International Transactions on Electrical Engineering and Computer Science
Salah et.al., Vol: 4, No: 2, pp: 53-63, June 2025

Enhancement of Caesar Cipher Algorithm using Four Keys

Salah Abdulghani Alabady[1], Thabita Fawaz Shawkat[1], Amina Waad Adrees[1]

Abstract: There are serious worries about possible cybersecurity risks and the necessity for reliable solutions as the Internet of Things (IoT) becomes more integrated into our daily lives. IoT security is crucial to preserving the privacy and accuracy of user data. Nowadays, cryptography is a key component of information security systems' defense against hostile attacks. The Caesar cipher is one of the most popular encryption-decryption methods. The encrypted text is evolved by the use of a substitution method. This type of replacement cipher uses a key, which is a numerical value, to split the letters and replaces each letter in the plaintext with a letter at a predetermined moment. The goal of this research is to create a technique with a higher level of security than the original algorithm. It suggests altering the encryption algorithm for the Caesar cipher, which would complicate the procedure. The Caesar cipher has been improved by using extended ASCII code instead of ASCII code, in addition to adding alphabets, integers, and symbols. Additionally, we use four different keys (Key1, Key2, Key3, and Key4) to illustrate the efficacy of the proposed method. The length of the primary key is 128 bits. It has been split into four halves, or four keys, each of which has a key length of 32 bits. The encryption strength of the modified Caesar cipher was assessed using the Brute Force attack, which was then compared to the Caesar method. The results showed that the Caesar method was broken and that the value of the key used in the encryption process was discovered, despite the fact that the proposed algorithm was unknown and revealed the value of the encryption key.

Keywords: Caesar Cipher, Network Security, Cybersecurity, Brute Force, Encryption

1. Introduction

The Internet of Things IoT is used in many different fields, including smart homes, government work, senior care, environmental protection, intelligent transportation, and personal health [1].

History
Received: 08-11-2024; Revised: 18-03-2025;
Accepted: 27-03-2025

Salah Abdulghani Alabady
eng.salah@uomosul.edu.iq

[1]Computer Engineering Department, College of Engineering, University of Mosul, Mosul – 41002, Iraq

All kinds of electronic objects and gadgets that can connect to the Internet and exchange data are referred to as Internet of Things (IoT) devices. The process of sharing data between smart devices may damage the privacy of persons and their personal information. The sent data may be faked, intercepted, altered, or interrupted by an attacker. IoT data flows via several network hops, so it's necessary to use a reliable encryption method to preserve data safety. In order to stop hackers from listening in on conversations and changing data as it is being transmitted, encryption is the process of ciphering data so that only those with permission may access it. One piece of unencrypted or unlocked data will be transparent, making it simple for hackers to exploit it [1]. Moreover, a tiny encryption key used by some IoT devices may leave them open to hacking [2-3].

https://doi.org/10.62760/iteecs.4.2.2025.121

While there are many methods for encryption and decryption, they can be broadly divided into two main categories. There are two types of cryptography: conventional and public. Conventional encryption uses a single key (symmetric key) for both encryption and decryption, while public key cryptography uses distinct keys (asymmetric key) [4-5]. One of the simplest instances of a classical substitution cipher is the Caesar cipher, which substitutes a letter three paces ahead of the alphabet for the letter in the alphabet. Because there are only 25 potential key combinations in the end, the Caesar cipher is readily cracked using a brute force approach [6].

This paper's suggested algorithm is considered as a block cipher, meaning it processes input in numerous blocks, each of which has a size of 32 bits (4 bytes), and it uses extended ASCII code rather than regular ASCII code. In the suggested algorithm, four different keys (Key1, Key2, Key3, and Key4) were also employed. The primary key is 128 bits long. It has been divided into four sections (four keys), each with a 32-bit key length. This increases the complexity of the Cesar cipher method and the difficulty of deciphering or estimating the ciphertext.

This research is organized into five sections. Section 1 provides an introduction to the research topic, highlighting its significance, objectives, and scope. Section 2 presents a comprehensive review of related works, offering an overview of previous studies and identifying gaps this research addresses. Section 3 details the methodology and design and implementation of proposed algorithm to achieve the research objectives. Section 4 discusses the results and findings, analyzing their implications in light of the research objectives. Finally, Section 5 concludes the study by summarizing the key outcomes, providing recommendations.

2. Related works

A way to enhance the Caesar cipher using a random number generation mechanism for key generation operations has been developed by Saudagar et al. [7]. The Caesar cipher has been extended to incorporate symbols and alphanumeric characters. The Caesar cipher of old was limited to alphabets. Caesar Substitution's key was generated with a key Matrix Trace value limited to modulo 94. A recursive random number generation equation, the result of which only depends on the value of the chosen seed, is used to create the matrix elements.

The author attempted to adapt classical cipher features to current ones. Columnar transposition with arbitrary random order column selection has been used for the second stage of encryption. Therefore, the suggested Scheme combines elements of both contemporary and classical cipher properties. The suggested approach uses the least amount of memory while offering significant security and excellent throughput. With a 93! combination of keys, the approach is impervious to brute-force attacks for Caesar.

The classic Caesar cipher has been modified by Goyal et al. [8], who maintain the key size at one. He checks the alphabet's index during substitution; if it is even, he increases the key value by one; if the index is odd, he decreases the key value by one. By combining strategies from contemporary ciphers, such as XOR key to the first letter of the plain text, XOR first letter of the plain text to the second letter, and so forth, Sinaga et al. [9] have proposed a modified hybrid of the Caesar cipher and Vigenere cipher to increase the diffusion and confusion characteristics of cipher text.

An algorithm to enhance the classical encryption technique was proposed by Atish Jain et al. [10]. The suggested modified algorithm makes use of affine ciphers, transposition ciphers, and randomized substitution techniques to produce a cipher text that is almost impossible to decode. Together with alphabets, it also expands the set of characters that the Caesar cipher algorithm can encrypt to include all ASCII and expanded ASCII characters. For increased security, a sophisticated key-generation method that creates two keys from one key is utilized.

A technique for data encryption and decryption was suggested by the authors in [11]. The approach is based on the ASCII values of the characters in the plain text. The ASCII values of the data to be encrypted are employed in this procedure to encrypt it. The data will be encrypted or decrypted using a key that is obtained by altering another string with the secret. As a result, this approach can be classified as symmetric encryption since it slightly modifies the same key that is used for

both encryption and decryption. When the length of the key and the input are same, the algorithm functions.

A bit-by-bit encryption scheme based on message position has been presented by Disina [12]. By moving the characters in the even position to the right and the characters in the odd position to the left, the sender will transpose the message's bits. According to Purnama et al. [13], there are two methods for turning data into an unreadable format: 1. The transposition method 2. The technique of substitution one illustration of a substitution technique is the Caesar cipher. Caesar cipher has several drawbacks.

A viewpoint on the integration of substitution and transposition techniques is presented in this study. To get around all of the Caesar cipher's drawbacks and create a far stronger and more secure cipher, utilize the double columnar transposition approach. According to Kothari et al. [14], if a cipher is computationally secure, it implies that there is a very slim, but not impossible, chance of breaking the encryption key using existing computational technology and techniques in a reasonable amount of time.

All cryptographic algorithms, with the exception of the Vernam Cipher, are theoretically breakable given sufficient time and cipher text. Here's where COMCRYPT enters the scene. An encryption technique called COMCRYPT was developed based on the Vernam cipher. A scrambling technique is applied to a passphrase once it is obtained from the user, producing two more random keys. To create the cipher text, these keys are layered on top of each other and XORed to the text. Upon monitoring this technique on several plaintexts, it was discovered to be nearly impenetrable. Multiple encryptions and decryptions are supported by this technology. The cipher text will change significantly with even a slight alteration to the text key.

Mishra [7] uses techniques such as encryption with different keys at each level, encryption with the same key at each level, and multiple-row transposition ciphers to increase security. The solution suggested by Greetta Pinheiro and Shruti Saraf [15] offers increased security and can defend the data against brute force assaults by employing extra encryption layers. Together with a key that is likewise a number, the cipher text that is produced during the various encryption phases is represented by numbers. It is unlikely that one could deduce the plaintext from those numbers.

3. Design and implementation of proposed algorithm

Cryptography provides a number of security purposes, including data privacy, non-alteration, and other issues. The following is a list of the many goals of cryptography [16].

- ***Confidentiality*:** This is the safeguard against information being revealed without authorization. It is possible to apply confidentiality to entire communications, portions of messages, or even the fact that messages exist [17]. Data that has been sent is shielded from passive attacks by confidentiality.
- ***Authentication*:** Ensuring the authenticity of a communication is the responsibility of the authentication service. It is the confirmation of a message's purported source. There are two kinds of authentication: data origin and peer entity.
- ***Data Integrity*:** This relates to the integrity of a single message, a stream of messages, or certain fields within a message. It guarantees that there are no replays, duplications, insertions, modifications, or reordering of the messages once they are transmitted. Another aspect of this service is data destruction.
- ***Access Control*:** refers to the capacity to restrict and manage user access through communications links to host systems and applications. To do this, in order to customize access rights for each entity attempting to obtain access, the entity must first be identified, or authenticated.
- ***No Repudiation*:** A communication that has been transmitted cannot be denied by the sender or the recipient. The recipient of a message has the ability to demonstrate that the purported sender actually transmitted the message [16].

3.1 Caesar Cipher

The Caesar cipher is a well-known encryption-decryption technique in the field of cryptography. Julius Caesar is credited with using the Caesar cipher,

a kind of substitution type, to communicate with his soldiers. Caesar is credited with being one of the first people to use encryption to protect communications. Caesar communicated his decision to all the generals and was able to send them encrypted messages after determining that his normal algorithm would be to move each letter three positions down the alphabet in the message. Modular arithmetic can be used to represent the encryption by first changing the letters into integers, as in $A = 0$, $B = 1$,..., $Z = 25$. There is a mathematical explanation for how a shift N can encrypt a letter x, as (1)

$$(x) = (x + n) \, mod \, 26 \qquad (1)$$

Decryption is performed similarly as (2),

$$Dn\,(x) = (x - n) \, mod \, 26 \qquad (2)$$

For example, with a shift of the 3, A would be replaced by the letter D, B would be replaced by the letter E, and so on as shown in Fig. 1.

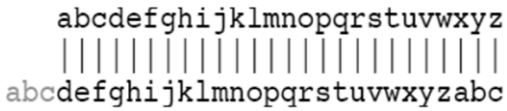

Fig. 1: Caesar Cipher [18]

The 'key' to the cipher must first be obtained by both parties in order for the sender to encrypt data using it and the recipient to decrypt it using the same key. This allows for the transfer of encrypted messages from one person to another as shown in Fig 2. The only thing needed to crack the Caesar cipher is the quantity of letters needed to change the cipher alphabet. Here is an illustration of how the Caesar cipher functions. The value 3 key is used for this. The original alphabet is displayed in the first row, and each original alphabet that will be substituted is displayed in the second row as in Table. 1.

After that, the algorithm can be written like this. For every letter p in the plaintext, replace it with the ciphertext letter (3).

$$C = E(3, p) = (p + 3) \, mod \, 26 \qquad (3)$$

Any number of a shift is allowed, consequently the standard Caesar algorithm is (4)

$$C = E(k, p) = (p + k) \, mod \, 26 \qquad (4)$$

where the value of k falls between 1 and 26. The decryption technique is easy to use as in (5).

$$p = D(k, C) = (C - k) mod \, 26 \qquad (5)$$

For example, the message to be encrypted is MEET ME AFTER TOGA PARTY. So, the ciphertext in this example is Phhwphdiwhuwrjdsduwb [19], is presented in Table. 2.

Table. 1: Original and substituted alphabets when 3 key is used

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
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

Table. 2: Original and substituted alphabets with proposed algorithm

M	E	E	T		M	E		A	F	T	E	R		T	O	G	A		P	A	R	T	Y		
P	H	H	W		P	H		D	U	W	H	U		W	R	J	D		S	D	U	W	B		

The cipher alphabet will obviously be moved differently if a different key is employed. As can be observed, the algorithm's simplicity leaves it opens to attacks because of repetitions, which make it simple to decrypt the encrypted data. The Caesar cipher has certain flaws that allow us to employ a brute force assault.

- The encryption and decryption algorithm are known.
- Only 25 keys are trying.

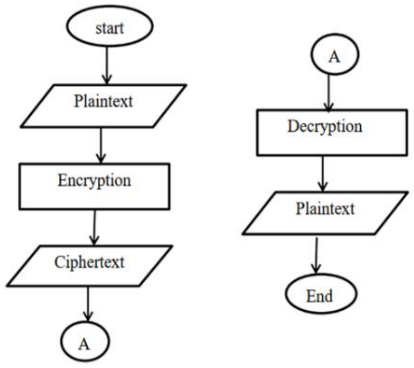

Fig. 2: Caesar cipher flowchart

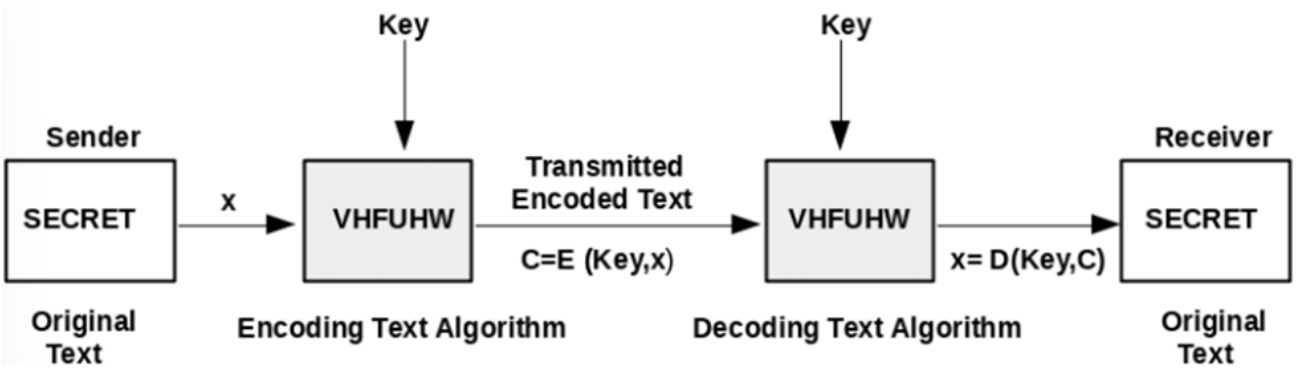

Fig. 3: Caesar Cipher block diagram

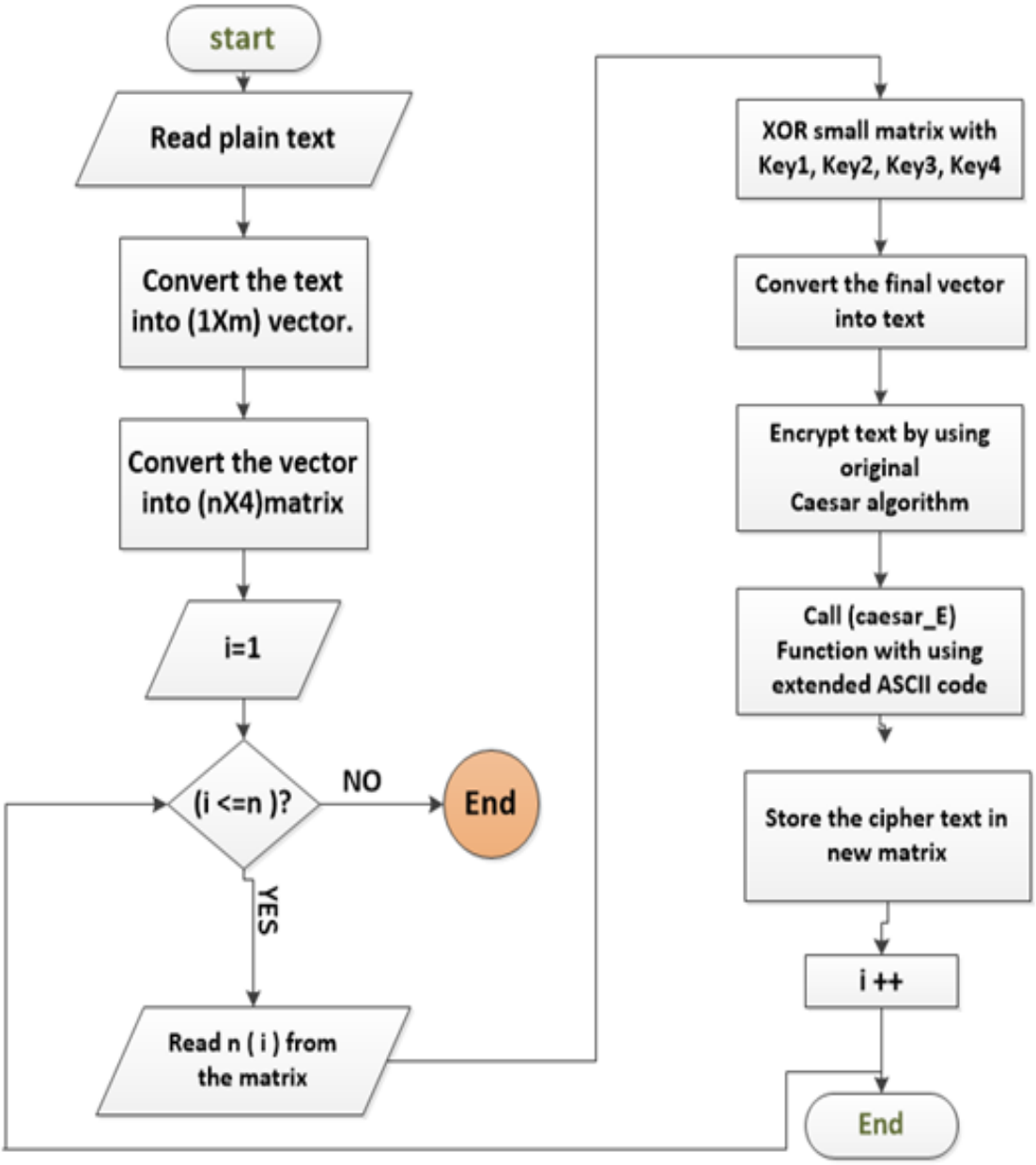

Fig. 4: Flowchart of encryption steps for improved Caesar algorithm

- The language of the plaintext is known and easily recognizable

Fig. 3, [20] shows Caesar's cipher block diagram.

3.2 Design and Implementation of Proposed Algorithm

The steps suggested for the Caesar cipher's enhancement are explained in this section. As a block cipher, the suggested method processes input in blocks, with each block having a size of 32 bits (4 bytes). Fig. 4 shows the flowchart of improved the Caesar algorithm, and the details of the steps of design and implementation used for the improvement:

- Read the current input text.
- Save and convert the input text into a vector.
- Convert the vector into a large (n × 4) matrix.
- Perform an XOR operation (⊕) on each row of the large matrix with the first key (32 bits).
- Repeat an XOR operation (⊕) on the same row with the (second, third, and fourth) key, the length of each key is 32 bits. This gives the impression that a key with 128 lengths was used.
- Convert the final vector into text.
- Perform the original Caesar algorithm on the text to be encrypted.
- Using Extended ASCII code instead of the general ASCII code that was used in the original Caesar algorithm. Fig. 7, [21] shows the General ASCII Code, and Fig. 8, [22] shows the Extended ASCII Code.
- Store the ciphertext in a large final matrix.
- Repeat steps 3-7 on all blocks until the entire input text is finished.

The following Fig. 5 and Fig. 6 describe the encryption and decryption processes using 4 keys.

4. Simulation results of the Caesar and proposed algorithms

In this section we present two scenarios of the simulation results for the text encryption. The first scenario is evolution the original Caesar algorithm when using the range of keys from 1 to 26 as shown in Table. 3, and when using the range of keys from 27 to 100 as shown in Table. 4.

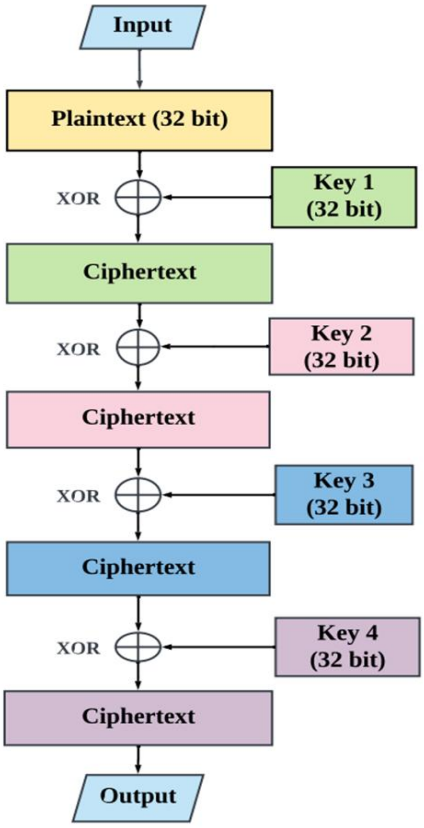

Fig. 5: Flowchart of encryption using 4 Keys

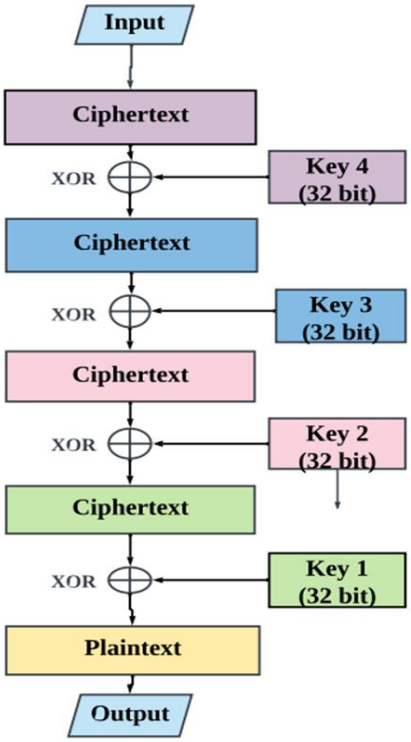

Fig. 6: Flowchart of decryption using 4 Keys

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	|
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

Fig. 7: General ASCII code

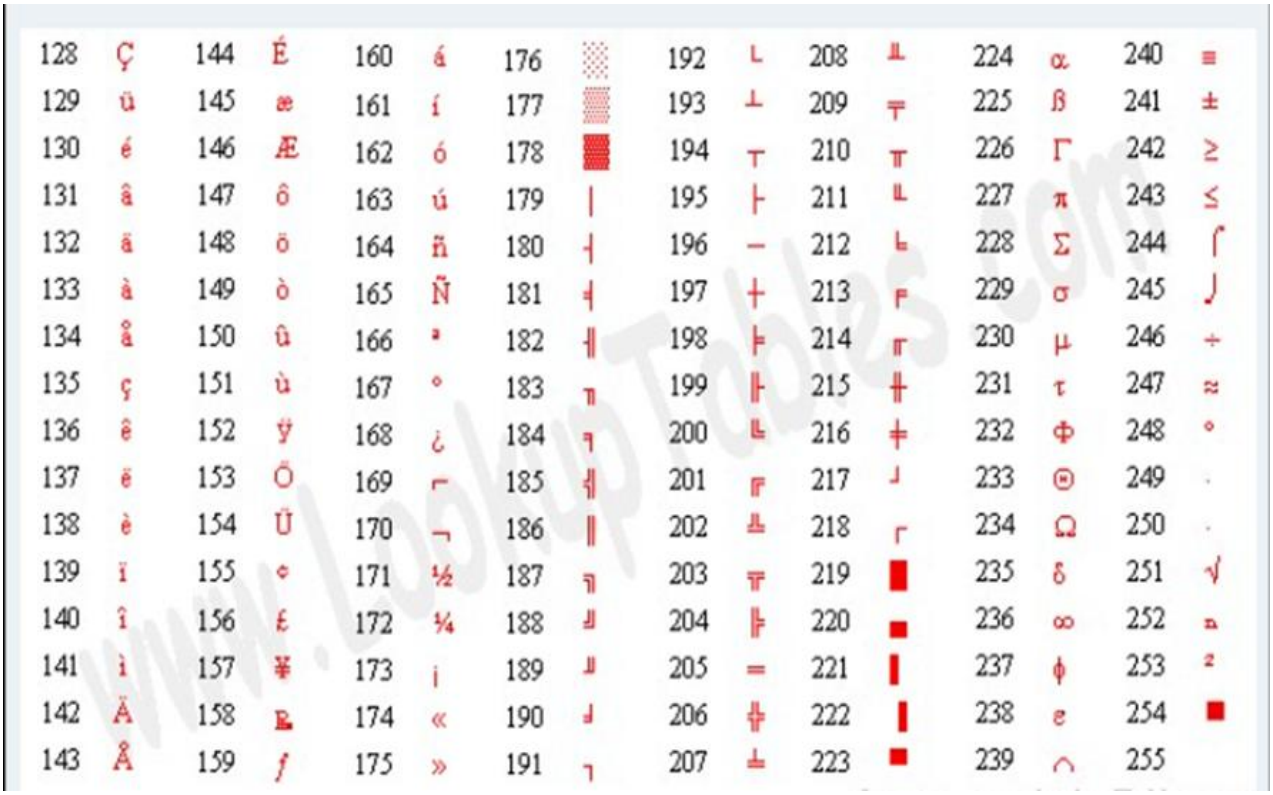

Fig. 8: Extended ASCII code

The simulation results demonstrate that the strength and complexity of encryption increases with the increase in the value of the key. In addition, the process revealing or guessing the ciphertext becomes more difficult and complicated as well as the key that used for encryption process.

In the second scenario we present the performance of the proposed algorithm using four different keys (Key1, Key2, Key3, and Key4). The length of the main key is 128 bits. It has been segmented into four parts (four keys), with a key length of 32 bits. Table. 5 shows the comparison of encryption between Caesar algorithm and proposed algorithm using Key1 and Key2, and Table. 6, shows the comparison of encryption between Caesar algorithm and proposed algorithm using Key3 and Key4. The MATLAB simulation program is used for the implementation and evaluation the proposed algorithm. The simulation results prove that the strength and complexity of encryption increases with the increase of the value of the key. The results indicate that the proposed algorithm is better than the Caesar algorithm (using different types of key) in terms of the strength of data encryption and in terms of the difficulty of being able to guess or decipher the code.

Table. 3: Encryption of caesar algorithm based on ASCII code in case the rang of keys from 1 to 26

Key	Input plain text: Computer engineering department 023
1	Dpnqvufs Fohjoffsjoh Efqbsunfou 134
2	Eqorwvgt Gpikpggtkpi Fgrctvogpv 245
3	Frpsxwhu Hqjlqhhulqj Ghsduwphqw 356
4	Gsqtyxiv Irkmriivmrk Hitevxqirx 467
5	Htruzyjw Jslnsjjwnsl Ijufwyrjsy 578
6	Iusvazkx Ktmotkkxotm Jkvgxzsktz 689
7	Jvtwbaly Lunpullypun Klwhyatlua 79:
8	Kwuxcbmz Mvoqvmmzqvo Lmxizbumvb 8:;
9	Lxvydcna Nwprwnnarwp Mnyjacvnwc 9;<
10	Mywzedob Oxqsxoobsxq Nozkbdwoxd :<=
11	Nzxafepc Pyrtyppctyr Opalcexpye ;=>
12	Oaybgfqd Qzsuzqqduzs Pqbmdfyqzf <>?
13	Pbzchgre Ratvarrevat Qrcnegzrag =?@
14	Qcadihsf Sbuwbssfwbu Rsdofhasbh >@A
15	Rdbejitg Tcvxcttgxcv Stepgibtci ?AB
16	Secfkjuh Udwyduuhydw Tufqhjcudj @BC
17	Tfdglkvi Vexzevvizex Uvgrikdvek ACD
18	Ugehmlwj Wfyafwwjafy Vwhsjlewfl BDE
19	Vhfinmxk Xgzbgxxkbgz Wxitkmfxgm CEF
20	Wigjonyl Yhachyylcha Xyjulngyhn DFG
21	Xjhkpozm Zibdizzmdib Yzkvmohzio EGH
22	YkilqpaT [jcejaanejc Zalwnpiajp FHI
23	ZljmrqbU \kdfkbbofQd [bmxoqjbkq GIJ
24	Amknsrcp]leglccpglK \cnyprkclr HJK
25	Bnlotsdq ^mfhmddqhmL]dozqsldms IKL
26	Computer _ngineerinM ^epartment JLM

Table. 4: Encryption of caesar algorithm based on extended ASCII code in case the rang of keys from 27 to 100

Key	Input plain text: Computer engineering department 023
27	Dpnqvufs `ohjoffsjoN _fqbsunfou KMN
28	Eqorwvgt apikpggtkpO `grctvogpv LNO
29	Frpsxwhu bqjlqhhulqj ahsduwphqw MOP
30	asqtyxiv crkmriivmrk bitevxqirx NPQ
31	btruzyjw dslnsjjwnsl cjufwyrjsy OQR
32	cusv{zkx etmotkkxotm dkvgxzsktz PRS
33	dvtw|{ly funpullypun elwhy{tlu{ QST
34	ewux}|mz gvoqvmmzqvo fmxiz|umv| RTU
35	fxvy~}n{ hwprwnn{rwp gnyj{}vnw} SUV
36	gywz☻~o| ixqsxoo|sxq hozk|~wox~ TVW
37	hzx{☻p} jyrtypp}tyr ip{l}☻xpy☻ UWX
38	i{y|☻q~ kzsuzqq~uzs jq|m~yqzVXY
39	j|z}☻r☻ l{tv{rr☻v{t kr}n☻☻zr{☻ WYZ
40	k}{~s m|uw|ssw|u ls~o{sbXZ[
41	l~|☻t☻ n}vx}tt☻x}v mt☻p☻|tci Y[\
42	m☻}u o~wy~uuy~w nuq}udj Z\]
43	n~☻v p☻xz☻vvz☻x ov☻r~vek []^
44	o☻☻wqy{ww{y pws☻wfl \^_
45	px r☻z|☻xx|☻z qxtxgm]_`
46	q☻ys{}yy}{ ryu☻yhn ^`a
47	rzt|~zz~| szvzi_ab
48	s{u}☻{{☻} t{w{j`bc
49	t|v~|\~ u|x| acd
50	u☻}w☻☻}}☻☻ v}y} bde
55	z☻☻ b☻a☻~☻ gij
60	e☻g☻☻☻f☻☻ lno
65	jl☻☻k qst
70	o¡ qp☻ vxy
75	t ¡¦¥£ v£u¡£¥¥ acd
80	y¥£¦«ª¨ {¤☻¤¨¤☻ z¦¨ª£¤ª fhi
85	~ª¨«°¯ - ©¢¤© -¤©¢ ☻ «-¯¨ ©¯ kmn
90	¯-°µ´¥² ®§©®¥¥²©®§ ¥°¡²´-¥®´ prs
95	´²µº¹ª. ³¬®³ªª.®³¬ ªµ¦.¹²ª³¹ uwx
100	¹·º¿¾¯¼ ☻¸±³¸¯¼³¸± ¯º«¼¾·¯¸¾ z| }

Table. 5: Comparison of encryption between Caesar algorithm and proposed algorithm using Key1 and Key2

Key	Input Text: Computer engineering department 023		
	Caesar algorithm	**Proposed algorithm using Key1**	**Proposed algorithm using Key2**
3	Frpsxwhu Hqjlqhhulqj Ghsduwphqw 356	u`aGGIYIw_W`bZYI[_WsWEUIHbY]H	FhHj-qhlj=hdwhwP5Z
6	Iusvazkx Ktmotkkxotm Jkvgxzsktz 689	xcdJJL\LzbZce]\L^bZvZHXLKe\`	I"sa Km!tkom@k gzkzS8]
9	Lxvydcna Nwprwnnarwp Mnyjacvnwc 9;<	{fgMMO_O}e]fh`_Oae]y]K[ONh_cN	L%vdn N"p$wnr-pCn jcncV;`
12	Oaybgfqd Qzsuzqqduzs Pqbmdfyqzf <>?	~ijPPRbRh`ikcbRdh`\`N^RQkbfQ	O(ygqQ%s'zqu!sFqmf-q fY>c
15	Rdbejitg Tcvxcttgxcv Stepgibtci?AB	lmSSUeUkclnfeUgkc©cQaUTneiT	R+bjtT(v*ctx$vItpO!t#i\Af
18	Ugehmlwj Wfyafwwjafy Vwhsjlewfl BDE	opVVXhXnfoqihXjnffTdXWqhlW	U.emwW+y-f"w a'yLwsR$w&l_Di
21	Xjhkpozm Zibdizzmdib Yzkvmohzio EGH	rsYY[k[qirtlk[mqiiWg[ZtkoZ	X1hpzZ.b0i%z#d*bOz-vo'z)obGl
24	Amknsrcp]leglccpglK \cnyprkclr HJK	uv\\^n^tluwon^ptllZj^]wnr]	A4k-scC1e3l(c&g-eRc!y"r*c,reJo

Table. 6: Comparison of encryption between Caesar algorithm and proposed algorithm using Key3 and Key4

Key	Input Text: Computer engineering department 023		
	Caesar Algorithm	**Proposed Algorithm using Key3**	**Proposed Algorithm using Key4**
3	Frpsxwhu Hqjlqhhulqj Ghsduwphqw 356	aEG Yb V^)U9 b/y_;T>Hb.^'ei	Fp tgLn"k ip"Elieq-mP5X
6	Iusvazkx Ktmotkkxotm Jkvgxzsktz 689	J4HJ=\<e Ya,X< e2bb>WAKe1a*hl	I"swjOq%n#ls%Hol ht!pS8[
9	Lxvydcna Nwprwnnarwp Mnyjacvnwc 9;<	M71M&_? h \d/[? h5eeAZDNh4d-k o	L%vzmR-t(q&ov(Krokw$s-V;^
12	Oaybgfqd Qzsuzqqduzs Pqbmdfyqzf <>?	j:4P)bBQ _g2^B k8hhD]GQk7g0nr	O(ycpU!w+"t)ry+Nurnz'v!Y>a
15	Rdbejitg Tcvxcttgxcv Stepgibtci ?AB	m=QSFeET bj5aE n;kkG`JTn:j3qu	R+bf sX$z.w,u b.Qxuqc*y$\Ad
18	Ugehmlwj Wfyafwwjafy Vwhsjlewfl BDE	p@TVIhHq em8dH q>nnJcMWq=m6tx	U.e i#v['c1(`/x#e1Ta xtf-b'_Dg
21	Xjhkpozm Zibdizzmdib Yzkvmohzio EGH	sCWYLkKt hp;gK tAqqMfPZt@p9wa	X1hl&y^*f4+c2a&h4Wd#awi0e*bGj
24	Amknsrcp]leglccpglK \cnyprkclr HJK	\FZ\OnNw Qs>jN wDttPiS]wCs<zd	A4k-o)|a-i7.f5d)k7Zg&dzl3h-eJm

5. Conclusion

In this paper, improvements of the Caesar cipher encryption algorithm have been proposed, these improvements would make the Cesar cipher algorithm more complex, and it aims to build the method with a security level that is either higher than the original algorithm's security level. In addition, the Caesar cipher r have been modified and expanded so as to include alphabets, numbers and symbols and at the same time by using extended ASCII code instead of only the ASCII code. In summary modified of Caesar cipher algorithm was tested data security on communicated message and it was found to be most secure compared to data security test on Caesar cipher algorithm. The MATLAB software was used to encrypt texts, as well as to determine the encrypting and decryption.

Conflict of interest

The authors declared 'No conflict of interest'.

References

[1] M. Albany, E. Alsahafi, I. Alruwili, and S. Elkhediri "A review: Secure Internet of thing System for Smart Houses", *Procedia Computer Science*, Vol. 201, pp. 437-444, 2022. https://doi.org/10.1016/j.procs.2022.03.057

[2] T. Abdullah, W. Ali, S. Malebary, and A. A. Ahmed "A review of cyber security challenges

attacks and solutions for Internet of Things based smart home," *International Journal of Computer Science and Network Security*, Vol. 19, No. 9, pp. 139-146, 2019. [CrossRef]

[3] M. Algarni, M. Alkhelaiwi, and A. Karrar, "Internet of things security: A review of enabled application challenges and solutions," *International Journal of Advanced Computer Science and Applications*, Vol. 12, No. 3, pp. 201-215, 2021. [CrossRef]

[4] S. Chandra, S. Bhattacharyya, S. Paira and S. S. Alam, "A study and analysis on symmetric cryptography," *2014 International Conference on Science Engineering and Management Research (ICSEMR)*, Chennai, India, pp. 1-8, 2014. https://doi.org/10.1109/ICSEMR.2014.7043664

[5] G. Sharma and A. Kakkar, "Cryptography Algorithms and approaches used for data security", *International Journal of Scientific & Engineering Research*, Vol. 3, No. 6, pp. 1-6, 2012. [CrossRef]

[6] Y. J. Lee, K. R. Park, S. J. Lee, K. Bae and J. Kim, "A New Method for Generating an Invariant Iris Private Key Based on the Fuzzy Vault System," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 38, No. 5, pp. 1302-1313, Oct. 2008, https://doi.org/10.1109/TSMCB.2008.927261

[7] S. Saudagar, N. Kamtalwar, H. Karadbhajne, M. Karmarkar, H. Kendre and O. Ketkar, "File Encryption-Decryption using Java," *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, pp. 855-859, 2023. https://doi.org/10.1109/IDCIoT56793.2023.10053514

[8] K. Goyal and S. Kinger, "Modified caesar cipher for better security enhancement", *International Journal of Computer Applications*, Vol. 73, No. 3, pp. 0975-8887, 2013. [CrossRef]

[9] M. D. Sinaga, N. S. B. Sembiring, F. Tambunan and C. J. M. Sianturi, "Hybrid Cryptography WAKE (Word Auto Key Encryption) and Binary Caesar Cipher Method for Data Security," *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, Parapat, Indonesia, 2018, pp. 1-5, 2018. https://doi.org/10.1109/CITSM.2018.8674346

[10] A. Jain, R. Dedhia, and A. Patil, "Enhancing the security of caesar cipher substitution method using a randomized approach for more secure communication," *arXiv preprint arXiv:1512.05483*, 2015. https://doi.org/10.48550/arXiv.1512.05483

[11] D. Buddh and V. K. Srivastav, "Hybrid three layer Fibonacci Caesar Cipher," *2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC)*, Bengaluru, India, pp. 1567-1573, 2022. https://doi.org/10.1109/IIHC55949.2022.10059848

[12] A. H. Disina "Robust Caesar Cipher against frequency cryptanalysis using bi-directional shifting", *Universiti Tun Hussein Onn Malaysia*, 2014. [CrossRef]

[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, pp. 195-204, 2015. https://doi.org/10.1016/j.procs.2015.07.552

[14] M. Kothari, M. Shah, M. Malde, and A. Wad, "Comcrypt: An Encryption Algorithm based on Vernam cipher", *International Journal on Computer Science and Technology (IJCST)*, Vol. 3, No. 4, pp. 364-367, 2012. [CrossRef]

[15] G. Pinheiro and S. Saraf, "Improved Caesar cipher algorithm using multistage encryption", *IJCSIT) International Journal of Computer Science and Information Technologies, IJCST*, Vol. 7, No. 1, pp. 117-119, 2016. [CrossRef]

[16] A. Albugmi, M. O. Alassafi, R. Walters and G. Wills "Data security in cloud computing", *2016 Fifth International Conference on Future Generation Communication Technologies (FGCT)*, London, UK, pp. 55-59, 2016. https://doi.org/10.1109/FGCT.2016.7605062

[17] M. Cardei, A. Marcus, I. Cardei and T. Tavtilov, "Web-based heterogeneous WSN integration using pervasive communication", *30th IEEE International Performance Computing and Communications Conference*, Orlando, FL, USA, pp. 1-6, 2011. https://doi.org/10.1109/PCCC.2011.6108065

[18] S. N. Gowda, "Innovative enhancement of the Caesar cipher algorithm for cryptography," *2016 2nd International Conference on Advances in*

Computing, Communication, & Automation (ICACCA) (Fall), Bareilly, India, pp. 1-4, 2016. https://doi.org/10.1109/ICACCAF.2016.7749010

[19] M. Srivastava, U. Srivastava and S. Srivastava, "Modified Caesar Cipher with Image Steganography," *2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India, pp. 1-6, 2023. https://doi.org/10.1109/ISCON57294.2023.10111954

[20] A. Saraswat, C. Khatri, P. Thakral, and P. Biswas. "An extended hybridization of vigenére and caesar cipher techniques for secure communication", *Procedia Computer Science*, Vol. 92, pp. 355-360, 2016. https://doi.org/10.1016/j.procs.2016.07.390

[21] Z. West, *ASCII Table*, Available at https://www.alpharithms.com/ascii-table-512119/

[22] *Extended ASCII Code* Available at https://www.asciitable.com/2018

***