Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

Research Article

No Sum (NS) Sequence: A Tool for Quantum-Safe Cryptography

Bharat S. Rawal[1], Anjan Biswas[2,3,4*], Raman Singh[5]

[1] Department of Computer Science and Digital Technologies, Grambling State University, Grambling, LA 71245-2715, USA
[2] Department of Mathematics and Physics, Grambling State University, Grambling, LA 71245-2715, USA
[3] Department of Applied Sciences, Cross-Border Faculty of Humanities, Economics and Engineering, Dunarea de Jos University of Galati, 111 Domneasca Street, Galati-800201, Romania
[4] Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Medunsa-0204, Pretoria, South Africa
[5] School of Computer Science, University of the West of Scotland, Scotland
E-mail: biswasa@gram.edu

Received: 15 April 2024; **Revised:** 31 July 2024; **Accepted:** 15 August 2024

Abstract: Quantum computing is emerging as a promising technology that can solve the world's most complex and computationally intensive problems, including cryptography. Conventional cryptography is now facing a severe threat because of the speed of computation offered by quantum computers. There is a need for the standardization of novel quantum-resistant public-key cryptographic algorithms. In this article, the No-Sum (NS) sequence-based public key cryptosystem is proposed, increasing the computational resource requirement against the quantum computer-assisted brute-force attack. In this paper, we demonstrate an elevation of security of the public key cryptosystem with NS sequence. The proposed methodology works in two phases; in the first phase, the conventional rivest−shamir−adleman (RSA) algorithm is used to share the first element and offset parameter between the receiver and sender. In the second phase, prime numbers are generated from the NS sequence and used for encryption/ decryption of messages/ciphers. This paper illustrates the potential of the NS sequence in developing a quantum-resistant cryptosystem, contributing to the advancement of secure communication in the quantum computing era.

Keywords: cryptography, encryption technique, no-sum sequence, post-quantum cryptography, special sequences

MSC: 81P94

1. Introduction

Unique sequence numbers play a crucial role in public-key cryptography applications, majorly because they can introduce freshness. Researchers emphasize novel encryption algorithm development and its implementation but generally lose focus on its perhaps most important ingredient, i.e., key generation. The various arithmetic concepts, like number systems, groups, rings, fields, etc., are the basic building blocks of cryptography [1]. Researchers in public-key cryptography have used unique sequences like Prime, Fibonacci, and Lucas numbers. In ideal conditions, the unique sequence numbers help keep the encryption algorithm's output nearly random and ensure secrecy intact. Substantial prime numbers have been used in public-key cryptography to make encryption/decryption key generation mechanisms secure.

The most appropriate examples where prime numbers are used for this are RSA algorithms, Taher and ElGamal improved algorithm, and elliptical curve cryptography applications [2–4]. The Fibonacci numbers have proved their usefulness in many fields, including cryptography. It is quite impressive to use Fibonacci sequences to produce encoding and decoding algorithms to secure message communication. The complexity of the encoded messages can be increased by generating sequential Fibonacci numbers [5]. The third-order Fibonacci series greatly enhances the cryptography characteristic as the Fibonacci universal code variation increases [6].

Lucas sequences help in improving the efficiency of the cryptosystem. If Lucas sequences are computed many times, it enhances the secrecy in real-time applications [7]. Lucas Fibonacci sequences are more efficient than regular binary coding if applied to the public key cryptosystem. These unique sequences have proved their ability to secure key distribution functionality [8].

NS sequence is one in which no element can be described as the sum of any subset in a given set. Each of its items cannot be equal to the total of any combinations of the other non-repeated elements in the sequence, regardless of whether any of them is positive [9]. Equation (1) represents a mathematical expression of NS sequence. There exist three types of NS sequences; however, we limit our analysis in this study to our own NS(+) sequence and just take into account addition operators. The NS (+-) sequence takes addition and subtraction into account. and the NS (+-*) sequence, considering addition, subtraction, and multiplication operations [10].

The NS sequence can be constructed either manually or using predefined algorithms. Manually, a set of NS sequences can be constructed by adding a new number that is not a sum of earlier existing numbers. This manual method is ponderous, and that is why it supports alternative algorithms to generate the NS number of large sets. It is found that some of the NS'S sets are periodic, while there is proof of aperiodic sets as well [9, 11]. The NS sequence properties make these sequences eligible candidates to be used in cryptography. This paper examines the properties of various unique sequences like Prime, Fibonacci, Lucas, and NS sequence numbers. The applicability of NS sequences is in detail analyzed and discussed based on some parameters like time etc.

Our contribution: 1) We introduce the No-Sum (NS) sequence. If any element cannot be represented as a sum of any subset in a given set, this sequence can be termed an NS sequence. 2) We introduce the superior prime numbers (SPN): All prime numbers in the NS sequence. 3) We introduced the RLP cryptosystem based on the NS sequence.

With the introduction of quantum computers, the risk of breaking public-key cryptography is increasing. The paper presents an NS sequence-based cryptosystem that is resistant to quantum computer-assisted brute-force attacks. The results of the analysis show that the proposed methodology provides promising results. This paper proposed an NS sequence-based cryptosystem, increasing the quantum computer-assisted brute-force attack's computational resource requirement. The one important NS sequence feature is to generate a reasonably separate set of new numbers if the starting element of the sequence is changed. Experiments show that NS sequence can efficiently select superior prime numbers with different cardinality, enhancing the proposed system's security and effectiveness. Many researchers seem to think that quantum computers will break public-key cryptosystems by performing an exhaustive search. It is not entirely true. Existing cryptosystems are insecure against quantum computers because Shor invented an algorithm that can solve the integer factorization problem (and the discrete log problem) in polynomial time. The effect of quantum computers on symmetric cryptosystems is slashing the time by one-half.

As quantum threats emerge, Post-Quantum Cryptography (PQC) shows promise in terms of efficiency and efficacy. In contrast to standard encryption schemes that are vulnerable to quantum attacks, PQC computations provide robustness without sacrificing computing performance. PQC achieves a compromise between higher security and functional productivity by using numerical problems, such as cross section-based or hash-based techniques. PQC's execution demonstrates ruthless timing, making it a sensible choice for gathering data under various figuring scenarios. PQC addresses the urgent need for security while keeping an eye on improving execution in the rapidly evolving field of digital communication [12].

Within the field of cryptography, the use of quantum computing platforms poses a serious challenge to the security of traditional public-key cryptographic algorithms such as Elliptic Curve Cryptography (ECC) and RSA. These methods rely on how hard it is to compute discrete logarithms or manage large numbers-problems that quantum computers are good at solving. In response to this impending threat, PQC has emerged with the goal of developing new cryptographic computations that are impervious to quantum attacks [13].

The normalization and reception of Post-Quantum Cryptography (PQC) addresses a basic stage in strengthening computerized protection from arising quantum dangers. As cryptographic frameworks face the approaching gamble of quantum assaults, worldwide endeavors are in progress to lay out normalized PQC calculations [14]. There are chances to investigate novel mathematical ideas and notions in post-quantum cryptography. Deep knowledge of mathematical concepts and their use in cryptography is necessary to design encryption algorithms that withstand attacks from quantum computers. As a result, new mathematical ideas and methods have been discovered that have applications in computer science and mathematics beyond PQC [15]. The study of lattice-based cryptography is one such field. A subclass of PQC referred to as "lattice-based cryptography" is predicated on how challenging certain algebraic problems using lattices are geometric structures called lattices can be utilized to build cryptographic algorithms that fend off attacks from quantum computing systems. Analyzing lattices and their characteristics [16].

Similar issues emerge as PQC looks to supplant traditional cryptographic strategies, requiring a sensitive harmony between current security goals and the limitations of the inheritance framework [16]. Overcoming this issue requires vital preparation, retrofitting, or, at times, a progressive change to additional coordinated frameworks. Effectively tending to heritage framework similarity is critical in guaranteeing the everywhere reception and adequacy of PQC in different mechanical scenes [17].

The rest of this research is ordered as follows. Section 2 describes the properties of unique sequences like Prime, Fibonacci, and Lucas numbers. Section 3 specifies the No-Sum (NS) sequences and examines their properties along with their usefulness. Section 4 discusses the quantum attack resistant cryptography approaches proposed by researchers and their various modalities. Section 5 presents a post quantum crypto system recommended by NIST. The proposed RLP cryptosystem is described in Section 6. The results and discussion about the proposed work are given in section 7, and the conclusion about this work, along with future works, is detailed in section 8.

2. Properties of some special sequences

2.1 *Prime numbers*

An example of a prime number is bigger than one and is not the result of multiplying two lesser natural numbers. ("Solved A prime number is a natural number greater than 1") Figure 1 shows the expansion characteristics of prime numbers. As the sequence of a prime number (at the x-axis) increases, then its value increases linearly. The first number considered is 3, and it is evident that the number linearly increases to 227 at the 50th number. With computing power readily available, big prime numbers can be calculated without much effort [18]. The public key infrastructure works on a large prime number, factoring complexity, and promising quantum computing proves its capability by breaking prime numbers-based cryptosystems. Shor's algorithm is capable of breaking RSA, Diffie-Hellman, and Elliptical curve-based encryption methodology. In response to this quantum computing capability, security agencies are moving their focus to quantum-resistant cryptography [19].

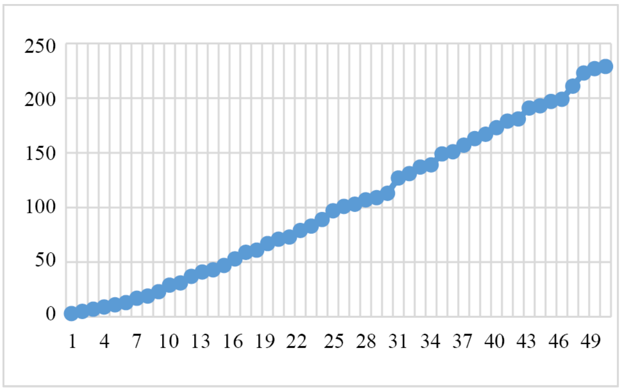

Figure 1. First 50 Prime numbers starting at 3

2.2 *Fibonacci numbers*

The Fibonacci sequence can be obtained by generating each next number by adding two preceding ones. The Fibonacci sequences provide many ways to generate and detect encryption and decryption keys because each sequence number has a unique representation. Since the same encrypted ciphertext cannot be generated for the different sequence numbers, it is hard to break [20]. Figure 2 shows that as the sequence number (at the x-axis) increases, the Fibonacci number value initially increases slowly, but the growth becomes exponential as the sequence passes 50. This suggests that as Fibonacci sequences' degree increases, computing for adding numbers increases marginally, but since the last two numbers are added, it remains a computationally solvable problem. Due to this particular sequence's recursive property, quantum computing is integrated to generate new keys to make communication secure. The Fibonacci sequence possesses a unique mathematical property that can be used in matrix coding of encryption/decryption scheme lai2017fast.

2.3 *Lucas numbers*

The Lucas numbers are similar to Fibonacci numbers but with a different starting point. The first two Fibonacci sequences are 0 and 1, whereas Lucas's numbers start with 2 and 1 and then keep adding the last two numbers to generate newer ones. Lucas number value's initial growth is similar to Fibonacci sequences, which is slow as sequence increase, as shown in Figure 3, but the growth increases faster than Fibonacci sequences. After sequence number 12, Lucas's growth number increases exponentially, making it the right candidate for cryptography. Lucas number coding is used in a high-capacity key distribution scheme that is based on quantum computing. The Lucas numbers have a close relationship with the Chebyshev maps, and it is found that Fibonacci numbers can easily replace them in terms of providing security. Only a few Lucas numbers can be picked to generate lower error rates but secure and long encryption keys to achieve the high-quality key distribution scheme. Lucas numbers are used as orbital angular momenta to produce entangled photons, and then the message is encoded with Chebyshev-map values. The quantum computing integration with the Lucas number makes this approach achieve a high-capacity key distribution mechanism [21].

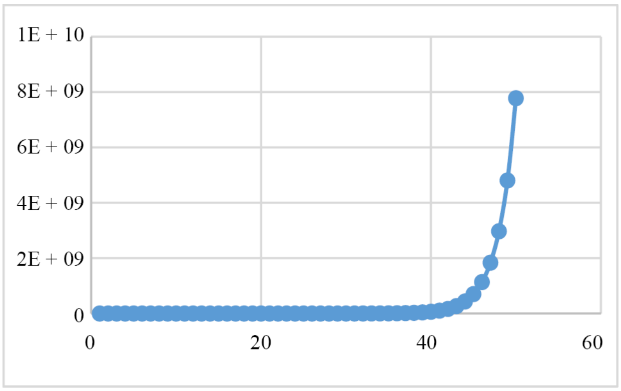

Figure 2. First 50 Fibonacci numbers starting at 4

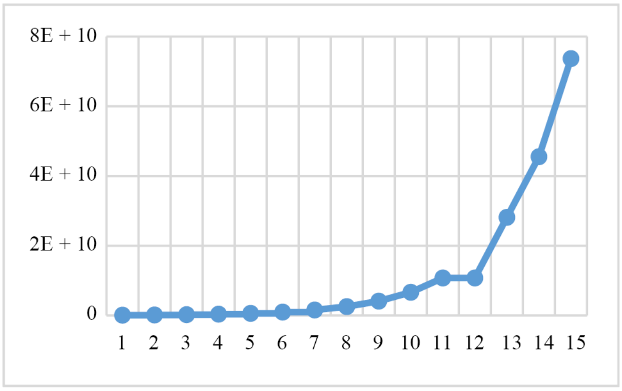

Figure 3. First 50 Lucas numbers starting at 4

3. No-sum sequence (NS): properties

3.1 *No-sum (NS) sequence*

We reproduce the NS sequence here [9] An NS sequence is one in which no element can be described as the sum of any subset in a given set. Each of its items cannot be equal to the total of any combinations of the other non-repeated elements in the sequence, regardless of whether any of them is positive. The NS sequence can be mathematically represented as given in equation (1).

$$\begin{aligned} &j,\ j+1,\ j+2,\ \ldots,\ 2*j,\ \mathrm{NV}_{1,\,\mathrm{j}},\ \mathrm{NV}(1,\ j+1,\ \mathrm{NV}_{1,\,\mathrm{j}}+2,\ \ldots,\ \mathrm{NV}_{1,\,\mathrm{j}} \\ &+j-2,\ \ldots,\ \mathrm{NV}_{i,\,\mathrm{j}},\ \mathrm{NV}_{i,\,\mathrm{j}}+1,\ \mathrm{NV}_{i,\,\mathrm{j}}+2,\ \ldots,\ \mathrm{NV}_{i,\,\mathrm{j}}+j-2,\ \ldots \end{aligned} \tag{1}$$

where, $j,\ j+1,\ j+2,\ \ldots,\ 2\ j]$ is the 1st set, $\mathrm{NV}_{1,\,\mathrm{j}},\ \mathrm{NV}_{1,\,\mathrm{j}}+1,\ \mathrm{NV}_{1,\,\mathrm{j}}+2,\ \ldots,\ \mathrm{NV}_{1,\,\mathrm{j}}+j-2$ is the 2nd set, and generally the ith set is $\left\{\mathrm{NV}_{i,\,\mathrm{j}},\ \mathrm{NV}_{i,\,\mathrm{j}}+1,\ \mathrm{NV}_{i,\,\mathrm{j}}+2,\ \ldots,\ \mathrm{NV}_{i,\,\mathrm{j}}+j-2\right\}$ Also, j is the value of the 1st element in this sequence and $j>2$. When $j=1,\ 2$ as the special cases, it is straightforward to form the sequence with the formulae below respectively. When $j=1$, the sequence will be in the form $1,\ 2,\ 4,\ 8,\ \ldots,\ 2i-1$ while when $j=2$, the sequence becomes $2,\ 3,\ 4,\ 8,\ \ldots,\ 2i-1$. For the general sequence with the first element over 2, based on the definition to form, the whole sequence is relatively computationally intensified. When adding one more element to the sequence, the computational time is dramatically increased nonlinearly. Fortunately, a set of formulae has been discovered for this sequence. Instead of direct calculation, accurate formulae are used to calculate each element in the sequence. The following variables and formulae are defined for this sequence.

Period: It is defined as the number of elements over which it leaps once. $Ps = j+1$ for the 1st set of elements, and otherwise $j-1$. It is observed that the period depends on the initial value j.

• Amplitude: Its amplitude measures how intensive it leaps, defined as given in equation (2).

$$Aj = (j-2)*(j-1)/2 \tag{2}$$

The amplitude depends on the initial value j of the 1st element. For example, if the first element is fixed as 3, then the amplitude will be computed as 1.

• First node: The value of the 1st node is obtained as per equation (3).

$$\mathrm{NV1},\ j = j*(j+2)+Aj \tag{3}$$

ith Node: The elements of the NS sequence are called nodes. In order to fetch the full sequence, nodes are computed. The following formula given in equation (4) is used to calculate the node values.

$$\mathrm{NV}i,\ j = \mathrm{NV1},\ j*ji-1+Aj*(ji-1-1)/(j-1) \tag{4}$$

where i is the index for ith node and $i > 0$, and j is the initial value of the 1st element in NS sequence and $j > 2$.

Node index i and offset k: When given the number of elements n, the following formulae can calculate the node index i and offset k in the sequence as given in equation (5).

$$i = [(n-3)/(j-1)] \tag{5}$$

where [] represents a floor function. For example, $[2.1] = 2$. The offset k can be calculated as per equation (6).

$$k = n-i(j-1)-3 \tag{6}$$

where k is the offset from the ith node, and $0 < k < j-1$.

Computation of the record: To compute any record whose index is n in the NS when given the initial value j of the 1st . element. From (2) and (6), any element can be computed via node value and offset below as given in equation (7).

$$Vn, j = \mathrm{NV}i, j+k \tag{7}$$

where n is the index of the nth element of the sequence. Once j is known, the above sequence equation (1) is known through straightforward calculation with (2) to (7).

3.2 *Proliferation property of NS sequence*

The one significant property of this sequence is its capability for exponential growth as the sequence increases. Due to this property, the first element of the NS sequence is 3, which is a one-digit number, whereas it grows to a 239-digit long number when the sequence reaches the 1,000th element. Figure 4 shows this exponential growth phenomenon. After sequence 45, the value increases exponentially. Unlike Fibonacci Sequence and Lucas numbers, which are computationally straightforward, the NS sequence requires significant computational efforts to calculate elements.

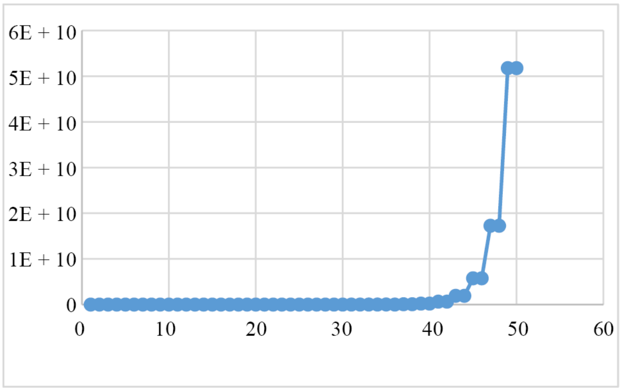

Figure 4. First 50 elements of NS sequence with 3

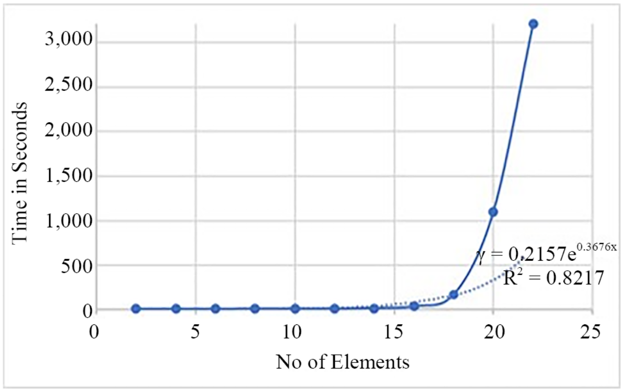

Figure 5. The time required to calculate several elements with brute force

The computationally intensive property of this particular sequence highlights the usefulness of the NS sequence in cryptography. For example, Figure 5 shows the time taken by a brute force attack to calculate any element in the NS set. As the sequence proceeds, it is evident that the time required to calculate element value by brute force attack increases exponentially.

Figure 6 demonstrates the time required to calculate the 1,000 elements from a starting point. For example, the x-axis depicts the starting element from where the 1,000 elements are computed. The y-axis shows the milliseconds required to calculate these ns-sequence numbers. It is evident that if the starting element is 10, the total time required to compute 1,000 elements is 13,800 milliseconds. If we increase the starting element to 100, now the time required is reduced to 3,000 milliseconds. It is also noticeable that the time required is not gradually decreasing as the starting point increases, but it is oscillating but decreasing in peak. This is a genuinely intriguing phenomenon that needs further research attention.

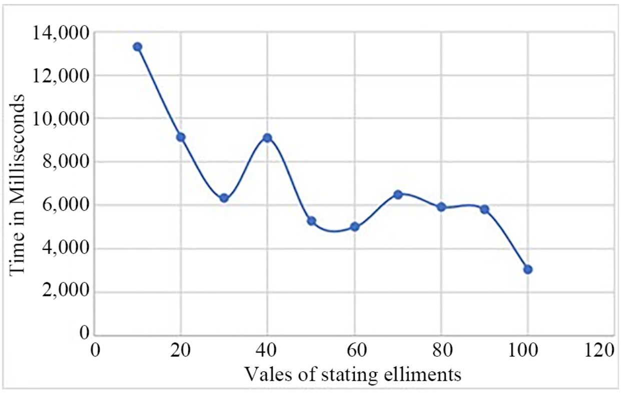

Figure 6. Time (in Millisecond) requires calculating 1,000

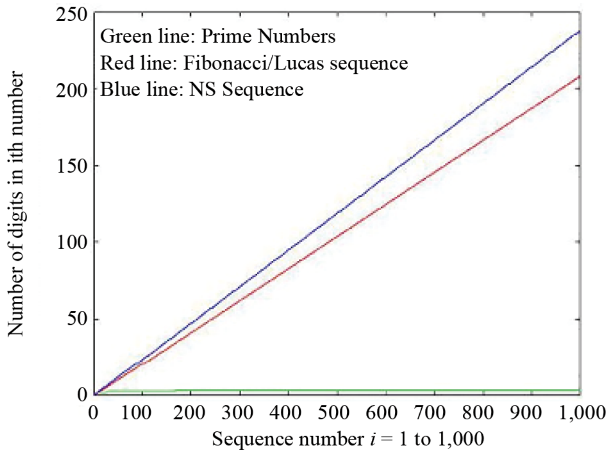

Figure 7. Growth in the number of digits as various sequence proceeds

There is one interesting property of these sequences: the growth of digits counts in the value of elements as the sequence progresses. This growth of digits provides an idea of how exponentially numbers increase in the sequences. Figure 7 compares the growth of digit counts for prime numbers, Fibonacci, Lucas, and NS sequences. The figure shows that prime numbers grow slowly, whereas the NS sequence grows fastest among these four studied sequences. Fibonacci and Lucas's sequences are at the same rate but lower than NS sequences but significantly higher than prime numbers. All four sequences start from single digit numbers and grow at different rates. For example, prime numbers start from a single digit and grow to a four-digit number at the 1,000th element. The growth is relatively slow as the prime number remains at four-digit values from the 169th element to the 1,000 th element. The growth of Fibonacci and Lucas's numbers is relatively higher than prime numbers. It starts from a single digit and goes up to 208 digits wide number at the 1,000th element, which is compelling higher than the growth of prime numbers. The growth of the NS sequence is higher than all of the other three sequences. It started from a single digit number to 238-digit vast number at the 1,000 th element. The growth seems to be linear but has a significant increment as it grows from a 119-digit wide number at the 500th element to a 238-digit vast number at the 1,000 th element.

3.3 *The convergence of NS sequence*

A convergence test is carried out to learn a property of any sequence whether it will converge, i.e., reach a finite value or diverge (not converged) as it is already proved that the NS sequence will start from a single digit number and increase towards a considerable number. For this convergence test, the first set of elements, i.e. (3, 4, 5, 6), will be ignored as it has finite numbers. The starting number j will also be fixed to 3 to test the convergence of a specific NS sequence used in this experiment. The value for the first node can be computed to 16 as per equation (3) if $j = 3$ and $Aj = 1$ (computed from equation (2). So, the equation (4) will be reduced to (8) for the convergence test:

$$NVi,\ 3 = 16 * 3i - 1 + 1 * (3i - 1 - 1)/(2) \tag{8}$$

Convergence can be defined as given in definition 1.

Definition 1 A sequence $a1,\ a2,\ a3$, converges to a limit L if:

$Limitlimi\infty NVi,\ 3 = L$. For any $E > 0$, there is a NEN such that $|aiL| <$, for all $i > N$.

Since it is evident that as the sequence progresses, the element's value increases exceptionally, the candidate for L can be taken infinity ( ). If we proceed towards a too high sequence number, i.e., the ith number's high value, the element's value will also be extremely high. Let us assume the ith is so large that it is replaced by, putting this value in equation (8) as shown in equation (9).

$$NV\infty,\ 3 = 16 * 3\infty - 1 + 1 * (3\infty - 1 - 1)/(2) \tag{9}$$

$$NV\infty,\ 3 = 16 * 3\infty + 1 * (3\infty)/(2) \tag{10}$$

$$NV\infty,\ 3 = 16 * \infty + (\infty)/(2) \tag{11}$$

$$NV\infty,\ 3 = 16 * \infty + \infty \tag{12}$$

$$NV\infty,\ 3 = \infty \tag{13}$$

If the equation is calculated using common mathematical expressions as shown in equations (10), (11), (12), and (13), it can be seen that the element value comes out to be ∞. This suggests that the given NS sequence is not convergence in nature and is likely to advance ∞.

3.4 *Unique properties of NS sequence*

NS sequences offer high security due to their combinatorial complexity, which makes them difficult to reverse-engineer or anticipate. They lack a sum structure, preventing attackers from using sum-based attacks. NS sequences are non-trivial to generate and often involve complicated algorithms, contributing to their unpredictability and strength in cryptographic applications. They also have random-like behavior, making them resistant to analytical attacks and statistical analysis. They can handle large key spaces, making brute-force attacks more difficult. NS sequences can be mathematically demonstrated, providing more confidence in their ability to withstand attacks. Their theoretical underpinnings have been thoroughly researched, providing a solid framework for their application in safe systems. They also have minimal correlation with other sequences, making them useful in multi-user systems.

4. Eventual cryptography: Quantum resistance

The advancement in futuristic technological development despairs the applicability of classical cryptography approaches. One such headway is adopting quantum computing to breach encryption algorithms. When Google's quantum computer executed the task in 200 seconds, it would have taken 10,000 years [22]; otherwise, it makes it clear that the future of traditional cryptography is not glorious. Researchers are already working and proposing new methods for transforming cryptography into quantum attack-resistant ones. The primary way to secure communication from a quantum attack is by securing key distribution mechanisms and using the quantum concept. Researchers started working on Quantum Key Distribution (QKD) protocol to improve security using single-photon polarization [23] long ago, but many research issues still need to be addressed. Authors have used Edholm's law of bandwidth along with Gaussian encoding and proposed a terahertz key distribution mechanism. They found that direct and reverse reconciliation is possible over short distances and can secure credit cards/cash machines or proximity transactions. Various concepts like the no-cloning theorem and the monogamy of entanglement make this short communication and the QKD technique a quantum attack-resistant mechanism [24]. Since wireless communication channels are also susceptible to quantum attacks, the researcher proposed a post-quantum cryptography and hash proof system (HPS), which focuses on quantum-resistant verification, to secure this sort of channel. These HPS structures often have a solid security foundation that focuses on single-bit encryption and can check the selected ciphertext attack (CCA). For wireless communications, this method can deliver excellent efficiency for authentication. Lattice-based cryptography makes a credible claim for post-quantum cryptography and its security can be predicated on worst-case hardness assumptions [9, 25].

It is a general perception that quantum computing may not be an excellent idea for small devices with limited computing and memory resources because of its processing-intensive characteristics. However, researchers are working on lightweight quantum.

Cryptography is like lattice-based cryptography for small devices like the Internet of Things (IoT). This type of cryptography ensures strong security guarantees using efficient worst-to-average case reduction along with small sizes of key and ciphertext [24]. Apart from the key distribution, quantum computing is also used in many other cryptography paradigms like quantum unchangeability, leakage resilience, quantum cryptanalysis, post-quantum cryptography, etc. The no-cloning property is asserted in quantum computing because no physical process exists in a quantum system that takes a single input and produces two different outputs. Quantum cryptography practices conjugate coding, also known as quantum coding or quantum multiplexing. This makes quantum cryptography everlasting because calculating one basis will destroy the encoding information present in its conjugate basis. Secondly, it is possible to check the authenticity of the quantum encoding originator, and the third party cannot create a verification procedure for approvable two quantum states by accessing a single encoded state [26]. Researchers also created cryptosystems based on isogenies on elliptic curves, possibly the first hardware implementation of isogeny-based encryption. To make cryptography quantum-resistant, a supersingular isogeny Diffie-Hellman (SIDH) based key exchange mechanism is presented. It is proposed that, in contrast to conventional Elliptic Curve Cryptography (ECC), isogeny computations over supersingular elliptic curves are immune to quantum attacks. This method computationally generates an algebraic map across elliptic curves [27].

Shor's algorithm, which can accelerate the calculation of factoring, discrete logarithm problem (DLP), and elliptic-curve discrete logarithm (ECDLP) issues, is one of the most intriguing and pioneering quantum algorithms. This algorithm alters how people once perceived various cryptography algorithms, such as Transport Layer Security (TLS), Secure Shell (SSH), Internet Protocol Security (IPSec), or communication protocols based on Diffie-Hellman key, digital signatures, public-key encryption, etc., which were once thought to be impenetrable but are now exposed as a result of Shor's algorithm. This algorithm can identify periodicity in a function by solving the Abelian hidden subgroup problem. For instance, Shor's approach effectively determines if a function $f(x+)$ admits a period [28]. Shor's algorithm takes the time to factor the RSA public key is almost the same as legitimate users decrypt the ciphertext using its genuine key. This algorithm uses exponentiation modulo n and almost matches the time of factorization with the real decryption time. Researchers even proposed a new quantum-based factorization algorithm faster than Shor's algorithm [29].

Majorly, three parametric issues will keep the researchers occupied in cryptography: length of public keys, private key lifetime, and computational cost. It is known that the security provided by any encryption algorithm depends on the length of the key. The longer the key, the more security is because a longer key takes more time to compute. Blowfish is considered the most secure algorithm because of its vital length of 32 bits to 448 bits. The other traditional cryptography algorithm has different vital lengths, like the DES algorithm's key length is 56 bits, whereas AES key length can be anything 128, 192, or 256 bits. The longer key length folds increase the computational resources and time required to detect it, but attacks generated from quantum computers may not significantly affect the target algorithm's key length. The brute force attack carried by quantum computers will also not have any apprehensions about the computing time required to compute any length key [30].

5. Post quantum crypto system recommended by NIST

Cryptography existed even before computers were invented. Cryptography refers to a range of techniques to secure information at rest or on flight. It also applies to symmetric keys, which are those that are used to both encrypt and decode communications. Public/private key pairs, sometimes known as asymmetric keys, have been described. Public/private key pairs can take the form of secret/public key pairs, in which the private key is kept secret, and the public key can be known to all parties with whom they have a chance to communicate and exchange information [31]. There is a jungle of private and public key cryptography. However, we will directly jump to the most popular and widely used public or private cryptography technics recommended by NIST [32]. According to NIST, recommend FIPS-approved, or NIST cryptographic algorithms [32]. These cryptographic methods must go through extensive security testing and analysis to achieve acceptable security. Using larger keys is frequently an option when better security is required.

Recently NIST has selected four cryptosystems CRYSTALS-Kyber, Dilithium, FALCON, and SPHINCS+. CRYSTA LS-Kyber is for use in general encryption. It offers several benefits including two parties can exchange relatively modest encryption keys and maintain the required speed of operation. It is fairly faster among cross-platforms, and it is designed for efficient constant time implementation, and some optimized routines across all parameter sets.

CRYSTALS-Kyber: The advance of Kyber originated from the seminal LWE-based crypto scheme of Regev. It is designed to provide secure encryption operations resistant to attacks from quantum computers. Here is an overview of how CRYSTALS-Kyber works:

CRYSTALS-Kyber uses a set of parameters to define the security level and efficiency of the scheme. The parameters include the dimension of the underlying lattice, the number of rounds for the encryption algorithm, and the number of bits used for public keys, secret keys, and ciphertexts. The specific values of these parameters can be chosen based on the desired security level.

For a more detailed understanding, you can refer to the specific research papers or documentation related to CRYSTALS-Kyber.

Crystle-Kyber $= [A;\ \text{Public Key}][s;\ \text{secrete key}] + [e:\ (\alpha,\ \beta,\ \theta)\text{small error terms}] = [t;\ \text{public key}]$

$V = t\alpha,\ +\beta + m$

$U = A\alpha + \theta$

$D = V - sU$

To decrypt we will remove the public key $(A,\ t)$

$d = V - sU = t\alpha,\ +\beta + m - s(A\alpha + \theta)$; we have $As + e = t$.

$d = (As + e)\alpha + \beta + m - sA\alpha - s\theta$

$As\alpha + e\alpha + \beta + m - sA\alpha - s\theta = e\alpha + \beta + m - s\theta.$

We can remove Noise (term with smaller coefficient) $= m + e\alpha + \beta - s\theta$ (everything underline are small).

6. Proposed RLP cryptosystem

The proposed cryptography method used the NS sequence to secure the public key cryptosystem. The proposed system uses the RSA algorithm to encrypt/decrypt but integrates the NS sequence numbers to enhance security. Figure 8 shows the distinct phases involved in the proposed cryptosystem. For the proposed system, the entities are named as sender and receiver. The sender is the entity that wants to send secret messages like emails, auction bidding, passwords, etc., to the other entity called the receiver. The security system is divided into two phases. In the first phase, the conventional RSA algorithm is used to share the first NS-sequence element (commonly known as N0 or *j*) by the sender to the receiver. Along with the first element, the sender also shares another variable offset. The receiver should receive these two values secretly, and hence the RSA cryptosystem is used. The receiver generates two prime numbers (say p1, q1) and initializes to get public and private key pairs. The receiver publishes its vital public pairs (say e1, n1) to all other entities, including the sender. The initialization process is explained in algorithm 1.

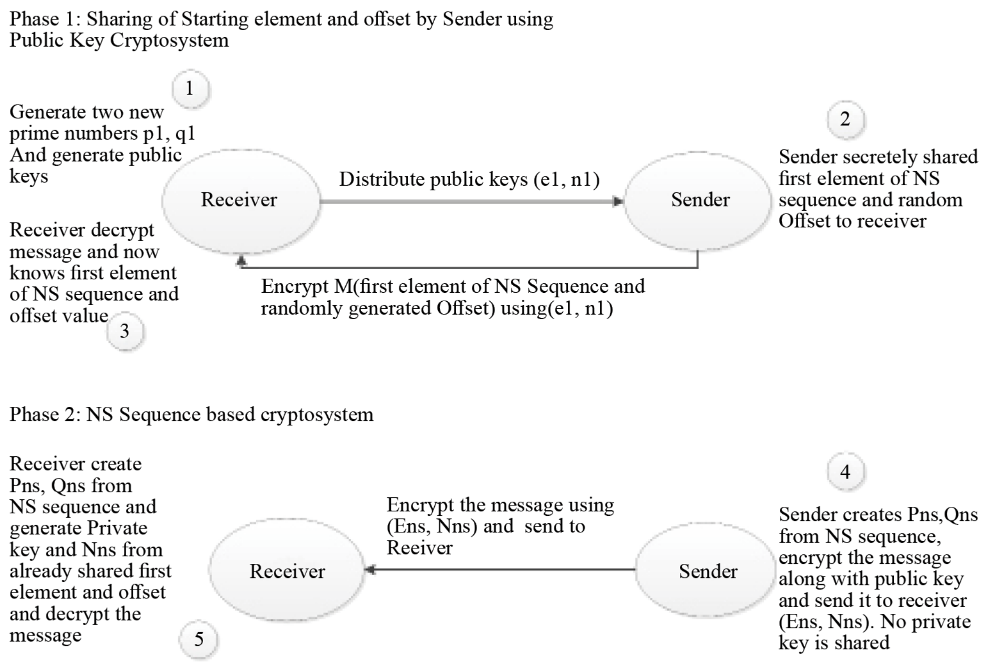
Phase 1: Sharing of Starting element and offset by Sender using Public Key Cryptosystem
1
Generate two new prime numbers p1, q1 And generate public keys
Receiver
Distribute public keys (e1, n1)
Sender
2
Sender secretely shared first element of NS sequence and random Offset to receiver
Receiver decrypt message and now knows first element of NS sequence and offset value
3
Encrypt M(first element of NS Sequence and randomly generated Offset) using(e1, n1)
Phase 2: NS Sequence based cryptosystem
Receiver create Pns, Qns from NS sequence and generate Private key and Nns from already shared first element and offset and decrypt the message
Receiver
Encrypt the message using (Ens, Nns) and send to Reeiver
Sender
4
Sender creates Pns,Qns from NS sequence, encrypt the message along with public key and send it to receiver (Ens, Nns). No private key is shared
5

Figure 8. proposed NS Sequence-based cryptosystem procedural flow chart

```
Algorithm 1 Initialization in RSA/NS Sequence-based Cryptosystem: initialization $(p, q)$ reproduced from [9].
Result: $Pk, Phi, d, e$
Input: Two prime numbers $p, q$ Start:
$Pk = p\ \ q; //N = p * q$
$Phi = (p\ \ 1)(q\ \ 1)$
$x = 2;\ e = 1$
while $x > 1$ do
$e = e + 1;$ //The public key $e$
$x = gcd(Phi, e)$
end
$i = 1,\ r = 1$
while $r > 0$ do
$k = (Phi\ i) + 1 r = rem(k, e)$
$i = i + 1$
end
$d = k/e;$ //The private key $e$
End
```

The sender randomly generates the first element (say j) of the NS-sequence and offset value. The sender uses algorithm 2 to generate the NS-sequence up to a given number (say ele). This algorithm can control the total number of NS-sequence elements, and users can use it to control the complexity of the overall cryptosystem. The randomly generated value offset will generate another pair of Superior prime numbers (SPN), which will be further used in the proposed system. The sender now encrypts the first element and offsets using a public key shared by the receiver and sends the ciphertext to the receiver. The receiver will decrypt the receiver ciphertext using its private key and get the first element and offset. After this first phase, both the sender and receiver know the NS sequence's first element and the offset value.

Each element is unique in a given NS sequence and cannot be regenerated using the additive property. Prime numbers only ever have two components: one and themselves, as opposed to composite numbers, which can have several factors. Number 1 is considered special, which is neither prime nor composite. SPN is derived from NS Sequence such its prime numbers are designated as SPN. Hence SPN numbers are present both in the NS sequence set as well as prime numbers set. SPN may exhibit properties like indivisibility and non-multiplicative, and it is not advisable to use the addition/ subtraction of two SPNs to get a new SPN. A natural number that is the sum of two prime numbers is referred to as a semiprime number [33]. The two prime numbers in the product may equal each other so that the semiprimes may include prime number squares.

```
Algorithm 2 NS Sequence generation from a given starting element: generate_nsseq ($j$, ele)
Result: NS Sequence NS_seq($N*$)
Input: starting element $j$ number of elements required in sequence ele.
Start: $NS\_seq = []$
$num\_ele = round(ele/(j1))$   $NS\_seq = [NS\_seq; j]$   $period\_first = j+1$
$period = j\ \ 1$
$t = j$
for $ab = 2:\ period\_first$ do
$t = t+1$
$NS\_seq = [NS\_seq;\ t]$
end
$A_j = (j-*2)(j-1)/2$
$NV_{1,\ j} = j(j*+2)+A_j$
for $i = 1:(num\_ele)$ do
$q = i-1\quad q\quad q$
$temp = ((NV_{1,\ j}(j))+(((j)1)/2))$
$add1 = temp$
for $tt = 1:period$ do $NS\_seq = [NS\_seq;\ add1]add1 = add1+1$
end end
$NS\_seq = truncate(NS\_seq, ele)$
End
```

Phase 2 of the proposed cryptosystem uses the NS sequence and offset value to generate the prime numbers. In this phase, no public key is shared by the receiver with the sender, but the sender itself generates the receiver's public key. The sender selected two prime numbers from the NS sequence. The selection of these prime numbers can be varied, and newer algorithms can be proposed to enhance the system's security and complexity. For this experiment, the method used to generate two prime numbers uses an offset and set of NS sub-sequences. The NS sequence is generated in sub-sequences, and the prime numbers are taken from these sub-sequences as per the offset value given. For example, the first prime number (say pns) is selected from the first sub-sequence. The first prime number that occurred from the offset element is chosen as the first prime number. The second prime number (say qns) is selected as the prime number after the second sub-sequence offset element. This arrangement of choosing prime numbers is an open research area that needs more attention. The sender now uses these prime numbers (say pns, qns) to generate public key pairs and encrypt the messages using this public key pair. The receiver receives the ciphertext along with public key pairs. The sender and receiver can agree in advance about the one prime fundamental generation methodology like picking the first prime numbers to form the first sub-sequence starting from offset value. Both sender and receiver also know about the first element and offset value in advance. The receiver now generates the NS sequences using algorithm two and parameters like the first element (j). The receiver picks the first prime number from the NS sequence and calculates the second prime number from the first prime and public keys. The receiver can now calculate the private key and decrypt the ciphertext shared by the sender. In the proposed cryptosystem, the public and private key pairs are common for both the sender and receiver and change for each communication. This type of mechanism enhances the security of the proposed cryptosystem. The proposed cryptosystem is explained in algorithm 3.

```
Algorithm 3 NS Sequence based RLP Cryptosystem
Result: Message and Cipher text
Input: Prime numbers
Start:
Phase 1: Sharing starting element and off set using RSA cryptosystem.
Receiver Side
$[p1, q1] = generate\_prime(\ )$
$[Pk1, Phi1, d1, e1] = initialization(p1, q1]$
Share public key (Pk, e1) to sender side Sender Side $j = random\_number(\ )$; //
First element of NS Sequence $offset = random\_number(\ )$; //
Off set for creating final prime numbers later
$cipher\_init = RSA\_encrypt((j, offset), Pk, e1)$; //
Secretly share the first element $j$ and off set to the receiver side
The receiver received the encrypted first element $j$ and off set.
Receiver Side
$(j, offset) = RSA\_decrypt(cipher\_init, Pk, d)$
Phase 2: NS sequence based cryptosystem Sender Side: Encryption $ele = 1,000$; //
Number of required elements in NS sequence $NS\_Seq = generate\_nsseq(j, ele)$
$(pns, qns) = find\_prim(NS\_seq, offset)[Pk_{ns}, Phi_{ns}, DNS, e_{ns}] = initialization(pns, qns]$
$cipher\_ns = RSA\_encrypt((Message, Pk_{ns}, e_{ns})$
$Sharethecipher\_nstexttoreceiverwithpublickeys(Pk_{ns}, e_{ns})$
Receiver Side: Decryption
$NS\_Seq = generate\ SSE\ (j, ele)(pns) = find\_prim(NS\_seq, offset)qns = Pk_{ns}/pns$
$[Pk_{ns}, Phi_{ns}, DNS, e_{ns}] = initialization(pns, qns]$ $Message = RSA\_decrypt(cipher\_ns, Pk_{ns}, DNS)$
End
```

7. Results analysis

The proposed cryptosystem is implemented using Matlab, a multi-paradigm numerical computing environment, and proprietary programming language. The system used to implement the proposed cryptosystem consists of 64-bit Microsoft Windows 10 Home Edition with Intel Core i5, 2.5 GHz x64 based processor. The system is configured with 8 GB of DDR3 Random Access Memory (RAM) and 1,000 GB of a solid-state drive. The implementation is divided into two parts, the sender and receiver sides to understand the time taken by different procedures on both sides. Figure 9 shows the sender side's time if it uses a direct RSA algorithm and the proposed cryptosystem. It shows the comparative time analysis. The x-axis gives the starting element of the NS sequence, which starts at 6 and goes up to 1,500. NS-sequence is generated as per the starting element and constitutes different elements depending on the starting element. It means a new set of elements is generated in the NS sequence if the starting element is changed. For this experiment, the total number of elements generated in NS-sequence is taken as 1,000, but the flexibility is provided to users to increase it as per their requirements. The y-axis shows the time taken in seconds by the sender to encrypt the message completely. The red graph line shows the time taken by the sender if the proposed cryptosystem is used. Since the proposed system provides enhanced security and is executed in two phases, the time taken by it is also increased following the starting element of the NS sequence. The blue graph line shows the sender side's time if the classical RSA cryptosystem is used. For this experiment, the same prime numbers are used for both cryptosystems, but RSA is executed in one phase, whereas the proposed system is executed in two phases.

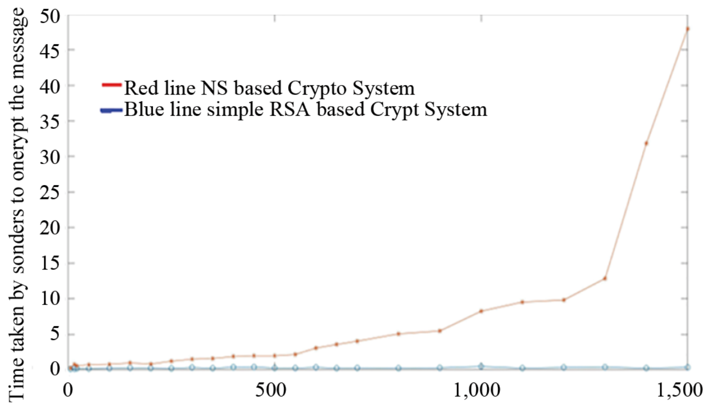

Figure 9. Time analysis of RLP and Plain RSA cryptosystem from the sender side

Since the low digit, prime number generation is easy, and elements do not change if the starting element is changed, the sender side's time is almost stable. If the starting element is taken low, the proposed system's time is comparable to the conventional RSA cryptosystem's time.

If the starting element is increasing, the time taken by the sender side is also significantly increasing because of two-phase systems and generation of NS-sequence each time, and then selecting SPN from the generated NS-sequence. The flexibility of new elements in NS-sequence and selection criteria of SPN from NS-sequence enhances the cryptosystem's security and makes it quantum attack resistant.

Figure 10 shows the comparative analysis of the time taken by the receiver side for both conventional RSA and the proposed cryptosystem. A similar pattern of time analysis is identified on the receiver side also. The time taken to decrypt the messages is almost stable for the conventional RSA system, whereas the time taken by the proposed cryptosystem is increasing following the starting element of the NS sequence. As the starting element of the NS sequence increases, larger prime numbers are generated, enhancing the system's security against brute-force attacks.

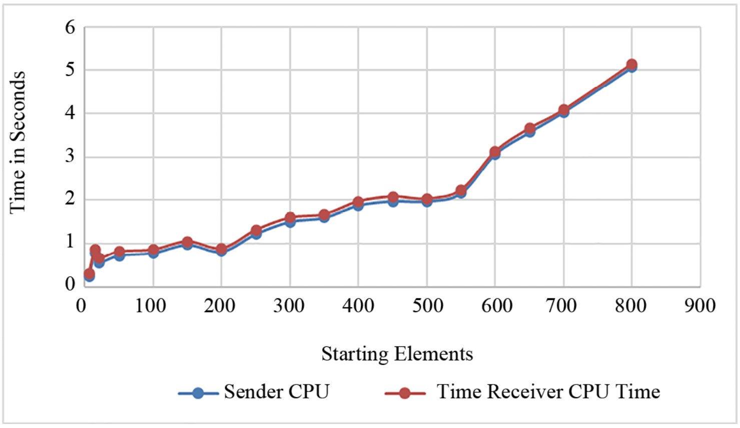

Figure 10. Time analysis of RLP and Plain RSA Cryptosystem from the receiver sid

Figure 11 shows the senders and receivers' CPU times. We can notice it gradually increases with higher starting elements in the NS sequence.

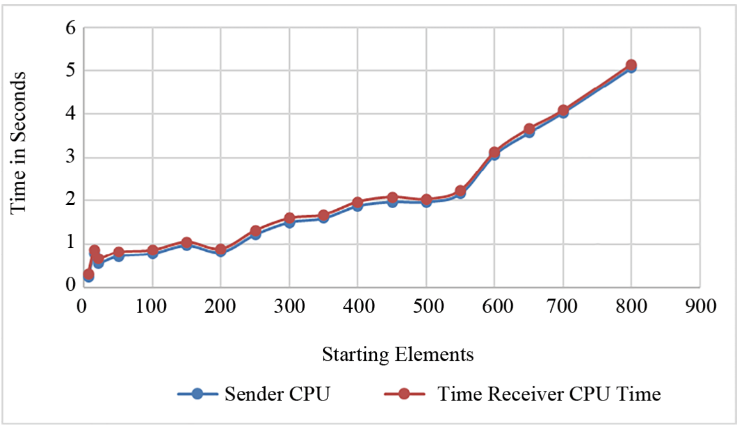

Figure 11. Sender and Receiver CPU times

Figure 12 shows the encryption and decryption times for RLP with respect to file size. We can notice that for smaller file size decryption time is always higher than encryption time. For large files start gradually smaller than encryption times.

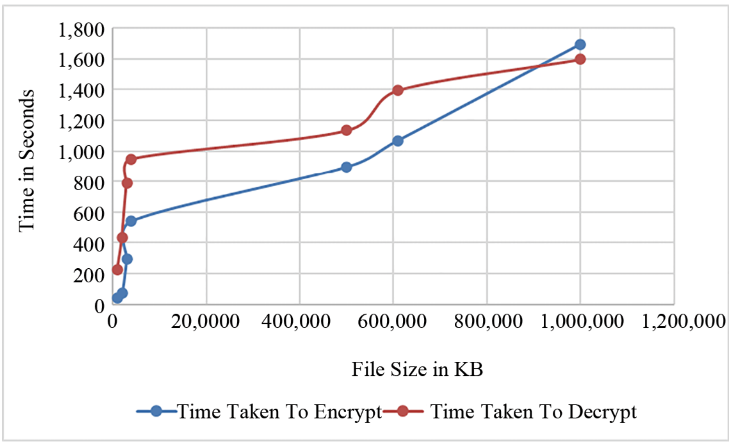

Figure 12. Encryption and decryption times for RLP

Figure 10 shows the comparative analysis of the time taken by the receiver side for both conventional RSA and the proposed cryptosystem. A similar pattern of time analysis is identified on the receiver side also. The time taken to decrypt the messages is almost stable for the conventional RSA system, whereas the time taken by the proposed cryptosystem is increasing following the starting element of the NS sequence. As the starting element of the NS sequence increases, larger prime numbers are generated, enhancing the system's security against brute-force attacks.

Table 1 shows the values of parameters and results from the analysis of the proposed cryptosystem. In this table, the starting element shows the various values taken as a starting element for generating different NS sequences in different experiments. The parameter in prime 1 and its prime 2 shows the prime numbers generated by the receiver in phase 1 to secretly share the starting element and offset by the sender and receiver.

Table 1. The results analysis of the proposed cryptosystem

Starting element	Init Prime 1 (p1)	Init Prime 2 (q1)	SPN 1 (pns)	SPN 2 (qns)	Offset	Sender cputime	Receiver cputime
6	43	17	7	61	3	0.25	0.3125
15	19	3	17	353	4	0.7813	0.8594
21	31	17	23	683	9	0.5625	0.6563
50	7	17	53	3,793	15	0.7188	0.8125
100	23	19	101	15,061	4	0.7813	0.8594
150	7	433	151	33,851	14	0.9688	1.0469
200	163	149	211	60,127	15	0.8125	0.8906
250	383	29	251	93,887	4	1.2188	1.3125
300	131	311	307	135,173	3	1.5	1.5938
350	347	31	353	183,943	10	1.5938	1.6719
400	211	373	411	240,257	11	1.875	1.9688
450	443	433	457	303,983	6	1.9688	2.0781
500	331	67	503	375,253	1	1.9688	2.0313
550	353	223	557	454,039	8	2.1563	2.2344
600	443	271	601	540,343	10	3.0625	3.125
650	421	383	653	634,091	9	3.5625	3.6563
700	61	83	701	735,367	14	4.0313	4.0781
800	5	211	809	960,419	3	5.0625	5.1406
900	487	331	907	1215,463	8	5.4688	5.5156
1,000	197	13	1,009	1,500,517	11	8.25	8.3281
1,100	11	17	1,103	1,815,559	8	9.5313	9.6406
1,200	29	389	1,201	2,160,617	13	9.8125	9.8906
1,300	347	47	1,301	2,535,671	10	12.8438	12.875
1,400	229	479	1,409	2,940,731	10	31.8438	31.9219
1,500	467	499	1,511	3,375,767	3	47.9688	48.1875

This is the first phase of security, and further research can be carried out to secure this phase further. SPN1 and SPN2 signify the superior prime numbers generated from the NS-sequence in phase 2, used to encrypt the sender's real message. Offset is another parameter that adds randomness and an extra layer of security in the proposed cryptosystem by picking SPN from the NS sequence. Sender and receiver CPU time is the actual CPU time taken by the sender and receiver side to encrypt and decrypt in both the proposed cryptosystem phases. The table shows that the SPN2 is becoming larger and larger as the starting element increases, whereas the p1 and q1 are still stable. The digit length of SPN 2 can even further be increased by selecting it from the higher sub-sequences, as for this experiment, it is chosen from the second sub-sequence only. The SPN1 is chosen among the first sub-sequence of the NS-sequence, and it explains the limited length of it.

By controlling the selection of both SPNs, the security of the proposed cryptosystem can be enhanced further. The time taken by the sender and receiver shows an increasing pattern as the starting element's size increases. From the table, it is evident that if the starting element and offset can increase the cryptosystem's complexity and provide significant security against a brute-force attack, the messages delivered are essential. If we further increase the starting element's size like 5,000, the cryptosystem selects the prime numbers 487 and 463 in the first phase, whereas the SPNs selected are 5,003 and 37,502,519 is somewhat a large prime number. The offset taken for this experiment is 8, and because of the large prime numbers' selection, the time taken by the sender and increases are 1,682.7 and 1,686.2, respectively.

8. Conclusions

Send her side analysis time comparison the time taken by the sender to encrypt messages was compared between the proposed cryptosystem and the classical RSA algorithm and sequence. The starting element of the sequence ranged from 6 to 1,500 generating 1,000 elements. The time taken increased with higher starting elements due to the two-phase execution, and sequence generation Graph analysis. The red line proposed crypt system showed increased time with higher starting elements while the blue line RSA remains stable, receiving side analysis. Time comparison is similar to the center side. The time taken to decrypt was analyzed. The proposed crypto system showed increasing time with higher starting elements while the RSA system remains stable , security enhancement, larger prime numbers generated with higher starting elements, enhance security against fruit force attack CPU time analysis center and receiver, CPU times and receiver CPU times increased with higher starting elements in the sequence.

File size analysis, encryption, and decryption and times for smaller file sizes decrypt in time was higher than encrypt in time for larger files description time gradually became smaller than encryption time sequence effectiveness number selection, sequence, effectively select superior prime numbers with different cardinality enhancing the system, security and, effectiveness flexibility. The proposed system allows flexibility in choosing prime numbers based on the starting element and differences in prime numbers selected for encryption, trade-off performance versus security well the simple RSA crypto system shows stable time progression the proposed system time increases significantly a trade-off is necessary to balance the proposed methodologies quantum resistant features and performance overall the proposed crypto system, while taking more time due to its two phase, execution and sequence generation offers enhance security and resistance to quantum attacks compared to the classical RSA algorithm Sequence has the potential to become a frontier tool against quantum computer attacks.

Overall, the proposed cryptosystem, while taking more time due to its two-phase execution and NS-sequence generation, offers enhanced security and resistance to quantum attacks compared to the classical RSA algorithm. The NS sequence has the potential to become a frontier tool against quantum computer-assisted attacks.

The risk of breaking out of public-key cryptography is on the horizon because of quantum computers' plausible introduction in the future. NS sequence-based cryptosystem is proposed to counter quantum computer-assisted brute force attacks. The results obtained show promising results and illustrate the proposed methodology's computational extensiveness, consisting of two phases.

The most important feature of the NS sequence is to generate a reasonably separate set of new numbers if the starting element of the sequence is changed. The results also show the effectiveness of the NS sequence in selecting superior prime numbers with different cardinality, enhancing the proposed system's security and effectiveness. The proposed system provides the flexibility of choosing prime numbers as per starting and differences of prime numbers selected for encryption in plain RSA and the proposed cryptosystem.

9. Future works

We are planning to investigate No-Sum (NS) sequences in cryptographic applications that have a lot of potential to improve security, effectiveness, and versatility. Subsequent investigations ought to concentrate on propelling algorithmic advancement, augmenting security attributes, customizing applications, refining statistical attributes, contemplating pragmatic implementation facets, and investigating interdisciplinary applications. The cryptographic community may further develop and enhance the usefulness of NS sequences and make them a strong instrument for safeguarding contemporary data and communication networks by tackling these research topics.

Conflict of interest

The authors declare there is no conflict of interest at any point with reference to research findings.

References

[1] Stinson DR, Paterson M. *Cryptography: Theory and Practice (Textbooks in Mathematics)*. USA: CRC Press; 2018.

[2] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*. 1978; 21(2): 120-126.

[3] Gaur SS, Mohapatra AK, Roges R. An approach to improve the complexity of el-gamal cryptographic algorithm and its simulation. *Technology*. 2017; 8(3): 75-83.

[4] Shankar K, Eswaran P. Rgb based multiple share creation in visual cryptography with elliptic curve cryptography aid. *China Communications*. 2017; 14(2): 118-130.

[5] Tas N, Uçar S, Özgür NY, Kaymak ÖÖ. A new coding/decoding algorithm using Fibonacci numbers. *Discrete Mathematics, Algorithms, and Applications*. 2018; 10(02): 1850028.

[6] Nalli A, Ozyilmaz C. The third order variations on the Fibonacci universal code. *Journal of Number Theory*. 2015; 149: 15-32. Available from: https://doi.org/10.1016/j.jnt.2014.07.010.

[7] Li ZH, Lu B, Xu HM, Li G, Li W, Li X. New algorithm for public key cryptosystems based on Lucas sequences. In: *2012 8th International Conference on Wireless Communications, Networking, and Mobile Computing*. Shanghai, China: IEEE; 2012. p.1-4.

[8] Lai H, Pieprzyk J, Orgun MA. Novel quantum key distribution with shift operations based on Fibonacci and Lucas valued orbital angular momentum entangled states. *Physica A: Statistical Mechanics and its Applications*. 2020; 554(6): 124694.

[9] Rawal BS, Shah J. SUDP: The frontier tool for security in 5g and beyond wired or wireless communication. In: *2021 IEEE Globecom Workshops (GC Wkshps)*. Madrid, Spain: IEEE; 2021. p.1-6.

[10] Rawal BS, Singh R, Liang S, Peter A, Biswas A. Augmenting AES to quantum safe level with no-sum sequence. *IEEE MAC 2024 Second International Conference Microwave, Antenna and Communication*. India: Dehradun, Uttarakhand; 2024.

[11] Wen ZX, Zhang JM, Wu W. On the regular sum-free sets. *European Journal of Combinatorics*. 2015; 49: 42-56. Available from: https://doi.org/10.1016/j.ejc.2015.02.030.

[12] Léo D, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehlé D. Crystals-dilithium: Digital signatures from module lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2018; 2018(1): 1-31.

[13] Melissa A, Bronchain O, Cassiers G, Hoffmann C, Kuzovkova Y, Renes J, et al. Protecting dilithium against leakage: revisited sensitivity analysis and improved implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*. 2023; 2023(4): 58-79.

[14] Ahmed M, Moustafa N, Barkat A, Haskell-Dowland P. *Next-Generation Enterprise Security and Governance*. USA: CRC Press; 2022.

[15] Whyte W. *Falcon: New Post-Quantum Cryptography Standard Advances Data Security*. Available from: https://www.qualcomm.com/news/onq/2022/07/falcon--how-this-new-u-s--adopted--qualcomm-backed-cryptography- [Accessed 22nd July 2022].

[16] Pornin T, Prest T. More efficient algorithms for the ntru key generation using the field norm. In: Lin D, Sako K. (eds.) *Public-Key Cryptography-PKC 2019. PKC 2019. Lecture Notes in Computer Science, vol. 11443*. Heidelberg: Springer; 2019.

[17] National Security Agency. *Quantum Computing and Post-Quantum Cryptography*. Available from: https://media.defense.gov/2021/Aug/04/2002821837/-1/-1/1/Quantum_FAQs_20210804.PDF [Accessed 22nd July 2022].

[18] Riesel H. *Prime Numbers and Computer Methods for Factorization, Vol. 126*. Germany: Springer Science and Business Media; 2012.

[19] Mailloux LO, Lewis II CD, Riggs C, Grimaila MR. Post-quantum cryptography: What advancements in quantum computing mean for its professionals. *IT Professional*. 2016; 18(5): 42-47.

[20] Flaut C. Some application of difference equations in cryptography and coding theory. *Journal of Difference Equations and Applications*. 2019; 25(7): 905-920.

[21] Lai H, Orgun MA, Pieprzyk J, Li J, Luo MX, Xiao JH, et al. High-capacity quantum key distribution using chebyshev-map values corresponding to Lucas numbers coding. *Quantum Information Processing*. 2016; 15(11): 4663-4679.

[22] Gibney E. Hello quantum world! Google publishes landmark quantum supremacy claims. *Nature*. 2019; 574(7779): 461-462.

[23] Bennett CH. Quantum cryptography. *Scientific American*. 1992; 267(4): 50-57.

[24] Liu Z, Choo K-KR, Grossschadl J. Securing edge devices in the post-quantum Internet of Things using lattice-based cryptography. *IEEE Communications Magazine*. 2018; 56(2): 158-162.

[25] Li ZP, Wang JR, Zhang WY. Revisiting post-quantum hash proof systems over lattices for the Internet of thing authentications. *Journal of Ambient Intelligence and Humanized Computing*. 2020; 11(6): 3337-3347.

[26] Broadbent A, Schaffner C. Quantum cryptography beyond quantum key distribution. *Designs, Codes, and Cryptography*. 2016; 78(1): 351-382.

[27] Koziel B, Azarderakhsh R, Kermani MM, Jao D. Post-quantum cryptography on FPGA based on isogenies on elliptic curves. *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2016; 64(1): 86-99.

[28] Aumasson J-P. The impact of quantum computing on cryptography. *Computer Fraud and Security*. 2017; 6: 8-11. Available from: https://doi.org/10.1016/S1361-3723(17)30051-9.

[29] Bernstein DJ, Heninger N, Lou P, Valenta L. Post-quantum RSA. In: *International Workshop on Post-Quantum Cryptography*. Germany: Springer; 2017. p.311-329.

[30] Hamid HAA, Rahman SkMM, Hossain MS, Almogren A, Alamri A. A security model for preserving medical significant data privacy in a healthcare cloud using a fog computing facility with pairing-based cryptography. *IEEE Access*. 2017; 5: 22313-22328.

[31] Ateniese G, Benson K, Hohenberger S. Key-private proxy re-encryption. In: *Lecture Notes in Computer Science*. Berlin Heidelberg: Springer; 2009. p.279-294.

[32] Gaur SS, Mohapatra AK, Roges R. An approach to improve the complexity of el-gamal cryptographic algorithm and its simulation. *Technology*. 2017; 8(3): 75-83.

[33] GitHub-derektypist/semiprime-numbers: According to Wikipedia. Available from: https://github.com/derektypist/semiprime-numbers [Accessed 27 November 2024].