Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1266~1271
e-ISSN: 2548-6861

Enhancing the Encryption Capabilities of the Generalization of the ElGamal Algorithm for Document Security

Imanuel Zega[1]*, Bagas Dwi Yulianto [2]**
* Sistem Informasi, Universitas Pignatelli Triputra
** Informatika, Universitas Pignatelli Triputra
nuelzega@upitra.ac.id[1], bagas19.yulianto@gmail.com [2]

Article Info

Article history:

Received 2025-06-03
Revised 2025-07-22
Accepted 2025-07-30

Keyword:

Algorithm efficiency,
Cryptography,
Digital document security,
Generalization of the ElGama,
Security.

ABSTRACT

The development of cryptographic algorithms that are efficient in terms of computation and resource usage, in addition to maintaining the confidentiality, integrity, and authentication of information, is driven by the growing need for digital document security. The generalization of the ElGamal, an expansion of the traditional ElGamal algorithm with more adaptable encryption features, is one algorithm with a lot of promise in this situation. The implementation of the technique of splitting the plaintext into three-digit blocks to lower the complexity of encryption per character and the use of large prime numbers to increase the algorithm's mathematical complexity are the two main ways that this study seeks to increase the algorithm's efficiency and security. It is anticipated that this method will speed up computation time and simplify the encryption process per character without compromising security. The overall findings demonstrate that, without compromising security, this method dramatically cuts down on computation time and ciphertext enlargement. Therefore, in the age of digital transformation, the findings of this study aid in the creation of contemporary cryptographic algorithms that are more flexible and effective and serve as a strategic guide when creating a strong digital data security system.

I. INTRODUCTION

One type of data or information that comes in the form of files is a document. All kinds of documents, including private letters, trade secrets, and official government data, are now kept digitally due to the growth of digital communication and sensitive data transactions in the modern era [1]. Digital documents are therefore more susceptible to hacking, theft, and misuse. Modern cryptography has evolved into a crucial basis for information security in the current digital era [2]. Thanks to robust encryption methods, modern cryptography enables the confidentiality and integrity of digital documents to be preserved even when they are kept in risky places like the internet [3]. Digital document security can be improved, and the risk of sensitive data theft or misuse decreased, by employing cryptographic algorithms like the Generalization of the ElGamal using block encryption [4]. Cryptographic algorithms are constantly evolving to meet emerging challenges in an effort to preserve the confidentiality, integrity, and authentication of data [5].

Data security is becoming more and more important, particularly for documents [6]. Documents contain important information, such as business, personal, and confidential data [7]. Only unsecured data can be accessed by unauthorized parties, which can cause losses for data owners [8]. Message security is a crucial aspect of data communication [9]. Insecure messages can be used by unauthorized parties to commit fraud, alter data, or disseminate false information [10].

Therefore, it is crucial to address delicate data security issues, particularly in the still-emerging modern digitalization era [11]. Investigating the development, improvement, and evolution of the digital ecosystem from an information technology standpoint is required to accomplish this goal, taking into account the requirement for a method known as cryptography to preserve data confidentiality [12].

Even though contemporary cryptography has attained a high degree of security, issues still exist with algorithm efficiency as well as cryptographic strength [13]. Algorithm performance becomes critical in many real-world applications, particularly in real-time systems and resource-constrained devices [14]. Delays, high energy consumption, and ciphertext enlargement can result from high computational complexity [15]. The creation of algorithms that can strike a balance between process efficiency and cryptographic strength is a significant problem that requires methodical attention.

The science or art of delivering data to the recipient while concealing it from the sender while preserving its authenticity and integrity is known as cryptography [16]. One significant cryptographic algorithm that has been shown to be secure is the ElGamal algorithm. Taher ElGamal developed this algorithm in 1985, and its most well-known uses are in secret key exchange, public key encryption, and digital signatures [17].

The most widely used asymmetric cryptography algorithm is ElGamal, which encrypts and decrypts messages or information using different digital keys [18]. ElGamal, on the other hand, depends on the intricacy of discrete logarithm computations [19]. The generalization of the ElGamal is a new version of the ElGamal algorithm that was developed in 2021 for an improved encryption process scheme that uses the discrete logarithm problem in addition to prime factorization of plaintext [20]. Like all cryptographic systems, the generalization of the ElGamal must be improved further to meet ever-more-difficult security challenges.

Thus, by employing large prime numbers and the method of splitting the plaintext into blocks with three digits, this study aimed to enhance the security feature of the generalization of the ElGamal algorithm. It is anticipated that this method will yield data on the growth in the ciphertext's size as well as the approximate time needed for document encryption and decryption. It is anticipated that the findings of this study will not only advance the science of cryptography but also serve as a helpful guide for attempts to increase the efficacy and efficiency of currently used encryption algorithms.

II. METHOD

A. *Fermat's Theorem*

A positive integer with exactly two components 1 and the number itself is called a prime number [21]. In cryptography, public and private keys are computed using prime numbers. The difficulty of breaking down large integers into their constituent factors has a significant impact on their strength. Using Fermat's theory is one method for determining whether a number is prime

Formally, Fermat's theorem states that a prime number p can be primed if p is prime and a is not a multiple of p, and $1 < a < p$ [22]. Thus, it can be denoted in t by the equation $a^{p-1} \equiv 1 \, mod \, p \, ; 1 < a < p$, or it can also be written as $a^{p-1} mod \, p \equiv 1 \, ; 1 < a < p$.

The steps in Fermat's theory are as follows.

1. Take a random *integer* $[1, n-1]$
2. Calculate $y \equiv a^{p-1} (mod \, p)$
3. If $y \neq 1$, then output ("not prime")

B. *Generalization of the ElGamal Algorithm*

The generalization of the ElGamal algorithm was formulated to augment the robustness of earlier encryption algorithms, yielding substantial advancements in efficiency and security. This algorithm, grounded in the fundamental principles of discrete logarithms, also tackles the challenges of prime factorization related to plaintext processing in cryptography. Consequently, the generalization of the ElGamal algorithm provides a more resilient solution for data security in the contemporary digital era.

1) *Encryption by dividing the plaintext into blocks of numbers*

The encryption procedure in the Generalization of ElGamal initiates by transforming the plaintext into numeric values that correspond to the ASCII values of each character within the text. The main objective of this transformation is to convert the character-based data into a numeric representation suitable for processing by the cryptographic system. Subsequent to acquiring the numeric representation, the plaintext is partitioned into blocks of three-digit numbers. This method enhances encryption security while simultaneously increasing the efficiency of encryption and decryption processes, thereby diminishing the computational complexity necessary for managing extensive data sets. For instance, for a message comprising the text "ABCDE," the initial step is to convert each character to its ASCII value: A=65, B=66, C=67, D=68, E=69. This yields a sequence of ASCII values: 65, 66, 67, 68, 69. The subsequent step involves merging two adjacent values to create blocks of three-digit numbers. This method diminishes intricacy in the encryption procedure and enables the system to manage larger data sets effectively.

- Merge the ASCII values of A (65) and B (66) to create 656.
- Merge the ASCII values of B (66) and C (67) to create 667.
- Merge the ASCII values of D (68) and E (69) to create 686.
- The residual value of E (69) will be represented solely as 9.

The plaintext "ABCDE" is segmented into the blocks: 656, 667, 686, and 9.

2) *Reducing Complexity in the Encryption Process*

One direct way to lower encryption complexity is by breaking the plaintext up into blocks. In the absence of this division, the encryption system has to process the entire message at once, which can lengthen computation time, particularly for lengthy messages. The message can be divided into smaller blocks so that the system can process each one independently. A predefined public key can be used to independently encrypt each block. Because fewer calculations are needed for each block, the algorithm is more

effective at processing massive volumes of data, which speeds up the encryption and decryption process overall.For instance, the complete plaintext is processed in its entirety in classical cryptography algorithms, which frequently leads to a large computational overhead [23]. Block division in the Generalization of the ElGamal algorithm reduces the time needed to encrypt and decrypt larger messages and enables parallel processing. This is one of the reasons this approach efficiently cuts down on computation time, particularly when encrypting large amounts of data or character-rich messages.

3) Expansion of Ciphertext

This plaintext partitioning technique decreases computation time; however, it results in an increase in ciphertext size (ciphertext enlargement). The encryption of each plaintext block independently results in ciphertext that is generally larger than that produced by other encryption methods, such as RSA or AES, which may yield more compact ciphertext. This alteration arises as each block generates an individual output, each encompassing data that requires transmission or storage. The augmentation of ciphertext size also confers security advantages [24]. Segmenting the plaintext into smaller blocks obscures data patterns, rendering analysis more challenging. Each block, individually encrypted with a public key, enhances data confidentiality. Moreover, despite the augmented ciphertext size, enhanced security is frequently prioritized in numerous applications managing sensitive information. This method enhances protection against cryptanalysis assaults, including ciphertext-only and chosen-plaintext attacks, which may exploit patterns in smaller or unsegmented ciphertext. Consequently, although the ciphertext size increases, segmenting the plaintext into 3-digit blocks continues to provide substantial benefits regarding security and system efficiency. Enhanced processing efficiency and accelerated computational speeds facilitate swifter encryption and decryption, rendering it more applicable for real-world scenarios, including digital communications and the storage of sensitive data in cloud systems. Segmenting the plaintext into 3-digit blocks diminishes the probability of pattern-based attacks and bolsters data confidentiality, integrity, and authentication.

The phases in the Generalization of the ElGamal algorithm encompass three processes: key generation, encryption, and decryption.

4) Key Generation

1. Choose a prime number p, with the condition that $p > 255$
2. Choose a primitive root number g modulo p with the condition that $g > p$
3. Choose a private key x, with the condition that 2 to $p-2$
4. Calculate the public key

5) Encryption

1. Express the message in plaintext blocks.

$$m \equiv {p_1}^{a1} \cdot {p_2}^{a2} \cdots {p_i}^{ai}$$

2. Choose a random number i, with the condition 2 to $p-2$
3. Choose a private encryption key y, with the condition $1 < iy < p-1$
4. Calculate the values $d, b,$ and c.

$$d \equiv g^i \ mod\ p$$
$$b \equiv d^y \ mod\ p$$
$$c \equiv m.a^{iy} \ mod\ p$$

5. Give the message's recipient $b,$ and c.

6) Decryption

1. Compute b^x to determine a^{iy} without the sender's private key

$$b^x = (d^y)^x \equiv (g^i)^{xy} (mod\ p)$$
$$\equiv (g^x)^{iy} (mod\ p)$$
$$\equiv a^{iy} (mod\ p)$$

2. Get the value of m by the equation

$$m \equiv c \cdot (b^x)(mod\ p)$$

where (b^x) is the inverse value of $b^x \ mod\ p$

3. Arrange the plaintext with $b1, b2, \cdots, bn$

As part of the implementation of the cryptographic algorithm used in this study, the steps involved in key generation, message encryption, and message decryption are described below.

III. Result And Discussion

This section will go over how to use large prime numbers, specifically 200 digits, to divide plaintext into blocks with three digits. To obtain the private and public keys, the implementation starts by splitting the plaintext value into blocks. Next, we will generate prime numbers. After that, the encryption and decryption processes are completed.

For example, as Table 1 below illustrates, a message that contains the word "SECURITY" is transformed into a value in the ASCII table, and the plaintext value is then divided into blocks with three digits.

TABLE I
Decimal Value Of Each Character

S	E	C	U	R	I	T	Y
83	69	67	85	82	73	84	89

Each character's decimal value will then be separated into blocks, as shown in table 2 below.

TABLE II
Decimal Value Of Each Character

$m1$	$m2$	$m3$	$m4$	$m5$	$m6$
836	967	858	273	848	9

1) Generate Key

1. Determine the value of p, g, x

$$p = 1009$$
$$g = 503$$
$$x = 721$$

2. Calculate the value of a

$$a = 503^{721}\ mod\ 1009$$
$$= 503$$

After determining the value of p, g, x and calculating the value of a, we get:
Public key = $(p, g, a) = (1009, 503, 503)$
Private key = $(x) = (721)$

2) Message encryption

TABLE III
ENCRYPTION PROCESS

Plaintext	*Encryption*		*Ciphertext*	
	$b \equiv d^y\ (mod\ p)$	$c \equiv m.(a^{iy} mod\ p) mod\ p$	b	c
$M_1 = 836$	$634^8 mod\ 1009$	$836(503^{12.8}) mod\ 1009$	374	883
$M_2 = 969$	$1008^9 mod\ 1009$	$40(503^{7.9}) mod\ 1009$	1008	40
$M_3 = 858$	$337^{20} mod\ 1009$	$572(503^{13.20}) mod\ 1009$	337	572
$M_4 = 273$	$374^y mod\ 1009$	$193(503^{112.6}) mod\ 1009$	374	193
$M_5 = 848$	$634^{12} mod\ 1009$	$844(503^{79.12}) mod\ 1009$	634	844
$M_6 = 9$	$561^{17} mod\ 1009$	$4(503^{56.17}) mod\ 1009$	561	4

So the ciphertext obtained by combining the plaintext block values is 374, 883. 1008, 40. 337, 572. 374, 193. 634, 844. 561, 4.

3) Message Decrytion

TABLE IV
DECRYPTION PROCESS

Ciphertext	Decryption		Plaintext
	$z = b^x (mod\ p)$	$w = z^{-1}\ (mod\ p)$	$m = c\ .w\ (mod\ p)$
374, 883	$z = 374^{721} (mod\ 1009) = 374$	$w = 374^{-1}\ (mod\ 1009) = 634$	$m = 883\ .634\ (mod\ 1009) = 836$
1008, 40	$z = 1008^{721} (mod\ 1009) = 1008$	$w = 1008^{-1}\ (mod\ 1009) = 1008$	$m = 40\ .1008\ (mod\ 1009) = 969$
337, 572	$z = 337^{721} (mod\ 1009) = 337$	$w = 337^{-1}\ (mod\ 1009) = 506$	$m = 572\ .506\ (mod\ 1009) = 858$
374, 193	$z = 374^{721} (mod\ 1009) = 374$	$w = 374^{-1}\ (mod\ 1009) = 634$	$m = 193\ .634\ (mod\ 1009) = 273$
634, 844	$z = 634^{721} (mod\ 1009) = 634$	$w = 634^{-1}\ (mod\ 1009) = 374$	$m = 844\ .374\ (mod\ 1009) = 848$
561, 4	$z = 561^{721} (mod\ 1009) = 561$	$w = 561^{-1}\ (mod\ 1009) = 759$	$m = 4\ .759\ (mod\ 1009) = 9$

So the decryption result is 8369698582738489.

Table 5 below shows the size of the ciphertext and the time needed for the encryption and decryption process using 200-digit prime numbers after the message value has been divided into multiple blocks.

TABLE V
DIFFERENCES IN CIPHERTEXT ENLARGEMENT SIZE

Number of characters	**Original data size**	**Generalization of the ElGamal (original)**	**Generalization of the ElGamal (block technique)**
20	20 bytes	7,91 KB	2,80 KB
200	200 bytes	78,4 KB	26,3 KB
500	500 bytes	100 KB	65,3 KB

The ciphertext is enlarged in both algorithms, as can be observed from the test results displayed in the above table.

Nevertheless, it appears smaller in the generalization of the Elgamal algorithm, which employs the method of splitting the plaintext into blocks.

TABLE VI
DIFFERENCE IN ENCRYPTION AND DECRYPTION PROCESS TIME

Number of characters	Encryption		Decryption	
	Generalization of the ElGamal	**Generalization of the ElGamal**	**Generalization of the ElGamal**	**Generalization of the ElGamal**
	(original)	(block technique)	(original)	(block technique)
20	0,109214783 second	0,039031982 second	0,08749795 second	0,02823591 second
200	2,168078661 second	0,328340769 second	0,878899813 second	0,21689367 second
500	2,917479515 second	0,858770371 second	1,488327503 second	0,50265765 second

The time disparity between the encryption and decryption processes presented in Table 6 demonstrates that the method of segmenting the plaintext into blocks, in conjunction with the utilization of 200-digit prime numbers, yields a more efficient and expedited encryption and decryption process compared to the method lacking block segmentation. Refer to the following figure for a clearer comprehension of the temporal disparity between the encryption and decryption phases.

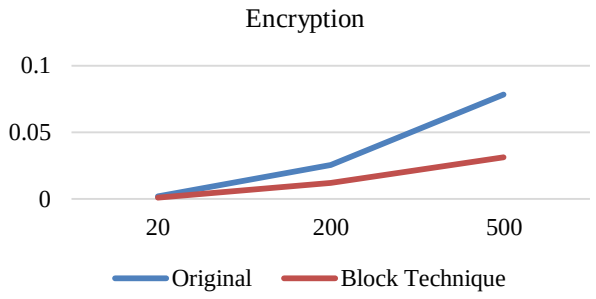

Figure 1. Duration of Encryption Computation

A comparison of the computational time between the conventional Generalization of ElGamal method and the block-division technique for plaintext is presented. The findings indicate that block encryption is significantly more efficient and rapid compared to the conventional method, which does not segment the plaintext into blocks. The block method facilitates expedited processing by segmenting the plaintext into smaller, more manageable units utilizing 200-digit prime numbers.

Figure 2 illustrates a comparison of decryption computation durations. The decryption process utilizing the block technique is markedly more rapid, akin to encryption. This is attributable to the more straightforward and efficient decryption process for smaller blocks, in contrast to the decryption employed by the standard Generalization of ElGamal, which necessitates greater processing time.

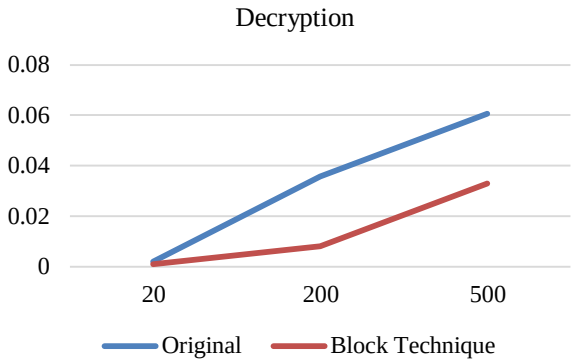

Figure 2. Decryption Processing Duration

IV. CONCLUSION

The application of the technique of dividing the plaintext into blocks, each consisting of three digits, has been proven to significantly increase efficiency, reduce computation time, and strengthen the security aspect of the Generalization of the ElGamal algorithm. This approach effectively reduces the complexity of the encryption process per character. In addition, the use of Fermat's Theorem has been proven effective in producing large prime numbers up to 200 digits, which are important cryptographic components in the encryption process. This block technique approach confirms that dividing the plaintext value into blocks can produce a faster encryption process and also contributes to reducing the enlargement of the ciphertext, making it a feasible and efficient solution to meet the needs of data security in today's digital era.

ACKNOWLEDGMENTS

We gratefully acknowledge that this research is funded by Universitas Pignatelli Triputra. The support is under the research grand UPITRA of Year 2024 Contract Number 14.126/LPPM-UPITRA/V/2024.

BIBLIOGRAPHY

[1] S. Kanojia, "Digitalization in Corporations: Integrating Utility of Digital Technology With Accessibility and Privacy of Data," 2024, pp. 227–245.

[2] S. Alotaibi, K. Alharbi, B. Abaalkhail, and D. M. Ibrahim, "Sensitive Data Exposure: Data Forwarding and Storage on Cloud Environment," *Int. J. online Biomed. Eng.*, vol. 17, no. 14, pp. 4–18, 2021, doi: 10.3991/IJOE.V17I14.27365.

[3] A. Bhandari, M. K. Ojha, D. K. Choubey, and V. Soni, "Research Article in Special Issue : Selected Papers from the 4th International Conference on Machine Learning , Image Processing , Network Security and Data IoT Based System for Accident Detection , Monitoring and Landslide Detection Using GSM in Hilly Ar," pp. 104–111, 2023.

[4] R. Hazra, P. Chatterjee, Y. Singh, G. Podder, and T. Das, "Data Encryption and Secure Communication Protocols," 2024, pp. 546–570.

[5] K. Raut, "A Comprehensive Review of Cryptographic Algorithms," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 12, pp. 1750–1756, 2021, doi: 10.22214/ijraset.2021.39581.

[6] P. Sharma, S. Namasudra, R. Gonzalez Crespo, J. Parra-Fuente, and M. Chandra Trivedi, "EHDHE: Enhancing security of healthcare documents in IoT-enabled digital healthcare ecosystems using blockchain," *Inf. Sci. (Ny).*, vol. 629, pp. 703–718, Jun. 2023, doi: 10.1016/j.ins.2023.01.148.

[7] S. Bevz, O. Tereshchuk, O. Kravchuk, V. Yehorova, I. Bodnarchuk, and M. Danevych, "Confidential information and the right to freedom of speech," *Int. J. Criminol. Sociol.*, vol. 10, pp. 648–651, 2021, doi: 10.6000/1929-4409.2021.10.75.

[8] A. Razaque, N. Shaldanbayeva, B. Alotaibi, M. Alotaibi, A. Murat, and A. Alotaibi, "Big data handling approach for unauthorized cloud computing access," *Electron.*, vol. 11, no. 1, pp. 1–20, 2022, doi: 10.3390/electronics11010137.

[9] Y. S. Santoso, "Message Security Using a Combination of Hill Cipher and RSA Algorithms," *J. Mat. Dan Ilmu Pengetah. Alam LLDikti Wil. 1*, vol. 1, no. 1, pp. 20–28, 2021, doi: 10.54076/jumpa.v1i1.38.

[10] H. N. Fakhouri, S. Alawadi, F. M. Awaysheh, F. Hamad, S. Alzubi, and M. N. AlAdwan, "An Overview of using of Artificial Intelligence in Enhancing Security and Privacy in Mobile Social Networks," in *2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC)*, Sep. 2023, pp. 42–51, doi: 10.1109/FMEC59375.2023.10305886.

[11] P. M. T. Untawale, "Importance of Cyber Security in Digital Era," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 8, pp. 963–966, 2021, doi: 10.22214/ijraset.2021.37519.

[12] R. Zahid *et al.*, "Secure Data Management Life Cycle for Government Big-Data Ecosystem: Design and Development Perspective," *Systems*, vol. 11, no. 8, p. 380, Jul. 2023, doi: 10.3390/systems11080380.

[13] M. Hamidouche, B. F. Demissie, and B. Cherif, "Real-time Threat Detection Strategies for Resource-constrained Devices," in *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, Apr. 2024, pp. 211–218, doi: 10.1109/DCOSS-IoT61029.2024.00038.

[14] O. Can, F. Thabit, A. O. Aljahdali, S. Al-Homdy, and H. A. Alkhzaimi, "A Comprehensive Literature of Genetics Cryptographic Algorithms for Data Security in Cloud Computing," *Cybern. Syst.*, vol. 56, no. 5, pp. 413–447, Jul. 2025, doi: 10.1080/01969722.2023.2175117.

[15] N. Wang, W. Zhou, Q. Han, J. Liu, W. Liao, and J. Fu, "A Lightweight Privacy-Preserving Ciphertext Retrieval Scheme Based on Edge Computing," *IEEE Trans. Cloud Comput.*, vol. 12, no. 4, pp. 1273–1290, Oct. 2024, doi: 10.1109/TCC.2024.3461732.

[16] J. A. Sarumi, "LASUSTECH Multidisciplinary Innovations Conference (LASUSTECH-MIC)," no. December 2021, pp. 63–82, 2022.

[17] A. Thakkar and R. Gor, "Cryptographic Method To Enhance Data Security Using Rsa Algorithm and Mellin Transform," *Int. J. Eng. Sci. Technol.*, vol. 7, no. 2, 2023, doi: 10.29121/ijoest.v7.i2.2023.490.

[18] S. S and B. V Nair, "Survey on Asymmetric Key Cryptographic Algorithms," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 7, no. 2, pp. 404–408, 2020, doi: 10.32628/ijsrset207292.

[19] Maxrizal, S. Irawadi, and Sujono, "Discrete Logarithmic Improvement for ElGamal Cryptosystem Using Matrix Concepts," *2020 8th Int. Conf. Cyber IT Serv. Manag. CITSM 2020*, 2020, doi: 10.1109/CITSM50537.2020.9268832.

[20] R. Ranasinghe and P. Athukorala, "A generalization of the ElGamal public-key cryptosystem," *J. Discret. Math. Sci. Cryptogr.*, no. May, 2021, doi: 10.1080/09720529.2020.1857902.

[21] J. Jeffries, "Differentiating by Prime Numbers," *Not. Am. Math. Soc.*, vol. 70, no. 11, p. 1, 2023, doi: 10.1090/noti2833.

[22] K. N. Adédji, F. Luca, and A. Togbé, "On the solutions of the Diophantine equation," *J. Number Theory*, vol. 240, pp. 593–610, Nov. 2022, doi: 10.1016/j.jnt.2021.12.008.

[23] D. Kumar Sharma, N. Chidananda Singh, D. A. Noola, A. Nirmal Doss, and J. Sivakumar, "A review on various cryptographic techniques & algorithms," *Mater. Today Proc.*, vol. 51, no. xxxx, pp. 104–109, 2021, doi: 10.1016/j.matpr.2021.04.583.

[24] F. R. Shareef, "A novel crypto technique based ciphertext shifting," *Egypt. Informatics J.*, vol. 21, no. 2, pp. 83–90, 2020, doi: 10.1016/j.eij.2019.11.002.