An Enhanced Rsa Algorithm For Data Security Using Gaussian Interpolation Formula

John Kwao Dawson (kwaodawson1@yahoo.com)
Sunyani Technical University
Frimpong Twum
Kwame Nkrumah University of Science and Technology
James Benjamin Hayfron Acquah
Kwame Nkrumah University of Science and Technology
Beklisi Kwame Ayawli
Sunyani Technical University

Research Article

Keywords: Gaussian Backward Interpolation, Gaussian First Forward Interpolation Formula, ASCII values, Cryptographic algorithm, RSA, Hybrid algorithm, SRNN, 2-Key pair

Posted Date: February 16th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1326669/v1

AN ENHANCED RSA ALGORITHM FOR DATA SECURITY USING GAUSSIAN INTERPOLATION FORMULA

John Kwao Dawson[1]
Sunyani Technical University
kwaodawson@yahoo.com

Dr. Frimpong Twum[2]
KNUST
twumf@yahoo.co.uk

Prof. James Benjamin Hayfron[2]
Acquah
KNUST
jbha@yahoo.com

Dr. Beklisi Kwame Ayawli[3]
Sunyani Technical University
bbkayawli@yahoo.com

ABSTRACT

Data 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 hybrid data security algorithm is proposed by integrating traditional RSA and Gaussian Interpolation formulas. The integration raises the security strength of RSA to the fifth degree. The Gaussian First Forward Interpolation is used to encrypt the ASCII values of the message after which the traditional RSA is used to encrypt and decrypt the message in the second and third levels. The last stage employs Gaussian Backward Interpolation to decrypt the data again. The integration helps to cater to the factorization problem of the traditional RSA. Comparative analysis was performed using four different algorithms; RSA, SRNN, 2-Key pair algorithms, and the proposed algorithm. It is proven that when the data size is small the encryption and decryption times are lower for the proposed algorithm but higher when the data size is big.

KEYWORDS: Gaussian Backward Interpolation, Gaussian First Forward Interpolation Formula, ASCII values, Cryptographic algorithm, RSA, Hybrid algorithm, SRNN, 2-Key pair

1. INTRODUCTION

The provision of protocol and processes necessary to secure a communication channel when an assumed third party exists is considered Cryptography [1]. Cryptographic algorithms are divided into symmetric and asymmetric keys. The symmetric algorithms require a single key only for the encryption and decryption of data. Asymmetric algorithms on the other hand require both public and private keys for the encryption and decryption of data. The scrambling of the data is done using the public key while the private key is made known only to the receiver which is meant for the decryption of the data.

A maiden asymmetric algorithm was proposed by Diffie-Hellman [2][3] which ensures secured communication as well as data security. A counter algorithm termed RSA which has lower time complexity based on prime number factorization was proposed in 1977 and is the patent of Ron Rivest, Adi Shamir, and Len Adleman (RSA), which was published in 1978 at the Massachusetts Institute of Technology [4]. In this algorithm, two prime numbers are used to produce the public and the private key. When the keys are created, the prime numbers are no more considered and are or can be discarded.

RSA is a block cipher algorithm and, as a result all palintext are scrambled at a time using bits based on the same key [5]. RSA complexity is assumed based on calculating the modulus of p as

a result of the multiplication of selected primes of r and s using a value e which is named public key to result in a scrambled data c [6]. It can be deduced that the complexity of RSA is based on computing e which is a combined modulus p [29]. This determines the condition that, upon the selection of modulus p with an open key e which guarantees for every value c $(0,1, \ldots . p-1)$, just one $t(0,1 \ldots \mathrm{t}-1)$ such that

$$C = p^e mod\ n \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

This suggests that if an attacker has access to plaintext and the nth modulus or the value for e, there can be a compromise by factoring n [30]. It can be said that cracking the cipher is by factoring integers [29] and [30]. This implies that if the initial primes are carefully selected, the computation time will be great to factor n even though it is possible using the factorization of n [7]. In the academia and the industry sector, a series of research has been conducted in finding better ways of improving data security. A lot of researchers directed their efforts towards achieving optimum encryption and decryption times while others concentrated on strengthening the security of data. The essence of this paper is to proposed an enhanced hybrid RSA algorithm by integrating the traditional RSA and Gaussian Interpolation formula. The proposed algorithm seeks to strengthen data security by raising the encryption and the decryption stages of the traditional RSA algorithm to fifth-degree, thereby making it resistant against the factorization problem of RSA.

2. RELATED WORKS

A lot of scholars have done various works in-line with data security enhancement. Some aimed at ensuring less execution cost of algorithms while others projected better security of data. To gain a sound understanding, there is therefore the need for a review of literary works in a summarized form as presented in this section.

In [9], the author proposed a modified RSA which uses linear order with chosen integer values with an n^{th} modulus similar to the RSA algorithm. Also, Wazery and Amin [10] proposed another variant of RSA which uses a multiple-level scheme to secure data by first employing RSA cryptosystem and then an embedded scheme that uses random placement for selecting data's coordinates when an image is to be considered. This works using dual-levels by first scrambling data and then mining to reestablish the data.

According to [11], a Certificateless RSA algorithm by integrating a Kilian-Petrank's RSA with a DDH algorithm was also proposed. In their scheme, the private key is the client secured key. The input value now becomes the user's partial key only to ensure the validity of the scheme. Their scheme had strong security features but was based on the assumption that integrating Kilian with DDH is complex.

Another variant of RSA cryptosystem was also proposed by Budiman et, al, which employed a multi-factor RSA scheme [12]. Their scheme worked based on the Agrawal-Biswas scheme which scrambles the data and finally unscrambles the ciphertext. Budiman et, al's work was further improved by [13] by using R Prime RSA which is based on large prime numbers which are more secure than traditional RSA which is based on dual prime values. The security of the R Prime RSA is based on the modulus of n. This means that the higher the modulus the more secured the encryption scheme. This then means if the modulus is less, the security strength becomes weak.

In the works of Bansal and Singh [14], the use of a concurrent indexed list of blocks of characters was also proposed. This has the potential of increasing the encryption and decryption speed of RSA as well as making it compatible with modern industry standards. In the works of Mittal and

Arora a hybrid algorithm that integrates Blowfish and RSA algorithm was proposed [15]. This technique serves both symmetric and asymmetric purposes which makes it efficient.

In [16], Amalarethinam and Leena proposed an Enhanced RSA (ERSA) that injects two additional prime values compared with the traditional RSA. This has the objective of lessening the execution time by breaking the data into units aiming at increasing the security strength of the algorithm. A hybrid algorithm was also proposed by Kaliyamoorthy and Ramalingam [17] that integrates RSA and image steganography. The RSA encrypts the data and the image steganography encapsulates the data from a hacker.

Quasi-modified levy flight integrated with RSA was also proposed by Bharathi et.al [18]. The RSA was used to encrypt the data while the Quasi based modified levy flight was used to generate the keys which helped to boost data integrity.

In [19] Khan et. al proposed a hybrid algorithm based on the Gulliou-Quisquater scheme and RSA. This is aimed at ensuring data integrity based on the generation of the key using the RSA while the Gulliou-Quisquater scheme does the integrity and confidentiality checks.

A three-level encryption technique with the objective of overcoming the use of a single key for the encryption and decryption of data through the merge of Advance Encryption Scheme, Data Encryption Standard, and RSA was proposed by Suhasini and Bushra [20].

In [21] Mondol and Mahmood proposed a hybrid algorithm utilizing RSA, Blowfish, and Secure Hash Algorithm -2. The RSA ensured the authentication of the clients while the confidentiality of the data was secured using Blowfish and the data integrity is secured using Secure Hash Algorithm-2.

Subasini and Bushra [22] proposed a hybrid cryptographic scheme based on RSA, AES, and other cryptographic keys. This was meant to secure the safe transfer of data from the client-side to the cloud service provider and vice versa.

Mondol and Mahmood [23] proposed the use of the RSA scheme to encrypt the data. The RSA is meant for the estimation of different attributes such as Moment Difficulty, Throughput, and Area Difficulty. The scrambling is performed at the cloud service provider's end and the encryption at the cloud client's end. In view of the various attempts to help provide algorithms to ensure the security of the cloud, there is still a gap as can be cited in the works of Kausar Khan et.al [8].

3. METHODOLOGY

This section presents our proposed methodology of integrating traditional RSA and Gaussian Interpolation formulas for the purpose of strengthening data security. The Gaussian Forward and Backward Interpolation are integrated with the traditional RSA algorithm to enhance security strength by addressing the factorization problem of RSA as indicated by [30]. Their approach involves 3rd degree of encryption and 2nd degree level of decryption.

4. PROPOSED FRAMEWORK

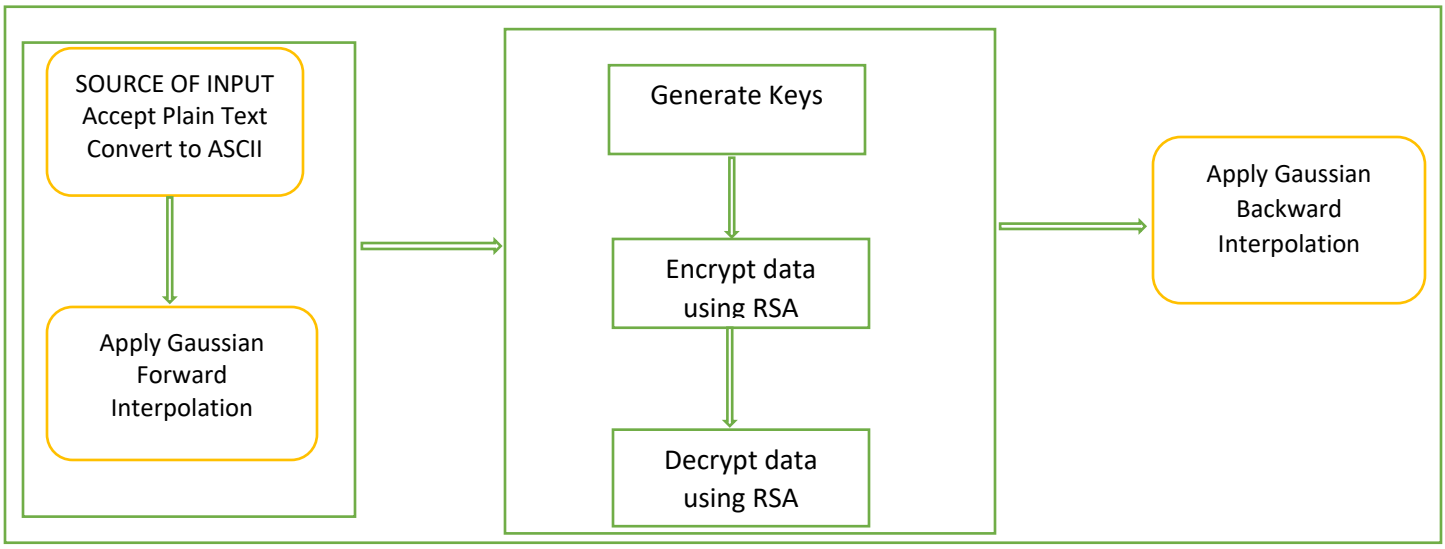

Figure 1: Workflow diagram of the proposed algorithm

Figure 1, gives a general overview of the proposed algorithm. There is an input section where the plaintext alphabets are converted to their corresponding ASCII values. Gaussian Forward Interpolation is applied on the ASCII values which gives it first encryption strength. RSA is then applied to the Ciphertext results. This involves the key generation, encryption, and decryption based on the results from the ASCII values. Apply Gaussian Backward Interpolation on the decrypted values from RSA to obtain the plaintext.

5. SOURCE OF INPUT

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
```

6. APPLY GAUSSIAN FORWARD INTERPOLATION

In this work, the First Level Gaussian Interpolation Formula will be considered. Computationally, assuming $t = g(h)$, this is a relation with variables h and t [24]. 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 1.

Table 1: Central Differential Table

x	y	First Degree	Second Degree	Third Degree	Fourth Degree
h 0-2h	y-2				
		$\Delta y - 1$			
h 0-1	y-1		$\Delta^2 y0$		
		$\Delta y0$		$\Delta^3 y1$	
h0	y0		$\Delta^2 y1$		$\Delta^4 y0$
		$\Delta y1$		$\Delta^3 y2$	
h 0 +1	Y1		$\Delta^2 y2$		
		$\Delta y2$			
h 0 + 2	Y2				

7. STATIC INITIAL ASCII VALUE

$m0 = c0$……1

Where $m0$ is the initial value for the ASCII values obtained from the first alphabet in the string of numbers generated. $c0$ is maintained as the same value for the ASCII values for $m0$

The First Forward Gaussian Differential Formula proposed in [24] is given in equation 2. Therefore the first forward difference formula is

$\Delta y - 1 = \left(y - 1\left(-(y - 2)\right)\right)$ ……..2

Where the First Degree values using equation 2 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

$y1 - y0, y2 - y1, y3 - y2$ … … … $yk - yk$……3

Hence $\Delta y0, \Delta y1, \Delta y2, \Delta yk - 1$ can be obtained using equation 3.

$\Delta y0$ represents the First Forward Differential and this can be deduced from equation 2.

From $eq3$ it can be deduced that the First Forward Difference will be

$\Delta y0 = y1 - y0$……4

Integrate equation 1 and equation 4 to obtain the proposed First Forward Gaussian Differential Formula

$m0 = c0$……5

$\Delta y0 = y1 - y0$……6

Algorithm 2 First Forward Gaussian Differential Formula

Input: ASCII values
Output: Ciphertext

```
1   : var ascii_values = Encoding.ASCII.GetBytes(message);
2   :var initialEncryptedValues = 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
```

8. RSA ALGORITHM

RSA was proposed in 1977 and is the patent of Ron Rivest, Adi Shamir, and Len Adleman, which was published in 1978 at the Massachusetts Institute of Technology [25]. RSA as a public cryptographic scheme per literature is known to have a lot of weaknesses and also with higher execution time [25]. Hence the effort by researchers to propose 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   :          //var squareDiff = (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
```

9. GAUSSIAN BACKWARD INTERPOLATION

STATIC INITIAL CIPHERTEXT VALUE

$d0 = p0$……..7

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

$Yp = y0 + p\Delta y0 + \frac{p(p-1)\Delta 2y-1}{2!} + ((p+1)p(p-1)3! + \frac{(p+1)p(p-1)(p-2)\Delta 4y-2}{4!} + \cdots$……..8

It can be deduced from $eq1$ that $\Delta 2y - 1 = \Delta y0 - \Delta y - 1$……..……9

From $equation\ 2$ it can be deduced that

$\equiv \Delta y0 = \Delta y1 + \Delta 2y - 1$……..10

This suggests that $\Delta 2y0 = \Delta 2y - 1 + y3 - 1$……..……11

$\Delta y0 = \Delta y3 - 1 + \Delta 4y - 1$ … … … … … … … … … … … … … …12

Hence the Gaussian Backward Interpolation Formula can be written as;

$d0 = p0$……..13

$\Delta y0 = \Delta y3 - 1 + \Delta 4y - 1$…….…….... 14

Algorithm 2 First Backward Gaussian Differential Formula

Input: Decrypted values
Output: Plaintext

```
1  :  Begin
2  :  var plainArr = new List<byte>();
3  :        var startIndex = 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
```

10. SETUP FOR THE EXPERIMENTAL WORK

For operative justification of the piece of the anticipated system, the model is carried out using five diverse projected hybrid schemes to compare the level of security as well as the execution

times of the various algorithms. The following are the specifications for the computer system used for the simulation of the hybrid algorithm. A computer with an NVidia GTX 1650Ti Intel i7 CPU with 16 GB memory and a GPU OF 8GB.

11. RESULTS AND DISCUSSION FOR THE EXPERIMENTAL SETUP

This section discusses the setup for the experimental work to assess the proposed hybrid algorithm, the output of the proposed algorithm as well as the comparison of the security strength and the execution time with other schemes.

12. EXPLORATORY OBSERVATION OF THE PROPOSED HYBRID ALGORITHM

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 ie. n = p * q
n acquired. n = 811
Applying Difference of squares $\emptyset$(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 d Compute d = e^{-1} mod $\emptyset$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

$$Decrypt = C^d \ \% \ \text{n}$$

101990249715101
STAGE VI: Apply Gaussian Backward Differential information on
10111099114121112116

13. RESULTS OF THE PROPOSED ALGORITHM

The hybrid algorithm is implemented employing C# language. The security strength and the execution metrics is compared with [26], [27], [28]. The proposed algorithm is an integration of the Gaussian Interpolation formula with Traditional RSA. This raises the encryption and

decryption degree to the fifth level. Table 1, depicts the comparison of the scrambling and unscrambling time of three algorithms, RSA, 2-KEY Pair, SRNN, and the proposed algorithm.

Table 1: Comparing the encryption and decryption time of four algorithms

	Encryption Time (ms)				Decryption Time (ms)			
Text Size	RSA[28]	2-Key[25] Size	SRNN[26]	Proposed	RSA[28]	2-Key Size [25]	SRNN[26]	Proposed
1KB	12	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

From table 1, 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 is higher than the 2-key size and SRNN but lower than traditional RSA algorithms. Again the decryption time for the proposed algorithm is higher than RSA and SRNN but lower than the 2-Key size algorithm. This is 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 2 and 3.

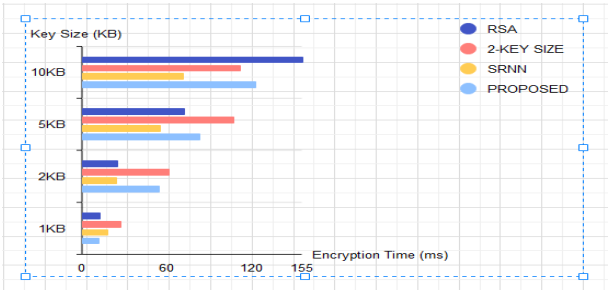

Figure 2: Scrambling Time (ms)

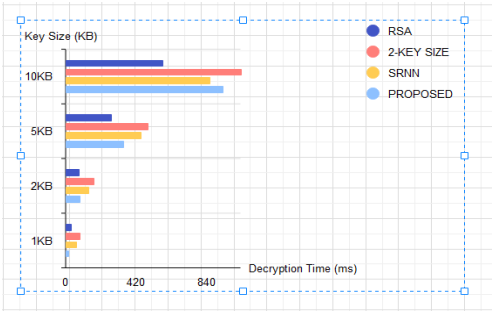

Figure 3: Unscrambling Time (ms)

14. THE PROS OF THE HYBRID ALGORITHM

The hybrid algorithm's security strength is based on the conversion of the alphabets in the plaintext to its ASCII values and applying First Forward Gaussian Interpolation Formula on the ASCII values. This helps to make it difficult for any intruder to fish out the plaintext being sent unto the cloud. The result is then acted upon using the traditional RSA scheme. The RSA scheme consists of three stages Generation of Keys, Encryption, and Decryption of the data. Another robustness of the hybrid algorithm depends on the second decryption process performed on the decryption from

the RSA decoded output. This is attained through the use of the Gaussian Backward Interpolation formula.

15. CONCLUSION

This paper developed an enhanced RSA scheme by the integration of the 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.

FUTURE WORKS

This proposed algorithm has a stronger security strength than the traditional RSA but it would be appropriate that future works are done to compare execution metrics of the hybrid algorithm and the traditional RSA on different machines with higher specifications.

REFERENCES

[1]M. Raut and P. Itkar, "Provable Data Possession at Untrusted Cloud Storage server", *International Journal of Engineering And Computer Science*, 2016. Available: 10.18535/ijecs/v5i2.20.

[2]V. Basili and H. Rombach, "The TAME project: towards improvement-oriented software environments", *IEEE Transactions on Software Engineering*, vol. 14, no. 6, pp. 758-773, 1988. Available: 10.1109/32.6156.

[3]U. Vazirani and T. Vidick, "Fully device independent quantum key distribution", *Communications of the ACM*, vol. 62, no. 4, pp. 133-133, 2019. Available: 10.1145/3310974.

[4] P. Rewagad and Y. Pawar, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing," *2013 International Conference on Communication Systems and Network Technologies*, 2013, pp. 437-439, doi: 10.1109/CSNT.2013.97.

[5]A. Rawat, K. Sehgal, A. Tiwari, A. Sharma, and A. Joshi, "A novel accelerated implementation of RSA using parallel processing", *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 2, pp. 309-322, 2019. Available: 10.1080/09720529.2019.1582864.

[6]V. Rayward-Smith, T. Cormen, C. Leiserson, and R. Rivest, "Introduction to Algorithms", *The Journal of the Operational Research Society*, vol. 42, no. 9, p. 816, 1991. Available: 10.2307/2583667.

[7]N. Kartha, "Review of the algorithm design manual, second edition by Steven S. Skiena", *ACM SIGACT News*, vol. 42, no. 4, pp. 29-31, 2011. Available: 10.1145/2078162.2078169.

[8]H. K. Khan, R. Pradhan and B. R. Chandavarkar, "Hybrid Cryptography for Cloud Computing," *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456210.

[9]É. Gouzien and N. Sangouard, "Factoring 2048-bit RSA Integers in 177 Days with 13 436 Qubits and a Multimode Memory", *Physical Review Letters*, vol. 127, no. 14, 2021. Available: 10.1103/physrevlett.127.140503.

[9]S. Aboud, M. AL-Fayoumi, M. Al-Fayoumi and H. Jabbar, "An Efficient RSA Public Key Encryption Scheme", *Academia.edu*, 2022. [Online]. Available: https://www.academia.edu/5008239/An_Efficient_RSA_Public_Key_Encryption_Scheme. [Accessed: 10- Jan- 2022].

[10]Y. Wazery, S. Gamal and A. Amin, "A Hybrid Technique based on RSA and Data Hiding for Securing Handwritten Signature", *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, 2021. Available: 10.14569/ijacsa.2021.0120489.

[11]X. Lin, L. Sun and H. Qu, "An efficient RSA-based certificateless public key encryption scheme", *Discrete Applied Mathematics*, vol. 241, pp. 39-47, 2018. Available: 10.1016/j.dam.2017.02.019.

[12]M. Budiman, P. Sihombing and I. Fikri, "A cryptocompression system with Multi-Factor RSA algorithm and Levenstein code algorithm", *Journal of Physics: Conference Series*, vol. 1898, no. 1, p. 012040, 2021. Available: 10.1088/1742-6596/1898/1/012040.

[13]M. Ariffin, S. Abubakar, F. Yunos and M. Asbullah, "New Cryptanalytic Attack on RSA Modulus N=pq Using Small Prime Difference Method", *Cryptography*, vol. 3, no. 1, p. 2, 2018. Available: 10.3390/cryptography3010002.

[14] V. P. Bansal and S. Singh, "A hybrid data encryption technique using RSA and Blowfish for cloud computing on FPGAs," 2015 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), 2015, pp. 1-5, doi: 10.1109/RAECS.2015.7453367.

[15] S. Mittal, S. Arora and R. Jain, "PData security using RSA encryption combined with image steganography," *2016 1st India International Conference on Information Processing (IICIP)*, 2016, pp. 1-5, doi: 10.1109/IICIP.2016.7975347.

[16]I. Amalarethinam and H. Leena, "Enhanced RSA Algorithm with Varying Key Sizes for Data Security in Cloud", *2017 World Congress on Computing and Communication Technologies (WCCCT)*, 2017. Available: 10.1109/wccct.2016.50 [Accessed 10 January 2022].

[17]P. Kaliyamoorthy and A. Ramalingam, "QMLFD Based RSA Cryptosystem for Enhancing Data Security in Public Cloud Storage System", *Wireless Personal Communications*, vol. 122, no. 1, pp. 755-782, 2021. Available: 10.1007/s11277-021-08924-z.

[18]P. Bharathi, G. Annam, J. B. Kandi, V. K. Duggana and A. T., "Secure File Storage using Hybrid Cryptography," *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, 2021, pp. 1-6, doi: 10.1109/ICCES51350.2021.9489026.

[19]H. K. Khan, R. Pradhan and B. R. Chandavarkar, "Hybrid Cryptography for Cloud Computing," *2021 2nd International Conference for Emerging Technology (INCET)*, 2021, pp. 1-5, doi: 10.1109/INCET51464.2021.9456210.

[20] C. A. Subasini and S. Nikkath Bushra, "Securing of Cloud Data with Duplex Data Encryption Algorithm," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 252-256, doi: 10.1109/ICCMC51019.2021.9418247.

[21]B. Mondol and M. A. Mahmood, "An Efficient Approach for Multiple User Data Security in Cloud Computing," *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 1130-1135, doi: 10.1109/ICAIS50930.2021.9395815.

[22]V. Rajkumar, M. Prakash and V. Vennila, "Secure Data Sharing with Confidentiality, Integrity and Access Control in Cloud Environment", *Computer Systems Science and Engineering*, vol. 40, no. 2, pp. 779-793, 2022. Available: 10.32604/csse.2022.019622.

[23]M. Uddin, Md. Kowsher and Mir Md. Moheuddin, "A New Method Of Central Difference Interpolation", *Applied Mathematics and Sciences An International Journal (MathSJ)*, vol. 6, no. 3, pp. 01-14, 2019. Available: 10.5121/mathsj.2019.6301.

[24]K. Balasubramanian, "Variants of RSA and their cryptanalysis," *2014 International Conference on Communication and Network Technologies*, 2014, pp. 145-149, doi: 10.1109/CNT.2014.7062742.

[25] S. Y. Bonde and U. S. Bhadade, “Analysis of encryption algorithms (RSA, SRNN and 2 key pair) for information security,” *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017.

[26] M. S. devi, “Threshold SR2N public key cryptosystem,” *International Journal of Engineering Trends and Technology*, vol. 31, no. 1, pp. 15–17, 2016.

[27] P.K.Panda and S. Chattopadhyay, “A hybrid security algorithm for RSA cryptosystem,” in 2017 4th International *Conference on Advance Computing and Communication Systems* (ICACCS), 2017.

[28] R.Rivest, A. Shamir and L.Adleman, “A method for obtaining digital signatures and public key cryptosystem,” Communications of ACM vol.21 (2), pp. 120-126, 1978.

[29] A. Karakra and A. Alsadeh, "A-RSA: Augmented RSA," 2016 SAI Computing Conference (SAI), 2016, pp. 1016-1023, doi: 10.1109/SAI.2016.7556103.

[30] E. Rutkowski and S. Houghten, "Cryptanalysis of RSA: Integer Prime Factorization Using Genetic Algorithms," 2020 IEEE Congress on Evolutionary Computation (CEC), 2020, pp. 1-8, doi: 10.1109/CEC48606.2020.9185728.

Funding: There is no funding support from any organization or entity.
Availability of supporting data: All documents needed for better explanations of the articles have all been inserted in the manuscript.
Competing interests: There is no competing interest with any organization
Acknowledgment: We would like to acknowledge the hard work of all the authors and the works of other authors that have been duly referenced.
Authors' contributions: The abstract and the introduction were jointly written by Dawson and Twum. The literature review was written by Prof Acquah. The mathematical component was written by all the authors and typeset by Dawson.