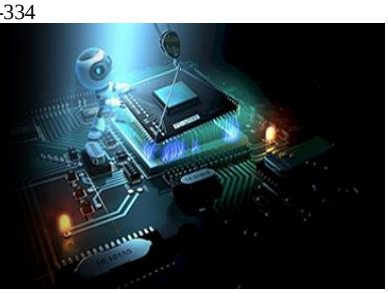
International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
Impact Factor (RJIF): 5.52
www.computersciencejournals.com/ijecs
IJECS 2025; 7(2): 318-334
Received: 20-10-2025
Accepted: 25-11-2025

Firas Sameer Ashour
Independent Researcher, Iraq

Corresponding Author:
Firas Sameer Ashour
Independent Researcher, Iraq

Comparative between classical and quantum algorithms for solving path optimization problems

Firas Sameer Ashour

DOI: https://www.doi.org/10.33545/26633582.2025.v7.i2d.235

Abstract
It is essential to maintain users' rights to privacy, confidentiality, and integrity of their information. Cryptography is the only technology that allows users to protect and guarantee their rights. As such, encryption is the most important policy to be employed by users to make it virtually impossible for attackers to decipher transmitted messages. This research has two major objectives: In Part 1, we will test six traditional encryption methods to determine which produces the greatest amount of complexity for an attacker attempting to decipher an identical plain-text message. The six traditional encryption algorithms to be tested include: Beaufort, Garsfeld, Porta, Trithemuis, Autokey, and Vigenère. In addition to testing the encryption methods, we will analyze the results using techniques such as Entropy, Histograms, and Autocorrelation. In Part 2, we will build upon our findings from Part 1 to develop and test two hybrid ciphering models utilizing Vigenère to enhance the security of the Data Encryption Standard (DES). Each trial will have three cases, representing three scenarios. For example, in the first trial, we will test the Vigenère algorithm used alone to encrypt plaintext, the ECB DES algorithm used alone to encrypt plaintext, and a combination of both algorithms to create a multilevel encrypted ciphertext. In the second trial, we will use the same cases, however, we will add the CBC DES algorithm to the previous case to create another hybrid ciphering model. We expect to find that all six of the classical encryption algorithms have differing levels of effectiveness, and that the Vigenere method is the most complex and therefore the most successful. Furthermore, when the Vigenere method is used in conjunction with additional methods, such as the DES, the overall complexity of the encryption and the level of security provided to the user can be increased.

Keywords: Cryptography, classical encryption algorithms, quantum algorithms, vigenère cipher, data encryption standard (DES), encryption complexity, information security

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]. As illustrated in Figure 1, it is executed at the receiving end and transforms the encrypted content into plaintext [7, 20].

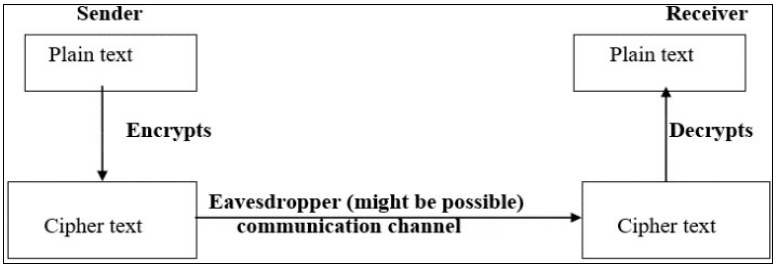


Fig 1: Encryption and Decryption Operations [7]

Before being encrypted into cipher text, plain text (at the sender) is a simple, readable text in cryptography [21]. The encrypted message is called the cipher text at the receiving end [22]. To encrypt a message, we need a key and some sort of algorithm to convert plain text to cipher text [23]. Encryption can be categorized as either transposition (when the content of the plain text is modified) and substitution (when the content of the plain text is not changed), which fall under two major types of encryption algorithms [7]. Finally, there are two types of cryptography, symmetric and asymmetric [15], based on the type of keys that are used for both encryption and decryption [7].

Symmetric key Cryptography

While the terms private key and secret key appear in discussions of cryptography, they refer to a single entity that uses the same key for both encryption of plaintext and decryption of ciphertext [24, 25], i.e., Symmetric Cryptosystems such as the One Time Pad (OTP), the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) [7, 15].

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.

The authors of [28], have taken steps to increase the size of the DES encryption key from its original 56-bits to 1024-bits; this larger key will be broken into sixteen 64-bit keys; each one of the sixteen 64-bit keys are generated independently per cycle of the algorithms, so as to greatly improve the time it would take to try every possible key (blind search). Thus, the authors' method has provided an exponentially better performance than previously implemented DES methods due to the added complexity of the search space due to the addition of the new larger key sizes.

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. To execute the parallel 2DES algorithm, we use the Message Passing Interface (MPI) module. We demonstrated that with the parallel 2DES method, the run time for executing both encryption and decryption was significantly less than in the case of sequential. Furthermore, when employing a greater number of CPUs, the parallel 2DES method resulted in better parallel performance. As such, it is expected that due to the significant improvement in processing speed over sequential methods, the parallel 2DES will be an attractive solution for parallelizing multimedia encryption and decryption applications.

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.

A few researchers have also proposed enhancements for DES that could increase its resistance to various forms of attacks, such as brute-force attacks or cryptanalysis (e.g., [30]). The authors' enhancement to DES included a different method of utilizing the F-Function which combined striding methods with an XOR function between the 56 bit key, and the plaintext. This made us better able to defend against such types of attacks. Using the Avalanche Impact, the authors were able to test the new DES algorithm versus the original, and determine that 1-bit vibrations in the plaintext resulted in approximately 55% Avalanche Impact.

For enhancing the safety and protection of attacks on DES, the author developed a structure to update the traditional DES in [25] and we have used the standard DES with the novel approach of generating two keys for the enhancement of the structure. The author generated one key directly (simple) and generated the second key indirectly by encrypting it using the improved Caesar cipher. After the first eight rounds of the encryption, the algorithm uses the first key as a simple key and after the ninth round until the sixteenth round uses the second key as the encrypted key. Our research results show that compared to the traditional DES the enhanced DES has improved security, performance and search complexity.

A technique of protecting text based on keys has been proposed by the authors of [31]: KE-DES. In order to implement the KE-DES method there are two steps. First, the odd/even bit transformation of each of the key bits in the DES method need to be combined. Second, the K-D function needs to be utilized instead of the original DES's right side expansion to replace it. 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] authors increased the block-size and key-size of the DES-algorithm from 64-bit to 128-bits, which resulted in a modification of the internal-structure of the proposed method; as such, this alteration of all other aspects of DES made the system more secure through an increase in its confusion and diffusion components. The encryption is applicable for both simple text and graphics. The enhanced encryption-method was tested utilizing standard benchmark-data to validate the intended methodology.

The only non-linear component of the proposed approach is the replacement process; this was the area that the authors modified for an enhancement to DES in [33]. Therefore, due to this modification, it has allowed for improved performance out of DES and yielded very good results. In addition, during the substitution phase of DES, a new, superior design (a 6 x 6) of the S-box has been employed. The architecture uses the Galois Field methodology to generate S-Boxes that are robust against differential and linear attacks. Additionally, the upgraded DES will provide a larger key space that can be used to protect the system from brute-force attacks. The authors have demonstrated that the replacement S-Boxes are effective, through a number of performance analysis tests, and therefore demonstrate that the entire DES is also effective.

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]. Since the cipher text undergoes various shift operations, the Vigenère approach differs from exhibition-to-frequency analysis in that it does not always encrypt the same plaintext character 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:

$$Ci = (Pi + Ki)\ mod\ m \quad (1)$$

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: One example of an algorithm is this classical substitution. To create the keyword, this algorithm contacts the keyword at the starting point of the plain text; otherwise, it is identical to Vigenère [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. Once you've discovered the

plain-text letter that was provided, find the leftmost letter in that row; it contains the encryption key [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 (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) [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)

With DES, a symmetric-key encryption method, both the encryption and decryption processes use the identical secret key [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.

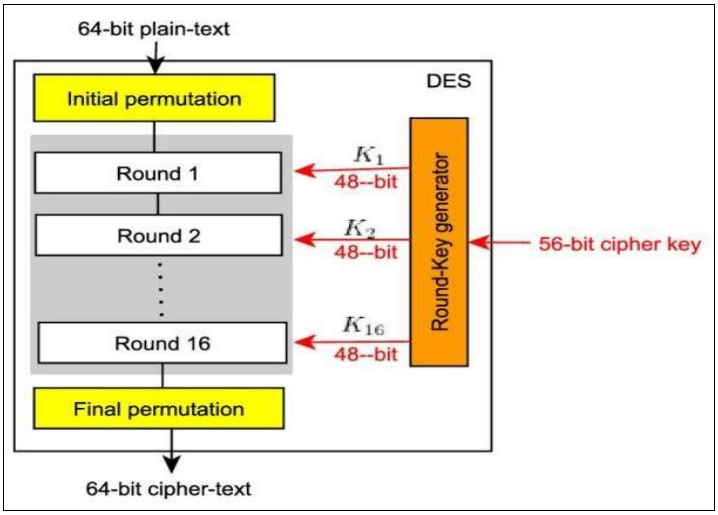


Fig 2: Structure of the DES Algorithm

There is little difference between the DES encryption and decryption processes; the only difference is that the cipher text is sent into the DES algorithm instead of the keys, and vice versa. 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

- Direct the 64-bit plaintext block to the preliminary permutation (IP) function.
- 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).

There is a 16-step encryption process for both LPT and RPT blocks. Key transformation, expansion permutation, S-Box permutation, P-Box permutation, XOR, and swap are the five steps that make up this step.

- Merge LPT and RPT blocks, then perform final permutation (FP) on the resulted block.
- The outcome 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

Using this tool, you may visually depict each range of data with a vertical bar. On the one hand, we have the character's appearance rate, and on the other, we have the character's horizontal axis of appearance. Frequency is another name for histogram.

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.

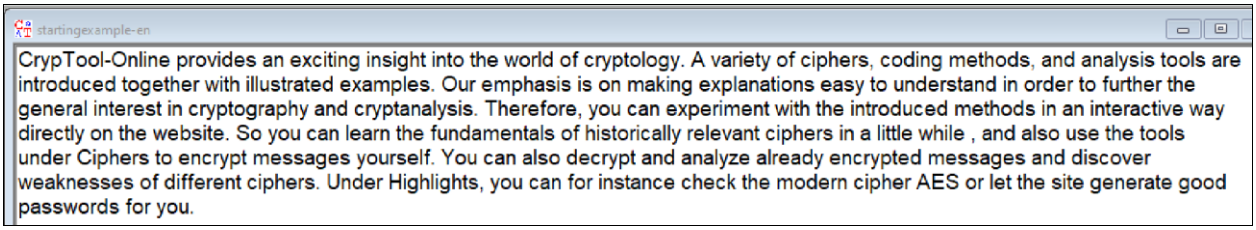


Fig 3: Plain-Text Cryptool S/W Interface

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	**Cipher-Text Name**
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 ftthtjp 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.

The encryption technique works better when there is diversity of content within the cipher text, resulting in a higher number of different characters. The greater the number of diverse characters, the harder the cipher text is for attackers to decode. In accordance with the data presented here, the Vigenère encryption was the most effective of the six methods evaluated, based upon its highest entropy rate (5.26) and the highest letter variety (up to 47).

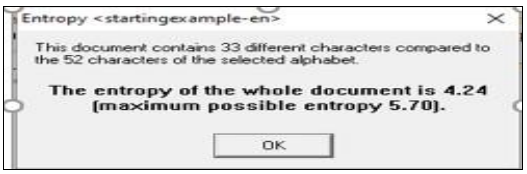


Fig 4: Entropy of Plain-Text

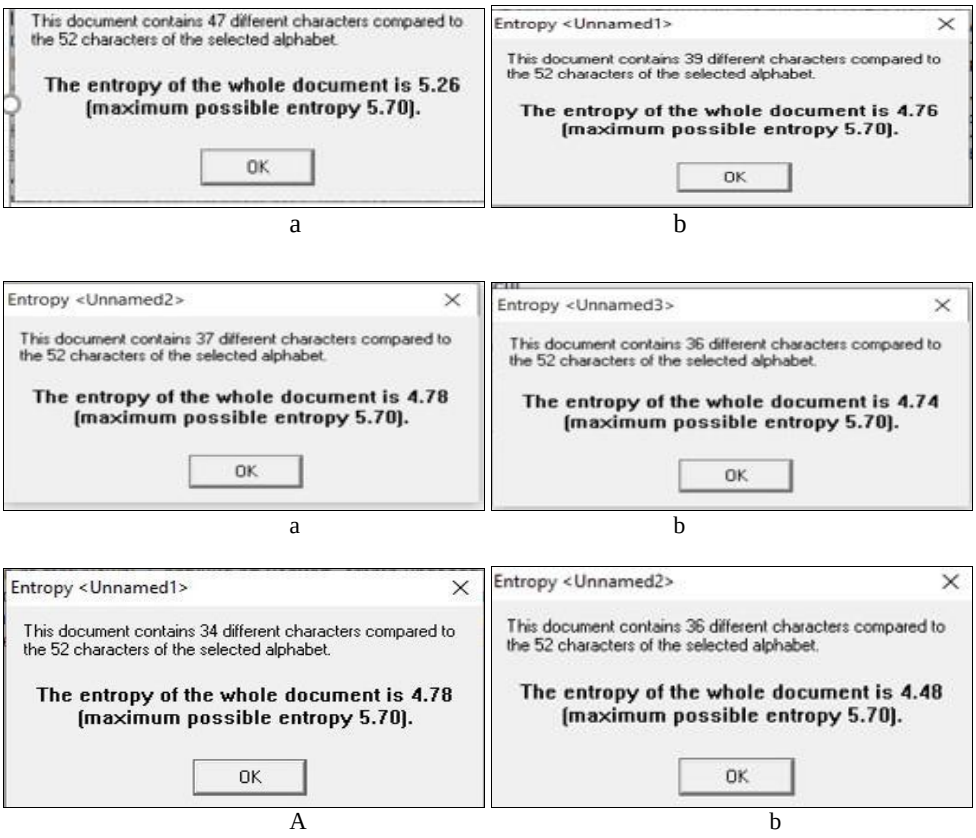


Fig 5: Entropy for Six Used Algorithms (a) Vigenère (b) Porta (c) Autokey (d) Beaufort (e) Trithemuis (f) Gronsfeld

Table 5: Entropy Values Relative to the Number of Characters

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

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

An attempt was made to discover the number of times a letter appears in the clear text using the same device as before. Figures (a-f) 7 show the program that has been used to determine the number of times each letter appears in the encrypted message. Also, the same tool has been used for this purpose. From these tables, and from the figures demonstrating how the Vigenere algorithm operates, it can be concluded that the most effective algorithm will provide the highest variability of characters used in the cipher text, and decrease the frequency of similar values among letters, thus making it more difficult for an attacker to decipher the original text.

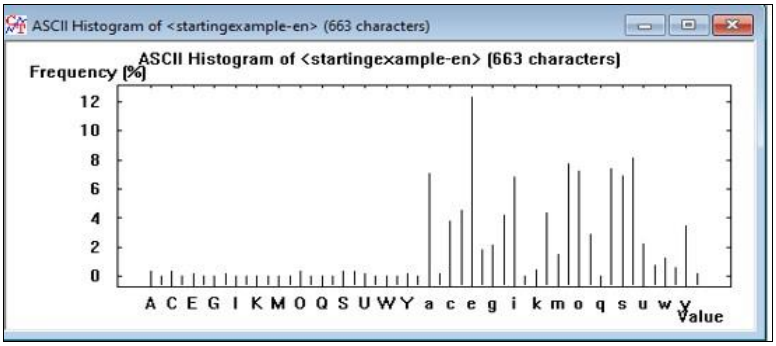
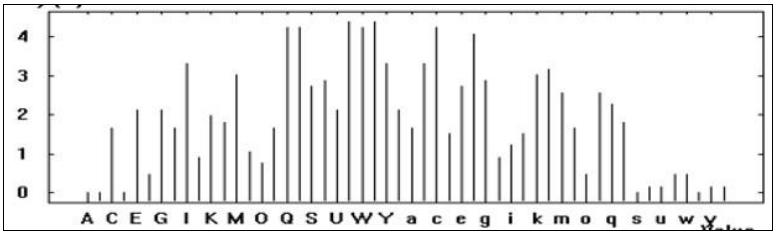
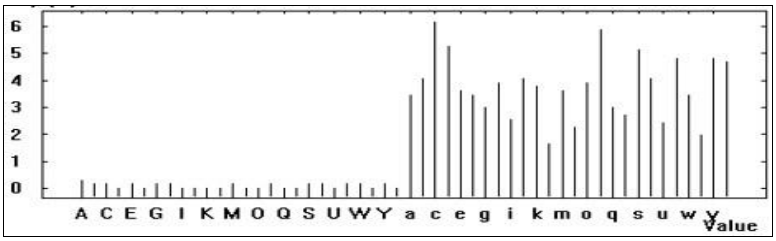


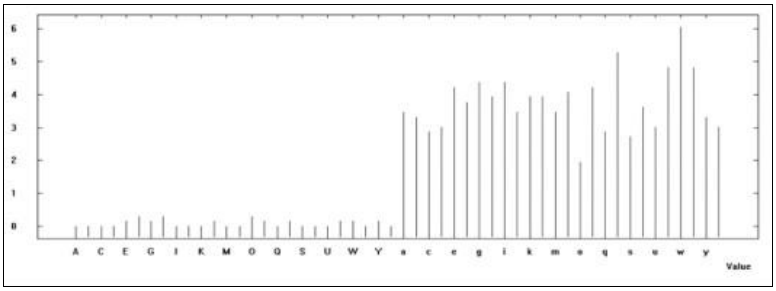
Fig 6: Histogram of Plain Text



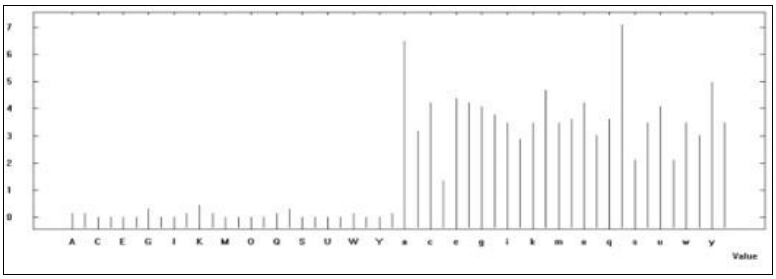
(a)



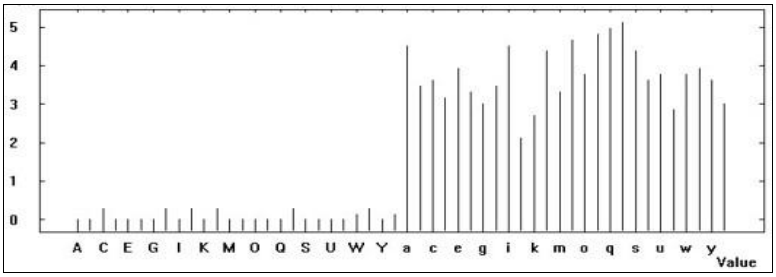
(b)



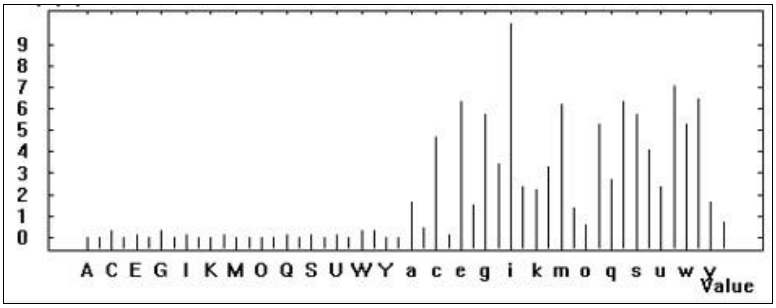
(c)



(d)



(e)



(f)

Fig 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.

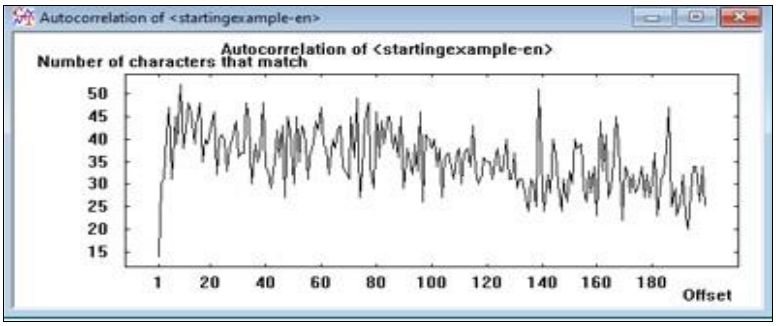
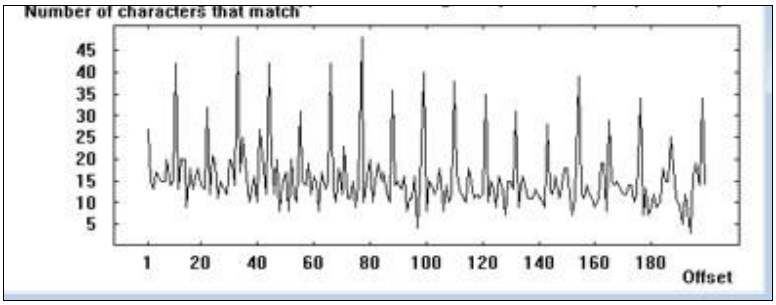
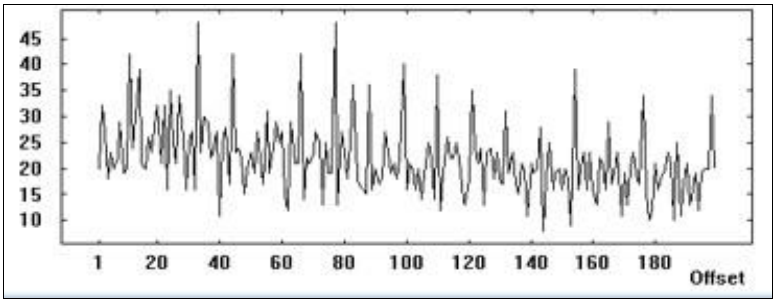


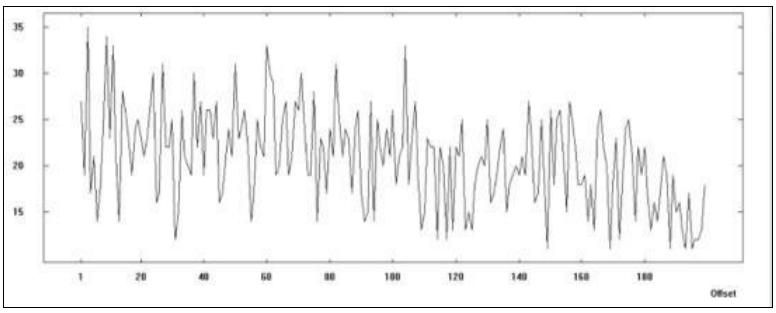
Fig 8: Autocorrelation of Plain Text



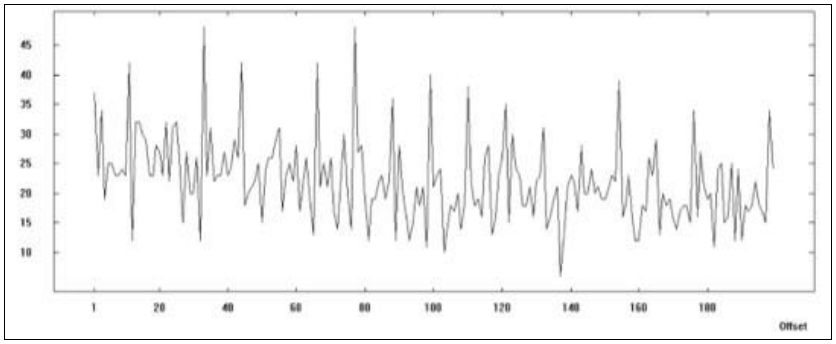
(a)



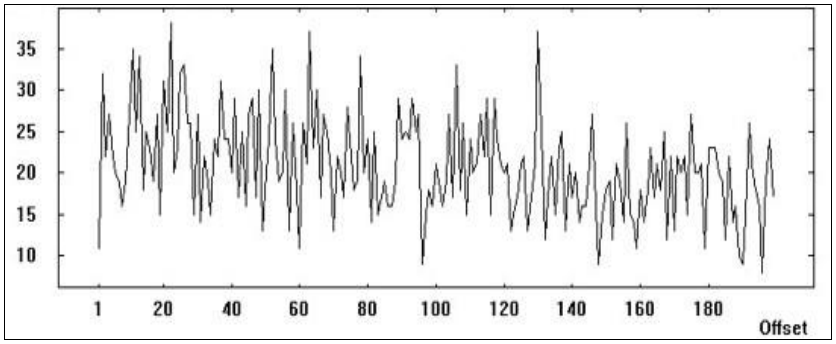
(b)



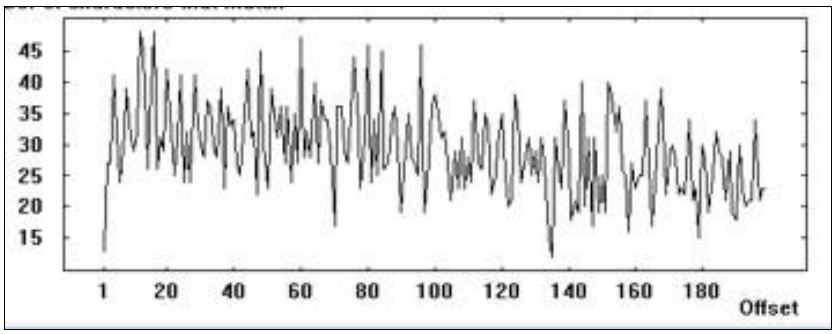
(c)



(d)



(e)



(f)

Fig 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.

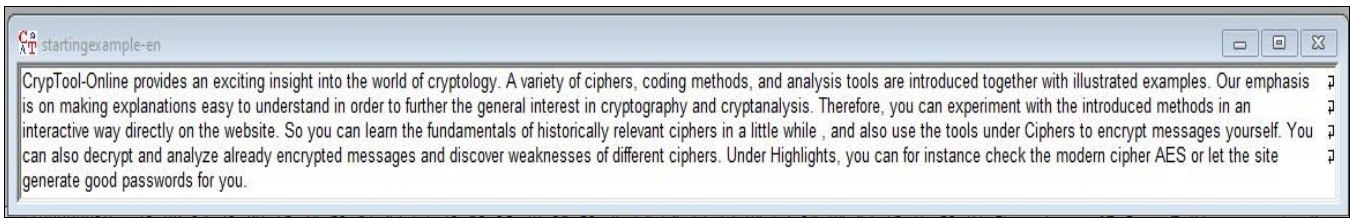


Fig 10: Plain Text

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

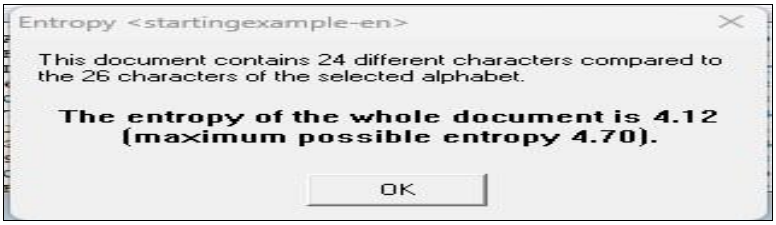


Fig 11: Entropy of Plain Text

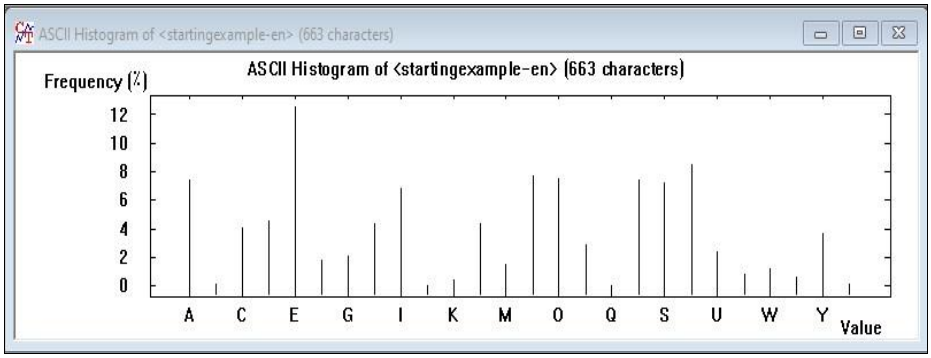


Fig 12: Histogram of Plain Text

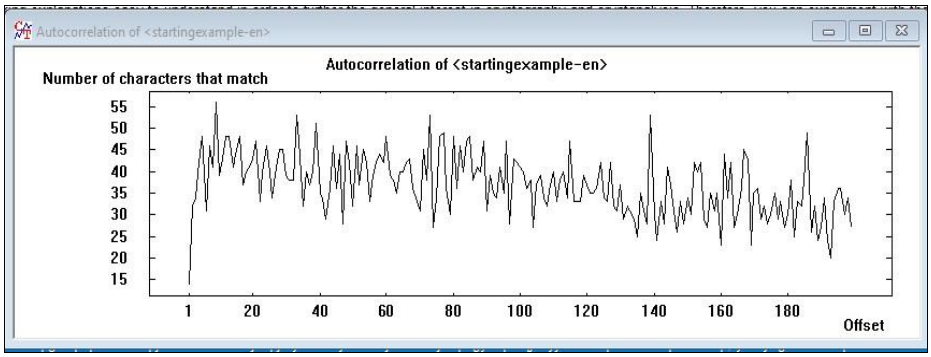


Fig 13: Autocorrelation of Plain Text

A. First Experiment

This experiment includes three cases, as follows:

Scenario 1: Encrypt plaintext by using solely the Vigenère method.

Scenario 2: Encrypt plaintext by using solely ECB-DES.

In the third scenario, use the Vigenère method and ECB-DES in layers to encrypt plaintext.

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

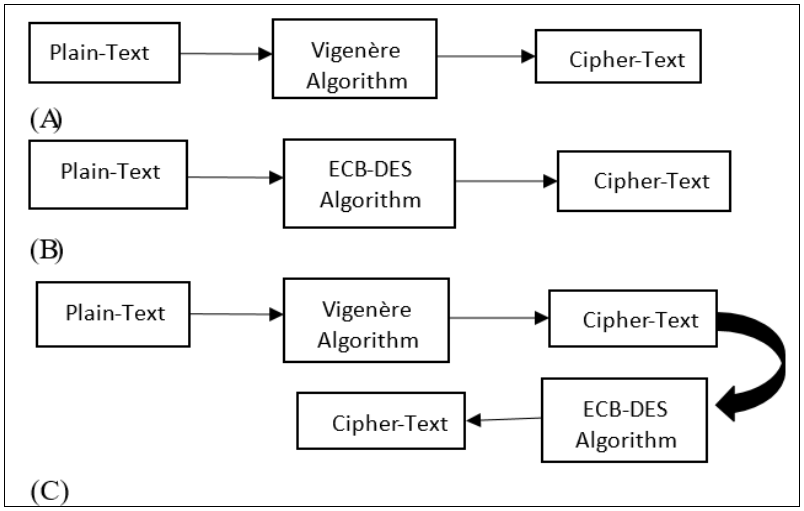
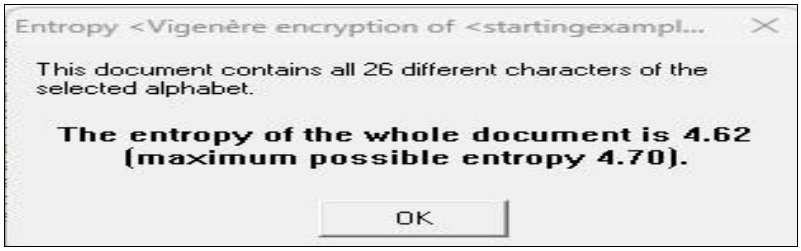
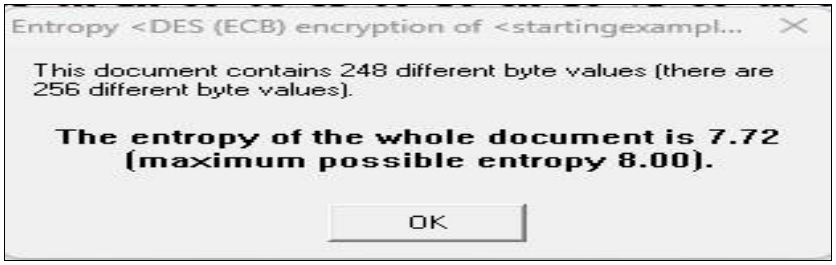


Fig 14: First experiment (a) Vigenère Ciphering (b) ECB-DES Ciphering (c) Vigenère, then ECB-DES Ciphering (Hybrid)

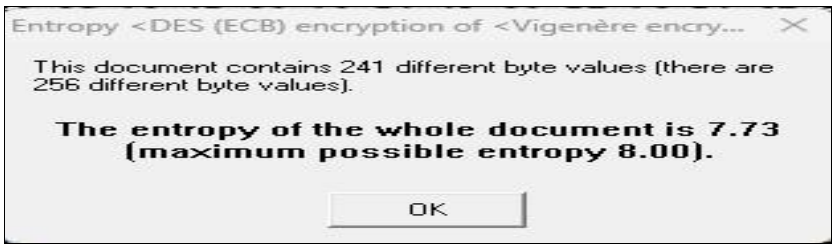
See Figures (15-17) to illustrate the value of the entropy, histogram, and autocorrelation sequentially for three cases.



(a)

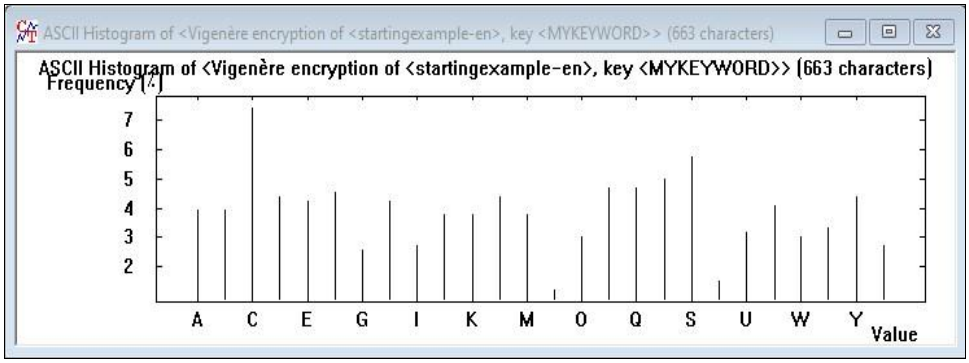


(b)

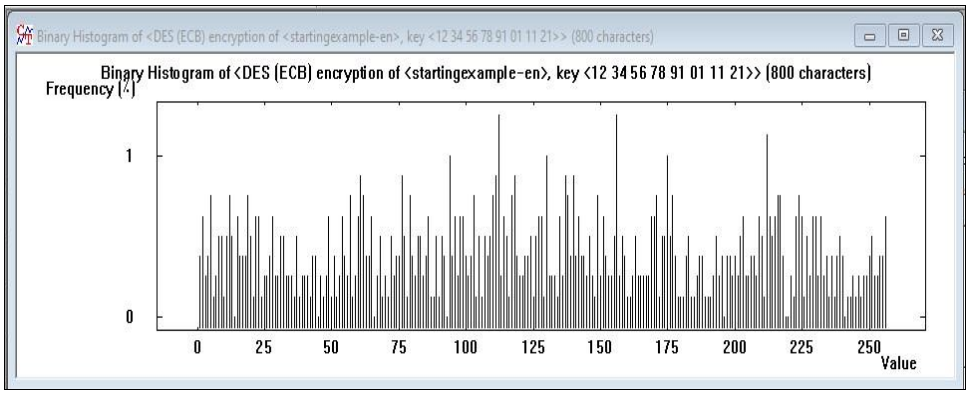


(c)

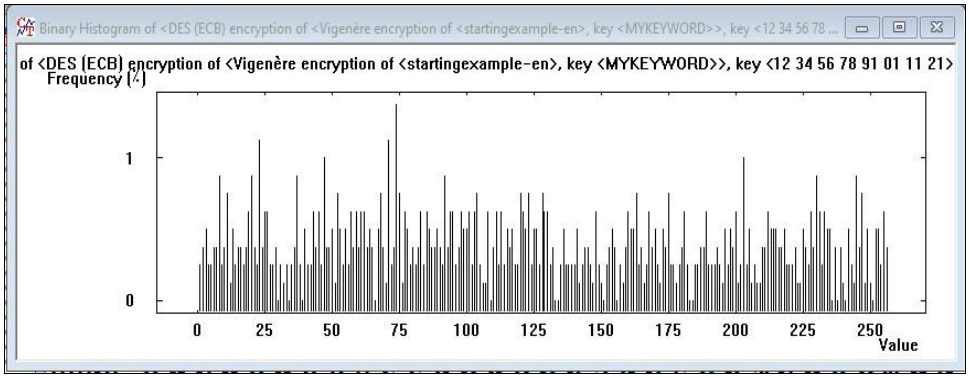
Fig 15: Entropy of Cipher Text for Three Cases of Experiment1 (a) Case1 (b) Case2 (c) Case3



(a)

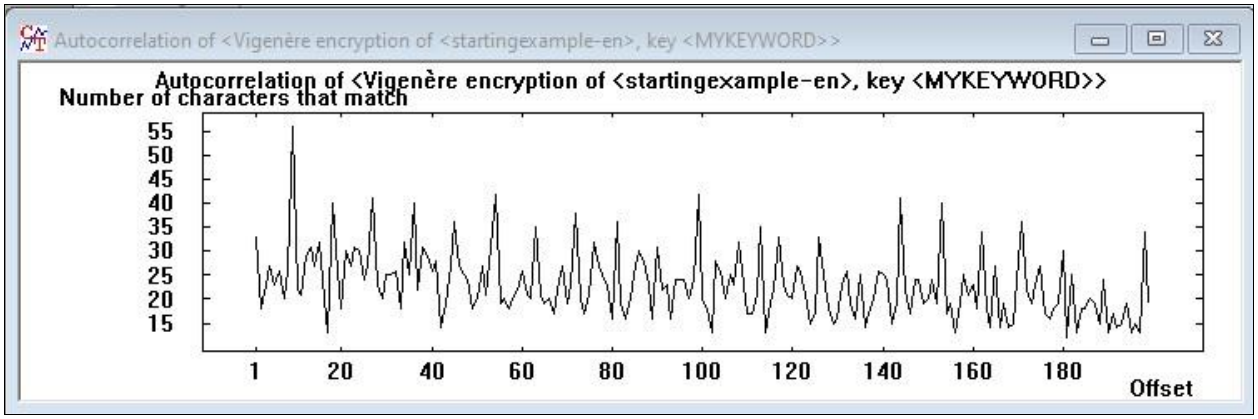


(b)

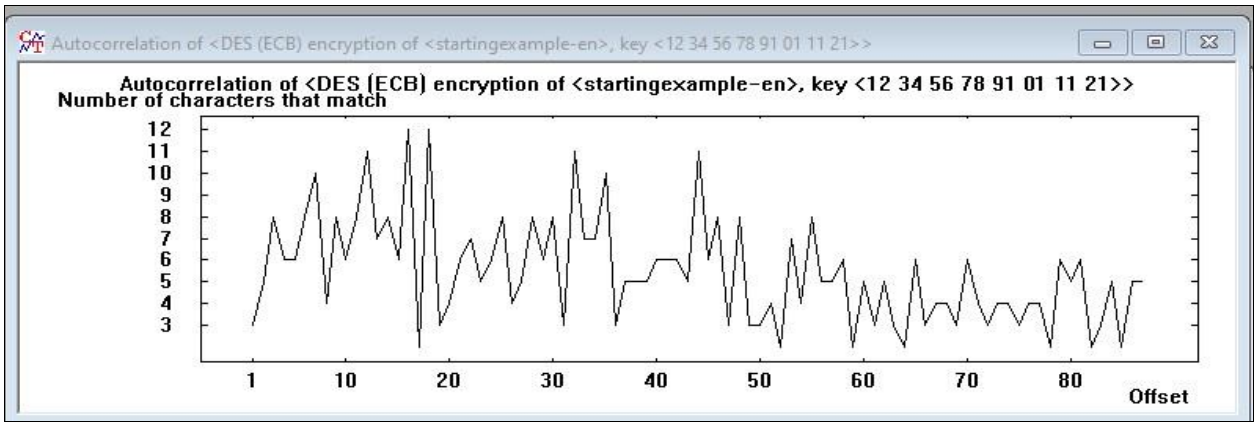


(c)

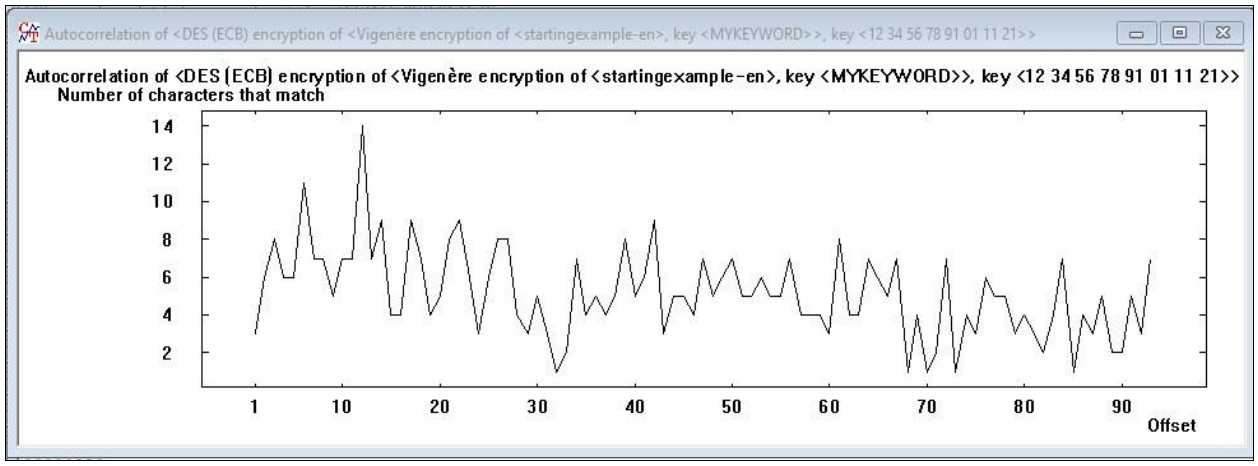
Fig 16: Histogram of Cipher Text for Three Cases of Experiment 1 (a) Case 1 (b) Case 2 (c) Case 3



(a)



(b)



(c)

Fig 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. Therefore, the best scenario is case 3 of Experiment 1 in which the Vigenère algorithm was used as an enhancement for the ECB-DES algorithm to make it more efficient and more difficult for an attacker to find a way around the encryption.

B. Second Experiment

This experiment includes three cases, as follows:

The first scenario is to use only Vigenère for encrypting plaintext.

In the second scenario, the CBC-DES method will be used alone for encrypting plaintext.

In the third scenario, both of the Vigenère and CBC-DES methods will be employed in a multilevel fashion for encrypting plaintext.

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

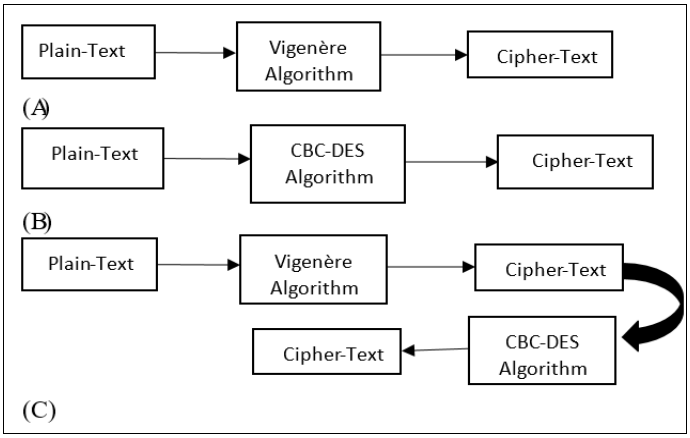
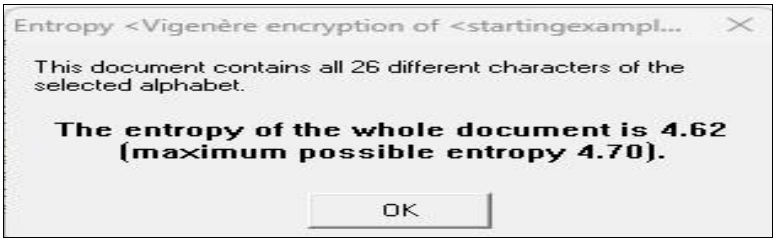
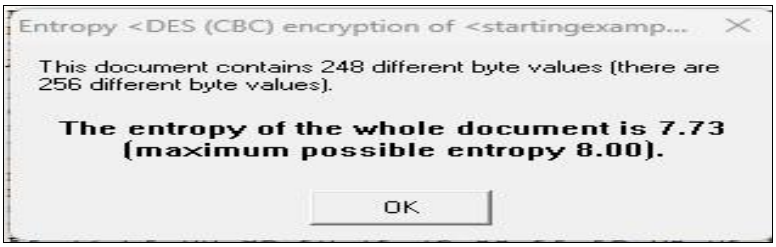


Fig 18: Second experiment (a) Vigenère Ciphering (b) ECB-DES Ciphering (c) Vigenère then CBC-DES Ciphering (Hybrid)

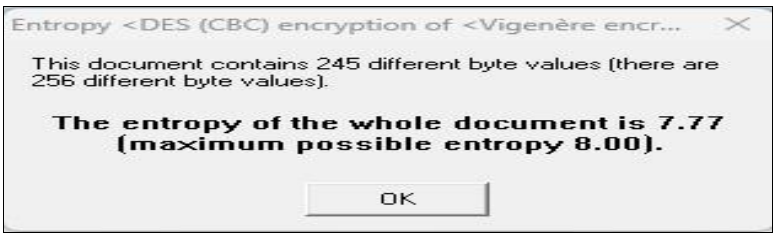
This is followed by Figures (19-21) to illustrate the value of the entropy, histogram, and autocorrelation sequentially for three cases.



(a)

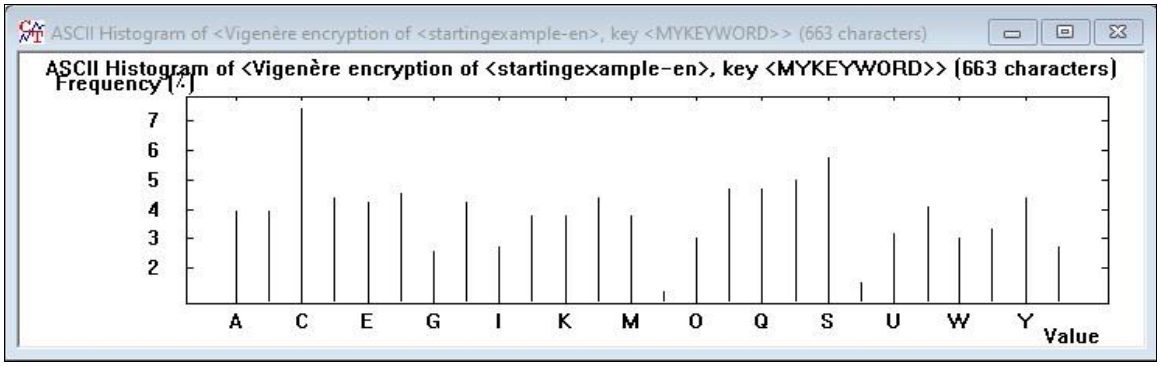


(b)

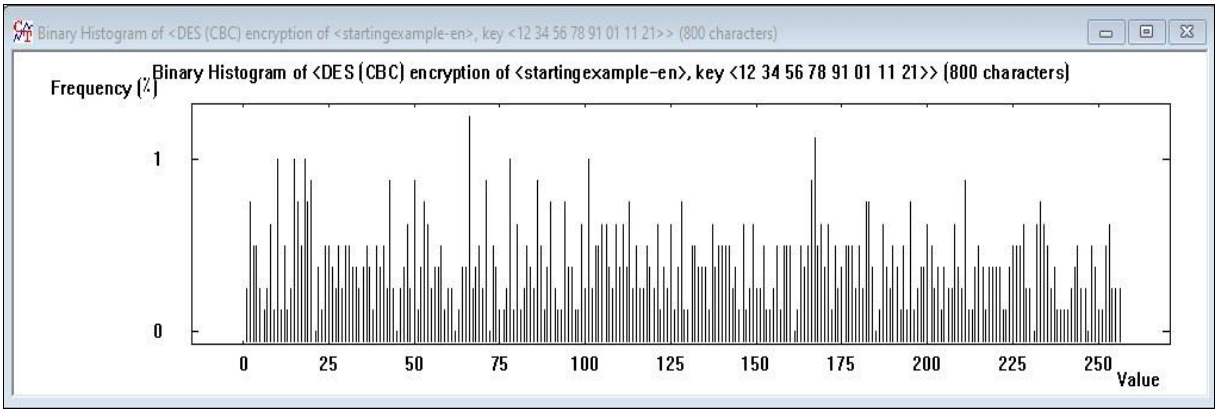


(c)

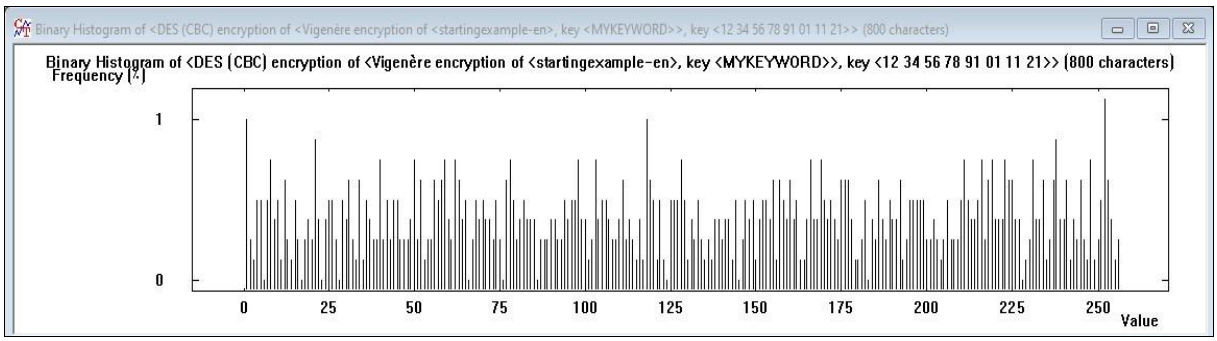
Fig 19: Entropy of Cipher Text for Three Cases of Experiment 2 (a) Case 1 (b) Case 2 (c) Case 3



(a)

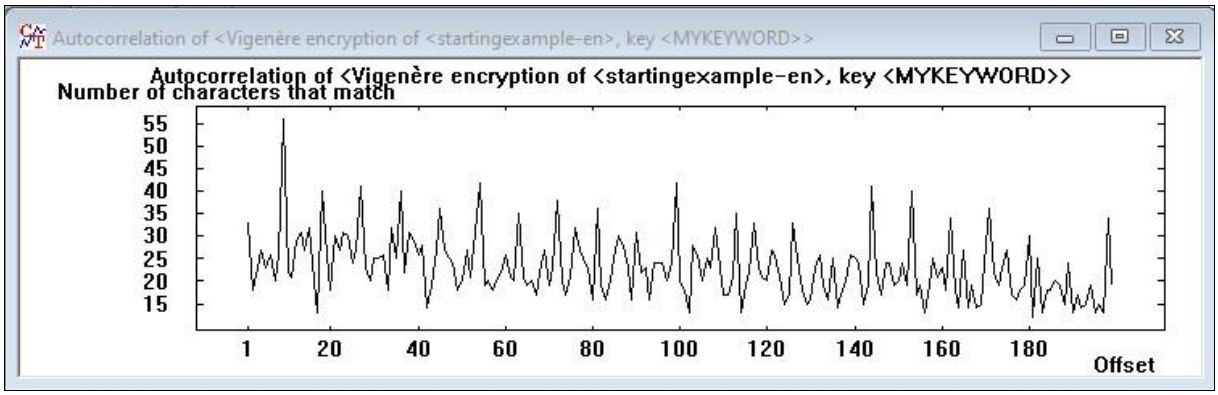


(b)

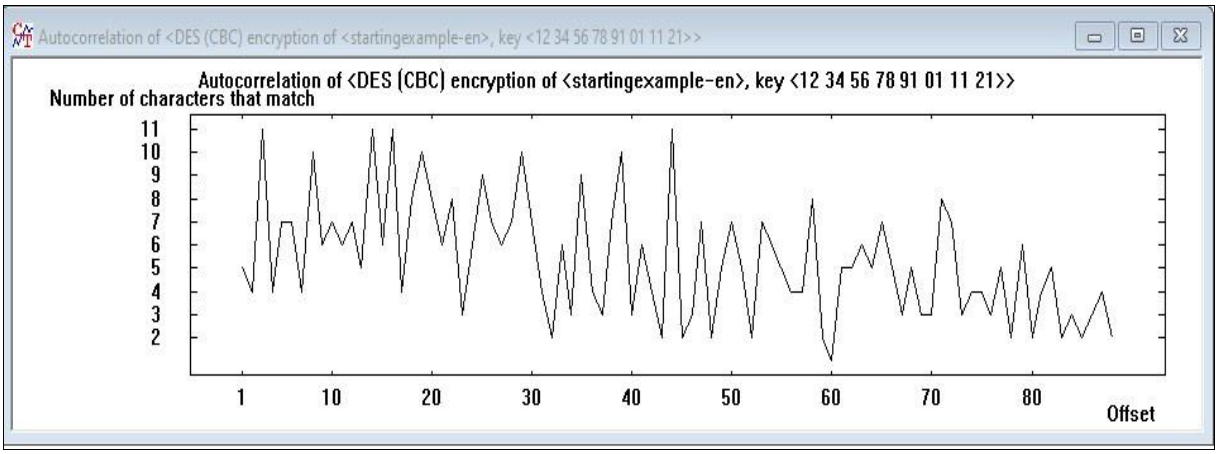


(c)

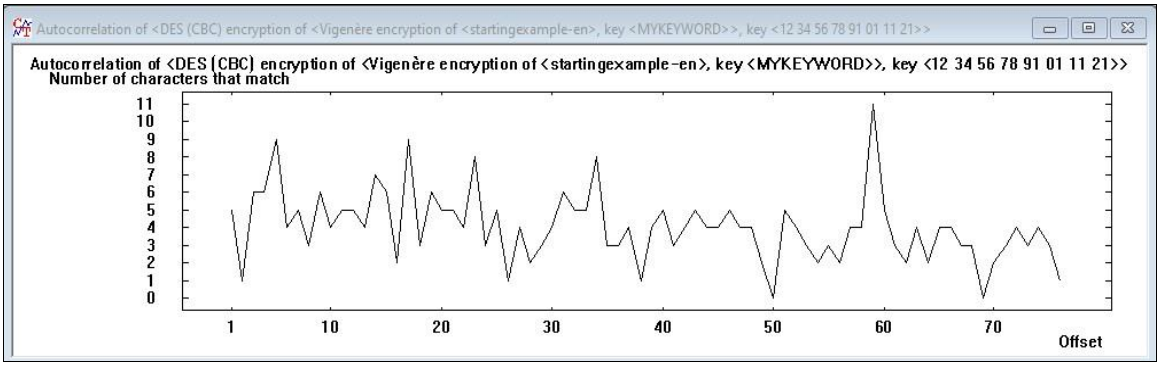
Fig 20: Histogram of Cipher Text for Three Cases of Experiment 2 (a) Case 1 (b) Case 2 (c) Case 3



(a)



(b)



(c)

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

As revealed 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. The last step for Case 3 of the experiment is the autocorrelation results of Case 3; these show that the characters are no longer correlated, showing that Case 3 is the best of the three cases from Experiment 2. Also, Case 3 of Experiment 2 is the gold standard of the CBC-DES method. To improve the efficiency of the CBC-DES method and increase the complexity of the cryptanalysis of the CBC-DES method, the Vigenère cipher was used with the CBC-DES method to increase the security of the CBC-DES method. The reason why the key search space is 2m, is because, as stated above, in the first test case of both tests described above, if a hacker knows the key of the algorithm then the hacker will be able to successfully break into the encrypted message. Therefore, since in the second test case of both experiments, all the hacker needs to know is the DES key, therefore the key search space is 264. The use of a smaller key search space makes it easier for hackers to perform brute-force attacks on the message. In Table 5, there are many examples of this. In addition, the third test case for both experiments, involves the proposed system's hybrid encryption scheme; the third test case shows how much longer it takes a hacker to obtain the key to decrypt the message. 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	264
	3	264*2m
2	1	2^m
	2	264
	3	264*2m

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 to that, the search space of a key was increased in case-3 (2m*264), with the proposed multi-level model; it used two keys for two algorithms: the DES algorithm, whose key size is 64 bits, and the Vigenere method, whose key size is m bits.

References

1. Ashty M, AblhdA Z. New cryptography method based on Hill and Rail Fence algorithms. Diyala J Eng Sci. 2017;10(1):39-47.
2. Shaker SA, Naser AG, Ali FH. New design of efficient non-linear stream key generator. In: Proc 4th Int Sci Conf Eng Sci Adv Technol; 2022.
3. Ali NA, Rahma AS, Shaker SH. 3D content encryption using multi-level chaotic maps. Iraqi J Sci. 2023;64(5):2521-2532.
4. Al-Karkhi AAS, Hassan NF, Azeez RA. A secure private key recovery based on DNA bio-cryptography for blockchain. Iraqi J Sci. 2023;64(2):958-972.
5. Elwinus HAM, Purba EY, Siahaan BY, Sembiring RW. Collaborative encryption algorithm between Vigenère cipher, rotation of matrix (ROM), and one-time pad (OTP) algorithm. Adv Sci Technol Eng Syst J. 2017;2(5):13-21.
6. Hoobi MM. Improved structure of data encryption standard algorithm. J Southwest Jiaotong Univ. 2020;55(5):1-10.
7. Soofi AAS, Riaz I, Rasheed U. An enhanced Vigenère cipher for data security. Int J Sci Technol Res. 2016;5(3):141-145.
8. Sciacovelli A, Vittorio V, Enrico S. Entropy generation

analysis as a design tool: a review. Renew Sustain Energy Rev. 2015;43:1167-1181.
9. Brumen T, Makari T. Resilience of students' passwords against attacks. In: Proc 40th Int Conv Inf Commun Technol Electron Microelectron (MIPRO); 2017.
10. Hoobi MM. Strong triple data encryption standard algorithm using Nth degree truncated polynomial ring unit. Iraqi J Sci. 2017;58(3C):1760-1771.
11. Mahmoud HM, Hoobi MM. Improved Rijndael algorithm by encryption S-Box using NTRU algorithm. Iraqi J Sci. 2015;56(4A):2982-2993.
12. Harba ESI. Advanced password authentication protection by hybrid cryptography and audio steganography. Iraqi J Sci. 2018;59(1C):600-606.
13. Shaker SA, Nasir AG, Ali FH. Constructing a digital certificate authentication system for classified documents. Iraqi J Sci. 2023;64(3):1391-1400.
14. Rachmawati D, Lubis AN. Combining Beaufort cipher and RSA-CRT algorithm in a hybrid scheme to secure images. In: Proc 2nd TALENTA-Int Conf Sci Technol; 2023.
15. Sari RN, Hayati RS. Beaufort cipher algorithm analysis based on the Power Lock-Blum Blum Shub in securing data. In: Proc 6th Int Conf Cyber IT Serv Manag (CITSM); 2018.
16. Sumartono I, Siahaan APU, Mayasari N. An overview of the RC4 algorithm. IOSR J Comput Eng. 2016;18(6):67-73.
17. Biswas SH, Ali MA, Rahman M, Sohel K, Hasan M, Sarkar K, *et al*. A systematic study on classical cryptographic cipher to design a smallest cipher. Int J Sci Res Publ. 2019;9(12):507-511.
18. Ashok S, Kiran R, Pradeep S, Devara R. A new variant of rail fence cipher using hybrid block-swap method. Int Res J Eng Technol. 2021;8(7):1735-1739.
19. Yousif YE. Improving the efficiency of DES algorithm using neural networks. Int J Eng Appl Sci Technol. 2020;5(1):26-29.
20. Subhi RMZ. DES encryption and decryption algorithm implementation based on FPGA. Indones J Electr Eng Comput Sci. 2020;18(2):774-781.
21. Stallings W. *Cryptography and network security: principles and practice*. 7th ed. Pearson India; 2018.
22. Lakshmi PS, Murali G. Comparison of classical and quantum cryptography using QKD simulator. In: Proc Int Conf Energy Commun Data Anal Soft Comput (ICECDS); 2017. p. 3543-3547.
23. Budiman MA, Rachmawati D, Jessica. Implementation of super-encryption with Trithemius algorithm and double transposition cipher in securing PDF files on Android platform. In: Proc 2nd Int Conf Comput Appl Inform; 2017.
24. Ramadhan Z, Putera A, Siahaan U. Protection of important data and information using Gronsfeld cipher. Int J Innov Res Multidiscip Field. 2018;4(10):128-132.
25. Harba ES, Harba HS, Abdulmunem IA, Hussein SS. Improving security of the CryptoStego approach using time sequence dictionary and spacing modification techniques. Iraqi J Sci. 2021;62(5):1721-1733.
26. Hoobi MM. Keystroke dynamics authentication based on naïve Bayes classifier. Iraqi J Sci. 2015;56(2A):1176-1184.
27. Hoobi MM. Modified robust AES architecture. Technol Rep Kansai Univ. 2020;26(10).
28. Subramani S, Munuswamy S, Arputharaj K, Svn SK. Review of security methods based on classical cryptography and quantum cryptography. Cybern Syst. 2023.
29. Deepthi DVV, Benny BH, Sreenu K. Various ciphers in classical cryptography. J Phys Conf Ser. 2019;1228:012014.
30. Abdulla QZ, Al-Hassani MD. Robust password encryption technique with an extra security layer. Iraqi J Sci. 2023;64(3):1477-1486.
31. Nahar J, Chakraborty P. Improved approach of rail fence for enhancing security. Int J Innov Technol Explor Eng. 2020;9(9):583-585.
32. Hoobi MM. Efficient hybrid cryptography algorithm. J Southwest Jiaotong Univ. 2020;55(3):1-9.
33. Hoobi MM. Survey: efficient hybrid algorithms of cryptography. MINAR Int J Appl Sci Technol. 2020;2(4):1-16.
34. Hoobi MM, Sulaiman SS, AbdulMunem IA. Enhanced multistage RSA encryption model. IOP Conf Ser Mater Sci Eng. 2020;455.
35. Brown DC. A cryptanalysis of the autokey cipher using the index of coincidence. In: Proc ACMSE 2018 Conf; 2018. p. 1-8.
36. Rubinstein-Salzedo S. Other types of ciphers. In: *Cryptography*. Springer; 2018. p. 63-73.
37. Agustini S, Rahmawati WM, Kurniawan M. Modified Vigenère cipher to enhance data security using monoalphabetic cipher. Int J Artif Intell Robot. 2019;1(1):26-32.
38. Zhu H, Li Z. An efficient biometric authenticated protocol for arbitrary-domain-server with blockchain technology. Int J Netw Secur. 2021;23(3):386-394.
39. Dubey P, Yadav O. A survey on quantum cryptography versus classical cryptography. Int J Curr Eng Technol. 2020;10(6):910-913.
40. Kareem SM, Rahma AMS. A modification on key stream generator for RC4 algorithm. Eng Technol J. 2020;38(28):54-60.
41. Saraswata A, Khatria C, Sudhakara, Thakrala P, Biswas P. An extended hybridization of Vigenère and Caesar cipher techniques for secure communication. In: Proc 2nd Int Conf Intell Comput Commun Converg; 2016. p. 355-360.
42. Nasution SD, Ginting GL, Syahrizal M, Rahim R. Data security using Vigenère cipher and Goldbach codes algorithm. Int J Eng Res Technol. 2017;6(1):360-363.
43. Darari R, Winarko E, Damayanti A. Encryption and decryption application on images with hybrid algorithm Vigenère and RSA. Contemp Math Appl. 2020;2(2):109-117.
44. Hussein AA, Ayoob NK. Key generation for Vigenère ciphering based on genetic algorithm. J Univ Babylon. 2022;30(1):200-208.
45. Ali SM, Mahmood NT, Yousif SA. Meerkat clan algorithm for solving N-Queen problems. Iraqi J Sci. 2021;62(6):2082-2089.
46. Shukur WA, Qurban LK, Aljuboori A. Digital data encryption using a proposed W-method based on AES and DES algorithms. Baghdad Sci J. 2023;20(4):1414-1424.
47. Abdulameer SA, Kashmar AH, Shihab AI. A cryptosystem for database security based on TSFS algorithm. Baghdad Sci J. 2020;17(2):567-574.
48. Khudair ET, Naser EF, Mazher AN. Comparison between RSA and CAST-128 with adaptive key for video frames encryption with highest average entropy. Baghdad Sci J. 2022;19(6):1378-1386.
49. Somsuk K. A new methodology to find private key of RSA based on Euler totient function. Baghdad Sci J. 2021;18(2):338-348.