BAREKENG: Journal of Mathematics and Its Applications
December 2026 Volume 20 Issue 4 Page 3185-3196
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol20iss4pp3185-3196

ALGORITHM DESIGN OF ELGAMAL MAX-PLUS ALGEBRA FOR LETTER AND NUMBER CRYPTOGRAPHY WITH UNI-CUSTOM CIPHERTEXT

Zumrotus Sya'diyah [1*], Nur Salamah [2], Faradillah Alfiani [3]

[1, 2,3]*Mathematics Study Program, Faculty of Mathematics and Natural Sciences, Universitas Billfath Komplek PP. Al-Fattah Siman Kec. Sekaran, Lamongan, 62261, Indonesia*

*Corresponding author's e-mail: * **zuma.yakuza@gmail.com***

Article Info

Article History:

Received: 2nd October 2025
Revised: 28th April 2026
Accepted: 31st May 2026
Available online: 24th August 2026

Keywords:

Algorithm;
Cryptography
ElGamal;
Max-Plus Algebra;
Uni-Custom Ciphertext.

ABSTRACT

This study focuses on the design and implementation of a prototype cryptographic algorithm using the integration of Uni-Custom ciphertext transformation and Max-Plus algebra in the ElGamal scheme. The integration is performed through a sequential process, where the Uni-Custom ciphertext transformation converts plaintext into a dynamic numeric form, and then we use the ElGamal max plus-based algorithm in the encryption process through a public-key mechanism. The primary objective of this research is to explore the feasibility and functional correctness of the proposed hybrid framework rather than to provide a complete cryptographic performance evaluation. The implementation results show that the system exhibits high computational and structural complexity. The security of the proposed system is associated with the hardness of the discrete logarithm problem in the ElGamal scheme, while additional transformation layers enhance variability and reduce pattern predictability. Experimental results show consistent reconstruction of the original message without data loss, indicating promising potential for the development of alternative cryptographic methods based on non-conventional mathematical structures.

How to cite this article:

Z. Sya'diyah, N. Salamah, and F. Alfiani, "ALGORITHM DESIGN OF ELGAMAL MAX-PLUS ALGEBRA FOR LETTER AND NUMBER CRYPTOGRAPHY WITH UNI-CUSTM CIPHERTEXT", *BAREKENG: J. Math. & App.*, vol. 20, no. 4, pp. 3185-3196, Dec. 2026.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

The rapid development of digital and information technology has brought many benefits to mankind. The use of technology and information can be felt in various fields, ranging from education to the economy to health. However, this progress also presents new challenges, especially in the aspect of data security. Threats such as hacking and information theft are increasingly complex as technology develops, so data protection is very crucial [1], [2].

One of the main challenges in the digital era is the issue of data security, which has become crucial in many sectors such as banking, communications, and information storage. As technology becomes more sophisticated, threats to data protection, such as hacking and information theft, also grow in complexity. Therefore, the protection of data is no less important than the utilization of technology itself. Moreover, failure to address these security risks can lead to significant financial losses and damage to public trust in digital systems [3], [4].

One of the main methods to ensure data integrity and confidentiality is cryptography. Cryptography is a set of techniques to provide information security by concealing messages during transmission [5]. However, new research shows that there are some problems when implementing it. This is especially true when multiple systems have to work together. Attackers can exploit this weakness to perform cross-configuration attacks [6], [7]. As the threats to data increase, innovations in the field of cryptography become very important. The development of cryptographic methods that are stronger, more efficient, and adaptive to new technologies such as quantum computing is the main focus in maintaining information security in this digital era [8].

ElGamal algorithm is one type of algorithm in cryptography that is based on the concept of a public key. This algorithm serves to maintain the security of messages or data by forming keys involving prime numbers and discrete logarithms [9], [10]. These characteristics make the ElGamal algorithm difficult to crack, so it is effectively used in information protection. In addition, the algorithm's asymmetric structure allows secure key exchange between communicating parties without the need to share private information directly, enhancing its robustness against interception [6].

In an effort to improve the security and efficiency of cryptographic algorithms, various new approaches have begun to be developed by utilizing alternative mathematical structures, one of which is Max-Plus algebra. This algebra is a branch of mathematics that uses maximum and sum operations as the basis of its operations and has proven to have wide applications in various fields, such as optimization, networking, distributed storage systems, and network coding. Interestingly, research by Subiono [11] revealed that the application of wavelet transformation based on Max-Plus algebra can be used to develop cryptographic algorithms that are not only efficient but also more resistant to various types of attacks.

Combining the power of the ElGamal algorithm and Max-Plus algebra will help us to design a stronger and more efficient cryptographic system. This approach allows the development of algorithms that are capable of encrypting data in the form of letters and numbers with great flexibility. One of the most widely used character encoding systems is Unicode. The Unicode standard is one of the most popular character encoding systems, created to facilitate the exchange, processing, and display of written text consisting of various languages and symbols in the modern world. In this research, we will use the "Uni-Custom" one as the method in the encryption and decryption process [12], [13].

The concept of 'Uni-Custom' in this context refers to a cryptographic system that can be uniquely customized for each user or application, allowing for personalization in the security of letters and numbers. This Uni-Custom ciphertext system in cryptography can be built by converting each plaintext character into a Unicode value and then processing it using summation with repeated numeric keys and XOR operations. Combining maximization and summation operations in a Max-Plus algebraic framework can add a higher level of personalization to the encryption system. Such a customizable approach, especially when layered on dynamic encoding and user-specific keys, enhances resistance against generic cryptanalysis by increasing key-space diversity and algorithmic unpredictability [14], [15]. Although the Uni-Custom approach involves XOR operations, its role is mainly to support the overall system by contributing to a more flexible key that can be used by the user.

There have not been many studies that combine the ElGamal algorithm with Max-Plus algebra in the Uni-Custom ciphertext for letter and number cryptography. Therefore, in this research, we design a new cryptographic algorithm that integrates ElGamal and Max-Plus algebra in the Uni-Custom ciphertext framework, with the hope of improving security, efficiency, and flexibility in securing letter and number data.

This framework is used because most ElGamal frameworks are developed using standard Unicode, which is generally known, thus providing a tendency to reduce the level of security. Therefore, Uni-Custom ciphertext is a uniquely customized system that will provide extra security for its users.

This study is limited to a prototype-level implementation, focusing on the design of the proposed algorithm. As such, computational performance aspects, including execution time and memory usage, have not been evaluated in detail. In addition, formal security testing through attack simulations, such as brute-force or known-plaintext attacks, is beyond the scope of this work. The present study mainly concentrates on the structural design and functional verification of the algorithm. Further evaluation, including performance analysis and more comprehensive security testing, will be addressed in future work to better assess the robustness of the proposed system.

2. RESEARCH METHODS

This research was conducted to develop a Uni-Custom-based cryptography algorithm that integrates the ElGamal algorithm with Max-Plus algebra. The method used is applied with a descriptive-qualitative approach. The main objective of this research is to form a secure and flexible encryption system for processing data in the form of letters and numbers.

The research stages can be seen in Fig. 1 below:

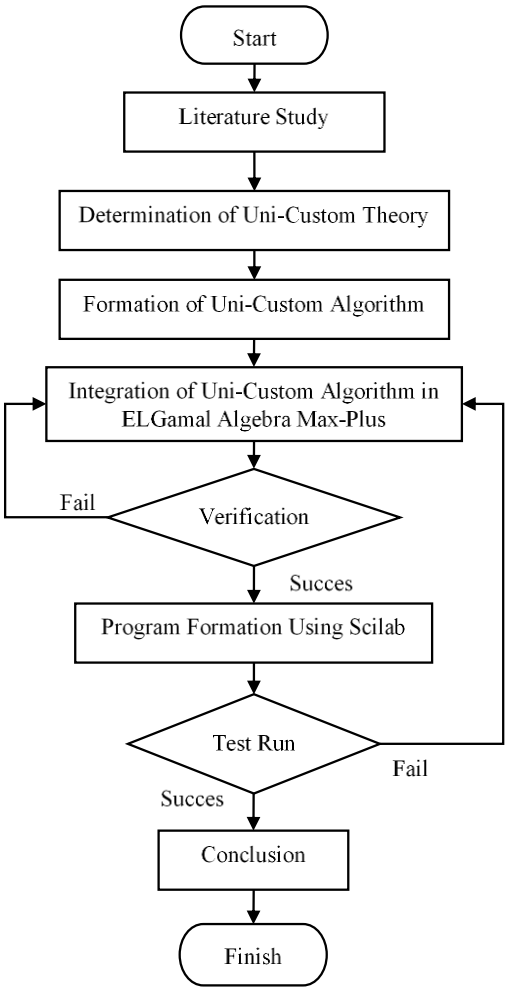

Figure 1. Flowchart of the Research

2.1 Cryptographic Foundation

This section describes the main cryptographic foundations used in the development of the Uni-Custom-based encryption system, which integrates the ElGamal algorithm, Max-Plus algebra, Unicode encoding, and the Uni-Custom XOR mechanism.

2.1.1 ElGamal-MaxPlus

ElGamal algorithm is one of the public key cryptography methods introduced by Taher ElGamal in 1985. It is an asymmetric cryptography, which uses a pair of keys: a public key for encryption and a private key for decryption. Although the public key can be known to many parties, only the owner of the private key can read the message. ElGamal works by dividing the message into blocks, which are then encrypted into ciphertext and decrypted back using the private key [9].

One of the foundations of ElGamal is the Diffie-Hellman key exchange protocol which allows two parties to form mathematically identical secret keys. Diffie-Hellman utilizes the commutative nature of the mathematical structure used. Based on the formulation proposed by Fitriani [16], if given two elements $k_A = v^k \ mod \ p$ and $k_B = u^l \ mod \ p$, with $u = g^k \ mod \ p, v = g^l \ mod \ p$ where $g \in \mathbb{Z}_p$, it can be proven that $k_A = k_B$ [16]. This equivalence ensures that both communicating parties can generate the same secret key even though they never directly exchange their private keys. The strength of this approach lies in the computational difficulty of the discrete logarithm problem, which serves as the backbone of ElGamal's security.

To strengthen its security, Stickel introduced a non-commutative algebra-based approach, where the order of operations affects the result but still yields the same secret key for both parties. Based on Stickel's theorem, if given $k_A = a^n v b^m$ and $k_B = a^r u b^s$, with m, n, r, s are integers and $u = a^n b^m, v = a^r b^s$, where $a, b \in G$, then $k_A = k_B$ is still obtained [16]. This adaptation expands the applicability of Diffie-Hellman-type key exchange to more complex algebraic systems beyond simple commutative groups. As a result, Stickel's approach opened a new perspective for developing cryptosystems based on non-commutative structures that are potentially more resistant to algebraic attacks.

In an effort to develop further, alternative algebraic structures such as Max-Plus algebra began to be used. Max-Plus algebra is an idempotent semiring that uses maximum operation ($\oplus$) instead of addition and ordinary addition ($\otimes$) instead of multiplication. Although it does not form groups or fields, this structure is useful in discrete systems such as scheduling. Its algebraic characteristics make Max-Plus a promising tool for modeling synchronization and optimization in cryptography. This potential motivates researchers to integrate Max-Plus algebra into existing cryptographic frameworks such as ElGamal [10], [17].

In cryptography, the Max-Plus approach has been applied to modify the ElGamal algorithm, as studied by [16] and [18]. The Diffie-Hellman protocol has also been adapted in Max-Plus by [19], allowing for a more efficient computation process if $k_A = v^{\otimes k}$ and $k_B = u^{\otimes l}$ with k, l are integers, $u = g^{\otimes k}$ and $v = g^{\otimes l}$, where g is element of $\mathbb{R}_{max}$, then equality $k_A = k_B$ is obtained. This adaptation shows that the Max-Plus structure can preserve the essential symmetry required for key exchange while reducing computational complexity. Thus, the Max-Plus modification not only maintains security but also enhances performance efficiency in encryption schemes [16], [18], [19].

Stickel also adapted his approach to Max-Plus by using non-commutative semigroups. In this form, given $k_A = a^{\otimes n} v b^{\otimes m}$ and $k_B = a^{\otimes r} u b^{\otimes s}$ with m, n, r, s are integers, $u = a^{\otimes n} b^{\otimes m}$, and $v = a^{\otimes r} b^{\otimes s}$, where $a, b \in \mathbb{G}_{n \times n}(\mathbb{Z}_{max})$, then both parties will obtain the same secret key, namely $k_A = k_B$ [16]. With its flexibility, Max-Plus algebra provides a powerful structural alternative for developing modern cryptographic systems that are efficient and resistant to attacks. This combination of non-commutativity and idempotent operations offers new directions for research in algebraic cryptography.

2.1.2 Unicode

Unicode is a universal character encoding system designed to digitally represent symbols and alphabets from various languages [12]. In cryptography, text messages need to be converted to numeric form in order to be processed by encryption algorithms. Unicode provides that numerical basis by assigning a unique value to each character, usually in decimal, hexadecimal, or encoding formats such as UTF-8 and UTF-16 [20]. This standardization ensures that data can be consistently interpreted across different platforms, programming languages, and systems.

The use of Unicode allows cryptographic algorithms, such as ElGamal or Max-Plus algebra-based models, to work on multilingual texts and complex symbols. The encryption process starts with the conversion of characters into Unicode values, which are then mathematically processed. Conversely, the decryption result in the form of numbers will be converted back into the original characters. This bidirectional

transformation plays an important role in maintaining message integrity, ensuring that no data loss or misinterpretation occurs during the encryption and decryption stages [4], [21].

In the Scilab-based ElGamal-MaxPlus system, text-to-number conversion is performed with the syntax a = **ascii**(txt), and vice versa to return to the original text. This approach enables efficient, consistent, and compatible text processing without the need for specialized tables. Moreover, it facilitates seamless integration between the mathematical operations of the Max-Plus algebra and the string handling features of Scilab. It also supports high flexibility in securing data in diverse digital environments.

2.1.3 Uni-Custom

Uni-Custom is a code customization system used to uniquely and flexibly represent letters, numbers, or symbols in the encryption process. Each character is encoded based on a key or system parameter, unlike standard systems such as ASCII or Unicode that use fixed encoding. With dynamic encoding, Uni-Custom provides a higher level of security because it is difficult to analyse or predict by third parties [22].

Uni-Custom allows high flexibility in encryption and decryption, as each character can be changed differently depending on the key used. This makes the system more difficult to analyse or predict, making it more secure compared to standard coding systems that use fixed rules. In the Uni-Custom ciphertext system, users can specify a numeric key based on two main parameters, namely the number of key elements and the interval of allowed values. This flexibility means that even small modifications in the key configuration can generate entirely different encoded results, increasing resistance to brute-force or pattern-based attacks. This key is a string of integers that plays a role in character transformation during the encryption and decryption process. Being self-configurable and flexible, this approach adds complexity to the system as well as enhances security by creating a very large key space that varies between users [23].

The main operation used in this encryption scheme is the XOR (Exclusive OR) logical operation. In Boolean algebra, XOR can be expressed using the standard Boolean expression:

$$Y = A'B + AB', \text{ where } A, B \text{ are binary inputs.}$$

In a cryptographic context, XOR is symmetric in a bit-by-bit context, which means that the XOR result between two numbers can be restored to its original value by XORing the result again using the same key number. This characteristic makes XOR very useful in data encryption and decryption [14], [24]. In combination with Uni-Custom ciphertext encoding, XOR further enhances data protection by ensuring that even identical plaintexts produce distinct ciphertexts when different keys are applied [25].

The Uni-Custom encryption process involves three main stages:

1. Unicode Conversion

   Each character of the message (plaintext) is converted into a Unicode numeric representation using the Syntax a = **ascii**(txt) on the Scilab platform.

2. XOR Transformation

   The resulting Unicode value is then subjected to an XOR operation with the elements of a numeric key. The key application is done cyclically to handle messages whose length exceeds the number of elements in the key. This key can be chosen according to the user's individual preferences.

3. Ciphertext Generation

   After the XOR operation-based transformation process is completed, the next stage is the generation of ciphertext through the application of Max-Plus algebra-based encryption.

As an illustration, the encryption of the text "HELLO!" with a random numeric key consisting of four elements with values in the range of 1 to 9 produces the following output:

Table 1. Uni-custom Ciphertext Trial Program of The Encryption Process

Plaintext	Unicode	Numeric key	Uni-Custom	
H	72	9	65	A
E	101	6	99	C
L	108	1	109	M

Plaintext	Unicode	Numeric key	Uni-Custom	
L	108	5	105	I
O	111	9	102	F
!	33	6	39	‘ (Apostrophe)

The decryption process is performed using the same key, ensuring the original message can be recovered intact.

Table 2. Uni-custom Ciphertext Trial Program of The Decryption Process

Ciphertext	Numeric key	Unicode	Uni-Custom Description
65	9	72	A
99	6	101	C
109	1	108	M
105	5	108	I
102	9	111	F
39	6	33	‘ (Apostrophe)

The advantages of this system lie in its simplicity while still guaranteeing security, as well as its ability to be integrated with other cryptographic algorithms such as ElGamal and Max-Plus algebraic structures. With its modular and adaptive nature, Uni-Custom is suitable for securing character data in modern systems, as well as being a potential basis for the development of more complex cryptographic algorithms in the future.

It should be noted that the Uni-Custom ciphertext mechanism is not designed to operate as a standalone cryptographic method. Rather, it acts as a preprocessing step that introduces additional variability through dynamic, key-dependent character encoding. This transformation helps diversify the input data before it is processed by the ElGamal–Max-Plus encryption stage. Although the Uni-Custom approach involves XOR operations, its role is mainly to support the overall system by contributing to a more flexible key that can be used by the user.

3. RESULTS AND DISCUSSION

This section presents the implementation and experimental results of a cryptographic system based on the integration of the ElGamal max plus based algorithm and Uni-Custom character ciphertext transformation. The proposed method adopts a Max-Plus-based ElGamal approach, where Max-Plus operations are directly incorporated into the ElGamal encryption and decryption processes. As such, the two components form a unified framework rather than separate procedures.

The main objective of the system is to secure the text message exchange process, which contains letters, numbers, and symbols, through a layered encryption approach. The process starts by converting the text to Unicode numeric form and performing an XOR operation with a numeric key cyclically (Uni-Custom). The result is then encrypted using a modified ElGamal algorithm with Max-Plus operations.

Table 3. The Proposed Algorithm of Uni-Custom Encryption

Algorithm 1. Uni-Custom Encryption

Input: Plaintext (*text*), Numeric key (*key*).
Output: Numeric vector (*result*)

1: Validate input types
   If *text* is not a string → Error: “Input text must be a string.”
   If *key* is not a numeric vector → Error: “Input key must be a numeric vector.”
2: Convert text to ASCII codes → *ascii_code* = asciimat(*text*)
3: Determine the length of the text and the key → *length_text*, *length_key*

4: Generate a repeated cyclic key to match the text length →
cyclic_key = *key*(modulo(0 : *length_text* − 1, *length_key*) + 1)
5: Perform bitwise XOR operation between *ascii_code* and *cyclic_key* →
result = bitxor(*ascii_code*, *cyclic_key*)
6: Return *result*

The Uni-Custom encryption algorithm, as outlined in **Algorithm 1**, represents the initial phase of the cryptographic framework. In this stage, plaintext is transformed into a numeric vector through ASCII conversion and a cyclic XOR operation using a predefined numeric key. The resulting vector serves as input for the subsequent ElGamal–Max-Plus encryption process, while the cyclic key enhances variability and strengthens resistance against frequency-based attacks.

Table 4. The Proposed Algorithm of Uni-Custom Decryption

Algorithm 2. Uni-Custom Decryption

Input: Decryption results (*decrypted_data*), Numeric key (*key*).
Output: Original text (*plaintext*)

1: Validate input types
If d*ecrypted_data* is not a numeric vector → Error: "Input decryption data must be a numeric vector."
If *key* is not a numeric vector → Error: "Input key must be a numeric vector."
2: Compute the length of the decryption data and the key → *length_data*, *length_key*
3: Generate a repeated cyclic key to match the data length →
cyclic_key = *key*(modulo(0 : *length_data* − 1, *length_key*) + 1)
4: Perform bitwise XOR operation between decryption data and cyclic key →
ascii_code = bitxor(d*ecrypted_data*, *cyclic_key*)
5: Convert ASCII code to characters → *plaintext* = asciimat(*ascii_code*)
6: Return *plaintext*

After the ElGamal–Max-Plus decryption phase, as shown in **Algorithm 2**, the Uni-Custom decryption process converts the obtained numerical vector back to its original form. Through XOR operations and ASCII reconversion with the same key, this stage ensures the reversibility and consistency of the entire cryptographic system, accurately restoring the original plaintext without data loss.

To ensure the reliability and robustness of the proposed Uni-Custom–ElGamal–Max-Plus cryptographic system, a comprehensive testing process was conducted using multiple input variations. The testing procedure was designed to evaluate correctness, consistency, and reversibility of the encryption–decryption mechanism. A total of 30 different test messages were used in the experiment. These messages were intentionally varied in structure and complexity, including:

a). Short words (e.g., 3–5 characters).
b). Medium-length sentences (6–15 characters).
c). Mixed-character inputs containing letters, numbers, and symbols.
d). Repeated-character patterns to evaluate resistance to pattern-based analysis.

The experimental results confirm that the proposed system operates reliably across diverse input types. The use of dynamic Uni-Custom encoding combined with Max-Plus algebra and ElGamal encryption ensures both functional correctness and enhanced security. The system demonstrates strong robustness, as all tested messages were accurately recovered.

Table 5. Example of Encryption and Decryption Results

Type	Parameter / Process	Value	Description
Input	Public Key (User 1)	123	The public key generated as part of the ElGamal Max Plus Algebra key pair used for encryption.
Input	Private Key (User 1)	12	The private key corresponding to the public key is utilized during the encryption process.
Input	Plaintext	Hallo!123	The original message to be encrypted in the Uni-Custom–ElGamal–Max-Plus framework.

Type	Parameter / Process	Value	Description
Input	Number of Key Elements	4	The number of numeric components forming the Uni-Custom key vector.
Input	Lower Limit of Key Interval	1	Minimum possible value for key element generation in the Uni-Custom configuration.
Input	Upper Limit of Key Interval	100	Maximum possible value for key element generation in the Uni-Custom configuration.
Output	Numeric Key	[1,7,34,71]	Numeric key vector generated within the defined interval limits.
Output	Uni-Custom Output	[73,102,78,43]	Result of the Uni-Custom operation between the plaintext and numeric key, forming a numeric vector for subsequent encryption
Input	Private Key (User 2)	23	Recipient's private key used for decryption.
Output	Ciphertext	[34021, 34050, 34026, 33991]	Encrypted numeric vector obtained through ElGamal–Max-Plus encryption.
Output	Decrypted Numeric Data	[73, 102, 78, 43]	Numeric vector obtained after performing the decryption process.
Output	Decrypted Text	Hallo!123	Final plaintext reconstructed using the Uni-Custom decryption algorithm.

In this paper, we give one of the simplest testing mixed-character inputs. The test results demonstrate the full integration of Uni-Custom with the ElGamal–Max-Plus encryption framework. For illustration, public and private key pairs (123, 12) are applied to the input message "Hallo!123," using a randomly generated numeric key [1, 7, 34, 71] from the interval [1, 100]. The Uni-Custom transformation produces the numeric vector [73, 102, 78, 43], which is subsequently encrypted using ElGamal–Max-Plus operations into the ciphertext [34021, 34050, 34026, 33991]. The decryption process successfully reconstructs the original message, confirming that the system functions symmetrically and accurately.

To further demonstrate the application of Max-Plus algebra within the proposed scheme, a numerical example consistent with the parameters shown in Table 3 is provided. Let $g = 123$ denote the public generator, while the private keys of User 1 and User 2 are $k = 12$ and $l = 23$, respectively. In Max-Plus algebra, exponentiation with respect to the operator $\otimes$ corresponds to repeated addition, such that $g^{(\otimes k)} = k \times g$.

Based on this definition, the public values are obtained as

$$u = g^{(\otimes k)} = 123^{(\otimes 12)} = 12 \times 123 = 1476,$$
$$v = g^{(\otimes l)} = 123^{(\otimes 23)} = 23 \times 123 = 2829.$$

Both parties subsequently compute the shared secret key using

$$k_A = v^{(\otimes k)}, k_B = u^{(\otimes l)}$$

which results in

$$k_A = 2829^{(\otimes 12)} = 33948,$$
$$k_B = 1476^{(\otimes 23)} = 33948.$$

Thus, both users obtain the same shared secret key, namely $K = 33948$. According to the Max-Plus multiplication rule $a \otimes b = a + b$, the ciphertext elements are generated from the Uni-Custom output vector [73,102,78,43,110,38,19,50] as

$$C_i = M_i \otimes K = M_i + K$$

for instance,

$$73 \otimes 33948 = 73 + 33948 = 34021$$

which corresponds to the first ciphertext value presented in Table 3. The same procedure is applied to the remaining elements to produce the ciphertext vector [34021, 34050, 34026, 33991, 34058, 33986, 33967, 34065, 33998].

During the decryption stage, the same shared key obtained from the Max-Plus Diffie–Hellman mechanism is utilized. Since Max-Plus multiplication corresponds to ordinary addition, the inverse operation for recovering the plaintext is performed by subtracting the shared key from each ciphertext element. Therefore,

$$M_i = C_i - K$$

where $K = 33948$. For example,

$$34021 - 33948 = 73$$

Similarly,

$$34050 - 33948 = 102, \qquad 34026 - 33948 = 78, \qquad 33991 - 33948 = 43$$

which reconstructs the original numeric vector [73,102,78,43,110,38,19,50].

Finally, the inverse Uni-Custom transformation is applied to this vector using the same cyclic key $[1, 7, 34, 71]$. For example,

$$\begin{aligned} 73 \oplus 1 &= 72 \rightarrow H, \\ 102 \oplus 7 &= 97 \rightarrow A, \\ 78 \oplus 34 &= 108 \rightarrow L, \\ 43 \oplus 71 &= 108 \rightarrow L, \\ 110 \oplus 1 &= 111 \rightarrow O, \\ 38 \oplus 7 &= 33 \rightarrow !, \\ 19 \oplus 34 &= 49 \rightarrow 1, \\ 117 \oplus 71 &= 50 \rightarrow 2, \\ 50 \oplus 1 &= 51 \rightarrow 3. \end{aligned}$$

The remaining characters are reconstructed in the same manner, resulting in the recovery of the original plaintext message “Hallo!123”.

With the aim of making it easier to understand the input and output of each stage of our cryptography process, the following is a picture illustration from Table 5.

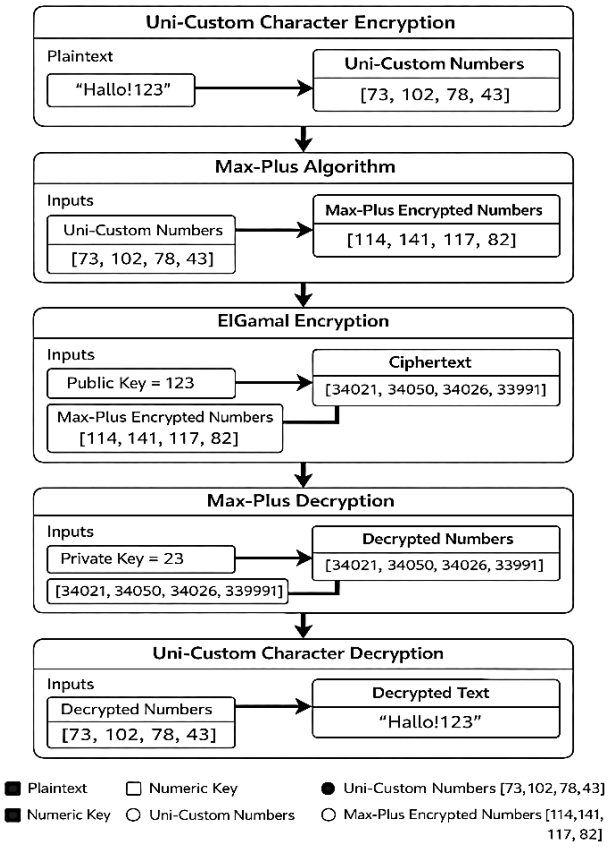

Figure 2. Illustration Figure of Each Stage of ElGamal–Uni-Custom Cryptography Showed in Table 5

From a security perspective, the Uni-Custom transformation effectively obscures character patterns. Identical sequences such as "AAA" yield distinct ciphertexts under different key elements, making frequency-based attacks ineffective. When combined with ElGamal and Max-Plus algebra, the system achieves enhanced resistance against classical cryptanalytic methods. The nonlinear transformation inherent to Max-Plus operations increases the numerical complexity and randomness of the ciphertext, thereby hindering predictive decryption attempts.

The results of the program show the complete process stages from key generation, character transformation, encryption, to decryption. The result reflects the success of the system in maintaining the integrity of the message through a clear and modular workflow. This is reinforced by experimental data showing that each stage runs as expected without data loss or logical errors.

The advantages of this system include: (1) Uni-Custom ciphertext increases the complexity of the cryptographic system by adding a flexible (we can use various keys to change from the Unicode form to the Uni-Custom one) and lightweight character transformation layer, (2) The Max-Plus-based ElGamal scheme enables secure key exchange while introducing an alternative numerical transformation compared to classical modular approaches. This integration adds structural complexity and may reduce pattern predictability. The system is suitable for short to medium-length text data and remains relatively efficient within a prototype implementation. The performance metrics, including execution time, memory usage, and throughput, have not been systematically evaluated in this study. As a result, the efficiency of the proposed system is discussed qualitatively based on its prototype implementation, because we focused this research only on designing the proposed algorithm, which integrated Elgamal-Max Plus and Uni-Custom ciphertext. Further quantitative evaluation will be addressed in future research.

Overall, the integration between ElGamal, Max-Plus algebra, and Uni-Custom results in an innovative, secure, and logical cryptographic system. The system is able to effectively disguise message structure, maintain data integrity, and exhibit resistance to conventional attacks. With improvements on efficiency and interface, this algorithm has great potential to be further developed in character-based data security applications. The criteria of the cryptographic system's effectiveness are the accuracy, Integrity and Security behavior. The accuracy and integrity have been obtained successfully through 30 tests on various types of input characters. For the security behavior, it is found that with the uni-custom ciphertext process, many possible keys can be used in the process, thus increasing the level of data transfer security.

The main contribution of this work lies in the integration of a dynamic character transformation scheme (Uni-Custom) with Max-Plus algebra and the ElGamal cryptosystem. This hybrid approach introduces a multi-layer encryption framework that combines classical cryptographic security with non-conventional algebraic structures. Unlike conventional systems that rely solely on fixed encoding schemes, the proposed method enhances security through adaptive encoding and increased structural complexity.

4. CONCLUSION

According to the result above, we can conclude that:

1. Max-Plus algebra can be involved in cryptography theory, especially for the ElGamal algorithm using Uni-Custom ciphertext-based numerical transformation.
2. This proposed cryptography method consists of three main stages, i.e., the process of converting characters into numerical form with Uni-Custom ciphertext, data obfuscation using XOR operations, and the application of Max-Plus algebra-based encryption, which adds to the complexity of numerical transformation, together increasing the structural complexity of the transformation.
3. This algorithm design successfully combines the deterministic and non-deterministic methods in one modular framework that is relatively simple to implement and suitable for prototype-level development.
4. The algorithm enhances the structural complexity of the cryptographic system through the Uni-Custom encoding process, which allows flexible key selection based on user-defined preferences.

Building on these findings, future work may explore the application of the proposed algorithm in file-based security systems, real-time processing environments, and its integration into digital communication protocols that demand robust encryption. Further investigation is also needed to evaluate computational

performance, scalability, and security through standard attack models. In addition, comparative analysis with established cryptographic schemes and testing on larger or real-world datasets would provide a more comprehensive assessment of the proposed method.

Author Contributions

Zumrotus Sya'diyah: Validation, writing–original draft; Nur Salamah: Conceptualization, software, writing–original draft; Faradillah Alfiani: Review, editing, supporting references.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment

The author extends sincere gratitude to all individuals who contributed to the development of this research. Special appreciation is given to the reviewers and academic colleagues whose insights and constructive feedback significantly enhanced the quality of this work.

Declarations

The authors declare no competing interests.

Declaration of Generative AI and AI-assisted Technologies

Generative AI tools (e.g., ChatGPT) were used solely for language refinement, including grammar, spelling, and clarity. The scientific content, analysis, interpretation, and conclusions were developed entirely by the authors. All final text was reviewed and approved by the authors.

REFERENCES

[1] Adi Ahmad, Riyan Maulana, and Muhammad Yassir, "CYBERSECURITY CHALLENGES IN THE ERA OF DIGITAL TRANSFORMATION A COMPREHENSIVE ANALYSIS OF INFORMATION SYSTEMS," *Journal Informatic, Education and Management (JIEM)*, vol. 6, no. 1, pp. 7–11, Feb. 2024. doi: https://doi.org/10.61992/jiem.v6i1.57.

[2] S. Quach, P. Thaichon, K. D. Martin, S. Weaven, and R. W. Palmatier, "DIGITAL TECHNOLOGIES: TENSIONS IN PRIVACY AND DATA," *J Acad Mark Sci*, vol. 50, pp. 1299–1323, 2022. doi: https://doi.org/10.1007/s11747-022-00845-y

[3] Y. T. Y. Azura, M. A. Azad, and Y. Ahmed, "AN INTEGRATED CYBER SECURITY RISK MANAGEMENT FRAMEWORK FOR ONLINE BANKING SYSTEMS," *Journal of Banking and Financial Technology*, May 2025. doi: https://doi.org/10.1007/s42786-025-00056-3

[4] N. Alanazi, E. Khan, and A. Gutub, "INCLUSION OF UNICODE STANDARD SEAMLESS CHARACTERS TO EXPAND ARABIC TEXT STEGANOGRAPHY FOR SECURE INDIVIDUAL USES," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1343–1356, Apr. 2022. doi: https://doi.org/10.1016/j.jksuci.2020.04.011

[5] S. N. Nugraha, "PENERAPAN ALGORITMA KRIPTOGRAFI ELGAMAL PADA APLIKASI PENGAMANAN PESAN BERBASIS WEBSITE," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, pp. 2513–2524, Aug. 2024. doi: https://doi.org/10.23960/jitet.v12i3.4794.

[6] L. De Feo, B. Poettering, and A. Sorniotti, "ON THE (IN)SECURITY OF ELGAMAL IN OPENPGP," ACM, 2021. Accessed: Oct. 01, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3460120.3485257. doi: https://doi.org/10.1145/3460120.3485257

[7] H. Li, G. Yang, J. Ming, Y. Zhou, and C. Jin, "TRANSPARENCY ORDER VERSUS CONFUSION COEFFICIENT: A CASE STUDY OF NIST LIGHTWEIGHT CRYPTOGRAPHY S-BOXES," *Cybersecurity*, vol. 4, no. 1, pp. 1–20, Dec. 2021. doi: https://doi.org/10.1186/s42400-021-00099-1.

[8] F. Liu *et al.*, "A SURVEY ON LATTICE-BASED DIGITAL SIGNATURE," *Cybersecurity*, vol. 7, no. 1, pp. 1–18, Dec. 2024. doi: https://doi.org/10.1186/s42400-023-00198-1

[9] F. Husaini, A. M. H. Pardede, and I. Gultom, "PENERAPAN ENKRIPSI MENGGUNAKAN METODE ELGAMAL GUNA MENINGKATKAN KEAMANAN DATA TEKS DAN GAMBAR," *JUKI : Jurnal Komputer dan Informatika*, vol. 4, pp. 67–73, May 2022.

[10] K. A. Sattar, T. Haider, U. Hayat, and M. D. Bustamante, "AN EFFICIENT AND SECURE CRYPTOGRAPHIC ALGORITHM USING ELLIPTIC CURVES AND MAX-PLUS ALGEBRA-BASED WAVELET TRANSFORM," *Applied Sciences (Switzerland)*, vol. 13, no. 14, pp. 1–19, Jul. 2023. doi: https://doi.org/10.3390/app13148385.

[11] Subiono, J. Cahyono, D. Adzkiya, and B. Davvaz, "A CRYPTOGRAPHIC ALGORITHM USING WAVELET TRANSFORMS OVER MAX-PLUS ALGEBRA," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 627–635, Mar. 2022. doi: https://doi.org/10.1016/j.jksuci.2020.02.004.

[12] Supriyanto, Isbandiyah, and T. A. Yeni, "INTEGRASI AKSARA DAERAH KE DALAM SISTEM INFORMASI KOMPUTER SEBAGAI UPAYA MELESTARIKAN AKSARA DAERAH," *Bina Gogik*, vol. 9, no. 1, pp. 185–191, Mar. 2022.

[13] A. Wibowo, K. Wijayanti, and R. Budhiati Veronica, "PENERAPAN ALJABAR MAX-PLUS PADA PENGATURAN SISTEM ANTRIAN TRAFFIC LIGHT," *UNNES Journal of Mathematics*, vol. 7, pp. 192–205, Nov. 2018, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/ujm

[14] K. Khairani and M. Z. Siambaton, "PENGAMANAN DATA TEKS MENGGUNAKAN ALGORITMA KRIPTOGRAFI ELGAMAL DAN XOR DARI SERANGAN HACKER," *sudo Jurnal Teknik Informatika*, vol. 2, no. 4, pp. 176–187, Dec. 2023. doi: https://doi.org/10.56211/sudo.v2i4.401.

[15] K. K. Johar, "GENERATION OF A NOVEL CRYPTOGRAPHIC ALGORITHM FOR IMPLEMENTATION OF 'PLAY COLOR CIPHER' SUBSTITUTION TECHNIQUE WITH UNICODE TRANSFORMATION FORMAT-8." [Online]. Available: www.ijert.org

[16] R. N. Fitriani, "MODIFIKASI ENKRIPSI ELGAMAL DALAM ALJABAR MAX-PLUS UNTUK PENGAMANAN DOKUMEN BAHASA ARAB," Undergraduate thesis, Universitas Islam Negri Walisongo, Semarang, 2023.

[17] A. S. Septiany, Siswanto, and V. Y. Kurniawan, "PERMANENT AND DOMINANT OF MATRIX OVER INTERVAL MIN-PLUS ALGEBRA," *BAREKENG: Journal of Mathematics and Its Applications*, vol. 18, no. 3, pp. 2029–2034, Aug. 2024. doi: https://doi.org/10.30598/barekengvol18iss3pp2029-2034.

[18] A. Muanalifah, A. I. Riana, and R. Artes Jr, "THE TROPICAL VERSION OF EL GAMAL ENCRYPTION," *Journal of Natural Sciences and Mathematics Research J. Nat. Scien. & Math. Res*, vol. 9, no. 2, pp. 151–157, 2023, [Online]. Available: http://journal.walisongo.ac.id/index.php/jnsmr. doi: https://doi.org/10.21580/jnsmr.v9i2.18704

[19] D. Grigoriev and V. Shpilrain, "TROPICAL CRYPTOGRAPHY," Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.1195

[20] R. S. Roring, T. I. Gunawan, and Y. B. Samponu, *DASAR DAN TEORI SISTEM MULTIMEDIA*. Bekasi: JUI Press, 2022.

[21] A. Kumar, R. Yadav, and R. Kumar, "UNICODE AND COLOR INTEGRATION TECHNIQUE FOR ENCRYPTION AND DECRYPTION," *International Journal of Advanced Research in Computer Science*, vol. 4, no. 8, [Online]. Available: www.ijarcs.info

[22] D. Sutanto and A. Hariyanto, *PENGENALAN KRIPTOGRAFI DAN KEAMANANNYA*. Yogyakarta: Andi Offset, 2020.

[23] D. Ibrahim, K. Ahmed, M. Abdallah, and A. A. Ali, "A NEW CHAOTIC-BASED RGB IMAGE ENCRYPTION TECHNIQUE USING A NONLINEAR ROTATIONAL 16 × 16 DNA PLAYFAIR MATRIX," *Cryptography*, vol. 6, no. 2, pp. 1–27, Jun. 2022. doi: https://doi.org/10.3390/cryptography6020028.

[24] R. Sulthanah and N. Ahmad, "PENERAPAN METODE MCCULLOCH-PITTS MENGGUNAKAN PYTHON UNTUK PENGUJIAN PENGENALAN POLA OPERATOR AND, OPERATOR OR, OPERATOR XOR PADA FUNGSI LOGIKA," *J-INTECH(Journal of Information and Technology)*, pp. 347–360, Nov. 2022. doi: https://doi.org/10.32664/j-intech.v11i2.1022

[25] L. Adrián Lizama-Pérez, "XOR Chain and Perfect Secrecy at the Dawn of the Quantum Era †," vol. 7, p. 50, 2023.. doi: https://doi.org/10.3390/cryptography7040050