheme by

* *E-mail:* satish.kumar.rs.mat18@itbhu.ac.in (Corresponding Author)
† *E-mail:* harshdeep.sag@gov.in
§ *E-mail:* indivar_gupta@yahoo.com
‡ *E-mail:* agupta.apm@itbhu.ac.in

running the NIST Statistical test suite be analysing the impact of change in the secret key and plaintext on ciphertext through an avalanche effect analysis. Thereafter, a comparative analysis has been done referring the results with existing schemes based on quasigroups [8,40]. Moreover, we also analyse the computational complexity in terms of number of operations needed for encryption and decryption.

Subject Classification: *(2010) 20N05, 05B15, 94A60, 68W20.*

Keywords: *Quasigroups, Symmetric encryption scheme, Mode of operation, IND-CPA, IND-CCA2.*

1. Introduction

In the nineteenth century, Euler [20] introduced a new theory that delved into the concept of those Latin squares which are pair-wise orthogonal, termed as MOLS (mutually orthogonal Latin squares). This concept bears a close connection to the structure of quasigroups using combinatorics. In 1935, Moufang [34] first introduced the term 'quasigroup' and later referred to the term 'loop' as a quasigroup possessing a unity element. The versatility of quasigroups, with their numerous structures, properties and existence of quasigroups of specific order, enables their application in diverse theories, like coding theory, cryptography, telecommunications etc., for details readers may kindly refer [9, 16, 26, 38]. Dénes and Keedwell, in their work of 1979, 1991 and 2001 [15–17], have conducted an extensive research on applications of quasigroups in diverse fields like cryptology and coding theory.

Quasigroup is an algebraic structure that is almost similar to a group, with the difference as the operation in former need not be associative. Hence, we can say every group is a quasigroup not other way around. It is well known that the properties such as closure and inversion of elements in an algebraic structure plays compelling role in the design of various cryptographic primitives. Groups, Rings and Finite fields which are types of associative algebraic structures have been widely employed in the creation of diverse cryptographic primitives [1, 24, 25, 33], as well as in the development of algebraic codes [41, 42] that can detect and correct errors simultaneously. Similarly, the security of a cryptographic Hash function NaSHA [32], designed by Markovski et al. in 2008, relied upon factors including the abundance of quasigroup with the same order, the utilization of secret quasigroup operations, large number of isotopes etc. Similarly, the Edon-R encryption scheme, developed by Gligoroski et al. in 2009, [21] anchors its security on the intricacies of several factors like solving the systems of quasigroup operations and Multivariate Quadratic (MQ)

equations and finding the order of elements in a given quasigroup. Additionally, several public key cryptographic primitives [23, 27, 38] have been proposed by utilizing the structure of quasigroups. Quasigroups find wide applications in the design of various other cryptographic primitives like block ciphers [4, 30, 40], stream ciphers [31, 35], lightweight block ciphers [4, 8, 40] and numerous other areas. For detailed survey on the application of quasigroups in cryptography, readers are encouraged to refer the detailed survey articles [27, 38].

Similarly, the orthogonal properties of Latin squares and quasigroups render them suitable for application in coding theory [10–13, 26, 28]. In 2007, Gligoroski et al. [22] proposed an error correcting code based on quasigroup transformations which are non-linear and almost random. Additionally, they compare their code with Reed-Muller (RM) code of rate $3/16$ that can recover up to 7 errors in 32 bits by showing that the proposed non-linear code can correct even 5 errors in 16 bits which is much better than Reed-Muller code.

This paper focuses on the development of a symmetric encryption scheme based on quasigroup (SEBQ) using a chaining-like mode of operation. Before delving deeper into this topic, we will explore various security aspects related to symmetric encryption schemes. The concept of provable security for public key cryptosystems, proposed by Golwasser and Micali in 1982 [36], marked a significant advancement. They demonstrated that the security of such systems could be polynomial-time reducible from quadratic residuosity. Subsequently, in 1997, Bellare et al. [5] adapted this concept to the symmetric setting. Prior to this, Shannon introduced the concept of perfect secrecy for cryptosystems, well explained by Stinson [39]. In a perfectly secure cryptosystem, for any two distinct plaintexts P_1, P_2, and for given legitimate ciphertext C corresponding to any one of plaintexts, the probability of C being the result of encrypting either P_1 or P_2 is equally likely.

1.1 *Security notions for symmetric encryption scheme*

For the context of this paper, primarily we discuss two notions of security: chosen-plaintext attack (CPA) and chosen ciphertext attack (CCA). To understand these security of notions, we mainly focus on conceptual framework discussed by Bellare et al. in [5], detailed in [25].

Consider a symmetric encryption scheme $SE = \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ where $\mathcal{K}$, $\mathcal{E}$ and $\mathcal{D}$ denoted the key generation, encryption and decryption algorithms respectively.

1.1.1 Indistinguishability under chosen-plaintext attack (IND-CPA) and chosen-ciphertext attack (IND-CCA) security

Definition 1.1: [25] Suppose $\mathcal{A}_{cpa}$ represents an adversary that can access to encryption oracle $\mathcal{E}_K(\cdot)$ only and $\mathcal{A}_{cca}$ is an adversary that can access both encryption $\mathcal{E}_K(\cdot)$ and decryption $\mathcal{D}_K(\cdot)$ oracles. Now we define the following experiments: for $b \in \{0,1\}$ and $k \in \mathbb{N}$

Algorithm 1: Experiment $\mathbf{Exp}^{ind-cpa-b}_{SE,\mathcal{A}_{cpa}}(k)$

$K \xleftarrow{R} \mathcal{K}(k)$

$(x_0, x_1) \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}_{cpa}(k)$

$c \leftarrow \mathcal{E}_K(x_b)$

$b' \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot)}_{cpa}$

Return b'

Algorithm 2: Experiment $Exp^{ind-cca-b}_{SE,\mathcal{A}_{ccawhitep}}(k)$

$K \xleftarrow{R} \mathcal{K}(k)$

$(x_0, x_1) \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{cca}(k)$

$c \leftarrow \mathcal{E}_K(x_b)$

$b' \leftarrow \mathcal{A}^{\mathcal{E}_K(\cdot),\mathcal{D}_K(\cdot)}_{cca}$

Return b'

Note that $|x_0| = |x_1|$ (i.e. x_0 and x_1 both have equal lengths) and $\mathcal{A}_{cca}$ cannot query $\mathcal{D}_K(\mathcal{E}_K(\cdot))$ The advantage of the adversaries $\mathcal{A}_{cpa}$ and $\mathcal{A}_{cpa}$ can be defined as:

$$\begin{aligned}
\mathrm{Adv}^{ind-cpa}_{SE,\mathcal{A}_{cpa}}(k) &= Pr[\mathrm{Exp}^{ind-cpa-1}_{SE,\mathcal{A}_{cpa}} = 1] - Pr[\mathrm{Exp}^{ind-cpa-0}_{SE,\mathcal{A}_{cpa}} = 1] \\
\mathrm{Adv}^{ind-cca}_{SE,\mathcal{A}_{cca}}(k) &= Pr[\mathrm{Exp}^{ind-cca-1}_{SE,\mathcal{A}_{cca}} = 1] - Pr[\mathrm{Exp}^{ind-cca-0}_{SE,\mathcal{A}_{cca}} = 1]
\end{aligned}$$

Now the advantage functions for the symmetric encryption scheme SE can be defined as: For any $t, q_e, \mu_e, q_d, \mu_d \in \mathbb{Z}^+$,

$$\mathrm{Adv}_{SE}^{ind-cpa}(k,t,q_e,\mu_e) = \max_{\mathcal{A}_{cpa}}\{\mathrm{Adv}_{SE}^{ind-cpa}(k)\}$$
$$\mathrm{Adv}_{SE}^{ind-cca}(k,t,q_e,\mu_e,q_d,\mu_d) = \max_{\mathcal{A}_{cca}}\{\mathrm{Adv}_{SE}^{ind-cca}(k)\}$$

The maximum is taken from all potential adversaries $\mathcal{A}_{cpa}$ and $\mathcal{A}_{cca}$ with time complexity t. Both adversaries are constrained to make at most q_e queries to the oracle $\mathcal{E}_K(\cdot)$ and resulting in μ_e ciphertexts that both adversaries observe in response to these encryption oracle queries. Additionally, in case of adversary $\mathcal{A}_{cca}$, can make at most q_d queries to the oracle $\mathcal{D}_K(\cdot)$, and as result μ_d amount of plaintext they see in response to decryption oracle queries. We say that the symmetric scheme SE is IND-CPA secure (resp. IND-CCA secure), if the advantage function $\mathrm{Adv}_{SE}^{ind-cpa}(\cdot)$ (resp. $\mathrm{Adv}_{SE}^{ind-cca}(\cdot)$) is negligible for legitimate adversaries $\mathcal{A}$ with time complexity polynomial in security parameter k.

1.2 *Main contribution of the paper*

We propose a novel construction of symmetric encryption scheme SEBQ based on quasigroup using chaining-like mode of operation. The transformed initial vector is being utilized to encrypt the subsequent message block. We will prove that SEBQ scheme is IND-CPA secure and after applying *unbalanced Feistel transformation*, we establish its security strength to IND-CCA2 secure. We assess the randomness of SEBQ scheme by running the NIST-STS test suite and compare the results with the existing schemes like INRU [40], BCWST [8], AES-128. Also, we analyze the avalanche effect concerning the secret key and plaintext within the SEBQ scheme. Finally we perform the computational analysis of SEBQ focusing on the complexity in terms of number of operations required.

Organization of the paper

This article is structured into several sections: First, in Section 2 we give the preliminaries of quasigroups that are required to understand this paper and the string transformations based on structure of quasigroups. Section 3 is dedicated to the construction of SEBQ using chained like mode of operations in which instead of ciphertext transformed random vector will be passed to next block for the encryption of succeeding blocks of plaintext. Section 4 is committed for the security analysis of SEBQ scheme, in which we prove main results of this paper and we also do the randomness testing of the SEBQ scheme. In Section 5, we find out the computational

complexity of the SEBQ scheme. Finally, In Section 6, we draw the conclusions of the paper.

2. Preliminaries

This section contains a concise overview related to the theory of quasigroups, the definition of symmetric encryption scheme and the string transformations based on quasigroup. For detailed theory of quasigroup and the symmetric encryption schemes, readers may refer [5, 14, 38, 39].

Consider a finite set Q and a binary operation $*: Q^2 \to Q$ possessing the following property: for all $a, b \in Q$ there exist unique $x, y \in Q$ satisfying the equations $a * x = b$ and $y * a = b$. Then, $(Q, *)$ is known as quasigroup.

For a given quasigroup $(Q, *)$, the parastrophe or adjoint binary operation '$\backslash$' can be derived from the given binary operation '$*$' by utilizing the following equation:

$$x * y = z \Leftrightarrow y = x \backslash z \tag{1}$$

The algebra $(Q, *, \backslash)$ satisfies the following identities:

$$x \backslash (x * y) = y \text{ and } x * (x \backslash y) = y,$$

and $(Q, \backslash)$ is also a quasigroup.

Definition 2.1: [26] A Latin square is an array of size $n \times n$ filled with n distinct symbols, ensuring that each symbol appears precisely once in each row and once in each column.

Every quasigroup $(Q = \{x_1, x_2, \ldots, x_n\}, *)$ can be represented by a Latin square L of size n^2. For example, for the set $Q = \{1, 2, 3, 4, 5\}$, a Latin square and its corresponding parastrophe's Latin square are represented in Table 1:

Table 1

Latin squares corresponding to quasigroup $(Q, *)$ and its parastrophes $(Q, \backslash)$

$*$	1	2	3	4	5
1	1	2	3	4	5
2	2	1	4	5	3
3	3	5	1	2	4
4	4	3	5	1	2
5	5	4	2	3	1

$\backslash$	1	2	3	4	5
1	1	2	3	4	5
2	2	1	5	3	4
3	3	4	1	5	2
4	4	5	2	1	3
5	5	3	4	2	1

Definition 2.2: [29] Consider a matrix $A=(a_{i,j})$ of order $n\times n$. Then, the permanent of A is defined as:

$$perm(A)=\sum_{\tau\in S_n}\prod_{j=1}^{n}a_{j,\tau(j)}.$$

The summation is running for all τ belongs to symmetric group S_n

The cardinality of Latin square plays a significant role in theoretical security of quasigroup-based cryptographic protocols. However, calculating the number of Latin square [37] poses a challenge because there is exponential growth in the number of terms involved. Consequently, extensive research efforts have been dedicated to estimating the number of Latin square for specific order by employing methods like volunteer counting or utilized specialized software like Singular / SageMath [2].

In [37], Shao and Wei determined a formula to calculate the number of Latin squares. Suppose $\tau_0(A)$ represents the total number of zeroes in matrix A and $perm(A)$ is the permanent of A defined as (2.2), the respective formula is described below:

Theorem 2.3: *Let* $B_n=\{(b_{i,j})_{n\times n}\mid b_{i,j}\in\{0,1\}\}$ *be the set of all* $n\times n$ *sized matrices. Then number of Latin squares is:*

$$L(n)=n!\sum_{A\in B_n}(-1)^{\tau_0(A)}\binom{perm(A)}{n} \tag{2}$$

Above formulae (2) are not easy for computation, however Dénes and Keedwell [26] used them to compute bounds on number of Latin squares, i.e.

$$\prod_{j=1}^{n}(j!)^{n/j}\geq L(n)\geq\frac{(n!)^{2n}}{n^{n^2}}. \tag{3}$$

For specific values of $n=2^l$, where $l=7,8$, the approximate number of $L(n)$ is estimated as:

$$0.164\times10^{21091}\geq L(128)\geq0.337\times10^{20666}$$
$$0.753\times10^{102805}\geq L(256)\geq0.304\times10^{101724}$$

We give a Table 2 contains exact values of Latin square $L(n)$ with n-number of symbols.

Table 2
Number of Latin squares of size *n*.

n	**Number of Latin squares of size *n***
1	1
2	2
3	12
4	576
5	161, 280
6	812, 851, 200
7	61, 479, 419, 904, 000
8	108, 776, 032, 459, 082, 956, 800

In order to set-up an encryption scheme, we make use of the quasigroup $(Q,*)$ based transformations on $Q^l \times Q^n$. This is formally defined below:

Definition 2.4: [22] **($\mathfrak{e}$-transformation)** Consider a quasigroup $(Q,*)$ $(\mathbb{F}_2^k,*)$. For a given $b \in Q$, we define $F_b : Q \to Q$ as $F_b(a) = b * a$. Also, we define $f : Q^n \to Q^n$ as $f(b_1 b_2 \ldots b_n) = d_1 d_2 \ldots d_n$, where each $d_i = b_i$ for $i = \overline{1, n-1}$ and $d_n = b_1 + b_2 + \cdots + b_n$ (where $\overline{1, n-1}$ represents integers ranging from 1 to $n-1$). Using these functions, we define the following:

(i) For a given $\beta = b_1 b_2 \ldots b_n \in Q^n$, we define a map $F_\beta : Q \to Q$ as $F_\beta = F_{b_n} \circ F_{b_{n-1}} \circ \ldots \circ F_{b_1}$.

(ii) For a given $a \in Q$, we define a map $f_a^n : Q^n \to Q^n$ as $f_a^n(b_1 b_2 \ldots b_n) = d_1 d_2 \ldots d_n$, where each component maps differently as $d_i = F_{b_i} \circ F_{b_{i-1}} \circ \cdots \circ F_{b_1}(a)$, $i = \overline{1, n}$; additionally, we define $F_a^n : Q^n \to Q^n$ as $F_a^n = f \circ f_a^n$.

(iii) Finally, we define $\mathfrak{e}$-transformation $F^{l,n} : Q^l \times Q^n \to Q^l \times Q^n$ as $F^{l,n}(a_1 a_2 \ldots a_l, b_1 b_2 \ldots b_n) = (c_1 c_2 \ldots c_l, d_1 d_2 \ldots d_n)$, where each component is computed as:

$$c_i = F_{\delta_{i-1}}(a_i), \delta_i = F_{a_i}^n(\delta_{i-1}) \text{ for } i = \overline{1, n};$$

assuming $\delta_0 = b_1 b_2 \ldots b_n$, and $d_1 d_2 \ldots d_n = \delta_l$.

Definiton 2.5: [22] **($\mathfrak{d}$-transformation)** Consider a quasigroup $(Q,*) = (\mathbb{F}_2^k,*)$ and its corresponding parastrophe $\backslash$. For a given $b \in Q$,

we define $G_b : Q \to Q$ as $G_b(c) = b \setminus c$. Also, we define $f : Q^n \to Q^n$ as $f(b_1 b_2 \ldots b_n) = d_1 d_2 \ldots d_m$ where each $d_i = b_i$ for $i = \overline{1, n-1}$ and $d_n = b_1 + b_2 + \ldots + b_n$. Using these functions, we define the following:

(i) For a given $\beta = b_1 b_2 \ldots b_n \in Q^n$, we define a map $G_\beta : Q \to Q$ as $G_\beta = G_{b_1} \circ G_{b_2} \circ \ldots \circ G_{b_n}$.

(ii) For a given $c \in Q$, we define a map $g_c^n : Q^n \to Q^n$ as $g_c^n(b_1 b_2 \ldots b_n) = d_1 d_2 \ldots d_m$ and each component map differently as $d_{n-i+1} = G_{b_n} \circ G_{b_{n-1}} \circ \ldots \circ G_{b_{n-i+1}}(c)$, $i = \overline{1, n}$; additionally, we define $G_c^n : Q^n \to Q^n$ as $G_c^n = f \circ g_c^n$.

(iii Finally, we define $\mathfrak{d}$-transformation $G^{l,n} : Q^l \times Q^n \to Q^l \times Q^n$ as

$G^{l,n}(c_1 c_2 \ldots c_l, b_1 b_2 \ldots b_n) = (a_1 a_2 \ldots a_l, d_1 d_2 \ldots d_n)$ where each component computed as:

$$a_i = G_{\delta_{i-1}}(c_i), \delta_i = G_{c_i}^n(\delta_{i-1})$$

assuming $\delta_0 = b_1 b_2 \ldots b_n$ and $d_1 d_2 \ldots d_n = \delta_l$.

Theorem 2.6: [22] *From the functions defined in Definition 2.4 and Definition 2.5, we have the following:*

(i) $G_\beta^l(F_\beta^l(\alpha)) = \alpha$ and $F_\beta^l(G_\beta^l(\gamma)) = \gamma$,

(ii) F_β^l and G_β^l are permutations of Q^l.

Symmetric encryption scheme: In this paper, our objective is to construct a symmetric encryption scheme $SE = \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ that comprises of three algorithms: Key Generation Algorithm, Encryption Algorithm and Decryption Algorithm.

(i) *Key Generation*: In Key Generation Algorithm $\mathcal{K}$, it accepts a security parameter $k \in \mathbb{N}$ as input and returns a key secret key K as output, denoted as $K \xleftarrow{R} \mathcal{K}(k)$.

(ii) *Encryption algorithm*: The Encryption Algorithm $\mathcal{E}_K : Q \to Q$ can be either randomized or stateful. When instantiated with the key K, it

takes a plaintext $M \in Q^l$ as input and returns a ciphertext C, denoted as $C \xleftarrow{R} \mathcal{E}_K(M)$.

(iii) *Decryption algorithm*: The decryption algorithm $\mathcal{D}_K$ is both deterministic and stateless. It initializes with a key K and processes a string C to return the corresponding plaintext $x \in Q^l$, denoted as $x \leftarrow \mathcal{D}_K(C)$. It is important to ensure that the design of $\mathcal{E}$ and $\mathcal{D}$ guarantees that $\mathcal{D}_K(\mathcal{E}_K(x)) = x$ holds for all $x \in Q^l$.

3. Symmetric encryption scheme based on quasigroup (SEBQ)

In this section, we design a symmetric encryption scheme based on the $\mathfrak{e}$ and $\mathfrak{d}$ -transformations using Definition 2.4 and 2.5 of the quasigroup by utilizing chaining-like mode of operation. We design a block cipher of size k-bit long blocks and the operation of quasigroup $*$ as secret of our encryption scheme.

Consider a set $Q = \mathbb{F}_2^k$ for some positive integer k. Now, $a \in Q$ implies a can be written as k-tuple of elements from set $\{0,1\}$. In the SEBQ scheme, three algorithms have been put forth, including *Key Generation algorithm*, an *Encryption algorithm* and the *Decryption algorithm*. The security of SEBQ depends on the number of quasigroups of particular order.

3.1 *Key Generation*

Generate a random Latin square of size $2^k \times 2^k$ by giving an input a parameter $k \in \mathbb{N}$, utilizing an algorithm proposed by Artamonov et al. [2]. The generated Latin square is being treated as a secret key for SEBQ scheme and the dimension of the key space is $|L(n)|$ for $n = 2^k$ where as $|L(n)|$ can be calculated by utilizing (2). To compute the parastrophe of the corresponding Latin square, we utilize (1), which states that the knowledge of $(Q, *)$ or combinatorial equivalent Latin square is equivalent to the knowledge of $(Q, \backslash)$.

3.2 *Encryption Process*

Consider $M = (m_1, m_2, \ldots, m_l) \in (\mathbb{F}_2^k)^l$, where each $m_i \in \mathbb{F}_2^k$, i.e. each block m_i is k-bit long and M is kl-bit long plaintext. In order to begin the encryption process, user needs to consider an initial vector $R^{(1)} = r_1^{(1)}, r_2^{(1)}, \ldots, r_n^{(1)}$, i.e. a kn-bit long string, where each $r_i^{(1)} \in Q$.

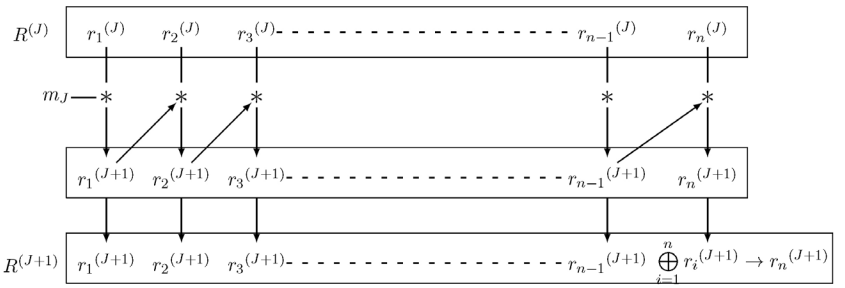

Figure 1

Transformation of initial vector for encryption function

Now, the encryption scheme proceeds by making use of $\mathfrak{e}$ -transformation as defined in Definition 2.4. Define the intermediate map $E_* : Q \times Q^n \to Q \times Q^n$ as

$$E_*(m,(r_1,r_2,\ldots,r_n)) = (r_n * (r_{n-1} * (\ldots * (r_1 * m)\cdots)),(r_{1'},r_{2'},\ldots,r_{n'})). \tag{4}$$

Here, $(r_1',r_2',\ldots,r_n') = F_m^n(r_1,r_2,\ldots,r_n)$. For the pictorial representation, refer Figure 1.

The encryption process runs sequentially by first applying E_* on m_1 using the initial vector. Subsequently, the initial vector gets updated and using this together with m_2, the E_* results in second block of ciphertext. Similarly, the initial vector is updated and utilized to encrypt m_i till the complete message is encrypted. We say that the ciphertext corresponding to M is given by the vector $(E_*(m_1),E_*(m_2),\ldots,E_*(m_l))$, where by $E_*(m_i)$ we consider only the first component. This process is explained in Algorithm 3.

To encrypt the i^{th} block of message M we need two inputs, i.e. a message block m_i and a random vector $R^{(i)}$, after encrypting the message block m_i we get output $(c_i,R^{(i+1)})$. For pictorial representation and algorithm of encryption scheme refer Figure 2 and Algorithm 3 respectively.

Here, the encryption function, i.e. a function from $Q^l \to Q^l$, can be decomposed into l maps E_* each from $Q \times Q^n \to Q \times Q^n$ and same structure follows for the decryption function as well operation $*$ being replaced with $\backslash$.

Algorithm 3 Encryption algorithm

Require: An $n \times n$ sized Latin square $*$ over Q, an initial (or, leader) random vector $R = (r_1, r_2, \ldots, r_n) \in Q^n$, and a message $M = (m_1, m_2, \ldots, m_l) \in Q^l = \mathbb{F}_2^{kl}$.

1: **for** $j = 1$ to l **do**
2: $\quad k_{0,j} \leftarrow m_j$ ▷ Assign $(k_{0,1}, k_{0,2}, \ldots, k_{0,l}) = (m_1, m_2, \ldots, m_l)$
3: **end for**
4: **for** $i = 1$ to n **do**
5: $\quad k_{i,0} \leftarrow r_i$ ▷ Assign $(k_{1,0}, k_{2,0}, \ldots, k_{n,0}) = (r_1, r_2, \ldots, r_n)$
6: **end for**
7: **for** $j = 1$ to l **do**
8: $\quad$ **for** $i = 1$ to n **do**
9: $\quad\quad k_{i,j} \leftarrow k_{i,j-1} * k_{i-1,j}$ ▷ Compute $k_{1,j}, k_{2,j}, \ldots, k_{n,j}$
10: $\quad$ **end for**
11: $\quad c_j \leftarrow k_{n,j}$ ▷ Compute the ciphertext block c_j as $k_{n,j}$
12: $\quad k_{n,j} \leftarrow k_{1,j} \oplus k_{2,j} \oplus \cdots \oplus k_{n,j}$ ▷ Update the value of $k_{n,j}$
13: **end for**

Ensure: $C = (c_1, c_2, \ldots, c_l) \in Q^l = \mathbb{F}_2^{kl}$

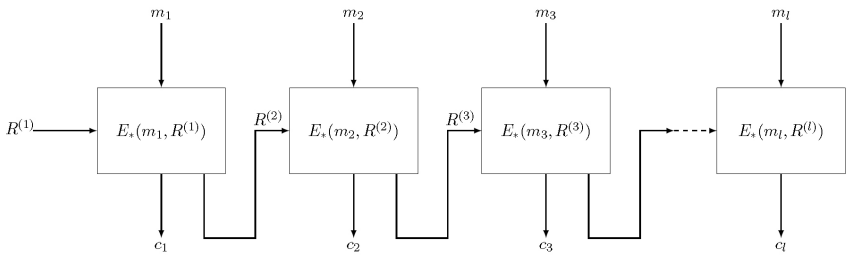

Figure 2

Encryption algorithm

3.3 *Decryption Process*

The decryption process takes input a ciphertext $C = (c_1, c_2, \ldots, c_l) \in \mathbb{F}_2^{kl}$ along with the initial vector $R^{(1)} = r_1^{(1)}, r_2^{(1)}, \ldots, r_n^{(1)} \in Q^n$. The algorithm requires the multiplication table of the parastrophe $\backslash$ of quasigroup $(Q, *)$ which acts as the secret key.

For the decryption scheme, we use $\mathfrak{d}$-transformation of quasigroup using Definition 2.5. Define a map $D_\backslash : Q \times Q^n \to Q \times Q^n$ as $D_\backslash(c, (s_1, s_2, \ldots, s_n)) = (r_1 \backslash (r_2 \backslash (\ldots \backslash (r_n \backslash c) \ldots)), (s_1', s_2', \ldots, s_n'))$. Here, $(s_1', s_2', \ldots, s_n') = G_c^n(s_1, s_2, \ldots, s_n)$.

Using same definition, transform the vector S after the decryption of each block of ciphertext given the initial vector $S^{(1)} = R^{(1)} = (r_1^{(1)}, r_2^{(1)}, \ldots, r_n^{(1)})$.

Algorithm 4 Decryption algorithm

Require: An $n \times n$ sized multiplication table of parastrophe $\backslash$ of Latin square $*$ over Q, an initial (or, leader) random vector $R = (r_1, r_2, \ldots, r_n) \in Q^n$, and a ciphertext $C = (c_1, c_2, \ldots, c_l) \in Q^l = \mathbb{F}_2^{kl}$.

1: **for** $i = 1$ to n **do**
2: $k_{i,0} \leftarrow r_i$ ▷ Assign $(k_{1,0}, k_{2,0}, \ldots, k_{n,0}) = (r_1, r_2, \ldots, r_n)$
3: **end for**
4: **for** $j = 1$ to l **do**
5: $k_{n,j} \leftarrow c_j$ ▷ Assign $(k_{n,1}, k_{n,2}, \ldots, k_{n,l}) = (c_1, c_2, \ldots, c_l)$
6: **end for**
7: **for** $j = 1$ to l **do**
8: **for** $i = n$ down to 2 **do**
9: $k_{i-1,j} \leftarrow k_{i,j-1} \backslash k_{i,j}$ ▷ Compute $k_{n-1,j}, k_{n-2,j}, \ldots, k_{1,j}$
10: **end for**
11: $m_j = k_{1,j-1} \backslash k_{1,j}$ ▷ Compute the message block $m_j = k_{1,j-1} \backslash k_{1,j}$
12: $k_{n,j} = k_{1,j} \oplus k_{2,j} \oplus \cdots \oplus k_{n,j}$ ▷ Update the value of $k_{n,j}$
13: **end for**

Ensure: $M = (m_1, m_2, \ldots, m_l) \in Q^l = \mathbb{F}_2^{kl}$.

Similar to encryption function, here the decryption function, i.e. a function from $Q^l \to Q^l$ can be decomposed into l maps $D_\backslash$ each from $Q \times Q^n \to Q \times Q^n$ and to decrypt the i^{th} block of ciphertext C we compute $D_\backslash(c_i, S^i) = (m_i, S^{i+1})$. For pictorial representation of the decryption scheme, one can refer to Figure 3.

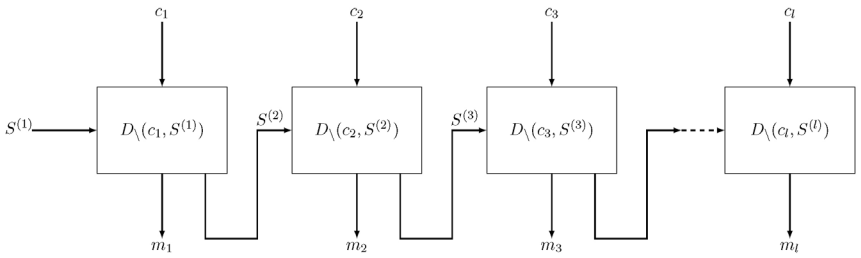

Figure 3

Decryption algorithm

4. Security Analysis of the Proposed Cryptosystem SEBQ

This section contains the security analysis of the proposed scheme SEBQ. For the security analysis, we apply the conceptual framework of Bellare et al. [5] for IND-CPA and IND-CCA security assessments. We will show that the SEBQ is IND-CPA secure and subsequently, through the application of an unbalanced Feistel transformation, we establish its IND-CCA2 security.

Theorem 4.1: *The symmetric encryption scheme SEBQ as defined in Section 3 based on quasigroup is immune to chosen-plaintext attacks. In other words, any adversary having access to encryption oracle, has negligible advantage in distinguishing correct from his chosen two plaintexts, corresponding to any arbitrary ciphertext.*

Proof: In order to prove that the symmetric encryption SEBQ scheme is secure against CPA attack heuristically, first, we consider an experiment (Exp) conducted by adversary $\mathcal{A}$ which generates a key, two plaintexts and a ciphertext of randomly selected one plaintext out of those. The experiment outputs 1 if $\mathcal{A}$ is able to identify the correct plaintext. Algorithmically, it is explained below:

Algoritm 5 $\mathrm{Exp}^{cpa}_{\mathcal{A},SEBQ}$(security parameter)

1. The adversary $\mathcal{A}$ is given input 1^* and has access to the oracle $\mathcal{E}_K(\cdot)$. It then produces a pair of messages m_0, m_1 of the equal length.
2. A random bit $b \in \{0,1\}$ is selected, and subsequently, a ciphertext $c \leftarrow \mathcal{E}_K(m_b)$ is generated and given to adversary $\mathcal{A}$.
3. The adversary $\mathcal{A}$ continues to have the oracle access $\mathcal{E}_K(\cdot)$, and following a series of computations, produces an outputs bit b'.
4. The output of the experiment is 1 if $b' = b$, and 0 otherwise. In the former case, we say that the adversary $\mathcal{A}$ succeeds.

Suppose adversary chooses plaintexts m_0 and m_1 from Q and an initial vector $R \in Q$; and gets a ciphertext C of one of the plaintexts. We consider the cases where adversary can query repeated or non-repeated messages to the oracle.

- **Adversary queries the repeated messages with different initial vectors (IVs)**: Since, the encryption scheme SEBQ is probabilistic (not deterministic) and adversary $\mathcal{A}$ queries the repeated message as many times as it wants. Suppose adversary $\mathcal{A}$ repeatedly queries $m_0 \in Q$ with initial vectors R, with $R' \neq R$ to oracle $\mathcal{E}_K(\cdot)$. Then, the adversary can compute the complete column of Latin square. Eventually, $\mathrm{Exp}^{cpa}_{\mathcal{A},SEBQ} = 1$, i.e. Adversary is able to distinguish the plaintext given its ciphertext with probability 1. Thus, the encryption scheme is insecure. However, this kind of liberty is not given to adversary. So we consider the other case only.
- **Adversary is not allowed to query the repeated messages**: The main idea for adversary to be successful is to construct the Latin square

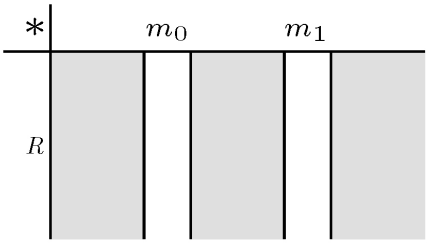

Figure 4

Construction of Latin square during IND-CPA attack

using the adaptive interaction with the $\mathcal{E}_K(\cdot)$ oracle. Adversary queries m_i, $i \notin \{0,1\}$ to the oracle and get different ciphertexts. This results in detection of whole Latin square except two columns corresponding to m_0 and m_1 as given in Figure 4. However, even with this knowledge adversary is unable to identify whether the given ciphertext corresponds to which message m_0 or m_1 as the R-th row can have C in either column with equal probability, i.e. $Pr[b=0] = Pr[b=1] = \frac{1}{2}$, which implies $\mathrm{Adv}_{\mathcal{A}}^{ind-cpa} = 0$.

Proposition 4.2: The SEBQ encryption scheme in not IND-CCA2 secure.

Proof: Suppose adversary $\mathcal{A}$ has access to both encryption $\mathcal{E}_K(\cdot)$ and decryption $\mathcal{D}_K(\cdot)$ oracle. Upon querying $\mathcal{D}_K(\cdot)$ for $C' \neq C$, the complete Latin square can be computed expect the places where C is present. Using the Definition 2.1 of Latin square, those left out places can be easily filled. Thus, $\mathrm{Adv}_{SEBQ}^{ind-cca}(\mathcal{A}) = 1$, hence the proposition follows.

Similar to the above, it can be readily observed that the same result holds for Left-or-Right indistinguishability [25]. Thus, we aim on applying some transformations to modify the scheme to get indistinguishability under CCA2 attacks.

4.1 *Unbalanced Feistel transformation of SEBQ encryption scheme*

We transform SEBQ scheme using unbalanced Feistel transformation [18] and secure it against adaptive chosen ciphertext attacks. Additionally, using this transformation, the cryptosystems achieves another security goal of non-malleability as introduced by Dolev et al. [19].

Let SEBQ $= \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ be a symmetric encryption scheme described in Section 3. We proved that it is secure against IND-CPA attack in Theorem 4.1 and subsequently, in Proposition 4.2 we have shown that it is not IND-

CCA2 secure. So, to shield SEBQ scheme against IND-CCA2 attack we transform the SEBQ into $\widetilde{SEBQ}$ using unbalanced Feistel transformation.

Let M be a variable-length input pseudorandom function, which takes input of any pre-specified length and outputs some fixed length vector. Similarly, let G be a variable-length output pseudorandom function. Our transformation yields a better version of work carried out by [6, 18].

Let us first recall that the fixed-length pseudorandom functions come from a family of keyed multi-set in which functions have fixed (finite) domain and range corresponding to a key. It must be noted that a pseudorandom function is indistinguishable from a random function within the same domain and range. Variable-length input pseudorandom functions accepts an input of arbitrary and variable lengths, generating output of a fixed length. Most of the MACs are examples for this family. The variable-output pseudorandom functions takes fixed length input and outputs a variable length vector of user's choice.

The transformation that we apply to our encryption scheme that makes it secure against CCA depends on security of these variable-length output pseudorandom function.

Let k_0 be the positive integer. The transformation of $SEBQ$ to $\widetilde{SEBQ}$ is describe by following algorithm:

Algorithm 6 Transformed symmetric scheme $\widetilde{SEBQ}$

1. Consider a quasigroup $(Q,*)$ of order 2^k, assuming $Q=\mathbb{F}_2^k$, an initial vector $R=(r_1,r_2,\ldots,r_{k_0})\in Q^{k_0}$, and message $M=(m_1,m_2,\ldots,m_l)\in Q^l$.
2. Consider a variable-length output pseudorandom function $G:\mathbb{F}_2^{k_0}\to(\mathbb{F}_2^k)^*$.
3. In addition to quasigroup, the key generation algorithm also outputs a positive integer $a>1$ for variable-length output pseudorandom function G.
4. The transformed scheme modifies the function E_* as described in Subsection 3.2 to $\widetilde{E}_*$ as:

$$\widetilde{E}_*(m_i,R^{(i)})=E_*(m_i,G_a(R^{(i)})) \tag{5}$$

The function $D_\backslash$ as described in Subsection 3.3 gets transformed to $\widetilde{D}_\backslash$ as:

$$\widetilde{D}_\backslash(c_i,S^{(i)})=D_\backslash(c_i,G_a(S^{(i)})). \tag{6}$$

Recall that the encryption scheme $SEBQ$ is insecure to adaptive chosen-ciphertext attacks as it can be easily observed that adversary can distinguish the correct plaintext from out of two, given ciphertext of any one. This distinguisher is possible only when message and initial vector R are both elements of Q, but on applying the variable-length output pseudorandom function the length of $G_a(R)$ becomes equal to a, which is a positive integer greater than 1. Now upon querying decryption oracle $\widetilde{D}_{\setminus}(\cdot)$ as described in (6), it becomes impossible to recover the quasigroup $(Q,*)$. This is due to the increase in length of initial vector that is to be used in generation of Latin square corresponding to $(Q,*)$. As if initial vector was of length 1, then on querying decryption oracle gives the desired entry of Latin square as explained in Proposition 4.2. However, this transformation yields in slight increase of number of bit-operations in the encryption scheme, yet it provides security under adaptive chosen-ciphertext attacks. With these in-sights, we provide the following theorem.

Theorem 4.3: *The transformed symmetric encryption scheme* $\widetilde{SEBQ}$, *based on quasigroup, as described in Algorithm 6 is IND-CCA2 secure.*

4.2 *Randomness testing of SEBQ*

This section delve into the assessment of the entropy in the symmetric encryption (SEBQ) scheme, as proposed in Algorithm 3. To accomplish that, we conduct tests using the NIST test suite [3] and recorded the outcome into the Tables 3, 4 and 5. Additionally, we performed a comparative analysis of the outcomes against BCWST [8], INRU [40] in CBC mode and AES-128. In this experiment we will examine a 4000-bit long binary sequence serving as ciphertext. To generate the sequence of ciphertext we utilize a secret key and random initial vector both have 400-bit long sequence. The random initial vector was generated by following the ANSI x9.31 standard random number generation. These experiments were carried out with a significance level set at 99%. Tables 3, 4 and 5 display the success rate and consistency of p-values acquired through the execution of the NIST test suite on three types of plaintexts: 0x00, 0xFF and randomly generated plaintext. The p-values are the results of a chi-square test. We analyze the randomness of the cipher produced by Algorithm 3. Our analysis revealed that the randomness of SEBQ encryption is comparable with exiting symmetric encryption schemes like AES-128 and in some instances, it surpassed that of the previous schemes based on quasigroups [8, 40] and AES-128, affirming its superiority in terms of randomness and security.

Table 3

A comparative analysis on the success rates and uniformity of the p-values by running the NIST Test Suite for randomly generated plaintext across SEBQ, BCWST, INRU and AES-128 (where, NOTM-Non-overlapping template matching)

Test	SEBQ		BCWST [8]		INRU [40]		AES-128	
	Success %	p value	Sucess %	p value	Success %	p value	Success %	p value
Frequency	100	0.534146	100	0.1025	100	0.1626	99	0.8165
Block Frequency	99	0.739918	100	0.6371	100	0.4943	99	0.9914
Runs	100	0.739918	99	0.1626	100	0.6579	98	0.4190
Longest runs of ones in a block	100	0.534146	100	0.5543	98	0.2896	100	0.9716
Binary matrix rank	100	0.035174	99	0.0457	99	0.1916	98	0.2896
Discrete Fourier transform	100	0.350485	98	0.4944	98	0.4372	99	0.7197
Overlapping template matching	99	0.739918	100	0.5749	99	0.0711	98	0.6993
NOTM (149 templates)	98-100	0.122325-0.991413	96-100	0.0134-0.9978	96-100	0.0066-0.9963	95-100	0.0034-0.9914
Maurer's universal statistical	100	0.520354	100	0.475	98	0.1025	100	0.8676
Linear complexity	99	0.350485	99	0.5341	99	0.1025	100	0.9114
Serial 1	100	0.534146	99	0.5341	99	0.0965	98	0.2133
Serial 2	99	0.911413	99	0.4011	100	0.5141	99	0.9357
Approximate entropy	100	0.024011	100	0.0146	100	0.7981	99	0.0220
Cumulative sum forward	100	0.971736	99	0.4373	100	0.0046	99	0.9988
Cumulative sum backward	100	0.965373	100	0.6371	99	0.3345	100	0.6371
Random excursions (8 states)	98.46-100	0.000714-0.524432	98.24-100	0.0008-0.5544	98.46-100	0.0956-0.9411	98.24-100	0.0008-0.5544
Random excursions variant (18 states)	96.34-100	0.064321-0.974522	98.24-100	0.329-0.9879	95.38-100	0.0704-0.9705	98.24-100	0.0329-0.9879

Table 4

A comparative analysis on the success rates and uniformity of the p-values by running the NIST Test Suite for plaintext consisting solely of zeros (0x00) across SEBQ, BCWST, INRU and AES-128 (where, NOTM-Non-overlapping template matching)

Test	SEBQ		BCWST [8]		INRU [40]		AES-128	
	Success %	p value	Sucess %	p value	Success %	p value	Success %	p value
Frequency	99	0.513309	99	0.9717	100	0.4944	97	0.5141
Block frequency	100	0.911413	100	0.6579	99	0.8513	98	0.0135
Runs	98	0.934146	98	0.9781	99	0.6163	97	0.6371
Longest run of ones in a block	100	0.739918	100	0.6787	99	0.5141	99	0.4559
Rank	100	0.122325	100	0.9357	99	0.5141	97	0.0329
Discrete Fourier transform	100	0.534146	100	0.7598	99	0.1537	95	0.6579
Overlapping template matching	99	0.350485	98	0.5749	98	0.6163	99	0.0167
NOTM (149 templates)	96-100	0.017912-0.991468	96-100	0.004-0.9943	96-100	0.004-0.9978	95-100	0.0102-0.9942
Maures's universal statistical	100	0.52032	98	0.7981	99	0.0668	98	0.5141
Linear complexity	99	0.739918	99	0.2133	99	0.6786	99	0.9357
Serial 1	100	0.350485	100	0.1816	100	0.3669	100	0.2897
Serial 2	99	0.534146	99	0.7981	99	0.4559	99	0.1626
Approximate entropy	100	0.112300	100	0.1154	98	0.3041	99	0.8514
Cumulative sum forward	98	0.876384	98	0.7598	98	0.4190	99	0.2023
Cumulative sum backward	98	0.986294	99	0.5141	99	0.9879	99	0.7399
Random excursions (8 states)	98.46-100	0.200743-0.764423	97.18-100	0.2053-0.8810	98-100	0.0965-0.8832	97.10-100	0.0151-0.7565
Random excursions variant (18 states)	96.34-100	0.03852-0.974556	95.77-100	0.3863-0.9643	96-100	0.1223-0.9914	94.2-100	0.0151-0.8486

Table 5

A comparative analysis on the success rates and uniformity of the p-values by running the NIST Test Suite for plaintext consisting solely of ones (0xFF) across SEBQ, BCWST, INRU and AES-128 (where, NOTM-Non-overlapping template matching)

Test	SEBQ		BCWST [8]		INRU [40]		AES-128	
	Success %	p value	Sucess %	p value	Success %	p value	Success %	p value
Frequency	100	0.350485	99	0.9558	99	0.6371	100	0.1626
Block frequency	100	0.991468	100	0.3345	100	0.978	99	0.8677
Runs	98	0.350485	100	0.4012	99	0.0519	99	0.4373
Longest run of ones in a block	100	0.911413	100	0.3346	99	0.2368	100	0.3669
Rank	99	0.066882	99	0.1719	99	0.0519	100	0.5955
Discrete Fourier transform	98	0.035174	97	0.9643	98	0.3505	95	0.6163
Maurer's universal statistical	99	0.499323	99	0.5141	99	0.1223	100	0.4943
Overlapping template matching	99	0.122325	98	0.6163	99	0.9114	99	0.1626
NOTM(148 template)	96-100	0.008879-0.911413	96-100	0.0046-0.9915	96-100	0.0076-0.9914	95-100	0.0179-0.9781
Linear complexity	98	0.066882	98	0.6993	100	0.1537	100	0.9463
Serial 1	99	0.534146	98	0.0205	99	0.3505	100	0.3505
Serial 2	99	0.739918	98	0.1373	99	0.2757	100	0.2757
Approximate entropy	100	0.112300	100	0.6993	100	0.4749	95	0.3345
Cumulative sum forward	98	0.994708	99	0.8832	98	0.4349	100	0.7399
Cumulative sum backward	98	0.823133	99	0.6579	100	0.9558	99	0.6993
Random excursions (8 states)	98.46-100	0.120073-0.876423	96.88-100	0.1480-0.8623	98.21-100	0.00003-0.9114	98.55-100	0.0514-0.8755
Random excursions variant (18 states)	96.34-100	0.08522-0.975564	98.44-100	0.0822-0.9114	94.64-100	0.0020-0.9558	97.1-100	0.0190-0.7887

4.3 *Avalanche criterion*

We examine the impact of the avalanche effect on the ciphertext produced by Algorithm 3. When assessing the security of any symmetric encryption scheme, it is crucial to verify that any modification of input plaintext and key even a single bit leads to a significant alteration in the resulting ciphertext [7] also known as diffusion in ciphertext. The concept of avalanche criterion has been mathematically defined in various ways across different contexts, including block ciphers, Boolean functions and hash functions.

Furthermore, we conduct a comparative analysis of the avalanche criterion for SEBQ against the existing schemes like INRU [40], BCWST [8] and AES-128. Subsequently, present the findings in a tabular format.

4.3.1 Avalanche analysis by modification in key

We examine the variation in ciphertext percentages for SEBQ, as presented in Algorithm 3, whenever a key alteration occurs, specifically when the Latin square is modified. To assess whether SEBQ fulfills the avalanche criterion, we conducted experiments with 10 randomly generated Latin squares. We encrypted a randomly generated plaintext, comprising 4000 bits long using each unique secret keys along with the 400-bit long initial vector. The outcomes of the experiments are mentioned in the Table 6. Comparative analysis of the SEBQ with INRU [40], BCWST [8] and AES-128 is mentioned in Table 6.

Table 6

Comparative analysis of avalanche effect of secret key

Encryption scheme	Max % change in ciphertext	Min % change in ciphertext
BCWST [8]	50.054	49.931
INRU [40]	50.164	49.822
AES-128	50.046	49.937
SEBQ	**50.352**	**49.90**

4.3.2 Avalanche analysis in ciphertext by change in plaintext

To examine the influence of modifying a bit in the plaintext on the resultant ciphertext of SEBQ, we carry out a sequence of experiments. Specifically, we generate 100 random plaintext, each consisting of a 4000-

Table 7

Comparative analysis for maximum and minimum avalanche effect of plaintext

Encryption scheme	Max % change in ciphertext	Min % change in ciphertext
BCWST [8]	50.041	49.959
INRU [40]	50.042	49.943
AES-128	50.032	49.957
SEBQ	**52.550**	**48.000**

bit long vector. These plaintexts were then individually encrypted using a randomly generated secret Latin square of order 16×16 in conjunction with a 400-bit long initial vector. The resulting data is also compared with the existing schemes such as INRU [40], BCWST [8] and AES-128. The comparative outcomes are presented in Table 7. To ensure statistical significance, we repeated these experiments 10 times and record the outcomes of avalanche effect in Table 7 and 8.

Table 8

Observation of avalanche effect due to change in plaintext bit positions

Pos.	Percentage change in ciphertext with corresponding experiment										
	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}	Avg.
1	49.875	51.200	52.050	50.050	49.650	48.525	49.925	49.300	50.625	50.800	50.200
2	49.875	50.375	49.075	50.949	51.075	50.625	48.949	49.450	50.349	50.324	50.105
3	49.375	49.750	50.050	50.675	49.825	50.800	51.025	50.849	49.475	48.850	50.067
4	49.400	51.675	49.675	50.625	49.000	51.050	49.225	50.050	49.425	50.125	50.025
5	49.150	50.324	49.550	50.324	50.575	50.550	50.349	49.675	50.275	48.699	49.947
6	49.600	49.825	50.400	**48.000**	50.125	50.949	51.100	51.200	49.375	51.050	50.162
7	49.675	49.950	49.900	49.075	49.675	**52.550**	49.950	49.700	49.625	49.375	49.847
8	50.324	49.525	51.200	49.025	49.700	50.149	50.075	49.775	49.950	49.920	49.965
9	50.025	50.000	49.925	50.224	49.525	50.975	47.500	50.300	51.200	50.525	50.020
10	49.425	48.725	50.775	49.250	49.500	51.100	50.324	49.775	50.600	50.675	50.015

5. Analysis of SEBQ scheme

This section is committed to find out the total number of operations needed for encryption and decryption of plaintext in the proposed SEBQ

scheme as detailed in Section 3. Additionally, we will show that the number of operations require to encrypt and decrypt the plaintext with specific parameters is more efficient compared to the existing schemes such as INRU [40] and BCWST [8].

Consider $Q = \mathbb{F}_2^k$ and a Latin square of order 2^k. Now, let's assume that message $M = m_1, m_2, \ldots, m_l \in (\mathbb{F}_2^k)^l$ is being encrypted with the given Latin square in conjunction with the initial vector $IV = r_1, r_2, \ldots, r_n$.

Proposition 5.1: The number of operations required to encrypt the given message M using the given Latin square and the initial vector IV is $n + (l-1)(n+k)$.

Proof: The number of operations needed to encrypt the i^{th} block of message M is $(n+k)$ and for the entire message M is $(l-1)(n+k)$ operations. Additionally, updating the vector requires performing n xor operations. Therefore, the total number of operations required for the encryption scheme is equal to $n + (l-1)(n+k)$.

Propositon 5.2: The number of operations required to decrypt the given ciphertext C using the given Latin square and the initial vector IV is $n + (l-1)(n+k)$.

Hence, the total operations required for the symmetric encryption scheme SEBQ are equal to $2(n + (l-1)(n+k))$ which is proportional to $\mathcal{O}(nl)$.

Consider a scenarios where a user intends to secure a message $M \in \mathbb{F}_2^{64}$ through encryption and by using 128-bit long secret key along with 16-bit long initial vector. The number of operations required to encrypt the message M is just 70. In contrast, for the same message and parameter set, the INRU: A quasigroup based lightweight block cipher [40] necessitates 154 operations and to encrypt the same message with same parameter 120 number of operations are needed to perform in the encryption scheme BCWST [8]. In a nutshell, the symmetric encryption scheme SEBQ described in Algorithm 3 demonstrated superior efficiency compared to the lightweight block ciphers proposed in [40], [8].

Suppose $Q = \mathbb{F}_2^4$ and a Latin square of order m has been utilized to encrypt the message $M = m_1, m_2, \ldots, m_l \in \mathbb{F}_2^{4l}$ where $m_i \in \mathbb{F}_2^4$ in conjunction with random vector $R = r_1, r_2, \ldots, r_8 \in \mathbb{F}_2^{32}$ where each $r_i \in \mathbb{F}_2^4$. Suppose adversary $\mathcal{A}$ possesses information about the 128-bit ciphertext. The number of operations needed to make an educated guess about the plaintext corresponding to the provided ciphertext are detailed in

Proposition 5.3. The minimum order of Latin square to provide security against 128-bit is given by following Proposition 5.3.

Proposition 5.3: In order to attain 128-bit security against known ciphertext attack in SEBQ, it is imperative that the order of Latin square must be greater than 11.

Proof: According to the Proposition 5.2 the decryption of a 128-bit ciphertext necessitates 380 operations. In order for an adversary $\mathcal{A}$ to successfully guess the correct plaintext corresponding to given ciphertext it needed to perform operations greater than $L(m)\times 380$ where $L(m)$ represents the count of Latin squares of order m. To achieve 128-bits security level, parameter m ought to be selected in a manner that $L(m)\times 380 \geq 2^{128}$ which implies that the parameter m should be greater than 11. Conclusively, achieving a 128-bit security level implies that order of Latin square should be greater than 11.

6. Conclusion

This paper presents a symmetric encryption scheme (SEBQ) based on the structure of quasigroups and the string transformations. The SEBQ scheme employs a chaining-like mode of operation. In this mode, unlike the CBC, a transformed initial vector is utilized to encrypt each subsequent block of message. We proved that the SEBQ scheme is IND-CPA secure and further enhanced its security to withstand IND-CCA2 attack through the application of unbalanced Feistel transformation. To assess the performance and randomness of the algorithm, we subjected to the NIST test suite for statistical analysis and conducted comparative analysis with the existing schemes like INRU [40], BCWST [8] and AES-128. Additionally, we conducted an avalanche analysis of secret key and plaintext to gauge the diffusion properties of the SEBQ encryption scheme, comparing it to INRU [40], BCWST [8] and AES-128. Furthermore, our research included an analysis of SEBQ computational aspects, determining the number of operations needed to perform encryption and decryption process. Lastly, we identified the minimum order of Latin squares necessary to attain 128-bit security level against known-ciphertext attack.

The $\mathfrak{e}$ and $\mathfrak{d}$ -transformations as defined in Definition 2.4 and 2.5 can play a crucial role in improving the security of cryptographic algorithms based on quasigroups like SEBQ. This augmentation occurs by employing more than one quasigroup and more than one round with different initial

vectors [8]. Furthermore, the proposed design's security can be evaluated by experimenting with various quasigroups to tailor the algorithms for specific applications.

References

[1] F. H. Al-Rubbiay, "Cipher block chaining-based color medical image encryption for secure transmission," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 26 (2023).

[2] V. A. Artamonav, S. Chakrabarti, S. K. Tiwari, and V. T. Markov, "Algebraic properties of subquasigroups and construction of finite quasigroups," *Algebra and Logic*, vol. 61, pp. 251-270 (2022).

[3] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, et al., "SP 800-22 rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications," *National Institute of Standards & Technology* (2010).

[4] M. Battey and A. Parakh, "An efficient quasigroup block cipher," *Wireless Personal Communications,* vol. 73, pp. 63-76 (2013).

[5] M. Bellare, A. Desai, E. Jokiph, and P. Rogaway, "A concrete security treatment of symmetric encryption," Proceedings 38th Annual Symposium on Foundations of Computer Science, *IEEE*, pp. 394-403 (1997).

[6] M. Bellare and P. Rogaway, "Entity authentication and key distribution," Advances in Cryptology - CRYPTO-93. CRYPTO 1993, *Lecture Notes in Computer Science*, pp. 232-249 (1994).

[7] J. C. H. Castro, J. M. Sierra, A. Seznec, A. Izquierdo, and A. Ribagorda, "The strict avalanche criterion randomness test," *Mathematics and Computers Simulation*, vol. 68, pp. 1-7 (2005).

[8] D. Chauhan, I. Gupta, P. R. Mishra, and R. Verma, "An ultra-lightweight block cipher with string transformations," *Cryptologia*, pp. 1-7 (2023).

[9] W. L. Chee, "Public key cryptography based on Moufang loops," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, pp. 2411-2427 (2022).

[10] E. Couselo, S. Gonzalez, V. Markov, and A. Nechav, "Recursive MDS-codes and recursive differentiable quasigroups," *Discrete Math. Appl.*, vol. 8, pp. 217-246 (1998).

[11] L. Berardi, S. Jain, and B. K. Dass, "Bound for solid burst error correction with Lee weight," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 8, pp. 225-235 (2005).

[12] B. Biagio, B. K. Dass, and S. Jain, "High-density-burst error detection," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 7, pp. 5-21 (2004).

[13] B. K. Dass, "Burst Error Locating Linear Codes," *Journal of Information and Optimization Sciences*, vol. 3, pp. 77-80 (1982).

[14] B. K. Dass, N. Sharma, and R. Verma, "MDS and I-perfect poset block codes," *Finite Fields and Their Applications*, vol. 62, 101620 (2020).

[15] J. Dénes, "Latin squares and non-binary encoding," Proc. Conf. Information Theory, CNRS, Paris, pp. 215-221 (1979).

[16] J. Dénes and A. D. Keedwell, "Some applications of non-associative algebraic systems in cryptology," *Pure Mathematics and Applications*, vol. 12, pp. 147-195 (2001).

[17] J. Dénes and A. D. Keedwell, "Latin squares: New developments in the theory and applications," Elsevier, vol. 46 (1991).

[18] A. Desai, "New paradigms for constructing symmetric encryption schemes secure against chosen-ciphertext attack," *Annual International Cryptology Conference*, pp. 394-412 (2000).

[19] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing, pp. 542-552 (1991).

[20] L. Euler, "Recherches sur une espece de carrés magiques," *Commentationes Arithmeticae Collectae,* vol. 2, pp. 302-361 (1849).

[21] D. Gligoroski, S. Markovski, and L. Kocarev, "Edon-R, An Infinite Family of Cryptographic Hash Functions," *Int. J. Netw. Secur.*, vol. 8, no. 3, pp. 293-300 (2009).

[22] D. Gligoroski, S. Markovski, and L. Kocarev, "Error-Correcting Codes Based on Quasigroups," 16th International Conference on Computer Communications and Networks, pp. 165-172 (2007).

[23] D. Gligoroski, S. Markovski, and S. J. Knapskog, "A public key block cipher based on multivariate quadratic quasigroups," arXiv preprint arXiv:0808.0247 (2008).

[24] A. Joshi and A. K. Mohapatra, "A novel lightweight authentication protocol for body area networks based on elliptic-curve cryptography," *Journal of Information and Optimization Sciences,* vol. 41, pp. 1645-1672 (2020).

[25] J. Katz and Y. Lindell, Introduction to Modern Cryptography, CRC Press (2020).

[26] A. D. Keedwell and J. Dénes, Latin Squares and Their Applications, Elsevier (2015).

[27] S. Kumar, I. Gupta, and A. J. Gupta, "A study of public key cryptosystems based on quasigroups," *Cryptologia*, pp. 511-540 (2022).

[28] S. Kumar, H. Singh, I. Gupta, and A. J. Gupta, "MDS codes based on orthogonality of quasigroups," *Applicable Algebra in Engineering, Communication and Computing* (2023).

[29] M. Marcus and H. Minc, "Permanents," *The American Mathematical Monthly,* vol. 77, pp. 577-591 (1965).

[30] S. Markovski, V. Dimitrova, Z. Trajcheska, M. Petkovska, M. Kostadinoski, and D. Buhov, "Block cipher defined by matrix presentation of quasigroups," Cryptology ePrint Archive (2021).

[31] S. Markovski, D. Gligoroski, and S. Andova, "Using quasigroups for one-one secure encoding," Proc. VIII Conf. Logic and Computer Science, 'LIRA', vol. 97, pp. 157-162 (1997).

[32] S. Markovski, A. Mileva, S. Samardziska, and B. Jakimovski, NaSHA, in First SHA-3 Candidate Conference, February 25-28, K.U. Leuven, Belgium (2009).

[33] G. Mittal, S. Kumar, and S. Kumar, "Novel public-key cryptosystems based on NTRU and algebraic structure of group rings," *Journal of Information and Optimization Sciences,* vol. 42, pp. 1507-1521 (2021).

[34] R. Moufang, "Zur Struktur von Alternativkörpern," *Mathematische Annalen*, vol. 10, pp. 416-430 (1935).

[35] E. Ochodková and V. Snášel, "Using quasigroups for secure encoding of file systems," Proceedings of the International Scientific NATO PfP/PWP Conference Security and Information Protection, pp. 175-181 (2001).

[36] G. Shafi and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, pp. 270-299 (1984).

[37] J. Yu Shao and Wan-di Wei, "A formula for the number of Latin squares," *Discrete Mathematics,* vol. 110, pp. 293-296 (1992).

[38] V. Shcherbacov, Elements of Quasigroup Theory and Applications, CRC Press (2017).

[39] D. R. Stinson, Cryptography: Theory and Practice, CRC Press (2005).

[40] S. K. Tiwari, A. Awasthi, S. Chakrabarti, and S. Yadav, "INRU: A quasigroup-based lightweight block cipher," arXiv preprint arXiv:2112.07411 (2011).

[41] J. A. Tsimi and P. B. Gabriel Cedric, "On the generalized modal q-valent Reed-Muller codes," *Journal of Information and Optimization Sciences*, vol. 42, pp. 1885-1906 (2021).

[42] J. A. Tsimi and R. C. Youdom, "The modal q-valent extensions of BCH codes," *Journal of Information and Optimization Sciences*, vol. 42, pp. 1723-1764 (2021).

Received December, 2023