Generalization of Lattice-Based Cryptography on Hypercomplex Algebras

Sonika Singh, Sahadeo Padhye, and Ankal Pal

1 Introduction

Quantum computing is not a far-fetched reality. The mathematical model of quantum computing was first proposed by Feynman [18], but the practical implementation has still engineering limitations pertaining to cryogenics. IBM has come up with its small quantum computers, which can solve hard optimization problems [20]. In the post-quantum era, computational capabilities would increase exponentially. By Shor's algorithm [17], the discrete logarithm problem (DLP) and factorization problems can be solved efficiently. It poses a direct threat to the security of the elliptic curve discrete logarithm problem (ECDLP) and RSA cryptosystems. The national institute of standards and technologies (NIST) has emphasized the efficacy of quantum-resistant algorithms to be used in the future. In 2019, NIST already announced the second round of candidates for post-quantum cryptography [21]. Lattice-based schemes are inherently quantum-resistant for higher dimensions ($n \geq 100$). The recent implementation of the "NewHope" [19] software has shown that it is possible to have a hardware implementation of lattice-based protocols with memory and speed constraints. This provides us with new hope that similar implementations are possible for our proposed scheme, STRU cryptosystem, based on the shortest

S. Singh (✉)
Department of Mathematics, CMP Degree College, University of Allahabad, Pryagraj, India
e-mail: sonikasinghcool10@gmail.com

S. Padhye
Department of Mathematics, Motilal Nehru National Institute of Technology Allahabad, Pryagraj, India
e-mail: sahadeomathrsu@gmail.com

A. Pal
Department of Mathematics, University of L'Aquila, L'Aquila, Italy
e-mail: ankanpal100@gmail.com

P. Stănică et al. (eds.), *Security and Privacy*, Lecture Notes in Electrical Engineering 744,
https://doi.org/10.1007/978-981-33-6781-4_6

vector problem (SVP) on sedenion algebra obtained through the Cayley–Dickson process.

A sequence of algebras was constructed by A. Cayley and L. E. Dickson over the field of real numbers by defining specific/compatible multiplication and conjugation rules in such a way that each algebra has twice the dimension of the previous algebra.

In this construction, we keep losing some important properties of the algebras at each step. The gradual loss of properties manifests in the depletion in the algebraic structure of the higher dimensional algebras. When we double $\mathbb{R}$, we get $\mathbb{C}$. The ordering property of $\mathbb{R}$ is lost. Similarly, when we construct quaternions ($\mathbb{H}$) (Dimension 4), the commutative property is lost. Continuing the CD process, we arrive at octonions ($\mathbb{O}$) from quaternions; the associative property is lost. Continuing this process, we proceed further and construct the sedenions $\mathbb{S}$, and we observe that it is non-alternative. We can continue this process and attain the concept of generalized 2^n-ions. The power associativity remains intact for 2^n-ions, which gives us the freedom to construct and manipulate polynomials [3, 14, 16].

The question is "Why should one use these algebras for Cryptography?" The pivotal motivation is that a set of hierarchies can be developed. Moreover, the algebras provide a way for using obfuscation techniques as the number of bases in the higher dimensional algebra are more. Hence, heuristically we can safely assume that it might provide more security and interesting application scenarios. During the instantiation of this type of framework, we define a new property "inverse associative property (IAP)" for the composition of the basis elements.

The remaining parts are given as follows: in Sect. 2, we discuss cryptographic hierarchies. In Sect. 3, we discuss the algebraic structure of the sedenion algebra for the proposed STRU scheme. In the next section, we present our desired as well as anticipated inverse associative property. We propose the STRU scheme and its decryption verification in Sect. efsec5. Consequently, we analyze the proposed scheme in the context of different attacks in Sect. 6. In Sect. 7, we provide a comparative analysis of the generalized structure. After that, we conclude the article in the last section.

2 Cryptographic Hierarchies

The aim of the cryptographic hierarchy is to create different security levels using the same protocol [10]. It is an attempt to vary the security levels by keeping the same encryption–decryption process but changing the base algebraic configuration.

As the SVP is implemented over the quaternions, we notice that although it is four times slower than NTRU [7], QTRU [11] is much more secure to lattice-based attacks than NTRU. Hence, one can easily compensate for speed loss by reducing the dimension of the three parameters (N, p, q) and still gaining the same level of security. Similar arguments can be made for the non-associative counterpart OTRU [2]. The way forward from OTRU needs a structured framework as they are no more division algebras. We have implemented SVP on sedenions, and we call it STRU cryptosystem. A detailed explanation of the proposed scheme is provided in Sect. 7.

Table 1 A general hierarchy

Cryptosystem	Underlying hard lattice problem	Dimension
GTRU (Power associative but other properties lost)	SVP (Shortest Vector Problem)	2^n $(n = 5, 6, \ldots)$
STRU (Non-alternative)	SVP	2^4
OTRU [12] (Non-associative)	SVP	2^3
QTRU [11] (Non-commutative)	SVP	2^2
G-NTRU [9] (Non-ordered)	SVP	2^1
NTRU [7]	SVP	2^0

3 Sedenion Algebra

As we have discussed earlier that if we apply the CD construction to the octonions (an eight-dimensional non-commutative and non-associative algebra over the reals) [1], then we can obtain a 16-dimensional non-commutative, non-associative, and non-alternative algebra over the reals. This algebra is called sedenion algebra. We denote the set of sedenions by $\mathbb{S}$. The addition and the subtraction of sedenions are coefficient-wise, and the multiplication of sedenions is non-commutative and non-associative. Sedenions are power associative and flexible. Since sedenions have zero divisors, they are not division algebra [8]. For more details for sedenion algebra, please refer to Imaeda's work [8] or [13].

Sedenions: The real sedenions denoted by $\mathbb{S}$ can be viewed as an algebra of dimension 16 over real number field $\mathbb{R}$. We define $\mathbb{S}$ by

$$\mathbb{S} = \{y_0 + \Sigma_{i=1}^{15} y_i k_i : y_0, \ldots, y_{15} \in \mathbb{R}\},$$

where y_i's are the real scalar values and the set $\{1, k_1, \ldots, k_{15}\}$ are the basis elements (unit sedenions, we are using $k_0 = 1$). For our implementation, we consider only integer coefficients because of the modularity restrictions.

We elaborate the structure of $\mathcal{A}$, $\mathcal{A}_p$, and $\mathcal{A}_q$ sets which contains the desired polynomials of integer coefficients. Considering the convolution polynomial rings $\mathcal{R} = \frac{Z[x]}{x^N-1}$, $\mathcal{R}_p = \frac{Z_p[x]}{x^N-1}$, and $\mathcal{R}_q = \frac{Z_q[x]}{x^N-1}$, the structures of $\mathcal{A}$, $\mathcal{A}_p$, and $\mathcal{A}_q$ are given by

$$\mathcal{A} = \{a_0 + \Sigma_{i=1}^{15} a_i(x)k_i : a_0(x), ..., a_{15}(x) \in \mathcal{R}\},$$
$$\mathcal{A}_p = \{a_0 + \Sigma_{i=1}^{15} a_i(x)k_i : a_0(x), ..., a_{15}(x) \in \mathcal{R}_p\},$$
$$\mathcal{A}_q = \{a_0 + \Sigma_{i=1}^{15} a_i(x)k_i : a_0(x), ..., a_{15}(x) \in \mathcal{R}_q\}.$$

4 Inverse Associative Property in the Basis Elements of Sedenions

We assume that k_i are basis elements where $1 \leq i \leq 15$ and $i \in \mathbb{N}$ and $\circ$ denotes the sedenionic multiplication. We recall some properties of the basis elements which would be necessary to verify the inverse associativity property in the basis elements of $\mathbb{S}$:

Property-1 [8] (**Anti-Commutativity**):

$$-k_i \circ k_j = k_j \circ k_i$$
$$k_i \circ (-k_j) = k_j \circ k_i.$$

Property-2 [8]:

$$k_i \circ k_i = -k_0.$$

The second property is a particularly nice property because from it we can deduce that

$$-k_i = {k_i}^{-1}.$$

Types of Inverse Associativity Property: We assume that $f, g, h \in \{k_i \mid 1 \leq i \leq 15\}$ and $f \neq k_0, g \neq k_0, h \neq k_0$. We did the computations and found out that there are two types of inverse associativity that is followed by the basis elements:

1. Elements which satisfy *Inverse Associativity-D (Desired)* (IAP-D): $f \circ ((g \circ f) \circ h) = (g \circ h)$.

2. Elements which satisfy *Inverse Associativity-A (Anticipated)* (IAP-A): $f \circ ((g \circ f) \circ h) = (h \circ g)$.

We elaborate some of the algebraic manipulations here to show that the above properties which are the results of the computations can have many forms. We start from the following assumptions and analyze how Property-1 and Property-2 can be used to do the manipulations. We assume the following:

$$f \circ ((f^{-1} \circ g) \circ h) = (f \circ f^{-1}) \circ (g \circ h) = (g \circ h).$$

By using Property-2, it can be rewritten as

$$f \circ ((f^{-1} \circ g) \circ h) = (f \circ f^{-1}) \circ (g \circ h) = k_0 \circ (g \circ h).$$

Again, using Property-1, we can rewrite it as

$$f \circ ((f^{-1} \circ g) \circ h) = (f \circ f^{-1}) \circ (g \circ h) = (h \circ g).$$

We again use Property-2 to deduce that $f^{-1} = -f$:

$$f \circ ((-f \circ g) \circ h) = (h \circ g).$$

We again use Property-1 to deduce that

$$f \circ ((g \circ f) \circ h) = (h \circ g) \,(\text{InverseAssociativityProperty} - \text{A}).$$

We also define (*Inverse Associativity Property-D*):

$$f \circ ((g \circ f) \circ h) = (g \circ h).$$

Interestingly, we see that actually all the basis elements satisfy either of the two properties.

Extending the Property to Polynomials: We impose a condition that this property to be checked while constructing $\mathcal{A}$. Every polynomial in $\mathbb{S}$ will have a form:

$$p(x) = \sum_{i=0}^{N-1} a_i x^i,$$

where $a_i's$ are the sedenionic coefficients. Every $a_i's$ can be written in the form of the basis elements as

$$a_i = \sum_{j=0}^{15} y_{ij} k_j$$

, i.e., our $p(x)$ will be of the form:

$$p(x) = \sum_{i=0}^{N-1} (\sum_{j=0}^{15} y_{ij} k_j) x^i.$$

Hence, we need to multiply the basis elements and then do an iterative process. In this way, all the elements of $\mathcal{A}$ will follow either of the two properties. This is an additional computational task which needs to be performed for sedenions.

5 Proposed Scheme: STRU

There is an article, namely, "STRU: A Non-Alternative and Multi-dimensional Public Key Cryptosystem" given by Thakur and Tripathi in 2017 [15]. They proposed an STRU cryptosystem based on sedenions, but in the decryption process, they used associativity directly. Since we know that sedenions are non-associative, we cannot use associativity directly, as Thakur et al. did. So, their scheme does not follow the sedenionic requirements.

Here, we propose a cryptographic scheme based on sedenion algebra and overcome the flaw of Thakur et al. scheme. We use the inverse associative property in the decryption phase of the system. In our proposed scheme, the elements of $\mathcal{A}$ are taken and called sedenion polynomial for brevity. The encryption and decryption in STRU are done in a multi-dimensional space as in OTRU cryptosystem [12]. This cryptosystem has parameters (N, p, q) and four subsets L_f, L_g, L_ϕ, and L_m of $\mathcal{A}$. N, p, q, d_f, d_g, d_ϕ all are constant parameters and perform a similar role as in NTRU. The scheme of STRU cryptosystem is as follows.

Key-Generation: For generating key pairs, two random sedenion polynomials $F \in L_f$ and $G \in L_g$ are generated, where

$$F = f_0(x) + f_1(x)k_1 + \cdots + f_{15}(x)k_{15} : f_0, f_1, \ldots, f_{15} \in L_f,$$
$$G = g_0(x) + g_1(x)k_1 + \cdots + g_{15}(x)k_{15} : g_0, g_1, \ldots, g_{15} \in L_g.$$

The sedenion polynomial F should have an inverse over $\mathcal{A}_q$ and $\mathcal{A}_p$. If the inverses do not exist, then a new sedenion polynomial F will be generated. The inverse of F over $\mathcal{A}_p$ is denoted by F_p^{-1} and over $\mathcal{A}_q$ is by F_q^{-1}. Then, a new sedenion polynomial H is computed by

$$H = F_q^{-1} \circ G \in \mathcal{A}_q.$$

This sedenion polynomial H acts as the public key, and the sedenion polynomial pair (F, F_p^{-1}) is kept secret. When the same parameters are used in both cryptosystems, then the key-creation process in STRU is 256 times slower in comparison to NTRU.

Encryption: For the encryption process, firstly, a random sedenion polynomial $\phi \in L_\phi$ is generated. The message M, which is to be encrypted, is first expressed in terms of a sedenion polynomial. Consequently, the ciphertext E is obtained by

$$E = pH \circ \phi + M \in \mathcal{A}_q,$$

where, $\phi = \phi_0(x) + \phi_1(x)k_1 + \cdots + \phi_{15}(x)k_{15} : \phi_0, \phi_1, \ldots, \phi_{15} \in L_\phi$ and $M = m_0(x) + m_1(x)k_1 + \cdots + m_{15}(x)k_{15} : m_0, m_1, \cdots, m_{15} \in L_m$.
The encryption requires one sedenionic multiplication involving 256 convolution multiplications and 16 polynomial additions having complexity $O(N^2)$ and $O(N)$,

respectively. Each encryption round takes a total of 16 data vectors.

Decryption: For decryption, we compute

$$\begin{aligned} B =& (F \circ E) \bmod q = F \circ (pH \circ \phi + M) \bmod q \\ &= pF \circ (H \circ \phi) + F \circ M \bmod q \\ &= pF \circ ((F_q^{-1} \circ G) \circ \phi) + F \circ M \bmod q. \end{aligned}$$

Now we use IAP-D to get

$$B = pG \circ \phi + F \circ M \ \ mod\ q.$$

If we select advisable parameters, the coefficients of the 16 polynomials in $pG \circ \phi + F \circ M \in \mathcal{A}_q$ will fall into the range $(-q/2, +q/2\ ]$ so that the last reduction modulo q will not be required. So, we can proceed to the next step. $B \in \mathcal{A}_q$ should be analyzed with its corresponding candidate in $(-q/2, +q/2]$ and all coefficients in 16 polynomials should be reduced mod p. Thus, we get $(B \bmod p) = F \circ M \in \mathcal{A}_p$. To obtain the actual plaintext M, we multiply B on the left by ${F_p}^{-1}$.

The encryption and decryption algorithms in this cryptosystem are about 16 and 32 times slower than NTRU for similar dimension N. However, in STRU, we can deal with a lesser dimension N, without compromising security of the cryptosystem. Also, as in NTRU, the efficiency of STRU encryption/decryption may be optimized using various optimization methods [6] under appropriate assumptions. Additionally, a message of size $16N$ can be encrypted/decrypted in a single encryption/decryption process, whereas in NTRU the message of size $16N$ can be encrypted/decrypted using 16 times of encryption/decryption process.

5.1 Successful Decryption

We calculated the successful decryption probability in STRU cryptosystem as like as in the NTRU cryptosystem and by taking similar assumptions as in the standard version [9]. The decryption will be successful in STRU if all coefficients of $pG \circ \phi + F \circ M$ lie in $(\frac{-q+1}{2}, \frac{+q-1}{2})$. So, we have

$$B = F \circ E = F \circ (pH \circ \phi) + F \circ M = r_0 + \Sigma_{i=1}^{15} r_i(y) k_i,$$

where, for instance, r_0, a degree N polynomial is computed as

$$
\begin{aligned}
r_0 &= [r_{0,0}, r_{0,1}, ..., r_{0,N-1}],\\
r_1 &= [r_{1,0}, r_{1,1}, ..., r_{1,N-1}],\\
&\cdots\\
&\cdots\\
&\cdots\\
r_{15} &= [r_{15,0}, r_{15,1}, ..., r_{15,N-1}].
\end{aligned}
$$

If we consider all NTRU assumptions, then one can easily estimate the expected values for all coefficients of $r_0, r_1, ..., r_{15}$ in B will remain zero and their variances are 16 tuples. We assume that the coefficients of f_i, g_i, and ϕ_i are random independent variables and take one of the values from $\{1, 0, -1\}$. Then we can easily deduce

$$
\begin{aligned}
f_i &= [f_{i,0}, f_{i,1}, ..., f_{i,N-1}], i = 0, 1, ..., 15,\\
g_i &= [g_{i,0}, g_{i,1}, ..., g_{i,N-1}], i = 0, 1, ..., 15,\\
\phi_0 &= [\phi_{i,0}, \phi_{i,1}, ..., \phi_{i,N-1}], i = 0, 1, ..., 15.
\end{aligned}
$$

$\Pr(f_{i,j} = 1) = \frac{d_f}{N}$, $\Pr(f_{i,j} = -1) = \frac{d_f - 1}{N} \approx \frac{d_f}{N}$, $\Pr(f_{i,j} = 0) = \frac{N-2d_f}{N}$,
$\Pr(g_{i,j} = 1) = \frac{d_g}{N}$, $\Pr(g_{i,j} = -1) = \frac{d_g}{N}$, $\Pr(g_{i,j} = 0) = \frac{N-2d_g}{N}$,
$\Pr(\phi_{i,j} = 1) = \frac{d_\phi}{N}$, $\Pr(\phi_{i,j} = -1) = \frac{d_\phi}{N}$, $\Pr(\phi_{i,j} = 0) = \frac{N-2d_\phi}{N}$,
$\Pr(m_{i,j} = j) = \frac{1}{p}$ where $i = 0, 1, ..., 15$ and $j = \frac{-(p-1)}{2}, ..., \frac{-(p+1)}{2}$.

Considering the above assumptions, expected values are $E(f_{i,j}) \approx 0$, $E(g_{i,j}) = 0$, $E(r_{i,j}) = 0$, and $E(m_{i,j}) = 0$. Thus, $E(r_{i,j}) = 0, i = 0, 1, ..., 15$.

The variances are calculated as in NTRU, i.e., $Var[g_{i,l}\phi_{j,t}] = \frac{4d_g d_\phi}{N^2}$, $i, l = 0, 1, ..., 15$, $j, t = 0, 1, ..., N-1$,
$Var[f_{i,l}m_{j,t}] = \frac{d_f(p-1)(p+1)}{6N}$, $i, l = 0, 1, ..., 15$, $j, t = 0, 1, ..., N-1$.

Therefore,
$Var[r_{0,l}] = Var[\Sigma_{i+j=l \bmod N}(p.g_{0,i}\phi_{0,j} - p.g_{1,i}\phi_{1,j} - p.g_{2,i}\phi_{2,j} - p.g_{3,i}\phi_{3,j} - p.g_{4,i}\phi_{4,j} - p.g_{5,i}\phi_{5,j} - p.g_{6,i}\phi_{6,j} - p.g_{7,i}\phi_{7,j} - p.g_{8,i}\phi_{8,j} - p.g_{9,i}\phi_{9,j} - p.g_{10,i}\phi_{10,j} - p.g_{11,i}\phi_{11,j} - p.g_{12,i}\phi_{12,j} - p.g_{13,i}\phi_{13,j} - p.g_{14,i}\phi_{14,j} - p.g_{15,i}\phi_{15,j} + f_{0,i}m_{0,j} - f_{1,i}m_{1,j} - f_{2,i}m_{2,j} - f_{3,i}m_{3,j} - f_{4,i}m_{4,j} - f_{5,i}m5, j - f_{6,i}m_{6,j} - f_{7,i}m_{7,j} - f_{8,i}m_{8,j} - f_{9,i}m_{9,j} - f10, im_{10,j} - f_{11,i}m_{11,j} - f_{12,i}m_{12,j} - f_{13,i}m_{13,j} - f_{14,i}m_{14,j} - f_{15,i}m_{15,j})]$.

Putting the values of $Var[g_{i,l}\phi_{j,t}]$ and $Var[f_{i,l}m_{j,t}]$, we get
$Var[r_{0,l}] = \frac{1024p^2 d_g d_\phi}{N} + \frac{128d_f(p-1)(p+1)}{3}$.

In same manner, we have
$Var[r_{1,l}] = Var[r_{2,l}] = ... = Var[r_{15,l}] = \frac{1024p^2 d_g d_\phi}{N} + \frac{128d_f(p-1)(p+1)}{3}$.

It is required to compute the probability that $r_{i,l}$ lies between $\frac{-(q-1)}{2}$ to $\frac{q-1}{2}$ which would result in a successful decryption. Considering that $r_{i,l}$ have normal distribution having mean zero and the variance obtained as above, we have

$$\Pr\left(|r_{i,l}| \leq \frac{q-1}{2}\right) = \Pr\left(\frac{-q+1)}{2} \leq |r_{i,l}| \leq \frac{q-1}{2}\right) = 2\Phi\left(\frac{q-1}{2\sigma}\right) - 1,$$

where Φ denotes the distribution of the standard normal variable and

$$\sigma = \sqrt{\frac{1024p^2 d_g d_\phi}{N} + \frac{128 d_f (p-1)(p+1)}{3}}.$$

According to the above observations, the probability that STRU has successful decryption can be obtained from the two investigated points:

1. Each of the messages $m_0, m_1, ..., m_{15}$ to be correctly decrypted has the probability

$$\left(2\Phi\left(\frac{q-1}{2\sigma}\right) - 1\right)^N.$$

2. All of the messages $m_0, m_1, ..., m_{15}$ to be correctly decrypted has the probability

$$\left(2\Phi\left(\frac{q-1}{2\sigma}\right) - 1\right)^{16N}.$$

6 Cryptanalysis of STRU

6.1 Brute Force Attack

An attacker tries each possible sedenion polynomial in L_f to find a short key for decryption for mounting a brute force attack. The size of the search space L_f will be

$$|L_f| = \binom{N}{d_f}^{16} \binom{N-d_f}{d_f-1}^{16}, \text{ where } \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Similar to NTRU, F along with all its rotations $y^i F$ can be taken as the decryption key. Hence, search space to find the key is $\frac{|L_f|}{N}$. Similarly, if an attacker wants to find the plaintext directly, then he has to search on $\frac{|L_g|}{N}$ possibilities where

$$|L_g| = \binom{N}{d_g}^{16} \binom{N-d_g}{d_g}^{16}.$$

6.2 Meet-in-the-Middle (MITM) Attacks

If sufficient memory is available, then for attacking with MITM attack [5], an attacker requires search spaces $\sqrt{\frac{|L_f|}{N}}$ and $\sqrt{\frac{|L_g|}{N}}$ for finding the key and plaintext, respectively, i.e., meet-in-the-middle attack shortens the search space by the order of the square root.

6.3 Message Expansion Scheme

For obtaining encryption speed, there is an important factor, namely, message expansion, which cannot be ignored. Message expansion is the ratio of the sizes of ciphertext search space and plaintext search space and given as $\frac{\log|C|}{\log|P|}$ where P is the plaintext space and C is the ciphertext search space. In STRU cryptosystem:

$$\text{messageexpansion} = \frac{\log|C|}{\log|P|} = \frac{\log q^{16N}}{\log p^{16N}} = \frac{\log q}{\log p}.$$

As this ratio involves with q and p, q should be chosen in such a manner that will give the smaller decryption failure.

6.4 Lattice Attacks

Lattice attacks [4] are prevalent in lattice-based cryptographic protocols. The key idea is to find an element having a small norm, satisfying the relations as per the protocol to construct a lattice, and then use lattice reduction schemes to find the secret key. In STRU, finding a sedenion polynomial having a small norm that satisfies the relation $F \circ H = G \pmod q$ is difficult. Also, it is exigent for attacker to build a sedenionic lattice on which lattice reduction schemes can be applied to render the protocol insecure. The sole strategy for applying the lattice attack on the STRU cryptosystem and to find a suitable decryption key is to analyze the relation $F \circ H = G \pmod q$ in the following way:

$$f_0h_0 - f_1h_1 - f_2h_2 - f_3h_3 - f_4h_4 - f_5h_5 - f_6h_6 - f_7h_7 - f_8h_8 - f_9h_9 - f_{10}h_{10} - f_{11}h_{11} - f_{12}h_{12} - f_{13}h_{13} - f_{14}h_{14} - f_{15}h_{15} = g_0 + qu_0,$$

$$f_0h_1 + f_1h_0 + f_2h_3 - f_3h_2 + f_4h_5 - f_5h_4 - f_6h_7 + f_7h_6 + f_8h_9 - f_9h_8 - f_{10}h_{11} + f_{11}h_{10} - f_{12}g_{13} + f_{13}h_{12} + f_{14}h_{15} - f_{15}g_{14} == g_1 + qu_1,$$

.
.

.

$f_0h_{15} + f_1h_{14} - f_2h_{13} - f_3h_{12} + f_4h_{11} + f_5h_{10} - f_6h_9 + f_7h_8 - f_8h_7 + f_9h_6 - f_{10}h_5 - f_{11}h_4 + f_{12}h_3 + f_{13}h_2 - f_{14}h_1 + f_{15}h_0 = g_{15} + qu_{15}$.

Let $M_{N\times N}$ be the linear representation of the polynomials $h_0, h_1, ..., h_{15}$:

$$M_{N\times N} = \begin{bmatrix} h_{i,0} & h_{i,1} & h_{i,2} & ... & h_{i,N-1} \\ h_{i,N-1} & h_{i,0} & h_{i,1} & ... & h_{i,N-2} \\ h_{i,N-2} & h_{i,N-1} & h_{i,0} & ... & h_{i,N-3} \\ ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... \\ h_{i,2} & h_{i,3} & h_{i,4} & ... & h_{i,1} \\ h_{i,1} & h_{i,2} & h_{i,3} & ... & h_{i,0} \end{bmatrix}$$

.

We can construct a $32N$-dimensional STRU lattice (L_{STRU}) generated by the rows of the matrix M defined above. It can be observed from the above equations that the vector $\langle f_0, f_1, ..., f_{15}, g_0, g_1, ..., g_{15}\rangle_{1\times 32N}$ is contained in STRU lattice. In this lattice, a short vector may be used as our key as we do in the NTRU lattice (L_{NTRU}) [7]. For the STRU lattice, we have

1. $\text{Det}(L_{STRU}) = q^{16N}$.
2. $\|\langle f_0, f_1, ..., f_{15}, g_0, g_1, ..., g_{15}\rangle\| = \sqrt{64d} \approx 4.62378\sqrt{N}$ (we are assuming $d_f = d_g = d_\phi = d = N/3$).
3. The expected length of the shortest non-zero vector (by Gaussian heuristic) in the L_{STRU} is

$$\lambda_0 = \sqrt{\frac{n}{2\pi e}} Det(L)^{1/n} = \sqrt{\frac{32N}{2\pi e}} Det(q^{16N})^{1/32N} = \sqrt{\frac{16Nq}{\pi e}} \approx 1.368752\sqrt{Nq}.$$

4. $c = \frac{\|f_0, f_1, ..., f_{15}, g_0, g_1, ..., g_{15}\|}{\lambda_0} = \frac{4.62378\sqrt{N}}{1.368752\sqrt{Nq}} \equiv \frac{3.3780}{\sqrt{q}}$.

The target vectors in L_{STRU} are about $O(\sqrt{q})$ shorter than the expected shortest vector given by Gaussian heuristic. We can say that the STRU lattice has the same structure as NTRU lattice with the only difference that STRU lattice is not fully circular. When we choose the same parameter N in both the cryptosystems NTRU and STRU, the dimension of L_{STRU} is 16 times than L_{NTRU}. Hence, STRU lattice possesses all the properties of NTRU lattice.

All advantages of taking non-associative algebra [16] are same here as in OTRU cryptosystem [12].

7 Comparative Analysis

If we compare the speed of encryption and decryption processes of the STRU cryptosystem with the NTRU cryptosystem with equal dimensions, these are almost 16 times and 32 times slower, respectively. If we reduce N with the power of two, then it will affect the computation speed, given that the complexity of the convolution multiplication is $O(N^2)$. Consequently, NTRU with a size of $16N$ is almost 256 times slower than an NTRU with a dimension of N and also, naturally, much slower than an STRU. Therefore, we argue that higher security can be achieved by reducing N within reasonable limit, but then one can meet a claim about reducing the STRU speed. It could also be argued that the length of the parameter q in STRU is longer and should not be prime at an insignificant cost. Our proposed scheme, STRU cryptosystem, is efficient, fast, and cost-effective as the NTRU public key cryptosystem because of the nature of its underlying algebraic structure (having basic operations).

8 Conclusion

A public key cryptosystem, STRU, based on sedenion algebra (non-associative and non-alternative) containing all strengths and strong points of NTRU cryptosystem is introduced. It encrypts 16 data vectors at each encryption round. A new property "inverse associative property," which is required for successful decryption is also introduced. To attack STRU cryptosystem with the lattice threats is a very massive task than NTRU. The speed of STRU cryptosystem can be increased even higher than that of NTRU by reducing the size of the underlying convolution polynomial ring.

References

1. Feynman RP (1982) Simulating physics with computers. Int J Theor Phys 21(6/7)
2. Online Resource (2019). https://www.research.ibm.com/ibm-q/
3. Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Goldwasser S (ed) Proceedings of the 35th annual symposium on foundations of computer science. IEEE Computer Society Press
4. Online Resource (2019). https://quantumcomputingreport.com/news/nist-to-announce-round-2-pqc-candidates-on-january-10-2019/
5. Alkim E, Ducas L, Poppelmann T, Schwabe P (2016) Post-quantum key exchange—A new hope. In: USENIX security symposium
6. Carmody K (1988) Circular and hyperbolic quaternions, octonions, and sedenions. Appl Math Comput 28(1):47–72
7. Tian Y (2000) Similarity and consimilarity over real cayley dickson algebra. arxiv: math-ph/0003031
8. Wonenburger MJ, Schafer RD (1969) An introduction to nonassociative algebras. Bull Am Math Soc 75(4):7–12

9. Kutyłowski M (2004) Anonymity and rapid mixing in cryptographic protocols. In: The 4th central European conference on cryptology, Wartacrypt
10. Malekian E, Zakerolhosseini A (2010) A non-associative lattice-based public key cryptosystem. Secur Commun Netw 5:145–163
11. Malekian E, Zakerolhosseini A, Mashatan A (2011) QTRU: quaternionic version of the NTRU public-key cryptosystems. ISC Int J Inf Secur 3(1):29–42
12. Kouzmenko R (2006) Generalizations of the NTRU cryptosystem
13. Hoffstein J, Pipher J, Silverman JH (1998) NTRU: A ring based public key cryptosystem. In: Proceedings of the ANTS, LNCS, vol 1423. Springer, pp 267–288
14. Bagheri K, Sadeghi MR. A new non-associative cryptosystem based on NTRU public key cryptosystem and octonions algebra. ACM Commun Comput Algebra 49(1)
15. Baez JC (2002) The octonions. Bull Am Math Soci 39:145–205
16. Imaeda K, Imaeda M (2000) Sedenions: algebra and analysis. Appl Math Comput 115:77–88
17. Schafer RD (1996) An introduction to non associative algebras. Dover Publications Inc., NewYork, corrected reprint of the 1966 original
18. Thakur K, Tripathi BP (2017) STRU: a non alternative and multidimensional public key cryptosystem. Glob J Pure Appl Math 13(5):1447–1464
19. Hoffstein J, Silverman J (2000) Optimizations for NTRU. In: Public key cryptography and computational number theory, pp 11–15
20. Graham NH, Silverman NH, Whyte W (2003) A meet-in-the-middle attack on an NTRU private key, NTRU Technical Report-004
21. Coppersmith D, Shamir A (1997) Lattice attacks on NTRU. In: proceeding of EUROCRYPT 1997, vol 1233. LNCS, Springer, pp 52-61