Modified Vigenère cipher algorithm based on new key generation method

Thamer Hassan Hameed[1], Haval Tariq Sadeeq[2]
[1]Basic Sciences Department, College of Agricultural Engineering Sciences, University of Duhok, Duhok, Iraq
[2]Information Technology Department, Technical College of Informatics-Akre, Duhok Polytechnic University, Duhok, Iraq

Article Info

Article history:

Received Apr 12, 2022
Revised Jun 27, 2022
Accepted Jul 29, 2022

Keywords:

Cryptography
Keystream generation
Modified Vigenère cipher
Symmetric encryption
Vigenère cipher

ABSTRACT

Nowadays, as communication and network technologies evolve in modern life, ensuring the confidentiality of a cryptography system has become a critical requirement. The Vigenère cipher is attracting the attention of cryptography specialists, although the Vigenère cipher algorithm has a problem. The problem is due to a repeating encryption key. As a result of the multiple cryptographic approaches described in the literature, this paper proposes a novel encryption strategy for safe and secure data exchange by utilizing a new key generation process. The proposed encryption approach avoids the issue of repeating keys. Additionally, the classic Vigenère cipher encrypts the plaintext using a 26x26 Vigenère table, the researcher modified the original Vigenère table to 95x95, which adds more potential letters, mathematical symbols, numerals, and punctuation to a standard QWERTY keyboard layout. Additionally, the researcher added case sensitivity. To observe the performance of the proposed method, the index of coincidence and entropy have been calculated. The obtained results confirm the high performance of the proposed algorithm in comparison to the other algorithms used in this paper. The primary goal of this paper is to make cryptanalysis extremely complex and to promote data security.

Corresponding Author:

Haval Tariq Sadeeq
Information Tecchnology Department, Technical College of Informatics-Akre
Duhok Polytechnic University
61 zakho Road, 1006 Mazi Qr Duhok, Kurdistan, Iraq
Email: haval.tariq@dpu.edu.krd

1. INTRODUCTION

Communication has gotten faster and easier in this century as a result of advancements in networking technology and the Internet. Thus, data security is gaining prominence as a vital issue each day. Cryptography protects the integrity of data transmitted across a variety of channels. Cryptography employs mathematical functions to perform encryption and decryption. The function makes use of a key that is utilized for both encryption and decryption [1], [2]. A cryptographic algorithm's strength is contingent upon its ability to retrieve the key value from the domain space. Thus, the algorithm's strength is proportional to the time required to get the key. Cryptanalysis is the study and decryption of cryptographic methods. Plaintext is converted to ciphertext using an encryption technique, and the plaintext is recovered using decryption algorithms and keys. Private key (or symmetric) algorithms and public key (or asymmetric) algorithms are the two basic categories of algorithms. Asymmetric encryption uses a single key for both encryption and decryption, whereas symmetric encryption employs different keys on both sides [3]. The necessity for cryptography arises from the possibility of an observing opponent who is interested in learning the contents of the communication between the communicating parties. It is assumed that an eavesdropping adversary is knowledgeable and capable of decrypting the ciphertext if he has the key. In an attempt to locate the encryption key, an adversary may conduct an exhaustive search within the key space. Thus, in cryptography, a vast key space that cannot be exhaustively searched in a reasonable amount of time is a needed. However, a huge key space does not ensure security, as demonstrated by the easy-to-break mono-alphabetic substitution cipher. Thus, a large key space is a necessary but not sufficient criterion [4]. Modern cryptography holds that an encryption system is safe if an opponent cannot decode the ciphertext to derive any function of the plaintext. As a result, a powerful mathematical encrypting function that an opponent cannot decrypt in real polynomial time is required for secure encryption.

Blaise de Vigenère, a French mathematician, came up with an innovative polyalphabetic cipher called the Vigenère cipher, which relates to symmetric encryption [5]. To look at it another way, it was a cipher that could readily be broken using a frequency analysis of letters, like the Caesar cipher. Numeric values for each of the 26 letters of the alphabet are assigned based on their position in the alphabetical sequence, from 0 to 25. Figure 1 illustrates the fundamental configuration of private-key encryption, including ciphertext. Symmetric encryption, also known as conventional encryption or single-key encryption, was the sole type of encryption that was used prior to the introduction of public-key encryption in the 1970s. In the context of encryption, it's worth noting that all of the most commonly used algorithms are private-key. Symmetric encryption schemes are made up of the following five parts: i) “Plaintext” - original message, ii) “Encryption algorithm” - accomplishes plaintext substitutions and transformations, iii) “Secret key” - ensure that the encryption algorithm uses the correct substitutions/transformations, iv) “Ciphertext” - scribble text, and v) “Decryption algorithm” - an encryption technique that is the contrary of.

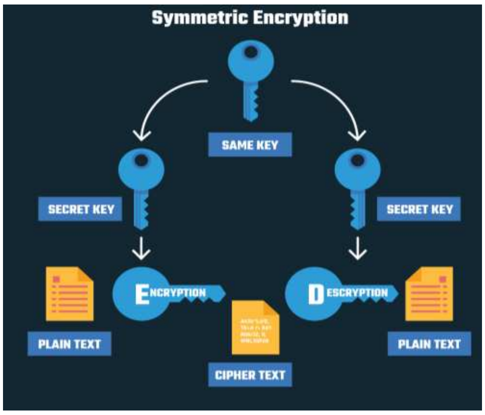


Figure 1. Symmetric encryption

2. RELATED WORK

Researchers have made various improvements to the Vigenère cipher during the last few years to bring it up to date with today's higher security requirements. In order to make the Vigenère cipher more secure, random padding bits were inserted into the original code [6]. Encryption algorithms that automatically change the cipher key after each encryption step are being developed. During the encryption process, the new approach will employ consecutive keys that are reliant on the starting key value. Text can be encrypted and decrypted using the technique [7].

Rahmani [8] increases the character set of the classic Vigenère cipher from the 26 English alphabets to 92 characters. More messages, including passwords and other transactions, can be supported because of the additional character set's large size. Both the message and the key are more difficult to decipher when symbols are used instead of the English alphabets. Kester [9] introduced a new hybrid encryption method for plaintext. A columnar transposition cipher is used to decrypt the plaintext, and the ciphertext is used to decrypt the plaintext by using Vigenère cipher. The ciphertext was deciphered using cryptanalysis at the end.

Omolara *et al.* [10] presented a Caesar-Vigenère cipher hybrid with diffusion and confusion that classical ciphers could not match. By including alphabets, numerals, and symbols into the modified ciphers, the Caesar and Vigenère ciphers have become completely incomprehensible and diffuse.

Subandi *et al.* [11] offers an improved encryption technique that enhances the reliability of the Vigenère approach via being used with modern ciphers such as Stream Ciphers. The combination cipher described above has a high level of security when using the proposed technique, whereas a cipher based solely on the Vigenère method does not.

The keystream generator of the Vigenère encryption was modified for the proposed three-pass protocol [12]. As a final step, 26-character protection messages utilizing the standard alphabet were implemented. In order to overcome the low security and low computational efficiency of the standard confusion-diffusion framework, a unique key substitution encryption architecture was presented. It uses key scheming and substitution to encrypt various types of images [13].

The Caesar cipher method can be modified to produce ciphertext that can be deciphered. Cryptanalysis will not be concerned if the ciphertext can be deciphered. The consonant alphabet was replaced with a consonantal alphabet, and the alphabet was split into two sections to accommodate the new vocalizations [14]. On the basis of key domain maximization in a finite field, [15] came up with a more secure encryption-decryption approach. Keys for encryption and decryption are obtained using a random main key in the suggested method.

3. VIGENÈRE CIPHER

In the field of polyalphabetic substitution and symmetric key cryptography, Vigenère cipher is a traditional algorithm that makes use of the same keys for both encryption and decryption. This Cipher uses a table called tabula recta as illustrated in Table 1, which is a 26×26 matrix comprising alphabet letters, to encrypt and decode data [16]. The Vigenère cipher's encryption and decryption can be understood in the (1) and (2).

$$CTi = (PTi + Key)\%26 \tag{1}$$

$$PTi = (CTi - Key)\%26 \tag{2}$$

Table 1. Tabula Recta

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

Where CT is the ciphertext, PT is the plaintext, Key is the Key. Alphabet ciphertext is the intersection of the ciphertext's plaintext and alphabet keys, and it is used to encrypt sensitive information. There are times when the key alphabet is less than the plaintext, hence the key will be repeated until the plaintext is equal to that of the key.

Key repetition is an issue when the length of the key is less than the length of the plaintext. in Vigenère cipher, because the algorithm most likely produces the same ciphertext for plaintexts of equal length. For example, an encryption of the message “C R Y P T O G R A P H Y” using the key “L U C K” gives ciphertext “N L A Z E I I B L J J I” as it declared in Table 2.

Table 2. Vigenère example

PT	*C*	*R*	*Y*	*P*	*T*	*O*	*G*	*R*	*A*	*P*	*H*	*Y*
ID	2	17	24	15	19	14	6	17	0	15	7	24
Key	L	U	C	K	L	U	C	K	L	U	C	K
ID	11	20	2	10	11	20	2	10	11	20	2	10
CT	N	L	A	Z	E	I	I	B	L	J	J	I
ID	13	11	0	25	4	8	8	1	11	9	9	8

Where ID is the index of the letter.

There are certain disadvantages to The Vigenère cipher, such as a key length that is too short for the plaintext length. This means that the key will be repeated, which cryptanalysts can utilize this to decrypt the ciphertext. Vigenère cipher techniques were broken by the Kasiski method, which uses the same characters in the ciphertext to determine the distance to the key length. For an exhaustive key search, the next step is to identify the keywords that should be used [17].

3.1. Cryptanalysis of Vigenère cipher

It is possible to discern some similarities in the Vigenère cipher, such as the fact that if “H” is the most commonly occurring symbol in a cipher text, one may assume that “H” corresponds to “E”, since “E” is perhaps the most commonly used letter in English. This is because E can be encoded as multiple ciphertext characters at various times in the message, making it impossible to do a basic frequency analysis of the message.

This cipher's primary flaw is that its key repeats itself. Ciphertext can be deciphered if a cryptanalyst correctly guesses the key's length, which makes it possible to break individual Caesar ciphers. The length of the key can be determined using the Kasiski and Friedman tests.

The Kasiski test, often known as the Kasiski examination, exploits the fact that, by chance, some words may be encrypted using the same key letters, resulting in repeating groups in the cipher text.
key length can be used to rewrite the ciphertext into how many columns the key is known or guessed. A single Caesar cipher is used to encrypt each column of plaintext. The cipher text's letters can be decoded using techniques similar to those employed to decipher the Caesar cipher.

It is worth mentioning, that swarm intelligence algorithms such as particle swarm optimization [18], firefly algorithm [19], and bird mating optimizer [20]. can be used for cryptoanalysis of Vigenère cipher and recently, there are a number of swarm intelligence algorithms have been applied for that purpose such as [21]-[24].Finally, it is noticed that neural networks [25], [26] which is a typical model for machine learning and application is used also for cryptoanalysis of Vigenère cipher [27].

4. PROPOSED METHOD

In this paper, firstly, the proposed approach aims to increase the original Vigenère cipher's 26-character length to a 95-character case-sensitive cipher that includes numerals and other regularly used English symbols. The new Vigenère table is shown in Table 3. Then, after receiving a user key, the process of keystream generation obtains the preliminary fills (as a user key). When the process reaches Kn, the insert from the preceding process becomes the next one in line, and so on. To generate a key when the user-specified key is not equal to the plaintext, the key length must be more than the plaintext length. However, if the key length is equal to the plaintext length, the key generating process is not required. The (3) is used to generate the keys:

$$Ki = (Ki + 3 + r)\%95 \tag{3}$$

where Ki is a character key that will be generated, $Ki + 3$ is the index of the third key character and r is the remaining length of the keys to the length of the plaintext.

However, if the length of the key is equal to the length of the plaintext, then the following formula can be used (4).

$$CT = (PT + Key)\%95 \tag{4}$$

It is worth mentioning that for decryption process, when the length of key equals the length of ciphertext then the following formula can be used (5).

$$PT = (CT - Key)\%95 \tag{5}$$

The flow chart of both encryption and decryption process is shown in Figure 2 and Figure 3 respectively.

Table 3. The proposed 95x95 Vigenère table

		PLANTEXT																																									
			!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~
			!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~
	!	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~	
	"	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!
	#	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"
	$	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#
	%	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$
	&	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%
	'	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&
	(	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'
	)	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(
	*	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)
	+	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*
	,	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+
	-	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,
	.	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-
	/	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.
	0	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
	…	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0
	9	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…
KEY	:	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9
	;	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:
	<	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;
	=	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<
	>	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=
	?	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>
	@	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?
	A	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@
	…	…	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A
	Z	Z	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…
	[	[	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z
	\	\	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[
	]	]	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\
	^	^	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]
	_	_	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^
	`	`	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_
	a	a	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`
	…	…	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a
	z	z	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…
	{	{	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z
	|	|	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{
	}	}	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|
	~	~		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/	0	…	9	:	;	<	=	>	?	@	A	…	Z	[	\	]	^	_	`	a	…	z	{	|	}

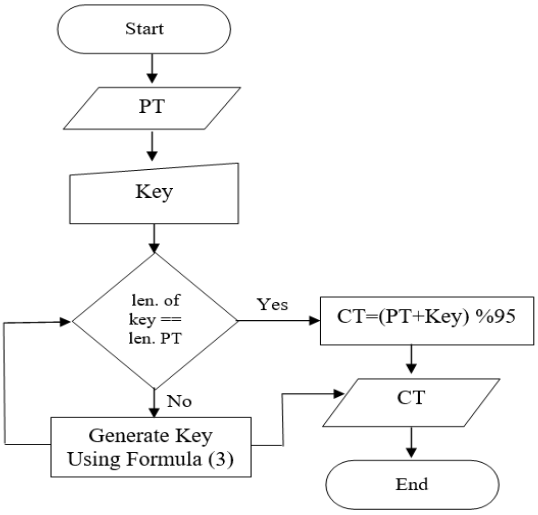


Figure 2. Modified Vigenère cipher encryption process

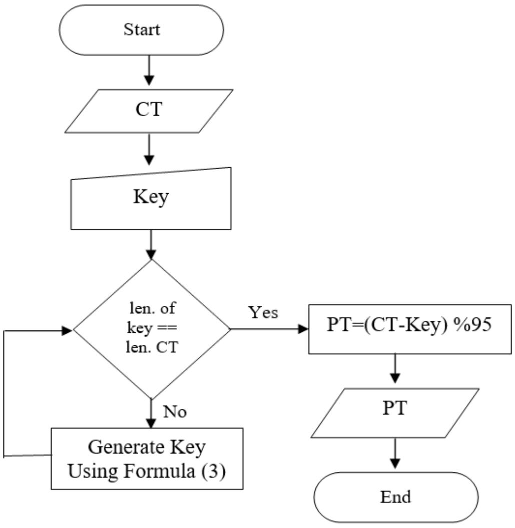
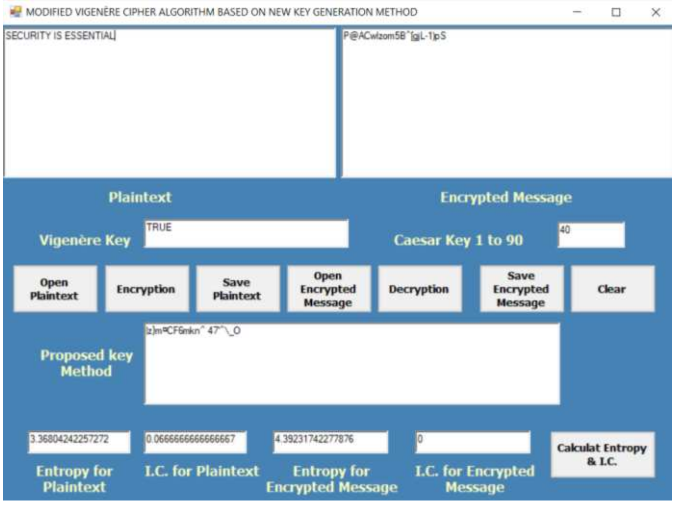


Figure 3. Modified Vigenère cipher decryption process

5. EXPERIMENTAL RESULTS

The proposed method has been evaluated in a real-world application. The model was developed in visual studio C# and encrypts plain text into cipher text using the user-supplied key or decrypts the cipher text using the keyword. The project's main side is depicted in Figure 4.


MODIFIED VIGENÈRE CIPHER ALGORITHM BASED ON NEW KEY GENERATION METHOD

Plaintext: SECURITY IS ESSENTIAL

Encrypted Message: P@ACwlzom5B`[gL-1}pS

Vigenère Key: TRUE

Caesar Key 1 to 90: 40

Proposed key Method: z}m¤CF6mkn^ 47`_O

Entropy for Plaintext: 3.36804242257272

I.C. for Plaintext: 0.0666666666666667

Entropy for Encrypted Message: 4.39231742277876

I.C. for Encrypted Message: 0

Figure 4. C# interface of the proposed method

In order to evaluate the performance of the proposed algorithm, the index of coincidence (IC) and entropy have been calculated. The IC determines the chance that two characters in a string are identical when chosen at random. The IC for a particular letter-frequency distribution can be calculated mathematically as (6):

$$IC = \frac{\sum_{i=a}^{i=z} f_i(f_i - 1)}{N(N-1)} \qquad (6)$$

where fi represents the number of times the given litter appears in the ciphertext and N refers to the total number of letters in the ciphertext.

In order to compute both the index of coincidence and entropy, consider an example plaintext “SECURITY IS ESSENTIAL” with Key “TRUE”. When this is used for our modified encryption method, the ciphertext is: “P@ACwlzom5B^[gjL-1)pS”.

Table 4 compares the newly proposed method to some other methods in terms of index of coincidence and entropy. The index of coincidence of the proposed method is calculated to be 0 and entropy is 4.392317 bits. This improved and increased the security of the Vigenère cipher and indicates that the new method’s effectiveness is extremely high due to its hybrid design.

Table 4. Comparison of the proposed method along with some methods

Method	IC	Entropy
Vigenere Cipher	0.02925	3.7216
Double columnar	0.02925	3.7216
Columnar and Vigenere cipher	0.05849	3.4058
Modified Caesar Cipher and Vigenere	-	4.2479
Proposed method	0	4.392317

6. CONCLUSION

According to the research, the classical algorithm Vigenère cipher appears to have a better degree of trust than the ordinary Vigenère cipher when the keys are reconfigured. This is because the keys are adjusted in such a way that when the length of the key exceeds the length of the plaintext, the key is not repeated but created by a function. As a result of not needing to repeat the key, more random keys are generated. In this paper, a new encryption strategy for safe and secure data exchange has been proposed. It is worth mentioning that the proposed encryption approach avoids the issue of repeating keys by utilizing a novel key generation process. The main aim of this paper is to make cryptanalysis complicated and to focus on promoting data security. As evidenced by the literature, various modifications were performed based on a different method of transposition in order to generate the key for the Vigenère cipher. It's worthwhile to mention that in order to validate the performance of the proposed method, index of coincidence and entropy have been computed. The results clearly demonstrated that the suggested method's superior performance as compared to the other techniques used in this paper.

REFERENCES

[1] D. R. Ibrahim, J. Sen Teh, and R. Abdullah, “An overview of visual cryptography techniques,” *Multimedia Tools and Applications*, vol. 80, no. 21–23, pp. 31927–31952, Sep. 2021, doi: 10.1007/s11042-021-11229-9.

[2] X. You *et al.*, “Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts,” *Science China Information Sciences*, vol. 64, no. 1, p. 110301, Jan. 2021, doi: 10.1007/s11432-020-2955-6.

[3] N. A. Kako, H. T. Sadeeq, and A. R. Abrahim, “New symmetric key cipher capable of digraph to single letter conversion utilizing binary system,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 2, p. 1028, May 2020, doi: 10.11591/ijeecs.v18.i2.pp1028-1034.

[4] D. K. Sharma, N. C. Singh, D. A. Noola, A. N. Doss, and J. Sivakumar, “A review on various cryptographic techniques &amp; algorithms,” *Materials Today: Proceedings*, vol. 51, pp. 104–109, 2022, doi: 10.1016/j.matpr.2021.04.583.

[5] A.-A. Mohammed and A. Olaniyan, “Vigenere Cipher: trends, review and possible modifications,” *International Journal of Computer Applications*, vol. 135, no. 11, pp. 46–50, Feb. 2016, doi: 10.5120/ijca2016908549.

[6] P. Wilson and M. Garcia, “A modified version of the Vigenère algorithm,” *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 6, no. 3, pp. 140–143, 2006.

[7] Q. Kester, “A cryptosystem based on Vigenère cipher with varying key,” *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, pp. 108–113, 2012.

[8] M. K. I. Rahmani, “Alpha-qwerty Cipher: an extended Vigenere Cipher,” *Advanced Computing: An International Journal*, vol. 3, no. 3, pp. 107–118, May 2012, doi: 10.5121/acij.2012.3311.

[9] Q.-A. Kester, “A hybrid cryptosystem based on Vigenere Cipher and columnar transposition Cipher,” vol. 3, no. 1, pp. 141–147, Jul. 2013, [Online]. Available: http://arxiv.org/abs/1307.7786.

[10] O. E. Omolara, A. I, Oludare, and S. E. Abdulahi, “Developing a modified hybrid Caesar Cipher and Vigenere Cipher for secure data communication omolara computer engineering and intelligent systems,” *Computer Engineering and Intelligent System*, vol. 5, no. 5, pp. 34–46, 2014.

[11] A. Subandi, R. Meiyanti, C. L. M. Sandy, and R. W. Sembiring, “Three-pass protocol implementation in Vigenere Cipher classic cryptography algorithm with keystream generator modification,” *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 5, pp. 1–5, Jun. 2017, doi: 10.25046/aj020501.

[12] Y. Song, Z. Zhu, W. Zhang, H. Yu, and Y. Zhao, “Efficient and secure image encryption algorithm using a novel key-substitution architecture,” *IEEE Access*, vol. 7, pp. 84386–84400, 2019, doi: 10.1109/ACCESS.2019.2923018.

[13] B. Purnama and A. H. H. Rohayani, “A new modified Caesar Cipher cryptography method with legibleciphertext from a message to be encrypted,” *Procedia Computer Science*, vol. 59, no. Iccsci, pp. 195–204, 2015, doi: 10.1016/j.procs.2015.07.552.

[14] D. N. Uniyal, D. G. Dobhal, and M. P. Semwal, “Enhanced security of encrypted text by KDMT: key-domain maximization technique,” *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 5, pp. 1385–1388, Jan. 2020, doi: 10.35940/ijrte.E6326.018520.

[15] R. S Kartha and V. Paul, "Survey: recent modifications in Vigenere Cipher," *IOSR Journal of Computer Engineering*, vol. 16, no. 2, pp. 49–53, 2014, doi: 10.9790/0661-16294953.

[16] F. Mushtaq and S. Ali, "Enhancing security of Vigenere Cipher by Stream Cipher," *International Journal of Computer Applications*, vol. 100, no. 1, pp. 0975 – 8887, 2014.

[17] A. L. Hananto, A. Solehudin, A. S. Y. Irawan, and B. Priyatna, "Analyzing the Kasiski method against Vigenere Cipher," vol. 6, no. 6, pp. 1–8, Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.04519.

[18] S. Sengupta, S. Basak, and R. Peters, "Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 157–191, 2018, doi: 10.3390/make1010010.

[19] H. Sadeeq and A. M. Abdulazeez, "Hardware implementation of firefly optimization algorithm using FPGAs," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*, Oct. 2018, pp. 30–35, doi: 10.1109/ICOASE.2018.8548822.

[20] A. Askarzadeh, "Bird mating optimizer: an optimization algorithm inspired by bird mating strategies," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 1213–1228, Apr. 2014, doi: 10.1016/j.cnsns.2013.08.027.

[21] A. K. Bhateja, A. Bhateja, S. Chaudhury, and P. K. Saxena, "Cryptanalysis of Vigenere cipher using cuckoo search," *Applied Soft Computing*, vol. 26, pp. 315–324, Jan. 2015, doi: 10.1016/j.asoc.2014.10.004.

[22] A. Bhateja, S. Kumar, and A. K. Bhateja, "Cryptanalysis of Vigenere Cipher using Particle Swarm Optimization with Markov chain random walk," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 5, no. 05, pp. 422–429, 2013.

[23] Z. Kochladze and G. Gelashvili, "Using genetic algorithm for the breaking Vigenere Cipher," *Computer Science & Telecommunications*, vol. 2, no. 2, pp. 53–57, 2017.

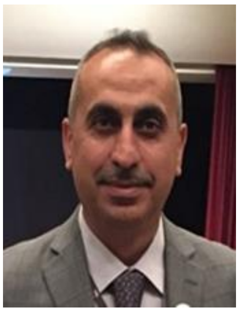
[24] R. Navatejareddy, M. Jayabhaskar, and B. Sathyanarayana, "Elliptical curve cryptography image encryption scheme with aid of optimization technique using gravitational search algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 1, pp. 247–255, 2022, doi: 10.11591/ijeecs.v25.i1.pp247-255.

[25] H. T. Sadeeq, T. H. Hameed, A. S. Abdi, and A. N. Abdulfatah, "Image compression using neural networks: a review," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 17, no. 14, pp. 135–153, Dec. 2021, doi: 10.3991/ijoe.v17i14.26059.

[26] D. A. Q. Shakir and A. J. Dawood, "3D chaos graph deep learning method to encrypt and decrypt digital image," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 941–951, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp941-951.

[27] M. A. Budiman, Handrizal, and William, "A neural cryptography approach for digital image security using Vigenre cipher and tree parity machine," *Journal of Physics: Conference Series*, vol. 1898, no. 1, 2021, doi: 10.1088/1742-6596/1898/1/012039.

BIOGRAPHIES OF AUTHORS



Thamer Hassan Hameed received a B.Sc. degree in Soil Science College of Agriculture and Forestry; University of Mosul, Iraq, received a B.Sc. degree in Computer science, College of Computers and Mathematical Sciences. University of Mosul, Iraq, the M.Sc. degree in Computer Sciences, University of Zakho, Kurdistan Region, Iraq. He is a lecturer at Branch, Basic Sciences, College of Agricultural Engineering Sciences, University of Duhok, Kurdistan Region of Iraq. His interests are Swarm Intelligence, Machine Learning and Image Processing. He can be contacted at email: thamer.hameed@uod.ac.

Haval Tariq Sadeeq is a lecturer in the Information Technology Department at Duhok Polytechnic University in Duhok, Kurdistan Region of Iraq. Currently, he is a PhD student at the Technical College of Informatics/DPU. His interests are metaheuristic algorithms, swarm intelligence, stochastic optimization, global and engineering optimization, data cryptography and compression. He can be contacted at email: haval.tariq@dpu.edu.krd.