# Chapter 1

# Preliminaries

This chapter provides a concise overview of quasigroup theory, coding theory and cryptography. We present basic definitions and key results which are essential for understanding the concepts explored in this thesis. For in-depth mathematical background, readers are encouraged to refer (see [9–11, 33, 34, 78, 116, 118]).

## 1.1 Basics of quasigroups and loops

**Definition 1.1.1.** [17] Let $Q$ be a non-empty set and '$\cdot$' a binary operation on $Q$. Then, the structure $(Q, \cdot)$ is called a *groupoid*.

As referred in Figure 1, a groupoid $(Q, \cdot)$ with divisibility property is referred as *quasigroup*. It is formally defined as following:

**Definition 1.1.2.** [116] A non-empty set $Q$ with a binary operation $* : Q \times Q \to Q$ is called a *quasigroup* if for all $a, b \in Q$ there exist unique $x, y \in Q$ satisfying the equations $a * x = b$ and $y * a = b$.

Definition 1.1.2 explicitly indicates that every group is a quasigroup but the converse does not necessarily hold. The non-associative property being a prominent distinction in quasigroups which distinguish them from groups. Consequently, quasigroups are also known as *non-associative groups*. In instances where a quasigroup satisfies the associative law, it becomes a group. Several examples of quasigroups, that are not groups, include $(\mathbb{Z}, -)$, $(\mathbb{Q}\backslash\{0\}, \div)$ and $(\mathbb{R}\backslash\{0\}, \div)$. A quasigroup that possesses an identity element is referred as a *loop*. Throughout this thesis, only finite quasigroups have been considered.

Let $Q$ be a finite set containing $n$ elements. A *Latin square* of order $n$ over $Q$ is an $n \times n$ square matrix whose entries come from $Q$ and each elements appears exactly once in each row and column of the matrix [33].

To establish the ordering of the set of quasigroups in a specified finite order, we use the lexicographic ordering for certain choices of $Q$. This involves arranging the set of rows from the corresponding Latin square into a string of $n^2$ letters. The lexicographic ordering of these representations subsequently determines the ordering of the quasigroups as discussed in [56].

**Definition 1.1.3.** [116] Let $Q$ be a non-empty set and $f$ an $n$-ary operation on $Q$ (i.e., $f : Q^n \to Q$). The $n$-ary groupoid $(Q, f)$, is called an $n$-ary quasigroup if, using the equation $f(x_1, x_2, \ldots, x_n) = x_{n+1}$, the knowledge of any $n$ elements from the set $\{x_1, x_2, \ldots, x_n, x_{n+1}\}$ uniquely determines the remaining element.

When $n = 2$ in the Definition 1.1.3, a 2-ary quasigroup is referred as a *binary quasigroup* (or, simply a *quasigroup*) and is defined in alignment with Definition 1.1.2. Similarly, by considering $n = 3$ in the Definition 1.1.3, a 3-ary quasigroup is known as *ternary quasigroup*.

Given a binary quasigroup $(Q, *)$, we can construct five other quasigroups $(Q, *^{(1\,2)})$, $(Q, *^{(1\,3)})$, $(Q, *^{(2\,3)})$, $(Q, *^{(1\,2\,3)})$ and $(Q, *^{(1\,3\,2)})$ known as the *parastrophes* of the quasigroup $(Q, *)$. The operations, namely $*^{(1\,2)}$, $*^{(1\,3)}$, $*^{(2\,3)}$, $*^{(1\,2\,3)}$ and $*^{(1\,3\,2)}$ can be determined using the given binary operation $*$.

$$x * y = z \Leftrightarrow y *^{(12)} x = z \Leftrightarrow z *^{(13)} y = x \Leftrightarrow x *^{(23)} z = y \Leftrightarrow y *^{(123)} z = x \Leftrightarrow z *^{(132)} x = y$$

The operations denoted as $*^{(1\,2)}$, $*^{(1\,3)}$, $*^{(2\,3)}$, $*^{(1\,2\,3)}$ and $*^{(1\,3\,2)}$ are sometimes represented as $\cdot, /, \backslash, //$ and $\backslash\backslash$ respectively. These operations are also referred as opposite multiplication, right division, left division, opposite right division and opposite left division respectively. The algebra $(Q, *, \backslash, /)$ satisfies the following properties:

$$x * (x\backslash y) = y;\ \ x\backslash(x * y) = y; (y/x) * x = y;\ \ (y * x)/x = y. \tag{1.1}$$

Then, $(Q, \backslash)$ and $(Q, /)$ also form quasigroups.

**Remark 1.1.4.** Sometimes the notations being changed and parastrophe of a quasigroup can be defined as following. For a given binary quasigroup $(Q, f)$ and $\sigma \in S_3$:

$$f^{\sigma}(x_i, x_j) = x_k \iff f(x_{\sigma^{-1}i}, x_{\sigma^{-1}j}) = x_{\sigma^{-1}k} \tag{1.2}$$

For example $f^{(1\,3\,2)}(x_1, x_2) = x_3$ if and only if $f(x_2, x_3) = x_1$.

**Example 1.1.5.** Consider a set $Q = \{a, b, c\}$ and the binary operation $*$ mentioned in Table 1.1.

| $*$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $b$ | $c$ | $a$ |
| $b$ | $a$ | $b$ | $c$ |
| $c$ | $c$ | $a$ | $b$ |

Table 1.1: Latin square corresponding to $*$

Here $(Q, *)$ forms a quasigroup of order 3. The $(1\,2)$, $(1\,3)$, $(2\,3)$, $(1\,2\,3)$ and $(1\,3\,2)$-parastrophe of quasigroup $(Q, *)$ i.e., $(Q, *^{(1\,2)})$, $(Q, *^{(1\,3)})$, $(Q, *^{(2\,3)})$, $(Q, *^{(1\,2\,3)})$ and $(Q, *^{(1\,3\,2)})$, respectively, are described in Table 1.2.

| $*^{(1\,2)}$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $b$ | $a$ | $c$ |
| $b$ | $c$ | $b$ | $a$ |
| $c$ | $a$ | $c$ | $b$ |

| $*^{(1\,3)}$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $b$ | $c$ | $a$ |
| $b$ | $a$ | $b$ | $c$ |
| $c$ | $c$ | $a$ | $b$ |

| $*^{(2\,3)}$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $c$ | $a$ | $b$ |
| $b$ | $a$ | $b$ | $c$ |
| $c$ | $b$ | $c$ | $a$ |

| $*^{(1\,2\,3)}$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $b$ | $a$ | $c$ |
| $b$ | $c$ | $b$ | $a$ |
| $c$ | $a$ | $c$ | $b$ |

| $*^{(1\,3\,2)}$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $a$ | $c$ | $a$ | $b$ |
| $b$ | $a$ | $b$ | $c$ |
| $c$ | $b$ | $c$ | $a$ |

Table 1.2: Parastrophes of quasigroup $(Q, *)$

**Definition 1.1.6.** [116] Let $(Q, \cdot)$ be a groupoid, and $a$ be a fixed element in $Q$. We define the following maps:

- *Left translation map* is defined as $L_a : Q \to Q$ such that $L_a(x) = a \cdot x$ for all $x \in Q$.
- *Right translation map* is defined as $R_a : Q \to Q$ such that $R_a(x) = x \cdot a$ for all $x \in Q$

For a given quasigroup $(Q, \cdot)$, there exist one more translation map on $Q$ referred as *Middle translation map* $P_a : Q \to Q$ and it is defined as $x \cdot P_a x = a$ for all $x \in Q$.

Alternatively, a binary quasigroup can be defined using the Left and Right translation maps [116] in the following definition.

**Definition 1.1.7.** [116] In a groupoid $(Q, *)$, if the left and right translation maps $L_a : Q \to Q$ and $R_a : Q \to Q$ are bijective for all $a \in Q$, then $(Q, *)$ is defined as a quasigroup.

**Remark 1.1.8.** The left translation map $L_a$ (or the right translation map $R_a$) is said to be a bijective map on set $Q$ if and only if, for all $(a, b) \in Q^2$, the equation $a * x = b$ (or $x * a = b$, respectively) has a unique solution $x \in Q$ .

**Definition 1.1.9.** [116] An element $p(x)$ (or, $q(x)$) of a quasigroup $(Q, *)$ is called a left local identity element (or, a right local identity element) of an element $x \in Q$, if $p(x) * x = x$ (or, $x * q(x) = x$). Alternatively, left local identity $p(x)$ (or, right local identity element $q(x)$) can be determined by using $p(x) = x/x = R_x^{-1}x$ (or, $q(x) = x\backslash x = L_x^{-1}x$).

An element $r(x)$ of a quasigroup $(Q, *)$ is called a middle local identity element of an element $x \in Q$, if $x*x = r(x)$. It can be determined using $r(x) = L_x x = R_x x$.

In a quasigroup there exist a unique left, right and middle local identity element of any fixed element $x$.

**Definition 1.1.10.** [116] An element $p \in Q$ (or, $q \in Q$) is called a left (or, right) identity element of a quasigroup $(Q, *)$ if and only if $p(x) = p$ (or, $q(x) = q$) for all $x \in Q$. Similarly, an element $r \in Q$ is called as middle identity element of a quasigroup $(Q, *)$ if and only if $r(x) = r$ for all $x \in Q$.

**Definition 1.1.11.** [116] A quasigroup $(Q, *)$ with a left, right and middle identity elements $p \in Q$, $q \in Q$ and $r \in Q$ are called a *left loop, a right loop* and a *unipotent quasigroup*, respectively. A quasigroup $(Q, *)$ in which left and right identity element coincide is referred as *loop*.

In a quasigroup, the left, right or middle identity elements are not necessarily present. However, according to the definition of a quasigroup, it's evident that each element within a quasigroup holds unique left, right and middle local identity elements if it exists.

**Definition 1.1.12.** [116] A quasigroup $(Q, *)$ is said to be an *idempotent quasigroup* if it satisfies the identity $a * a = a$ for all $a \in Q$. In such a quasigroup, any element $a$ that fulfills this condition is referred as an *idempotent element*.

**Remark 1.1.13.** In an idempotent quasigroup $(Q, *)$, the maps $p$, $q$ and $r$ are identity permutations maps on the set $Q$.

**Example 1.1.14.** Consider a set $Q = \{a, b, c\}$ and four different binary operations $*_1, *_2, *_3$ and $*_4$ as mentioned in Table 1.3.

| $*_1$ | a | b | c |
|---|---|---|---|
| a | b | c | a |
| b | a | b | c |
| c | c | a | b |

| $*_2$ | a | b | c |
|---|---|---|---|
| a | b | c | a |
| b | c | a | b |
| c | a | b | c |

| $*_3$ | a | b | c |
|---|---|---|---|
| a | a | b | c |
| b | b | c | a |
| c | c | a | b |

| $*_4$ | a | b | c |
|---|---|---|---|
| a | a | c | b |
| b | c | b | a |
| c | b | a | c |

Table 1.3: Latin squares with four different operations $*_1$, $*_2$, $*_3$ and $*_4$

It can be easily observed that the quasigroup $(Q, *_1)$ is a left loop but not a right loop, quasigroup $(Q, *_2)$ is a right loop but not a left loop, quasigroup $(Q, *_3)$ is a loop with identity $a$, $(Q, *_1)$ is a unipotent quasigroup and $(Q, *_4)$ is an idempotent quasigroup from the above Table 1.3.

A unipotent quasigroup which is also a left loop is called a *unipotent left loop* and a unipotent quasigroup which is also a right loop is called a *unipotent right loop*.

Table 1.4 shows the connections among local identity elements in all the six parastrophes of a quasigroup [116].

| | $\epsilon$ | $(1\,2)$ | $(1\,3)$ | $(2\,3)$ | $(1\,2\,3)$ | $(1\,3\,2)$ |
|---|---|---|---|---|---|---|
| $p$ | $p$ | $q$ | $r$ | $p$ | $q$ | $r$ |
| $q$ | $q$ | $p$ | $q$ | $r$ | $r$ | $p$ |
| $r$ | $r$ | $r$ | $p$ | $q$ | $p$ | $q$ |

Table 1.4: Connection among local identity elements in parastrophes of a quasigroup

**Remark 1.1.15.** If $(Q, \cdot)$ is a loop, then $(Q, \cdot^{(1\,2)})$ is a loop, $(Q, \cdot^{(1\,3)})$ is a unipotent right loop, $(Q, \cdot^{(2\,3)})$ is a unipotent left loop, $(Q, \cdot^{(1\,2\,3)})$ is a unipotent left loop, $(Q, \cdot^{(1\,3\,2)})$ is a unipotent right loop.

**Definition 1.1.16.** [116] Consider a system of equations involving $n$-ary groupoids $(Q, g_1)$, $(Q, g_2), \ldots, (Q, g_n)$ and for any fixed tuples $(b_1, \ldots, b_n) \in Q^n$, the system of equations:

$$\begin{cases} g_1(x_1, x_2, \ldots, x_n) & = b_1 \\ g_2(x_1, x_2, \ldots, x_n) & = b_2 \\ & \vdots \\ g_n(x_1, x_2, \ldots, x_n) & = b_n \end{cases} \tag{1.3}$$

has unique solution $(x_1, \ldots, x_n) \in Q^n$. Then, tuple $\langle g_1, g_2, \ldots, g_n \rangle$ is an orthogonal system of $n$-ary operations over $Q$.

### 1.1.1 Morphisms

A morphism is a structure-preserving map from one algebraic structure to another one of the same type. In set theory, morphisms are functions; in group theory, morphisms are group homomorphisms; and in ring theory, morphisms are the ring homomorphisms. In a similar way, we discuss some basic morphism maps in

quasigroup theory. For in-depth understanding of morphisms of different algebraic structures reader may refer (see [17, 96, 116]).

**Definition 1.1.17.** Let $Q = \{1, 2, \ldots, n\}$ be a finite set. A permutation of $Q$ is then defined as a bijection from $Q$ to itself.

**Definition 1.1.18.** [116] Two $n$-ary groupoids $(G, f)$ and $(G, g)$ are considered isotopic (or isotopic images) if there exist permutations $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $\alpha$ of the set $G$ such that for all $x_1, x_2, \ldots, x_n \in G$, the following equation holds:

$$f(x_1, x_2, \ldots, x_n) = \alpha^{-1} g(\alpha_1 x_1, \alpha_2 x_2, \ldots, \alpha_n x_n). \tag{1.4}$$

The equality (1.4) can also be expressed as $(G, f) = (G, g)T$, where $T = (\alpha_1, \alpha_2, \ldots, \alpha_n, \alpha)$ represents an isotopism or isotopy of $n$-ary groupoid. Notably, it's evident that an isotopic image of a quasigroup remains a quasigroup.

In cases where the $n$-ary operations $f$ and $g$ coincide in Definition 1.1.18, the tuple $T = (\alpha_1, \alpha_2, \ldots, \alpha_n, \alpha)$ is known as an *autotopism* or *autotopy* of an $n$-ary groupoid $(G, f)$. Denote the set of all autotopies of a groupoid $(G, f)$ as $Aut(G, f)$, and $Aut(G, g)$ forms a group with respect to standard component-wise multiplication of autotopies. The last component of an autotopy of a groupoid is represented by $\alpha$ and known as a *quasiautomorphism* of a groupoid.. Some special cases of isotopy like isomorphism and automorphism of groupoid $(Q, f)$ can be defined as following:

- If in Equation (1.4), all the permutations are same (i.e., $\alpha_1 = \alpha_2 = \cdots = \alpha_n = \alpha$), then the groupoid $(Q, f)$ and $(Q, g)$ are said to be isomorphic.
- Furthermore, when $\alpha_1 = \alpha_2 = \cdots = \alpha_n = \alpha$ and the $n$-ary operations $f$ and $g$ are same, this special case defines an *automorphism of groupoid* $(Q, f)$. In other words, a permutation $\alpha$ of the set $Q$ is called an automorphism of an $n$-ary groupoid $(Q, f)$ if the following condition holds for all elements $x_1, x_2, \ldots, x_n \in Q$:

$$\alpha f(x_1, \ldots, x_n) = f(\alpha x_1, \ldots, \alpha x_n).$$

For the case where $n = 2$, we obtain the definition of isotopism of binary quasigroups.

**Definition 1.1.19.** [116] Two binary groupoid $(Q, \circ)$ and $(Q, \cdot)$ are considered isotopic, if there exist permutations $\alpha_1, \alpha_2$ and $\alpha$ of the set $Q$ such that the following equation holds for all $x, y \in Q$:

$$\alpha(x \circ y) = \alpha_1 x \cdot \alpha_2 y.$$

**Remark 1.1.20.** An isotopism of the form $(\alpha_1, \alpha_2, \varepsilon)$ is called a *principal isotopism*. Up to isomorphism, any isotopism is a principal isotopism. Any isotopy of a quasigroup $(Q, \cdot)$ of the form $(R_a^{-1}, L_b^{-1}, \varepsilon)$ is called *loop isotopy (LP-isotopy)*, where $a, b \in Q$ and $R_a, L_b$ are right and left translation maps of $(Q, \cdot)$, respectively.

Let $T = (\alpha_1, \alpha_2, \alpha_3)$ be an isotopy of a groupoid $(Q, \cdot)$. Then, the action of $\sigma \in S_3$ on $T$ is denoted as $T^\sigma$ and defined as: $T^\sigma = (\alpha_{\sigma^{-1}1}, \alpha_{\sigma^{-1}2}, \alpha_{\sigma^{-1}3})$. If $(Q, f)$ is a quasigroup, then $(fT_1)^\sigma = f^\sigma T_1^\sigma$ and $(T_1T_2)^\sigma = T_1^\sigma T_2^\sigma$, where $T_1$ and $T_2$ are isotopisms of $(Q, f)$.

**Definition 1.1.21.** [116] A quasigroup $(Q, B)$ is an *isostrophic image* or *isostrophe* of $(Q, A)$, if there exists a collection of permutations $(\sigma, (\alpha_1, \alpha_2, \alpha_3)) = (\sigma, T)$, where $\sigma \in S_3$ and $T = (\alpha_1, \alpha_2, \alpha_3)$ is triplet of permutations $\alpha_1, \alpha_2, \alpha_3$ of the set $Q$ such that

$$B(x_1, x_2) = A(x_1, x_2)(\sigma, T) = \alpha_3^{-1} A(\alpha_1 x_{\sigma^{-1}1}, \alpha_2 x_{\sigma^{-1}2}) \tag{1.5}$$

for all $x_1, x_2 \in Q$.

The tuple $(\sigma, T)$ is known as *isostrophism* or *isostrophy* of the quasigroup $(Q, A)$. The equality of (1.5) can be rewritten as $B = (A^\sigma)T$. We shall call $\alpha_i$, the $i^{\text{th}}$ component of the isostrophism $(\sigma, T)$, for $i = 1, 2, 3$.

In other words, an isostrophic image of a quasigroup is defined as an isotopic image of its parastrophe.

**Definition 1.1.22.** [70] A quasigroup $(Q, *)$ satisfies the $(r, s, t)$-*inverse property* if there exist a permutation $J$ of the set $Q$ such that the following equation holds for all $x, y \in Q$:

$$J^r(x * y) * J^s x = J^t y. \tag{1.6}$$

If we replace $r = m$, $s = m + 1$, $t = m$, then we can obtain the definition of $m$-*inverse property* of the quasigroup $(Q, *)$.

**Definition 1.1.23.** [70] A quasigroup $(Q, *)$ satisfies the $(\alpha, \beta, \gamma)$-*inverse property* if there exist permutations $\alpha, \beta$ and $\gamma$ of the set $Q$ such that following equation holds for all $x, y \in Q$:

$$\alpha(x * y) * \beta x = \gamma y.$$

For an extensive survey and in-depth mathematical understanding on quasigroups and its morphisms, readers may refer [9, 10, 33, 116, 118].

### 1.1.2 String transformations based on quasigroup

In 1997, Markovski et al. [85] proposed several string transformations using quasigroup. It attracts significant attention in the field of cryptography. Consequently, various researchers have proposed additional string transformations using quasigroups [86–88].

Let $(Q, *)$ be a finite quasigroup. We denote a finite string over $Q$ as $Q^n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in Q, n \geqslant 2\}$. The functions $e_{l,*} : Q^n \to Q^n$ and $d_{l,*} : Q^n \to Q^n$ are defined as follows:

$$e_{l,*}(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n) \Leftrightarrow y_j = \begin{cases} l * x_1, & j = 1, \\ y_{j-1} * x_j, & 2 \leq j \leq n \end{cases} \tag{1.7}$$

and

$$d_{l,*}(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n) \Leftrightarrow y_j = \begin{cases} l * x_1, & j = 1, \\ x_{j-1} * x_j, & 2 \leq j \leq n. \end{cases} \tag{1.8}$$

Where $l \in Q$ is considered as an initial (or, leader) element. These functions $e_{l,*}$ and $d_{l,*}$ are called an *elementary quasigroup transformations*. The pictorial representations are shown in Figures 1.1 and 1.2.

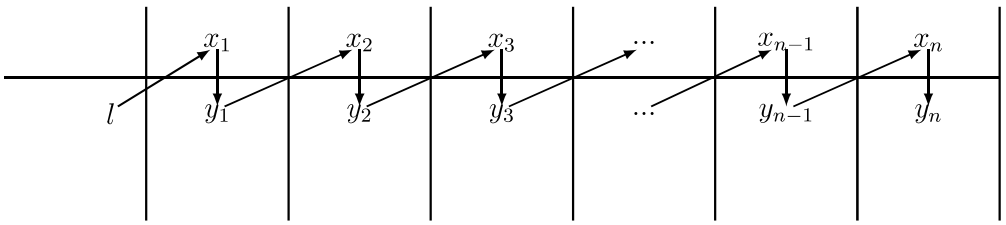


Figure 1.1: Pictorial representation of $e_{l,*}$ function

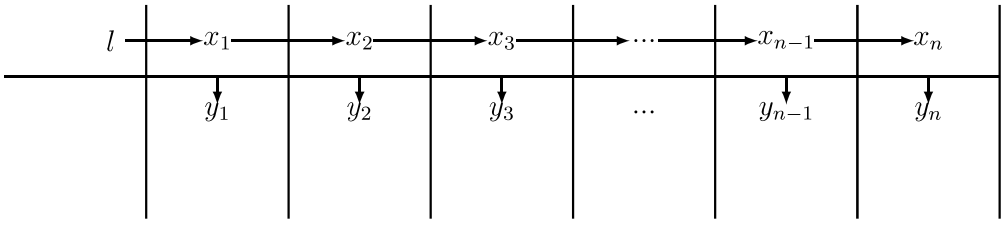


Figure 1.2: Pictorial representation of $d_{l,*}$ function

**Example 1.1.24.** Let $Q = \{1, 2, 3, 4, 5\}$ be a finite set. Suppose $*$ and $\backslash$ define two binary operations on $Q$. The quasigroup $(Q, *)$ and its parastrophe $(Q, \backslash)$ portrayed in Table 1.5.

Consider a string $x = (3, 3, 4, 5, 1, 2, 3, 4, 4, 5)$ and a leader element $l = 1$. Using the transformation map $e_{l,*}$ mentioned in Equation (1.7) on $x$ and Table 1.5, we get $e_{1,*}(x) = (3, 1, 4, 2, 2, 1, 3, 2, 5, 1)$. Subsequently, using transformation $d_{l,\backslash}$ mentioned in Equation (1.8) on the string $e_{1,*}(x)$ and Table 1.5, we get back the string $x$. It is straightforward to show that $d_{1,\backslash}(e_{1,*}(x)) = e_{1,*}(d_{1,\backslash}(x)) = x$.

| $*$ | 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 |

Table 1.5: A quasigroup $(Q, *)$ and its parastrophes $(Q, \backslash)$.

Given the algebra $(Q, *, \backslash, /)$ satisfying the identities in Equation (1.1), we can establish the following property:

**Theorem 1.1.25.** [116] *Let $Q$ a finite set equipped with three operations $*$, $\backslash$ and $/$. We define three quasigroups: $(Q, *)$, $(Q, \backslash)$ and $(Q, /)$. The operations $\backslash$ and $/$ are left and right parastrophes of operation $*$ respectively and can be calculated using Equation (1.1). Then, for all $x \in Q^n$ and a leader element $l \in Q$, the elementary transformations $e_{l,*}$ and $d_{l,\backslash}$ are mutually inverse permutations of $Q^n$* (i.e. $d_{l,\backslash}(e_{l,*}(x)) = x = e_{l,*}(d_{l,\backslash}(x))$).

Moreover, we delve into some important quasigroup transformations that have played pivotal roles in designing renowned cryptographic primitives. In 2004, Gligoroski [55] presented a string transformation, where the leaders are the elements of the input string arranged in reverse order, known as the reverse string transformation. This transformation has found application in the construction of the Edon-$\mathcal{R}$ family of cryptographic hash functions [60]. The quasigroup reverse string transformation defined as follows.

**Definition 1.1.26.** [60] Consider a quasigroup $(Q, *)$. Let $n \in \mathbb{N}$ and $x_j \in Q$ for all $j = \{1, \ldots, n\}$. The quasigroup reverse string transformation, denoted by $\mathcal{R} : Q^n \to Q^n$, can be defined as the composition of elementary $e$-transformations:

$$\mathcal{R}(x_1, x_2, \cdots, x_n) = (e_{x_1} \circ e_{x_2} \circ \cdots \circ e_{x_n})(x_1, x_2, \ldots, x_n). \tag{1.9}$$

In 2008, Markovski and Mileva [90] introduced several quasigroup string transformations, including the quasigroup additive string transformation and quasigroup reverse additive string transformation. These transformations have played

a key role in the design of hash families like NaSHA [90]. We now discuss the definitions for the quasigroup additive and reverse additive transformations.

Consider a quasigroup $(\mathbb{Z}_{2^n}, *)$. Let $t \in \mathbb{N}$, $1 \leq j \leq n$ and $+$ represents the addition modulo $2^n$.

**Definition 1.1.27.** [90] The quasigroup additive string transformation is a map $\mathcal{A}_l : \mathbb{Z}_{2^n}^t \to \mathbb{Z}_{2^n}^t$ with leader element $l$ and defined as follows:

$$\mathcal{A}_l(x_1, x_2, \ldots, x_t) = (z_1, z_2, \ldots, z_t) \Leftrightarrow z_j = \begin{cases} (l + x_1) * x_1; \ j = 1 \\ (z_{j-1} + x_j) * x_j; \ 2 \leq j \leq t \end{cases} \tag{1.10}$$

where $x_j, z_j \in \mathbb{Z}_{2^n}$.

**Definition 1.1.28.** [90] The quasigroup reverse additive string transformation is a map $\mathcal{RA}_l : \mathbb{Z}_{2^n}^t \to \mathbb{Z}_{2^n}^t$ with leader element $l$ and defined as follows:

$$\mathcal{RA}_l(x_1, x_2, \ldots, x_t) = (z_1, z_2, \ldots, z_t) \Leftrightarrow z_j = \begin{cases} x_j * (x_j + z_{j+1}); \ 1 \leq j \leq t-1 \\ x_t * (x_t + l); \ j = t \end{cases} \tag{1.11}$$

where $x_i, z_i \in \mathbb{Z}_{2^n}$.

Consider transformations $m_{l_i} \in \mathcal{A}_{l_i}$ or $\mathcal{RA}_{l_i}$, for fixed $l_i$, $1 \leq i \leq s$. Define a string transformation $M$ as follows:

$$M = m_{l_1} \circ m_{l_2} \circ \cdots \circ m_{l_s}.$$

Let $\rho(z, \lfloor \frac{n}{2} \rfloor)$ denotes the element of $\mathbb{Z}_{2^n}$ obtained by rotating the $n$-bit representation of $z$ to the left by $\lfloor \frac{n}{2} \rfloor$ bits. For a given string $Z = (z_1, z_2, \ldots, z_t) \in \mathbb{Z}_{2^n}^t$, $\rho(Z)$ denotes the string given by:

$$\rho(Z) = (\rho(z_1, \lfloor \frac{n}{2} \rfloor), \ldots, \rho(z_t, \lfloor \frac{n}{2} \rfloor)) \in \mathbb{Z}_{2^n}^t.$$

The function $\rho(f) = \rho(f)(Z)$ is defined for $f = f(Z)$ as $\rho(f)(Z) = f(\rho(Z))$.

**Definition 1.1.29.** [90] Let $t \in \mathbb{N}$ and $k$ is an even natural number. The quasigroup main transformation with complexity $k$ is a map $\mathcal{MT} : \mathbb{Z}_{2^n}^t \to \mathbb{Z}_{2^n}^t$ and defined as:

$$\mathcal{MT}(x_1, x_2, \ldots, x_t) = \rho(\mathcal{RA}_{l_1}) \circ \mathcal{A}_{l_2} \circ \ldots \rho(\mathcal{RA}_{l_{k-1}}) \circ \mathcal{A}_{l_k}(x_1, \ldots, x_t)$$

for all $x_i \in \mathbb{Z}_{2^n}$.

### 1.1.3 Quasigroup as a vector valued Boolean functions

A finite quasigroup that can be represented as a vector valued Boolean functions, offer a means to determine its algebraic degree. The categorization of quasigroups relies on the given algebraic degree. Several post-quantum public key cryptosystems have been designed using multivariate quadratic quasigroups [58, 61, 62], which can be formulated utilizing the vector valued representation of quasigroups.

A Boolean functions is a map of the form $\mathbb{F}_2^k \to \mathbb{F}_2$. A *vector valued Boolean function* is a map $f : \mathbb{F}_2^k \to \mathbb{F}_2^t$ and defined as follows:

$$f(x_1, \ldots, x_k) = (f_1(x_1, \ldots, x_k), f_2(x_1, \ldots, x_k), \ldots, f_t(x_1, \ldots, x_k))$$

where $f_i$ is a Boolean function and also referred as the *coordinate functions* of $f$. The Boolean functions obtained by the non-zero $\mathbb{F}_2$ linear combinations of $f_i$ are called *component functions of* $f$.

Every Boolean function in $k$ variables, denoted as $f_i(x_1, x_2, \ldots, x_k)$, can be uniquely represented in its *Algebraic Normal Form (ANF)*. The ANF of the function $f_i$ can be represented as:

$$f_i(x_1, x_2, \ldots, x_k) = \sum_{I \subseteq \{1,2,\ldots,k\}} \alpha_I \left( \prod_{i \in I} x_i \right),$$

where the coefficients $\alpha_I \in \mathbb{F}_2$ and the operations addition and multiplication are over $\mathbb{F}_2$. The ANF representation allows us to calculate the algebraic degree of a vector valued Boolean function $f$. Mathematically, degree of $f$ can be expressed as:

$$\deg(f) = \max\{\deg(f_i) \mid i \in \{1, \ldots, t\}\}. \tag{1.12}$$

This value is also equal to the maximum degree of any component functions of $f$.

Gligoroski et al. [56] presented that every quasigroup $(Q, *)$ with an order of $2^d$, where $d \geq 2$ can be expressed as a vector-valued Boolean function. In simpler terms, the quasigroup operation $*$ can be represented as $f : \mathbb{F}_2^{2d} \to \mathbb{F}_2^d$. Furthermore, for any elements $x, y, z \in Q$, with their binary representations $(x_1, x_2, \ldots, x_d)$, $(y_1, y_2, \ldots, y_d)$ and $(z_1, z_2, \ldots, z_d)$, respectively. The quasigroup operation $x * y = z$ can be represented as:

$$\begin{aligned} &f(x_1, x_2, \ldots, x_d, y_1, y_2, \ldots, y_d) = (z_1, z_2, \ldots, z_d) \\ &= (f_1(x_1, \ldots, x_d, y_1, \ldots, y_d), \ldots, f_d(x_1, \ldots, x_d, y_1, \ldots, y_d)), \end{aligned}$$

where each $z_i$ can be uniquely expressed as $z_i = f_i(x_1, \ldots, x_d, y_1, \ldots, y_d)$ for $i = 1, \ldots, d$ and $f_i : \mathbb{F}_2^{2d} \to \mathbb{F}_2$. If the degree of $f_i$ (denoted by $\deg(f_i)$) is equal to 1,

it implies that $f_i$ is a linear function.

**Definition 1.1.30.** [17] Let $(G, +)$ be a group. A function $f : G \to G$ is an *affine function*, if for all $x, y \in G$: $f(x + y) = f(x) + f(y) - f(0)$, where $0 \in G$ denotes the identity element of the group $G$. An affine function $f$ is further classified as a linear function if it satisfies the additional condition $f(0) = 0$.

Quasigroups can be categorized based on their algebraic degree, which reflects the complexity of their underlying Boolean functions. This classification yield two main categories: linear and non-linear quasigroups [37, 56].

- **Linear quasigroups:** These quasigroups Boolean representation have a degree equal to 1, meaning all their component Boolean functions are linear.
- **Non-Linear quasigroups:** These quasigroups Boolean representation have a degree greater than 1, meaning all their component Boolean functions are non-linear. Non-Linear quasigroups can be further subdivided:
  - **Weak Non-Linear quasigroups:** These quasigroups Boolean representation contains a linear (degree 1) and non-linear (degree greater than 1) component Boolean functions.
  - **Pure Non-Linear quasigroups:** These quasigroups Boolean representation consists entirely of non-linear component Boolean functions.

Gligoroski et al. [58] gave the following definitions of Multivariate Quadratic Quasigroups (MQQ) that have been used in designing public key cryptosystems [58, 61, 62].

**Definition 1.1.31.** [58] Consider a quasigroup $(Q, *)$ of order $2^d$. It is referred as *Multivariate Quadratic Quasigroup (MQQ)* of type $Quad_{d-k}Lin_k$, if its vector valued Boolean representation contains exactly $d - k$ quadratic polynomials and $k$ linear polynomials, where $0 \leq k < d$.

The sufficient condition for a quasigroup $(Q, *)$ to be a MQQ is given by the following theorem:

**Theorem 1.1.32.** [58] *Consider two $d \times d$ sized matrices $P$ and $Q$ of linear Boolean expressions, and two vectors $b_1$ and $b_2$ of linear or quadratic Boolean expressions. Let elements of $P, b_1$ depends on the variables $X = (x_1, \ldots, x_d)$ and the elements of $Q, b_2$ depends upon the variables $Y = (y_1, y_2, \ldots, y_d)$. If $\det(P) = \det(Q) = 1$ and $P \cdot Y^T + b_1 \equiv Q \cdot X^T + b_2$ in $\mathbb{F}_2$. Then the vector-valued operation $*_{vv}$ of the given quasigroup $(Q, *)$ with order $2^d$ can be expressed as:*

$$*_{vv}(x_1, \cdots, x_d, y_1, \cdots, y_d) = P \cdot Y^T + b_1$$

*The quasigroup $Q$ with vector valued operation $*_{vv}$ is known as Multivariate Quadratic Quasigroup (MQQ).*

**Example 1.1.33.** [58] Consider a quasigroup $(Q, *)$ of order 8. The quasigroup $(Q, *)$ is represented by following table:

| $*$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 6 | 7 | 1 | 0 | 4 | 5 |
| 1 | 5 | 3 | 7 | 1 | 0 | 6 | 2 | 4 |
| 2 | 0 | 6 | 3 | 5 | 4 | 2 | 7 | 1 |
| 3 | 6 | 7 | 2 | 3 | 5 | 4 | 1 | 0 |
| 4 | 7 | 1 | 4 | 2 | 3 | 5 | 0 | 6 |
| 5 | 1 | 0 | 5 | 4 | 2 | 3 | 6 | 7 |
| 6 | 4 | 5 | 1 | 0 | 6 | 7 | 3 | 2 |
| 7 | 2 | 4 | 0 | 6 | 7 | 1 | 5 | 3 |

Table 1.6: A quasigroup $(Q, *)$ of order 8

The Algebraic normal form of the operation $*$ as a vector valued Boolean function representation is: $*_{vv}(x_1, x_2, x_3, x_4, x_5, x_6) = (f_1, f_2, f_3)$, where

$$f_1 = x_1 + x_3 + x_1x_4 + x_2x_4 + x_3x_4 + x_5 + x_1x_5 + x_2x_5 + x_3x_5 + x_1x_6 + x_2x_6 + x_3x_6,$$
$$f_2 = 1 + x_2 + x_3 + x_4 + x_1x_4 + x_2x_4 + x_3x_4 + x_1 + x_5 + x_2x_5 + x_3x_5 + x_1x_6 + x_2x_6 + x_3x_6,$$
$$f_3 = 1 + x_2 + x_3x_4 + x_5 + x_3x_5 + x_6 + x_1x_6 + x_2x_6 + x_3x_6.$$

The corresponding matrix vector representations of $*$ by $P$, $b_1$, $Q$ and $b_2$ are: $*_{vv}(x_1, x_2, x_3, x_4, x_5, x_6) = P \cdot (x_4, x_5, x_6)^T + b_1$, where

$$P = \begin{bmatrix} x_1 + x_2 + x_3 & 1 + x_1 + x_2 + x_3 & x_1 + x_2 + x_3 \\ 1 + x_1 + x_2 + x_3 & x_1 + x_2 + x_3 & x_1 + x_2 + x_3 \\ x_3 & 1 + x_3 & 1 + x_1 + x_2 + x_3 \end{bmatrix} \text{ and } b_1 = \begin{bmatrix} x_1 + x_3 \\ 1 + x_2 + x_3 \\ 1 + x_2 \end{bmatrix}$$

or, $*_{vv}(x_1, x_2, x_3, x_4, x_5, x_6) = Q \cdot (x_1, x_2, x_3)^T + b_2$, where

$$Q = \begin{bmatrix} 1 + x_4 + x_5 + x_6 & x_4 + x_5 + x_6 & 1 + x_4 + x_5 + x_6 \\ x_4 + x_5 + x_6 & 1 + x_4 + x_5 + x_6 & 1 + x_4 + x_5 + x_6 \\ x_6 & 1 + x_6 & 1 + x_4 + x_5 + x_6 \end{bmatrix} \text{ and } b_2 = \begin{bmatrix} x_1 + x_3 \\ 1 + x_2 + x_3 \\ 1 + x_2 \end{bmatrix}.$$

Here, it can be observed that $\det(P) = \det(Q) = 1$ in $\mathbb{F}_2$.

**Theorem 1.1.34.** [94] *Polynomial $p(x)$ is said to be permutation polynomial of degree 2 if $p(x) = ax^2 + b$, where $a, b \in \mathbb{F}(p^k)$, $a \neq 0$ and prime $p = 2$.*

## 1.2 Basics of coding theory

In practical scenarios where data storage and communication systems are not completely reliable due to noise or other form of interference. Hence, one of the primary objectives in coding theory is to devise techniques for error detection and correction. Typically, coding is employed in two main domains: *source coding* and *channel coding*. Source coding entails converting the message effectively into a suitable codeword for transmission through the channel. For instance, the ASCII code exemplifies source coding by converting each character into a byte consisting 8 bits. On the other hand, channel coding involves encoding the message again after source coding, introducing redundancy to enable the detection or correction of errors.

**Definition 1.2.1.** [81] Let $\mathbf{A} = \{a_1, a_2, \ldots, a_q\}$ be a set of size $q$, refer as a *code alphabet*. Its elements, $a_i$, are called *code symbols*.

(*i*) A $q$-ary word of length $n$ over $\mathbf{A}$ is a sequence of symbols $w = w_1 w_2 \ldots w_n$ where each $w_i \in \mathbf{A}$ for all $i$. We can also represent this word $w$ as a vector $(w_1, \ldots, w_n) \in \mathbf{A}^n$.

(*ii*) A $q$-ary block code of length $n$ over $\mathbf{A}$ is a non-empty set $C$ of $q$-ary words, all having the same length $n$. These words in the collection $C$ are called codewords.

(*iii*) The number of codewords in $C$, denoted by $|C|$, is the size of code $C$. The (information) rate of a code $C$ of length $n$ is $(\log_q |C|)/n$.

(*iv*) An $(n, M)$-code represents a code of length $n$ and cardinality $M$.

**Example 1.2.2.** Consider an alphabet set $\mathbb{F}_2 = \{0, 1\}$. The code defined over $\mathbb{F}_2$ is referred as *binary code*. For better understanding of binary codes we give the following examples:

(*i*) $C_1 = \{00, 01, 10, 11\}$ is a $(2, 4)$-code;

(*ii*) $C_2 = \{000, 011, 101, 110\}$ is a $(3, 4)$-code;

(*iii*) $C_3 = \{0011, 0101, 1010, 1100, 1001, 0110\}$ is a $(4, 6)$-code.

Similarly, a code over the alphabet set $\mathbb{F}_3 = \{0, 1, 2\}$ is referred as *ternary code*.

**Definition 1.2.3.** [83] Let $\mathbf{A}$ be an alphabet set. We consider words $x$ and $y$ of length $n$ over $\mathbf{A}$. The *Hamming distance* between $x$ and $y$, denoted by $d_H(x, y)$, is defined as the number of positions where the corresponding symbols in $x$ and $y$ differ.

Suppose the representation of $x$ and $y$ are $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ respectively, then the Hamming distance ($d_H(x, y)$) can be defined as:

$$d_H(x, y) = d_H(x_1, y_1) + \cdots + d_H(x_n, y_n) \tag{1.13}$$

where $x_i$ and $y_i$ can be referred as separate words with length 1, and the distance between them can be defined as

$$d_H(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i \end{cases}$$

**Definition 1.2.4.** [83] The minimum distance $d(C)$ for a code $C$ containing at least two words is defined as:

$$d(C) = \min\{d_H(x, y) : x, y \in C, x \neq y\}.$$

A code $C$ characterized by its length $n$, cardinality $M$ and distance $d$, is often referred using its parameters. Such a code $C$ can be denoted as $(n, M, d)$-code.

**Definition 1.2.5.** [83] The (Hamming) weight of a word $x \in \mathbb{F}_q^n$, denoted by $wt(x)$, is the number of non-zero coordinates in $x$ (i.e., $wt(x) = d_H(x, \mathbf{0})$, where $\mathbf{0}$ is the zero word).

The distance $d(C)$ of a code $C$ is closely tied to its error detecting and error correcting capabilities [83, 110].

**Definition 1.2.6.** [81] Let $u$ be a positive integer. A code $C$ is said to be $u$-error-detecting, if whenever a codeword incurs at least one but atmost $u$ errors, the resulting word is not considered a valid codeword. There is a codeword for which some $u + 1$ changes gives another valid codeword.

**Theorem 1.2.7.** [81] *A code $C$ is $u$-error-detecting if and only if its minimum distance $d(C)$, satisfies $d(C) \geq u + 1$. In other words, a code with a minimum distance $d$ is an exactly $(d - 1)$-error-detecting code.*

Suppose codewords from a code $C$ are transmitted over a communication channel, errors may occur during the transmission. To recover the original message, a decoding technique is employed. The *nearest neighbour decoding rule* (also known as the *minimum distance decoding rule*) aims to correct these errors.

Upon receiving a potentially corrupted word $x$, the decoder compares it (using, Hamming distance, $d_H$) to all the codewords $c$ within the code $C$. The decoder chooses the codeword $c_x$ that is closest to the received word $x$ in terms of Hamming distance i.e., $d(x, c_x) = \min_{c \in C} d_H(x, c)$.

**Definition 1.2.8.** [81] Let $t$ be a positive integer. A code $C$ is defined as $t$-error-correcting if minimum distance decoding rule can correct $t$ or fewer errors, under the assumption that this decoding rule are applied. Moreover, a code $C$ is precisely $t$-error-correcting if it meets the condition of being $t$-error-correcting but fails to be $(t+1)$-error-correcting.

**Theorem 1.2.9.** [81] *A code $C$ is $t$-error-correcting if and only if $d(C) \geq 2t+1$; i.e., a code with distance $d$ is an exactly $\lfloor (d-1)/2 \rfloor$-error-correcting code. Here, $\lfloor x \rfloor$ is the greatest integer less than or equal to $x$.*

For given $q$, $n$ and $d$, several well-known upper and lower bounds exist [83, 110] for determining the largest possible value of $M$. When $M$ attains one of these well-known bounds, it often leads to the discovery of interesting codes including perfect codes [83] and maximum distance separable (MDS) codes [83].

**Definition 1.2.10.** [83] Consider an alphabet set $\mathbf{A}$ containing $q$ distinct symbols (where $q > 1$). Let $A_q(n,d)$ denote the largest possible value of $M$, for which there exists an $(n, M, d)$-code over $\mathbf{A}$. Mathematically, it can be expressed as:

$$A_q(n,d) = \max\{M : \text{ there exists an } (n,M,d)\text{-code over } \mathbf{A}\}$$

Any $(n, M, d)$-code $C$ that achieves the maximum size, meaning it has $M = A_q(n,d)$ codewords, is called an *optimal code*.

Indeed, it is important to note that $A_q(n,d)$ relies solely on the values of $n$, $d$ and $M$.

**Definition 1.2.11.** [83] Consider a finite field $\mathbb{F}_q$ with $q$ elements. A code $C$ of length $n$ is said to be linear code over $\mathbb{F}_q$, if it is a subspace of $\mathbb{F}_q^n$.

A linear code $C$ of length $n$ and dimension $k$ over the finite field $\mathbb{F}_q$ is commonly referred to as a $q$-ary $[n,k]$-code. Alternatively, it can be denoted as an $(n, q^k)$-linear code. When the distance $d$ of $C$ is specified, it is denoted as an $[n,k,d]$-linear code over $\mathbb{F}_q$.

**Definition 1.2.12.** [83] Let $C$ be a linear code in $\mathbb{F}_q^n$. The *dual code* of $C$, denoted by $C^\perp$, is defined as the orthogonal complement of the subspace $C$ of $\mathbb{F}_q^n$. (i.e., $C^\perp = \{c' \in \mathbb{F}_q^n : \langle c', c\rangle = 0 \text{ for all } c \in C\}$, where $\langle \cdot, \cdot \rangle$ is an inner product on $\mathbb{F}_q^n$).

**Definition 1.2.13.** [83] A *generator matrix* for a linear code $C$ is a matrix $G$ whose rows constitutes a basis for $C$. On the other hand, a *parity-check matrix* $H$ for a linear code $C$ serves as a generator matrix for the dual code $C^\perp$.

**Definition 1.2.14.** [83] For a given prime power $q$ and fixed values of $n$ and $d$, let $B_q(n,d)$ represents the largest possible size $q^k$ for which there exists an $[n,k,d]$-code over $\mathbb{F}_q$. Hence, it can be expressed as:

$$B_q(n,d) = \max\{q^k : \text{ there exist an } [n,k,d]\text{-code over } \mathbb{F}_q\}$$

Let's explore the upper bound for $A_q(n,d)$ resulting from the Singleton bound [81].

**Theorem 1.2.15.** [81] (Singleton bound) *For any integers $n, d, q > 1$ and $1 \leq d \leq n$, we have*

$$A_q(n,d) \leq q^{n-d+1}.$$

*In particular, when $q$ is a prime power, the parameters $[n,k,d]$ of any linear code over $\mathbb{F}_q$ satisfy the following condition:*

$$k + d \leq n + 1.$$

**Definition 1.2.16.** [81] A linear code with parameters $[n,k,d]$ is called a *maximum distance separable* (MDS) code if $k + d = n + 1$.

The following theorem presents some intriguing properties of MDS codes:

**Theorem 1.2.17.** [81] *Let $C$ be a linear code over the finite field $\mathbb{F}_q$ with parameters $[n,k,d]$. Suppose $G$ and $H$ be a generator matrix and a parity-check matrix for $C$ respectively. The following statements are equivalent:*

(*i*) *$C$ is an MDS code;*

(*ii*) *every set of $n-k$ columns of $H$ is linearly independent;*

(*iii*) *every set of $k$ columns of $G$ is linearly independent;*

(*iv*) *$C^\perp$ is an MDS code.*

**Definition 1.2.18.** [81] A subset $S$ of $\mathbb{F}_q^n$ is called a cyclic set if a vector $(a_{n-1}, a_0, \ldots, a_1, a_{n-2}) \in S$ whenever $(a_0, a_1, \ldots, a_{n-1}) \in S$. A linear code $C$ is called a *cyclic code* if $C$ is a cyclic set.

The word $(u_{n-r}, \ldots, u_{n-1}, u_0, u_1, \ldots, u_{n-r-1})$ is said to be obtained from the word $(u_0, \ldots, u_{n-1}) \in \mathbb{F}_q^n$ by cyclically shifting $r$ times.

### 1.2.1 Some special types of error-correcting codes

In this section, we explore certain special types of cyclic codes that can correct errors resulting from noisy channels.

### Bose-Chaudhuri-Haucquenghem codes (BCH codes)

BCH codes were introduced by Alexis Hocquenghem in 1959, and independently by Raj Chandra Bose and D. K. Ray Chaudhuri in 1960. It is possible to design binary BCH codes that can correct multiple bit errors depending upon requirements of the systems. These codes have efficient method for decoding known as syndrome decoding [83]. Special classes of BCH codes are used in numerous applications including satellite communications, CD players, USB flash drives, data storage, mobile communications, QR codes etc.

**Definition 1.2.19.** [81] Let $\alpha$ be a primitive element of $\mathbb{F}_{q^m}$, and let $M^{(i)}(x)$ denote the minimal polynomial of $\alpha^i$ over $\mathbb{F}_q$. A (primitive) BCH code over $\mathbb{F}_q$ of length $n = q^m - 1$ with designed distance $\delta$ is a cyclic code generated by $g(x) := \text{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$ for some integer $a$. Furthermore, the code is referred to as narrow-sense if $a = 1$.

The Reed-Solomon codes are the most important subclass within the domain of BCH codes.

### Reed-Solomon code (RS codes)

Consider a $q$-ary BCH code $C$ of length $q^m - 1$ generated by $g(x) := \text{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$, where $M^{(i)}(x)$ represents the minimal polynomial of $\alpha_i$ with respect to $\mathbb{F}_q$ for a primitive element $\alpha$ of $\mathbb{F}_{q^m}$. When $m = 1$, we obtain a $q$-ary BCH code of length $q - 1$. This code is known as *RS code* (cyclic). In this scenario, $\alpha$ serves as a primitive element of $\mathbb{F}_q$ and the minimal polynomial of $\alpha^i$ with respect to $\mathbb{F}_q$ is $x - \alpha^i$. Thus, for $\delta \leq q - 1$, the generator polynomial becomes

$$\begin{aligned} g(x) &= \text{lcm}(x - \alpha^a, x - \alpha^{a+1}, \ldots, x - \alpha^{a+\delta-2}) \\ &= (x - \alpha^a)(x - \alpha^{a+1}) \ldots (x - \alpha^{a+\delta-2}) \end{aligned}$$

since $\alpha^a, \alpha^{a+1}, \ldots, \alpha^{a+\delta-2}$ are pairwise distinct.

**Definition 1.2.20.** [81] A $q$-ary *Reed-Solomon code* (RS code) is a $q$-ary BCH code of length $q - 1$ generated by

$$g(x) = (x - \alpha^a)(x - \alpha^{a+1}) \ldots (x - \alpha^{a+\delta-2})$$

with $a \geq 0$ and $2 \leq \delta \leq q - 1$, where $\alpha$ is a primitive element of $\mathbb{F}_q$.

**Theorem 1.2.21.** [81] *Reed-Solomon codes are maximum distance separable (MDS)*

*code; i.e., a q-ary Reed-Solomon code of length* $q-1$ *generated by* $g(x) = \prod_{i=a+1}^{a+\delta-1}(x-\alpha^i)$ *is a* $[q-1, q-\delta, \delta]$*-cyclic code for any* $2 \leq \delta \leq q-1$.

## 1.3 Basics of cryptography

The term cryptology has two main branches one is cryptography and other is cryptanalysis. Cryptography is the study of scientific methods used to secure user communications via insecure channel. Cryptography offers security for communications in terms of confidentiality, integrity, authentication and non-repudiation. Cryptanalysis refers to the study of methods used to decipher partially or fully encrypted information (or, message) without access to the necessary key. In essence, it involves the exploration of technique to break encryption algorithms or their implementations. This section mainly focus on basics of cryptography and multivariate polynomials based public key cryptography. For detailed theory about the cryptography, readers may refer [38, 63, 69, 75, 119, 122, 131]

**Definition 1.3.1.** [119] A cryptosystem is formally defined as a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$. Here, $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{K}$ denote the possible plaintext space, ciphertext space and keyspace respectively. For each $k \in \mathcal{K}$, there exists an encryption scheme $e_k \in \mathcal{E}$ and a corresponding decryption scheme $d_k \in \mathcal{D}$. Each $e_k : \mathcal{P} \to \mathcal{C}$ and $d_k : \mathcal{C} \to \mathcal{P}$ are functions such that $d_k(e_k(x)) = x$ for every plaintext element $x \in \mathcal{P}$.

Cryptography is broadly categorised into two main categories: symmetric (private) key cryptography and asymmetric (public) key cryptography.

**Definition 1.3.2.** [63] A symmetric encryption scheme $SE = \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ comprises of three algorithms: key generation algorithm ($\mathcal{K}$), encryption algorithm ($\mathcal{E}$) and decryption algorithm ($\mathcal{D}$).

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

(*ii*) *Encryption algorithm*: The encryption algorithm $\mathcal{E}_K : \mathcal{P} \to \mathcal{C}$ can be either randomized or stateful. When instantiated with the key $K$, it takes a plaintext $m \in \mathcal{P}$ as an input and returns a ciphertext $c$, denoted as $c \longleftarrow \mathcal{E}_K(m)$.

(*iii*) *Decryption algorithm*: The decryption algorithm $\mathcal{D}_K : \mathcal{C} \to \mathcal{P}$ is both deterministic and stateless. It initializes with a key $K$ and processes a string $c$ to return the corresponding plaintext $m \in \mathcal{P}$, denoted as $m \leftarrow D_K(c)$. It is important to ensure that the algorithms $\mathcal{E}$ and $\mathcal{D}$ guarantee that $\mathcal{D}_K(\mathcal{E}_K(m)) = m$ holds for all $m \in \mathcal{P}$.

In an asymmetric key cryptosystem, two keys are used: a public key and a private key, instead of same private key for both encryption and decryption processes. The public key is employed for the encryption process, while the private key is utilized for decryption. Formally, the asymmetric encryption scheme can be defined as follows:

**Definition 1.3.3.** [69] A asymmetric encryption scheme $ASE = \{\mathcal{K}, \mathcal{E}, \mathcal{D}\}$ comprises of three algorithms: key generation algorithm ($\mathcal{K}$), encryption algorithm ($\mathcal{E}$) and decryption algorithm ($\mathcal{D}$).

(*i*) *Key generation*: In key generation algorithm $\mathcal{K}$, it accepts a security parameter $\mathtt{k} \in \mathbb{N}$ as input and returns a tuple $K = (sk, pk)$ as output, denoted as $(sk, pk) \xleftarrow{R} \mathcal{K}(\mathtt{k})$.

(*ii*) *Encryption algorithm*: The encryption algorithm $\mathcal{E}_{pk} : \mathcal{P} \to \mathcal{C}$ can be either randomized or stateful. When instantiated with the public key $pk$, it takes a plaintext $m \in \mathcal{P}$ as input and returns a ciphertext $c$, denoted as $c \longleftarrow \mathcal{E}_{pk}(m)$.

(*iii*) *Decryption algorithm*: The decryption algorithm $\mathcal{D}_{sk} : \mathcal{C} \to \mathcal{P}$ is both deterministic and stateless. It initializes with a secret key $sk$ and processes a string $c$ to return the corresponding plaintext $m \in \mathcal{P}$, denoted as $m \leftarrow D_{sk}(c)$. It is important to ensure that the algorithm $\mathcal{E}$ and $\mathcal{D}$ guarantee that $\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)) = m$ holds for all $m \in \mathcal{P}$.

A digital signature scheme is a mathematical scheme which helps is maintaining the integrity and authenticity of the digital messages and documents. It can be defined as follows:

**Definition 1.3.4.** [69] A *Digital Signature scheme* $DSig = \{\mathcal{K}, Sign, Verify\}$ typically consists three algorithms:

(*i*) *Key generation* ($\mathcal{K}$): In key generation algorithm ($\mathcal{K}$), it accepts a security parameter $\mathtt{k} \in \mathbb{N}$ as input and returns a tuple $K = (sk, pk)$ as output, denoted as $(sk, pk) \xleftarrow{R} \mathcal{K}(\mathtt{k})$.

(*ii*) *Signing algorithm* (*Sign*): It takes $m \in \mathcal{P}$ and $sk$ as an input and returns a signature $\sigma$, denoted as $\sigma \longleftarrow Sign_{sk}(m)$

(*iii*) *Signature verification algorithm* (*Ver*): It takes $m \in \mathcal{P}$, $\sigma$ and $pk$ as input and returns either True or False, denoted as $\{True,\ False\} \longleftarrow Ver_{pk}(\sigma, m)$

The security of classical public key cryptosystems such as Rivest-Shamir-Adlemann (RSA) scheme relies on the difficulty of factoring large integers. On the other hand, the security of cryptographic schemes like ElGamal and Elliptic curves based cryptosystems relies on the hardness of solving the discrete logarithm problem. The existence of large scale quantum computers would pose a significant threat to cryptographic systems whose security relies on factoring large numbers and discrete logarithm problem using Shor's polynomial time algorithm for prime factorization [117]. As a result, it is imperative for cryptographers to propose a new cryptographic system or redesign the existing ones to evolve and adapt to the potential challenges posed by quantum computers. This idea led to a new research area called *Post Quantum Cryptography (PQC)*. The post-quantum cryptography [15] mainly divided into six categories: (*i*) *Lattice based cryptography*; (*ii*) *Multivariate polynomial based public key cryptography*; (*iii*) *Hash based cryptography*; (*iv*) *Code based cryptography*; (*v*) *Isogeny based cryptography*; (*vi*) *Symmetric key quantum resistance*.

In this thesis, we propose a quantum secure digital signature scheme using multivariate quadratic quasigroup (MQQ). The security of the proposed signature scheme relies on Multivariate Quadratic (MQ) problem. For in-depth mathematical understanding on the MQ problem and public key cryptographic schemes based on MQ problem, readers may refer [38, 127].

### 1.3.1 Multivariate equations and multivariate polynomials based digital signature scheme

Suppose $x_1, x_2, \ldots, x_n \in \mathbb{F}_q$ denotes $n$ number of variables. First we set $x_0 := 1$ which is the multiplicative neutral element of $\mathbb{F}_q$. Furthermore, for given $n, d \in \mathbb{N}$ we define

$$\mathcal{V}_n^d := \begin{cases} \{0\} & , \text{for } d = 0 \\ \{v \in \{0, \ldots, n\}^d : 1 \leq j \Rightarrow v_i \leq v_j\} & , \text{otherwise.} \end{cases}$$

It means, vector $v$ is represented as $v = (v_1, \ldots, v_d)$, where $v_i \in \{0, \ldots, n\}$. Suppose $\mathcal{F}$ consists a system of $m$ polynomials in $n$ variables with maximum degree $d \in \mathbb{N}$, i.e., $\mathcal{F} := (f_1, \ldots, f_m)$, where each $f_i$ can be represented as:

$$f_i(x_1, \ldots, x_n) := \sum_{v \in \mathcal{V}_n^d} \gamma_{i,v} \prod_{j=1}^{d} x_{v_j} \text{ for } 1 \leq i \leq m, \tag{1.14}$$

where the coefficient $\gamma_{i,v} \in \mathbb{F}_q$ and vectors $v \in \mathcal{V}_n^d$.

Suppose $y_1, \ldots, y_m$ be fixed elements from $\mathbb{F}_q$ and multivariate polynomials $f_1, f_2, \ldots, f_m$ be defined as in (1.14). Consider a system of simultaneous multivariate equations (SME) as given below:

$$\begin{cases} y_1 = f_1(x_1, x_2, \ldots, x_n) \\ y_2 = f_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ y_m = f_m(x_1, x_2, \ldots, x_n). \end{cases}$$

Finding the solution $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ for above given simultaneous system of equations is known as SME-problem. This proven to be a challenging and NP-complete problem [127]. For fixed $d = 2$, the SME problem referred as *Multivariate Quadratic* (MQ) problem.

Any high degree multivariate polynomial can be reduced to the multivariate quadratic polynomial. Therefore, finding the solution of a high degree system of multivariate polynomial equations is equivalent to finding the solution of a multivariate quadratic system of equations [38]. Therefore cryptographers worldwide are focusing on designing public key cryptographic protocols rely on MQ problem instead of SME problem. In literature, numerous public key cryptosystems have been proposed utilizing the MQ problem [15, 38]. Researchers have attempted to solve the random MQ problem in polynomial time using algebraic techniques like Gröbner bases technique, XL algorithm, Faugére's $F_4$ algorithm, etc. [38, 127]. To find the solution using these algorithms assumes certain strong conditions. Therefore, solving the system of random multivariate polynomials over finite field is still an open research problem.

A Multivariate Public Key Cryptosystem (MPKC) [38] is dependent on the trapdoor property of finding the solution of multivariate quadratic system of equations problem over finite field $\mathbb{F}_q$. In MPKC, the cipher is obtained by the multivariate quadratic map $\bar{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, i.e., $\bar{F}(x_1, \ldots, x_n) = (f_1, \ldots, f_m)$, where each $f_i \in \mathbb{F}_q[x_1, x_2, \ldots, x_n]$ is a polynomial and is represented as:

$$f_i(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} \gamma_{ij} x_i x_j + \sum_{i=1}^{n} \beta_i x_i + \alpha \tag{1.15}$$

for some coefficients $\gamma_{ij}, \beta_i, \alpha \in \mathbb{F}_q$. A typical construction of this kind of system begins with building a central map $F : \mathbb{F}_q^n \to \mathbb{F}_q^m$, which fulfills the following two conditions:

(*i*) $F(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$ where $f_i \in \mathbb{F}_q[x_1, \ldots, x_n]$;

(*ii*) For the decryption/signature process a legitimate user can solve the equation of type $F(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$ in polynomial time and equivalently, a legitimate user can efficiently find the pre-image of $(y_1, \ldots, y_m)$, which must be unique, and is denoted as: $F^{-1}(y_1, \ldots, y_m)$.

Once such a map is found, we define a public key $\bar{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ as:

$$\bar{F} = L \circ F \circ M$$

where the maps $L : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $M : \mathbb{F}_q^n \to \mathbb{F}_q^n$ are two randomly chosen bijective affine mappings and serve the purpose of masking the function $F$. The mappings $L$, $M$ and $F$ are part of the secret key.

**Encryption Scheme** ($m \geq n$)

*Encryption*: Suppose message $z = (z_1, \ldots, z_n) \in \mathbb{F}_q^n$, user need to simply computes $e = \bar{F}(z)$. The encrypted message (or, ciphertext) of the message $z$ is $e \in \mathbb{F}_q^m$.

*Decryption*: To decrypt the cipher text $e \in \mathbb{F}_q^m$, user need to compute $z = \bar{F}^{-1}(e) = M^{-1} \circ F^{-1} \circ L^{-1}(e) \in \mathbb{F}_q^m$.

**Signature Scheme** ($m \leq n$)

*Signature scheme*: Suppose a message $y = (y_1, \ldots, y_m)$ need to be signed by a user (system). Then, user (system) need to find the solution of system of equations $\bar{F}(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$. The solution $x = (x'_1, \ldots, x'_n)$ will be the signed document of message $y$.

*Verification*: Any user (system) can authenticate the signature scheme by verifying the equation $\bar{F}(x'_1, \ldots, x'_n) = (y_1, \ldots, y_m)$.