Article can be accessed online at http://www.publishingindia.com

Enhanced Caeser Cipher to Exclude Repetition and Withstand Frequency Cryptanalysis

Abdulkadir H. Disina*, Zahraddeen A. Pindar, Sapiee Bin Hj. Jamel*****

Abstract

Cryptography is the art of encoding messages into an unreadable form from a sender and re-transforming back to its readable form at the receiver end. This is an enhanced Caesar cipher against frequency analysis using bidirectional shift. This algorithm encrypts message bit by bit or character by character (stream cipher) and uses one key ideology (symmetric key cipher), the sender encrypts the message before transmitting and the receiver decrypts upon receiving using the same key as the one used for the encryption. It shifts the plain text characters to different direction which eliminates repetition of characters in the cipher text. Previous versions of Caesar cipher hadonly 26 English alphabets to be encrypted, which the attacker knew that there were only 26 choices to choose from, to determine the plain text. And when the alphabet of the same type are encrypted, they will have the same symbol representing each, which gives hint to attacker on how to break it by using frequency cryptanalysis. The enhanced method (Enhanced Caesar cipher) has 95 characters as digital messages as against 26 alphabets. Based on this method, the sender will transposition the bits in the message according to their sequence arrangement (odd and even position) to shift the characters in the odd position to the left and characters in the even position to the right side, based on the key given by the user, as the key to both shifts. Shifting the plain text to different directions mitigates the problem of repetition which the previous version suffers from. To make decryption more difficult, each character will switch position with the next to randomize their arrangement. The cipher was successfully developed and working accurately. It was developed on java platform using java eclipse IDE and NETBEANS IDE 7. 1. 2. It has successfully passed all the tests and proven its accuracy obtaining the result of 100% repetition free. Thus, the proposed method is highly resistant to frequency analysis. This provides more security than the earlier versions and it serves as an option to be integrated with other algorithms to strengthen the security.

Keyword: Caesar Cipher, Cryptography, Repetition Exclusion, Encryption, Decryption, Shift, Switch, Symmetric-Key Cipher, Key Generation

Introduction

In this research, a proposed modified version of Caesar cipher is introduced which at the end of the research achieves exclusion of repetitive characters in the message, when it is to be encrypted. It also randomizes the characters in the processes of encryption to make it very difficult for the crypto analyst to decipher it using frequency analysis (Dey, 2012a). In modern world, cryptography hackers try to break a code or cryptographic algorithm or try to retrieve the key, which is needed to encrypt a message, by analyzing the insertion or presence of repetitive bits/ characters (bytes) in the message and encrypted message

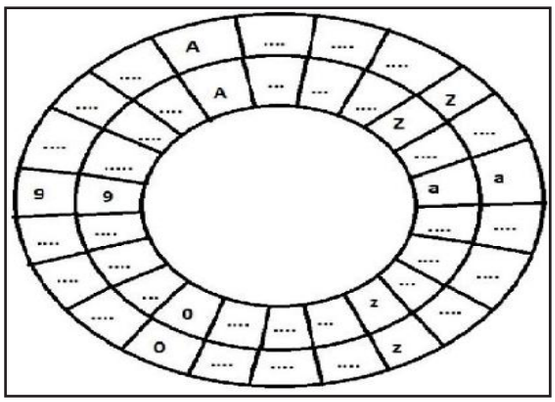
* Department of Information Security, UniversitiTun Hussein Onn Malaysia, Batu Pahat, Malaysia.
E-mail: deenpindar@gmail.com

to find out the encryption algorithm or the key used for it (Dey, 2012; Goyaland Srivastava, 2012). Therefore, it is must for a good encryption method to exclude the repetitive terms such that no trace of repetitions can be tracked down (Deand Maiti, 2013, Dey, 2012b). It is developed to compete with other ciphers of its type by bringing simple and reliable security to the messages that do not require advanced or extremely powerful encryption technique. This stream cipher uses more characters than the usual 26 English letters because now-a-days the messages are far more than just that. All the characters in the ASCII table of characters will be included so as to have all the numbers (0,1,2…. 9) and all the special characters to have more permutations (all possible combinations) as all could be found in a single message (www. asciitable. com). This research will yield a better Caesar cipher as the cipher text will be very difficult to predict as the number of characters are increased and randomized in the process of encryption.

Related Work DEDD

Recently (in 2013) another version of modified shift cipher is released in a paper as the researcher hasquoted "DEDD means Double Encryption and Double Decryption" (De, and Maiti, 2013). In this cryptosystem, the author created two concentric circles representing characters (A……. Z□, □ a……z□, □0…. . 9□) and some special characters ((,),!,@,$,%etc.) as the standard table for the encryption and decryption. The first queue of the circle is used to get cipher 1 and decipher1, whereas the second queue is used to get cipher 2 and decipher 2 by applying the methods.

Figure 1: DEDD Algorithm
Courtesy: De and Maiti (2013)

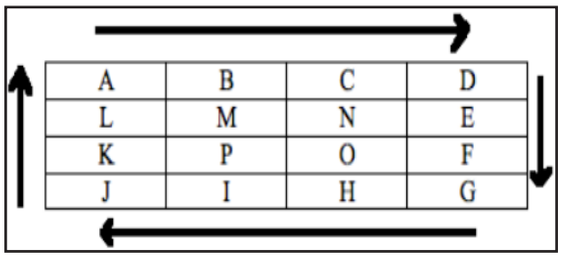


Consider Bob and Alice as a sender and receiver, respectively. Bob generates a key and assigns it to Alice. After receiving the key, Alice counts the length of the message and encrypts the message by shifting each character according to the length of the message. By shifting each character of the string found in the 1st queue up to the length, we get cipher 1. Alice enciphers the message by applying "shift Cipher" and encrypts the message by the length of it and get cipher 1 (for 1st time). Alice encrypts the message twice with the public key, and Bob, too,decrypts that encrypted message twice to recover the original message (De and Maiti (2013).

SD-AREE Cryptographic Method

Similarly, another cipher was created to improve and overcome the earlier version of the Caesar cipher by excluding repetitive terms in a message, when it is to be encrypted. This strengthens the message to be more difficult for a crypto analyst to predict the original message (plain text) from the encrypted. The cipher is called SD-AREE cryptographic method, in which the repetitive bits or characters are moved and there is no trace of any repetition in the message. For example, if a message has two characters of the same type, they become totally different characters in the cipher text. There should be no room for predicting any character or a character next to it. They extract the ASCII value of each character of the text, which is produced after bit level encryption, and add the code with the ASCIIvalue of each character(Dey, 2012; Sharma *et al.,* 2012).

Figure 2: SD-Aree Algorithm
Courtesy: Dey (2012)



The Proposed Frame Work

The proposed enhanced Caesar cipher is to improve the previous versions or to give an option to the users of Caesar cipher. The modified version is very difficult to decrypt without the use of the key. Attackers on this kind of cipher usually use frequency analysis to calculate the probability of having each character at every single location (Babu *et al.,* 2012). In all the previous versions, they used same 26 English alphabets which made it easier to decrypt (Stallings, 2006).

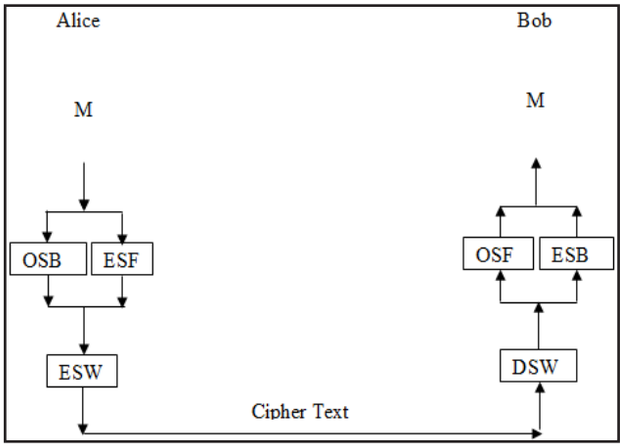
This cipher is going to be built with 95 characters, and not 26. Today's messages include phone and bank account numbers, programming codes or other type of passwords. To strongly cipher these, more than 26 characters are needed. Therefore, there is need to use one of the most accepted computer printable codes which is ASCII printable code or character. Table 1 shows the ASCII printable characters arranged in tabular form, starting from character number 32 to 126 (if we start counting from 0) (De and Maiti, 2013).

Table 1: Printable ASCII Characters.

	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	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	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	{	|	}	~	

Proposed Logical Framework

Figure 3: Proposed New Algorithm



C = M ((OSB, ESF) ESW)

In Figure 3, letter M represent the message before and after the decryption, while OSB and ESF represent Odd-position Shift back and Even-position Shift Forward, respectively. The last acronym at the sender's side, ESW means Encryption Switch and cipher text represents the encrypted message. DSW on the other hand means Decryption Switch. "C" represents the encrypted message (cipher) in this cryptographic scheme.

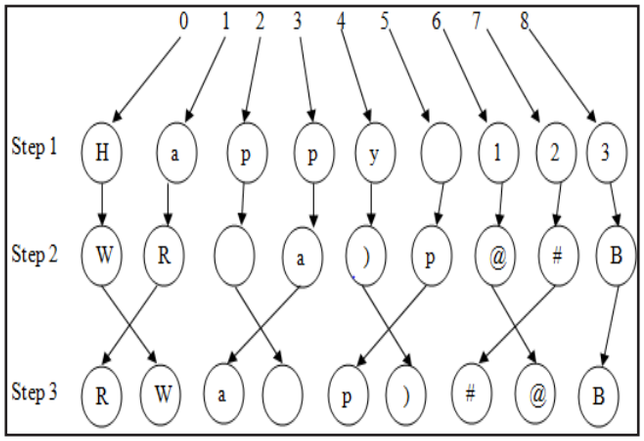
The proposed algorithm works in the following way: when the sender prepares a stream of text message for transmission, the message has to be arranged in an array. Each character has a position usually 0 to n. From 0 to n there are a series of odd as well as even numbers (Babu *et al.*, 2012). The algorithm uses shift technique to shift the character in the odd-position fifteen (15) times or any number of shift as the key given by the sender. The character in the even-position will be shifted fifteen (15) times backward and the ESW function will switch the characters in both odd and even position (Kumar *et al.*, 2012). Table 1 shows the proposed characters in tabular form and the position of each character before the shift operation.

Encryption

Consider "Happy 123" as the message (note that the key used in this diagram is constant)

C = M ((OSB, ESF) ESW)

Figure 4: Encryption Algorithm (with constant key 3)



Steps of the Encryption

Step 1: When the string of ciphers is received, the algorithm will arrange them in a dynamic array. The array usually starts counting the location from 0, 1…n (0,1… until the end of the message). Space and all other special characters will also be included in the array.

Step 2: The algorithm will shift every character in the ciphertext base on the proposed method (odd position characters will be shifted 15 times backward and even positioned characters will be shifted 15 times forward individually). It is not necessarily 15, it is simply the key given by the sender.

Step 3: After obtaining the result from the second step, the algorithm will then switch the position of each character with another character next to it, starting from position 0 of the array until the last character.

Decryption

Figure 5: Decryption Algorithm (with constant key 3)

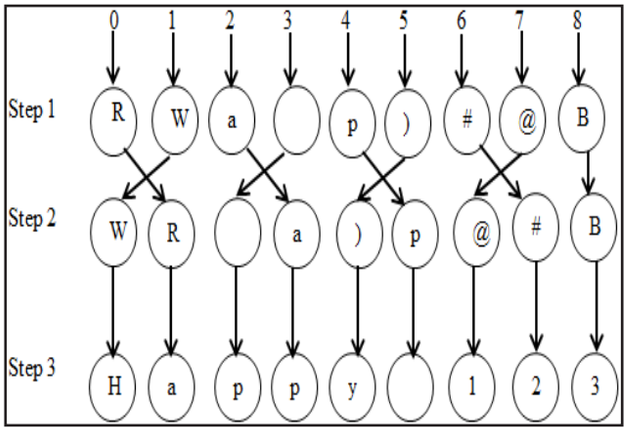


Figure 5 demonstrates the steps of decryption in this cipher scheme.

Steps of Decryption

Step 1: The system receives encrypted message and reverses the decryption process from the last step until the first one. It will first of all switch the position of the cipher base on the agreed key as it is showed in Figure 5.

Step 2: The next step is to reverse the shift other than what was done during the encryption. Shift the odd positioned characters forward and even positioned characters backward to undo (decrypt) the encryption.

Step 3: It is to obtain the result as it was in the original format before the encryption.

Key Scheduling Algorithm (KSA)

1-2 characters of only numbers greater than 3 and less than 50 preferably (3-50). The key to this algorithm has to be selected by Alice and when Alice chooses to use incremented key, then it should be as small as possible (3-20) because the longer the message to be encrypted, the bigger the key will become.

A variable "K" has to be created in the algorithm to store the value given by Alice.

Algorithm Pseudo Code

This section describesthe step-by-step algorithm of the proposed methodology.

//Encryption

{

Step 1: declare the variable to store the characters as p;

Step2 : The input string is provided by Alice.

Step3: Alice input the key of her choice.

Encrypt2() {

Algorithm Encrypt Right shift()

{

Step1: The input string is the input of the function (odd position).

Step2: use loop to access the element in odd position of the inputstring;

Step3: if p== inputstring[i] then transfer the content of input string to another variable

Step4: Then perform the shift operation base on the key value.

}

Algorithm Encrypt Left shift()

{

Step1: The input string is the input of the function (even position).

Step2: use loop to access the element in even position of the inputstring;

Step3: if p== inputstring[i] then transfer the content of input string to another variable

Step4: Then perform the shift operation base on the key value.

Step4: increment the key by 1 (optional)

}

Algorithm switch between even and odd positioned ()

{

Switch elements from even to odd position and vice visa

}

Display the result(){

}

//Decryption

Algorithm switch between even and odd positioned ()

{

Switch elements from even to odd position and vice visa

}

Right shift decryption()

{

Step1: use loop to access the element in the odd position of the variable;

Step2: compare with the content of the master variable declared in the beginning of the program;

Step4: Then perform the reverse shift operation base on the key value.

Step4: decrement the key by 1 (if using the key increment in the encryption)

}

Left shift decryption()

{

Step1: use loop to access the element in the even position of the variable;

Step2: compare with the content of the master variable

Step4: Then perform the reverse shift operation base on the key value.

Step4: decrement the key by 1 (if using the key increment in the encryption)

}

Display the result()

{

}

Result

As shown in the result in Table 2, there are many characters of the same type, but they do not represent the same alphabets. It is very tricky when two or more alphabets are being represented by only one type of character as clearly shown in example 2 where the word "January" is represented by "RYf} cpp", the first "p" and the second "p" represent "r" and "y" respectively, but only one type of character (i. e. p) represented both of them.

User can choose to use either a constant key or an incremented key that will keep on increasing until the plain text ends. Using the incremented key provides more accuracy in terms of repetition but when the message is longer, the key increases.

Table 2: Result of Encryption

Key type	*Message (plaintext)*	*Encrypted (ciphertext)*
	Happy 123	RWa p)#@B
Constant key	January 16, 2012 is the foundation day	RYf}cpp) '@p;!A#@Z/ p#Y$pt'u_%RsZ$_~u/jp
	1111111111	"?!@ A~B}C
Key ++	aaaaaaaaaaaa	RpQqPrOsNtMuv

Comparison

To compare the proposed cipher with the previous ciphers, specifically SD-aree, it looks like there is a repletion detected in the cipher text, even though it is only once, but crypt analyst could use it as a trace to carry out an attack. The more same characters we have in the message, the more chances of getting repetition. There is no possibility where 10 characters of the same type would be found in sequence of the same message.

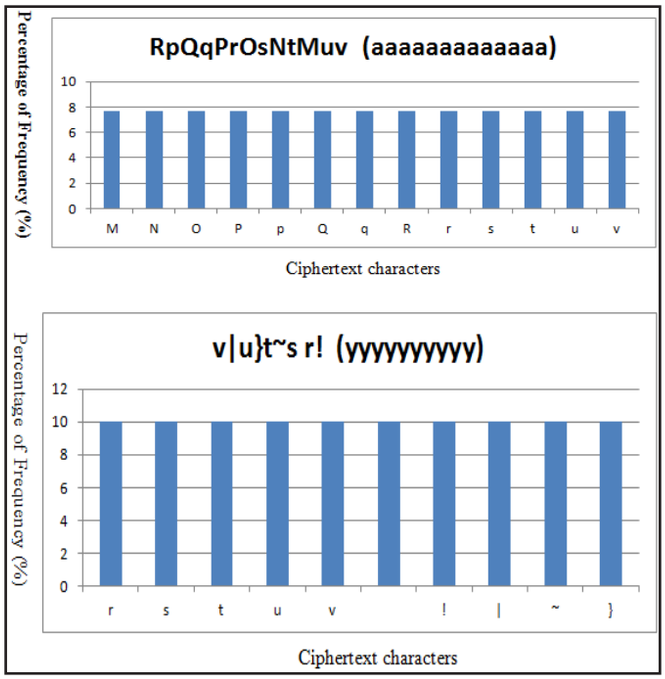
Table 3: Comparisons

Algorithm	**plaintext**	**ciphertext**
	aaaaaaaaaaaa	Óš²©ÛpÙM>□oÄ□i
	aaaaaaaabbbbaaaaaaaa	@X²ÉÛâp[□Ÿ?"□ "([`E□□ð
SD-aree	ÿÿÿÿÿÿÿÿÿÿ	□|*o□,s}~ox
	111aaaaaaaa111	Ù@rÉÛ`ù~o®?„Ô*
	aaaaaaaaaaaa	RpQqPrOsNtMuv
	aaaaaaaabbbbaaaaaaaa	d^e]f[gZiZjYjWkVlU mT
Enhanced	yyyyyyyyyy	v|u}t~s r!
	111aaaaaaaa111	. 4]5[fZgYh)i(:

Analysis

Looking at the obtained result, plain text characters of the same type in the message are being represented by different characters after the encryption. For example, in the cipher text "RpQqPrOsNtMuv",every character text represents "a" (small letter a) and each character appeared only once. The crypto analyst could not find any character of the same type to conduct his analysis. "R" and "r" and "P" and "p" are four (4) different characters. To analyze, there should be more characters of the same type, frequency analysis can only work when the single character is subsequently encrypted with the same key and got the same result as 7. 7% is the frequency of all the characters in this test equally.

Figure 6: Cipher Text Characters



The frequency of all the characters in this test is also equal (10% each) as the previous tests, they all have the same frequency. Usually frequency analysis uses "common pairs" to determine the plain text, but the enhanced Caesar cipher gave no room for any trace as some characters can be represented by a space as displayed in Figure 6.

Each character represent an English alphabet, and even if a character is detected twice or more, they don't represent the same character. It is poly alphabetic cipher where a letter can be mapped to many characters in the same message and that mitigates the treat of using frequency cryptanalysis. There is no trace of any space between the above characters in the cipher which is very difficult or even impossible to determine the beginning and the end of the words.

Conclusion and Future Work

The hybrid of substitution and shift cipher has yielded a stronger stream cipher that is meant to be used with other cryptographic algorithms to strengthen security and also to explore new research possibilities. It is concluded that it is difficult to predict the plain text using frequency analysis from the cipher text when this method is used, because it is poly alphabetic, where a single type of character is represented differently at different occasions in one message. It achieves its repetition exclusion by shifting a stream of message to different direction (b-directional shifting) and substitutes each character with its next at the last step of the encryption. The receiver of the message should reverse the process using the key that was exchanged prior to the communication. Based on the result obtained from the conducted test, it is very difficult for a crypto analyst to predict the plain text of a message if encrypted using this method. It has achieved 100% repletion free based on the test because of the number of characters in the ASCII printable codes which increases the permutation and difficulty level of decipherment. This research can further be modified by randomizing the result of the first and second shifts instead of just switching the position of the cipher text and it can also be integrated with other algorithms.

References

Ascii Characters and Table. Retrieved from http://www.asciitable. com/ (accessed on August 21, 2013).

Babu, R., Abraham, G. & Borasia, K. (2012). A review on securing distributed systems using symmetric key cryptography. *International Journal of Advances in Science and Technology*, April, 4(4), 1-7.

De, P. S. & Maiti, P. (2013). DEDD symmetric-key cryptosystem. *International Journal of Advanced Computer Research*, March, 3(8), 171-176.

Dey, S. (2012a). *An Integrated Symmetric Key Cryptographic Method- Amalgamation of TTJSA Algorithm, Advanced Caesar Cipher Algorithm, Bit Rotation and Reversal Method: SJA Algorithm*. India, West Bengal: Department of Computer Science, St. Xavier's College.

Dey, S. (2012b). *SD-AREE: A New Modified Caesar Cipher Cryptographic Method Along with Bit-*

Manipulation to Exclude Repetition from a Message to be Encrypted. India, West Bengal: Department of Computer Science St. Xavier's College.

Goyal, D. & Srivastava, V. (2012). RDA Algortihm: Symmetric Key Algorithm. *International Journal of Information and Communication Technology Research*, 2(4), 342-347.

Kumar, A., Varshney, A. K. & Kumar, P. (2012). PA substitution Cipher. *International Journal of Engineering Research & Technology*, December, 1(10), 1-4.

Sharma, A., Bhatnagar, A., Tak, N., Sharma, A., Avasthi, J. & Sharma, P. (2012). An approach of substitution method based on ascii codes in encryption technique. *International Journal of Advanced Studies in Computers, Science & Engineering*, 1(3), 1-7.

Stallings, W. (2006). Cryptography and Network Security 4/E., Pearson Education India. Retrieved from http://www. pearsonhighered. com/educator/ product/Cryptography-and-Network- (accessed on September 1, 2013)