Detection Optimization of Brute-Force Cyberattack Using Modified Caesar Cipher Algorithm Based on Binary Codes (MCBC)

Muhannad Tahboush[1]*, Adel Hamdan[2], Mohammad Klaib[3], Mohammad Adawy[4], Firas Alzobi[5]
Information Systems and Network Department, The World Islamic Sciences and Education University, Amman, Jordan[1, 4, 5]
Computer Science Department, The World Islamic Sciences and Education University, Amman, Jordan[2]
Intelligent Systems Engineering Department, Middle East University, Amman, Jordan[3]

***Abstract*—Information security is considered vital aspects that are employed to protect user credentials and digital information from cyber security threats. A Caesar cipher is an ancient cryptography algorithm, and it is susceptible to being easily broken and vulnerable to brute-force attack. Brute-force attack is a cyberattack that uses trial and error to crack passwords, login credentials, and encryption keys to unauthorized access and illegal to a system and individual accounts. However, several research has been developed to defeat the existing vulnerabilities in Caesar cipher, but are still suffering from their limitations and failing to provide a high level of attack detection and encryption strength. Therefore, Modified Caesar Cipher Algorithm Based on Binary Codes (MCBC) has been proposed to mitigate brute-force attack more optimistically based on different scenarios. First scenario, converting message to binary numbering system and the second scenario, employ binary shifting technique and then convert it to hexadecimal code. The performance metrics that were taken into consideration to evaluate the MCBC proposed algorithm are detection rate, strength rate, true positive rate and time required for decryption. The experimental results show that the proposed approach MCBC performance metrics outperformed other algorithms against brute force attack by ensuring the confidentiality of information.**

***Keywords*—Brute-force attack; encryption; Caesar cipher; binary code; security**

I. Introduction

Cybersecurity issues are become increasingly important, due to the increasing volume of sensitive data and credentials targeted by cybercriminals. Thus, it has become an urgent need to find a security system that can maintain confidentiality and prevent data from being misused, changed, or compromised by third party. Counterfeit authentication schemes allow attackers to use tactics such as social engineering and brute force attacks to obtain user database login information [1][2]. Therefore, cryptography can be employed to secure communication by encryption data on the sending side and decryption process on the receiving side of the communication system [2].

Encryption algorithms are usually used in addition to protecting data from theft, burglary or even alteration to verify the user's identity. Some of these algorithms are based on character representative which consist of substitution ciphers to convert one letter in the plaintext into an alternative form called cipher text [2][3] this type of substitution called Caesar cipher. Ideally only authorized parties can decrypt the cipher text and get access to the original information. Symmetric cryptography is a method that uses the same key for the encryption and decryption process [4]. The advantages of symmetric key are that managing the key is much easier and faster than the public key method. The Caesar cipher is considered as the most widely used symmetric encryption technique as illustrated in Fig. 1.

Fig. 1. Symmetric cryptography.

In cryptography techniques, Caesar cipher is a part of substitution cipher and susceptible to being easily cracked through brute-force cryptanalysis in a short period of time [4][5]. The reason behind this, is that there are only 25 possible options of keys are available [6]. Caesar encryption algorithm will replace each plaintext letter with a different one in a fixed number of positions [7]. The alphabet used to create the plaintext is assigned an index number that is used as keys, as shown in Fig. 2.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Fig. 2. Alphabetical order index.

Brute-force attacks are very challenging in detection and considered as high-risk security threats in cyberattacks. Brute force attack occurs when the adversary uses trial and error methods to crack passwords, login credentials, and encryption keys [8]. However, cryptography algorithms could transmit sensitive information over an insecure network to prevent the

data from being read by unauthorized recipients other than the intended recipient [9]. There are several issues that need to be resolved through MCBC proposed modified algorithm such as: easy to decrypt data by an unauthorized user and by looking at the letters pattern, the entire message can be decrypted, also provide higher attack detection rate and encryption strength. Moreover, the main limitation in Caesar cipher is the limited key space, which contains only 25 possible keys. This makes it easy for an attacker to systematically check brute force attacks and try all possible keys and passphrases till they find the right one [10]. Therefore, this paper presents an algorithm based on binary numbering system and shifting technique to provide high level of encryption and overcome the limitations faced by classical Caesar Cipher.

In this research, we perform the binary numbering system and shifting technique to strengthen the Caesar algorithm and increase the effectiveness of the MCBC. The outcomes of this research demonstrate the significant impact on password cracking techniques using brute force attack. The difference between MCBC algorithms and other algorithms in the literature is that the proposed algorithm used a binary system (base-2 and base 16) that will perform some other operations such as encryption and decryption. Doing so will help to protect data, storage and achieve high performance of encryption. Thus, the contributions of this paper are summarized as follows:

1) We proposed a modified MCBC secure algorithm for Caesar cipher. MCBC can provide encryption strength and is considered more secure and resistant to brute force attacks through performing the binary numbering system and shifting technique.

2) The concept of binary shifting technique and hexadecimal conversion will improve the performance and accuracy of MCBC which will avoid the chances of decryption operation by the attacker making the system strength against brute force attack.

3) The hexadecimal code will be input into Caesar cipher algorithm for complex processes in decryption operations.

4) The MCBC algorithm has been compared with that of classical Caesar cipher algorithm averse to brute force attack. The results demonstrate that the MCBC algorithm outperforms classical Caesar cryptography algorithm.

The remainder of this paper is organized as follows. Section I provide the introduction. Section II about literature review. Section III about preliminaries and background. Section IV shows the proposed approach. Section V shows the security analysis. Section VI shows results comparison and evaluation. Section VII about research summary. Finally, Section VIII, concludes the paper.

II. Literature Review

Several algorithms and myriad solutions have been developed to overcome the limitation of Caesar cipher encryption. However, the literature will discuss and point out the most recent developed solutions in cryptographic algorithms of the relevant literature reviewed.

M. D. Hossain et al. in [11] providing brute force attacks detection. This detection of SSH and FTP brute force attacks by employing LSTM (Long Short-Term Memory) deep learning technology. In addition, the detection mechanism used machine learning classifiers such as J48, naive Bayes, decision table, random forest, and k-nearest neighbors to enhance our detection capabilities and CICIDS2017 dataset. The evaluation of LSTM and ML algorithms has been shown that the LSTM model outperforms ML algorithms in terms of performance, achieving an accuracy level.

E. Ahmadzadeh et al. in [12] proposed a modified hybrid technique consisting of Caesar cipher and Vigenère cipher as well. The modification will improve the diffusion and confusion properties of the cipher text by incorporating modern encryption techniques such as XORing the key to the first letter of the plaintext, and then to the second letter and so.

M. M. Najafabadi et al. in [13] proposed mechanism detection about SSH brute force attacks at the network level, which can be detected through analyzing Net Flow data. A dataset has been employed for attack detection, using (ML) machine learning techniques that have been shown to be effective in recognizing brute force attacks. The proposed method authors have distributed SSH brute force attacks and evaluated, they conclude that some methods for detecting individual attacks were shown to have difficulties in implementation, as indicated by (AUC) Area Under the Receiver Operating Characteristic Curve values.

M. Srivastava et al. in [14] propose a modification that consists of two various encryption methods. Firstly, employ Caesar cipher techniques include image steganography. The image is first encoded and then stored inside the available image in order to increase the level of security. Secondly, a third security level will be involved. The encrypted image of the message is associated by the sender with a security key that can contain n digits. The receiver also receives the key with the image and if it matches the sender's key, then the image is decrypted.

Q. A. Kester in study [15] proposed an algorithm that uses a Vigenere square and a key in the encryption process. However, the new method uses successive keys that depend on the value of the initial key during the encryption process. The keys used later are based on the value of the original key during the encryption process. The key for the first stage is different from the key for the second stage, but they are related to each other, with the key for the second stage being derived from the function used in the first stage, and so on. The algorithm ultimately allows the text to be encrypted and decrypted and makes it more difficult to defend against common attacks with the Vigenère cipher. This is due to the different keys used in each encryption process.

D. Veera et al. in study [16] proposed a new technique which make the encryption more efficient based on a combination of the modified Caesar cipher and the Card Deck Shuffle algorithm for encryption operation of the image. The Card Deck Shuffle algorithm will reconstruct all available pixels based on the outcomes of the modified Caesar algorithm. The method uses variable keys, therefore, to have successful brute-force attack, it

requires more than 2^{26} attacks. The method can be used in various multimedia applications.

III. PRELIMINARIES AND BACKGROUNDS

In this section, we will characterize the preliminaries that are required in this research that are necessary for successful achievement of this research.

A. Adversary Model

The network is initiated in an environment with antagonistic activities, where opponents are present. We assume that the attackers can guess the username and password to gain unauthorized access to the system. Additionally, some attackers can also be used to discover applications and scripts as brute force tools to bypass authentication processes [8]. The adversary can access the web application by searching for the corresponding session ID. This gives the adversary the opportunity to control resources, steal information and infect websites with malware, resulting in disruption of available services.

B. Cryptography

Cryptography is a method to secure information and communication by ensuring integrity and confidentiality using codes in the presence of adversarial behavior. The privacy of individuals and organizations is guaranteed by a high level of cryptography to be sure the information that has been transmitted is accessible by authorized users only [17][18]. Therefore, the most common use of cryptography would be using it to transmit data through an insecure channel.

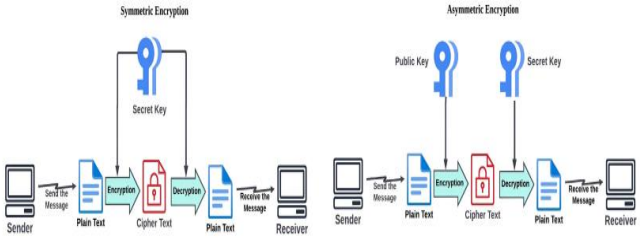

Fig. 3. Symmetric and asymmetric encryption [17].

Fig. 3 shows the cryptographic methods that can be categorized into two types: symmetric and asymmetric key cryptography. Symmetric key cryptography is a technique that uses the identical key for both the encryption and decryption process, such as Caesar cipher and XOR encryption techniques. While Asymmetric cryptography is employed a couple of different keys, one for encryption process and another for decryption process but mathematically related to each other [17][19].

C. Caesar Cipher

Caesar's encryption algorithm is one of the early and famous cryptographic algorithms realized, which uses 25 letters of the alphabet for encryption. In this type of algorithm, the given text is replaced by a letter with a fixed number of positions. In other words, it works by taking a message (plaintext) and substituting each letter in plaintext with another letter in the alphabet (cipher text). Consider Fig. 4 below, if we assume that the position shift value is 3, thus A will be replaced by the letter D and B will be replaced by the letter E and so on [5][20].

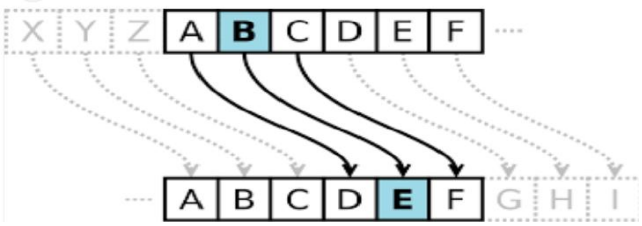

Fig. 4. Symmetric and asymmetric encryption [5].

Therefore, to be able to process with cipher a specific text, you need a shift value that indicates how many positions each letter in the text has been shifted or moved. The shift can be any number, a shift of 0 will not be considered as a shift at all, because all the alphabetic letters will remain in their position. If the alphabet of the plaintext is 26, then a shift of 26 also will not be considered as a shift at all since the cipher text would be the same as the plaintext. The first step is to convert the alphabet to numeric alphabets, where A is zero, B is one, and finally 25 is equal to Z [21]. Caesar's encryption mathematically expressed as illustrated in Eq. (1).

$$\text{Ciphertext} = (\text{Plaintext} + \text{Key}) \bmod 26 \quad (1)$$

While Eq. (2), expressed the mathematical form the decryption process of the cipher text using Caesar cipher encryption as follows:

$$\text{Plaintext} = (\text{Ciphertext} - \text{Key}) \bmod 26 \quad (2)$$

Where the (key) indicates the shift value that has been applied during the encryption and decryption process.

However, with the use of several decryption methods, Caesar cipher became vulnerable to easily cracked in a second, even in a scenario where only cipher text is used. To decrypt ciphered text using Caesar cipher, you need to move it backward by a certain number of positions depending on the key used to encrypt it [12]. However, there are only 25 possible shifts, so one way to break the code is by brute force until a solution is found [5] [10]. Namely, one can simply try all possible shifts.

D. Brute Force Attack

Brute force password attack is the most common network attack that relies heavily on raw computing power rather than the intelligence of the attacker. In a brute force attack, the attacker exploits the vulnerabilities of the credentials of a victim and checks all possible passwords and phrases with the hope of guessing and discovering them correctly [22][23]. Brute force attacks can be categorized into various types, credential stuffing and reverse brute force attacks. Generally, Brute-force attacks are considered more effective when weak or relatively predictable passwords are used. Brute force attack is considered as a type of cyberattack that use trial and error method because of a large record of usernames and passwords to gain unauthorized access to the available resource [22][24] as shown in Fig. 5.

Fig. 5. Brute-force attack [24].

This type of attack needs to check whether the credentials are authenticated and depending on the response of the application or whether the credentials were right or wrong. If not, the attackers will try another credential combination until they get unauthorized access to the system [25][26] to achieve their goals. A successful brute force attack can lead to several impacts on the resources and systems such as data breaches, leaking hidden files or interfaces and disrupting the service if it service is attacked to the point of causing a denial of service (DoS) [25].

E. Description of Binary Shifting

The methodology of this research relied on binary shifting (moving bits one position), because binary shifting technique can be used to enhance the Caesar cipher. Binary shifting technique related to the case of taking any binary number to the left or the right, according to the systematic method which will prevent its real contents from appearing to attackers as shown in Fig. 6 [27][28].

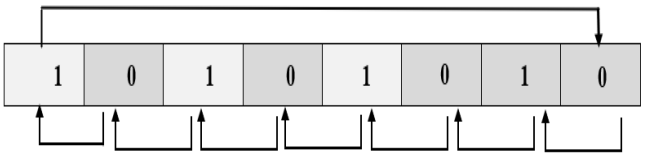

Fig. 6. Binary shifting algorithm.

The binary shifting can be used on a selected set of variables, where binary number (bits) shifting conceals the identity of sensitive binary code, thus preventing direct inference attacks. The binary shifting technique mathematically represented as indicated in Eq. (3) below.

v_i : is the bit value (0 or 1) in the i^{th} place

The function $f(i) = v_i$, $i = 1,2,\cdots,8$ gives one byte filled as shown below:

1	2	3	4	5	6	7	8
$f(1) = v_1$	$f(2) = v_2$	$f(3) = v_3$	$f(4) = v_4$	$f(5) = v_5$	$f(6) = v_6$	$f(7) = v_7$	$f(8) = v_8$

Now define the shifting function $g(i)$ as

$$g(i) = \begin{cases} f(1) \ , & i = 8 \\ f(i+1) \ , & i = 1,2,,\cdots,7 \end{cases} \quad (3)$$

The function $g(i)$ gives a new byte filled as shown below:

1	2	3	4	5	6	7	8
$g(1) =$	$g(2) =$	$g(3) =$	$g(4) =$	$g(5) =$	$g(6) =$	$g(7) =$	$g(8) =$
$f(2) = v_2$	$f(3) = v_3$	$f(4) = v_4$	$f(5) = v_5$	$f(6) = v_6$	$f(7) = v_7$	$f(8) = v_8$	$f(1) = v_1$

IV. PROPOSED APPROACH

The proposed algorithm is based on modifying the Caesar cipher algorithm and using a binary shifting technique between all available binary numbers (bits) after converting the unencrypted text to binary numbers. A successfully binary shifting (moving one position) technique has been employed to avert the decryption of the message, discontinue guessing credentials correctly through brute force attack and finally increase the complexity of the MCBC proposed algorithm against the adversary. Furthermore, the proposed algorithm will be able to resolve the security drawback in Caesar cipher algorithm and it would be difficult to perform brute force cryptanalysis. The proposed algorithm steps are as follow:

Step 1: Employ a binary numbering system technique to convert the message into a certain number of even bits.

Step 2: After that, use binary shifting technique to change the position of the available bits between one another in the converted message.

Step 3: Then, convert the shifted binary numbers to hexadecimal numbers to be processed to the Caesar cipher algorithm, so that it is not clear to the adversary.

Step 4: Finally, employ Caesar algorithm with certain shift key to encrypt the message and prevent trial and error methods to crack passwords and login credentials.

A. Assumptions:

In this section, some assumptions about the network and the capabilities of the adversaries in the proposed design are presented as follows.

Assumption1: An even number of bits should be resulted after converting message into binary code.

Assumption 2: The adversary can launch many kinds of brute force attacks.

Assumption 3: The algorithm proposed that the targeted password or key is susceptible enough to be unveiled through a trial-and-error approach.

Assumption 4: The adversary may exploit vulnerabilities present in the authentication process of the system being targeted.

B. Modified Encryption Technique

One of the simplest encryption techniques that are used to protect information and communication systems over insecure channels is the processes of encryption information using Caesar cipher. Generally, Caesar cipher is increasingly susceptible to various types of attack and security threats, where adversaries are capable of decrypting an encrypted message in a short period of time and guessing login credentials through brute-force attack. Therefore, a modified Caesar cipher technique has been employed to overcome the vulnerability of Caesar cipher and threats against brute force attack.

When a source is willing to transmit encrypted message. The proposed algorithm (MCBC) will convert the plaintext into binary numbering system (bits) using decimal code of character from ASCII table, for instance a letter of ***(and)*** will be converted into binary code as shown in Fig. 7, where the even number of bits is involved.

0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
0	1	1	0	0	1	0	0								

Fig. 7. Converting letter to binary (3 byte).

Then, the use of binary shifting technique falls in the idealist place, which is the backbone of proposed algorithm. The binary shifting required to move between all available binary code (bits) one position to the left or the right in every separated single byte. The shifting process starts by changing the position of the first bit to be in the next position and so on, the last bit will be in the first position in each byte, as illustrated below in (Algorithm 1).

Algorithm 1: Binary Shifting Technique

Input: $i_1, i_2, \ldots\ldots, i_8$
Output: Every single bit will be shifted to one position
Start
Input: arr[]
Begin
Set Length of Binary values
Length [arr] = 8
Create a new empty array newArr[] of size 8
newArr[7] = arr[0]
For (***i***=6; ***i*** >= 0; i--)
newArr[i] = arr[i+1]
output: newArr[]
End
Continue till End of binary number in each Byte
Display Output Shifted Values …….
End
End of Pseudocode

After binary shifting processes for every single byte, the result will appear as in Fig. 8.

1	1	0	0	0	0	1	0	1	1	0	1	1	1	0	0
1	1	0	0	1	0	0	0								

Fig. 8. Shifted binary system (3 byte).

The number of binary codes will always be even. Therefore, every bit was replaced by the position of other bit in the binary system. After that, it becomes important to convert the available shifted binary code into hexadecimal number as shown in (Algorithm 2) to result with (c2dcc8).

Algorithm 2: Convert binary to hexadecimal

Input: Enter Binary code (Figure 7)
Output: hexadecimal number to be processed with Caesar Cipher Algorithm
Start
While Length (Binarycode_N) MOD 8≠ 0 **Do**
Binarycode_N ← "0" + binarycode_N.
End while
Loop {
Binarycode_8 bit ← Substring (Binarycode_N)
Loop {
Binarycode_4bit ← Substring (Binarycode_8 bit)
HexChar_4bit ← BinaryToHexMAP (Binarycode_4 bit)
HexChar_8bit ← HexChar_8bit + HexChar_4bit
End Loop
Hexadecimal_N ← Hexadecimal_N + HexChar_8bit
End Loop
Combine the Hexadecimal result of all groups to get the complete output
End of Pseudocode

Subsequently, the operation processed into Caesar cipher algorithm, where each converted letter/number in the plaintext is replaced by a letter with some fixed number of positions in the alphabet.

C. Input Caesar Cipher Algorithm

To illustrate this last phase of the proposed algorithm, it's important to identify the converted hexadecimal code resulted from (Algorithm 2). Firstly, when starting using Caesar cipher to encrypt data, it's important to determine the shift key and start replacing (shifting) each letter of the message in the "plaintext" line and write down the corresponding letter in the "cipher text" line. This process can be achieved through mathematical expressions of the encryption process that has been used as in Eq. (1), and Eq. (2) for the decryption process to retrieve the message back to its original form.

Secondly, it's important to make a table where the top row contains original hexadecimal code resulted from (Algorithm 2), and the bottom row is for the new shifted alphabet according to the selected shift key.

Third, an encoded message will be obtained with the equivalent shifted letter, here assume shift key is (2) for 6 groups of 4 bits each, as shown in Fig. 9.

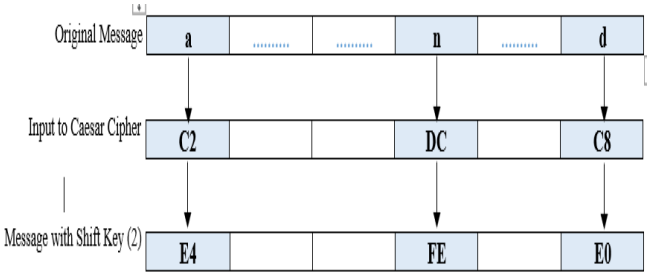

Fig. 9. Encrypted text using MCBC algorithm.

Finally, to decrypt a message encoded with a Caesar cipher, the recipient should know the number of binary codes used in (Algorithm 1), shifted binary technique and the hexadecimal number with shift key, then processes with the encoded message to return it back to its original form. To evaluate the results using both algorithm Caesar cipher and MCBC proposed algorithm with the same input text (and) and to demonstrate the effectiveness of the proposed algorithm over original Caesar cipher. Fig. 10 shows the encryption operations of both algorithms using same shift key value (2).

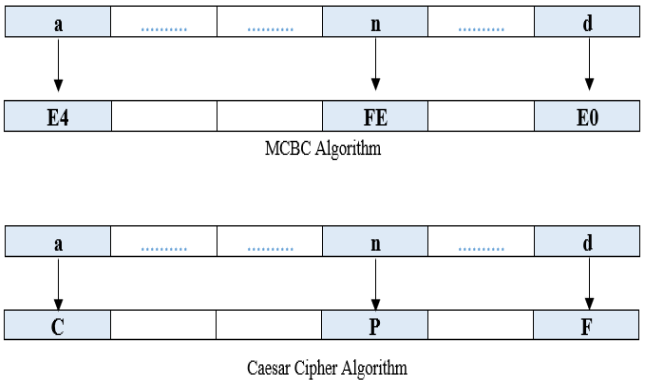

Fig. 10. MCBC and Caesar cipher results.

Based on the available results, the encrypted message using the MCBC proposed algorithm will be unreadable and un-understandable by malicious entities and brute-force attack while excessively forceful attempts to gain access to user accounts. Therefore, the MCBC algorithm has proven its efficiency over Caesar cipher and brute force cryptanalysis will not be easily performed.

V. Security Analysis of the Proposed Algorithm

Adversaries are more likely to camouflage malicious and aggressive behavior as if it were normal by evade detection, where attackers can temporarily stop submitting data or guessing credentials once a detection event is observed. The attack can also be executed when the attacker realizes that the network is using a Caesar cipher as a form of protection. Therefore, MCBC will overcome the security weakness that allows the attacker to submit and guess many passwords of the victim through converting text to binary codes as shown in first phase. In the second phase, binary shifting techniques have been used to prevent the malicious actor discovering and understand the mechanism that was employed. And being unable to understand the transmitted original message through the process of converting binary shifted code to hexadecimal. In this section, we analyzed the security of MCBC algorithm under presented attack.

VI. Result Comparison and Evaluation

To have a comprehensive evaluation of the proposed algorithm against brute-force attack effect, the performance of MCBC algorithm has been simulated using MATLAB R2015a environment. The performance parameters required to evaluate and measure the proposed algorithms are detection rate, true positive rate, accuracy, strength rate, time required for decryption. To evaluate the efficiency of the MCBC algorithm, we compare its performance with the well-known detection algorithm in the event of a brute-force attack.

A. Detection Rate

Detection rate is the ratio of the number of detected malicious activities to the total number of actual malicious activities, as shown in Eq. (4).

$$DR = \frac{TPR}{TPR+FNR} \times 100 \quad (4)$$

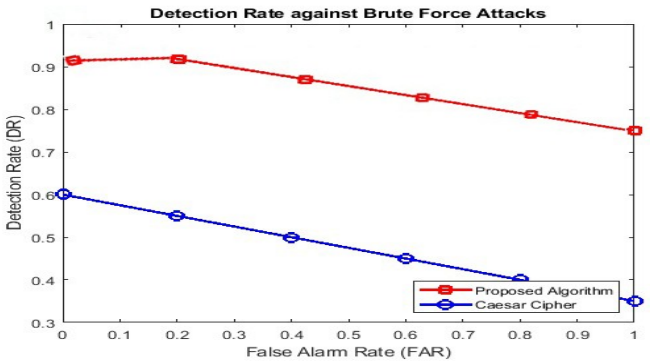

Fig. 11. Relation between detection rate and false alarm rate.

Data in Fig. 11 shows the evaluation of MCBC algorithm that has been performed and represents the trade-off between attack detection rate and false alarm rate. The MCBC provides the maximum detection rate (0.92) compared with the Caesar Cipher and decreases slightly while increasing FAR. This decrease is due to the false positives and increase in delays while processing the encryption and decryption process. On the other hand, traditional Caesar cipher provides DR (0.6) when the false alarm rate is approximately null and decreases to reach (0.3) while increasing FAR. Therefore, it demonstrates the ability of the proposed algorithm has a promising and optimistic detection rate compared with Caesar cipher.

B. Strength Rate

The strength rate of the algorithm can be measured by the amount of time required and computational effort needed to break the encryption algorithm over. This plot illustrates the strength of these algorithms against time, providing a visual representation of their security efficacy over extended periods.

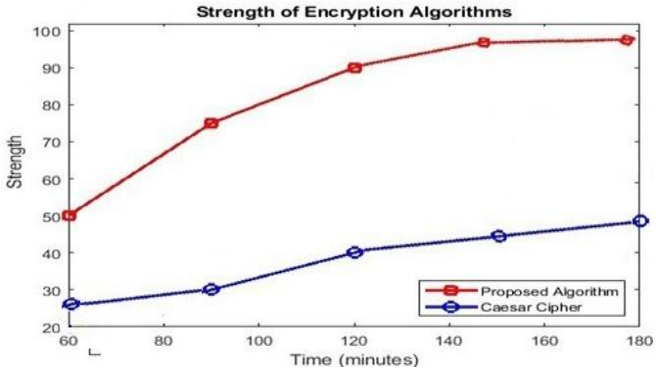

Fig. 12. Strength rate comparison.

Data in Fig. 12 shows the performance analysis and evaluation rate of the MCBC against Caesar algorithm. The encryption strength of the Caesar Cipher increases over time to reach approximately (48%) maximum strength rate, while in MCBC it rises in strength rate to reach approximately (95%). The reason behind that, the MCBC provides binary code conversions, hexadecimal number and binary shifting technique which will strengthen the proposed encryption algorithm, whereas the Caesar cipher is based on substitution method that leads to have lower strength encryption algorithm, which reduces the strength against brute-force attacks. Therefore, the

performance analysis and evaluation rate of the proposed algorithm outperformed the Caesar Cipher algorithm.

C. True Positive Rate (TPR)

TPR is the rate at which true attacks are identified correctly and measure of encryption algorithms to identify the brute-force threats, as shown in Eq. (5).

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

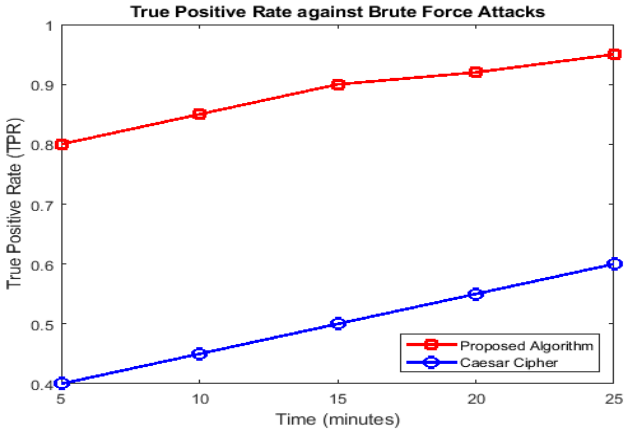

Fig. 13. True Positive Rate.

Data in Fig. 13 shows the comparative analysis between MCBC and the Caesar Cipher presented through a graph plotting their TPR against time. The MCBC shows a gradual increase over time to reach approximately (0.93). This indicates that the algorithm's detection mechanisms allow it to maintain a high level of sensitivity in identifying brute force attacks. Whereas the Caesar Cipher's shows a low TPR compared with MCBC algorithm to reach maximum (0.57) which considers as lack of detection mechanisms, due to the substitution method used by the Caesar cipher, which creates predictable encryption patterns that can be easily exploited by attackers.

D. Time Required for Decryption

The time required to decrypt encrypted data using brute force attacks is a fundamental measure of an encryption algorithm strength and resilience that mainly based on computational complexities.

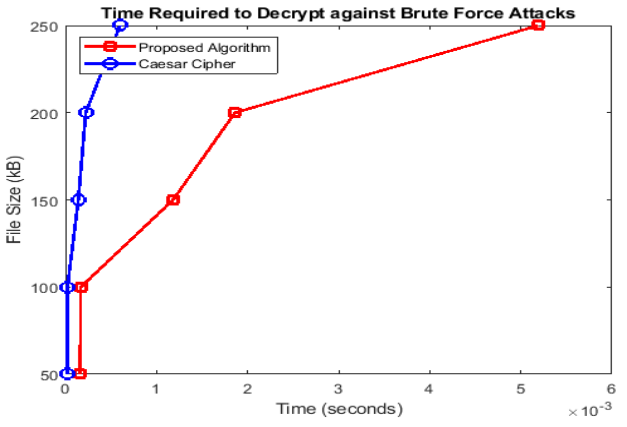

Fig. 14. Decryption time against Brute-force attack.

Data in Fig. 14 shows the time required to decrypt encrypted data against file size using brute force attacks. The decryption time of the MCBC shows a steep increase as file size grows, making brute force attacks impractical. The reason behind that is that, the binary code, hexadecimal number and binary shifting techniques increase the complexity of the algorithm. Conversely, the Caesar Cipher's relatively flat line decryption time curve, indicating minimal increases in decryption time as file size grows. This focuses on the cipher's inherent weaknesses and its vulnerability to rapid brute force attacks. Finally, the proposed algorithm reflects the effectiveness in resisting such attacks.

VII. Summary

In this part, it is important to present the functionality and performance of MCBC in the analyzed environment. In the study, the MCBC algorithm will be compared with Caesar cipher algorithm when exposed to attack instances. The experimental outcomes can be concluded as follows:

- The MCBC algorithm provides a higher strength rate which is approximately 95% compared with Caesar cipher algorithms that have lower strength that reach 48%.
- The MCBC maximizes decryption time, making brute force attacks impractical due to the computational complexities.
- The proposed algorithm provides optimal value of TPR approximately 0.93 in comparison with Caesar cipher algorithms. Thus, it has a high level of sensitivity in identifying brute force attacks and the ability to detect the real attackers.

VIII. Conclusion and Future Work

This research examined the adversary effect of brute-force attack which considered as serious threats to cybersecurity and obstacles to ensuring credential protection. Modified Caesar Cipher Algorithm Based on Binary Codes (MCBC) has been employed based on on two various scenarios, firstly, binary codes will convert the message into binary codes (bits) and second scenario uses binary shifting mechanisem to change the position of the available bits among each other in the message to bolster encryption against brute force attacks. MCBC is considered as suitable for the evaluation of brute-force attack and provide accurate detection and high strength rate that reduce bruken of the proposed algorithm. However, the proposed MCBC algorithm generally outperformed the Caesar cipher algorithms. It is of the utmost that in the future, we will focus on using other approaches that provide greater flixability and more accurate detection performance in networks that are based on different features.

References

[1] S. S G, "Improved Caesar Cipher with Random Number Generation Technique and Multistage Encryption," Int. J. Cryptogr. Inf. Secur., vol. 2, no. 4, pp. 39–49, 2012, doi: 10.5121/ijcis.2012.2405.

[2] M. Victor, D. D. W. Praveenraj, R. Sasirekha, A. Alkhayyat, and A. Shakhzoda, "Cryptography: Advances in Secure Communication and Data Protection," E3S Web Conf., vol. 399, 2023, doi: 10.1051/e3sconf/202339907010.

[3] S. Kulkarni, “Cryptographic algorithm using data structure using C concepts for better security,” 2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015, vol. 00, no. c, pp. 15–17, 2015, doi: 10.1109/PERVASIVE.2015.7087028.

[4] S. N. Gowda, “Innovative enhancement of the Caesar cipher algorithm for cryptography,” Proc. - 2016 Int. Conf. Adv. Comput. Commun. Autom. (Fall), ICACCA 2016, 2016, doi: 10.1109/ICACCAF.2016.7749010.

[5] Fibriyanto Farrel, “Decrypting an Unknown Caesar Cipher Using Brute Force," Institut Teknologi Bandung,” 2022.

[6] R. Devi.T, “Importance of cryptography in network security,” Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013, pp. 462–467, 2013, doi: 10.1109/CSNT.2013.102.

[7] S. Kumar, M. S. Gaur, P. Sagar Sharma, and D. Munjal, “A Novel Approach of Symmetric Key Cryptography,” Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021, vol. 26, no. 2, pp. 593–598, 2021, doi: 10.1109/ICIEM51511.2021.9445343.

[8] S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, “SSH-Brute Force Attack Detection Model based on Deep Learning,” Int. J. Comput. Appl. Technol. Res., vol. 10, no. 01, pp. 42–50, 2021, doi: 10.7753/ijcatr1001.1008.

[9] J. Sasi, K. M. Anusha, A. Vijaykumar, and M. Kavya, “Cryptography: The Sciene of Secure Communication,” IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 16, no. 4, pp. 129–134, 2016.

[10] I. M. Keshta, “Caesar Cipher Method Design and Implementation Based on Java , C ++ , and Python Languages,” vol. 16, no. 4, pp. 298–307, 2018.

[11] M. D. Hossain, H. Ochiai, F. Doudou, and Y. Kadobayashi, “SSH and FTP brute-force attacks detection in computer networks: Lstm and machine learning approaches,” 2020 5th Int. Conf. Comput. Commun. Syst. ICCCS 2020, pp. 491–497, 2020, doi: 10.1109/ICCCS49078.2020.9118459.

[12] E. Ahmadzadeh, H. Kim, O. Jeong, and I. Moon, “A Novel Dynamic Attack on Classical Ciphers Using an Attention-Based LSTM Encoder-Decoder Model,” IEEE Access, vol. 9, pp. 60960–60970, 2021, doi: 10.1109/ACCESS.2021.3074268.

[13] M. M. Najafabadi, T. M. Khoshgoftaar, C. Kemp, N. Seliya, and R. Zuech, “Machine learning for detecting brute force attacks at the network level,” Proc. - IEEE 14th Int. Conf. Bioinforma. Bioeng. BIBE 2014, pp. 379–385, 2014, doi: 10.1109/BIBE.2014.73.

[14] Srivastava, M., Srivastava, U., & Srivastava, S." Modified Caesar Cipher with image steganogra-phy". International Conference on Information Systems and Computer Networks (ISCON)(pp. 1–6), 2023 .

[15] Q.-A. Kester, “A cryptosystem based on Vigenère cipher with varying key (virtual) View project,” Int. J. Adv. Res. Comput. Eng. Technol., vol. 1, no. 10, pp. 15–17, 2021.

[16] D. Veera, R. Mangrulkar, C. Bhadane, K. Bhowmick, and P. Chavan, “Modified Caesar Cipher and Card Deck Shuffle Rearrangement Algorithm for Image Encryption,” J. Inf. Telecommun., vol. 8, no. 2, pp. 280–300, 2024, doi: 10.1080/24751839.2023.2285549.

[17] K. Sasikumar and S. Nagarajan, “Comprehensive Review and Analysis of Cryptography Techniques in Cloud Computing,” IEEE Access, vol. 12, no. February, pp. 52325–52351, 2024, doi: 10.1109/ACCESS.2024.3385449.

[18] A. Mehmood, A. Shafique, M. Alawida, and A. N. Khan, “Advances and Vulnerabilities in Modern Cryptographic Techniques: A Comprehensive Survey on Cybersecurity in the Domain of Machine/Deep Learning and Quantum Techniques,” IEEE Access, vol. 12, no. February, pp. 27530–27555, 2024, doi: 10.1109/ACCESS.2024.3367232.

[19] L. C. Han and N. M. Mahyuddin, “An implementation of caesar cipher and XOR encryption technique in a secure wireless communication,” 2014 2nd Int. Conf. Electron. Des. ICED 2014, pp. 111–116, 2011, doi: 10.1109/ICED.2014.7015781.

[20] A. Jain, R. Dedhia, and A. Patil, “Enhancing the Security of Caesar Cipher Substitution Method using a Randomized Approach for more Secure Communication,” Int. J. Comput. Appl., vol. 129, no. 13, pp. 6–11, 2015, doi: 10.5120/ijca2015907062.

[21] S. B. Dar, “Enhancing The Security of Caesar Cipher Using Double Substitution Method,” Int. J. Comput. Sci. Eng. Technol., vol. 5, no. 7, pp. 772–774, 2014.

[22] J. Luxemburk, K. Hynek, and T. Cejka, “Detection of HTTPS Brute-Force Attacks with Packet-Level Feature Set,” 2021 IEEE 11th Annu. Comput. Commun. Work. Conf. CCWC 2021, pp. 114–122, 2021, doi: 10.1109/CCWC51732.2021.9375998.

[23] Ezenwobodo and S. Samuel, “International Journal of Research Publication and Reviews,” Int. J. Res. Publ. Rev., vol. 04, no. 01, pp. 1806–1812, 2022, doi: 10.55248/gengpi.2023.4149.

[24] A. A. Hamza and R. J. surayh Al-Janabi, “Detecting Brute Force Attacks Using Machine Learning,” BIO Web Conf., vol. 97, pp. 1–15, 2024, doi: 10.1051/bioconf/20249700045.

[25] M. Tahboush, A. Hamdan, F. Alzobi, M. Husni, and M. Adawy, “NTDA: The Mitigation of Denial of Service (DoS) Cyberattack Based on Network Traffic Detection Approach,” Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 3, pp. 692–698, 2024, doi: 10.14569/IJACSA.2024.0150370.

[26] A. Ghandour and B. J. Woodford, "Guidelines to Develop a Cybersecurity Policy in Schools, Perspectives Informed from Jordanian Cybercrime Law" International Arab Conference on Information Technology (ACIT), Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10876919.

[27] A. Y. A. Bani Ahmad, M. Allahham, W. I. Almajali, F. T. Ayasrah and S. Sabra, "Blockchain's Role in Emerging Markets: Accelerating Digital Supply Chain Management and Unlocking New Opportunities," 2024 25th International Arab Conference on Information Technology (ACIT), Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10877053.

[28] T. Jamil, “Impact of shift operations on (-1+j)-base complex binary numbers,” J. Comput., vol. 3, no. 2, pp. 63–71, 2008, doi: 10.4304/jcp.3.2.63-71.