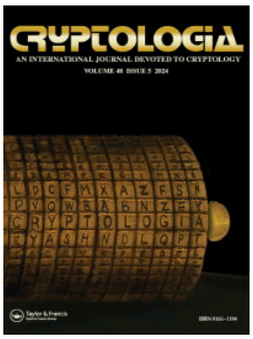

Taylor & Francis
Taylor & Francis Group

Cryptologia

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ucry20

An ultra-lightweight block cipher with string transformations

Dimpy Chauhan, Indivar Gupta, P. R. Mishra & Rashmi Verma

To cite this article: Dimpy Chauhan, Indivar Gupta, P. R. Mishra & Rashmi Verma (2024) An ultra-lightweight block cipher with string transformations, Cryptologia, 48:5, 437-468, DOI: 10.1080/01611194.2023.2224107

To link to this article: https://doi.org/10.1080/01611194.2023.2224107

Published online: 24 Aug 2023.

CRYPTOLOGIA
2024, VOL. 48, NO. 5, 437–468
https://doi.org/10.1080/01611194.2023.2224107

Taylor & Francis
Taylor & Francis Group

An ultra-lightweight block cipher with string transformations

Dimpy Chauhan, Indivar Gupta, P. R. Mishra, and Rashmi Verma

Department of Mathematics, University of Delhi, Delhi, India

ABSTRACT

Security efficiency and hardware efficiency are equally important when designing a block cipher. In this paper, we propose a new block cipher that uses string transformations and requires less memory and fewer computational resources, making it suitable for highly constrained environments. We compare the performance of our design with AES-128 and quasigroup-based block cipher INRU in the cipher block chaining (CBC) mode of operation using the National Institute of Standards and Technology (NIST) test suite. We then analyze our design against the standard linear, differential, and algebraic attacks. Choosing suitable S-boxes and a quasigroup of a smaller order does not compromise the security of the cipher, but the storage space is reduced compared with AES-128 and INRU.

KEYWORDS

block cipher; quasigroup; S-box; string transformation;

2010 MATHEMATICS SUBJECT CLASSIFICATION

94A60; 20N05

1. Introduction

One area of interest in the study of cryptography is the design of cryptographic algorithms using novel approaches. Over the past two decades, the application of quasigroups in cryptography has evolved at a high speed and received a lot of attention from researchers. Quasigroups are widely used in designing various cryptographic primitives, including S-boxes (Chauhan et al. 2022; Mihajloska and Gligoroski 2012; Yu and Xu 2017), block ciphers (Battey and Parakh 2012; Battey and Parakh 2013; Markovski et al. 2014; Tiwari et al. 2021; Zhao and Xu 2017), stream ciphers (Markovski, Gligoroski, and Andova 1997; Ochodková and Snášel 2001; Vojvoda 2004), hash functions (Gligoroski, Dimitrova, et al. 2009; Markovski and Mileva 2009), secret sharing schemes (Cooper, Donovan, and Seberry 1994), message authentication codes (Meyer 2006), and many more (Chauhan, Gupta, and Verma 2021; Mileva 2010). The structure and properties (nonassociativity and noncommutativity) of quasigroups and the existence of large numbers of them have led to their application in various fields such as cryptography, coding theory, telecommunications, and experimental designs.

CONTACT Dimpy Chauhan dimpychauhan21jan@gmail.com Department of Mathematics, University of Delhi, Delhi 110007, India.

The closure and inversion properties of quasigroups make them useful tools for constructing an efficient cryptosystem, and the lack of associativity facilitates the creation of one-way functions. Further, because quasigroups lack commutative and associative properties, many well-known traditional cryptographic approaches to attack quasigroup-based cryptographic designs cannot be applied directly. Quasigroups have an algebraic and combinatorial structure equivalent to that of Latin squares. The theory of Latin squares is well studied and includes several fascinating areas of research such as mutually orthogonal Latin squares, transversals of Latin squares, and Latin subsquares (Belousov and Belyavskaya 1989; Keedwell and Dénes 2015; Laywine and Mullen 1998). Thus, quasigroups and Latin squares play a significant role in designing cryptographic primitives. The reader can refer to other sources (Belousov and Belyavskaya 1989; Dénes and Keedwell 1974; Keedwell and Dénes 2015; Laywine and Mullen 1998) for an introduction to and applications of the Latin square. Markovski, Gligoroski, and Andova (1997) proposed quasigroup string transformations, and other authors have shown that these are the key component in the development of quasigroup-based encryption algorithms (Battey and Parakh 2012; Battey and Parakh 2013; Dimitrova 2010; Markovski, Gligoroski, and Bakeva 1999; Zhao and Xu 2017). The security of these algorithms is highly reliant on the selection of appropriate quasigroups. In block ciphers, S-boxes are the only nonlinear part, so they play an important role in the design and security of any block cipher. The rows of a Latin square/quasigroup can be considered S-boxes. Thus, to make a block cipher secure against different attacks, S-boxes/Latin squares should be chosen with care during the design of the cipher.

The widespread use of small computing devices such as sensors, smart phones, and tablets necessitates the development of encryption methods that are suitable for low-resource devices. Furthermore, as cloud services become more prevalent, the amount of data sent to and from these devices is increasing at an exponential rate (Parakh and Kak 2009). As a result, lightweight cryptography has received a lot of attention in recent years. Pervasive computing, as demonstrated by the use of smart cards, radio-frequency identification (RFID) tags, and sensor nodes, is transforming and improving everyday life but also introducing security concerns and threats. Several new cryptographic primitives have been proposed for use in resource-constrained situations, led by lightweight block ciphers (Battey and Parakh 2012; Battey and Parakh 2013; Bogdanov et al. 2007; Tiwari et al. 2021; Zhao and Xu 2017). In addition, the National Institute of Standards and Technology (NIST) has initiated a project to solicit, evaluate, and standardize lightweight cryptographic algorithms that are suitable for highly constrained environments. On February 7, 2023, NIST announced the selection of the Ascon family for lightweight cryptography standardization.

In this paper, we define new string transformations over an ordered set of m bijective S-boxes of order m. Using the string transformations over an ordered set of four S-boxes of order 4 and elementary string transformations over quasigroups of order 4 and 2, an extremely lightweight block cipher has been designed that we call BCWST. The security of BCWST depends on the selection of appropriate S-boxes and quasigroups used in the string transformations. The round functions in the encryption algorithm of BCWST consist of a string transformation over an ordered set S of four S-boxes of order 4, a string transformation over a quasigroup of order 4, and a string transformation over a quasigroup of order 2. In the decryption algorithm, the elementary string transformations based on their conjugate quasigroups and string transformations over S^{-1} are used. We analyzed the security of our design and compared its randomizing ability with that of AES-128 (Daemen and Rijmen 2002) and quasigroup-based block cipher INRU (Tiwari et al. 2021) using the NIST statistical test suite in the cipher block chaining (CBC) mode of operation. Further, an avalanche analysis of keystream and plaintext of BCWST, AES-128, and INRU is presented. We then show the robustness of our design against linear, differential, and algebraic attacks. The AES and INRU ciphers use an 8×8 S-box and a polynomially complete quasigroup (Artamonov et al. 2013) of order 16 that does not have any proper subquasigroups, respectively. To achieve hardware efficiency in BCWST, four 4-bit S-boxes and a quasigroup of order 4 were used. The implementation of these S-boxes and quasigroup is much more compact than the use of an 8×8 S-box or a 16-order quasigroup. Moreover, finding a nonlinear quasigroup of order 4 and four cryptographically strong 4×4 S-boxes is easier than finding a polynomially complete quasigroup of order 16 with no proper subquasigroup, as used in the INRU cipher. By choosing suitable S-boxes and a quasigroup of a smaller order in designing BCWST, storage space is reduced compared with the AES-128 and INRU ciphers, without compromising security.

In Section 2, we provide the definitions, properties, and results related to the quasigroups and S-boxes that are required to understand subsequent sections. Section 3 defines the elementary quasigroup string transformations used in our cipher. In Section 4, we define new string transformations over S-boxes, followed by an illustration of BCWST. The security analysis of the proposed cipher, including the design's randomness and robustness against standard linear, differential, and algebraic attacks, is discussed in Section 5. Section 6 concludes the paper.

2. Quasigroups and S-boxes

In this section, we discuss quasigroups, their representation, and S-boxes. The reader may refer to other sources (Chauhan, Gupta, and Verma 2021;

Chauhan et al. 2022; Markovski, Gligoroski, and Bakeva 1999; Wu and Feng 2016) for a more detailed study.

Definition 1. A binary groupoid $(Q, *)$ is called a *quasigroup* if there exist unique solutions $x, y \in Q$ of the equations $x * a = b$ and $a * y = b$ for all ordered pairs $(a, b) \in Q^2$.

In view of Definition 1, given a quasigroup $(Q, *)$, it is possible to associate five other quasigroups $(Q, \cdot), (Q, /), (Q, \backslash), (Q, //)$, and $(Q, \backslash\backslash)$, known as *parastrophes* or *conjugates* of quasigroup $(Q, *)$, as follows:

$x*y=z \Longleftrightarrow y \cdot x=z$ (opposite multiplication) $\Longleftrightarrow z/y=x$ (right division) $\Longleftrightarrow x\backslash z=y$ (left division) $\Longleftrightarrow y//z=x$ (opposite right division) $\Longleftrightarrow z\backslash\backslash x=y$ (opposite left division).

The operations $\cdot, /, \backslash, //$, and $\backslash\backslash$ are sometime denoted as $*^{(12)}, *^{(13)}, *^{(23)}, *^{(123)}$, and $*^{(132)}$, respectively.

Let Q be a finite set of order n. A *Latin square* on Q is an n ordered square matrix with elements from Q such that each element of Q occurs exactly once in each row and once in each column of the matrix. Note that every quasigroup of order n can be represented as an n ordered Latin square.

If $N(n)$ is the number of quasigroups of order n, then $N(n)$ is given by the following inequality (van Lint and Wilson 2001):

$$\frac{(n!)^{2n}}{n^{n^2}} \leq N(n) \leq \prod_{k=1}^{n} (k!)^{\frac{n}{k}}. \tag{1}$$

This shows that the number of quasigroups increases exponentially with the order. This is one of the properties of quasigroups that enables their application in cryptography (van Lint and Wilson 2001).

Let $\mathbb{F}_2$ be a Galois field of order 2, and n and m positive natural numbers. A Boolean function is a map $f : \mathbb{F}_2^n \to \mathbb{F}_2$, and a Boolean map or vector valued Boolean function is a map $g : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Every such Boolean map can be written as m n-ary Boolean functions, such as $g_0, g_1, ..., g_{m-1}$ (known as coordinate functions of g), in the following way:

$$g(x_0, x_1, ..., x_{n-1}) = (g_0(x_0, x_1, ..., x_{n-1}), g_1(x_0, x_1, ..., x_{n-1}), ..., g_{m-1}(x_0, x_1, ..., x_{n-1})). \tag{2}$$

The nonzero linear combinations of g_i's over $\mathbb{F}_2$ are called component functions of g. Note that a coordinate function is also a component function. Every Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ can be uniquely represented in an algebraic normal form (ANF) in the following way:

$$f(x_0, x_1, ..., x_{n-1}) = \sum_{I \in P(N)} a_I \left(\prod_{i \in I} x_i \right) \tag{3}$$

where $a_I \in \mathbb{F}_2$ and $P(N)$ denotes the power set of $N = \{0, 1, ..., n-1\}$.

Every quasigroup (Q, f) of order 2^n $(n \geq 2)$ can be represented as a Boolean map $f : \mathbb{F}_2^{2n} \to \mathbb{F}_2^n$, as follows:

$$f(x_0, x_1, ..., x_{n-1}; y_0, y_1, ..., y_{n-1}) = (f_0(x_0, x_1, ..., x_{n-1}; y_0, y_1, ..., y_{n-1}), ..., f_{n-1}(x_0, x_1, ..., x_{n-1}; y_0, y_1, ..., y_{n-1})), \tag{4}$$

where $x_0, x_1, ..., x_{n-1} \in \mathbb{F}_2^n$ and $y_0, y_1, ..., y_{n-1} \in \mathbb{F}_2^n$, and $f_i : \mathbb{F}_2^{2n} \to \mathbb{F}_2$ are Boolean functions, $0 \leq i \leq n-1$. The algebraic degree of quasigroup (Q, f) is the algebraic degree of the vector valued Boolean function f, which is equal to the maximum of the degree of its coordinate functions f_i; $0 \leq i \leq n-1$, if written in ANF. That is,

$$\deg(f) = \max\{\deg(f_i) : i = 0, 1, ..., n-1\}. \tag{5}$$

An $n \times m$ S-box S is a vector valued Boolean function $S : \mathbb{F}_2^n \to \mathbb{F}_2^m$. They are used mainly to infuse nonlinearity in a block cipher. Note that the function f in Equation (4) can be viewed as a $2n \times n$ S-box.

A quasigroup (Q, f) is a linear quasigroup if all the component Boolean functions of f are linear functions; otherwise, the quasigroup is a nonlinear quasigroup (Dimitrova 2010; Gligoroski, Dimitrova, et al. 2009).

Remark 1. The rows/columns of a Latin square of order 2^n can be considered $n \times n$ S-boxes. There are 2^n such S-boxes corresponding to the 2^n rows/columns of the Latin square.

Definition 2 (Chauhan et al. 2022). An S-box S of order 4 is said to be a *cryptographically strong S-box* if it satisfies the following conditions:

1. S is balanced,
2. nonlinearity of S is 4,
3. differential uniformity of S is 4, and
4. algebraic degree of S is 3.

3. Quasigroup string transformations

Definition 3 (Markovski, Gligoroski, and Andova 1997). Let (Q, f) be a finite quasigroup, and $Q^n = \{(x_0, x_1, ..., x_{n-1}) | x_i \in Q, n \geqslant 2\}$ denote the set of all n-tuples over Q. For a fixed element $l \in Q$, two functions $\mathcal{L}e_{l,(Q,f)} : Q^n \to Q^n$ and $\mathcal{L}d_{l,(Q,f)} : Q^n \to Q^n$ are defined as follows:

$$\mathcal{L}e_{l,(Q,f)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \Longleftrightarrow y_j = \begin{cases} f(l, x_0), j = 0 \\ f(y_{j-1}, x_j), 1 \leq j \leq n-1 \end{cases} \tag{6}$$

and

$$\mathcal{L}d_{l,(Q,f)}(x_0,x_1,\ldots,x_{n-1})=(y_0,y_1,\ldots,y_{n-1}) \Longleftrightarrow y_j=\begin{cases} f(l,x_0), j=0 \\ f(x_{j-1},x_j), 1\le j\le n-1. \end{cases} \tag{7}$$

These functions are called *elementary quasigroup transformations*, or *e-transformation* and *d-transformation* of Q^n with leader l (over the quasigroup(Q, f)), respectively.

In this paper, we call the e-transformation $\mathcal{L}e_{l,(Q,f)}$ and the d-transformation $\mathcal{L}d_{l,(Q,f)}$ *left e-transformation* and *left d-transformation* or *left elementary quasigroup transformation* of Q^n with leader l, respectively. Their functions are shown graphically in Figures 1 and 2.

Theorem 1 (Markovski, Gligoroski, and Bakeva 1999). *Let $(Q,f,\backslash,/)$ be a finite quasigroup. Then for each leader $l\in Q$ and for each string $x\in Q^n$ we have $\mathcal{L}e_{l,(Q,f)}$ and $\mathcal{L}d_{l,(Q,\backslash)}$, which are mutually inverse permutations of Q^n, i.e., $\mathcal{L}d_{l,(Q,\backslash)}(\mathcal{L}e_{l,(Q,f)}(x))=x=\mathcal{L}e_{l,(Q,f)}(\mathcal{L}d_{l,(Q,\backslash)}(x))$.*

From Theorem 1, we can conclude that the transformations $\mathcal{L}e_{l,(Q,f)}$ and $\mathcal{L}d_{l,(Q,\backslash)}$ can be used to define suitable encryption and decryption functions.

Definition 4. Let (Q, f) be a finite quasigroup. For a fixed element $l\in Q$, two more functions $\mathcal{R}e_{l,(Q,f)}:Q^n\to Q^n$ and $\mathcal{R}d_{l,(Q,f)}:Q^n\to Q^n$ are defined as follows:

$$\mathcal{R}e_{l,(Q,f)}(x_0,x_1,\ldots,x_{n-1})=(y_0,y_1,\ldots,y_{n-1}) \Longleftrightarrow y_j=\begin{cases} f(l,x_{n-1}), j=n-1 \\ f(y_{j+1},x_j), n-2\ge j\ge 0 \end{cases} \tag{8}$$

and

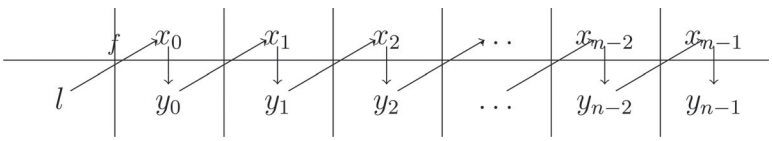

Figure 1. Graphical representation of $\mathcal{L}e_{l,(Q,f)}$.

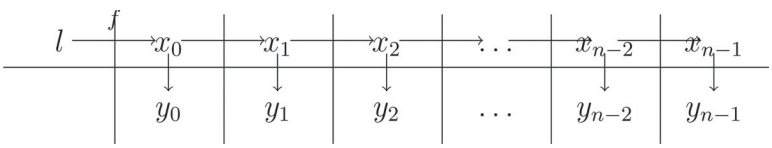

Figure 2. Graphical representation of $\mathcal{L}d_{l,(Q,f)}$.

$$\mathcal{R}d_{l,(Q,f)}(x_0,x_1,\ldots,x_{n-1})=(y_0,y_1,\ldots,y_{n-1}) \iff y_j=\begin{cases} f(l,x_{n-1}), j=n-1 \\ f(x_{j+1},x_j), n-2\geq j\geq 0. \end{cases} \tag{9}$$

These functions are called *right elementary quasigroup transformations,* or *right e-transformation* and *right d-transformation* of Q^n with leader l (over the quasigroup(Q, f)), respectively. Their graphical representations are shown in Figures 3 and 4.

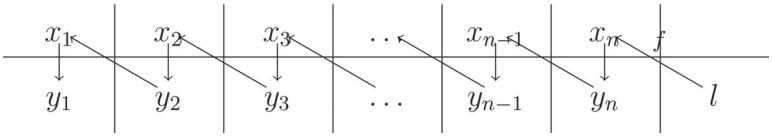

Figure 3. Graphical representation of $\mathcal{R}e_{l,(Q,f)}$.

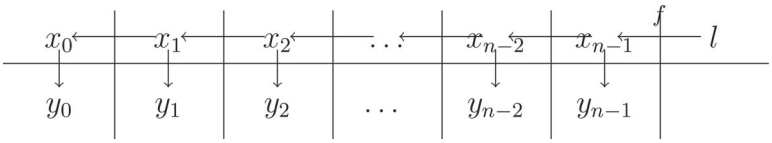

Figure 4. Graphical representation of $\mathcal{R}d_{l,(Q,f)}$.

Theorem 2. *Let $(Q,f,\backslash,/)$ be a finite quasigroup. Then for each leader $l\in Q$ and for each string $x\in Q^n$ we have $\mathcal{R}e_{l,(Q,f)}$ and $\mathcal{R}d_{l,(Q,\backslash)}$, which are mutually inverse permutations of Q^n, i.e., $\mathcal{R}d_{l,(Q,\backslash)}(\mathcal{R}e_{l,(Q,f)}(x))=x=\mathcal{R}e_{l,(Q,f)}(\mathcal{R}d_{l,(Q,\backslash)}(x))$.*

Proof. Clearly $\mathcal{R}e_{l,(Q,f)}$ and $\mathcal{R}d_{l,(Q,\backslash)}$ are permutations of Q^n (since $(Q,f,\backslash,/)$ is a finite quasigroup). Let $x=x_0,x_1,\ldots,x_{n-1}\in Q^n$ and $l\in Q$ be a leader. Let

$$\mathcal{R}e_{l,(Q,f)}(x_0,x_1,\ldots,x_{n-1})=(y_0,y_1,\ldots,y_{n-1}) \iff y_i=\begin{cases} f(l,x_{n-1}), i=n-1 \\ f(y_{i+1},x_i), n-2\geq i\geq 0, \end{cases} \tag{10}$$

and

$$\mathcal{R}d_{l,(Q,\backslash)}(y_0,y_1,\ldots,y_{n-1})=(z_0,z_1,\ldots,z_{n-1}) \iff z_i=\begin{cases} l\backslash y_{n-1}, i=n-1 \\ y_{i+1}\backslash y_i, n-2\geq i\geq 0. \end{cases} \tag{11}$$

For $i=n-1$,

$$\begin{aligned} & z_{n-1}=l\backslash y_{n-1} \text{ (from (11))} \\ \text{i.e. } & z_{n-1}=l\backslash f(l,x_{n-1}) \text{ (from (10))} \\ \iff & f(l,z_{n-1})=f(l,x_{n-1}) \text{ (from definition of parastrophes)} \\ \iff & z_{n-1}=x_{n-1} \text{ (from left cancellation law of quasigroup } (Q,f)). \end{aligned}$$

For $0 \leq i \leq n-2$,

$$\begin{gathered} z_i = y_{i+1} \backslash y_i \ \text{(from (11))} \\ \text{i.e.} \quad z_i = y_{i+1} \backslash f(y_{i+1}, x_i) \ \text{(from (10))} \\ \iff f(y_{i+1}, z_i) = f(y_{i+1}, x_i) \ \text{(from definition of parastrophes)} \\ \iff z_i = x_i \ \text{(from left cancellation law of quasigroup } (Q,f)). \end{gathered}$$

Hence, we get $\mathcal{R}d_{l,(Q,\backslash)}(\mathcal{R}e_{l,(Q,f)}(x_0, x_1, ..., x_{n-1})) = x_0, x_1, ..., x_{n-1}$.

For the other equality, suppose

$$\mathcal{R}d_{l,(Q,\backslash)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \iff y_i = \begin{cases} l \backslash x_{n-1}, i = n-1 \\ x_{i+1} \backslash x_i, n-2 \geq i \geq 0, \end{cases} \tag{12}$$

and

$$\mathcal{R}e_{l,(Q,f)}(y_0, y_1, ..., y_{n-1}) = (z_0, z_1, ..., z_{n-1}) \iff z_i = \begin{cases} f(l, y_{n-1}), i = n-1 \\ f(z_{i+1}, y_i), n-2 \geq i \geq 0. \end{cases} \tag{13}$$

For $i = n-1$,

$$\begin{gathered} z_{n-1} = f(l, y_{n-1}) \ \text{(from (13))} \\ \text{i.e.} \quad z_{n-1} = f(l, (l \backslash x_{n-1})) \ \text{(from (12))} \\ \iff l \backslash z_{n-1} = l \backslash x_{n-1} \ \text{(from definition of parastrophes)} \\ \iff z_{n-1} = x_{n-1} \ \text{(from left cancellation law of quasigroup } (Q, \backslash)). \end{gathered}$$

For $0 \leq i \leq n-2$,

$$\begin{gathered} z_i = f(z_{i+1}, y_i) \ \text{(from (11))} \\ \text{i.e. } z_i = f(z_{i+1}, (x_{i+1} \backslash x_i)) \ \text{(from (10))} \\ \iff z_{i+1} \backslash z_i = x_{i+1} \backslash x_i \text{(from definition of parastrophes)}. \end{gathered}$$

Thus, for $i = n-2$, we have $z_{n-1} \backslash z_{n-2} = x_{n-1} \backslash x_{n-2}$. Also, $z_{n-1} = x_{n-1}$, which implies $z_{n-2} = x_{n-2}$ (from left cancellation law of $(Q, \backslash)$). Continuing in a similar way, we get $z_i = x_i$, for all $0 \leq i \leq n-2$. Hence, $\mathcal{R}e_{l,(Q,f)}(\mathcal{R}d_{l,(Q,\backslash)}(x_0, x_1, ..., x_{n-1})) = x_0, x_1, ..., x_{n-1}$. □

From Theorem 2, we can conclude that the transformations $\mathcal{R}e_{l,(Q,f)}$ and $\mathcal{R}d_{l,(Q,\backslash)}$ can also be used to define suitable encryption and decryption functions.

Remark 2. The pair of transformations $(\mathcal{L}e_{l,(Q,f)}, \mathcal{L}d_{l,(Q,/)})$ and $(\mathcal{R}e_{l,(Q,f)}, \mathcal{R}d_{l,(Q,/)})$ can also be used suitably for encryption and decryption purposes.

4. Description of BCWST cipher

In this section, we define new string transformations over an ordered set of bijective S-boxes. Then, using these string transformations and elementary string transformations over a quasigroup, we proposes the block cipher BCWST.

4.1. String transformations over S-boxes

Definition 5. Let $S = \{S_0, S_1, ..., S_{m-1}\}$ be an ordered set of m bijective S-boxes of order m and $Q = \mathbb{F}_2^m$. For a fixed element $l \in Q$, define functions $L\varepsilon_{l,(Q,S)} : Q^n \to Q^n$ and $R\varepsilon_{l,(Q,S)} : Q^n \to Q^n$ as follows:

$$L\varepsilon_{l,(Q,S)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \iff y_i = \begin{cases} S_{l \bmod m}(x_0), i = 0 \\ S_{y_{i-1} \bmod m}(x_i), 1 \le i \le n-1 \end{cases} \tag{14}$$

and

$$R\varepsilon_{l,(Q,S)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \iff y_i = \begin{cases} S_{l \bmod m}(x_{n-1}), i = n-1 \\ S_{y_{i+1} \bmod m}(x_i), n-2 \ge i \ge 0, \end{cases} \tag{15}$$

where $S_{t \bmod m} := S_k$; $t \in Q$ and $k = 0, 1, 2, ..., m-1$ are the m S-boxes of S. We call these functions *left string transformation* (or *left* ε*-transformation*) and *right string transformation* (or *right* ε*-transformation*) over the set S of S-boxes with leader l, respectively.

Note that the ordered set means that the order of the S-boxes in set S matters. That is, for a different order, such as $S' = \{S_1, S_0, S_2, ..., S_{m-1}\}$, we get a different function $L\varepsilon_{l,(Q,S')}$.

The graphical representations of the left and right string transformations over the set S with leader l are shown in Figures 5 and 6, respectively.

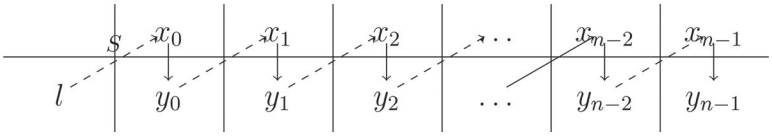

Figure 5. Graphical representation of $L\varepsilon_{l,(Q,S)}$.

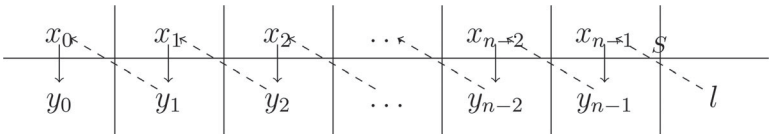

Figure 6. Graphical representation of $R\varepsilon_{l,(Q,S)}$.

Definition 6. Let $S = \{S_0, S_1, ..., S_{m-1}\}$ be an ordered set of m bijective S-boxes of order m and $Q = \mathbb{F}_2^m$. For a fixed element $l \in Q$, we define functions $LD_{l,(Q,S)} : Q^n \to Q^n$ and $RD_{l,(Q,S)} : Q^n \to Q^n$ as follows:

$$LD_{l,(Q,S)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \iff y_i = \begin{cases} S_{l \bmod m}(x_0), i = 0 \\ S_{x_{i-1} \bmod m}(x_i), 1 \le i \le n-1, \end{cases} \tag{16}$$

and

$$RD_{l,(Q,S)}(x_0, x_1, ..., x_{n-1}) = (y_0, y_1, ..., y_{n-1}) \iff y_i = \begin{cases} S_{l \bmod m}(x_{n-1}), i = n-1 \\ S_{y_{i+1} \bmod m}(x_i), n-2 \ge i \ge 0, \end{cases} \tag{17}$$

where $S_{t \bmod m} := S_k$; $t \in Q$ and $k = 0, 1, 2, ..., m-1$ are the m S-boxes of S. We call these functions *left inverse string transformation* (or *left D-transformation*) and *right inverse string transformation* (or *right D-transformation*) over the set S of S-boxes with leader l, respectively.

The graphical representations of the left and right inverse string transformations over the set S with leader l are shown in Figures 7 and 8, respectively.

Theorem 3. *Let S be an ordered set of m bijective S-boxes of order m and $Q = \mathbb{F}_2^m$. Then the string transformations $L\varepsilon_{l,(Q,S)}, R\varepsilon_{l,(Q,S)}, LD_{l,(Q,S)}$, and $RD_{l,(Q,S)}$ are permutations on Q^n.*

Proof. The proof of the theorem follows from the fact that S-boxes are bijective functions from Q to Q. □

Define the inverse of the ordered set $S = \{S_0, S_1, ..., S_{m-1}\}$ as an ordered set $S^{-1} = \{S_0^{-1}, S_1^{-1}, ..., S_{m-1}^{-1}\}$ and $S^{-1}_{t \bmod m} := S_k^{-1}$; $t \in Q$ and $k = 0, 1, ..., m-1$, where $Q = \mathbb{F}_2^m$.

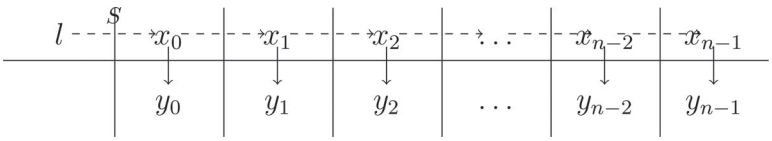

Figure 7. Graphical representation of $LD_{l,(Q,S)}$.

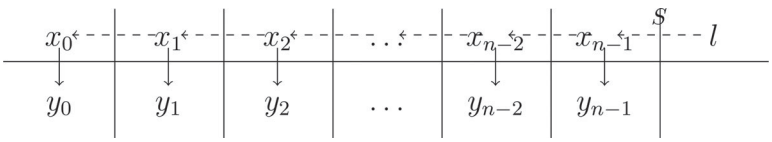

Figure 8. Graphical representation of $RD_{l,(Q,S)}$.

Theorem 4. *Let S be an ordered set of m bijective S-boxes of order m and $Q = \mathbb{F}_2^m$. Then for a fixed element $l \in Q$ and for each string $x \in Q^n$ we have $L\varepsilon_{l,(Q,S)}$ and $LD_{l,(Q,S^{-1})}$, which are mutually inverse permutations of Q^n, i.e., $LD_{l,(Q,S^{-1})}(L\varepsilon_{l,(Q,S)}(x)) = x = L\varepsilon_{l,(Q,S)}(x)(LD_{l,(Q,S^{-1})}(x))$.*

Proof. The proof of the theorem follows trivially from the definition of $L\varepsilon_{l,(Q,S)}$ and $LD_{l,(Q,S^{-1})}$ and is hence omitted. □

From Theorem 4, we can conclude that the transformations $L\varepsilon_{l,(Q,S)}$ and $LD_{l,(Q,S^{-1})}$ can be used to define suitable encryption and decryption functions.

Theorem 5. *Let S be an ordered set of m bijective S-boxes of order m and $Q = \mathbb{F}_2^m$. Then for a fixed element $l \in Q$ and for each string $x \in Q^n$ we have $R\varepsilon_{l,(Q,S)}$ and $RD_{l,(Q,S^{-1})}$, which are mutually inverse permutations of Q^n, i.e., $RD_{l,(Q,S^{-1})}(R\varepsilon_{l,(Q,S)}(x)) = x = R\varepsilon_{l,(Q,S)}(x)(RD_{l,(Q,S^{-1})}(x))$.*

Proof. The result follows trivially from the definition of $R\varepsilon_{l,(Q,S)}$ and $RD_{l,(Q,S^{-1})}$ and is hence omitted. □

From Theorem 5, we can conclude that the transformations $R\varepsilon_{l,(Q,S)}$ and $RD_{l,(Q,S^{-1})}$ can also be used to define suitable encryption and decryption functions.

Remark 3. It is worth noting that $LD_{l,(Q,S^{-1})} = L\varepsilon^{-1}_{l,(Q,S)}$ and $RD_{l,(Q,S^{-1})} = R\varepsilon^{-1}_{l,(Q,S)}$ are the inverse of the left and right string transformations over the ordered set S of S-boxes, respectively.

The string transformations over the ordered set S of four bijective S-boxes of order 4 and the elementary string transformations over quasigroups of order 2 and 4 are the basic building blocks of our design. The security of our cipher depends on the choice of S-boxes in set S and the quasigroups used in the string transformations. Hence, we take string transformations over the set of four cryptographically strong S-boxes of order 4 and over a nonlinear quasigroup of order 4. This is discussed in detail in the next section.

4.2. Components of the BCWST cipher

We designed a block cipher that comprises 10 rounds and operates on a 64-bit block length. We used an ordered set S of four bijective S-boxes of order 4, $S = \{S_0, S_1, S_2, S_3\}$, and two quasigroups $(Q_1, *_1)$ and $(Q_2, *_2)$ of orders 4 and 2, respectively, where $Q_1 = \mathbb{F}_2^2$ and $Q_2 = \mathbb{F}_2$. The four S-boxes of the set S are cryptographically strong and have been chosen in such a

way that when these S-boxes are written as rows, none of the elements in each column repeats. These S-boxes can be obtained from any of the methods given in a number of sources (Chauhan et al. 2022; Leander and Poschmann 2007; Mihajloska and Gligoroski 2012; Yu and Xu 2017). The quasigroup $(Q_1, *_1)$ is nonlinear. The quasigroup $(Q_2, *_2)$ is such that the element at the $(0,0)^{\text{th}}$ position of its Latin square is the XOR of the first and last bit of the last round key obtained from the round key generation algorithm (discussed in Section 4.3). The quasigroup $(Q_2, *_2)$ is linear and is determined by the last round key used as a post-whitening key. Here, we use the ordered set S whose four S-boxes $S_0, S_1, S_2,$ and S_3 are as given in Table 1 and the quasigroup $(Q_1, *_1)$ as given in Table 2 (L_{289} in lexicographical order). Note that the nonlinear quasigroups are correlated quasigroups and weak restricted quasigroups. The reason for using nonlinear quasigroups when designing cryptographic primitives is discussed by Mileva and Markovski (2008).

The ANF of the four S-boxes $S_0, S_1, S_2,$ and S_3 of the set S are as follows:

$$\begin{aligned} S_0(x) = (&x_0 + x_2 + x_0x_2 + x_1x_2 + x_0x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_0 + x_1 + x_2 + x_1x_2 + x_0x_3 + x_2x_3 + x_1x_2x_3,\\ &1 + x_1 + x_0x_1 + x_0x_2 + x_0x_3 + x_1x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_2 + x_3 + x_0x_2 + x_0x_3 + x_2x_3 + x_0x_1x_2) \end{aligned} \tag{18}$$

$$\begin{aligned} S_1(x) = (&1 + x_1 + x_2 + x_3 + x_0x_2 + x_0x_3 + x_1x_3 + x_1x_2x_3,\\ &1 + x_2 + x_3 + x_0x_2 + x_0x_3 + x_2x_3 + x_0x_1x_3,\\ &1 + x_0 + x_3 + x_0x_2 + x_0x_3 + x_1x_3 + x_2x_3 + x_0x_1x_3,\\ &x_1 + x_0x_1 + x_0x_2 + x_0x_3 + x_1x_2 + x_2x_3 + x_0x_1x_3) \end{aligned} \tag{19}$$

$$\begin{aligned} S_2(x) = (&x_0 + x_1 + x_3 + x_0x_1 + x_0x_2 + x_1x_2 + x_0x_3 + x_1x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_0 + x_0x_1 + x_0x_2 + x_1x_2 + x_1x_3 + x_0x_1x_2 + x_1x_2x_3,\\ &x_2 + x_3 + x_0x_2 + x_0x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_1 + x_2 + x_3 + x_0x_1 + x_1x_3) \end{aligned} \tag{20}$$

$$\begin{aligned} S_3(x) = (&1 + x_1 + x_2 + x_0x_1 + x_1x_2 + x_0x_3 + x_1x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_0x_1 + x_1x_2 + x_0x_3 + x_1x_3 + x_1x_2x_3,\\ &1 + x_2 + x_3 + x_0x_3 + x_2x_3 + x_0x_1x_2,\\ &1 + x_0 + x_1 + x_2 + x_0x_1 + x_1x_3) \end{aligned} \tag{21}$$

The quasigroup $(Q_1, *_1)$ can be presented as a Boolean map $*_1 : \mathbb{F}_2^4 \to \mathbb{F}_2^2$, and its ANF is given by:

Table 1. The four S-boxes $S_0, S_1, S_2,$ and S_3 of the set S.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	7	6	A	4	1	2	0	8	B	5	D	C	F	3	E	9
$S_1(x)$	E	0	2	B	7	3	A	1	C	D	F	9	4	8	6	5
$S_2(x)$	5	E	6	7	C	A	3	B	9	8	4	F	D	1	2	0
$S_3(x)$	F	D	4	C	6	9	1	0	E	2	5	3	A	B	7	8

Table 2. The Latin square of $(Q_1, *_1)$.

$*_1$	0	1	2	3
0	2	0	1	3
1	0	1	3	2
2	1	3	2	0
3	3	2	0	1

$$(x_0, x_1)*_1(y_0, y_1) = (1 + x_0 + x_1 + y_0 + y_1 + x_1y_1, x_0 + y_0 + x_1y_1) \tag{22}$$

The algebraic degree of $(Q_1, *_1)$ is 2, hence it is a nonlinear quasigroup.

The storage space required by the 16-order quasigroup used in the block cipher INRU (Tiwari et al. 2021) is 128 bytes, and that required by the 8×8 S-box used in the AES cipher is 256 bytes. However, the storage space required by the set S of four S-boxes and the quasigroup of order 4 used in our cipher is just 36 bytes, or approximately one-fourth and one-seventh that of the INRU and AES ciphers, respectively. The nonlinear quasigroup of order 4 and the four cryptographically strong S-boxes used in our design can be found much more easily than the polynomially complete quasigroup of order 16 used in INRU. Also, our cipher comprises only 10 rounds, and because of the small number of rounds, we get a relatively fast cipher.

4.3. Round key generation

In this step, we generate 11 64-bit round keys using an initial 128-bit long key (secret key), an initial value of length 64-bit, the ordered set S of four cryptographically strong S-boxes of order 4, and the quasigroup $(Q_1, *_1)$ of order 4. Let the initial key be $IK = (IK_0, IK_2, ..., IK_{n-1})$ and the initial value be $iv = (iv_0, iv_1, ..., iv_{31})$, where $n = 64$, IK_i is a 2-bit word from $\mathbb{F}_2^2$ $(0 \le i \le n-1)$ and iv_i is a 2-bit word from $\mathbb{F}_2^2$ $(0 \le i \le 31)$. Let $s = n \times 11$. First, the key IK is extended to $K\_ex = (K\_ex_0, K\_ex_1, ..., K\_ex_{s-1})$, where

$$K\_ex_i = \begin{cases} IK_{i \bmod n}, 0 \le i \le 2n-1 \text{ or } 8n \le i \le 10n-1, \\ iv_{i \bmod n}, 2n \le i \le 2n+31 \text{ or } 10n \le i \le 10n+31, \\ i \bmod 4, \text{ elsewhere.} \end{cases} \tag{23}$$

Initially, we apply the left e-transformation on K_ex over the quasigroup $(Q_1, *_1)$ with leader IK_0. Then, with leader IK_i, we apply the left and right string transformations over the ordered set S' of S-boxes obtained by swapping the IK_{i-1}^{th} and IK_{i+1}^{th} S-boxes in the ordered set S on K_ex alternatively for even and odd values of i, for $i = 1, 2, ..., n-2$. Next, we apply the right e-transformation over the quasigroup $(Q_1, *_1)$ with leader IK_{n-1} on K_ex. Finally, we obtain the 11 round keys from K_ex as: $RK_i = (K\_ex_{n\times i+0}, K\_ex_{n\times i+2}, ..., K\_ex_{n\times i+62}), 0 \le i \le 10$. The pseudo-algorithm of the round key generation is given in Algorithm 1.

Algorithm 1 Round key generation algorithm

Inputs:

(1) $IK = (IK_0, IK_1, ..., IK_{n-1})$– the initial key ($IK_i$ are 2-bit words)

(2) $iv = (iv_0, iv_1, ..., iv_{31})$– the initial value in 2-bit words

Outputs:

(1) Round key $RK = (RK_0, RK_1, ..., RK_{10})$

function Gen_RK(IK)
  $s := n \times 11$
  for $i = 0$ to $s - 1$ **do**
    if $i \in [0, 2n-1]$ or $i \in [8n, 10n-1]$ **then**
      $K\_ex_i := IK_{i \bmod n}$
    else if $i \in [2n, 2n+31]$ or $i \in [10n, 10n+31]$ **then**
      $K\_ex_i := iv_{i \bmod n}$
    else
      $K\_ex_i := i \bmod 4$
    end if
  end for
  $l := IK_0$
  $K\_ex := \mathcal{L}e_{l,(Q_1,*_1)}(K\_ex)$
  for $i = 1$ to $n-2$ **do**
    $l := IK_i$
    $S' := Swap(S, IK_{i-1}, IK_{i+1})$ ▷ S' obtained by swapping the IK_{i-1}^{th} and IK_{i+1}^{th} S-box in S
    if i is even **then**
      $K\_ex := L\varepsilon_{l,(Q,S')}(K\_ex)$
    else
      $K\_ex := R\varepsilon_{l,(Q,S')}(K\_ex)$
    end if
  end for
  $l := IK_{n-1}$
  $K\_ex := \mathcal{R}e_{l,(Q_1,*_1)}(K\_ex)$
  for $i = 0$ to 10 **do**
    $RK_i := (K\_ex_{n\times i+0}, K\_ex_{n\times i+2}, ..., K\_ex_{n\times i+62})$
  end for
  return $RK := (RK_0, RK_1, ..., RK_{10})$
end function

4.4. Encryption and decryption algorithms

Let M be a plaintext divided into N 64-bit blocks. Let $M^{(i)}$ be an i^{th} block of plaintext divided into 4-bit words from $\mathbb{F}_2^4$. Let $M^{(i)} = M_0^{(i)}, M_1^{(i)}, ..., M_{15}^{(i)}$. The 11 round keys $RK_0, RK_1, ..., RK_{10}$, each consisting of 64 bits, are generated using the round key generation algorithm as discussed in Section 4.3. Here, RK_{10} is the post-whitening key, that is, the XOR after the last round of encryption. Each round of the design consists of key mixing followed by a string transformation over the set S of S-boxes, an elementary transformation over quasigroup $(Q_1, *_1)$ that provides confusion in the design, and an elementary transformation over quasigroup $(Q_2, *_2)$ that provides diffusion in the design. The string transformation over the set S of S-boxes and the elementary string transformation over quasigroup $(Q_1, *_1)$ are nonlinear. However, the elementary string transformation over quasigroup $(Q_2, *_2)$ is linear. The left string transformation over the set S of S-boxes followed by right and left e-transformations over quasigroups $(Q_1, *_1)$ and $(Q_2, *_2)$, respectively, are applied in odd rounds, and the right string transformation over the set S of S-boxes followed by left and right e-transformations over quasigroups $(Q_1, *_1)$ and $(Q_2, *_2)$, respectively, are applied in even rounds. The leaders (denoted by l_1 and l_2) used in the first and second transformations in round r are the first two and last two bits of the round key RK_{r-1}, respectively.

Let $XOR : \mathbb{F}_2^{64} \times \mathbb{F}_2^{64} \to \mathbb{F}_2^{64}$, which denotes the function for XORing the round key, be defined as:

$$XOR(k_0, k_1, ..., k_{63}; x_0, x_1, ..., x_{63}) = (k_0 \oplus x_0, k_1 \oplus x_1, ..., k_{63} \oplus x_{63}), \quad (24)$$

where $k_0, k_1, ..., k_{63}$ and $x_0, x_1, ..., x_{63} \in \mathbb{F}_2^{64}$. The left and right string transformations $L\varepsilon_{l_1,(Q,S)}$ and $R\varepsilon_{l_1,(Q,S)}$ (as confusion layer) are functions on Q^{16}, where $Q = \mathbb{F}_2^4$. These transformations are defined in Equations (14) and (15). The other confusion transformations, $\mathcal{L}e_{l_2,(Q_1,*_1)}$ and $\mathcal{R}e_{l_2,(Q_1,*_1)}$, are functions on Q_1^{32}. And the last transformation layer for diffusion, $\mathcal{L}e_{1,(Q_2,*_2)}$ and $\mathcal{R}e_{0,(Q_2,*_2)}$, is a function on Q_2^{64}. These quasigroup transformations are defined in Equations (6) and (8) (for $n = 32$ and $n = 64$, respectively). If $\mathfrak{R}_{\text{odd}}$ and $\mathfrak{R}_{\text{even}}$ denote the odd and even round functions of the encryption algorithm, then mathematically, these functions can be written as:

$$\mathfrak{R}_{\text{odd}} = \mathcal{L}e_{1,(Q_2,*_2)} \circ \mathcal{R}e_{l_2,(Q_1,*_1)} \circ L\varepsilon_{l_1,(Q,S)} \circ XOR \quad (25)$$

and

$$\Re_{\text{even}} = \mathcal{R}e_{0,(Q_2,*_2)} \circ \mathcal{L}e_{l_2,(Q_1,*_1)} \circ \mathcal{R}e_{l_1,(Q,S)} \circ XOR. \tag{26}$$

Note that $\circ$ denotes the usual composition function. Algorithm 2 describes the encryption function in the form of pseudo-code. The workflow of the encryption algorithm is shown in Figure 9. The encryption of an i^{th} block of an odd round r of the design is shown in Figure 10. A similar figure can be drawn for an i^{th} block of an even round of the design. In total, there are 10 rounds in the block cipher design.

Algorithm 2 Encryption algorithm

Inputs:

(1) $M-$ the plaintext with elements from Q, where $Q = \mathbb{F}_2^4$. Let $M = M_0, M_1, ..., M_{N-1}$ be the N number of blocks to be encrypted, where each M_i is of 64-bit long.

(2) Round key $RK = (RK_0, RK_1, ..., RK_{10})$

Outputs:

(1) Ciphertext $C = C_0, C_1, ..., C_{N-1}$

for $i = 0$ to $N - 1$ **do**
  $C_i := M_i$
  for $r = 1$ to 10 **do**
    $l_1 :=$ first two bits of RK_{r-1}
    $l_2 :=$ last two bits of RK_{r-1}
    $C_i := XOR(C_i, RK_{r-1})$
    if r is odd **then**
      $C_i := L\varepsilon_{l_1,(Q,S)}(C_i)$
      $C_i := \mathcal{R}e_{l_2,(Q_1,*_1)}(C_i)$
      $C_i := \mathcal{L}e_{1,(Q_2,*_2)}(C_i)$
    else
      $C_i := R\varepsilon_{l_1,(Q,S)}(C_i)$
      $C_i := \mathcal{L}e_{l_2,(Q_1,*_1)}(C_i)$
      $C_i := \mathcal{R}e_{0,(Q_2,*_2)}(C_i)$
    end if
  end for
  $C_i := XOR(C_i, RK_{10})$
end for

In the decryption algorithm, we use the string transformations on Q^{16} over the inverse set S^{-1} of S-boxes and string transformations (d-transformations) over the inverse conjugates $(Q_1, \backslash_{*_1})$ and $(Q_2, \backslash_{*_2})$ of the quasigroups $(Q_1, *_1)$ and $(Q_2, *_2)$, respectively. Note that there are two different binary operations, so we have used these operations in the suffix corresponding to each of their conjugates. If $\Re'_{\text{odd}}$ and $\Re'_{\text{even}}$ denote the

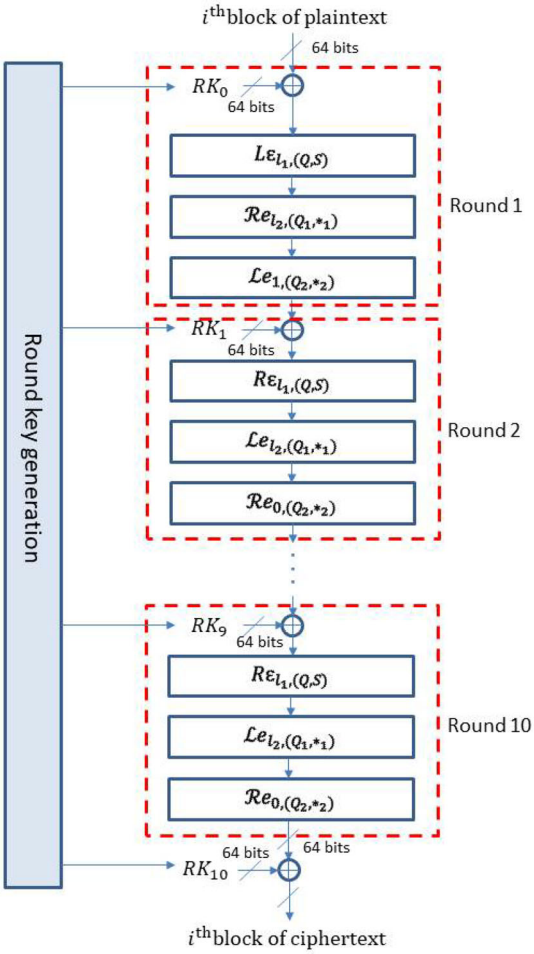

Figure 9. Graphical description of the encryption algorithm.

odd and even round functions of the decryption algorithm, then the round functions can be written mathematically as:

$$\Re'_{\text{odd}} = XOR \circ RD_{l_1,(Q,S^{-1})} \circ \mathcal{L}d_{l_2,(Q_1,\backslash_{*_1})} \circ \mathcal{R}d_{0,(Q_2,\backslash_{*_2})} \tag{27}$$

and

$$\Re'_{\text{even}} = XOR \circ LD_{l_1,(Q,S^{-1})} \circ \mathcal{R}d_{l_2,(Q_1,\backslash_{*_1})} \circ \mathcal{L}d_{1,(Q_2,\backslash_{*_2})}. \tag{28}$$

Algorithm 3 describes the decryption function in the form of pseudo-code. The workflow of the decryption algorithm is shown in Figure 11. The decryption of an i^{th} block of an odd round r of the design is shown in Figure 12. A similar figure can be drawn for the i^{th} block of an even round of the design.

Algorithm 3 Decryption algorithm

Inputs:

(1) $C-$ the ciphertext with elements from Q, where $Q = \mathbb{F}_2^4$. Let $C = C_0, C_1, ..., C_{N-1}$ be the N number of blocks to be decrypted, where each C_i is of 64-bit long.

(2) Round key $RK = (RK_0, RK_1, ..., RK_{10})$

Outputs:

(1) Plaintext $M = M_0, M_1, ..., M_{N-1}$

for $i = 0$ to $N - 1$ **do**

  $M_i := C_i$

  $M_i := XOR(M_i, RK_{10})$

  for $r = 1$ to 10 **do**

    $l_1 :=$ first two bits of RK_{10-r}

    $l_2 :=$ last two bits of RK_{10-r}

    if r is odd **then**

      $M_i := \mathcal{R}d_{0,(Q_2,\backslash_{*_2})}(M_i)$

      $M_i := \mathcal{L}d_{l_2,(Q_1,\backslash_{*_1})}(M_i)$

      $M_i := RD_{l_1,(Q,S^{-1})}(M_i)$

    else

      $M_i := \mathcal{L}d_{1,(Q_2,\backslash_{*_2})}(M_i)$

      $M_i := \mathcal{R}d_{l_2,(Q_1,\backslash_{*_1})}(M_i)$

      $M_i := LD_{l_1,(Q,S^{-1})}(M_i)$

    end if

    $M_i := XOR(M_i, RK_{10-r})$

  end for

end for

5. Security analysis

In this section, the security of the proposed block cipher BCWST is analyzed. The randomizing ability of BCWST is compared with that of AES-128 and the quasigroup-based block cipher INRU in the CBC mode of operation using the NIST test suite. We then provide an avalanche analysis of the keystream and plaintext of BCWST, AES-128, and INRU. In addition, the robustness of BCWST against linear, differential, and algebraic attacks is presented using the standard approach.

5.1. Randomness testing

In this section, the NIST test suite (National Institute of Standards and Technology 2001; Bassham 2010) is used to analyze the randomness of

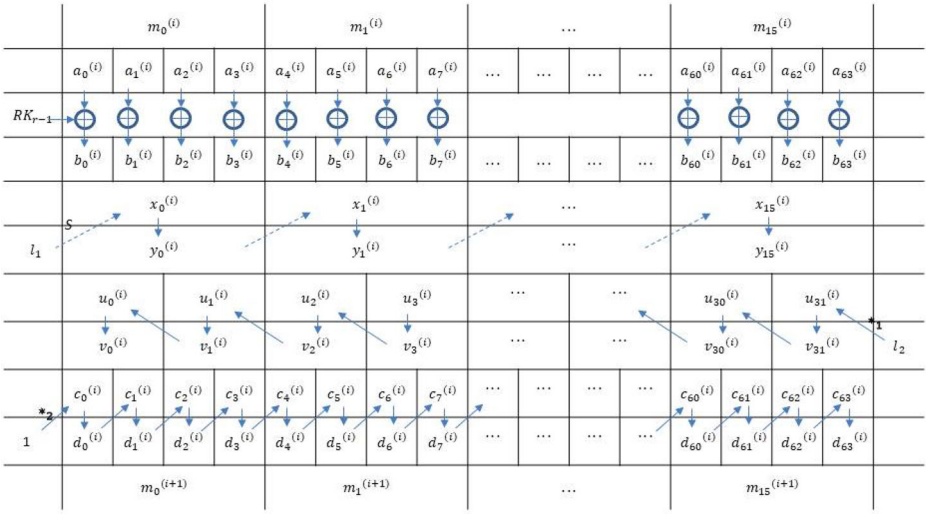

Figure 10. Graphical representation of the i^{th} block of an odd round of the encryption.

BCWST. We compare the results obtained for our design with those obtained for AES-128 and INRU (Tiwari et al. 2021) in the CBC mode of operation for 100 binary sequences of length 2^{20} produced using 100 random 128-bit keys generated by the ANSI X9.31 random number generator. The tests were conducted with a significance level of 99%. Tables 3–5 show the percentage of success and the uniformity of p values obtained from the NIST test suite experiment done with plaintext 0x00, 0xFF, and randomly generated in the ciphers BCWST, INRU, and AES-128, respectively. The p value in the tables determines the uniformity of the p value obtained from the statistical tests when the interval between 0 and 1 is divided into 10 subintervals. These values are obtained from the application of a chi-square test. From these tables, we can conclude that the randomness of the proposed cipher BCWST is comparable to that of INRU and AES-128. Table 9 lists the abbreviations for the NIST test used in Tables 3–5.

5.2. Avalanche criterion

During the security assessment of any cipher, it is desirable to check that whenever a single input bit is flipped (from 0 to 1 or from 1 to 0), an average of half the output bits are also changed, known as the avalanche criterion. The avalanche criterion has been given various mathematical formulations in different contexts, such as block ciphers, Boolean functions, and hash algorithms. In the case of a block cipher, a slight change in the plaintext or the key should result in a significant change in the ciphertext (Gustafson et al. 1994).

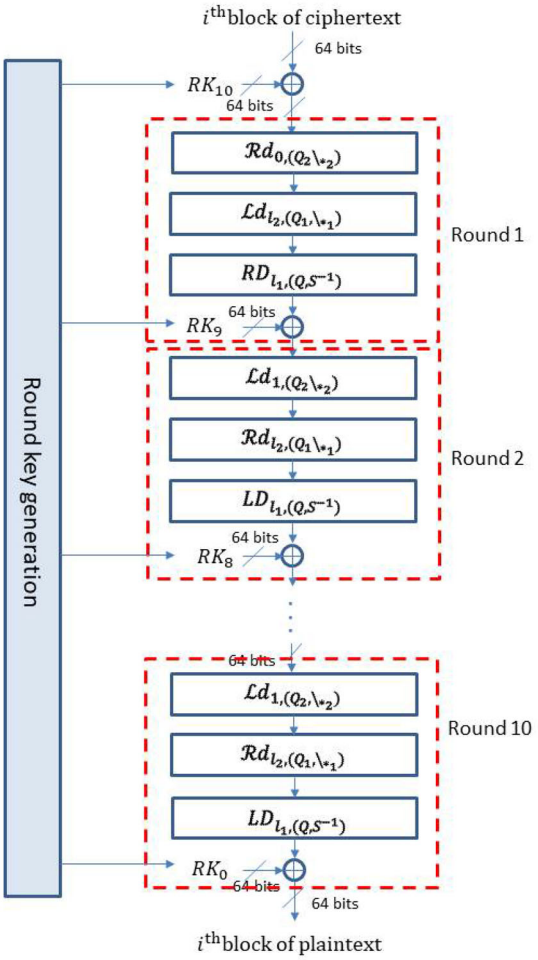

Figure 11. Graphical description of the decryption algorithm.

5.2.1. Avalanche analysis of keystream

In this section, we analyze the percentage change in the ciphertext of BCWST, INRU, and AES-128 whenever a single bit of the secret key is flipped (from 0 to 1 or from 1 to 0). To determine whether BCWST satisfies the avalanche criterion, we produced 100 random secret keys and encrypted a randomly generated plaintext 64,000 bits long using each of these keys. We then encrypted the random plaintext using the key obtained by altering a specific bit in each of the secret keys. We compared the obtained ciphertexts with the original ciphertexts to compute the percentage change in the three ciphers BCWST, INRU, and AES-128. Every bit in each of the secret keys was altered throughout the procedure over the length of the key, and some of the outcomes of the experiment are shown in Table 6. Each cell of the table represents the average percentage change in the ciphertext when the i^{th} bit of all 100 keys is flipped, where i is a

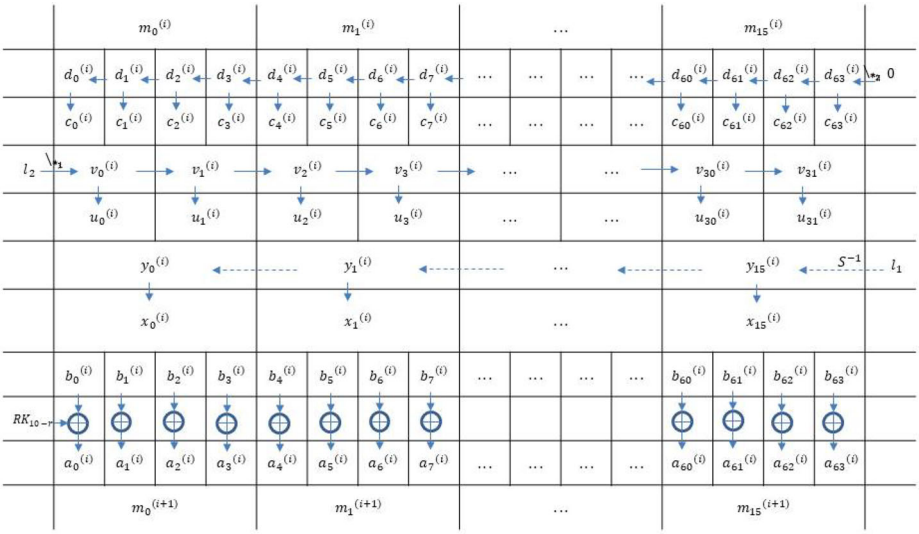

Figure 12. Graphical representation of the i^{th} block of an odd round of the decryption.

position in the key, $1 \leq i \leq 128$. For instance, the table indicates that when the 4^{th} bit of the keys is flipped, the percentage change in the ciphertext is 50.001 in BCWST, 49.918 in INRU, and 49.984 in AES-128. According to the experimental results obtained in Table 6, we can conclude that the ciphertext is changed with a probability of around 50% when an arbitrary bit of the keystream is changed. Hence, the proposed cipher satisfies the avalanche criterion. The minimum and maximum average percentages obtained in BCWST, INRU, and AES-128 are shown in the last two rows of Table 6.

5.2.2. Avalanche analysis of plaintext

We produced a random secret key and encrypted 10,000 different randomly generated plaintexts of the same length (block size) with this secret key to determine whether the three ciphers satisfied the avalanche criterion. We then encrypted the plaintexts obtained by altering a specific bit in each of the randomly generated plaintexts using the random secret key. The results were compared with the original ciphertexts, and we computed the average percentage change in the ciphertext when the i^{th} bit of the produced plaintexts (10,000 in number) were flipped, for all i from 1 to the block length. In this process, every bit in each of these plaintexts was changed one by one. The minimum and maximum average percentages obtained in BCWST, INRU, and AES-128 are shown in Table 7.

From Tables 6 and 7 we can conclude that BCWST has approximately the same diffusion effect as AES-128 and INRU, with relatively less space complexity.

Table 3. Percentage of success and uniformity of p values obtained from NIST tests for BCWST.

	Plaintext 0x00		Plaintext 0xFF		Random plaintext	
Test	Success %	p value	Success %	p value	Success %	p value
Freq	99	0.9717	99	0.9558	100	0.1025
BF	100	0.6579	100	0.3345	100	0.6371
Runs	98	0.9781	100	0.4012	99	0.1626
LRO	100	0.6787	100	0.3346	100	0.5543
Rank	100	0.9357	99	0.1719	99	0.0457
DFT	100	0.7598	97	0.9643	98	0.4944
NOTM (148 templates)	96–100	0.004–0.9943	96–100	0.0046–0.9915	96–100	0.0134–0.9978
OTM	98	0.5749	98	0.6163	100	0.5749
MUS	98	0.7981	99	0.5141	100	0.475
LC	99	0.2133	98	0.6993	99	0.5341
Serial 1	100	0.1816	98	0.0205	99	0.4559
Serial 2	99	0.7981	98	0.1373	99	0.4011
AE	100	0.1154	100	0.6993	100	0.0146
CSF	98	0.7598	99	0.8832	99	0.4373
CSB	99	0.5141	99	0.6579	98	0.4373
RE (8 states)	97.18–100	0.2053–0.8810	96.88–100	0.1480–0.8623	98.46–100	0.0635–0.922
REV (18 states)	95.77–100	0.3863–0.9643	98.44–100	0.0822–0.9114	96.92–100	0.0007–0.9809

Table 4. Percentage of success and uniformity of p values obtained from NIST tests for INRU.

	Plaintext 0x00		Plaintext 0xFF		Random plaintext	
Test	Success %	p value	Success %	p value	Success %	p value
Freq	100	0.4944	99	0.6371	100	0.1626
BF	99	0.8513	100	0.978	100	0.4943
Runs	99	0.6163	99	0.0519	100	0.6579
LRO	99	0.5141	99	0.2368	98	0.2896
Rank	99	0.5141	99	0.0519	99	0.1916
DFT	99	0.1537	98	0.3505	98	0.4372
NOTM (148 templates)	96–100	0.004–0.9978	96–100	0.0076–0.9914	96–100	0.0066–0.9963
OTM	98	0.6163	99	0.9114	99	0.0711
MUS	99	0.0668	99	0.1223	98	0.1025
LC	99	0.6786	100	0.1537	99	0.1025
Serial 1	100	0.3669	99	0.3505	99	0.0965
Serial 2	99	0.4559	99	0.2757	100	0.5141
AE	98	0.3041	100	0.4749	100	0.7981
CSF	98	0.4190	98	0.4372	100	0.0046
CSB	99	0.9879	100	0.9558	99	0.3345
RE (8 states)	98–100	0.0965–0.8832	98.21–100	0.00003–0.9114	98.46–100	0.0956–0.9411
REV (18 states)	96–100	0.1223–0.9914	94.64–100	0.0020–0.9558	95.38–100	0.0704–0.9705

5.3. Linear cryptanalysis

Linear cryptanalysis (Heys 2002; Matsui 1993) is a plaintext attack in which the attacker has knowledge of the set of plaintexts and their corresponding ciphertexts but there is no way to select which plaintexts and corresponding ciphertexts are available. In linear cryptanalysis, the attacker tries to approximate the nonlinear part of the design (the S-boxes) by using the linear relationship between the output bits obtained at a certain round with the key bits and the plaintext bits.

The conventional linear trail is not useful because the S-box is not fixed. Therefore, similar to the block cipher INRU, we opt for a standard

Table 5. Percentage of success and uniformity of p values obtained from NIST tests for AES-128.

	Plaintext 0x00		Plaintext 0xFF		Random plaintext	
Test	Success %	p value	Success %	p value	Success %	p value
Freq	97	0.5141	100	0.1626	99	0.8165
BF	98	0.0135	99	0.8677	99	0.9914
Runs	97	0.6371	99	0.4373	98	0.4190
LRO	99	0.4559	100	0.3669	100	0.9716
Rank	97	0.0329	100	0.5955	98	0.2896
DFT	95	0.6579	95	0.6163	99	0.7197
NOTM (148 templates)	95–100	0.0102–0.9942	95–100	0.0179–0.9781	95–100	0.0034–0.9914
OTM	99	0.0167	99	0.1626	98	0.6993
MUS	98	0.5141	100	0.4943	100	0.8676
LC	99	0.9357	100	0.9463	100	0.9114
Serial 1	100	0.2897	100	0.3505	98	0.2133
Serial 2	99	0.1626	100	0.2757	99	0.9357
AE	99	0.8514	95	0.3345	99	0.0220
CSF	99	0.2023	100	0.7399	99	0.9988
CSB	99	0.7399	99	0.6993	100	0.6371
RE (8 states)	97.10–100	0.0151–0.7565	98.55–100	0.0514–0.8755	98.24–100	0.0008–0.5544
REV (18 states)	94.2–100	0.0151–0.8486	97.1–100	0.0190–0.7887	98.24–100	0.0329–0.9879

Table 6. Avalanche effect of keystream.

	% change in ciphertext		
Position in key (in bits)	BCWST	INRU	AES-128
1	50.008	49.977	50.008
2	50.006	50.006	50.004
3	49.989	49.968	50.019
4	50.001	49.918	49.984
5	50.012	49.956	50.002
9	50.003	50.0236	49.994
13	49.999	49.918	50.008
27	49.994	50.064	49.968
34	50.002	50.146	50.005
55	49.957	50.037	49.985
63	49.990	50.076	50.023
80	49.993	50.051	49.967
97	49.978	49.973	49.998
100	50.006	49.961	50.003
106	50.004	49.965	50.005
111	50.002	50.065	50.001
125	50.005	49.963	49.970
126	50.011	50.102	50.012
127	50.012	49.964	49.997
128	50.015	50.020	49.994
Min	49.931	49.822	49.937
Max	50.054	50.164	50.046

approach and show that even at the end of the second round, the linear approximation is not possible for the nonlinear component of the design. To add confusion to our design, we used left and right string transformations over the ordered set S of four cryptographically strong S-boxes of order 4 and left and right e-transformations over quasigroup $(Q_1, *_1)$ of order 4. For a fixed element l of Q_1, the binary operation $*_1$ of the quasigroup $(Q_1, *_1)$ can be considered as a 2×2 S-box, as follows:

Table 7. Avalanche effect of plaintext.

	Min % (over length of plaintext)	Max % (over length of plaintext)
BCWST	49.959	50.041
INRU	49.943	50.042
AES-128	49.957	50.032

Table 8. The Latin square of $(Q_2, *_2)$.

$*_2$	0	1
0	1	0
1	0	1

$$S_l : Q_1 \to Q_1, \quad x \mapsto l *_1 x, \quad \forall\ x \in Q_1.$$

This implies that the rows of the quasigroup $(Q_1, *_1)$ give four 2×2 S-boxes, $S_0', S_1', S_2',$ and S_3'. Hence, the left and right string transformations over the set S of S-boxes of order 4 can be considered a layer of 16 4×4 S-boxes, and the left and right e-transformations over the quasigroup $(Q_1, *_1)$ with a fixed leader can be considered a layer of 32 2×2 S-boxes. In each of these layers, the output of an S-box determines the selection of the subsequent S-box. Hence, the S-box layers corresponding to left and right string transformations over both the set S of S-boxes and the quasigroup are output dependent.

Let $m_0, m_1, ..., m_{63}$ be an input block of plaintext and $RK_0 = rk_0^0, rk_1^0, ..., rk_{63}^0$. Then we have:

$$a_0, a_1, ..., a_{63} = m_0 \oplus rk_0^0, m_1 \oplus rk_1^0, ..., m_{63} \oplus rk_{63}^0, \tag{29}$$

where $a_0, a_1, ..., a_{63}$ is the output block obtained after key XORing. If (x_0, x_1, x_2, x_3) and (y_0, y_1, y_2, y_3) are the nonzero input-output masks corresponding to the first 4×4 S-box, then we get the following relation:

$$y_0 b_0 \oplus y_1 b_1 \oplus y_2 b_2 \oplus y_3 b_3 = x_0 a_0 \oplus x_1 a_1 \oplus x_2 a_2 \oplus x_3 a_3, \tag{30}$$

where $b_0, b_1, ..., b_{63}$ is the output of the left string transformation over the set S with leader $l_1 \in Q$, i.e., $L\varepsilon_{l_1,(Q,S)}(a_0, a_1, ..., a_{63}) = (b_0, b_1, ..., b_{63})$.

If $c_0, c_1, ..., c_{63}$ is the output of the right e-transformation over the quasigroup $(Q_1, *_1)$ with leader l_2, i.e., $\mathcal{R}e_{l_2,(Q_1,*_1)}(b_0, b_1, ..., b_{63}) = (c_0, c_1, ..., c_{63})$, then we have:

$$t_0 c_0 \oplus t_1 c_1 = z_0 b_0 \oplus z_1 b_1 \tag{31}$$

and

$$t_2 c_2 \oplus t_3 c_3 = z_2 b_2 \oplus z_3 b_3 \tag{32}$$

where $(z_0, z_1), (t_0, t_1)$ and $(z_2, z_3), (t_2, t_3)$ are input-output masks of the 31st and 32nd S-boxes, respectively.

If $d_0, d_1, ..., d_{63}$ is the output of the left e-transformation over the quasigroup $(Q_2, *_2)$ with leader 1, i.e., $\mathcal{L}e_{1,(Q_2,*_2)}(c_0, c_1, ..., c_{63}) = (d_0, d_1, ..., d_{63})$, then we get the following relations for the first 4 bits:

$$d_0 = 1*_2 c_0, d_1 = d_0*_2 c_1, d_2 = d_1*_2 c_2 \text{ and } d_3 = d_2*_2 c_3. \tag{33}$$

Suppose the quasigroup $(Q_2, *_2)$ is as given in Table 8:

Clearly, the ANF of $(Q_2, *_2)$ is $x_1*_2 x_2 = 1 + x_1 + x_2$. Hence, Equation (33) takes the form:

$$d_0 = c_0, d_1 = c_0 \oplus c_1 \oplus 1, d_2 = c_0 \oplus c_1 \oplus c_2, d_3 = c_0 \oplus c_1 \oplus c_2 \oplus c_3 \oplus 1. \tag{34}$$

Let $RK_1 = rk_0^1, rk_1^1, ..., rk_{63}^1$. Then, after XORing with the output of the first round, we get:

$$a'_0, a'_1, ..., a'_{63} = d_0 \oplus rk_0^1, d_1 \oplus rk_1^1, ..., d_{63} \oplus rk_{63}^1. \tag{35}$$

Similarly in the second round, if (x'_0, x'_1, x'_2, x'_3) and (y'_0, y'_1, y'_2, y'_3) are the nonzero input-output masks corresponding to the last (16^{th}) 4×4 S-box, then we get the following relation:

$$y'_0 b'_0 \oplus y'_1 b'_1 \oplus y'_2 b'_2 \oplus y'_3 b'_3 = x'_0 a'_0 \oplus x'_1 a'_1 \oplus x'_2 a'_2 \oplus x'_3 a'_3, \tag{36}$$

where $b'_0, b'_1, ..., b'_{63}$ is the output of the right string transformation over the set S with leader $l_1 \in Q$.

If $c'_0, c'_1, ..., c'_{63}$ is the output of the left e-transformation over quasigroup $(Q_1, *_1)$ with leader l_2, then for arbitrary nonzero input-output masks $(z'_0, z'_1), (t'_0, t'_1)$ and $(z'_2, z'_3), (t'_2, t'_3)$ corresponding to the first and second S-boxes, respectively, we get:

$$t'_0 c'_0 \oplus t'_1 c'_1 = z'_0 b'_0 \oplus z'_1 b'_1, \tag{37}$$

and

$$t'_2 c'_2 \oplus t'_3 c'_3 = z'_2 b'_2 \oplus z'_3 b'_3. \tag{38}$$

If $d'_0, d'_1, ..., d'_{63}$ is the output of the left e-transformation over the quasigroup $(Q_2, *_2)$ given in Table 8 with leader 0, then we have the following relations:

$$d'_3 = \oplus_{i=3}^{63} c'_i \oplus 1, d'_2 = \oplus_{i=2}^{63} c'_i, d'_1 = \oplus_{i=1}^{63} c'_i \oplus 1 \text{ and } d'_0 = \oplus_{i=0}^{63} c'_i. \tag{39}$$

The description of linear cryptanalysis is shown graphically in Figure 13. When we try to arrive at an approximation of input and output variables directly and try to find the bias, we encounter intermediate variables. We tried to eliminate the intermediate variables, but since the number of equations is far less than the number of intermediate variables, the process of elimination is computationally hard. In particular, at the end of the second round, we get equations with intermediate variables that are computationally infeasible to eliminate in terms of plaintext bits and key bits.

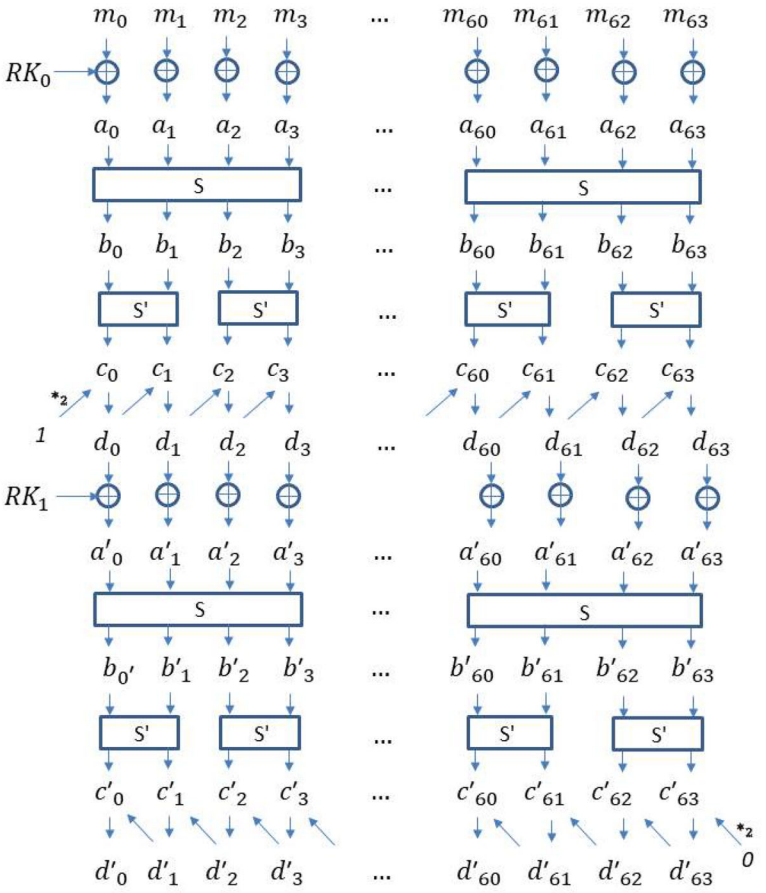

Figure 13. Graphical representation of linear cryptanalysis.

Therefore, it is not possible to obtain a linear approximation, and the design is resistant to linear attack.

5.4. Differential cryptanalysis

Differential cryptanalysis (Biham and Shamir 1991; Heys 2002) is a plaintext attack in which the attacker has the ability to choose plaintexts and examine their corresponding ciphertexts in an attempt to derive the secret key. In any block cipher design with block size n, given a particular input difference, the probability of a particular output difference occurring should not be greater than $\frac{1}{2^n}$.

Here, we show that our design does not have a differential distinguisher. Let m and m' be plaintexts with difference $\Delta m = m \oplus m'$, such that only their last 4 bits differ, that is, $m = 0\text{x}m_0, m_1, ..., m_{14}, m_{15}, m' = 0\text{x}m_0, m_1, ..., m_{14}, m'_{15}$ and $\Delta m = m \oplus m' = 0\text{x}0, 0, ..., 0, \Delta a$. This difference Δm passes through the

XOR layer of the round key without any change. After applying the left string transformation layer over the set S of S-boxes with leader l_1 from the left, we get the output difference of the form $0\text{x}\ 0, \ldots, 0\Delta b$, where $\Delta b \neq 0$. Let $0\text{x}b_0, b_1, \ldots, b_{14}, b_{15}$ and $0\text{x}b_0, b_1, \ldots, b_{14}, b'_{15}$ be the respective output of the left string transformation layer over S, where $b_{15} \neq b'_{15}$. Let $b_i = c_{2i}, c_{2i+1}, 0 \leq i \leq 15$ and $b'_{15} = c'_{30}, c'_{31}$. Equivalently, we have $c_0, c_1, \ldots, c_{29}, c_{30}, c_{31}$ and $c_0, c_1, \ldots, c_{29}, c'_{30}, c'_{31}$ as the input of the right e-transformation layer, where $c_i, c'_i \in Q_1 = \mathbb{F}_2^2$. Since $(c_{30}, c_{31}) \neq (c'_{30}, c'_{31})$, then $c_i \neq c'_i$ for at least one $i \in \{30, 31\}$. Let us suppose that $c_{31} \neq c'_{31}$. Thus, when the right e-transformation layer over the quasigroup $(Q_1, *_1)$ is applied from the right with leader l_2, we have input difference $(0, \Delta c_{31})$ corresponding to the first S-box (from the right), which gives a nonzero output difference Δd_{31}. Similarly, the second S-box has input difference $(\Delta d_{31}, \Delta c_{30})$ and produces output difference Δd_{30}. The next outputs are obtained from two different S-boxes, say S_i and S_j, from two different rows i and j of quasigroup $(Q_1, *_1)$, but they can be from the same column of its Latin square. Therefore, based on the structure of the Latin square, no element in the column of the Latin square of $(Q_1, *_1)$ repeats; hence, we definitely have $d_{30} \neq d'_{30}$ when $c_{30} = c'_{30}$. Continuing in the same way and using the same arguments, we get $d_j \neq d'_j$, for $j = 29, 28, \ldots, 1, 0$. Thus, $d_j \neq d'_j, \forall\ 0 \leq j \leq 31$. This shows that all the 2×2 S-boxes are active only in the first round. However, when $c_{30} \neq c'_{30}$, d_{30} and d'_{30} may or may not be distinct. If $d_{30} \neq d'_{30}$, we are done; that is, all the 2×2 S-boxes are active only in the first round. Otherwise, all the S-boxes are activated in the second round. For the case $c_{31} = c'_{31}$, we get $d_{31} = d'_{31}$ and $c_{30} \neq c'_{30}$. This implies $d_{30} \neq d'_{30}$, which means $d_{29} \neq d'_{29}$. Continuing in a similar way, we get $d_j \neq d'_j, \forall\ 0 \leq j \leq 30$. Thus, all the S-boxes are activated only in the first round. This is shown graphically in Figure 14. Thus, our design is resistant to differential attack.

5.5. Algebraic cryptanalysis

The algebraic cryptanalysis (Bard 2009) of our design was performed similarly to the analysis of the PRESENT block cipher (Bogdanov et al. 2007). Any 4×4 S-box can be described by at least 21 multivariate quadratic equations in the eight input/output-bit variables over $\mathbb{F}_2$. All four 4×4 S-boxes used in our cipher are described by 21 such equations. Thus, all the 4×4 S-boxes in the entire block cipher can be described by $e_1 = n_1 \times 21$ quadratic equations in $v_1 = n_1 \times 8$ variables, where n_1 is the number of 4×4 S-boxes in the encryption and key generation algorithms. In our cipher, we have $n_1 = (10 \times 16) + (31 \times 176)$, or $n_1 = 5616$. Thus, we get $e_1 = 117936$ and $v_1 = 44928$. It is well known that any 2×2 S-box can be

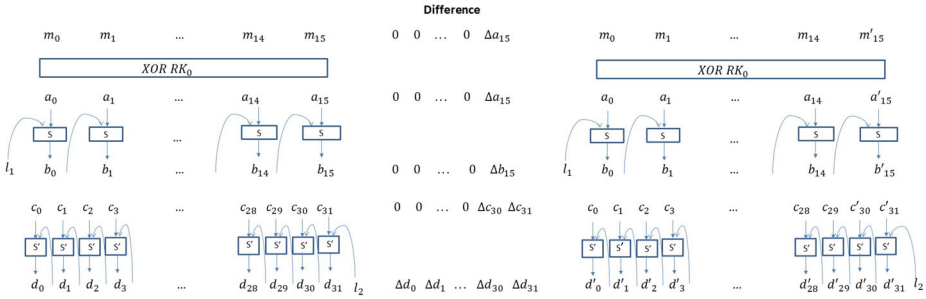

Figure 14. Graphical representation of differential cryptanalysis.

described by at least five multivariate quadratic equations in the four input/output-bit variables over $\mathbb{F}_2$. In our cipher, all four 2×2 S-boxes used as part of the quasigroup of order 4 are described by five such equations. Thus, all the 2×2 S-boxes in the entire block cipher can be described by $e_2 = n_2 \times 5$ quadratic equations in $v_2 = n_2 \times 4$ variables, where n_2 is the number of 2×2 S-boxes in the encryption and key generation algorithms. In our cipher, we have $n_2 = (10 \times 32) + (2 \times 352)$, or $n_2 = 1024$. Thus, we get $e_2 = 5120$ and $v_2 = 4096$. Hence, the entire system consists of $e_1 + e_2 = 123056$ quadratic equations in $v_1 + v_2 = 49024$ variables, which greatly exceeds the number for the PRESENT cipher (which has 1,1067 quadratic equations in 4,216 variables). Because the problem of solving a system of multivariate quadratic equations is NP-hard, the authors of the PRESENT cipher believe that it has resistance against an algebraic attack. Thus, we strongly believe the same is true of our design.

6. Conclusions and future work

In this paper, we proposed a block cipher based on string transformations with 10 rounds and for a key size of 128 bits. We used the NIST test suite for statistical analysis to verify the randomness of the algorithm, and these statistical experimental results were compared with the AES-128 and INRU ciphers. Further, using the standard approach, we showed that our design is resistant to linear, differential, and algebraic cryptanalysis. An avalanche analysis of keystream and plaintext was also performed, and the diffusion produced by our encryption algorithm was compared with that of AES-128 and INRU. From the security analysis, we can conclude that, with approximately one-fourth and one-seventh the space complexity of AES-128 and INRU, respectively, our cipher provides a roughly equivalent level of security.

Our design used an ordered set of four cryptographically strong S-boxes of order 4 and a nonlinear quasigroup of order 4. The security of the

proposed design can be tested with a selection of different quasigroups to customize the algorithm for different applications.

Appendix

Table 9. Abbreviations used in NIST statistical tests.

Abbreviation	Test
Freq	Frequency
BF	Block frequency
LRO	Longest run of ones in a block
Rank	Binary matrix rank
DFT	Discrete Fourier transform
NOTM	Nonoverlapping template matching
OTM	Overlapping template matching
MUS	Maurer's universal statistical
LC	Linear complexity
AE	Approximate entropy
CSF	Cumulative sum forward
CSB	Cumulative sum backward
RE	Random excursions
REV	Random excursions variant

Disclosure statement

The authors declare that they have no conflict of interest.

Funding

This research was supported by the University Grants Commission (UGC), Government of India, with reference no.: 20/12/2015(ii)EU-V.

About the authors

Dimpy Chauhan is a research scholar at the Department of Mathematics, University of Delhi, India. She has received her Masters in Mathematics from Panjab University, India. Her research interests are in algebra and its applications, especially cryptography and coding theory.

Indivar Gupta is a senior scientist in Scientific Analysis Group, DRDO, Delhi, India. He completed his Ph.D. from the IIT Delhi, India. His areas of research include computational algebra, number theory, cryptography, information security, and high-performance computing.

P. R. Mishra is a senior scientist in Scientific Analysis Group, DRDO, Delhi, India. He obtained his Ph.D. in number theory from Banaras Hindu University, Varanasi, India. His main research interests are cryptography and number theory.

Rashmi Verma is an associate professor at the Department of Mathematics, Mata Sundri College for Women, University of Delhi, India. She completed her M.Phil. and Ph.D. from the Department of Mathematics, University of Delhi, India. Her research interest is algebraic coding theory.

ORCID

Dimpy Chauhan http://orcid.org/0000-0002-3075-575X
Indivar Gupta http://orcid.org/0000-0003-2801-0254
P. R. Mishra http://orcid.org/0000-0003-0473-6616

References

Artamonov, V. A., S. Chakrabarti, S. Gangopadhyay, and S. K. Pal. 2013. On Latin squares of polynomially complete quasigroups and quasigroups generated by shifts. *Quasigroups Related Systems* 21 (2):117–30.

Bard, G. V. 2009. *Algebraic cryptanalysis*, xxxiv–356. Dordrecht: Springer.

Bassham, L. E. 2010. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, Gaithersburg, MD: National Institute of Standards and Technology. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762.

Battey, M., and A. Parakh. 2012. Efficient quasigroup block cipher for sensor networks. *21st International Conference on Computer Communications and Networks (ICCCN-2012)*, 1–5, IEEE.

Battey, M., and A. Parakh. 2013. An efficient quasigroup block cipher. *Wireless Personal Communications* 73 (1):63–76. doi:10.1007/s11277-012-0959-x.

Belousov, V. D., and G. B. Belyavskaya. 1989. Latin Squares, Quasigroups and Their Applications, Shtiinta, Kishinev, (in Russian).

Biham, E., and A. Shamir. 1991. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology* 4 (1):3–72. doi:10.1007/BF00630563.

Bogdanov, A., L. R. Knudsen, G. Le, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. 2007. PRESENT: An ultra-lightweight block cipher. In *Proceedings of CHES 2007, Lecture Notes in Computer Science*, ed. P. Paillier and I. Verbauwhede, vol. 4727, 450–66. Berlin: Springer.

Chauhan, D., I. Gupta, P. R. Mishra, and R. Verma. 2022. Construction of cryptographically strong S-boxes from ternary quasigroups of order 4. *Cryptologia* 46 (6):525–51. doi: 10.1080/01611194.2021.1934915.

Chauhan, D., I. Gupta, and R. Verma. 2021. Quasigroups and their applications in cryptography. *Cryptologia* 45 (3):227–65. doi:10.1080/01611194.2020.1721615.

Cooper, J., D. Donovan, and J. Seberry. 1994. Secret sharing schemes arising from Latin squares. *Bulletin of the Institute of Combinatorics and its Applications* 12:33–43.

Daemen, J., and V. Rijmen. 2002. *The design of rijndael, AES-the advanced encryption standard, information security and cryptography*, xviii–238. Berlin: Springer-Verlag.

Dénes, J., and A. D. Keedwell. 1974. *Latin squares and their applications*. New york and London: Academic Press, Inc.

Dimitrova, V. 2010. Quasigroup processing string, their boolean representations and application in cryptography and coding theory. PhD thesis., Ss. Cyril and Methodius University, Skopje, S. Markovski (promotor).

Gligoroski, D., V. Dimitrova, and S. Markovski. 2009. Quasigroups as Boolean functions, their equation systems and Gröbner bases. In *Gröbner bases, coding, and cryptography*, ed. M. Sala, T. Mora, L. Perret, S. Sakata, and C. Traverso. Berlin: Springer-Verlag.

Gligoroski, D., S. Markovski, and L. Kocarev. 2009. Edon-R, An infinite family of cryptographic hash functions. *International Journal Network Security* 8 (3):293–300.

Gustafson, H., E. Dawson, L. Nielsen, and W. Caelli. 1994. A computer package for measuring the strength of encryption algorithm. *Computer and Security* 13 (8):687–97. doi:10.1016/0167-4048(94)90051-5.

Heys, H. M. 2002. A tutorial on linear and differential cryptanalysis. *Cryptologia* 26 (3): 189–221. doi:10.1080/0161-110291890885.

Keedwell, D., and J. Dénes. 2015. *Latin squares and their applications*, 2nd ed. Amsterdam: Elsevier.

Laywine, C. F., and G. L. Mullen. 1998. *Discrete mathematics using Latin squares*. 1st ed., 328. USA: Wiley-Interscience.

Leander, G., and A. Poschmann. 2007. On the classification of 4 bit S-boxes. In *Arithmetic of finite fields*, ed. by C. Carlet, and B. Sunar, vol. 4547, 159–76. Berlin: Springer.

Markovski, S., V. Dimitrova, Z. Trajcheska, M. Petkovska, M. Kostadinoski, and D. Buhov. 2014. Block cipher defined by matrix presentation of quasigroups. *Proceedings of the 11th IEEE International Conference on Informatics and Information Technology*, CIIT Bitola.

Markovski, S., D. Gligoroski, and S. Andova. 1997. Using quasigroups for one-one secure encoding. *Proceedings of the VIII-th conference for logic and computing – LIRA '97*, Novi Sad, 157–167.

Markovski, S., D. Gligoroski, and V. Bakeva. 1999. Quasigroup string processing part 1, contributions. *Secondary Mathematics Technology Science MANU* 20:13–28.

Markovski, S., and A. Mileva. 2009. Nasha–cryptographic hash functions. *NIST The First SHA-3 Candidate Conference*, Leuven, Belgium

Matsui, M. 1993. Linear cryptanalysis method for DES cipher. In *EUROCRYPT '93*, ed. T. Helleseth, vol. 765, 386–97. LNCS, Berlin, Heidelberg: Springer.

Meyer, K. A. 2006. A new message authentication code based on the non-associativity of quasigroups. PhD thesis, Iowa State University, ProQuest LLC, Ann Arbor, MI, 91.

Mihajloska, H., and D. Gligoroski. 2012. Construction of optimal 4-bit S-boxes by quasigroups of order 4. *Proceedings of SECURWARE 2012, The Sixth International Conference on Emerging Security Information, Systems and Technologies*, Rome, Italy, 163–168. IARIA.

Mileva, A. 2010. Cryptographic primitives with quasigroup transformations. Doctoral dissertation, University Sts. Cyril and Methodius, Skopje.

Mileva, A., and S. Markovski. 2008. Correlation matrices and prop ratio tables for quasigroups of order 4. *6th International Conference for Informatics and Information Technology*, CIIT, 17–22.

National Institute of Standards and Technology. 2001. Advanced encryption standard (AES) (FIPS PUB 197). National institute of standards and technology. Federal information processing standards Publication 197 (FIPS 197), November.

Ochodková, E., and V. Snášel. 2001. Using quasigroups for secure encoding of file system. *Proceedings of the international science NATO PfP/PWP conference security and information protection 2001*, Brno, Czech Republic, 175–181.

Parakh, A., and S. Kak. 2009. Online data storage using implicit security. *Information Sciences* 179 (19):3323–31. doi:10.1016/j.ins.2009.05.013.

Tiwari, S. K., A. Awasthi, S. Chkrabarti, and S. Yadav. 2021. INRU: a quasigroup based lightweight block cipher. preprint arXiv:2112.07411.

van Lint, J. H., and R. M. Wilson. 2001. *A course in combinatorics*, 2nd ed., xiv–602. Cambridge: Cambridge University Press.

Vojvoda, M. 2004. Stream ciphers and hash functions-analysis of some new design approaches. PhD thesis, Slovak University of Technology, July.

Wu, C.-K, and D. Feng. 2016. *Boolean functions and their applications in cryptography, Advances in Computer Science and Technology* Heidelberg: Springer.

Yu, Z., and Y. Xu. 2017. Optimal S-boxes based on 3-quasigroups of order 4. *6th international conference on advanced materials and computer science (ICAMCS 2017)*, 114–121, Clausius Scientific Press (CSP).

Zhao, Y., and Y. Xu. 2017. A Lightweight block cipher based on quasigroups. *6th International Conference on Advanced Materials and Computer Science (ICAMCS 2017)*, 123–130, Clausius Scientific Press (CSP).