An enhanced RSA algorithm using Gaussian interpolation formula

John Kwao Dawson*

Sunyani Technical University,
P.O. Box 206, Sunyani-Bono Region, Ghana
Email: kwaodawson@yahoo.com
*Corresponding author

Frimpong Twum,
James Benjamin Hayfron-Acquah and
Yaw Marfo Missah

KNUST,
Kumasi, Ghana
Email: twumf@yahoo.co.uk
Email: jbha@yahoo.com
Email: ymissah@gmail.com

Ben Beklisi Kwame Ayawli

Sunyani Technical University,
P.O. Box 206, Sunyani-Bono Region, Ghana
Email: 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 factorisation problem of the traditional RSA. Comparative analysis was performed using four different algorithms: RSA, SRNN, two-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; ASCII values; Gaussian first forward interpolation formula; GFIF; cryptographic algorithm; RSA; hybrid algorithm.

Reference to this paper should be made as follows: Dawson, J.K., Twum, F., Hayfron-Acquah, J.B., Missah, Y.M. and Ayawli, B.B.K. (2022) ‘An enhanced RSA algorithm using Gaussian interpolation formula’, *Int. J. Computer Aided Engineering and Technology*, Vol. 16, No. 4, pp.534–552.

Biographical notes: John Kwao Dawson is a PhD candidate in Computer Science at the Kwame Nkrumah University of Science and Technology. He holds a Master of Philosophy in Information Technology and Bachelor’s in Information Technology from the Kwame Nkrumah University of Science and Technology and University of Education Winneba, respectively. His area of research is cloud computing, algorithm design, machine learning, artificial intelligence and data and network security.

Frimpong Twum received his BSc (hons.) in Electrical and Electronic Engineering and MSc in Internet and Multimedia Engineering from the London South Bank University, in 2004 and 2007, respectively. He also received his MSc in Information System from the Roehampton University, London, in 2011. He completed his PhD in Computer Science from the KNUST, Ghana, in 2017 with a specialisation in Computer Security. He is currently a Senior Lecturer at the Department of Computer Science, KNUST.

James Benjamin Hayfron-Acquah is the Head of the Department of Computer Science. He had his first degree in Computer Science from the Kwame Nkrumah University of Science and Technology (KNUST), in 1991. He then proceeded to have his Master’s in Computer Science and Application at the Shanghai University of Science and Technology (SUST), Shanghai, China, in 1996. He obtained his PhD from the Southampton University in the UK, in 2003. He joined the KNUST in March 1996 as a Lecturer. He was promoted to a Senior Lecturer in 2004 and Associate Professor in 2015.

Yaw Marfo Missah is a Lecturer in the Computer Science Department of Kwame Nkrumah University of Science and Technology. He obtained his PhD in Computer Science (DCS) in Enterprise Information System in 2013, Master of Science (MSIT) in Information Technology in 2004, and Bachelor of Science (BSc) in Computer Science in 2000.

Ben Beklisi Kwame Ayawli is a Lecturer in the Computer Science Department of Sunyani Technical University. He has been with the university since 2008. He holds a PhD in Power Engineering Automation from the Nanjing Tech University, China in 2019 and MSc in Information Technology obtained from Sikkim Manipal University, India, in 2011. He also obtained his BEd in Information Technology from the University of Education, Winneba, in 2008.

1 Introduction

The provision of protocol and processes necessary to secure a communication channel when an assumed third party exists is considered cryptography (Raut and Itkar, 2016). 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 (Basili and Rombach, 1988; Vazirani and Vidick, 2019) which ensures secured communication as well as data security. A counter algorithm termed RSA which has lower time complexity based on prime number factorisation 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 (Rewagad and Pawar, 2013). 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 as plaintext is scrambled at a time using bits based on the same key (Rawat et al., 2019). 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 (Rayward-Smith et al., 1991). It can be deduced that the complexity of RSA is based on computing e which is a combined modulus p (Gaussian Forward Interpolation Formula, https://www.mathworks.com/matlabcentral/fileexchange/42741-gaussian-forward-interpolation-formula). This determines the condition that, upon the selection of modulus p with an open key e which guarantees for every value c (0, 1, …, $p - 1$), just one t (0, 1, …, $t - 1$) such that

$$c = p^e \bmod n \tag{1}$$

This suggests that if an attacker has access to plaintext and the n^{th} modulus or the value for e, there can be a compromise by factoring n (Rutkowski and Houghten, 2020). It can be said that cracking the cipher is by factoring integers (Gaussian Forward Interpolation Formula, https://www.mathworks.com/matlabcentral/fileexchange/42741-gaussian-forward-interpolation-formula; Rutkowski and Houghten, 2020). 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 factorisation of n (Kartha, 2011). 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 propose 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 factorisation 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 summarised form as presented in this section.

Aboud et al. (2022) and Balasubramanian (2014) proposed a modified RSA which uses linear order with chosen integer values with an n^{th} modulus similar to the RSA algorithm.

Also, Wazery et al. (2021) 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 re-establish the data.

According to Lin et al. (2018), 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. (2021), which employed a multi-factor RSA scheme. Their scheme worked based on the Agrawal-Biswas scheme which scrambles the data and finally unscrambles the ciphertext.

Budiman et al.'s (2021) work was further improved by Ariffin et al. (2018) 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 (2015), 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 et al. (2016), a hybrid algorithm that integrates blowfish and RSA algorithm was proposed. This technique serves both symmetric and asymmetric purposes which makes it efficient.

Amalarethinam and Leena (2017) 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 (2021) 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. (2021). 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 Rajkumar et al. (2022) 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 to overcome 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 Subasini and Bushra (2021).

Mondol and Mahmood (2021) proposed a hybrid algorithm utilising 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 (2021) 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 (2021) 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. Given 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 Khan et al. (2021).

3 Methodology

This section presents our proposed methodology of integrating traditional RSA and Gaussian interpolation formulas to strengthen data security. The Gaussian forward and backward interpolations are integrated with the traditional RSA algorithm to enhance security strength by addressing the factorisation problem of RSA as indicated by Rutkowski and Houghten (2020). The approach involves third degree of encryption and second degree level of decryption.

3.1 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 (Bonde and Bhadade, 2017). RSA as a public cryptographic scheme per literature is known to have a lot of weaknesses and also with higher execution time (Bonde and Bhadade, 2017). Hence, the effort by researchers to propose variant RSA to raise its security strength while reducing execution time. In RSA, the selection of the primes determines the security strength of the algorithm. If the prime numbers selected are large the security strength is high likewise the execution time but if the selected prime numbers are smaller the execution time is low likewise the security strength (Bonde and Bhadade, 2017).

Traditional RSA considers three phases:

- generation of key
- scrambling
- unscrambling.

3.2 Proposed algorithm

The proposed algorithm is an integration of Gaussian Interpolation and the traditional RSA algorithm. This integration helps to raise the security strength of the proposed algorithm to a fifth degree by making it stream cipher and heterogeneous.

3.3 Proposed framework

Figure 2 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 then applied to the ASCII values which give it first encryption strength. RSA is then applied to the ciphertext results. This involves the generation of keys using the randomly selected prime numbers to get the private and public keys. This is then used to encrypt the data based on the private key and the decryption of the ciphertext using the public keys generated. Apply Gaussian backward interpolation on the decrypted values from RSA to obtain the plaintext.

Figure 1 Flowchart for traditional RSA cryptosystem

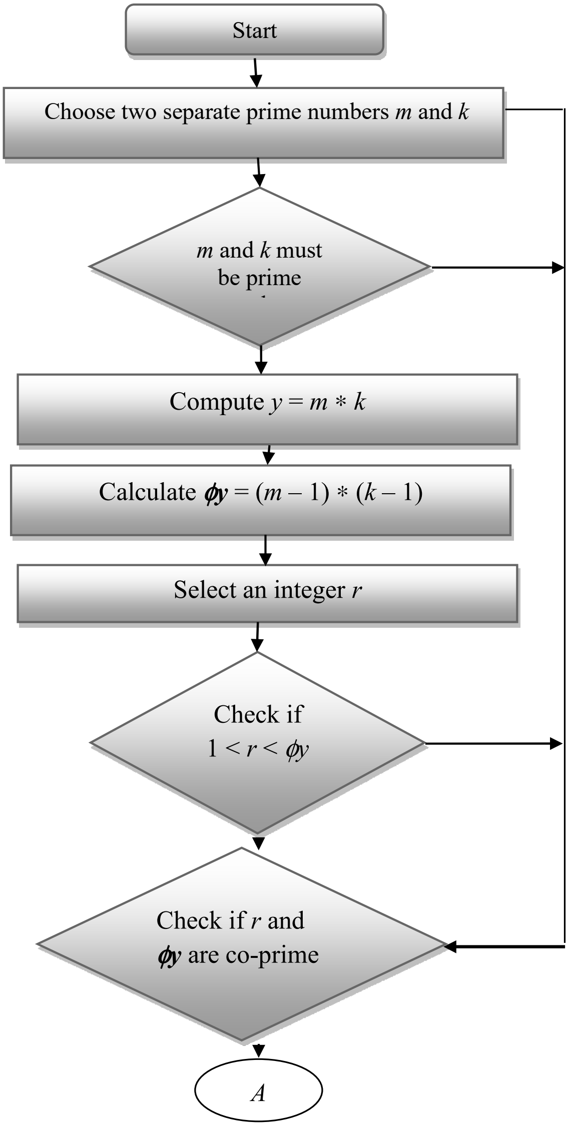

Start
Choose two separate prime numbers *m* and *k*
m and *k* must be prime
Compute $y = m * k$
Calculate $\phi y = (m - 1) * (k - 1)$
Select an integer *r*
Check if $1 < r < \phi y$
Check if *r* and ϕy are co-prime
A

Figure 1 Flowchart for traditional RSA cryptosystem (continued)

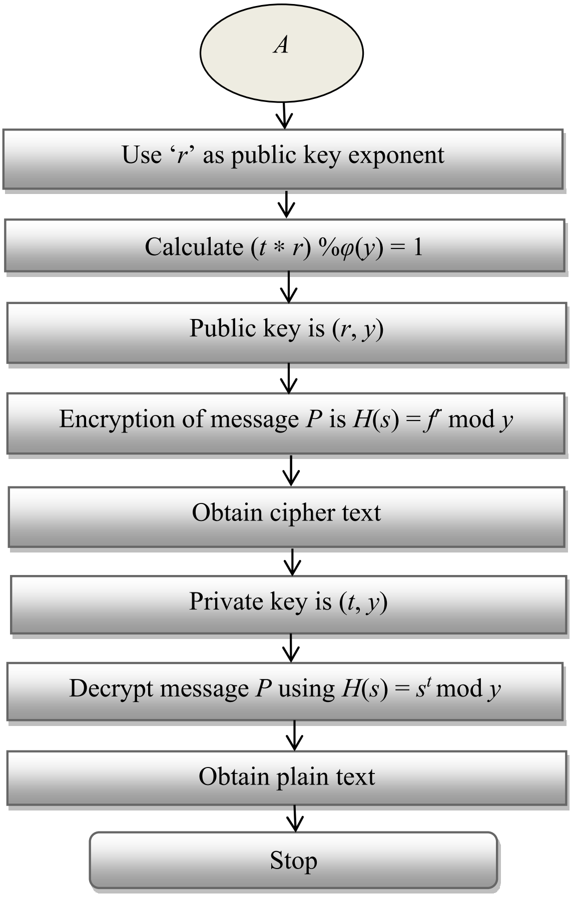
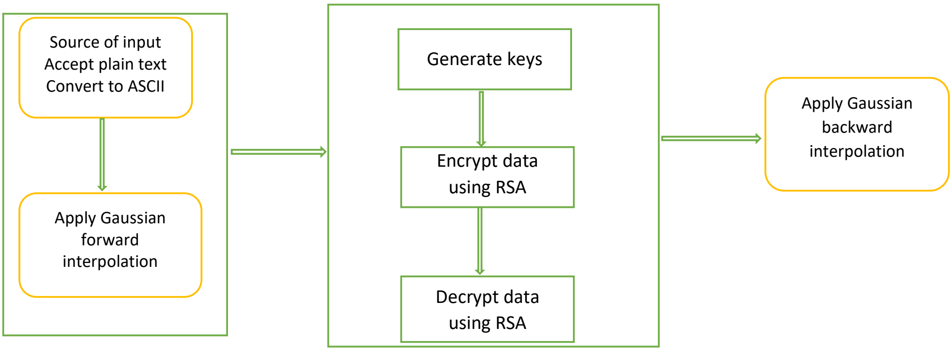

A
Use 'r' as public key exponent
Calculate $(t * r)\ \%\varphi(y) = 1$
Public key is (r, y)
Encryption of message P is $H(s) = f^r \bmod y$
Obtain cipher text
Private key is (t, y)
Decrypt message P using $H(s) = s^t \bmod y$
Obtain plain text
Stop

Figure 2 Workflow diagram of the proposed algorithm (see online version for colours)



3.4 Gaussian interpolation formula

Interpolation can be defined as the creation of new data points when a set of data is given. Gaussian interpolation considers central difference interpolation formula (Gaussian Forward Interpolation Formula, https://www.mathworks.com/matlabcentral/fileexchange/42741-gaussian-forward-interpolation-formula).

For instance, considering the formula $y = f(x)$ for values of x:

$$X : x0x1x2\ldots\ldots\ldots xn \tag{2}$$

$$Y : y0y1y2\ldots\ldots\ldots yn \tag{3}$$

The procedure for calculating the corresponding values for y regarding x variables is given as a range between $x = xi$ and $x0$ and xn is known as interpolation. Thus, interpolation is used to assume the functional values for intermediate units of independent values whiles considering the functions outside the assumed range known as extrapolation (Gaussian Forward Interpolation Formula, https://www.mathworks.com/matlabcentral/fileexchange/42741-gaussian-forward-interpolation-formula).

$$Yt = y0 + t.\Delta y0 + \frac{t(t-1)\Delta 2y0}{1.2} + t(t-1)(t-2)\Delta 3y0 / (1.2.3 + \cdots) \tag{4}$$

From this, we can deduce that

$$\Delta 3y - 1 = \Delta 2y0 - \Delta 2y - 1 \tag{5}$$

Again,

$$\Delta 3y0 = \Delta 3y - 1 + \Delta 4y - 1 \tag{6}$$

$$\Delta 4y0 = \Delta 4y - 1 + \Delta 5y - 1 \tag{7}$$

$$\Delta 3y - 1 = \Delta 3y - 2 + \Delta 4y - 2 \tag{8}$$

From equations (5), (6), (7) and (8) substituting into equation (4), the new formula becomes:

$$\Delta 2y0,\ \Delta 3y0,\ \Delta 4y0$$

$$\begin{aligned} Yt = {} & y0 + t\Delta y0 + \frac{t(t-1).(\Delta 2y - 1 + \Delta 3y - 1)}{1.2} + \left(p.\frac{(t-1)\left(t - 2(\Delta 3y - 1)\right)}{1.2.3} \right) \\ & + \left(t.(t-1)(t-2)(t-3)(\Delta 4y - 1 + \Delta 5y - 1)\right) \big/ (1.2.3.4 + \cdots) \end{aligned} \tag{9}$$

Then

$$\begin{aligned} Yt = {} & y0 + t\Delta y0 + \frac{t(t-1)\Delta 2y - 1}{2!} + \left[\left((t+1)t(t-1)3! \right) \right] \\ & + \frac{(t+1)t(t-1)(t-2)\Delta 4y - 2}{4!} + \cdots. \end{aligned} \tag{10}$$

3.5 Proposed algorithm

3.5.1 Phase 1: converting alphabets into ASCII values

The alphabets in the message are converted into their corresponding ASCII values.

3.5.2 Phase 2: first Gaussian forward difference

This is now considered as Gaussian first forward interpolation formula (GFIF). In this work, the first level Gaussian interpolation formula will be considered.

The central difference interpolation formula considers Gaussian interpolation as a subunit. For example, when the value $y = f(x)$ is given for x values

Then x: $x0, x1, x2, \ldots\ldots, xk$.

Then y: $y0, y1, y2, \ldots\ldots, yk$.

Table 1 Gaussian first forward difference interpolation

$x-2$	$y-2$				
		$\Delta y-1$			
$x-1$	$y-1$		$\Delta^2 y0$		
		$\Delta y0$		$\Delta^3 y1$	
$x0$	$y0$		$\Delta^2 y1$		$\Delta^4 y0$
		$\Delta y1$		$\Delta^3 y2$	
$x1$	$Y1$		$\Delta^2 y2$		
		$\Delta y2$			
$x2$	$Y2$				

Source: Uddin et al. (2019)

Therefore, the first forward difference formula is

$$\Delta y-1=\left(y-2\left(-(y-1)\right)\right) \tag{11}$$

The central differences, are deduced as $y1-y0, y2-y1, y3-y2, \ldots\ldots\ldots, yk-yk$.

This can be denoted as $\Delta y0$, $\Delta y1$, $\Delta y2$, $\Delta y3$, ………, $\Delta yk-1$ which gives the first forward difference.

From this, it can be deduced that the first forward difference will be

$$\Delta y0 = y1 - y0 \tag{12}$$

In this, formula Yp is the initial value for the ASCII value for the first letter in the message to be encrypted and is kept constant $y0$ = em. This value gives the first estimation for the message em and for the next alphabets to be encrypted as shown in equation (12).

3.5.3 Phase 3: generation of key

- Distinct two prime numbers are chosen m and k.
- Compute $y = m * k$.
- Compute $\varphi(y) = (m-1) * (k-1)$.

- Select r with the end goal such that $1 < r < \varphi(y)$ also r as well as r must be co-prime numbers.
- Calculate a quality for 't' with the end goal that $(t * r) \% \varphi(y) = 1$.
- The resulting public and private keys to help in the scrambling and unscrambling are:
  - a the public key is (r, y)
  - b the private key is (t, y).

3.5.4 Phase 4: scrambling

For any original data P, the encryption capacity is given as:

$$P(s) = f_r \,(\bmod\, y).$$

3.5.5 Phase 5: unscrambling

In this manner for any encrypted ciphertext C, the decryption capacity is

$$H(s) = s^t \,(\bmod\, y).$$

3.5.6 Phase 6: Gaussian backward interpolation

From the Gaussian forward interpolation formula

$$Yp = y0 + p\Delta y0 + \frac{p(p-1)\Delta 2y-1}{2!} + \left((p+1)p(p-1)3!+ \frac{(p+1)p(p-1)(p-2)\Delta 4y-2}{4!} \right) + \ldots \tag{13}$$

It can be deduced that $\Delta 2y - 1 = \Delta y0 - \Delta y - 1$

$$\equiv \Delta y0 = \Delta y1 + \Delta 2y - 1 \tag{14}$$

This suggests that

$$\Delta 2y0 = \Delta 2y - 1 + y3 - 1 \tag{15}$$

$$\Delta y0 = \Delta y3 - 1 + \Delta 4y - 1 \tag{16}$$

Replacing the variables $\Delta y0$, $\Delta 2y0$, $\Delta 3y - 1$ from equation (10) in equation (5), we can deduce that

$$f(x) \equiv y0 + t(\Delta y - 1 + \Delta 2y - 1) + \frac{t(t-1)}{2!(\Delta 2y - 1 + \Delta 3y - 1)} + \frac{t(t-1)(p-2)}{3!(\Delta 3y - 1 + \Delta 4y - 1)} + \frac{t(t-1)(t-2)(t-3)}{4!(\Delta 4y - 1 + \Delta 5y - 1)} + \cdots \tag{17}$$

In considering the variables $\Delta y - 1$, $\Delta 2y - 1$, $\Delta 3 - 1$, $\Delta 4y - 1$, will result in

$$f(x) \equiv y0 + t\Delta y - 1 + \frac{(t-1)t}{2!\Delta 2y} - 1 + \frac{(t-1)t(t-2)}{3!\Delta 3y} - 1 + \frac{(t-1)t(t-1)(t-2)}{4!\Delta 4y} - 1 + \frac{t(t-1)(t-2)(t-3)}{5!\Delta 5y} - 1 + \cdots \tag{18}$$

This can also be represented as

$$\Delta 3y-1=\Delta 3y-2+\Delta 4y-2 \text{ and } \Delta 4y-1=\Delta 4y-2+\Delta 5y-1+\ldots \tag{19}$$

It can now be deduced that the Gaussian backward interpolation formula is

$$f(x)\equiv y0+t\Delta y-1+\frac{(t-1)t}{2!\Delta 2y}-1+(t-1)t(t-1)/3!(\Delta 3y-2+\Delta 4y-2)+\ldots \tag{20}$$

Therefore, considering the first level backward Gaussian interpolation formula

$$F(X)\equiv y0+\Delta y-1 \tag{21}$$

$$GBIF = y0+\Delta y-1 \tag{22}$$

$$w0 = y0 \tag{23}$$

In this formula, $w0$ is the initial value for the ASCII value for the first letter in the message to be decrypted is kept constant $w0 = y0$; this value gives the first estimation for the message $w0$ and for the next alphabets to be encrypted the formula

$$GBIF = y0+\Delta y0-1 \tag{24}$$

where 0 represent the cardinal location of the letters in the word and $p\Delta y$ is the subsequent alphabets in the words in the message.

Figure 3 Flowchart of proposed algorithm

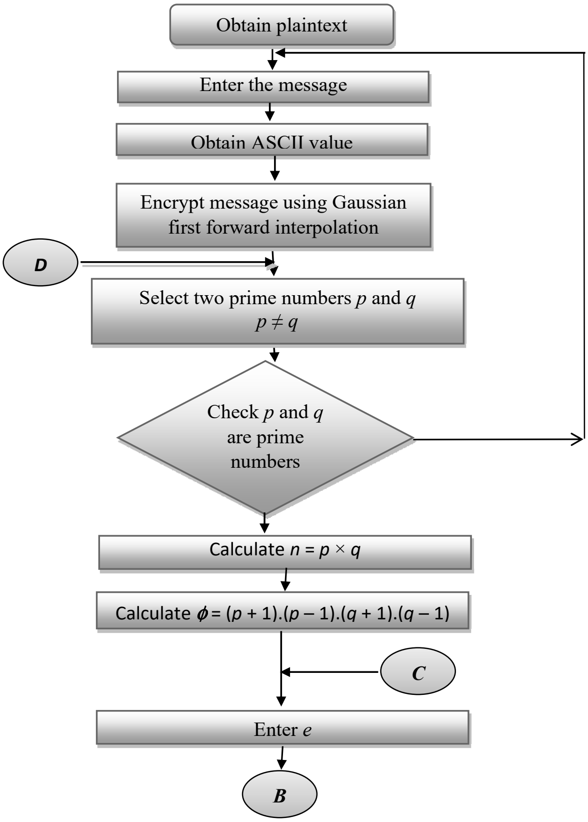
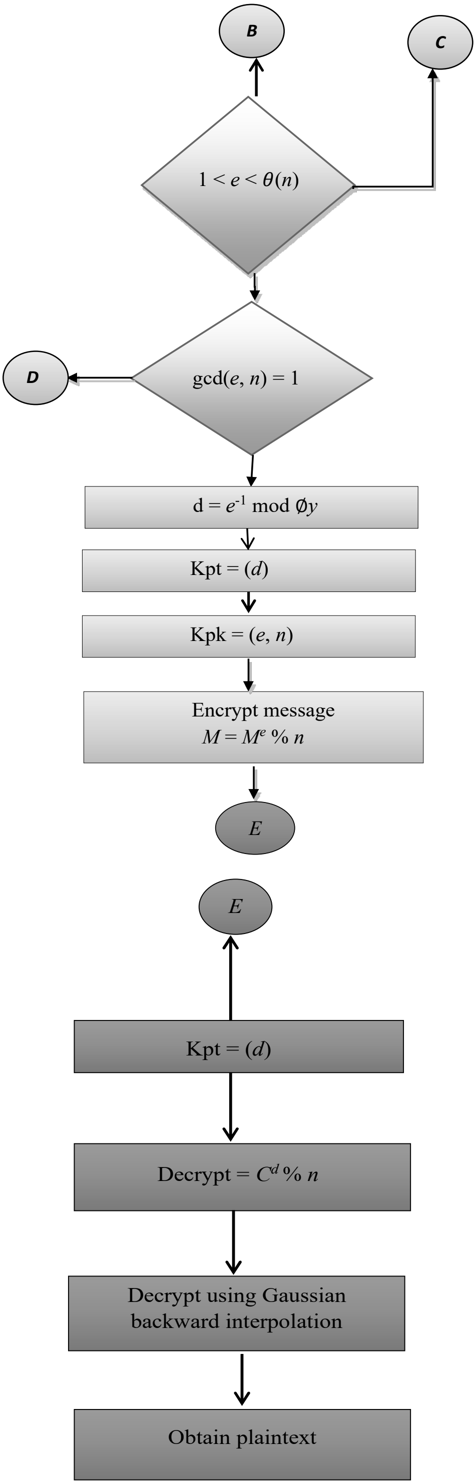

Obtain plaintext
Enter the message
Obtain ASCII value
Encrypt message using Gaussian first forward interpolation
D
Select two prime numbers p and q
p ≠ q
Check p and q are prime numbers
Calculate n = p × q
Calculate ϕ = (p + 1).(p – 1).(q + 1).(q – 1)
C
Enter e
B

Figure 3 Flowchart of proposed algorithm (continued)



4 Exploratory observation of the proposed hybrid algorithm

To be able to analyse the efficiency of the proposed algorithm, test data has been taken and completed using IntelliJ IDEA with java++.

Stage 1 Convert the message to be encrypted to its ASCII values

The message to be encoded is 'encrypt'

101110099114112116

Stage 2 Apply Gaussian first forward interpolation on ASCII values

$\Delta y0 = y1 - Y0$

101990249715101

Stage 3 Key generation

Choose two prime numbers p and $q, p \neq q$

$P = 811$

$Q = 281$

Computing n, i.e., $n = p * q$

n acquired. $n = 811$.

Applying difference of squares $\varnothing(n) = (p + 1) . (p - 1) . (q + 1) . (q - 1)$.

Compute PHI of n using formula $(p + 1) . (p - 1) . (q + 1) . (q - 1) = 433$.

Compute e such that $1 < e < \theta(n)$ and $\gcd(e, n) = 1$, i.e., $e = 62$.

Computing d, compute $d = e - 1 \bmod \varnothing n$, i.e., $d = = e^{-1} \bmod \theta(n)$ :: d is 0.016129032258064516.

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 4 Decrypt a message

$Decrypt = C^d \% n$

101990249715101

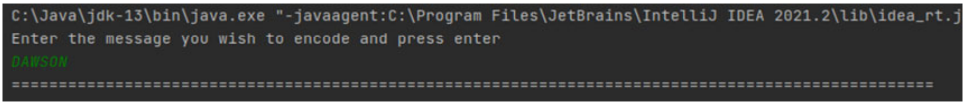
Stage 5 Apply Gaussian backward differential information on

1011199114121112116.

5 Experimentation results

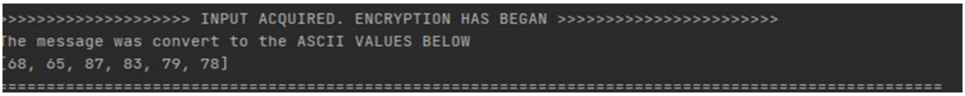
The experimental results of the proposed algorithm are depicted in Figures 4, 5, 6, 7, 8, 9, 10 and 11.

Figure 4 Plaintext (see online version for colours)



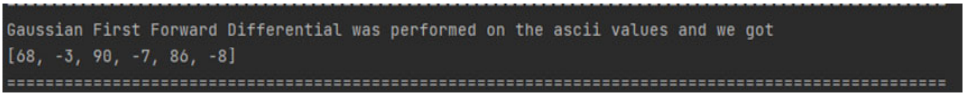
```
C:\Java\jdk-13\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.2\lib\idea_rt.j
Enter the message you wish to encode and press enter
DAWSON
==================================================================================================
```

Figure 5 ASCII values



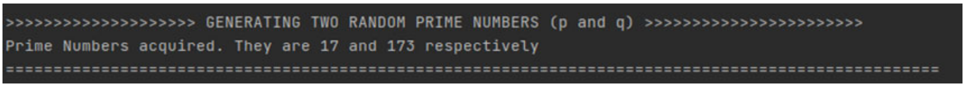
```
>>>>>>>>>>>>>>>>>>>> INPUT ACQUIRED. ENCRYPTION HAS BEGAN >>>>>>>>>>>>>>>>>>>>>>>>
he message was convert to the ASCII VALUES BELOW
68, 65, 87, 83, 79, 78]
==================================================================================================
```

Figure 6 Results of Gaussian first differentiation interpolation



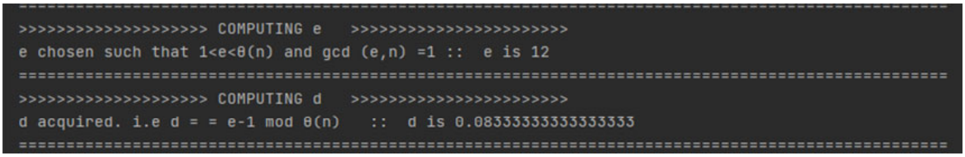
```
==================================================================================================
Gaussian First Forward Differential was performed on the ascii values and we got
[68, -3, 90, -7, 86, -8]
==================================================================================================
```

Figure 7 Key generation



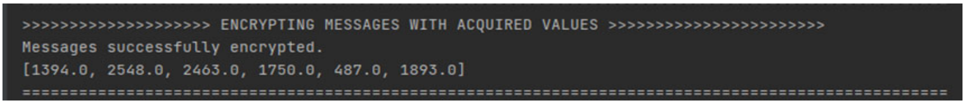
```
>>>>>>>>>>>>>>>>>>>>> GENERATING TWO RANDOM PRIME NUMBERS (p and q) >>>>>>>>>>>>>>>>>>>>>>>>>
Prime Numbers acquired. They are 17 and 173 respectively
==================================================================================================
```

Figure 8 Key generation



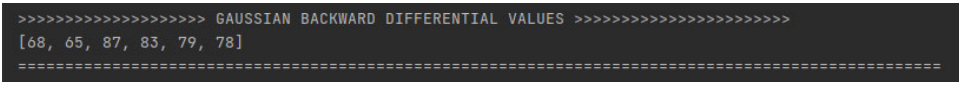
```
==================================================================================================
>>>>>>>>>>>>>>>>>>>>> COMPUTING e   >>>>>>>>>>>>>>>>>>>>>>>>
e chosen such that 1<e<θ(n) and gcd (e,n) =1 ::  e is 12
==================================================================================================
>>>>>>>>>>>>>>>>>>>>> COMPUTING d   >>>>>>>>>>>>>>>>>>>>>>>>
d acquired. i.e d = = e-1 mod θ(n)   ::  d is 0.08333333333333333
==================================================================================================
```

Figure 9 Encrypted message



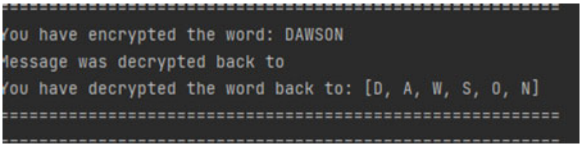
```
>>>>>>>>>>>>>>>>>>>>> ENCRYPTING MESSAGES WITH ACQUIRED VALUES >>>>>>>>>>>>>>>>>>>>>>>>>
Messages successfully encrypted.
[1394.0, 2548.0, 2463.0, 1750.0, 487.0, 1893.0]
==================================================================================================
```

Figure 10 Apply Gaussian backward interpolation



```
>>>>>>>>>>>>>>>>>>>>> GAUSSIAN BACKWARD DIFFERENTIAL VALUES >>>>>>>>>>>>>>>>>>>>>>>>>
[68, 65, 87, 83, 79, 78]
==================================================================================================
```

Figure 11 Decrypted message



```
========================================================
ou have encrypted the word: DAWSON
essage was decrypted back to
ou have decrypted the word back to: [D, A, W, S, O, N]
========================================================
========================================================
```

6 Results of the proposed algorithm

The hybrid algorithm is implemented employing java++. The security strength and the execution metrics are compared with Devi (2016), Panda and Chattopadhyay (2017) and Rivest et al. (1978). 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 2 depicts the comparison of the scrambling and unscrambling time of three algorithms, RSA, two-key pair, SRNN and the proposed algorithm.

Figure 12 Scrambling time (ms) (see online version for colours)

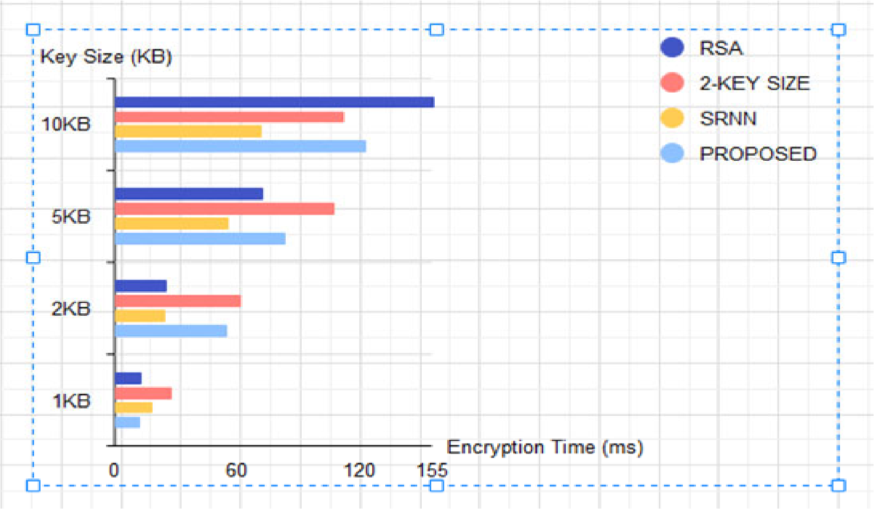


Figure 13 Unscrambling time (ms) (see online version for colours)

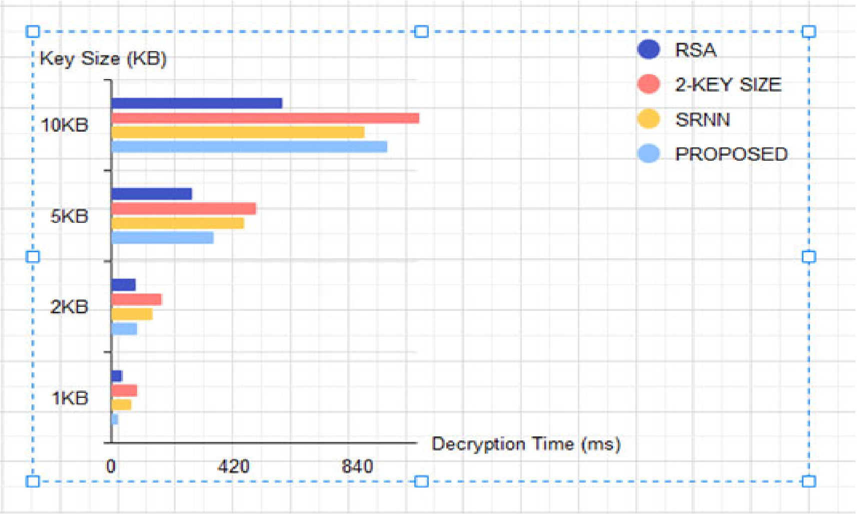


Table 2 Comparing the encryption and decryption time of four algorithms

Text size	*Encryption time (ms)*				*Decryption time (ms)*			
	RSA (Rivest et al., 1978)	*2-key (Bonde and Bhadade, 2017) size*	*SRNN (Devi, 2016)*	*Proposed*	*RSA (Rivest et al., 1978)*	*2-key size (Bonde and Bhadade, 2017)*	*SRNN (Devi, 2016)*	*Proposed*
1 KB	12	27	18	11.8	30	81	60	13.7
2 KB	25	61	24	54	76	167	133	82
5 KB	72	107	55	83	268	488	443	342
10 KB	155	111	71	122	573	1037	854	932

From Table 2, it can be inferred that the proposed algorithm's encryption and decryption time for data size of 1 KB is the lowest. On the other hand, as the text size increased to 10 KB, the encryption time is higher than the two-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 two-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 Figures 12 and 13.

7 Conclusions

This paper developed an ERSA 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.

8 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

Aboud, S., AL-Fayoumi, M., Al-Fayoumi, M. and Jabbar, H. (2022) *An Efficient RSA Public Key Encryption Scheme*, Academia.edu [online] https://www.academia.edu/5008239/An_Efficient_RSA_Public_Key_Encryption_Scheme (accessed 10 January 2022).

Amalarethinam, I. and Leena, H. (2017) 'Enhanced RSA algorithm with varying key sizes for data security in cloud', *2017 World Congress on Computing and Communication Technologies (WCCCT)*, DOI: 10.1109/wccct.2016.50 (accessed 10 January 2022).

Ariffin, M., Abubakar, S., Yunos, F. and Asbullah, M. (2018) 'New cryptanalytic attack on RSA modulus N=pq using small prime difference method', *Cryptography*, Vol. 3, No. 1, p.2, DOI: 10.3390/cryptography3010002.

Balasubramanian, K. (2014) 'Variants of RSA and their cryptanalysis', *2014 International Conference on Communication and Network Technologies*, pp.145–149, DOI: 10.1109/CNT.2014.7062742.

Bansal, V.P. and Singh, S. (2015) '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)*, pp.1–5, DOI: 10.1109/RAECS.2015.7453367.

Basili, V. and Rombach, H. (1988) 'The TAME Project: towards improvement-oriented software environments', *IEEE Transactions on Software Engineering*, Vol. 14, No. 6, pp.758–773, DOI: 10.1109/32.6156.

Bharathi, P., Annam, G., Kandi, J.B., Duggana, V.K. and Anjali, T. (2021) 'Secure file storage using hybrid cryptography', *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, pp.1–6, DOI: 10.1109/ICCES51350.2021.9489026.

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

Budiman, M., Sihombing, P. and Fikri, I. (2021) 'A cryptocompression system with multi-factor RSA algorithm and Levenstein code algorithm', *Journal of Physics: Conference Series*, Vol. 1898, No. 1, p.012040, DOI: 10.1088/1742-6596/1898/1/012040.

Devi, M.S. (2016) 'Threshold SR2N public-key cryptosystem', *International Journal of Engineering Trends and Technology*, Vol. 31, No. 1, pp.15–17.

Gaussian Forward Interpolation Formula [online] https://www.mathworks.com/matlabcentral/fileexchange/42741-gaussian-forward-interpolation-formula (accessed 14 February 2022).

Gouzien, É. and Sangouard, N. (2021) 'Factoring 2048-bit RSA Integers in 177 days with 13 436 qubits and a multimode memory', *Physical Review Letters*, Vol. 127, No. 14, DOI: 10.1103/physrevlett.127.140503.

Kaliyamoorthy, P. and Ramalingam, A. (2021) 'QMLFD based RSA cryptosystem for enhancing data security in public cloud storage system', *Wireless Personal Communications*, Vol. 122, No. 1, pp.755–782, DOI: 10.1007/s11277-021-08924-z.

Kartha, N. (2011) 'Review of the algorithm design manual, second edition by Steven S. Skiena', *ACM SIGACT News*, Vol. 42, No. 4, pp.29–31, DOI: 10.1145/2078162.2078169.

Khan, H.K., Pradhan, R. and Chandavarkar, B.R. (2021) 'Hybrid cryptography for cloud computing', *2021 2nd International Conference for Emerging Technology (INCET)*, pp.1–5, DOI: 10.1109/INCET51464.2021.9456210.

Lin, X., Sun, L. and Qu, H. (2018) 'An efficient RSA-based certificateless public-key encryption scheme', *Discrete Applied Mathematics*, Vol. 241, pp.39–47, DOI: 10.1016/j.dam.2017.02.019.

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

Mondol, B. and Mahmood, M.A. (2021) 'An efficient approach for multiple user data security in cloud computing', *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, pp.1130–1135, DOI: 10.1109/ICAIS50930.2021.9395815.

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

Rajkumar, V., Prakash, M. and Vennila, V. (2022) 'Secure data sharing with confidentiality, integrity and access control in cloud environment', *Computer Systems Science and Engineering*, Vol. 40, No. 2, pp.779–793, DOI: 10.32604/csse.2022.019622.

Raut, M. and Itkar, P. (2016) 'Provable data possession at untrusted cloud storage server', *International Journal of Engineering and Computer Science*, DOI: 10.18535/ijecs/v5i2.20.

Rawat, A., Sehgal, K., Tiwari, A., Sharma, A. and Joshi, A. (2019) 'A novel accelerated implementation of RSA using parallel processing', *Journal of Discrete Mathematical Sciences and Cryptography*, Vol. 22, No. 2, pp.309–322, DOI: 10.1080/09720529.2019.1582864.

Rayward-Smith, V., Cormen, T., Leiserson, C. and Rivest, R. (1991) 'Introduction to algorithms', *The Journal of the Operational Research Society*, Vol. 42, No. 9, p.816, DOI: 10.2307/2583667.

Rewagad, P. and Pawar, Y. (2013) '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*, pp.437–439, DOI: 10.1109/CSNT.2013.97.

Rivest, R., Shamir, A. and Adleman, L. (1978) 'A method for obtaining digital signatures and public-key cryptosystem', *Communications of ACM*, Vol. 21, No. 2, pp.120–126.

Rutkowski, E. and Houghten, S. (2020) 'Cryptanalysis of RSA: integer prime factorization using genetic algorithms', *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp.1–8, DOI: 10.1109/CEC48606.2020.9185728.

Subasini, C.A. and Bushra, S.N. (2021) 'Securing of cloud data with duplex data encryption algorithm', *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp.252–256, DOI: 10.1109/ICCMC51019.2021.9418247.

Uddin, M., Kowsher, M. and Moheuddin, M.M. (2019) 'A new method of central difference interpolation', *Applied Mathematics and Sciences an International Journal (MathSJ)*, Vol. 6, No. 3, pp.1–14, DOI: 10.5121/mathsj.2019.6301.

Vazirani, U. and Vidick, T. (2019) 'Fully device-independent quantum key distribution', *Communications of the ACM*, Vol. 62, No. 4, pp.133–133, DOI: 10.1145/3310974.

Wazery, Y., Gamal, S. and Amin, A. (2021) '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, DOI: 10.14569/ijacsa.2021.0120489.