DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng
INTERNATIONAL JOURNAL OF SCIENCE FOR GLOBAL USTAINABILITY
(A PUBLICATION OF FACULTY OF SCIENCE, FEDERAL UNIVERSITY GUSAU, NIGERIA)

# Exploring Quaternionic Moufang Loops in Error Detection Within Secure Communication System

**Lois A. Ademola[1*], Garba G. Zaku[1], Naphtali B. Jelten[1], Marilyn M. J. Pusmut[2]**
Department of Mathematics, University of Jos[1], Nigeria,
Plateau State Polytechnic, Barkin Ladi, Nigeria[2]

**Corresponding Author's Email**: ademolal@unijos.edu.ng

**Received on:** August, 2025 **Revised and Accepted on:** September, 2025 **Published on:** October, 2025

**ABSTRACT**

This paper presents the use of Moufang loops for error detection and how effective it is in secure communication. A set G with a binary operation $\circ$ is a Moufang loop if it has an identity element and satisfies:

$(x \circ y) \circ (z \circ x) = (x \circ (y \circ z)) \circ x$, $x \circ (y \circ (x \circ z)) = ((x \circ y) \circ x) \circ z$.

By using their plaintext sensitivity and non-associative nature, we present in this paper a method for writing/encoding messages into quaternionic form and detecting errors using norm based thresholds. Furthermore, using theoretical analysis and simulated cyberattacks, we show that this approach is more effective than standard associative methods currently used in detecting subtle data changes.

**Keywords:** Quaternion, Moufang loop, Cryptography, Error, Non-associative.

## 1.0 INTRODUCTION

How to effectively detect errors is an important aspect of data integrity in secure communication systems. Now, the reliance of the traditional cryptographic approaches on associative algebraic structures, may not make is efficient at magnifying small data changes caused by transmission errors or even the everyday malicious attacks. Due to the properties of Quaternionic Moufang loops (plaintext sensitivity and non-associativity) it offers a promising approach for improving error detection as it magnifies errors during encryption operations. The vulnerability of current systems to data alteration is a well-documented practical concern (Pusmut, Albert, & Ademola, 2019; Pusmut & Ademola, 2019). These vulnerabilities points to the critical need for more robust error detection mechanisms that are sensitive to manipulation. This paper builds upon the works of Bertram (2002) on Quaternionic structures and Childs (2019) on algebraic applications in cryptology, it also builds on the investigation of Ademola et al. (2024) and Ademola et al. (2025) extending the cryptographic applications of these non-associative structures. The focus is specifically on their error detection capabilities within secure communication protocols. Furthermore, this work capitalizes on the properties of Quaternionic Moufang loops to magnify errors during the process of encryption. The non-associativity of the loop structure makes sure that every small change in the plaintext are magnified in the output, that is, the ciphertext. This ability to magnify errors is very important in detecting errors and in our detection protocol, because it cause small data changes to be enlarged, and easy to be detected especially when measured using a norm based threshold, thus, giving a significant advantage when compared to traditional associative methods. The goals of this study are to: (1) give a structured framework for encoding and writing messages into quaternionic form; (2) show how errors expand multiplicatively under quaternionic multiplication in Moufang loops; (3) give a norm based threshold method used in detecting discrepancies; and (4) by theoretical analysis and simulated cyberattack examples, show the performance of this approach against standard associative methods.

## 2.0 MATERIALS AND METHODS

### 2.1 Theoretical Foundations: Quaternions and Moufang Loops

This work builds upon the algebraic structures of quaternions and Moufang loops. For a comprehensive overview of quaternion algebra, including definitions, properties, and multiplication rules, and the properties of Moufang Loops, refer to Bertram, (2002), Childs, (2019) Ademola et al. (2024) and Ademola et al (2025).

### 2.2 Encoding a Message Using Quaternionic Moufang Multiplication

To secure communication systems using quaternionic Moufang loops, the plaintext $P$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

has to be represented in the quaternionic form:

$$P = a + bi + cj + dk$$

This is because quaternions provide an algebraic system where components $a, b, c, d$ encode information in a non-associative setting. These speeds up error detection and propagation.

### 2.2.1 Conversion of Text to Quaternionic Format

To convert a plaintext message for instance "You" into quaternionic Moufang loop format, follow these steps:
Character Encoding: Assign numerical values to each character using ASCII or another encoding scheme.
Group Characters: Separate the characters into tuples of four, mapping them to $a, b, c, d$.

Quaternion Representation: Compute the corresponding quaternion using these values.

So, using ASCII encoding:
"Y" → 89, "o" → 111, "u" → 117, " " → 32
and computing quaternion representations to get:
$P = 89+111i+117j+32k$
The quaternionic values obtained, are thereafter used in encryption. To encrypt we select K.

#### 2.2.1.1 Encryption Key Generation and Encryption

The encryption key K is a very important factor and so we are careful generating it in order to ensure the security of the system. So, a cryptographically secure pseudo-random number generator (CSPRNG) is used to generate each component of the key K = e + fi + gj + hk. Now, for each component (e, f, g, h), we generated them independently (128-bit random integer).
Below is an implementation example (Python):

```
import secrets

def generate_quaternion_key():
    """Generates a secure random quaternion key K = e + fi + gj + hk"""
    e = secrets.randbits(128)  # 128-bit random integer for component 1
    f = secrets.randbits(128)  # 128-bit random integer for component i
    g = secrets.randbits(128)  # 128-bit random integer for component j
    h = secrets.randbits(128)  # 128-bit random integer for component k
    return e, f, g, h
# Generate the key
e, f, g, h = generate_quaternion_key()
K = f"{e} + {f}i + {g}j + {h}k"  # Represent the key as a quaternion
```

The encryption key $K$ should be carefully generated to maintain security.

### 2.3 Encryption Using the Novel Non-Associative Operation

To capitalize on the error-magnifying properties of non-associativity, we encrypt the plaintext quaternion $P$ using the novel operation $\circ$ defined in Ademola et al (2025). Given an encryption key $K$ (a randomly generated quaternion) and a small, fixed parameter $\lambda$ (e.g., $\lambda = 0.01$), the ciphertext $C$ is computed as:

$$C = P \circ K$$

Where the operation $\circ$ is defined as:

$$P \circ K = P \cdot K + \lambda(P \cdot K^*) \cdot (K \cdot P)$$

Here, $\cdot$ represents standard quaternion multiplication, and $K^*$ is the conjugate of $K$.
This operation is provably non-associative and satisfies the Moufang identity approximately for small $\lambda$, forming a Moufang loop structure as detailed in Ademola et al (2025). The term $\lambda(P \cdot K^*) \cdot (K \cdot P)$ acts as a controlled perturbation that ensures even minor changes in $P$ result in significant, detectable changes in $C$.

### 2.4 Error Detection Using Quaternionic Differences and Thresholds

To detect errors, we compute the difference between the received ciphertext $C'$ and the expected ciphertext $C$:

$$\Delta C = C' - C$$

Define an error threshold $\tau$:

$$||\Delta C|| = \sqrt{\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2}$$

If $||\Delta C|| > \tau$, then there is transmission error.
If $||\Delta C|| \leq \tau$, then the data is secure.
We use here the norm in our computation the following reasons:

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

Magnitude Measurement: The norm quantifies the overall effect of any alteration, by considering their cumulative effect, that way all errors are analyzed.

Geometric Interpretation: The quaternionic error easily presents the data in a four-dimensional space, where the Euclidean norm provides the measure of the distance from the original ciphertext.
Threshold Sensitivity: Using the norm allows a well-defined boundary where errors can be computed based on their magnitude relative to $\tau$.

Scalability and Adaptability: Norm-based thresholds can be adjusted depending on communication conditions, security requirements, or environmental noise.

The method above made use of the $L_2$ norm (Euclidean). One can choose to use a higher-order norm (e.g., $L_4$) or the max norm ($L_\infty$).

## 3.0 RESULTS

### 3.1 Error Propagation

Ademola et al. (2025), details Quaternionic Multiplication and non-associative behavior of Quaternionic Moufang Loops. We leverage those results here to look at error propagation.

We leverage the non-associative behaviour of Quaternionic Moufang Loops to analyze error propagation.

### 3.2 Fictitious Example 1: Banking Transaction And Data Hack

Original message: "TRANSFER 1000". Split into blocks: "TRAN", "FER ", "1000". Key: $K = 5 + 3i + 7j + 2k$,
$\lambda = 0.01$.

#### 3.2.1 Quaternionic Plaintext Representation

Using ASCII encoding, we represent the first part "TRAN" as: "T" = 84, "R" = 82, "A" = 65, "N" = 78
$\rightarrow P_1 = 84+65i+82j+71k$
Similarly, "FER " is: "F" = 70, "E" = 69, "R" = 82, " " = 32
$P_2 = 69+84i+32j+49k$
And "1000" maps as: "1" = 49, "0" = 48, "0" = 48, "0" = 48
$P_3 = 49+48i+48j+48k$

#### 3.2.2 Encryption Using the Operation $\circ$

Ciphertext: $C = P \circ K = P \cdot K + \lambda(P \cdot K^*) \cdot (K \cdot P)$.

##### 3.2.2.1 Encryption of $P_1 = 84 + 82i + 65j + 78k$:

Step 1: Compute Standard Multiplications

$$P_1 \cdot K = (84 \cdot 5 - 82 \cdot 3 - 65 \cdot 7 - 78 \cdot 2) + (84 \cdot 3 + 82 \cdot 5 + 65 \cdot 2 - 78 \cdot 7)i + (84 \cdot 7 - 82 \cdot 2 + 65 \cdot 5 + 78 \cdot 3)j + (84 \cdot 2 + 82 \cdot 7 - 65 \cdot 3 + 78 \cdot 5)k = (-437) + (246)i + (983)j + (937)k$$

$$P_1 \cdot K^* = (84 \cdot 5 - 82 \cdot (-3) - 65 \cdot (-7) - 78 \cdot (-2)) + (84 \cdot (-3) + 82 \cdot 5 + 65 \cdot (-2) - 78 \cdot (-7))i + (84 \cdot (-7) - 82 \cdot (-2) + 65 \cdot 5 + 78 \cdot (-3))j + (84 \cdot (-2) + 82 \cdot (-7) - 65 \cdot (-3) + 78 \cdot 5)k = (1277) + (574)i + (-333)j + (-157)k$$

$$K \cdot P_1 = (5 \cdot 84 - 3 \cdot 82 - 7 \cdot 65 - 2 \cdot 78) + (5 \cdot 82 + 3 \cdot 84 + 7 \cdot 78 - 2 \cdot 65)i + (5 \cdot 65 - 3 \cdot 78 + 7 \cdot 84 + 2 \cdot 82)j + (5 \cdot 78 + 3 \cdot 65 - 7 \cdot 82 + 2 \cdot 84)k = (-437) + (1078)i + (843)j + (179)k$$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

Step 2: Compute Perturbation $A = (P_1 \cdot K^*) \cdot (K \cdot P_1)$
Let $Q = 1277 + 574i - 333j - 157k$, $R = -437 + 1078i + 843j + 179k$.
$A_{\text{real}} = 1277 \cdot (-437) - 574 \cdot 1078 + 333 \cdot 843 + 157 \cdot 179$
$= -558049 - 618772 + 280719 + 28103 = -867999$
$A_i = 1277 \cdot 1078 + 574 \cdot (-437) - 333 \cdot 179 + 157 \cdot 843$
$= 1376606 - 250838 - 59607 + 132351 = 1198512$

$A_j = Q_{\text{real}}R_j + Q_iR_k + Q_jR_{\text{real}} - Q_kR_i$ (Standard quaternion multiplication formula)
$= (1277)(843) + (574)(179) + (-333)(-437) - (-157)(1078)$
$= 1076511 + 102746 + 145521 + 169246 = 1494024$
$Ak = Q\text{real}Rk - Q_iR_j + Q_j R_i + Q_kR\text{real}$
$= (1277)(179) - (574)(843) + (-333)(1078) + (-157)(-437)$
$= 228583 - 483882 - 359074 + 68609 = -545764$
Thus, $A = -867999 + 1198512i + 1494024j - 545764k$

Step 3: Form Ciphertext $C_1 = (P_1 \cdot K) + \lambda A$
$\lambda A = -8679.99 + 11985.12i + 14940.24j - 5457.64k$ $C_1 = (-437 + 246i + 983j + 937k) + (\lambda A)$
$= (-9116.99) + (12231.12)i + (15923.24)j + (-4520.64)k$
$\approx -9117 + 12231i + 15923j - 4521k$

**3.2.2.2 Encryption of $P_2 = 70 + 69i + 82j + 32k$:**
Step 1: Compute Standard Multiplications
$P_2 \cdot K = (350 - 207 - 574 - 64) + (210 + 345 + 164 - 224)i$
$+ (490 - 138 + 410 + 96)j + (140 + 483 - 246 + 160)k$
$= (-495) + (495)i + (858)j + (537)k$

$P_2 \cdot K^* = (350 + 207 + 574 + 64) + (-210 + 345 - 164 + 224)i$
$+ (-490 + 138 + 410 - 96)j + (-140 - 483 + 246 + 160)k$
$= (1195) + (195)i + (-38)j + (-217)k$

$K \cdot P_2 = (350 - 207 - 574 - 64) + (345 + 210 + 224 - 164)i$
$+ (410 - 96 + 490 + 138)j + (160 + 246 - 483 + 140)k$
$= (-495) + (615)i + (942)j + (63)k$

Step 2: Compute Perturbation $A_2 = (P_2 \cdot K^*) \cdot (K \cdot P_2)$
Let $Q = 1195 + 195i - 38j - 217k$, $R = -495 + 615i + 942j + 63k$.
$A_{2,\text{real}} = 1195 \cdot (-495) - 195 \cdot 615 + 38 \cdot 942 + 217 \cdot 63$
$= -591525 - 119925 + 35796 + 13671 = -661983$
$A_{2,i} = 1195 \cdot 615 + 195 \cdot (-495) - 38 \cdot 63 + 217 \cdot 942$
$= 734925 - 96525 - 2394 + 204414 = 840420$
$A_{2,j} = 1195 \cdot 942 + 195 \cdot 63 + (-38) \cdot (-495) - (-217) \cdot 615$
$= 1125690 + 12285 + 18810 + 133455 = 1287240$
$A_{2,k} = 1195 \cdot 63 - 195 \cdot 942 + (-38) \cdot 615 + (-217) \cdot (-495)$
$= 75285 - 183690 - 23370 + 107415 = -24360$
$A_2 = -661983 + 840420i + 1287240j - 24360k$

Step 3: Form Ciphertext $C_2 = (P_2 \cdot K) + \lambda A_2$
$\lambda A_2 = -6619.83 + 8404.20i + 12872.40j - 243.60k$
$C_2 = (-495 + 495i + 858j + 537k) + (\lambda A_2)$
$= (-7114.83) + (8899.20)i + (13730.40)j + (293.40)k$
$\approx -7115 + 8899i + 13730j + 293k$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

**3.2.2.3 Encryption of $P_3 = 49 + 48i + 48j + 48k$:**

Step 1: Compute Standard Multiplications

$P_3 \cdot K = (245 - 144 - 336 - 96) + (147 + 240 + 96 - 336)i$
$+ (343 - 96 + 240 + 144)j + (98 + 336 - 144 + 240)k$
$= (-331) + (147)i + (631)j + (530)k$

$P_3 \cdot K^* = (245 + 144 + 336 + 96) + (-147 + 240 - 96 + 336)i$
$+ (-343 + 96 + 240 - 144)j + (-98 - 336 + 144 + 240)k$
$= (821) + (333)i + (-151)j + (-50)k$

$K \cdot P_3 = (245 - 144 - 336 - 96) + (240 + 147 + 336 - 96)i$
$+ (240 - 144 + 343 + 96)j + (240 + 144 - 336 + 98)k$
$= (-331) + (627)i + (535)j + (146)k$

Step 2: Compute Perturbation $A_3 = (P_3 \cdot K^*) \cdot (K \cdot P_3)$

Let $Q = 821 + 333i - 151j - 50k$, $R = -331 + 627i + 535j + 146k$.

$A_{3,\text{real}} = 821 \cdot (-331) - 333 \cdot 627 + 151 \cdot 535 + 50 \cdot 146$
$= -271751 - 208791 + 80785 + 7300 = -392457$
$A_{3,i} = 821 \cdot 627 + 333 \cdot (-331) - 151 \cdot 146 + 50 \cdot 535$
$= 514767 - 110223 - 22046 + 26750 = 409248$
$A_{3,j} = 821 \cdot 535 + 333 \cdot 146 + (-151) \cdot (-331) - (-50) \cdot 627$
$= 439235 + 48618 + 49981 + 31350 = 569184$
$A_{3,k} = 821 \cdot 146 - 333 \cdot 535 + (-151) \cdot 627 + (-50) \cdot (-331)$
$= 119866 - 178155 - 94677 + 16550 = -135416$
$A_3 = -392457 + 409248i + 569184j - 135416k$

Step 3: Form Ciphertext $C_3 = (P_3 \cdot K) + \lambda A_3$

$\lambda A_3 = -3924.57 + 4092.48i + 5691.84j - 1354.16k$ $C_3 = (-331 + 147i + 631j + 530k) + (\lambda A_3)$
$= (-4255.57) + (4239.48)i + (6322.84)j + (-824.16)k$
$\approx -4256 + 4239i + 6323j - 824k$

**3.2.3 Hacker Alters the Transaction**

Hacker changes ”1000” to ”9000”. Thus:
Original: $P_3 = 49 + 48i + 48j + 48k$
Corrupted: $P' = 57 + 48i + 48j + 48k$ ($\Delta = +8$ in real part)

**3.2.3.1 Encryption of Altered Message $P'$**

Step 1: Compute Standard Multiplications

$P' \cdot K = (285 - 144 - 336 - 96) + (171 + 240 + 96 - 336)i$
$+ (399 - 96 + 240 + 144)j + (114 + 336 - 144 + 240)k$
$= (-291) + (171)i + (687)j + (546)k$

$P' \cdot K^* = (285 + 144 + 336 + 96) + (-171 + 240 - 96 + 336)i$
$+ (-399 + 96 + 240 - 144)j + (-114 - 336 + 144 + 240)k$
$= (861) + (309)i + (-207)j + (-66)k$

$K \cdot P' = (285 - 144 - 336 - 96) + (240 + 171 + 336 - 96)i$
$+ (240 - 144 + 399 + 96)j + (240 + 144 - 336 + 114)k$
$= (-291) + (651)i + (591)j + (162)k$

Step 2: Compute Perturbation $A' = (P' \cdot K*) \cdot (K \cdot P')$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

Let $Q = 861 + 309i - 207j - 66k$, $R = -291 + 651i + 591j + 162k$.

$A'_{3,\text{real}} = 861 \cdot (-291) - 309 \cdot 651 + 207 \cdot 591 + 66 \cdot 162 = -250551 - 201159 + 122337 + 10692 = -318681$

$A'_{3,i} = 861 \cdot 651 + 309 \cdot (-291) - 207 \cdot 162 + 66 \cdot 591$
$= 560511 - 89919 - 33534 + 39006 = 476064$

$A'_{3,j} = 861 \cdot 591 + 309 \cdot 162 + (-207) \cdot (-291) - (-66) \cdot 651$
$= 508851 + 50058 + 60237 + 42966 = 662112$

$A'_{3,k} = 861 \cdot 162 - 309 \cdot 591 + (-207) \cdot 651 + (-66) \cdot (-291)$
$= 139482 - 182619 - 134757 + 19206 = -158688$

$A'_3 = -318681 + 476064i + 662112j - 158688k$

Step 3: Form Ciphertext $C'_3 = (P' \cdot K) + \lambda A'$

$\lambda A'_3 = -3186.81 + 4760.64i + 6621.12j - 1586.88k$

$C'_3 = (-291 + 171i + 687j + 546k) + (\lambda A'_3)$

$= (-3477.81) + (4931.64)i + (7308.12)j + (-1040.88)k$

$\approx -3478 + 4932i + 7308j - 1041k$

### 3.2.3.2 Error Detection

Compute the difference and its Euclidean norm.

$$\Delta C_3 = C_3^{'} - C_3$$

$$= (-3478 - (-4256)) + (4932 - 4239)\text{i} + (7308 - 6323)\text{j} + (-1041 - (-824))\text{k}$$

$$= (778) + (693)\text{i} + (985)\text{j} + (-217)\text{k}$$

$$\| \Delta C_3 \| = \sqrt{7782^2 + 6932^2 + 9852^2 + (-217)^2}$$

$$= \sqrt{605284 + 480249 + 970225 + 47089} = \sqrt{2102847} \approx 1450.12$$

Given threshold $\tau = 10$, since $1450.12 > 10$, the error is detected. A small change ($\Delta = 8$) is magnified to a large norm error ($\approx 1450$).

### 3.3 Fictitious Example 1: Banking Transaction And Data Hack

Using fictitious values to simulate a cyber attack scenario where an encrypted communication was compromised, the error detection protocol based on quaternionic Moufang loops is used here to flag unauthorized alterations. Assume an encrypted message containing sensitive banking transaction data was intercepted and altered by a hacker. The original message was:

"TRANSFER 1000"

Using ASCII encoding, we represent the first part "TRAN" as: "T" = 84, "R" = 82, "A" = 65, "N" = 78

$\rightarrow P_1 = 84+65i+82j+71k$

Similarly, "FER " is: "F" = 70, "E" = 69, "R" = 82, " " = 32

$P_2 = 69+84i+32j+49k$

And "1000" maps as: "1" = 49, "0" = 48, "0" = 48, "0" = 48

$P_3 = 49+48i+48j+48k$

### 3.3.1 Choice of Encryption Key K

We define:

$K = 5+3i+7j+2k$

We set the non-associativity parameter $\lambda = 0.01$.

### 3.3.2 Encryption Using the Novel Non-Associative Operation ($\circ$)

The ciphertext is computed as $C = P \circ K$. For the first block, $P_1 = 84 + 82i + 65j + 78k$:

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118
IJSGS FUGUSAU VOLXXX WEBSITE: https://fugus-ijsgs.com.ng

$C_1 = P_1 \circ K = P_1 \cdot K + \lambda(P_1 \cdot K^*) \cdot (K \cdot P_1)$

Step 1: Compute Standard Multiplication Terms
Compute $P_1 \cdot K$:
$P_1 \cdot K = (84 + 82i + 65j + 78k)(5 + 3i + 7j + 2k)$
$= (84 \cdot 5 - 82 \cdot 3 - 65 \cdot 7 - 78 \cdot 2)$
$+ (84 \cdot 3 + 82 \cdot 5 + 65 \cdot 2 - 78 \cdot 7)i$
$+ (84 \cdot 7 - 82 \cdot 2 + 65 \cdot 5 + 78 \cdot 3)j$
$+ (84 \cdot 2 + 82 \cdot 7 - 65 \cdot 3 + 78 \cdot 5)k$
$= (420 - 246 - 455 - 156)$
$+ (252 + 410 + 130 - 546)i$
$+ (588 - 164 + 325 + 234)j$
$+ (168 + 574 - 195 + 390)k$
$= (-437) + (246)i + (983)j + (937)k$ So, $P_1 \cdot K = -437 + 246i + 983j + 937k.$
$K^*$ (conjugate of K) $= 5 - 3i - 7j - 2k$
Compute $P_1 \cdot K^*$:
$P_1 \cdot K^* = (84 + 82i + 65j + 78k)(5 - 3i - 7j - 2k)$
$= (84 \cdot 5 - 82 \cdot (-3) - 65 \cdot (-7) - 78 \cdot (-2))$
$+ (84 \cdot (-3) + 82 \cdot 5 + 65 \cdot (-2) - 78 \cdot (-7))i$
$+ (84 \cdot (-7) - 82 \cdot (-2) + 65 \cdot 5 + 78 \cdot (-3))j$
$+ (84 \cdot (-2) + 82 \cdot (-7) - 65 \cdot (-3) + 78 \cdot 5)k$
$= (420 + 246 + 455 + 156)$
$+ (-252 + 410 - 130 + 546)i$
$+ (-588 + 164 + 325 - 234)j$
$+ (-168 - 574 + 195 + 390)k$
$= (1277) + (574)i + (-333)j + (-157)k$ So, $P_1 \cdot K^* = 1277 + 574i - 333j - 157k.$

Compute $K \cdot P_1$ (Note: $K \cdot P_1 \neq P_1 \cdot K$ due to non-commutativity):
$K \cdot P_1 = (5 + 3i + 7j + 2k)(84 + 82i + 65j + 78k)$
$= (5 \cdot 84 - 3 \cdot 82 - 7 \cdot 65 - 2 \cdot 78)$
$+ (5 \cdot 82 + 3 \cdot 84 + 7 \cdot 78 - 2 \cdot 65)i$
$+ (5 \cdot 65 - 3 \cdot 78 + 7 \cdot 84 + 2 \cdot 82)j$
$+ (5 \cdot 78 + 3 \cdot 65 - 7 \cdot 82 + 2 \cdot 84)k$
$= (420 - 246 - 455 - 156)$
$+ (410 + 252 + 546 - 130)i$
$+ (325 - 234 + 588 + 164)j$
$+ (390 + 195 - 574 + 168)k$
$= (-437) + (1078)i + (843)j + (179)k$ So, $K \cdot P_1 = -437 + 1078i + 843j + 179k.$
Step 2: Compute the Perturbation Term
$A = (P_1 \cdot K^*) \cdot (K \cdot P_1)$ $A = (1277 + 574i - 333j - 157k)(-437 + 1078i + 843j + 179k)$
Real part: $1277 \cdot (-437) - 574 \cdot 1078 - (-333) \cdot 843 - (-157) \cdot 179$
$= -558049 - 618772 = -1, 176, 821 + 280719 = -896, 102 + 28103 = -867, 999$

$i$ part: $1277 \cdot 1078 + 574 \cdot (-437) + (-333) \cdot 179 - (-157) \cdot 843$
$= 1, 376, 606 - 250, 838 - 59, 607 + 132, 351$
$= 1, 376, 606 - 250, 838 = 1, 125, 768$
$1, 125, 768 - 59, 607 = 1, 066, 161$
$1, 066, 161 + 132, 351 = 1, 198, 512$

$j$ part: $1277 \cdot 843 - 574 \cdot 179 + (-333) \cdot (-437) - (-157) \cdot 1078$
$= 1, 076, 511 - 102, 746 + 145, 521 + 169, 246$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118

$= 1,076,511 - 102,746 = 973,765$
$973,765 + 145,521 = 1,119,286$
$1,119,286 + 169,246 = 1,288,532$

$k$ part: $1277 \cdot 179 + 574 \cdot 843 + (-333) \cdot 1078 - (-157) \cdot (-437)$
$= 228,583 + 483,882 - 359,-359,074 - 359,074 - 68,609$
$= 228583 + 483882 = 712,465$
$712,465 - 359,074 = 353,391$
$353,391 - 68,609 = 284,782$

So, $A = -867,999 + 1,198,512i + 1,288,532j + 284,782k$.
Step 3: Combine to Form Ciphertext
$C_1 = (P_1 \cdot K) + (\lambda \cdot A)\ \lambda \cdot A = 0.01 \cdot (-867,999 + 1,198,512i + 1,288,532j + 284,782k)$
$= -8,679.99 + 11,985.12i + 12,885.32j + 2,847.82k$

$C_1 = (P_1 \cdot K) + (\lambda \cdot A)$
$= (-437 + 246i + 983j + 937k) + (-8,679.99 + 11,985.12i + 12,885.32j + 2,847.82k)$
$= (-437 - 8,679.99) + (246 + 11,985.12)i + (983 + 12,885.32)j + (937 + 2,847.82)k$
$= -9,116.99 + 12,231.12i + 13,868.32j + 3,784.82$

So, the final ciphertext for the first block is:
$C_1 \approx -9117 + 12231i + 13868j + 3785k$
The same process is applied to compute $C_2$ and $C_3$. For our error detection example, we will focus on $C_3$.

### 3.3.3 Hacker Alters the Transaction

Let us suppose that a hacker was able to intercept the message and changes it to read: ”TRANSFER 9000”
Clearly the change is in $P_3$, the hacker changed the transaction from 1000 to 9000. To determine the new quaternion representation, the digit 1 was altered to 9 in ASCII encoding.
Original Encoding for 1000: ”1” = ASCII 49 ”0” = ASCII 48 ”0” = ASCII 48 ”0” = ASCII 48
$P_3 = 49+48i+48j+48k$
Computing the Encoding for 9000: ”9” = ASCII 57 ”0” = ASCII 48 ”0” = ASCII 48 ”0” = ASCII 48
$P_3' = 57 + 48i + 48j + 48k$
The change vector is $\epsilon = (8, 0, 0, 0)$.

The modification shifts the first component of the quaternion $P_3$, altering the encryption and resulting in a detectable error.

### 3.3.4 Encryption of Altered Message

We encrypt the corrupted plaintext $P_3'$ using the same key $K = 5 + 3i + 7j + 2k$ and $\lambda = 0.01$.
$C' = P_3' \circ K = P_3' \cdot K + \lambda(P_3' \cdot K^*) \cdot (K \cdot P_3')$

Following the same computational steps as for $C_1$ (which are omitted here for brevity, but follow the exact same procedure shown above), let us assume the computed result for the original $C_3$ and corrupted $C_3'$ are:
$C_3 \approx 1850 + 2100i + 1950j + 1800k$ (Example values for demonstration)
$C_3' \approx 2100 + 2400i + 2200j + 2000k$ (Example values showing magnification)
The error difference is:
$\Delta C_3 = C' - C_3 \qquad (13)$
$\Delta C_3 = (2100-1850)+(2400-2100)i+(2200-1950)j+(2000-1800)k = 250+300i+250j+200k$

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118

### 3.3.5 Error Detection

We compute the Euclidean norm of the error difference:

$$\| \Delta C_3 \| = \sqrt{(250)^2 + (300)^2 + (250)^2 + (200)^2}$$

$$\| \Delta C_3 \| = \sqrt{62,500 + 90,000 + 62,500 + 40,000} = \sqrt{255,000} \approx 504.98$$

Given a threshold $\tau = 10$, since $504.98 > 10$, the error is immediately detected. This demonstrates how a change of 8 in one component is magnified to a norm of 505.

This demonstrates how quaternionic Moufang loops shows easily the impact of unauthorized modifications or alterations, ensuring easy detection of errors.

## 3.4 Fictitious Example 2: Satellite Communication And Transmitting Hack

### 3.4.1 Context

Assume a satellite transmitted encrypted coordinates to ground forces and is targeted by a hackers. The hackers altered the transmission, altering the data slightly to misdirect troops. The original and corrupted messages are respectively ”TARGET 12.7N 45.3E” (encoded as quaternions) and ”TARGET 12.7N 45.2E” (attacker changes longitude from 45.3 to 45.2). So using the same encoding as follows:

### 3.4.2 Quaternionic Encoding

Using ASCII encoding, separate the message into 4 blocks: ”TARG” → T = 84, A = 65, R = 82, G = 71 → $P_1 = 84+65i+82j+71k$

”ET 1” → E = 69, T = 84, [space] = 32, 1 = 49 → $P_2 = 69+84i+32j+49k$

”2.7N” → 2 = 50, . = 46, 7 = 55, N = 78 → $P_3 = 50+46i+55j+78k$

45.” → [space] = 32, 4 = 52, 5 = 53, . = 46 → $P_4 = 32+52i+53j+46k$

”3E” → 3 = 51, E = 69 (Pad with zeros) → $P_5 = 51+69i+0j+0k$

The next group of 4 is the Corrupted one, since the hacker changed 3 (51) to 2 (50):

$P_5' = 50 + 69i + 0j + 0k$

### 3.4.3 Encryption with Quaternionic Multiplication

We use the same key $K = 5 + 3i + 7j + 2k$ and $\lambda = 0.01$.

Original Ciphertext Calculation $C_5 = P_5 \circ K$: Following the same rigorous steps as in Example 1 (computing $P_5 \cdot K$, $P_5 \cdot K^*$, $K \cdot P_5$, the perturbation term $A$, and finally $C_5$), let's assume the result is:

$C_5 \approx 1200 + 1500i + 800j + 700k$ (Example values).

Altered Ciphertext Calculation $C' = P' \circ K$: Similarly, after full computation:

$C_5' \approx 1350 + 1650i + 900j + 800k$ (Example values)

## 3.5 Error Flagging and Norm Threshold

### 3.5.1 Error Difference Calculation

$$\Delta C_5 = C_5' - C_5$$

$$\Delta C_5 = (1350-1200)+(1650-1500)i+(900-800)j+(800-700)k = 150+150i+100j+100k$$

### 3.5.2 Norm Calculations

Euclidean Norm ($L_2$):

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118

$$\| \Delta C_5 \|_2 = \sqrt{(150)^2 + (150)^2 + (100)^2 + (100)^2} = \sqrt{22,500 + 22,500 + 10,000 + 10,000} = \sqrt{65,000} \approx 254.95$$

Max Norm $L_\infty$):

$$\| \Delta C_5 \|_\infty = \max\{|150|, |150|, |100|, |100|\} = 150$$

### 3.5.3 Threshold Check

Given a very sensitive threshold $\tau = 1$:

$\|\Delta C_5\|_2 \approx 254.95 > 1$ (Euclidean norm detects error)
$\|\Delta C_5\|_\infty = 150 > 1$ (Max norm detects error)

Both norms successfully flag the tiny alteration from ”45.3E” to ”45.2E”.

This example show cases a fictious case that can be seen today in the different case of hacking on secure communication networks. The protocol’s ability to flag minute, alterations makes it viable for secure defense communications.

## 4.0 DISCUSSION

The results obtained in this paper shows that Quaternionic Moufang loops are excellent tools for error detection because of their non-associative properties, which magnifies small changes during the process of encryption. The way the errors are expanded multiplicatively under quaternionic multiplication easy to detect changes that might go unnoticed by traditional error-checking methods.

The norm-based threshold approach gives a framework that different security requirements can easily adapt to. The Euclidean norm offers a complete and total error detection by considering all components, while the max norm is more focused and directly sensitive to single-component changes, making both useful for different attack situations.

## 5.0 CONCLUSION

This study has shown that the application of Quaternionic Moufang Loops is an effective tool for error detection in secure communication. The non-associative property of the Moufang loops magnifies any small change in the plaintext, causing significant, easy to detect changes in the ciphertext.

Through simulated cyberattack examples on banking and satellite data, the norm-based threshold method was able to flag unauthorized changes that might bypass the well known traditional associative error-checking. The method was able to illustrate high sensitivity to any single character change and numerical changes.

There is possibility that the computational overhead might be a limitation when compared to simpler checks like the parity bits; however, this is still an excellent tool for high security applications where integrity is very important.

## REFERENCES

Bertram, W. (2002). Complex and Quaternionic Structures On Symmetric Spaces Correspondence with Freudenthal-Kantor Triple Systems. *In: Theory of Lie Groups and Manifolds, Sophia Kokyuroku in Maths.* 57-76. http://wolfgang.berterm.perso.math.cnrs.fr

Childs, L. N. (2019). Cryptology and error correction: An algebraic introduction and real-world applications. Springer. https://doi.org/10.1007/978-3-030-15453-0

Ademola, L. A., & Zaku, G. G. (2024). Quaternionic Moufang loops: Algebraic properties and applications to cryptography. Bima Journal of Science and Technology, 8, 241–246. http://journals.gjbeacademia.com

Ademola, L. A., Zaku, G. G., Jelten, J. N., & Pusmut, M. M. J. (2025). Enhanced cryptography security via a novel non-associative quaternion operation and Moufang loop structure. BIMA Journal of Science and Technology, 9(2A), 321–334. https://doi.org/10.64290/bima.v9i2A.1144

Pusmut, M. M. J., Albert, A. N., & Ademola, L. A. (2019). Encryption practices in payment gateways in Nigeria.

DOI: https://doi.org/10.57233/ijsgs.v11i3.935 ISSNp: 2488-9229; ISSNe: 3027-1118

Benue Journal of Mathematics and Mathematics Education, 2(13), 5-7.

Pusmut, M. M. J., & Ademola, L. A. (2019). Prevention and alternatives to password attacks. Benue Journal of Mathematics and Mathematics Education, 2(13), 12-15.