Nigerian Journal of Theoretical and Environmental Physics (NJTEP)
ISSN Print: 3026-9601
DOI: https://doi.org/10.62292/njtep.v2i2.2024.22
Volume 2(2), June 2024

Quantum Cryptography for Enhanced Data Security: A Comparative Survey of Data Encryption and Decryption of Error Correction

Alfa, B., *Tsepav, M. T., Adamu, Y., and Adesiyan Victor Adedeji

Department of Physics, Ibrahim Badamasi Babangida University, Lapai, Niger State.

*Corresponding author's email: tmathew39@ibbu.edu.ng

ABSTRACT

Data information security is a crucial concern that ought to be managed to help protect vital data. Cryptography is one of the conventional approaches for securing data and is generally considered a fundamental data security component that provides privacy, integrity, confidentiality, and authentication. In this paper, a comparative survey of Rivest-Shamir-Adleman(RSA)data security algorithm is proposed in order to compare their security strength. In doing so, we integrate traditional RSA and Gaussian Interpolation formulas to test their security level. The integration raised the security strength of RSA to the fifth degree,whilethe Gaussian First Forward Interpolation was used to encrypt the ASCII values of the message after which the traditional RSA was used to encrypt and decrypt the message in the second and third levels. Comparative data security analysis was performed using four different algorithms; RSA, SRNN, 2-Key pair algorithms, and the results showed that when the data size was small, the encryption and decryption times were lower for the proposed algorithm but higher when the data size was big. Thus, the two public keys used and some mathematical relations made the eavesdropper not to get much knowledge about the key and therefore, unable to decrypt the message.

Keywords:
Gaussian Interpolation,
Cryptographic algorithm,
Quantum key distribution,
Quantum information,
Data Security.

INTRODUCTION

The increasing dependence on digital technologies has led to a growing demand for secure and privacy-preserving protocols. Quantum cryptography has emerged as a promising solution to addressing these challenges, particularly in the field of cyber security and crypto-currency. The use of quantum cryptographic protocols can provide secure transactions that are resistant to attacks from eavesdroppers. However, the implementation of these protocols poses several challenges, and security and privacy issues need to be carefully considered. Some of the primary challenges in the implementation of these protocols are the characteristics of the quantum computer, many existing public key quantum cryptography (RSA) and elliptic curve cryptography (ECC) may no longer be safe in the quantum computer(Tseng. 2007). The well-known discrete logarithm problem (DLP) or the integer factorization problem will no longer be difficult under quantum computer. This suggests that in order toresist attacks from eavesdroppers, new unbreakable encryption and decryption protocol systems that are notbased on discrete logarithms problem should be explored. This is one of the ways that the information security of cyberspace couldbe ensured in thefutureInternet.In this paper, we explore a rangeof quantum cryptographic protocols that are essential for improving quantum information security, including quantum key distribution (encryption), post-quantum cryptography (decryption), counterfactual quantum key distribution, and key management. Through the analysis, we aim to develop a robust and new improved quantum cryptographic protocols and information processing techniques based on the theoretical analysis and experimental results, making it harder for malicious entities to intercept or tamper with sensitive communication information.

Here, we propose a method that can increase the secret key rate in QKD by simple additional actions of the legitimate users, namely the encryption and decryption of information that is disclosed during error correction. Encryption of the post processing data is also used to simplify the security proof which does not allow an eavesdropper to perform the best possible measurement.

Preliminaries and Assumptions

In many respects, quantum communication and information processing are superior to that of classical,

which is rooted in the characteristics of quantum information. These characteristics are enshrined in the uncertainty principle, quantum no-cloning theory; the quantum teleportationetc, such hidden characteristics of quantum information can be employed to resist attack (passive or active attack in cyberspace communication.Here, we focus on the four stages of QKD protocol that are the most significant forour study:

Stage I. The quantum states are sent via a quantum channel between two users namely, First user and second user, then the legitimate users utilize a classicalauthenticated channel to perform key sifting and/or basis reconciliation.

Stage II. The legitimate users estimate the intervention of a third party and the information availablebased on the data observed at the receiver's side.

Stage III: The legitimate users perform error correction, which provides them with coinciding
keys correlated with the third party. The legitimate users take into account that some secret dataare disclosed during error correction. The disclosed data actually specify the set ofcodewords used by the first User.

Stage IV:Finally, the privacy amplification stage follows. This results in a shorter key with very
low correlation with the eavesdropper (Third Party). The length of the final key depends on the dataobserved by the legitimate users and their estimate of the third party's own information. In addition, the security proof, i.e., the proof of the statementthat the key obtained with this formula is secure according to the security parameteris the main theoretical element for the QKD protocol.

MATERIALS AND METHODS

The methodological approach for the study is based on integrating traditional RSA algorithm and Gaussian Interpolation formalism to enhance and strengthen data security that may be prone to eavesdroppers. Here, two approaches were adopted for comparison namely; the Gaussian Forward and Backward Interpolation are integrated with the traditional RSA algorithm and the RSA algorithm for Quantum Key Distribution (QKD) using an encryption key (e,n) and decryption key (d,n) to enhance data security strength.Their approach involves 3rd degree of encryption and 2nd degree level of decryption. This approach has also been used in most digital data, information and telephone security applications and it has the advantages of being a reliable and safe system. For this reason, it was used in hybrid cryptographic systems that simultaneously use symmetric algorithms (AES) for the communication and data encryption phase and public key algorithms (RSA) for the safe delivery of the symmetric key (or session key) that is necessary for encrypting and decrypting the message.

RSA Approach

Here, we used the Rivest-Shamir-Adleman (RSA) algorithm in approach, which is one of the most popular and secure public-key encryption methods. The algorithm capitalizes on the fact that there is no efficient way to factor very large (100-200 digit) numbers. Therefore, an encryption key (e,n) was used. The bases of the algorithm used are as follows:

i. Represent the message as an integer between 0 and (n-1). Large messages can be broken up into a number of blocks. Each block would then be represented by an integer in the same range.
ii. Encrypt the message by raising it to theethpower modulo n. The result is a cipher text message C. To decrypt cipher text message C, raise it to another power d modulo n. The encryption key (e, n) is made public. The decryption key (d, n) is kept private by the user.
iii. Determine the Appropriate Values for e, d, and n
iv. Choose two very large (100+ digit) prime numbers and denote these numbers as p and q.
v. Set n equal to p * q.
vi. Choose any large integer, d, such that GCD (d, ((p-1) * (q-1))) = 1
vii. Find e such that e * d = 1 (mod ((p-1) * (q-1)))

Rivest, Shamir, and Adleman(1977) provided efficient algorithms for each required operation as shown in Figure 1.

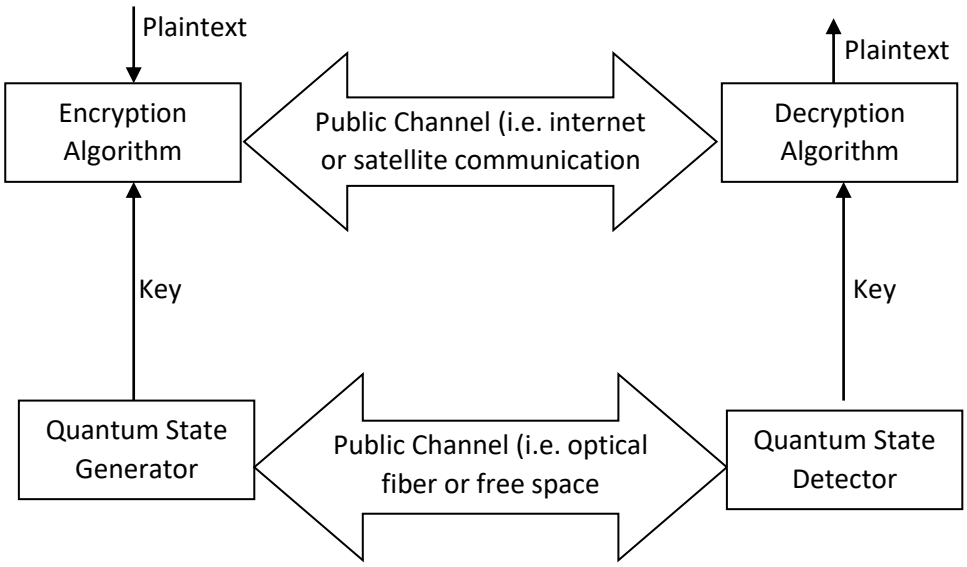

Figure 1: Proposed workflow diagram for Quantum Key Distribution Implementation in Matlab

Quantum cryptography Approach

Quantum Cryptography is a technology based on the phenomenon of Quantum Mechanism and entanglement, to generate one state at a time. The quantum cryptography uses the polarization properties for encoding the data in photon form. In quantum cryptography, security provides the key to encryption and decryption of information for communication. Here, we used Gaussian Interpolation formalism to enhance and strengthen data security; therefore, in order to simulate real situations for data security, the QKD protocol was first analyzed.In doing so, the algorithm for the encryption and decryption time fordata sizes (IKB, 2KB, 5KB and 10KB) were encoded using SRNN, RSA and the proposed protocol.

The probability of the existence of an eavesdropper on the QKD protocol is as follows.

$$P_{rob} = P_{rob}\{Base_S = Base_R \text{^} Measure_S \neq Measure_R\} \quad (1)$$

where$Measure_S$ and$Measure_R$ are the measurements based from sender and receiver respectively. The probability that the eavesdropper is found for 1-bit quantum information is calculated as:

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8} \quad (2)$$

Thus, in this scenario the more the number of transmission of data the higher the probability of the eavesdropper being detected, no matter whether there is noise interference or not. Figure 2 is the proposed workflow diagram of the algorithm for Gaussian Forward and Backward interpolation.

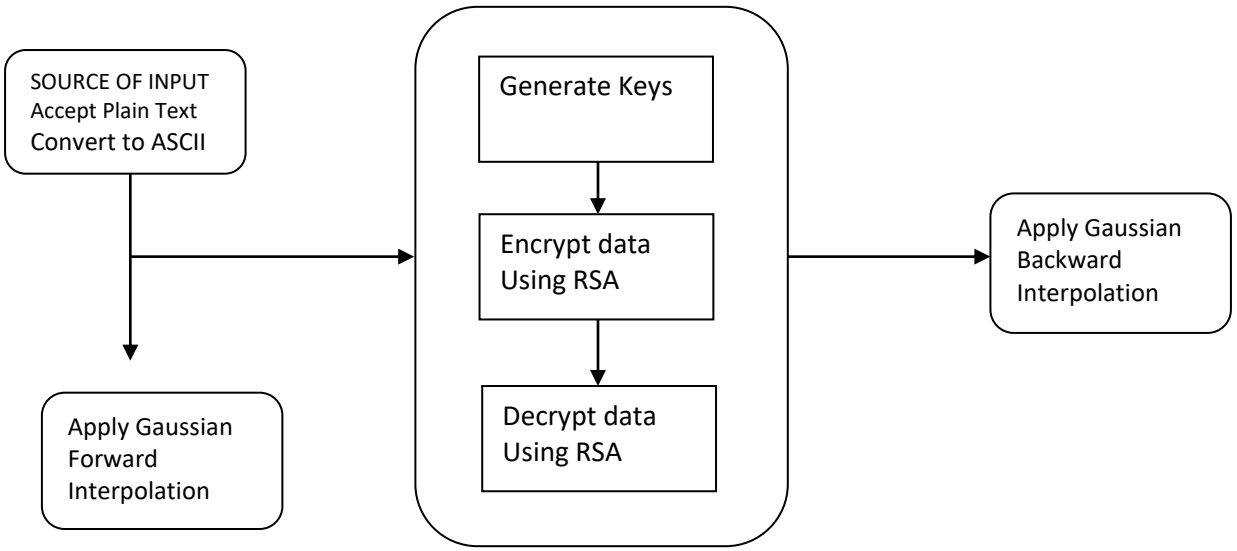

Figure 2: Proposed workflow diagram of the algorithm for Gaussian Forward and Backward interpolation

Implementation

The researchwas undertaken in order to compare and strengthen data security using a better and faster algorithms for implementation of RSA system in Quantum Cryptography. In doing so, the proposed approach was implemented by integrating traditional RSA system and Gaussian Interpolation formula for the purpose of strengthening of data security. The Quantum Cryptography for RSA system was used to secure textual data with RSA cryptography using MATLAB to generate a faster public-key cryptography while Quantum Cryptography for the Gaussian Interpolation uses the Gaussian Forward and Backward Interpolation which are integrated with the traditional RSA algorithm to enhance security strength by addressing the factorization problem of RSA as indicated by Rutkowski and Houghten (2020). This approach involves 3rd degree of encryption and 2nd degree level of decryption.

The general overview of the algorithms is that an input partition is created where the plaintext alphabets are converted to their corresponding ASCII values. Thus, Gaussian Forward

Interpolation is applied on the ASCII values which give it first encryption strength. RSA is then applied to the Cipher text results. This involves the key generation, encryption, and decryption based on the results from the ASCII values. Finally, the Gaussian Backward Interpolation is applied on the decrypted values from RSA to obtain the plaintext.

RESULTS AND DISCUSSION

Source of input for ASCII values in Matlab

Step 1: Here, a Matlab 2023a based simulation was used to "Run" the rsa.mMatlab code. The command used here was to enter the values of p (as prime number) as shown below

Implementation of RSA Algorithm

fx: Enter the value of p:

Enter the value of p: 13

fx: Enter the value of q:

Enter the value of q: 17

Note that: p and q could be any large prime number

Step 2: Enter to obtain the calculated public and private keys for RSA as shown below:

The value of (N) is : 221
The public Key (e) is : 5
The value of (Phi) is : 192
The private key (d) is : 77

fx: Enter the message:-

Step 3: Here, a message is entered which can be any plain text "VictorDeji"

The value of (N) is : 221
The public Key (e) is : 5
The value of (Phi) is : 192
The private key (d) is : 77

fx: Enter the message: VictorDeji

Note that the enteredmessageis encrypted at this stage

Step 4: After entering the ASCII code of entered message is shown, Cipher text ASCII message , Decrypted ASCII message and finally shows Decrypted Message as entered message.

The value of (N) is : 221
The public Key (e) is : 5
The value of (Phi) is : 192
The private key (d) is : 77
Enter the message: VictorDaji
ASCII Code of the entered Message:
83 104 105 112
114 97 32 83 97
104 117
Cipher Text of the entered Message:
70 117 209 125
173 54 2 70 54
117 104
Decrypted ASCII of message:
83 104 105 112 114 97
32 83 97 104 117
Decrypted message is: Victor Deji

fx>>

Source of input for ASCII values

The source input is where data is accepted from the user as plain text. The individual characters in the string are converted to ASCII values. This converts the alphabets to numbers which is a form of encryption. For example, the ASCII value for 'A' is "065".

Algorithm 1 Source of Input

Input: Accept message from user
Output: ASCII values
1: **begin**
2: message entered;
3: string s = value;
4: var buffer = Encoding.ASCII.GetBytes(s);
5: for each (byte c in buffer)
6: {
7: Console. Write (c);
9: }
10:
11: ASCII
11: **End**

Gaussian Forward Interpolation

Here, we consider the first level Gaussian Interpolation formulae, assuming $t = g(h)$ computationally, this is a relation with variables h and t by Balasubramanian; (2014). In any condition of the values h, then the values will be $h0 - 2h, h0 - 1, h0, h0 + 1x0 + 2h.$ This produces the corresponding values as shown in Table I

Table 1: Central Differential Table

x	y	First Degree	Second Degree	Third Degree	Fourth Degree
h0-2h	y-2				
		$\Delta y - 1$			
h0-1	y-1		$\Delta^2 y0$		
		$\Delta y0$		$\Delta^3 y1$	
h0	y0		$\Delta^2 y1$		$\Delta 4 y0$
		$\Delta y1$		$\Delta^3 y2$	
h0 +1	Y1		$\Delta^2 y2$		
		$\Delta y2$			
h0 + 2	y2				

Static Initial ASCII Value

$m0 = c0$ (3)

where m_0 is the initial value for the ASCII values obtained from the first alphabet in the string of numbers generated. c_0 is maintained as the same value for the ASCII values for $m0$

The First Forward Gaussian Differential Formula proposed in Uddin, *et al*. (2019) is given in equation 3.

Therefore the first forward difference formula is

$\Delta y - 1 = (y - 1(-(y - 2)))$ (4)

where the First Degree values using equation 3 is obtained from the difference between the former and the initial y values. Thus $y-1$ is the former value and $y-2$ is the initial value.

The central difference is deduced as

$y1 - y0, y2 - y1, y3 - y2 \ldots \ldots \ldots yk - yk$ (5)

Hence $\Delta y0, \Delta y1, \Delta y2, \Delta yk - 1$ can be obtained using equation 4. $\Delta y0$ represents the First Forward Differential and this can be deduced from equation 3. From $eq4$ it can be deduced that the First Forward Difference will be

$\Delta y0 = y1 - y0$ (6)

Integrate equation 3 and equation 5 to obtain the proposed First Forward Gaussian Differential Formula

$m0 = c0$

$\Delta y0 = y1 - y0$ (7)

Algorithm 2 First Forward Gaussian Differential Formula

Input: ASCII values

Output:Ciphertext

```
1       : varascii_values = Encoding.ASCII.GetBytes (message);
2       :varinitialEncryptedValues = new List<int>();
3       :       int initial = ascii_values[0];
4       :       initialEncryptedValues. Add (ascii_values [0]);
5       :       for (byte i = 1; i<ascii_values. Length; i++)
6:              {
7:              initialEncryptedValues.Add(ascii_values[i] - initial);
8:              initial = ascii_values[i];
9:              }
10: End
```

RSA Algorithm

RSA has been used as a public cryptographic scheme and as per literature is known to have a lot of weaknesses and also with higher execution time (Bonde and Bhadade, 2017). Hence, this work aimed at using the variant RSA to raise its security strength while reducing execution time.

Algorithm 2 RSA

Input: two random prime numbers (p, q)

Output: public key (), private key ().

Encrypted data;

Decrypted data;

```
1       : begin
2       : //generating
3       :       var byte primeNumber1;
4       :       var byte primeNumber2;
5       :       var byte e;
6       :       var byte d;
7       :       double n = primeNumber1 * primeNumber2;
8 :             //      varsquareDiff = (primeNumber1 - 1) * (primeNumber2 - 1);
9 :
10      :       var encrypted = new List<byte>();
11      :       for (int i = 0; i<initialEncryptedValues.Count; i++)
12      :       {
13      :       var remainder = Math. Pow(initialEncryptedValues[i], e) % n;
14      :       encrypted. Add ((byte) remainder);
15      :       }
16      :        //decipher
17      :       var decrypted = new List<byte>();
18      :       for (int k = 0; k <encrypted.Count; k++)
19      :       {
20      :       var remainder = Math.Pow(encrypted[k], d) % n;
21       :          decrypted. Add ((byte)remainder);
22       :      }: End
```

Gaussian Backward Interpolation

Static Initial Ciphertext Value

Here , we conside that

$d0 = p0$ (8)

Where $d0$ is the initial Ciphertext values obtained for the first decrypted alphabet in the string of numbers generated.$p0$ is maintained as the plaintext value for the decrypted values for $d0$

From the existing Gaussian Forward Interpolation Formula

$Yp = y0 + p\Delta y0 + (p-1)\Delta 2y-1/2! + ((p + 1)(p - 1)3! + (p+1)(p-1)(p-2)\Delta 4y-2/4! +\ldots$ (9)

It can be deduced from (4) that

$\Delta 2y - 1 = \Delta y0 - \Delta y -1$ (10)

Such that

$\Delta y0 = \Delta y1 + \Delta 2y - 1$ (11)

This suggests that

$\Delta 2y0 = \Delta 2y - 1 + y3 - 1$ (12)

$\Delta y0 = \Delta y3 - 1 + \Delta 4y - 1$ (13)

Hence the Gaussian Backward Interpolation Formula can be written as;

$d0 = p0$ (14)

$\Delta y0 = \Delta y3 - 1 + \Delta 4y - 1$ (15)

Algorithm 2 First Backward Gaussian Differential Formula

Input: Decrypted values

Output: Plaintext

```
1       :       Begin
2       :       varplainArr = new List<byte>();
3       :               varstartIndex = 1;
4       :               plainArr.Add (ascii_values[0]);
5       :               while (startIndex<decrypted.Count)
6       :               plainArr.Add ((byte)(ascii_values[startIndex - 1] + decrypted[startIndex]));
7       :               startIndex++;
8       : End
```

Stages of the Algorithm Implementation
STAGE I: Convert the message to be encrypted to its ASCII values
The message to be encoded is "encrypt"
101110099114112116
STAGE II: Apply Gaussian First Forward Interpolation on ASCII values
$\Delta y0 = y1 - Y0$
101990249715101
STAGE III: Key Generation
Choose two prime numbers p and q, p ≠ q
P = 811
Q = 281
Computing n i.e. n = p * q
n acquired. n = 811
Applying Difference of squares ∅ (n) = $(p + 1).(p - 1).(q + 1).(q - 1)$
Compute PHI of $n$using formula (p+1).(p -1). (q+1).(q-1)= 433
Compute $e$such that 1<e<θ(n) and gcd (e,n) =1 ie. e = 62
Computing dCompute d = e-1 mod ∅n
i.e d = = e-1 mod θ(n) :: d is 0.016129032258064516
STAGE IV: Encrypt message using the formula; $Kpt = (d, n)$
Messages successfully encrypted.
36610.0, 159006.0, 102987.0, 144536.0, 67850.0, 77584.0, 102102.0
STAGE V: Decrypt a message
$Decrypt = Cd$ % n
101990249715101
STAGE VI: Apply Gaussian Backward Differential information on
10111099114121112116

The experimental setup for this work was carried out using a 64bit system to implement the cryptographic hybrid algorithm, the output of the algorithm as well as the comparison of the security strength and the execution time with other schemes. The hybrid algorithm was implemented employing C++ language and Matlab. The security strength and the execution metrics was compared with Devi (2016) Rivest, Shamir and Adleman (1978) and Panda and Chattopadhyay (2017). The proposed algorithm was an integration of the Gaussian Interpolation formula with Traditional RSA. This raises the encryption and decryption degree to the fifth level. Table 2 depicts the comparison of the scrambling and unscrambling time of three algorithms, RSA, 2-KEY Pair, SRNN, and the proposed algorithm

Table 2: Comparing the encryption and decryption time of the four algorithms ASCII values

Text Size	RSA	2-Key Size	SRNN	Proposed	RSA	2-Key Size	SRNN	Proposed
1KB	112	27	18	11.8	30	81	60	13.7
2KB	25	61	24	54	76	167	133	82
5KB	72	107	55	83	268	488	443	342
10KB	155	111	71	122	573	1037	854	932

Note: RSA adopted (Rivest, Shamir and Adleman, 1978), 2-key Size (Bonde and Bhadade, 2017) and SRNN (Devi, 2016)

From Table 2, it can be deduced that the proposed algorithm's encryption and decryption time for data size of IKB is the lowest. On the other hand, as the text size increased to 10KB, the encryption time was higher than the 2-key size and SRNN but lower than traditional RSA algorithms. Again the decryption time for the proposed algorithm was higher than RSA and SRNN but lower than the 2- Key size algorithm. This was as a result of obtaining the ASCII values for all the alphabets in the string and applying the Gaussian First Forward and Backward Interpolation formula on the values. This increases the computational time for both the encryption and decryption processes. The encryption and decryption cost of the comparison is shown in Figure 3 and 4.

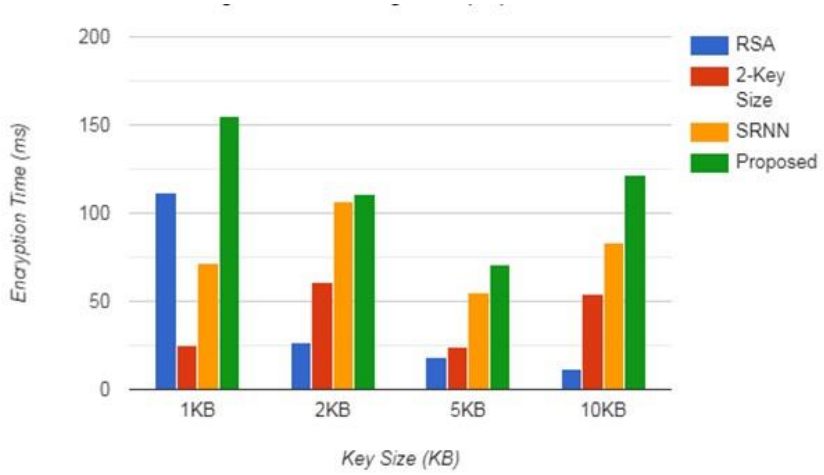

Figure 3: Scrambling Time for Four Algorithm (ms)

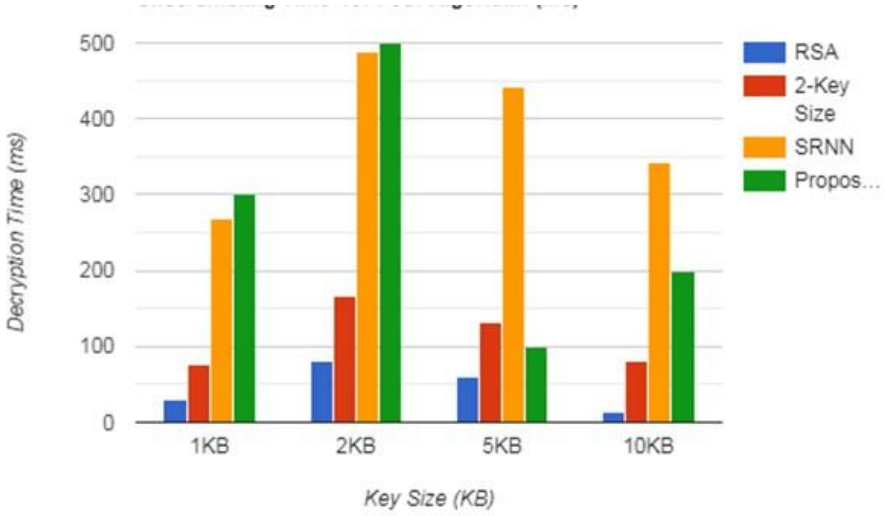

Figure 4: Unscrambling Time for Four Algorithms (ms)

In quantum cryptography, scrambling and unscrambling time relate to the procedures involved in encoding and decoding quantum information to enable secure communication. These notions are crucial to quantum key distribution (QKD), a method used to establish safe cryptographic keys between two parties. Figure 3 and 4shows the time representation of the scrambling and unscrambling process for RSA, 2-Key Size, SRNN and Proposed Algorithm respectively.

Time Considerations

Scrambling Time: The time needed to encode the information into qubits and prepare them for transmission is normally low and relies on the intricacy of the encoding operation as shown in Figure 3.

Unscrambling Time: Decoding the information at the receiver's end includes completing measurements and calculations to retrieve the encoded information effectively. This procedure may take longer and could be impacted by variables such as the efficiency of measuring methods and the complexity of reconciliation algorithms as shown in Figure 4.

CONCLUSION

This study has highlighted the significant potential of quantum cryptography in revolutionizing the security and privacy of data in the quantum era. This research developed an enhanced RSA scheme using a Matlab algorithm and a Gaussian Interpolation Formula with the traditional RSA, which has raised the security strength of traditional RSA. In addition, it has increased the encryption to a third-degree level and also the decryption to a second degree. The analysis from the simulation indicated that the execution time was lower when the text size is small but increased when the text size increases. Additionally, in this research work, an algorithm is proposed for RSA a method for

implementing a public-key cryptosystem (RSA) using two public key and some mathematical relation. These two public keys are sent separately, this makes the attacker not to get much knowledge about the key and unable to decrypt the message. The proposed RSA is used for system that needs high security with normal speed. As a future work multiple file encryption and decryption can be possible. Great level of security is achieved using this algorithm. Modified RSA algorithm for file transmission algorithm can be used where high security file transmission required in public forums.

REFERENCES

Balasubramanian, K. (2014)."Variants of RSA and their cryptanalysis. *InternationalConference on Communication and Network Technologies*,pp.145-149, doi:10.1109/CNT.2014.7062742.

Bonde S. Y. and U. S. Bhadade, "Analysis of encryption algorithms (RSA, SRNN and 2 keypair) for information security," *2017 International Conference on Computing,Communication, Control and Automation (ICCUBEA)*, 2017. **doi:** 10.1109/ICCUBEA.2017.8463720

Devi M. S.(2016). "Threshold SR2N public key cryptosystem," *International Journal of Engineering Trends and Technology*, vol. 31, no. 1, pp. 15–17. **doi**: 10.14445/22315381/IJETT-V9P265

Panda P.K.and S. Chattopadhyay (2017). "A hybrid security algorithm for RSA cryptosystem," in 2017 4th International *Conference on Advance Computing and Communication Systems*(ICACCS), **doi:** 10.1109/ICACCS.2017.8014644

Rivest R. L., A. Shamir, and L. Adleman (1978). "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126,.
https://doi.org/10.5772/intechopen.114973

Rivest, D. R., Shamir, A., & Adleman, L. (1977). RSA (cryptosystem). *Arithmetic Algorithms And Applications*, 19. https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Rutkowski E. and S. Houghten (2020). "Cryptanalysis of RSA: Integer Prime Factorization UsingGenetic Algorithms," IEEE Congress on Evolutionary Computation (CEC), p.1-8, **doi**: 10.1109/CEC48606.2020.9185728.

Tseng Y.-M., (2007). An efficient two-party identity-based keyexchange protocol," *Informatica*, vol. 18, no. 1, pp. 125–136, https://eprint.iacr.org/2009/441

UddinM., Md. Kowsher and Mir Md. Moheuddin, (2019)."A New Method Of Central DifferenceInterpolation", *Applied Mathematics and Sciences An International Journal (MathSJ)*, vol. 6, no.3, pp. 01-14, **do**i:10.5121/mathsj.2019.6301.