DOI: 10.24996/ijs.2024.65.12.37

ISSN: 0067-2904

Enhanced DES Algorithm Using Efficient Classical Algorithm

Inas Ali Abdulmunem*, Mays M. Hoobi

Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq

Received: 25/9/2023 Accepted: 9/12/2023 Published: 30/12/2024

Abstract

The privacy, confidentiality, and integrity of user information are very important concerns; therefore, cryptography is the solution to keep and satisfy these objectives. Encryption is the most important policy that is used to make transmitted messages have a high level of security and be so difficult to crack by attackers. The main aim of this research is divided into two parts: The aim of Part 1 is to obtain the highest level of complexity against the attacker by applying six classical cryptography algorithms to the same plain-text message. These six algorithms are Vigenère, Porta, Autokey, Beaufort, Trithemuis, and Gronsfeld. Also, in this part, several analysis tools are used to evaluate the performance of the six algorithms: entropy, histogram, and autocorrelation. Part 2 of this research aims to apply the results of Part 1 by using the best algorithm among these six algorithms (Vigenère) to improve the security level of the Data Encryption Standard (DES) algorithm by applying two experiments for two hybrid ciphering models. Each experiment has three cases. The first experiment included three cases: the first case included ciphering plain text by using only the Vigenère algorithm, the second case included ciphering plain text by using only the ECB DES algorithm, and the third case included multilevel ciphering plain text by using Vigenère and then ECB DES algorithms, while the second experiment for the second hybrid ciphering model included the same cases as the first experiment but applied CBC DES, another type of DES algorithm. The results of this research indicated different efficiency levels of six classical algorithms, and the Vigenère algorithm is the best and offers the highest level of complexity among the used algorithms. In addition, the Vigenère algorithm can be used to enhance the security level of cryptography algorithms and increase the complexity of cracking ciphertext when combined with another algorithm like the DES algorithm.

Keywords: Cipher-Text, Cryptography, Decryption, DES, Encryption, Plain-Text, Security, Vigenère.

خوارزمية معيار تشفير البيانات المحسنة باستعمال الخوارزمية الكلاسيكية الفعالة

ايناس علي عبدالمنعم *, ميس محمد هوبي

قسم علوم الحاسوب, كلية العلوم, جامعة بغداد, بغداد, العراق

الخلاصة

تعد خصوصية وسرية وسلامة معلومات المستخدم من الاهتمامات المهمة للغاية، وبالتالي فإن التشفير هو الحل للحفاظ على هذه الأهداف وتحقيقها. التشفير هو السياسة الأكثر أهمية التي يتم استعمالها لجعل الرسائل المرسلة بمستوى عالٍ من الأمان ويصعب اختراقها من قبل المهاجمين. الهدف الرئيسي من هذا البحث

*Email: inas.ali@uobaghdad.edu.iq

مقسم إلى جزئين: الهدف من الجزء الأول هو الحصول على أعلى مستوى من التعقيد ضد المهاجم من خلال تطبيق ست خوارزميات تشفير كلاسيكية مع نفس الرسالة ذات النص العادي. هذه الخوارزميات الستة هي: فيجينير وبورتا والمفتاح التلقائي و بوفورت و تريثيموس و جرونسفيلد. بالإضافة إلى ذلك، في هذا الجزء، يتم استعمال العديد من أدوات التحليل لتقييم أداء الخوارزميات الستة، هذه الأدوات هي إنتروبيا والرسم البياني والارتباط التلقائي. الجزء الثاني من هذا البحث يهدف الى استثمار نتائج الجزء الأول، من خلال استعمال التأثير الجيد لأفضل خوارزمية من بين هذه الخوارزميات الستة (فيجينير) لتحسين مستوى أمان خوارزمية معيار تشفير البيانات من خلال تطبيق تجربتان لنموذجين للتشفير الهجين. كل تجربة لها ثلاث حالات حيث تضمنت التجربة الأولى الثلاث حالات التالية: الحالة الأولى تضمنت تشفير النص العادي باستعمال خوارزمية فيجينير فقط، والحالة الثانية لتشفير النص العادي باستعمال خوارزمية تشفير البيانات الإلكترونية القياسية فقط، والحالة الثالثة لتشفير النص العادي متعدد المستويات باستعمال خوارزميات فيجينير ثم تشفير البيانات الإلكترونية القياسية’ بينما تضمنت التجربة الثانية لنموذج التشفير الهجين الثاني نفس حالات التجربة الأولى ولكنها طبقت خوارزمية تشفير كتلة التشفير القياسية للبيانات نوعا آخر من خوارزمية معيار تشفير البيانات.

أشارت نتائج هذا البحث إلى مستويات كفاءة مختلفة لست خوارزميات كلاسيكية مطبقة وخوارزمية فيجينير هي الأفضل وتقدم أعلى مستوى من التعقيد بين الخوارزميات المستعملة، بالإضافة إلى أنه يمكن استعمال خوارزمية فيجينير لتحسين مستوى أمان خوارزميات التشفير، وزيادة تعقيد كسر النص المشفر عند دمجه مع خوارزمية أخرى مثل خوارزمية معيار تشفير البيانات.

1. Introduction

Nearly all respectable sectors of commerce, government, and industry in today's more complex world use computers for their work [1]. There is no denying the capabilities of computer devices, which are demonstrated by the degree of accuracy and high rate of work completion [2]. Beyond the bias advantage derived from computer use, the most crucial factor to be taken into account is a part of its security, since if the information or data kept in the computer experienced damage or loss, it might result in significant losses [3] [4]. A computer that is not adequately secured will provide hackers with a fantastic opportunity to access the system and take any data they desire [5] [6].

The volume of data transferred in the modern world has expanded, making information security an increasingly important duty [7] [8]. It is crucial that public networks communicate data quickly, especially when it comes to information security. We must protect our information from intruders or attacks [9]. Information security also becomes necessary for us [10]. Given the importance of the data, sending and receiving it securely is important [11]. Cryptography addresses the facets of data security that include data integrity, confidentiality, origin authentication, and entity authentication [12] [13].

In its simplest form, cryptography refers to a concealed or secret method of communication between two parties [14]. This is made feasible when the communication data is concealed throughout the path that the data takes to get from sender to receiver [15] [16]. Encryption is the process of transforming plain text into cipher text; this operation is performed at the sender’s end [17]. It is also regarded as one of the most effective tools for safe data transmission through communication networks. Decryption, or deciphering, is the opposite process of encryption [18] [19]. It converts the cipher text into plain text and is carried out at the receiver end, as shown in Figure 1 [7] [20].

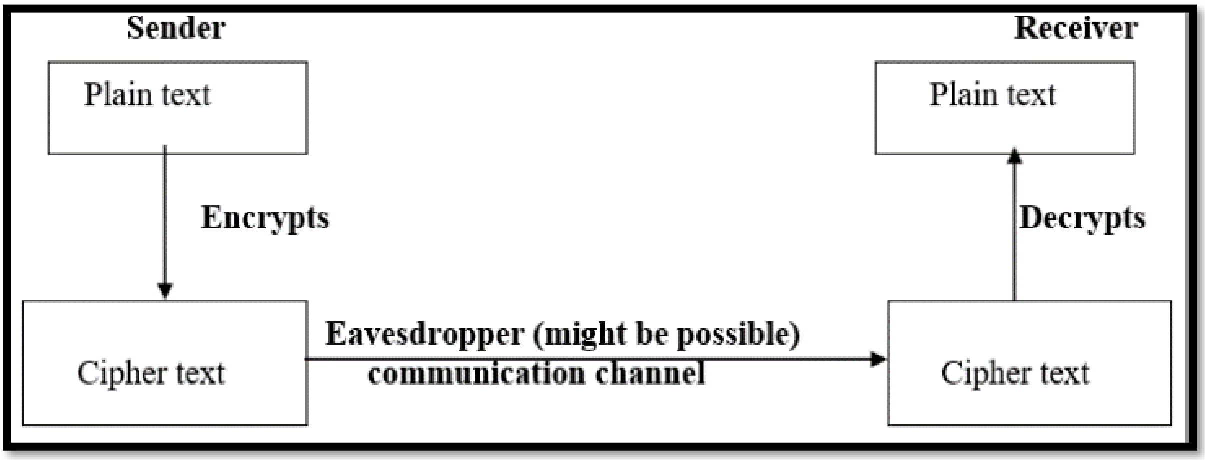


Figure 1: Encryption and Decryption Operations [7]

Before being encrypted into cipher text, plain text (at the sender) is a simple, readable text in cryptography [21]. To put it another way, the cipher text (at the receiver) can be defined as a message obtained using some type of encryption operation on plain text [22]. In order to convert plain text to cipher text, the encryption operation requires an algorithm and key [23]. Encryption algorithms play an important role in information security systems, and they are divided into two types: transposition and substitution, depending on whether the content of plain text changes at all or not [7]. Cryptography is classified into two basic types based on the key used for encryption and decryption [15], including symmetric cryptography and asymmetric cryptography [7].

a) *Symmetric key Cryptography*: It is also known as secret key or private key cryptography. It uses the same key for both plain-text encryption and cipher-text decryption [24] [25]. Some examples of symmetric cryptosystems are OTP (one time pad), DES (data encryption standard), AES (advanced encryption standard), etc. [7] [15].
b) *Asymmetric Key Cryptography*: Algorithms of the asymmetric type are used to encipher and decipher the message using different keys—for encryption, use public keys, and for decryption, use private keys [26] [27]. This type of algorithm is relatively slow, and for this reason, it makes it impossible to encipher large amounts of data. RSA, DH, and other asymmetric cryptosystems are some examples [7] [15].

2. Related Works

The enhancement of the DES algorithm was the important aim of several studies, some of which have been illustrated in this section.

In [28] the researchers improved the DES algorithm by increasing the key length (1024 bits) which is to be divided into 16 keys (64 bits each), each key is created independently for the various algorithm cycles. The results of the proposed algorithm were significantly better than the old algorithm in the detecting of the encryption key or the total number of keys that could be generated by following the blind search method (try all possible keys), Increasing the key length (degree of complexity) made it difficult to search in a vast space of numbers and attempts.

In [6], the double DES algorithm (2DES) is used to protect the security of information. The 2DES algorithm's implementations have a long execution time. The Message Passing Interface (MPI) library is used to implement the parallel 2DES algorithm. The results demonstrated that the run time of the parallel 2DES algorithm outperforms the sequential one. Moreover, on a large number of processors, parallel 2DES achieved better parallel efficiency. As a result, the parallel 2DES offered significantly better performance in terms of execution time than the serial ones and would be useful to apply to encrypt and decrypt multimedia.

In [29], the researchers proposed an enhanced Simplified DES (SDES) algorithm to protect the smart cards' data. It added complement and shift operations to the existing SDES algorithm. It offered higher security to protect the smart cards' data. The information is secured from any unauthorized access. This technique can be helpful for selecting the implementation of enhanced SDES for various applications. When compared to the SDES algorithm, the experimental results are better.

In [30], the researchers proposed modifications to enhance DES against brute force and cryptanalysis attacks. The proposed enhancement used the f-function by incorporating striding techniques instead of just performing the XOR operation between the 56-bit subkey and the plaintext. The security against the aforementioned attacks was enhanced. The enhanced DES had been evaluated and compared against the original algorithm using the avalanche effect. One-bit vibration in plain text caused an average of a 55% avalanche impact with enhanced DES.

In [25], the researcher suggested an improved structure for DES to make it secure and immune to attacks. The improved structure was accomplished using standard DES with a new way of using two key generations: one is simple, and the other is encrypted by using an improved Caesar algorithm. The encryption algorithm in the first 8 rounds uses simple key 1, and from round 9 to round 16, the algorithm uses encrypted key 2. The results of this research indicated an increase in DES security, performance, and complexity of search compared with standard DES.

In [31], the researchers proposed a key-based enhancement of the DES (KE-DES) technique for securing text. The KEDES algorithm is implemented by applying two steps: the first is merging the odd/even bit transformation of every key bit in the DES algorithm. The second step is replacing the right-side expansion of the original DES by using the Key-Distribution (K-D) function. The K-D allocation consists of 8 bits from the Permutation Choice-1 (PC-1) key outcome. The next 32-bit outcomes are from the right side of the data; there is also an 8-bit outcome from Permutation Choice-2 (PC-2) in each round. The key and data were created randomly. The results indicated that the enhancement provided adequate security, and the KEDES model is considered more efficient for text encryption.

In [32], the researchers increased the block size of the DES algorithm as well as the key size from 64 to 128 bits. The internal structure of the proposed mechanism ultimately changed all other components of DES accordingly. With this improvement, the anticipated scheme is more secure due to its robustness due to its confusion and diffusion components. The algorithm is applied to both plain text and images. The standard benchmark statistics have been performed to analyze the modified encryption scheme in order to authenticate the anticipated mechanism.

In [33], the researchers improved DES in the substitution step, which is the only nonlinear component of the algorithm. This alteration has been providing great outcomes and increasing the strength of DES. Accordingly, a novel 6 × 6 good-quality S-box construction scheme has been hired in the substitution phase of the DES. The construction involves the Galois field method and generates robust S-boxes that are used to secure the scheme against linear and differential attacks. Then again, the key space of the improved DES has been enhanced against brute-force attacks. The outcomes of different performance analyses depict the strength of the proposed substitution boxes, which also guarantees the strength of the overall DES.

3. Classical Cryptography Algorithms

This section explains several classical algorithms with a general view of each algorithm used in this research.

3.1 Vigenère Algorithm: This is a type of classical polyalphabetic substitution algorithm, i.e., each alphabet can be replaced by a different cipher alphabet [34] [35]. This algorithm uses a variety of Caesar ciphers, depending on the keyword letters, to encrypt the alphabetic text [36]. The strong point of the Vigenère algorithm is that it is not exhibition-to-frequency analysis because the cipher text passes through different shift operations, so the same plain-text character will not always be encrypted to the same cipher-text character. If the integers 0 to 25 are substituted for the letters A through Z, the Vigenère algorithm can be seen algebraically as follows:

$$Ci = (Pi + Ki) \bmod m \tag{1}$$

where, C_i = character of cipher text, P_i = character of plain text, K_i = character of key phrase, and m = alphabet length [7]. A key length must be less than the plain-text length. Characters from the same character set must be used in both the key and the text to be encoded [37]. For more illustration, see the following example for Vigenère encryption according to the above formula: If the plain text is (Vigenère is Type of Classical Polyalphabetic Substitution) and the key is (Vigresearch), then the encrypted text is (Qqmvrwve zu Atxk fj Upajupxir Gsdcacrovjkkmu Wusuadbakmgr) [38].

3.2 Porta Algorithm: This is a type of classical substitution cipher. This algorithm is similar to the Vigenère algorithm; the main distinction between this algorithm and the Vigenère algorithm is the use of 13 pairs of alphabets as keys as opposed to the individual usage of all 26 alphabets [39]. This will allow two different key characters to produce identical encrypted text for a given plain-text character. As seen in Table 1, the Porta algorithm encrypts plain text using the matrix [37].

Table 1: Porta Matrix [38]

Key	Substitution Alphabet
A, B	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
C, D	A B C D E F G H I J K L M Z N O P Q R S T U V W X Y
E, F	A B C D E F G H I J K L M Y Z N O P Q R S T U V W X
G, H	A B C D E F G H I J K L M X Y Z N O P Q R S T U V W
I, J	A B C D E F G H I J K L M W X Y Z N O P Q R S T U V
K, L	A B C D E F G H I J K L M V W X Y Z N O P Q R S T U
M, N	A B C D E F G H I J K L M U V W X Y Z N O P Q R S T
O, P	A B C D E F G H I J K L M T U V W X Y Z N O P Q R S
Q, R	A B C D E F G H I J K L M S T U V W X Y Z N O P Q R
S, T	A B C D E F G H I J K L M R S T U V W X Y Z N O P Q
U, V	A B C D E F G H I J K L M Q R S T U V W X Y Z N O P
W, X	A B C D E F G H I J K L M P Q R S T U V W X Y Z N O
Y, Z	A B C D E F G H I J K L M O P Q R S T U V W X Y Z N

For more illustration, see the following example: if the plain text= (Porta Algorithm is Similar to Vigenère algorithm) and the key= (porresearch), then the encrypted text= (Jimby Rwtjfsanr nh Btznxxl aj Dtxpawfo tryjgziur).

3.3 Autokey Algorithm: This is a classical substitution algorithm. This algorithm is like Vigenère, i.e., the only difference is that the keyword is developed by contacting the keyword at the beginning of the plain text [40] [41]. The autokey algorithm depends on the Vigenère algorithm except for the problem of keyword repetition periodically [37] [42]. For example, if you have the plain text = (Autokey Algorithm is like Vigenère Algorithm) and the key = (Autresearch), then after applying the Autokey algorithm, the encrypted text will be = (Aomfowc Acivrcmvw mq ltqs Mqzlzmjp Ivkjzoxuq).

3.4 Beaufort Algorithm: This algorithm, from the type of classical substitution, the Beaufort cipher, operates essentially identically to Vigenère encryption [14]. This algorithm is based on a matrix, as shown in Table 2 [43]. It looks for a plain-text letter in the matrix's first row before beginning a search for a keyword letter in a specific column. Track down the leftmost letter in that row, which is the encryption for the provided plain-text letter, when you've located it [37]. For example, if the plain text (Beaufort Cipher operates like Vigenère encryption) and key (Beresearch) are generated, then the cipher text will generate (Aarknqjy Azmxnn epwacoxm gwia Fjwdoaaa oryaesiwdr).

Table 2: Beaufort Matrix [38]

		Text	
		A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	
Key	**A**	A Z Y X W V U T S R Q P O N M L K J I H G F E D C B	**Geheimext**
	B	B A Z Y X W V U T S R Q P O N M L K J I H G F E D C	
	C	C B A Z Y X W V U T S R Q P O N M L K J I H G F E D	
	D	D C B A Z Y X W V U T S R Q P O N M L K J I H G F E	
	E	E D C B A Z Y X W V U T S R Q P O N M L K J I H G F	
	F	F E D C B A Z Y X W V U T S R Q P O N M L K J I H G	
	G	G F E D C B A Z Y X W V U T S R Q P O N M L K J I H	
	H	H G F E D C B A Z Y X W V U T S R Q P O N M L K J I	
	I	I H G F E D C B A Z Y X W V U T S R Q P O N M L K J	
	J	J I H G F E D C B A Z Y X W V U T S R Q P O N M L K	
	K	K J I H G F E D C B A Z Y X W V U T S R Q P O N M L	
	L	L K J I H G F E D C B A Z Y X W V U T S R Q P O N M	
	M	M L K J I H G F E D C B A Z Y X W V U T S R Q P O N	
	N	N M L K J I H G F E D C B A Z Y X W V U T S R Q P O	
	O	O N M L K J I H G F E D C B A Z Y X W V U T S R Q P	
	P	P O N M L K J I H G F E D C B A Z Y X W V U T S R Q	
	Q	Q P O N M L K J I H G F E D C B A Z Y X W V U T S R	
	R	R Q P O N M L K J I H G F E D C B A Z Y X W V U T S	
	S	S R Q P O N M L K J I H G F E D C B A Z Y X W V U T	
	T	T S R Q P O N M L K J I H G F E D C B A Z Y X W V U	
	U	U T S R Q P O N M L K J I H G F E D C B A Z Y X W V	
	V	V U T S R Q P O N M L K J I H G F E D C B A Z Y X W	
	W	W V U T S R Q P O N M L K J I H G F E D C B A Z Y X	
	X	X W V U T S R Q P O N M L K J I H G F E D C B A Z Y	
	Y	Y X W V U T S R Q P O N M L K J I H G F E D C B A Z	
	Z	Z Y X W V U T S R Q P O N M L K J I H G F E D C B A	

3.5 Trithemuis Algorithm: This algorithm is close to the Vigenère algorithm; it is also a classical substitution algorithm, but the Vigenère algorithm was developed before this one. It can be thought of as a Vigenère cipher in theory using the constant key (ABCDEFGHIJKLMNOPQRSTUVWXYZ) [38] [44]. See the following example: If the plain text= (Trithemuis Algorithm is also like Vigenère Algorithm), then the resultant encrypted text is: (Tskwljsbqb Kwsbfxjye bm vhpm kilg Ymlkumao Lxtcgykzf).

3.6 Gronsfeld Algorithm: This classical substitution, with the fact that keys can only be numbers, means that the cipher of this algorithm is the same as the Vigenère cipher. Table 3 shows the key matrix of Gronsfeld [38] [44] and shows that there are only 10 possible Caesar ciphers that can be utilized, as opposed to 26.

Table 3: Gronsfeld Matrix [38]

		Cleartext	
		A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	
Key	**0**	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	**Ciphertext**
	1	B C D E F G H I J K L M N O P Q R S T U V W X Y Z A	
	2	C D E F G H I J K L M N O P Q R S T U V W X Y Z A B	
	3	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C	
	4	E F G H I J K L M N O P Q R S T U V W X Y Z A B C D	
	5	F G H I J K L M N O P Q R S T U V W X Y Z A B C D E	
	6	G H I J K L M N O P Q R S T U V W X Y Z A B C D E F	
	7	H I J K L M N O P Q R S T U V W X Y Z A B C D E F G	
	8	I J K L M N O P Q R S T U V W X Y Z A B C D E F G H	
	9	J K L M N O P Q R S T U V W X Y Z A B C D E F G H I	

For instance, if the plain text= (Gronsfeld Algorithm has kay as numbers) was encoded using the key= (31415), the result would be the cipher text= (Jssoxifpe Fohssnwiq ifv lez fv oynghsw). Although there are much fewer possible mappings in the Gronsfeld cipher, the security is virtually identical to that of the Vigenère encryption. The Vigenère cipher can be mapped in 26x26x26 ways with a key length of 3; however, the Gronsfeld cipher can only be mapped in 10x10x10 ways [7].

4. Data Encryption Standard Algorithm (DES)

DES is a symmetric-key encryption scheme where the same secret key is used to encrypt and decrypt messages [45] [46]. Additionally, it is a block cipher, which means it operates on the blocks of a plaintext input message of fixed length (64-bit) and processes using the key. It then transforms the plain text through a complicated 16-round (i.e., permutation) operation to produce ciphertext of the same length [47]. The number of rounds is 16, perhaps to guarantee the elimination of any correlation between the cipher text and either the plain text or key. All blocks are numbered from left to right, which makes eight bits in each byte. Four fundamental operations—XOR, shift, LUT (look up table), and permutation—are needed to implement DES. A brief explanation of the DES structure is depicted in

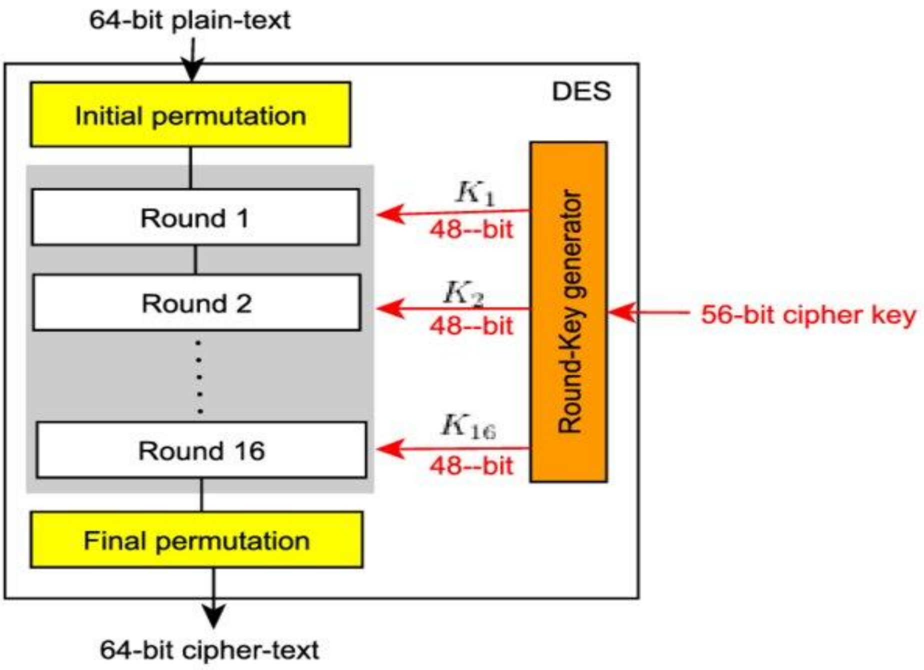


Figure 2: Structure of the DES Algorithm

The DES encryption and decryption procedures are essentially identical, with the exception that the cipher text is used as the input for the DES algorithm and the keys are used in reverse. Although the DES algorithm is almost impossible to break, some critical analysis has theoretically demonstrated its weaknesses. Since it has been publicly known as a standard of encryption, DES is no exception; hackers have taken advantage of its weaknesses to sneak secure encryption and steal critical information. The key length (56 bits) of DES security is one of the main issues. Intruders have developed attacks that they can use against it. Brute force (exhaustion attack), differential cryptanalysis, and linear cryptanalysis are attacks that have been known to successfully compromise DES security. The DES algorithm's key generation makes it weak [14].

To illustrate how the DES algorithm works in a simplified way, according to the following steps:

- 64-bit plain-text block forward to the function of initial permutation (IP).
- Perform IP on plain-text.
- Divide 64-bit plain-text block into two halves known as left plain- text (LPT) and right plain-text (RPT).
- All LPT blocks and RPT blocks are encrypted 16 times. This step includes five phases: key transformation, expansion permutation, S-Box permutation, P-Box permutation, XOR, and swap.
- Merge LPT and RPT blocks, then perform final permutation (FP) on the resulted block.
- The result represents 64-bit cipher ext.

5. Analysis Tools

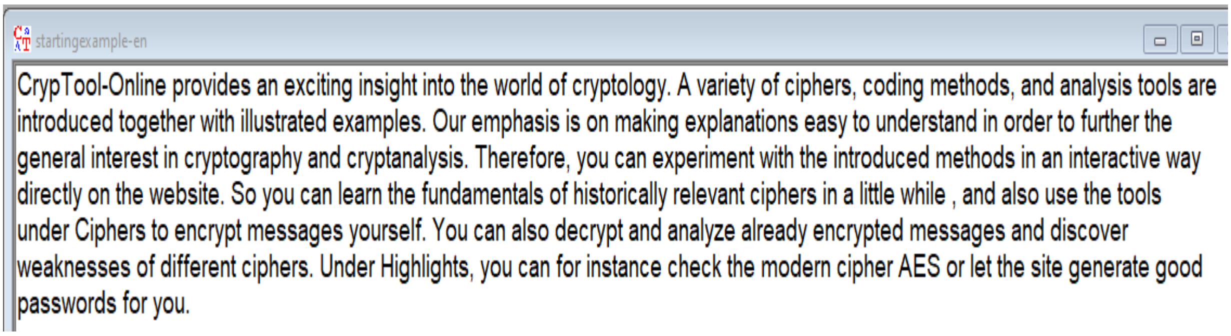
There are several types of analysis tools that can be used to analyze the efficiency and complexity of the security level against the attacker. Some of these tools are illustrated below and [48] [49]:

- *Entropy*: This tool of analysis includes collecting randomness used in cryptography or other applications. The security level is affected by the entropy value, so when the entropy value decreases, that leads to lower complexity and a lower security level [50].
- *Histogram*: This tool is used to group data into ranges, and every range is represented graphically by a vertical bar. The character appears on the horizontal axis, and the amount of appearance for each character is shown on the vertical axis. Histogram is also called frequency.
- *Autocorrelation:* This tool is used to show the autocorrelation of a specific text, which means the number of characters matched in the text. It can be helpful to relate each character to different points.

6. Experimental Results and Discussions

In this research, the experiments were divided into two parts:

6.1 Part 1: In this part, six algorithms of the classical type were applied, where a comparison of efficiency levels was made among these algorithms, in addition to a comparison of security levels according to complexity for confronting the attacker and making breaking the code as difficult as possible. Therefore, this research includes a comparative study of the performance of these six algorithms. The above algorithms were applied to the same plain text as shown in Figure 3.



CrypTool-Online provides an exciting insight into the world of cryptology. A variety of ciphers, coding methods, and analysis tools are introduced together with illustrated examples. Our emphasis is on making explanations easy to understand in order to further the general interest in cryptography and cryptanalysis. Therefore, you can experiment with the introduced methods in an interactive way directly on the website. So you can learn the fundamentals of historically relevant ciphers in a little while , and also use the tools under Ciphers to encrypt messages yourself. You can also decrypt and analyze already encrypted messages and discover weaknesses of different ciphers. Under Highlights, you can for instance check the modern cipher AES or let the site generate good passwords for you.

Figure 3: Plain-Text Cryptool S/W Interface

In addition, the encryption algorithms were dealt with under the same set of characters, as it included uppercase and lowercase English letters (52 letters), and the implementation was done using the s/w Cryptool 1.4.4.2. Table 4 illustrates the ciphertext for each algorithm used.

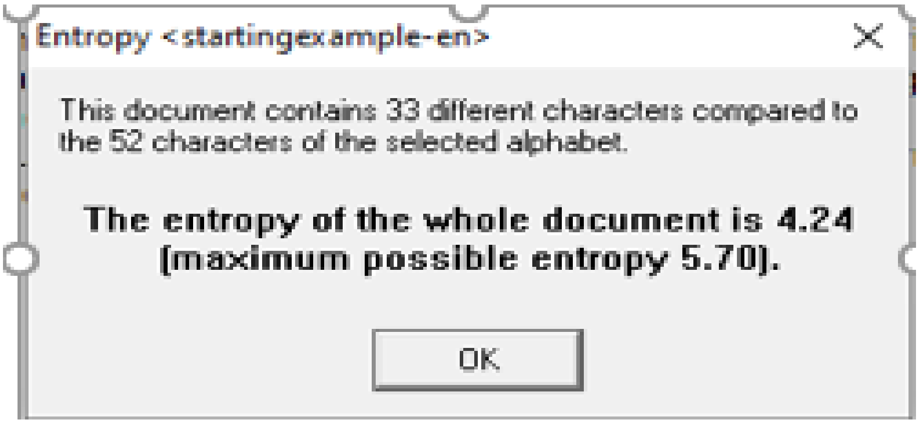
Table 4: Cipher Text for Six Algorithms

Algorithm Name	Cipher-Text
Vigenère	WdheHjdp-Gczpmv efjkzype um vmonizfr wgezrtb tjld fos fdfqo kn ndebcdxjrh. V kokwvim jq tqevydb, ncvtjo xsmvkog, scu vcoskbtg bdkth oks zchmdumnsx fkrqbsvj lwmv zwxchcjyfyr vmzrepzh. Aad vxdzybqh wl aj xzptjo pjizrczbtkfh sueh ic ccuzgemojo ui daypd ma wjfbsvj ivy uvcqmyp qcfydvhh nc tjabmaxgzksh vcr wdhehscrtaepe. Csqmpwggs, ead nzi pghpdpnvch etcp ivy wjifjodxpr tscscvh zf ym pmcpfsncqks coh oumptlwk hm csq epsktfy. Ek acc nrf wsudj itw qdfootsjizqh kn swlfkguuypta dyzvkzii tqevydb tb s wzlizy iytxw , yjy yzla dhq bsv ldase dcpwg Tqevydb ic wctjabm nvhgsrvk aaadbpxx. Akm nog ophc vptjabm ojo ziypdbs uzapzva vfndebcpp rpbkyuye rcp vtbxdhyd fpzpcvkhsl aw ouxqvjpmm qzetwgb. Mcryd Ytszwzosfl, kkj osc wgg wgecybup tppqr fyp yjovjc qpbypf SPB gg zyf csq atcz rsgsayhw rkgo bueblcmob ndd ead.
Porta	WdheHjdp-Gczpmv efjkzype um vmonizfr wgezrtb tjld fos fdfqo kn ndebcdxjrh. V kokwvim jq tqevydb, ncvtjo xsmvkog, scu vcoskbtg bdkth oks zchmdumnsx fkrqbsvj lwmv zwxchcjyfyr vmzrepzh. Aad vxdzybqh wl aj xzptjo pjizrczbtkfh sueh ic ccuzgemojo ui daypd ma wjfbsvj ivy uvcqmyp qcfydvhh nc tjabmaxgzksh vcr wdhehscrtaepe. Csqmpwggs, ead nzi pghpdpnvch etcp ivy wjifjodxpr tscscvh zf ym pmcpfsncqks coh oumptlwk hm csq epsktfy. Ek acc nrf wsudj itw qdfootsjizqh kn swlfkguuypta dyzvkzii tqevydb tb s wzlizy iytxw , yjy yzla dhq bsv ldase dcpwg Tqevydb ic wctjabm nvhgsrvk aaadbpxx. Akm nog ophc vptjabm ojo ziypdbs uzapzva vfndebcpp rpbkyuye rcp vtbxdhyd fpzpcvkhsl aw ouxqvjpmm qzetwgb. Mcryd Ytszwzosfl, kkj osc wgg wgecybup tppqr fyp yjovjc qpbypf SPB gg zyf csq atcz rsgsayhw rkgo bueblcmob ndd ead.
Autokey	OpqtVfse-Yrjkec ekcjtrrd ia imtwoqqk anfmdjb bvgu buw euyel by qkftpcczjm. F xrpxxhj cl aikhvzw, vmrnpo blxygfg, dvq gzeefglk tbrlf acc avlkcrfueu xwtxkvhl ymwa wrpnzxiwbxk milghevs. Hyu ijptpdmk wm fr yprifo wfhznzadqbtw bpdy go nvrrjwtslw wh buhvj mo sxzgvvu xyx ujhvkhp zgaixifx zn nzlixfkjtxua rls vfegtpuyllvkj. Rwxrrfzpw, ggn jee icdvvgaypt jmqw xym urgkklnjxk qmgafrv cp eq urmlfdubvvr enr hzrgvbgc kn rkm nidltrs. Fh fsq gbf txejb rvy huaoemvamhpx is kiexbkinszqf zwesmipt ntnyicw dn n ekbisi nzqye , lvw twwk bap xhr wozdg ofhxy Gbdvpjm gr ieeznwx dwlgetgj wdndwwdf. Esm aoh rdwz icqlapg ayv oqenpxt tlehaqy plbvyakid pcwfcxch trg pmkuobij wrdnvwugzw fb hipsijwrl qnspjww. Lrqxt Pxnlcaauww, pvc ihy nuy bfqhupcr hvvkx lae zqhgyr esioid OHW fe nmi alv smls xprxkhxw ohsj tnwjwhvjg trg ygm.
Beaufort	KhupJdqi-Wrnelo pldjlhag ml ohajllxy qzgkyvy wgrq ffu wqlgb ff chojzqrdyv. K jyvqole dz rcpriha, coowge sutrebk, rrq krybaawk yqfzm yvu krjaqqqcuj feyyyxpt iqtr ktrxmatefiv ohcfpigm. Ksh osnkebcm qu kf schwge abxnsrcywfxm umgu lo xrqgngtyfb ue qchah tk nklyxpt lri soryaei crfihomj jr rtgjtkmnccxv krv khupjrrtzggeg. Zxyaaownu, oky cce awvahemorj vwad lri qflldbziav auzxoom lx el elzalrcacju qyu buaarrta yl zxy vasswfi. Ge gox ctx tumhf lvn zzxbyauflcgm ff xqufenupeizg hinojcel rcpriha wp r tlrlni clwrn , egh enuk ymy yxp rqkbg yrznn Rcpriha lo nrrtgjt momkryps gkshaarm. Gfq cyz yhmo oartgjt yfb ceeimfu mnbacog pxchojzaz fabsesig srz owbiqdih wachrpsmuu kn bumzptalt wkpvnnb. Qrvih Lwwktlexfu, aek arr own qzgzeppa rdawc fla qdbptr wejlal RAB wn nif zxy zwag yuzubejn yfwb jmgaioabb fqh oky.

Trithemuis	CsasXtus-Wwvtzr dgemawyn wk cwcjvlrl ouarqsf vbie kzx qjnib nf dtbtyuswpi. L hnfxukq hz xemfdrt, erhntn undsaqg, pdu sgugupgr tpqow fxl qwdcaqiruu lhazpecq wjvk mqrbacblfrr tnreifzo. Lsq enrkexoz qb yy ynyxdx wqjgwkysippv ifyf bx eyprfhjrfw ci kobdr uq iywzoma dsq tscuise cipbpdsu kq gwewbxqcmcvn qev vltlqymamavmx. Zomaoqaes, nel uth ztmcqingqx boap crp uahgeumvyy ibrgoeu lr ft pvcocmphxlv ots yeocbtma rr ynl enldugs. He pgn wvj iczro vki kauljwpzgoai fx acnplphcbnoc wksmekyf pwexvjl ci w igstmg zlnrl , iwn lxfc jiv lay okljr uofhv Howpnbd fb scsiqin hapqzgfu bszxzmup. Jah qpd rdli yazpxpu cqh fththjp myftquq xhxnvnsee ohwxgnmb kyp qwhsfnxl raximetuhw tl kqoppdrbi szhaymo. Rlces Jlkmrpoqdd, kbi rqe xhl djprzndg fljir bqo xaqsgd taibzn XCR os nhx ynl ardp srbthrlx ajka nzstyrviy mwa izg.
Gronsfeld	FscqYxqr-Trojrf uaqbnhht eo jgeoymqh moxriny mqus umn yuwpg pj dwhrztprhc. B ajtojxb pj dnyjkww, fphjsp okylrew, bsm ctfpbtmt yxqrx euf moyaqjzghe xplnvnjv zjxi nunaxxubxfi nzgrtofw. Pza gsuldtmt nb qt renjrh jgrrfrdumpsb ggxc wp yointyyeqe mo tafkw xr gysyqgx ylh hiojacr nrwfvfxc kt hvbqxplacvmc doh dwhrzfrdmctnb. Vnjvhgssj, hqa heq fbqjaksjrw xmum cjk nrwssezlgj riwisex rp gs mquisflvoai zbc enagiypb pr umn ykgwlui. Tt hqa heq mibww vnj jxohbrnpzfpv pj inbvuwmfbpmd agrjzdox dnyjkww lo e mncvrj akjpf , fwf gqwr vwf yqg ztsot yoint Intkfvt yx gthvbqx njbugliv zsvwbgrk. Crv gbs jnyt hhdvzuc cti eqbpzen crwidec fslteuxhe qfxbcmjw doh enbeuaiu xibpwgyxiv pj enohkwiqu gjuqgxx. Yqeis Mrinqmjixt, dxw ifr ipv jsbvgsgh dlfht vnj qreiss lkvmiu BIT ta nky xkf wjyn iksiubxf lxqj uevtapwmu ltv bpy.

To evaluate the level of efficiency for each algorithm and analyze the results, three types of analysis tools were used (entropy, histogram, and autocorrelation). Figure 4 shows the entropy value for an algorithm using the entropy analysis tool. Figure 5 shows the entropy value of the cipher text for each algorithm. Table 5 shows how effective the algorithms were by looking at the relationship between the entropy value and the number of letters in the cipher text.

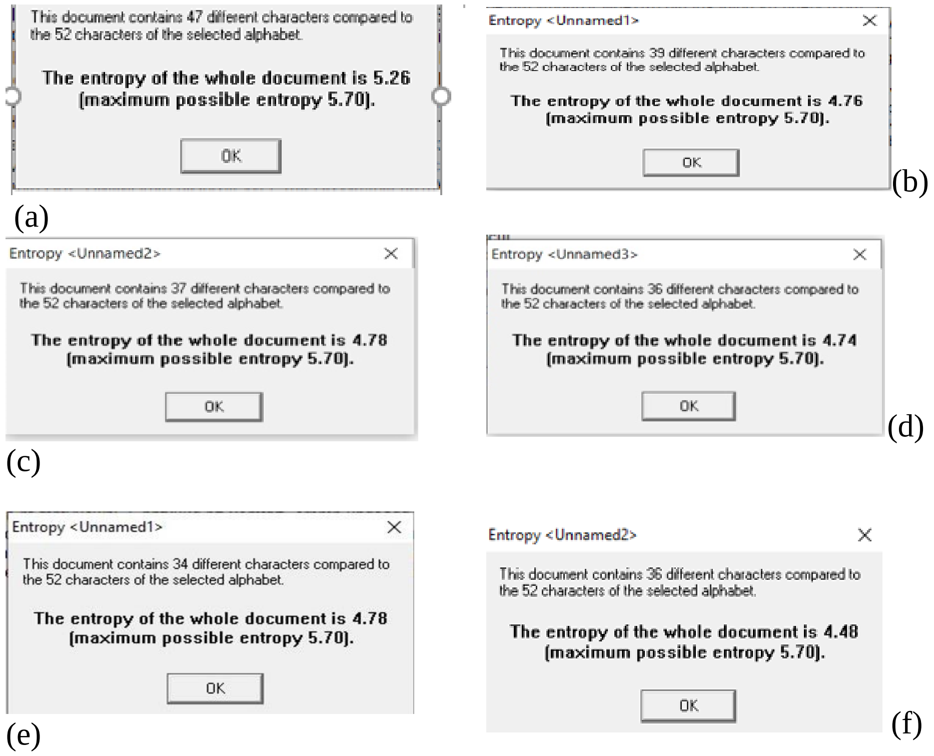
Where the most diversity in the content of the cipher text is preferred, so that the cipher text contains more different characters, this leads to an increase in the complexity of the cipher text against the attacker, and thus the algorithm is more efficient. From the above, it is clear that the Vigenère algorithm got the highest percentage of entropy (5.26), and with the highest variety of letters up to 47, it is the strongest and best among the six algorithms used.



This document contains 33 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.24 [maximum possible entropy 5.70].

Figure 4: Entropy of Plain-Text



(a)

This document contains 47 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 5.26 (maximum possible entropy 5.70).

(b)

Entropy <Unnamed1>

This document contains 39 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.76 (maximum possible entropy 5.70).

(c)

Entropy <Unnamed2>

This document contains 37 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.78 (maximum possible entropy 5.70).

(d)

Entropy <Unnamed3>

This document contains 36 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.74 (maximum possible entropy 5.70).

(e)

Entropy <Unnamed1>

This document contains 34 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.78 (maximum possible entropy 5.70).

(f)

Entropy <Unnamed2>

This document contains 36 different characters compared to the 52 characters of the selected alphabet.

The entropy of the whole document is 4.48 (maximum possible entropy 5.70).

Figure 5: Entropy for Six Used Algorithms (a) Vigenère (b) Porta (c) Autokey (d) Beaufort (e) Trithemuis (f) Gronsfeld

Table 5: Entropy Values Relative to the Number of Characters

Algorithm Name	Entropy of whole Document	Number of Characters
Vigenère	5.26	47
Porta	4.76	39
Autokey	4.78	37
Beaufort	4.74	36
Trithemuis	4.78	34
Gronsfeld	4.48	36

Figure 6 shows how to distribute the percentage of appearance of each letter in the plain text using the histogram analysis tool.

In addition to that, the same tool was used to count how many times each letter appears in the plain text. The same tool was also used to count how many times each letter appears in the encrypted message, using the algorithm shown in Figure 7(a–f). It is very clear through the diagrams in this table that the best algorithm is the one that makes the cipher text contain a number of characters with more variety and also the number of times the letters appear with close values so that it is difficult for the attacker to guess and break the cipher text, and this is what we notice clearly in the performance of the Vigenère algorithm.

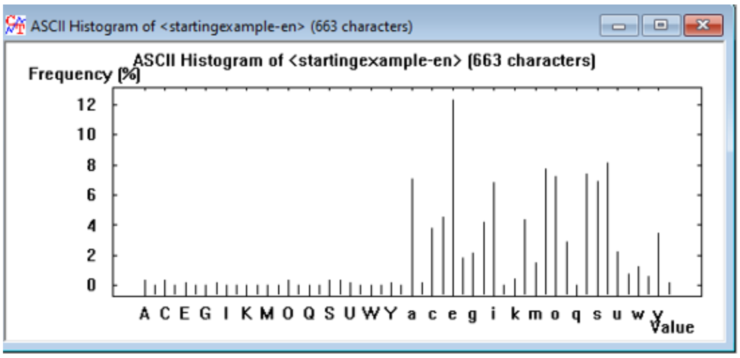
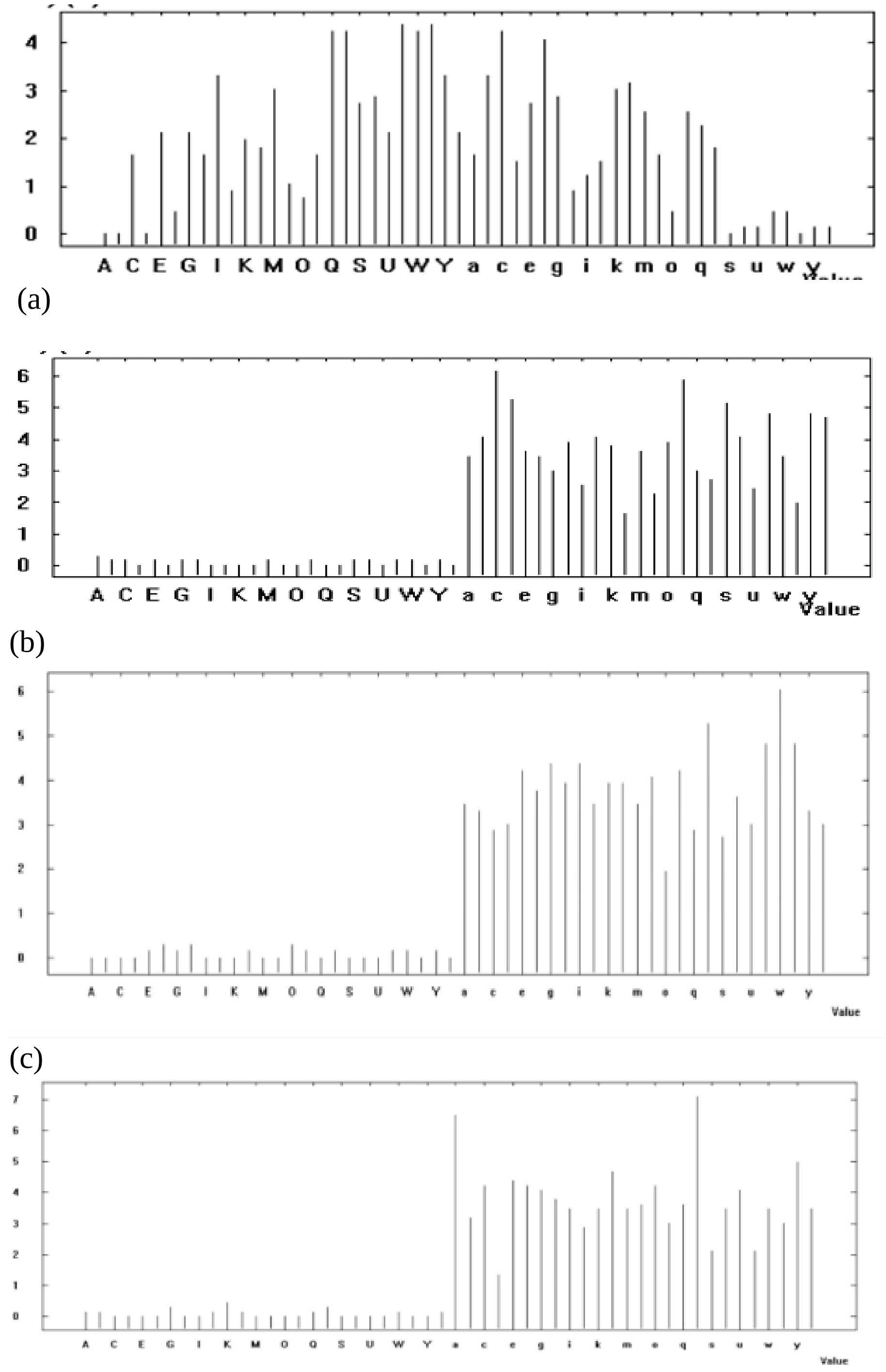

ASCII Histogram of <startingexample-en> (663 characters)
ASCII Histogram of <startingexample-en> (663 characters)
Frequency (%)
12
10
8
6
4
2
0
A C E G I K M O Q S U W Y a c e g i k m o q s u w y
Value

Figure 6: Histogram of Plain Text

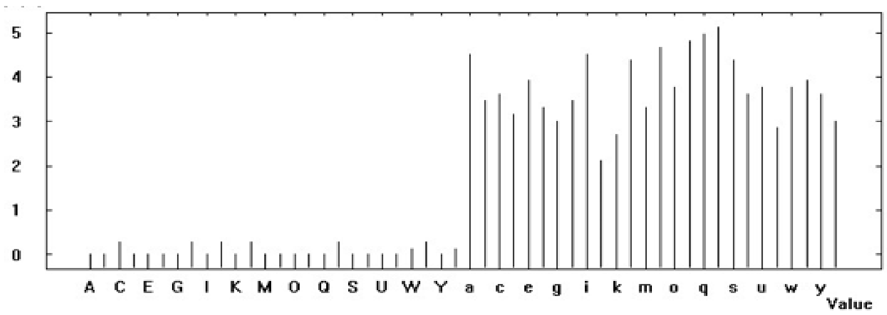


(a)

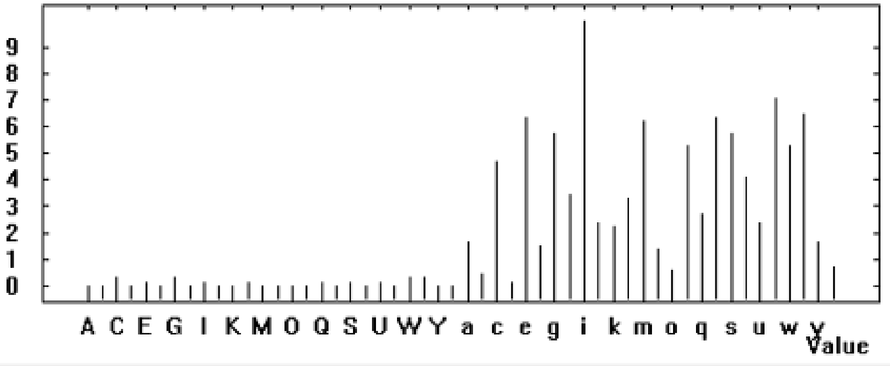
(b)

(c)

(d)



(e)



(f)

Figure 7: Histogram of Six Algorithms (a) Vigenère (b) Porta (c) Autokey (d) Beaufort (e) Trithemuis (f) Gronsfeld

Finally, as shown in Figure 8, that shows the diagram of autocorrelation for the plain text. In the same context, Figure 9 shows the diagrams of autocorrelation for each algorithm used in this research. It shows that the lower the percentage of correlation between the letters in a given block, the better the efficiency of the algorithm.

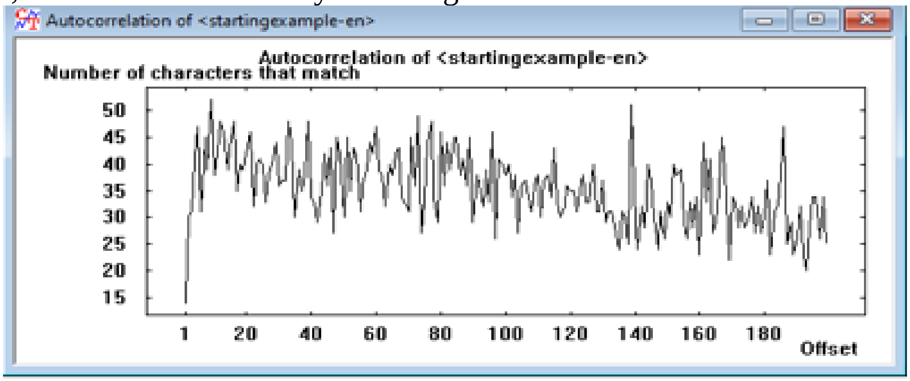
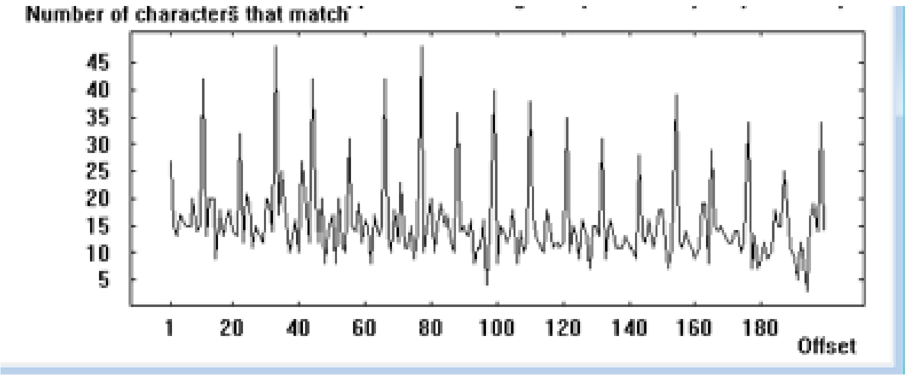
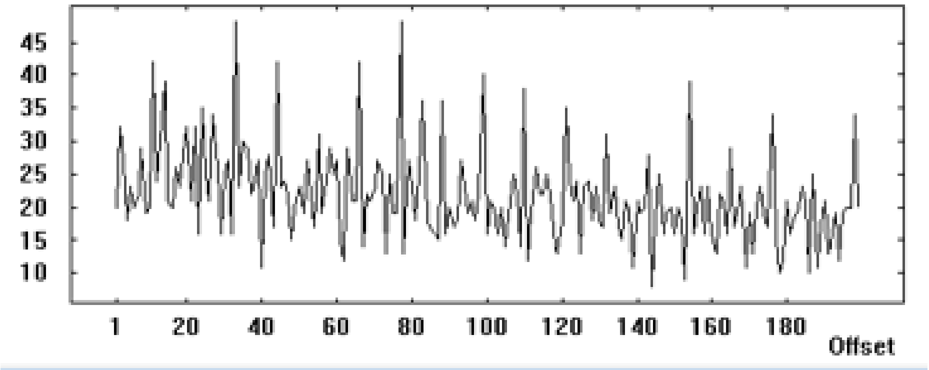

Autocorrelation of <startingexample-en>
Autocorrelation of <startingexample-en>
Number of characters that match
50
45
40
35
30
25
20
15
1
20
40
60
80
100
120
140
160
180
Offset

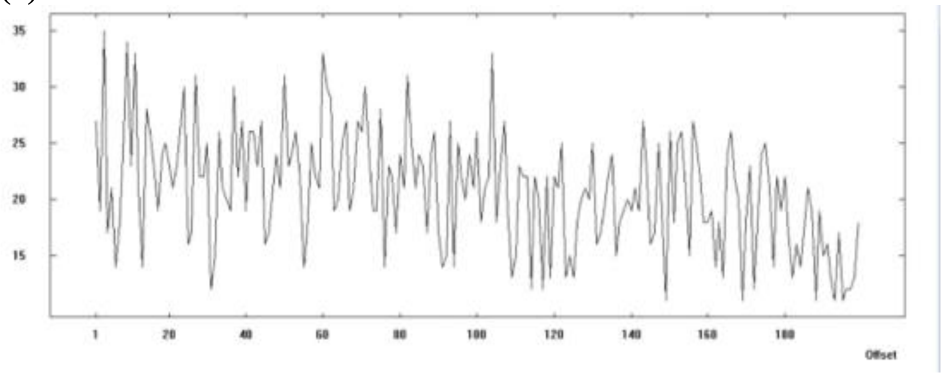
Figure 8: Autocorrelation of Plain Text



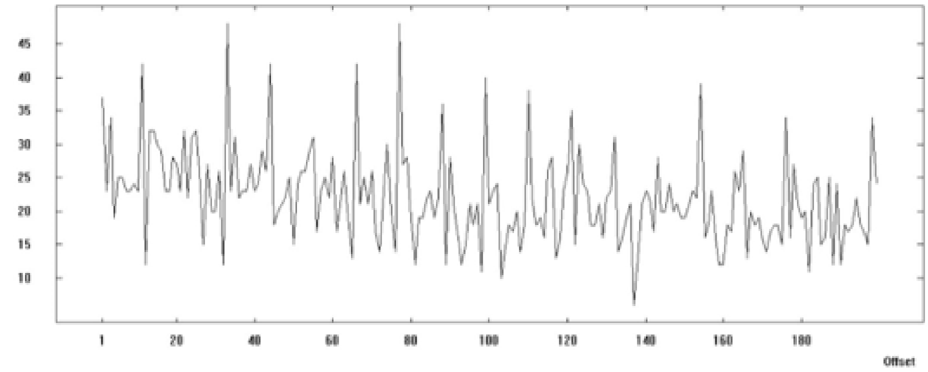
(a)


45
40
35
30
25
20
15
10
1
20
40
60
80
100
120
140
160
180
Offset

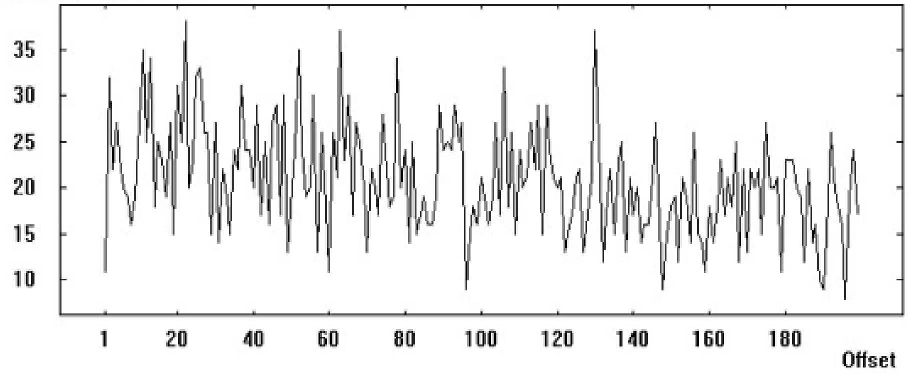
(b)


35
30
25
20
15
1
20
40
60
80
100
120
140
160
180
Offset

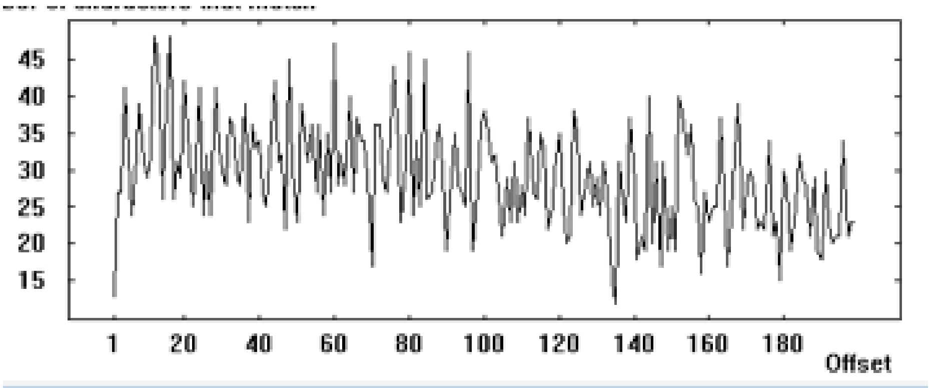
(c)


45
40
35
30
25
20
15
10
1
20
40
60
80
100
120
140
160
180
Offset

(d)


35
30
25
20
15
10
1
20
40
60
80
100
120
140
160
180
Offset

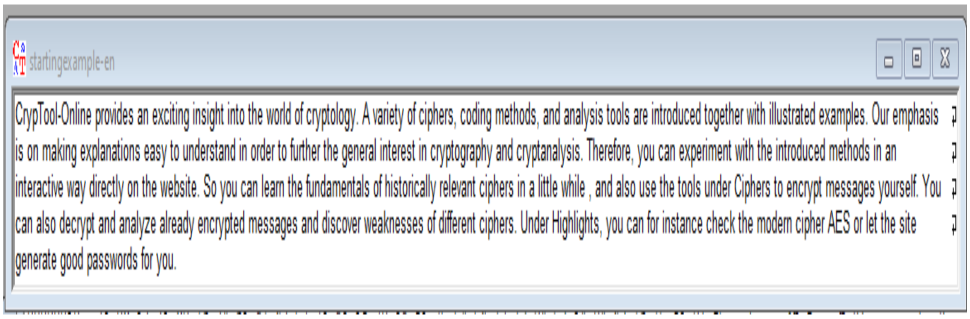
(e)



(f)

Figure 9: Autocorrelation of Plain-Text (a) Vigenère (b) Porta (c) Autokey (d) Beaufort (e) Trithemuis (f) Gronsfeld

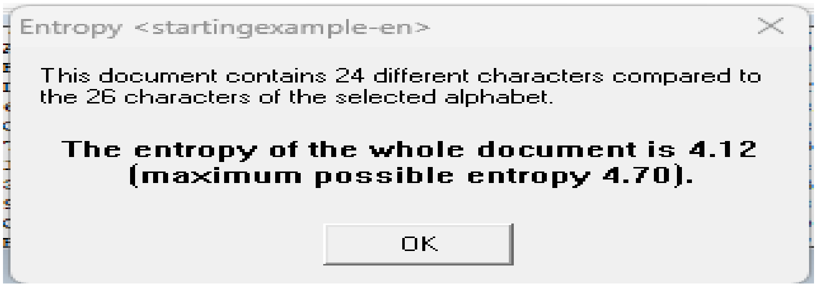
6.2 Part 2: In this part, the results of the first part are confirmed by clarifying the efficiency and positive effect of the Vigenère algorithm. Two hybrid ciphering models are designed; each model includes an experiment, and each experiment includes three cases. All these experiments were applied based mainly on the plain text shown in Figure 10.



CrypTool-Online provides an exciting insight into the world of cryptology. A variety of ciphers, coding methods, and analysis tools are introduced together with illustrated examples. Our emphasis is on making explanations easy to understand in order to further the general interest in cryptography and cryptanalysis. Therefore, you can experiment with the introduced methods in an interactive way directly on the website. So you can learn the fundamentals of historically relevant ciphers in a little while , and also use the tools under Ciphers to encrypt messages yourself. You can also decrypt and analyze already encrypted messages and discover weaknesses of different ciphers. Under Highlights, you can for instance check the modern cipher AES or let the site generate good passwords for you.

Figure 10: Plain Text

Figures (11–13) follow this to sequentially illustrate the plain-text entropy value, histogram, and autocorrelation.



This document contains 24 different characters compared to the 26 characters of the selected alphabet.

The entropy of the whole document is 4.12 (maximum possible entropy 4.70).

Figure 11: Entropy of Plain Text

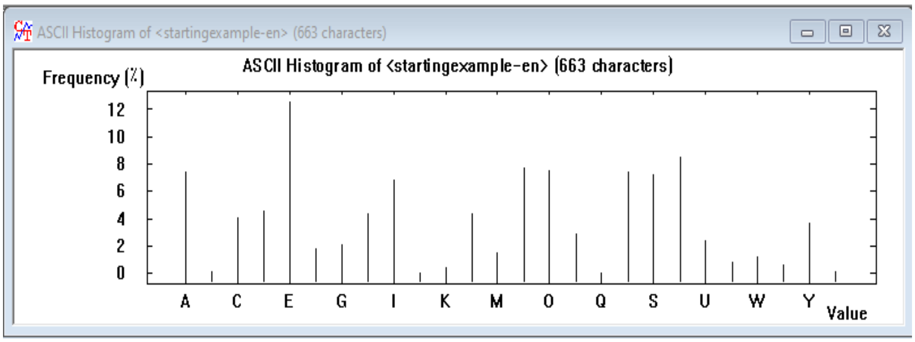


Figure 12: Histogram of Plain Text

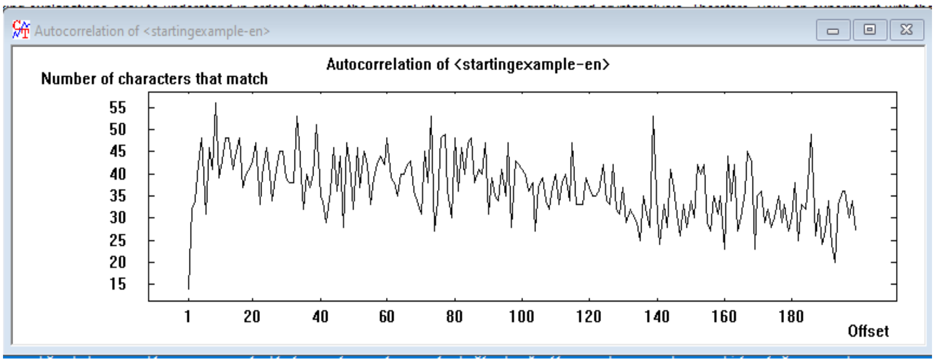


Figure 13: Autocorrelation of Plain Text

A- First Experiment

This experiment includes three cases, as follows:

Case 1: In this case, use only the Vigenère algorithm to encrypt plain text.

Case 2: In this case, use only the ECB-DES algorithm to encrypt plain text.

Case 3: In this case, use Vigenère and then ECB-DES algorithms as multilevel to encrypt plain text.

For more illustrations of the first experiment, see Figure 14(a–c).

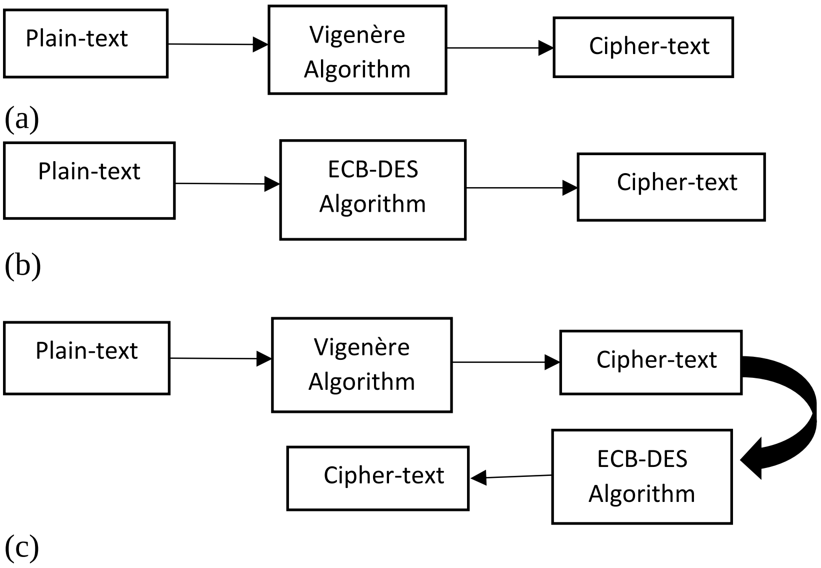
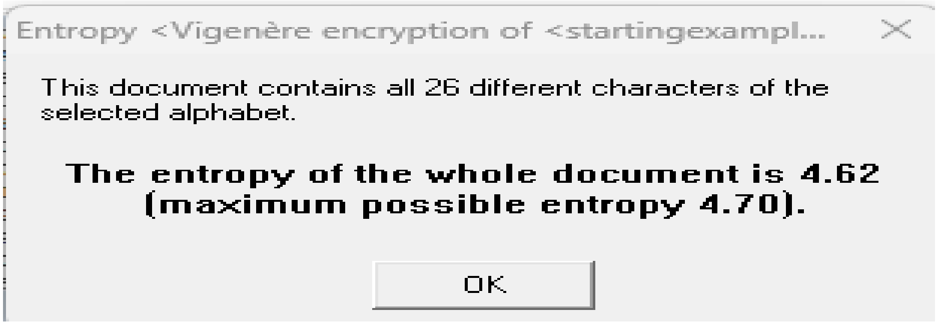


Figure 14: First experiment (a) Vigenère Ciphering (b) ECB-DES Ciphering (c) Vigenère, then ECB-DES Ciphering (Hybrid)

See Figures (15–17) to illustrate the value of the entropy, histogram, and autocorrelation sequentially for three cases.

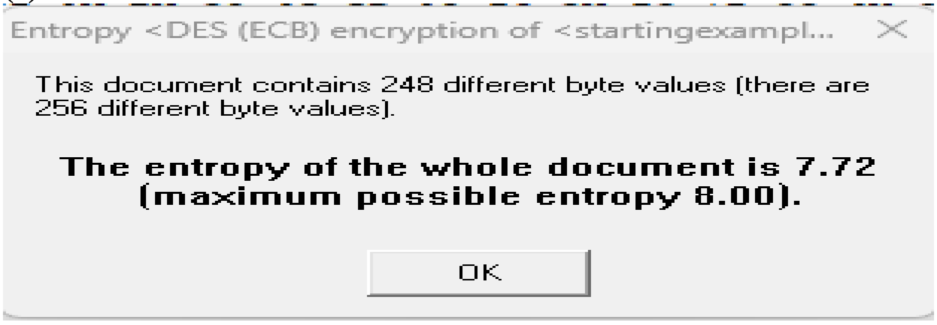


Entropy <Vigenère encryption of <startingexampl...

This document contains all 26 different characters of the selected alphabet.

The entropy of the whole document is 4.62 (maximum possible entropy 4.70).

(a)

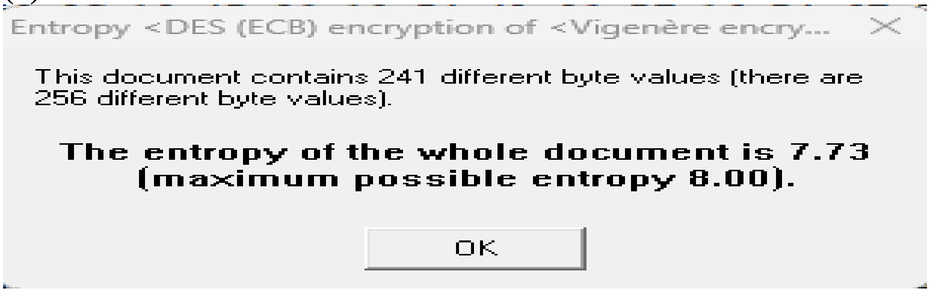


Entropy <DES (ECB) encryption of <startingexampl...

This document contains 248 different byte values (there are 256 different byte values).

The entropy of the whole document is 7.72 (maximum possible entropy 8.00).

(b)



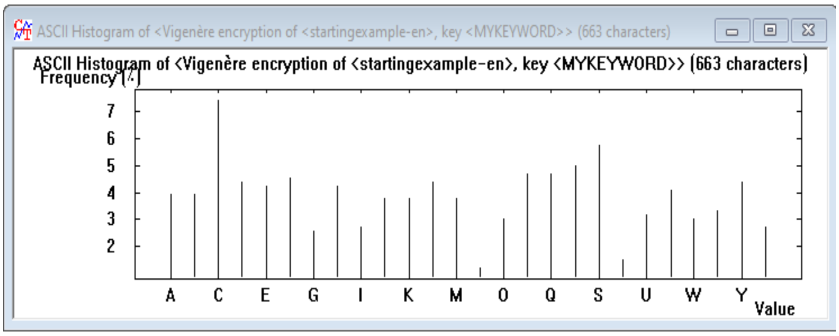
Entropy <DES (ECB) encryption of <Vigenère encry...

This document contains 241 different byte values (there are 256 different byte values).

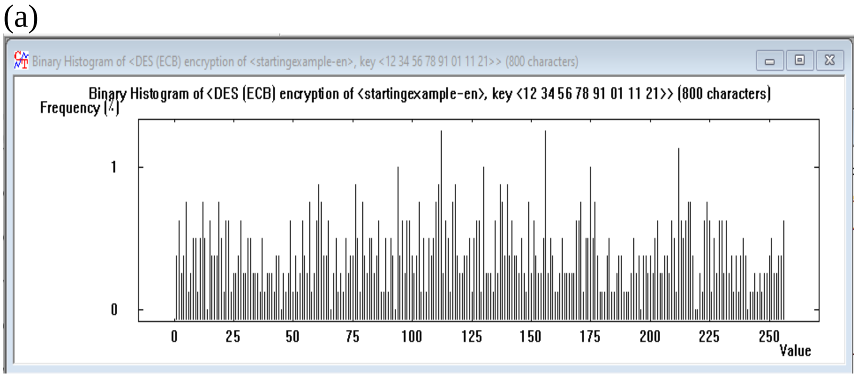
The entropy of the whole document is 7.73 (maximum possible entropy 8.00).

(c)

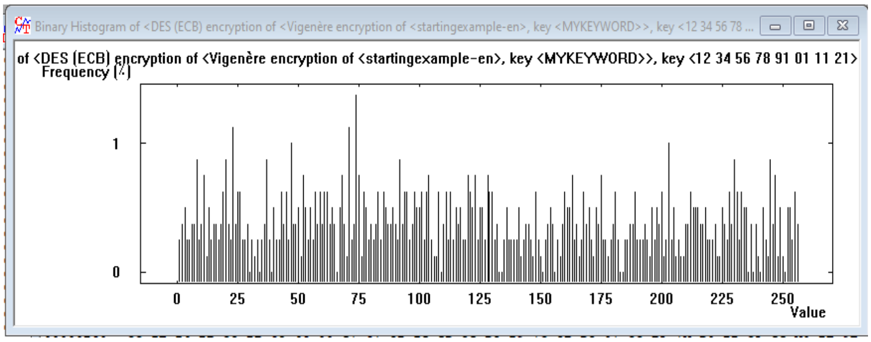
Figure 15: Entropy of Cipher Text for Three Cases of Experiment1 (a) Case1 (b) Case2 (c) Case3



(a)



(b)

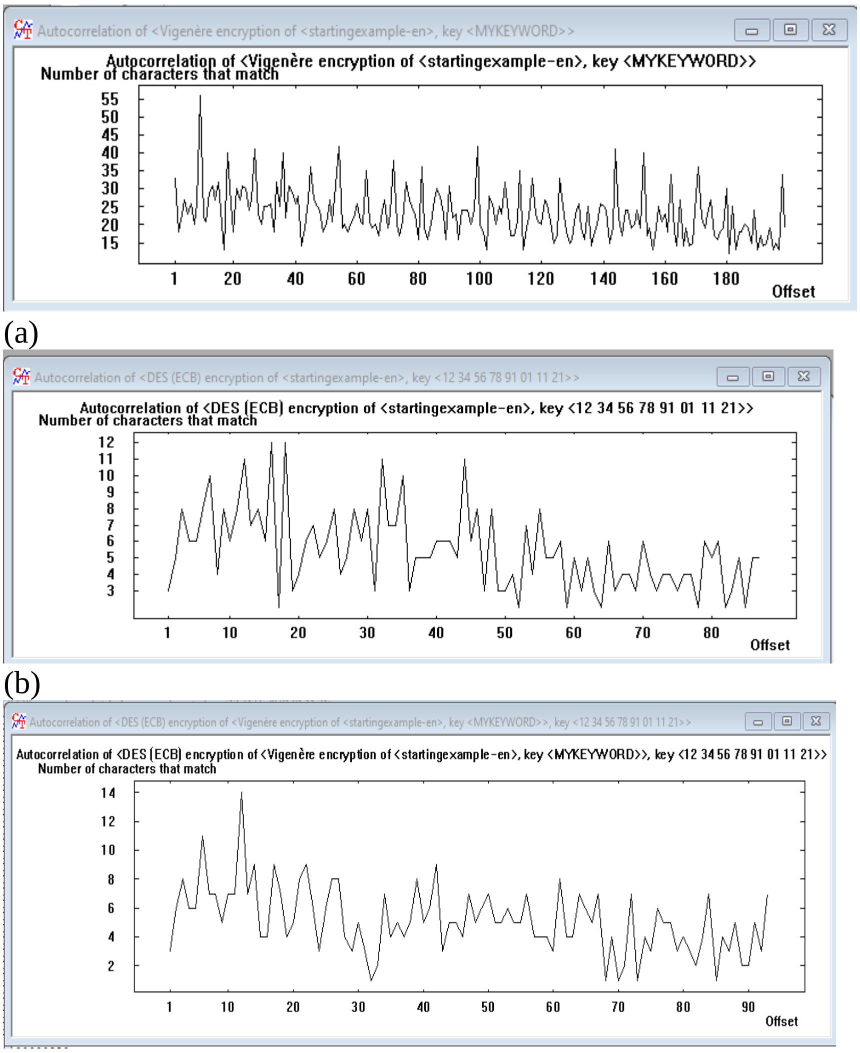


Binary Histogram of <DES (ECB) encryption of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>, key <12 34 56 78 ...

of <DES (ECB) encryption of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>, key <12 34 56 78 91 01 11 21>

(c)

Figure 16: Histogram of Cipher Text for Three Cases of Experiment 1 (a) Case 1 (b) Case 2 (c) Case 3



Autocorrelation of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>

Autocorrelation of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>

(a)

Autocorrelation of <DES (ECB) encryption of <startingexample-en>, key <12 34 56 78 91 01 11 21>>

Autocorrelation of <DES (ECB) encryption of <startingexample-en>, key <12 34 56 78 91 01 11 21>>

(b)

Autocorrelation of <DES (ECB) encryption of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>, key <12 34 56 78 91 01 11 21>>

Autocorrelation of <DES (ECB) encryption of <Vigenère encryption of <startingexample-en>, key <MYKEYWORD>>, key <12 34 56 78 91 01 11 21>>

(c)

Figure 17: Autocorrelation of Cipher Text for Three Cases of Experiment 1 (a) Case 1 (b) Case 2 (c) Case 3

As illustrated from the first experiment above, the entropy value in Case 3 increases with a slight difference, but this slight difference gives a big and strong reflection of the complexity. The histogram of Case3 indicated a greater distribution of cipher-text characters, and finally, the autocorrelation figure of Case3 showed that the autocorrelation was reduced between the characters. That means Case 3 in Experiment 1 is the best case, and the Vigenère algorithm was used to improve the efficiency of the ECB-DES algorithm and increase its complexity against attackers.

B- Second Experiment

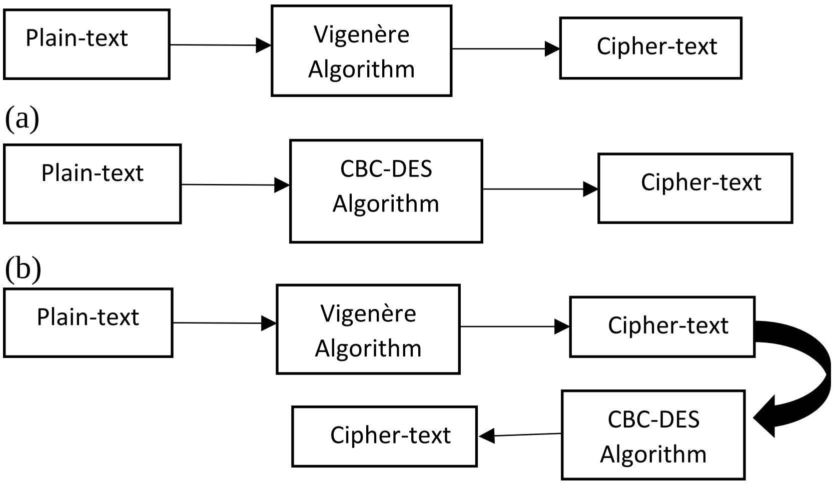
This experiment includes three cases, as follows:

Case 1: In this case, use only the Vigenère algorithm to encrypt plain text.

Case 2: In this case, use only the CBC-DES algorithm to encrypt plain text.

Case 3: In this case, use Vigenère and then CBC-DES algorithms as multilevel to encrypt plain text.

For more illustrations of the second experiment, see Figure 18(a–c).



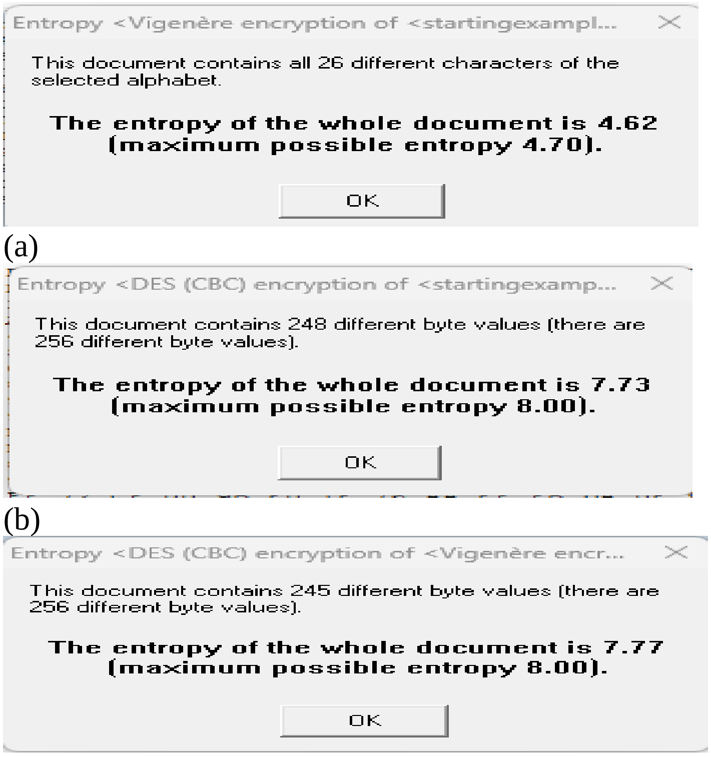
(a)

(b)

(c)

Figure 18: Second experiment (a) Vigenère Ciphering (b) ECB-DES Ciphering (c) Vigenère then CBC-DES Ciphering (Hybrid)

This is followed by Figures (19–21) to illustrate the value of the entropy, histogram, and autocorrelation sequentially for three cases.


Entropy <Vigenère encryption of <startingexampl...

This document contains all 26 different characters of the selected alphabet.

The entropy of the whole document is 4.62 (maximum possible entropy 4.70).

OK

(a)

Entropy <DES (CBC) encryption of <startingexamp...

This document contains 248 different byte values (there are 256 different byte values).

The entropy of the whole document is 7.73 (maximum possible entropy 8.00).

OK

(b)

Entropy <DES (CBC) encryption of <Vigenère encr...

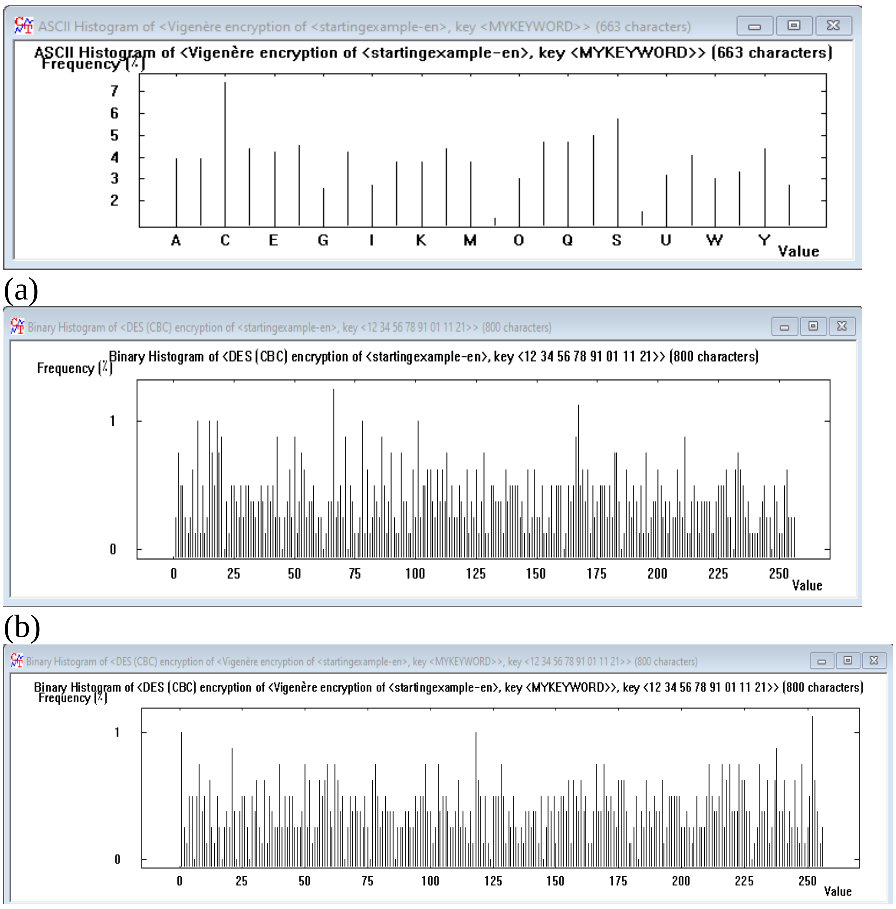
This document contains 245 different byte values (there are 256 different byte values).

The entropy of the whole document is 7.77 (maximum possible entropy 8.00).

OK

(c)

Figure 19: Entropy of Cipher Text for Three Cases of Experiment 2 (a) Case 1 (b) Case 2 (c) Case 3

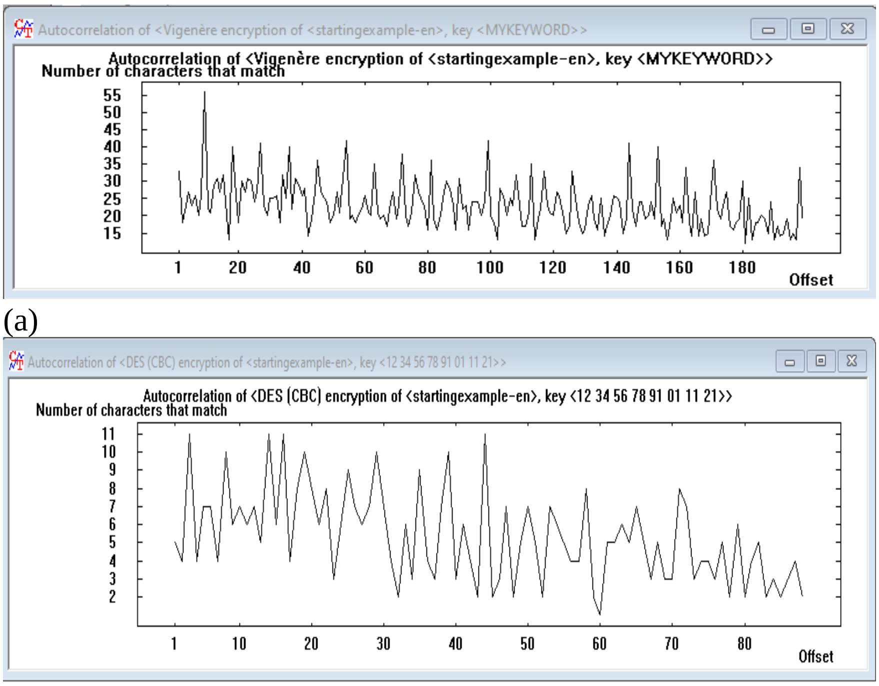


(a)

(b)

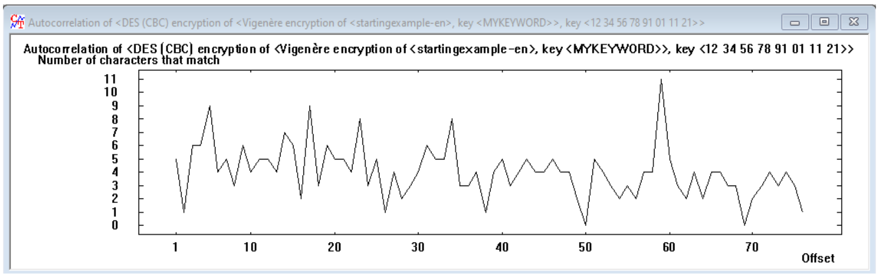
(c)

Figure 20: Histogram of Cipher Text for Three Cases of Experiment 2 (a) Case 1 (b) Case 2 (c) Case 3



(a)

(b)



(c)

Figure 21: Autocorrelation of Cipher Text for Three Cases of Experiment 2 (a) Case 1 (b) Case 2 (c) Case 3

As shown in the second experiment, the entropy value in Case 3 made the cipher-text characters more random. The histogram of Case3 also showed that the cipher-text characters were more evenly distributed. Finally, the autocorrelation figure of Case 3 showed that there was less correlation between the characters, which means that a good result was reached. In addition, Case 3 in Experiment 2 is the best case, and the Vigenère algorithm was used to add a security level to improve the efficiency of the CBC-DES algorithm and increase the complexity of the cryptanalysis. In addition to the above, in the first case for both above experiments, the attacker only needs to know the key of the Vigenère algorithm, so in this case the key search space is equal to 2m only, where m represents the key length of the Vigenère algorithm. In the second case for both above experiments, the attacker only needs to know the key of the DES algorithm, so in this case the key search space is equal to 264 only, which means the search space for the key will be less complicated, have fewer probabilities, and take less time for brute force attacks. For more illustrations, see Table 5. In the third case for both experiments, the proposed system provides a strong hybrid encryption system; in this case, it takes much more time to get the key needed to obtain the plain text. When the attacker tries to break the cipher text, they need to try 264*2m trails, i.e., the DES algorithm key size = 64 bits and the Vigenère algorithm key size = m bits.

Table 6: Key Search Space of Two Experiments

Experiment no.	Case no.	Key Search Space
1	1	2^{m}
	2	2^{64}
	3	$2^{64}*2^{m}$
2	1	2^{m}
	2	2^{64}
	3	$2^{64}*2^{m}$

7. Conclusions

Encryption is the best solution to maintain and achieve security goals, as there are many types of encryption algorithms, and each algorithm equips us with a certain level of efficiency and complexity. According to the previous sections mentioned in this research, six types of classic cryptography algorithms (Vigenère, Porta, Autokey, Beaufort, Trithemuis, and Gronsfeld) were used and compared among their efficiency and complexity levels, and through the use of only three types of analysis tools (entropy, histogram, and autocorrelation), it was noted that the Vigenère algorithm is the best, as it equips us with a high level of security and complexity on the attacker, followed by other algorithms with a slight variation in performance. In other words, the entropy value of the Vigenère cipher is equal to 5.26, which is the biggest among the entropy values of other classical algorithms, as shown previously in Table 5. In addition, the number of characters in the Vigenère cipher is equal to

47, which represents the largest produced character set as compared with the other classical algorithms. The Vigenère algorithm can also be used to make another algorithm, like DES, work better and be safer. This was shown in two previous experiments in Section 6. For example, the entropy value of the DES cipher is 7.72, but it is 7.73 when both the Vigenère and DES algorithms are used in the same encryption operation. This increment in entropy value gives more randomness to the cipher text. Furthermore, when an attacker wants to crack a specific ciphertext, he or she will need more time because there are multiple security levels to pass. In addition, the proposed multi-level model increased the key search space by using two keys for two algorithms in case-3 (2 m * 264), i.e., DES algorithm key size = 64 bits and Vigenère algorithm key size = m bits.

References

[1] M. Ashty, Z. AblhdA, "NEW CRYPTOGRAPHY METHOD BASED ON HILL AND RAIL FENCE ALGORITHMS," *Diyala Journal of Engineering Sciences*, vol. 10, no. 1, pp. 39-47, 2017.

[2] S. A. Shaker, A. G. Naser, F. H. Ali, "New Design of Efficient Non-Linear Stream Key Generator," in *4th International Scientific Conference of Engineering Sciences and Advances Technologies*, 2022.

[3] N. A. Ali, A. S. Rahma, S. H. Shaker, "3D Content Encryption Using Multi-Level Chaotic Maps," *Iraqi Journal of Science,* vol. 64, no. 5, pp. 2521-2532, 2023.

[4] A. A. S. Al-karkhi, N. F. Hassan, R. A. Azeez, "A Secure Private Key Recovery Based on DNA Bio-Cryptography for Blockchain," *Iraqi Journal of Science,* vol. 64, no. 2, pp. 958-972, 2023.

[5] H. A. M. Elwinus , E. Y. Purba, B. Y. Siahaan, R. W. Sembiring, "Collaborative Encryption Algorithm Between Vigenère Cipher, Rotation of Matrix (ROM), and One Time Pad (OTP) Algoritma," *Advances in Science, Technology and Engineering Systems Journal,* vol. 2, no. 5, pp. 13-21, 2017.

[6] M. M. Hoobi, "IMPROVED STRUCTURE OF DATA ENCRYPTION STANDARD ALGORITHM," *JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY,* vol. 55, no. 5, pp. 1-10, 2020.

[7] A. A. Soofi, I. Riaz, U. Rasheed, "An Enhanced Vigenère Cipher For Data Security," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* vol. 5, no. 3, pp. 141-145, 2016.

[8] A. Sciacovelli, V.Vittorio and S.Enrico, "Entropy generation analysis as a design tool—A review," *Renewable and Sustainable Energy Reviews,* vol. 43, pp. 1167-1181, 2015.

[9] B. Brumen, T. Makari, "Resilience of students' passwords against attacks," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* 2017.

[10] M. M. Hoobi, "Strong Triple Data Encryption Standard Algorithm using Nth Degree Truncated Polynomial Ring Unit," *Iraqi Journal of Science,* vol. 58, no. 3C, pp. 1760-1771, 2017.

[11] H. M. Mahmoud, M. M. Hoobi, "Improved Rijndael Algorithm by Encryption S-Box Using NTRU Algorithm," *Iraqi Journal of Science,* vol. 56, no. 4A, pp. 2982-2993, 2015.

[12] E. S. I. Harba, "Advanced Password Authentication Protection by Hybrid Cryptography & Audio Steganography," *Iraqi Journal of Science,* vol. 59, no. 1C, pp. 600-606, 2018.

[13] S. A. Shaker, A. G. Nasir, F. H. Ali, "Constructing a Digital Certificate Authentication System for Classified Documents," *Iraqi Journal of Science,* vol. 64, no. 3, pp. 1391-1400, 2023.

[14] D. Rachmawati, A. N. Lubis, "Combining Beaufort cipher and RSA-CRT algorithm in a hybrid scheme to secure images," in *2nd TALENTA-International Conference on Science and Technology*, 2023.

[15] R. N. Sari; R. S. Hayati, "Beaufort Cipher Algorithm Analysis Based on the Power Lock-Blum Blum Shub in Securing Data," in *2018 6th International Conference on Cyber and IT Service Management (CITSM),* 2018.

[16] I. Sumartono , A. P. U. Siahaan , N. Mayasari, "An Overview of the RC4 Algorithm," *IOSR Journal of Computer Engineering (IOSR-JCE),* vol. 18, no. 6, pp. 67-73, 2016.

[17] Sh. H. Biswas, Md. A. Ali, M. Rahman, K. Sohel, M. Hasan, K. Sarkar, A. A. razzaque, "A systematic study on classical cryptographic cypher in order to design a smallest cipher," *International Journal of Scientific and Research Publications,* vol. 9, no. 12, pp. 507-511, 2019.

[18] S. Ashok, R. Kiran, S. Pradeep, R. Devara, "A New Variant of Rail Fence Cipher using Hybrid Block-Swap Method," *International Research Journal of Engineering and Technology (IRJET),* vol. 8, no. 7, pp. 1735-1739, 2021.

[19] Y. E. Yousif, "IMPROVING THE EFFICIENCY OF DES ALGORITHM USING NEURAL NETWORKS," *International Journal of Engineering Applied Sciences and Technology,* vol. 5, no. 1, pp. 26-29, 2020.

[20] R. M. Z. Subhi, "DES encryption and decryption algorithm implementation based on FPGA," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 18, no. 2, pp. 774-781, 2020.

[21] W. Stallings, Cryptography and Network Security Principles and Practice, Pearson India; 7th edition, 2018.

[22] P. S. Lakshmi, G. Murali, "Comparison of Classical and Quantum Cryptography using QKD Simulator," *International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017),* pp. 3543-3547, 2017.

[23] M. A. Budiman, D. Rachmawati, Jessica, "Implementation of Super-Encryption with Trithemius Algorithm and Double Transposition Cipher in Securing PDF Files on Android Platform," in *2nd International Conference on Computing and Applied Informatics 2017*, 2017.

[24] Z. Ramadhan, A. Putera, U. Siahaan, "Protection of Important Data and Information using Gronsfeld Cipher," *INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD,* vol. 4, no. 10, pp. 128-132, 2018.

[25] E. S. Harba, H. S. Harba, I. A. Abdulmunem, S. S. Hussein, "Improving Security of the Crypto-Stego Approach using Time Sequence Dictionary and Spacing Modification Techniques," *Iraqi Journal of Science,* vol. 62, no. 5, pp. 1721-1733, 2021.

[26] M. M. Hoobi, "Keystroke Dynamics Authentication based on Naïve Bayes Classifier," *Iraqi Journal of Science,* vol. 56, no. 2A, pp. 1176-1184, 2015.

[27] M. M. Hoobi, "Modified Robust AES Architecture," *Technology Reports of Kansai University,* vol. 26, no. 10, 2020.

[28] S. Subramani, S. Munuswamy, K. Arputharaj, S. K. Svn, "Review of Security Methods Based on Classical Cryptography and Quantum Cryptography," *Cybernetics and Systems,* 2023.

[29] D. V. V. Deepthi, B. H. Benny, K. Sreenu, "Various Ciphers in Classical Cryptography," *IOP Conf. Series: Journal of Physics: Conf. Series 1228 (2019) 012014,* pp. 1228-1234, 2019.

[30] Q. Z. Abdulla, M. D. Al-Hassani, "Robust Password Encryption Technique with an Extra Security Layer," *Iraqi Journal of Science,* vol. 64, no. 3, pp. 1477-1486, 2023.

[31] K. Nahar, P. Chakraborty, "Improved Approach of Rail Fence for Enhancing Security," *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* vol. 9, no. 9, pp. 583-585, 2020.

[32] M. M. Hoobi, "EFFICIENT HYBRID CRYPTOGRAPHY ALGORITHM," *JOURNAL OF SOUTHWEST JIAOTONG UNIVERSITY,* vol. 55, no. 3, pp. 1-9, 2020.

[33] M. M. Hoobi, "Survey :Efficient Hybrid Algorithms of Cryptography," *MINAR International Journal of Applied Sciences and Technology,* vol. 2, no. 4, pp. 1-16, 2020.

[34] M. M. Hoobi, S. S. Sulaiman, I. A. AbdulMunem, "Enhanced Multistage RSA Encryption Model," *2nd International Scientific Conference of Al-Ayen University (ISCAU), IOP Conf. Series: Materials Science and Engineering,* p. 455, 2020.

[35] D. C. Brown, "A cryptanalysis of the autokey cipher using the index of coincidence," *ACMSE '18: Proceedings of the ACMSE 2018 Conference,* pp. 1-8, 2018.

[36] S. Rubinstein-Salzedo, "Other Types of Ciphers," in *Cryptography,* Springer, 2018, pp. 63-73.

[37] CrypTool-Online_Cryptography_for_everybody, "CrypTool-Online," CrypTool-Online Cryptography for everybody, [Online]. Available: https://www.cryptool.org/en/cto/.

[38] S. Agustini, W. M. Rahmawati, M. Kurniawan, "Modified Vigenère Cipher to Enhance Data Security Using Monoalphabetic Cipher," *International Journal of Artificial Intelligence & Robotics (IJAIR),* vol. 1, no. 1, pp. 26-32, 2019.

[39] H. Zhu, Z. Li, "An Efficient Biometric Authenticated Protocol for Arbitrary-domain-server with Blockchain Technology," *International Journal of Network Security,* vol. 23, no. 3, p. 386–394, 2021.

[40] P. Dubey, O. Yadav, "A Survey on Quantum Cryptography versus classical Cryptography," *International Journal of Current Engineering and Technology,* vol. 10, no. 6, pp. 910-913, 2020.

[41] S. M. Kareem, A. M. S. Rahma, "A Modification on Key Stream Generator for RC4 Algorithm," *Eng. Technol. J.,* vol. 38, no. 28, p. 54–60, 2020.

[42] A. Saraswata, C. Khatria, Sudhakara, P. Thakrala, P. Biswas, "An Extended Hybridization of Vigenère and Caesar Cipher Techniques for Secure Communication," *2nd International Conference on Intelligent Computing, Communication & Convergence,* pp. 355-360, 2016.

[43] S. D. Nasution, G. L. Ginting, M. Syahrizal, R. Rahim, "Data Security Using Vigenère Cipher and Goldbach Codes Algorithm," *International Journal of Engineering Research & Technology (IJERT),* vol. 6, no. 1, pp. 360-363, 2017.

[44] R. Darari, E. Winarko, A. Damayanti, "Encryption and Decryption Application on Images with Hybrid Algorithm Vigenère and RSA," *Contemporary Mathematics and Applications,* vol. 2, no. 2, p. 109–117, 2020.

[45] A. A. Hussein, N. K. Ayoob, "Key Generation for Vigenère Ciphering Based on Genetic Algorithm," *Journal of University of Babylon,* vol. 30, no. 1, pp. 200-208, 2022.

[46] S. M. Ali, N. T. Mahmood, S. A. Yousif, "Meerkat Clan Algorithm for Solving N-Queen Problems," *Iraqi Journal of Science,* vol. 62, no. 6, pp. 2082-2089, 2021.

[47] Wisam Abed Shukur, Luheb Kareem Qurban, and Ahmed Aljuboori, "Digital Data Encryption Using a Proposed W-Method Based on AES and DES Algorithms," *Baghdad Science Journal,* vol. 20, no. 4, pp. 1414-1424, 2023.

[48] Saad A. Abdulameer, Ali H. Kashmar, and Ammar I. Shihab, "A Cryptosystem for Database Security Based on TSFS Algorithm," *Baghdad Science Journal,* vol. 17, no. 2, pp. 567-574, 2020.

[49] Enas Tariq Khudair, Ekhlas Falih Naser, and Alaa Noori Mazher, "Comparison between RSA and CAST-128 with Adaptive Key for Video Frames Encryption with Highest Average Entropy," *Baghdad Science Journal,* vol. 19, no. 6, pp. 1378-1386, 2022.

[50] K. Somsuk, "A New Methodology to Find Private Key of RSA Based on Euler Totient Function Function," *Baghdad Science Journal,* vol. 18, no. 2, pp. 338-348, 2021.