SCIENTIST
Journal of Security, Computer, Information, Embedded, Network, and Intelligence System

P-ISSN : 3024-8701
E-ISSN : 3024-8329

(SCIENTIST) Journal of Security, Computer, Information,
Embedded, Network, and Intelligence System
Vol. 3, No. 1, 2025

A Computational Enhancement Of Base64 Algorithm Using Residue Number System Algorithms

***Logunleko Kolawole Bariu**
Department of Computer Science, D.S. Adegbenro ICT
Polytechnic, Eruku-Itori, Ewekoro, Ogun state, Nigeria.
kolawolelogunleko@gmail.com

Asaju-Gbolagade Ayisat Wuraola
Department of Computer Science, University of Ilorin,
Ilorin Nigeria.
ayisatwuraola@gmail.com

Logunleko Abolore Muhamin
Department of Computer Science, Gateway ICT
Polytechnic, Saapade, Ogun State, Nigeria.
aboloremlogunleko@gmail.com

Akinbowale Nathaniel Babatunde
Department of Computer Science, Kwara State University,
Malete, Ilorin, Nigeria.
akinbowale.babatunde@kwasu.ed.ng

Kazeem Alagbe Gbolagade
Department of Computer Science, Kwara State University, Malete, Ilorin, Nigeria.
kazeem.gbolagade@kwasu.edu.ng

ARTICLE INFO

Received : 12 December 2024
Accepted : 03 March 2025
Published : 01 June 2025

ABSTRACT

A significant part of information security has been played by cryptography techniques. Today's daily existence depends heavily on the advancement of data, information, and communication technologies. Consequently, there is a huge increase in the need for data and information. In communicating data over a public network, data security is a necessity that must be carefully taken into consideration. Thus, Base64 algorithm has been used in numerous security applications for ensuring data confidentiality, integrity and authentication. However, research shows that there is security vulnerabilities in most widely used Base64 algorithm due to the absence of key mechanism. To address this concern, this research employs residue number system (RNS) algorithms for the enhancement of base64 algorithm because of its cryptographic features whereby strengthen the transformation of the existing base64 algorithm to produce a novel symmetric-based cryptographic algorithm. The developed algorithm generates a symmetric key by shuffling the original key with the textual data, making the transformation of each character of the data better each time it is shuffled. Therefore, the research bridges the security gap in Base64 cryptographic algorithm by factoring key mechanism of RNS based algorithm into the newly developed algorithm. In addition, the developed symmetric-based cryptographic algorithm is more robust than the existing Base64 cryptographic algorithm because of the planned pattern and confusion produced during the methodology procedure thereby safeguards the data more effectively as shown in the cipher text generated.

Keywords : Base64 Algorithm, Cipher Key, RNS Algorithms, Xor, Encryption

I. INTRODUCTION

The usage increase of internet, the availability of digital data and the dissemination of data have made information security a crucial area for academics and professionals in information technology [1]. The development of information technology has brought about a number of important changes in people's lives, including a paradigm shift that gave rise to new forms of wealth and the reality that data is now replacing oil. [2]. As time goes by, people’s need for data and information are increasing tremendously therefore, the security of data, information and communication technology are currently essential part of everyday human life [3].

According to [4], privacy of data must be carefully taken into account before being sent or shared over a public network. Thus, users need to take necessary precautions by enforcing some protective measures on the information to be transmitted. For data security, encryption is the most significant automated technique [5]. Only a communication route shielded by encryption is protected from prying eyes attempting to steal private information. Furthermore, the information and communication technology era demands that the communication sector thrive since security is a crucial issue that needs to be resolved. In order to support data security measures in guaranteeing data privacy, secrecy and authentication, among other things, the Base64 cryptographic algorithm is employed. Several security applications have employed the Base64 algorithm to guarantee data integrity and authenticate the source of data but research found that the algorithm is not adequately secured due to absence of key mechanism. Moreover, the main goals in developing an algorithmic encryption and decryption process are to improve the security and efficiency of the base64 algorithm. Thus, this research integrates residue number system algorithms into base64 algorithm to produce a symmetric-based cryptographic algorithm called residue number system-base64 algorithm (RNS-B64) in order to ensure a better and more secured cryptosystem. The rest of the paper is structured as follows: related papers were discussed in section II. Section III focuses on the approach while Section IV and V are results and conclusion respectively.

II. RELATED WORK

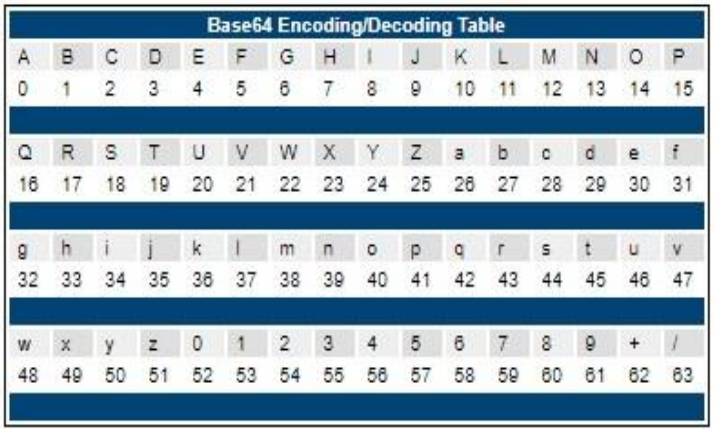
A number of studies [6, 7, 8] developed Base64 Algorithms (as stated in Algorithm1 and 2) and operation procedures in details as follows; the first stage dividing the three binary data bytes (24 bits) into four numbers of six bits each. Base64 utilizes 6 bits, i.e. $2^6 = 64 \text{ characters,}$ to guarantee that the encoded data is legible and does not employ any of the special characters present in ASCII, despite the ASCII standard requiring the usage of seven bits. These 64 ASCII characters are used in the base64 algorithm as shown in fig1.

Algorithm 1: **Encryption Algorithm**

```
bi = ascii(mi)
bi = bin(bi)
ci = split(bi, 6)
c = concat(c0, c1, …, cn)
ci = base64(ci)
ci = split(bi, 6)
ci = base64(ci)
```

Algorithm 2: **Decryption Algorithm**

```
ci = base64’(ci)
bi = join(ci)
bi = bin’(bi)
ci = split(bi, 6)
c = concat(c0, c1, …, cn)
ci = split(c, 8)
mi = ascii’(ci)
```



Base64 Encoding/Decoding Table															
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
w	x	y	z	0	1	2	3	4	5	6	7	8	9	+	/
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

Fig1: Base64 Encoding/Decoding Lookup Table Source: [6]

[3] applied Base64 algorithm to image files so as to ensure security of data from unauthorized users. The researcher revealed that Base64 was often used for simple encryption purposes such as hiding non-sensitive information and authentication processes. However, the study further pointed out that Base64 algorithm has security inadequacy mechanism because of no key technique. Therefore, the algorithms’ security needs to be improved so as to make the algorithm stronger for a symmetric based cryptography.

[9] did a research on enhancement of image security with RNS as DES is susceptible to brute force attack. After applying watermarking, the resulting model needs a secret key and RNS moduli set to retrieve the original image. By watermarking one image inside another, the watermarking technique was able to give an initial level of image protection, even confusing the original image for an impartial user. The computing time and memory space required to execute each of the algorithms under consideration were utilized to test and evaluate the model. The obtained results show that residual number did not only add another layer of security, but also helps the image to be processed faster during the encryption and decryption due to the fact that only a residual part of the image was worked upon rather than the whole image pixels.

[10] explained cryptography as the modern security protocol to protect the information from the outsiders and to communicate securely without the involvement of third parties. In addition to introducing a generic way of solvability series for a solvable problem that differs from others, the researcher suggested a few modified solvability propositions. Thus, a new basic numerical model of cryptography was developed based on solvability series.

[11] proposed a novel symmetric cryptography technique for the Residue Number System and its Modified Perfect Form. In the first technique, the ciphertext is considered as a collection of residues to the corresponding sets of modules (keys), and the Chinese Remainder Theorem was used to determine the decimal number recovery from its residues. The amount of arithmetic operations required for the decryption procedure is reduced as a result of the authours' use of a Modified Perfect Form of Residue Number System to streamline the computations. In a similar vein, where quick decryption is necessary, the study used a different symmetric encryption technique based on the Chinese Remainder Theorem. The Prime Number Theorem and the Euler Function were used to assess the security of the suggested techniques. The study examined the bitness and number of modules needed to provide the same level of security for the created symmetric security systems as the AES algorithm's longest length key. The study also, discovered that the bitness of the modules reduces with an increase in number.

[12] presented the design methods of information encryption and decryption using Residue Number System (RNS). The study selected three moduli set $\{2^n-1,\ 2^n, 2^n+1\}$ and design an effective forward conversion for the selected moduli set with 4n + 2 and 8n as delay and Area respectively, for the information encryption and reverse converter for the same moduli set with 4n + 3 and 8n + 3 as area and delay respectively, the proposed scheme outperforms the state of the art in terms of both security and computational efficiency.

[13] applied RNS and MPEG IV algorithm in video security. The work presents a cryptographic scheme for enhancing video transmission using a proposed modification of the traditional moduli set $\{2^n-1, 2^n, 2^n+1\}$.

[14] proposed encryption and decryption algorithm using algebraic matrix approach as an efficient data algorithm to protect the message with the help of key passed between sender and receiver. Utilizing the advantages of message splitting to the amount of words, data encryption lowers computational and storage overheads for the data owner. The ease of use and accessibility of the created model demonstrated that tools may be created without having to buy pricey software off the shelf.

The study revealed that Base64 algorithm has been used in numerous security applications for ensuring data confidentiality and integrity. However, Base64 transformation cryptography algorithm needed to be improved upon due to its non-availability of key. A cryptographic scheme's level of security is largely determined by the type and length of keys used, the degree of encryption used to create chaos, the algorithms' throughput rate, and their capacity to encrypt shorter messages [15]. Based on this, Base64 algorithm is therefore enhanced using Residue Number System due to its cryptographic features thereby creating a more confusion and transformation to the crypto algorithms to produce RNS-Base64 cryptosystem.

III. METHODS

This study develops an enhanced version of Base64 algorithm by factoring the key into the algorithm using residue number system (RNS) approach. The purpose of the developed algorithm is to boost the efficiency and performance of base64 algorithm current capabilities and also to increase the algorithm security level. The addition of the use of the restricted moduli set $\{\boldsymbol{2^n-1},\ \boldsymbol{2^n}, \boldsymbol{2^n+1}\}$ helps strengthen further the security of the Base64 algorithm to form RNS-B64 Algorithm.

A. RNS-Base64 Algorithm

The developed algorithm accepts inputs such as key, message and ciphertext which transformed to ASCII value. Thereafter, RNS was integrated into the ASCII value which was segmented into 8bits and transformed to Base64 Equivalent using Base64 Algorithm to form a symmetric encryption and decryption algorithms as shown in algorithm 3 and 4 respectively.

Algorithm 3: **Encryption Algorithm**

```
$k_i = ascii(k_i)$
$f_i = (k_i * len(k))/count(k_i))$
$d = xor(f_0, f_1, …, f_n)$
$m_i = rns(d)$
$m = xor(m_0, m_1, …, m_n)$
$b = bin(m)$
$m_i = ascii(m_i)$
$r_{ni}, r_{ni+1}, r_{ni+2} = rns(m_i)$
$r_i = bin(r_i)$
$r_i = xor(r_i, b)$
$r = concat(r_0, r_1, …, r_n)$
$c_i = r_{[6i:6i+6]}$
$ci = base64(c_i)$
```

Algorithm 4: **Decryption Algorithm**

```
$k_i = ascii(k_i)$
$f_i = (k_i * len(k))/count(k_i))$
$d = xor(f_0, f_1, …, f_n)$
$m_i = rns(d)$
$m = xor(m_0, m_1, …, m_n)$
$b = bin(m)$
$c_i = base64'(c_i)$
$c = concat(c_0, c_1, …, c_n)$
$r_i = c_{[8i:8i+8]}$
$r_i = xor(r_i, b)$
$r_i = dec(r_i)$
$m_i = crt(r_{ni}, r_{ni+1}, r_{ni+2})$
$m_i = ascii(m_i)$
```

IV. RESULT AND DISCUSSION

A. Base64 Algorithm Evaluation

Supplying the plaintext "**Names**" having five characters. The illustration in Table 1 shows the steps of the existing Base64 algorithmic process design.

Table 1. Base64 Algorithm Illustration

Index	1	2	3	4	5
char	N	a	m	e	s
Decimal	78	97	109	101	115

The computational evaluation process begins:

Index 1: N	ASCII: 78	Binary: 01001110
Index 2: a	ASCII: 97	Binary: 01100001
Index 3: m	ASCII: 109	Binary: 01101101
Index 4: e	ASCII: 101	Binary: 01100101
Index 5: s	ASCII : 115	Binary: 01110011

Putting the binaries together gives 0100111001100001011011010110010101110011000000000.

The combined length of the concatenated binaries is 40 bits, so in order to make it 48 bits, a multiple of 24 bits and 8-bit padding methods were required. Because of this, the combined binaries became 010011100110000101101101||01100101011100110000000.

Eight 6-bit binaries would be created from this concatenated 48-bit, one for each grouping.
010011||100110||000101||101101||011001||010111|| 001100||000000

Next, as seen below, each 6-bit binary will be transformed to a decimal value.

Table 2. Binary 6-bit

After reviewing the decimal equivalent values for each index in the base64 lookup table, it is found that the plaintext "Names" converted into the ciphertext TmFtZXMA. As A denotes zero (0), it was substituted with the "=" sign, ultimately resulting in **TmFtZXM=** as the ciphertext.

From the computational analysis above, the amount of the plain text character is five (5), this result to a total of forty eight (48) bits emanating from 6×8 bits as a result of 8-bit padding mechanism. Thus, the 48-bits later grouped into 6-bits characters to produce cipher text of eight (8) characters.

Index	1	2	3	4	5	6	**7**	8
Binary 6-bit	010 011	100 110	000 101	101 101	0 11 001	010 111	001 100	000 000
Decimal Value	19	38	5	45	25	23	12	0

B. RNS-Base64 Algorithm Evaluation

The RNS-Base64 Algorithm is evaluated computationally in this section. The section is carried out theoretically using moduli set $\{2^n - 1,\ 2^n, 2^n + 1\}$ for both the key introduced and the textual data. The section contains two parts; key and the message respectively.

Step 1: Key Generation

This involves key calculation which entails the computational technique of the key feature using RNS concept with the restricted moduli set of $\{2^n - 1,$ $2^n, 2^n + 1\}$. This is narrated in Table 3.

Table 3: Key Generation

Index	Char	Decimal	Weight	Key Function	Xor	Secret RNS	Xor	Binary
1	K	75	1	300	1916	$\{25,28,2\}$	7	0000 0111
2	o	111	1	444				
3	l	108	1	864				
4	a	97	1	1164				

Step 2: Encryption Generation.

Encryption Generation

The techniques of the plaintext "**Names**" begins. This contains the computational evaluation technique of encryption generation using RNS-Base64 Algorithm with restricted moduli set to generate six-bit binaries as shown in table 4.

Table 4: Transformation of Decimal Value to Binary Equivalent

Character	Decimal Value	Secret RNS	Binary
N	78	$\{16, 14, 12\}$	00010000 00001110 00001100
a	97	$\{4, 1, 31\}$	00000100 00000001 00011111
m	109	$\{16, 13, 10\}$	00010000 00001101 00001010
e	101	$\{8, 5, 2\}$	00001000 00000101 00000010
s	115	$\{22, 19, 16\}$	00010110 00010011 00010000

Step 3

Xoring the plaintext binary with the key:

Binary:

$$00010000 \oplus 00000111 = 00010111$$
$$00001110 \oplus 00000111 = 00001001$$
$$00001100 \oplus 00000111 = 00001011$$

Binary:

$$00000100 \oplus 00000111 = 00000011$$
$$00000001 \oplus 00000111 = 00000110$$
$$00011111 \oplus 00000111 = 00011000$$

Binary:

$$00010000 \oplus 00000111 = 00010111$$
$$00001101 \oplus 00000111 = 00001010$$
$$00001010 \oplus 00000111 = 00001101$$

Binary:

$$00001000 \oplus 00000111 = 00001111$$
$$00000101 \oplus 00000111 = 00000010$$
$$00000010 \oplus 00000111 = 00000101$$

Binary:

$$00010110 \oplus 00000111 = 00010001$$
$$00010011 \oplus 00000111 = 00010100$$
$$00010000 \oplus 00000111 = 00010111$$

Step 4

Concatenating the result binaries in step 3, the following binary sequences were generated:

00010111000010010000101100000011
00000110000110000001011100001010
00001101000011110000001000000101
000100010001010000010111

Step 5

Here, splitting Step 4 into six-bit binary, the following binary sequences were obtained:

000101 110000 100100 001011 000000
110000 011000 011000 000101 110000
101000 001101 000011 110000 001000
000101 000100 010001 010000 010111

Step 6

In this step, the Six-bit binary sequences obtained from Step 5, is then converted each to decimal value and base64 equivalent is generated respectively. This is demonstrated and summarized in table 5.

Table 5: Transformation of Binary Value to Base64 Equivalent

S/NO	1	2	3	4	5	6	7	8	9	10
BINARY	00 01 01	11 00 00	10 01 00	00 10 11	00 00 00	11 00 00	01 10 00	01 10 00	00 01 01	11 00 00
DECIMAL VALUE	5	48	36	11	0	48	24	24	5	48
BASE64 EQUIVALENT	F	w	k	L	A	w	Y	Y	F	w

S/NO	11	12	13	14	15	16	17	18	19	20
BINARY	10 10 00	00 11 01	00 00 11	11 00 00	00 10 00	00 01 01	00 01 00	01 00 01	01 00 00	01 01 11
DECIMAL VALUE	40	13	3	48	8	5	4	17	16	23
BASE64 EQUIVALENT	o	N	D	w	I	F	E	R	Q	X

Cypher text: FwkLAwYYFwoNDwIFERQX is generated from the RNS-B64 algorithms.

Decryption Generation

Step 1

Perform the key-generating procedure in Table 3 again.

Step 2

Using the matching 6-bit binary to replace "FwkLAwYYFwoNDwIFERQX"

000101 110000 100100 001011 000000
110000 011000 011000 000101 110000
101000 001101 000011 110000 001000
000101 000100 010001 010000 010111

Step 3

Combining the result binary in step 2:

000101110000100100001011000000
110000011000011000000101110000
101000001101000011110000001000
000101000100010001010000010111

Step 4

Division of the step 3 binary into eight bits:

00010111 00001001 00001011 00000011
00000110 00011000 00010111 00001010
00001101 00001111 00000010 00000101
00010001 00010100 00010111

Step 5

Xoring the cipher-text binary and the key binary respectively:

Binary:

$$00010111 \oplus 00000111 = 00010000$$
$$00001001 \oplus 00000111 = 00001110$$
$$00001011 \oplus 00000111 = 00001100$$

Binary:

$$00000011 \oplus 00000111 = 00000100$$
$$00000110 \oplus 00000111 = 00000001$$
$$00011000 \oplus 00000111 = 00011111$$

Binary:

$$00010111 \oplus 00000111 = 00010000$$
$$00001010 \oplus 00000111 = 00001101$$
$$00001101 \oplus 00000111 = 00001010$$

Binary:

$$00001111 \oplus 00000111 = 00001000$$
$$00000010 \oplus 00000111 = 00000101$$
$$00000101 \oplus 00000111 = 00000010$$

Binary:

$$00010001 \oplus 00000111 = 00010110$$
$$00010100 \oplus 00000111 = 00010011$$
$$00010111 \oplus 00000111 = 00010000$$

Step 6

The Six-bit binary sequences obtained from step 5, is then converted back using Chinese Reminder Theorem (CRT) to produce its decimal value, thereby generating the equivalent initial character. The computational method in table 6 is achieved as a result of backward conversion of RNS.

Table 6. Transformation of Binary Value to Character Equivalent

Binary	CRT (Secret RNS)	Decimal Value	Character
00010000 00001110 00001100	{16,14,12}	78	N
00000100 00000001 00011111	{4,1,31}	97	a
00010000 00001101 00001010	{16,13,10}	109	m
00001000 00000101 00000010	{8,5,2}	101	e
00010110 00010011 00010000	{22,19,16}	115	s

Consequently, the plaintext **‘**Names’ is obtained. From the evaluation, the amount of the plain text character is five (5), this result to a total of one hundred and twenty (120) bits emanating from 8×15 bits (i.e. the 120 bits are grouped into fifteen (15) sections of 8-bits characters). Therefore, the 120-bits later grouped into 2-bits characters to produce the non-intelligible text referred to cipher text of Twenty (20) characters. Finally, the equivalent character of each of the above decimal from the ASCII table is **Names** which was initial plaintext, as the final output of the decrypted process.

V. CONCLUSION

The research revealed the essential application usage of base64 algorithm in information security and also pointed out the security gap in the existing Base64 cryptographic algorithm as it is not adequately secured. Therefore, the study employed RNS because of its cryptographic nature and efficiency peculiarity to enhance base64 algorithm in order to produce a better symmetric key-based cryptographic algorithm. Furthermore, the addition of the use of the moduli set also strengthens further the security of the developed algorithm. The algorithm pattern established during the process of methodology procedure safeguards the data more effectively than the existing Base64 algorithm as it is observed in the cipher text generated. The developed RNS-Base64 algorithm produces a greater confusion and adequately secured in the hands of an adversary.

The conducted computational analysis comes to the conclusion that, the research closes the security gap in Base64 cryptographic algorithm as compared to the developed Residue Number System-Base64 cryptographic algorithms. Therefore, the developed RNS-B64 algorithm is theoretically and computationally secure, and is suitable for exploration in information security systems for both encryption and decryption.

REFERENCES

[1] J. Eseyin and K. . A. Gbolagade, “A Residue Number System Based Data Hiding Using Steganography and Cryptography,” *Kampala Int. Univ. KIU KIU J. Soc. Sci.*, vol. 5, no. 2, pp. 345–351, 2019.

[2] H. Agusta, “Keamanan dan Akses Data Pribadi Penerima Pinjaman Dalam Peer to Peer Lending di Indonesia,” *KRTHA BHAYANGKARA*, vol. 15, no. 1, pp. 11–38, May 2021, doi: 10.31599/krtha.v15i1.289.

[3] F. Baso, “Analysis and Utilization of the Base64 Algorithm for Image Encryption and Decryption Security in Web-Based Images,” *J. Secur. Comput. Inf. Embed. Netw. Intell. Syst.*, pp. 52–57, Dec. 2023, doi: 10.61220/scientist.v1i2.20233.

[4] K. B. Logunleko, O. D. Adeniji, A. M. Logunleko, and M. O. Odufowora, “Data Encryption Scheme Using An Enhanced Base64 Algorithm,” *Univ. Ib. J. Sci. Log. ICT Res. UIJSLICTR*, vol. Vol. 3, no. No. 1, pp. 16–24, Aug. 2020.

[5] M. Victor, D. D. W. Praveenraj, S. R, A. Alkhayyat, and A. Shakhzoda, “Cryptography: Advances in Secure Communication and Data Protection,” *E3S Web Conf.*, vol. 399, p. 07010, 2023, doi: 10.1051/e3sconf/202339907010.

[6] C. Sarath, Sugadev. S, S. Chandragandhi, Akshai Kannan, and Kamalesh M, “Secure Message Transmission Using Base 64 Algorithm,” *Int. Adv. Res. J. Sci. Eng. Technol.*, vol. 08, no. 04, pp. 52–55, 2021, doi: 10.17148/IARJSET.2021.8411.

[7] M. Efendi, V. Sihombing, and S. Parulian, “Implementation and Use of Base64 Algorithm in Video File Security,” *Sinkron*, vol. 7, p. 1, 2022, doi: https://doi.org/10.33395/sinkron.v7i1.11256.

[8] I. Sumartono, A. P. U. Siahaan, and Arpan, “Base64 Character Encoding and Decoding Modeling,” *Int. J. Recent Trends Eng. Res. IJRTER*, vol. 02, no. 12, pp. 63–68, 2016.

[9] B. Asaju, D. Popoola, and K. Gbolagade, “Enhancing Image Security Using Data Encryption Standard, Discrete Wavelet Tranfrom Watermarking, Residue Number System and Gaussian Filtering,” *Afr. Sch. J. Afr. Innov. Adv. Stud.*, vol. 25, no. 2, pp. 19–40, 2022.

[10] S. M. Naser, “Cryptography: From the Ancient History to now, It’s applications and a new complete numerical model,” *Int. J. Math. Stat. Stud.*, vol. 9, no. 3, pp. 11–30, 2021.

[11] Stanislaw Zawislak, Mykhailo Kasianchuk, Igor Iakymenko, and Daniel Jancarczyk, “Methods of crypto-stable symmetric encryption in the residual number system,” *Elsevier*, vol. 207, pp. 128–137, 2022.

[12] I. A. Aremu and K. A. Gbolagade, “Information encoding and decoding using Residue Number System for {22n -1, 22n, 22n+1} moduli sets,” *Int. J. Adv. Res. Comput. Eng. Technol. IJARCET*, vol. 6, no. 8, pp. 1260–1267, 2017.

[13] A. N. Babatunde and A. A. Oloyede, “Application of Residue Number Systems in enhancing the transmission of secured videos,” *Rev. Rom. Informatică Și Autom.*, vol. 31, no. 4, 2021, doi: 10.33436/v31i4y202108.

[14] K. Thiagarajan, P. Balasubramanian, J. Nagaraj, and J. Padmashree, “Encryption and decryption algorithm using algebraic matrix approach,” *J. Phys. Conf. Ser.*, vol. 1000, p. 012148, Apr. 2018, doi: 10.1088/1742-6596/1000/1/012148.

[15] D. R. Stinson, Maura. B. Paterson, and M. P. Paterson, *Cryptography : Theory and Practice*. in Forth Edition. CRC Press Taylor & Francis Group LLC, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton London New York, FL 33487-2742, 2019. [Online]. Available: https://taylorandfrancis.com/