A Cryptographic Algorithm Based on ASCII and Number System Conversions along with a Cyclic Mathematical Function

Mahmudul Hasan Moon[1], Md. Palash Uddin[2], Masud Ibn Afjal[3], Md. Al Mamun[4], Md. Abu Marjan[5] and Adiba Mahjabin Nitu[6]

[1-3,6]Faculty of Computer Science and Engineering, Hajee Mohammad Danesh Science and Technology University (HSTU), Dinajpur-5200, Bangladesh

[4]Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology
Rajshahi-6204, Bangladesh

[5]CodersTrust, Dhaka, Bangladesh

[1]mahmudulmoon123@gmail.com, [2]palash_cse@hstu.ac.bd, [3]masud@hstu.ac.bd, [4]a.mamun@ruet.ac.bd, [5]mdabumarjan66@gmail.com, [6]nitu.hstu@gmail.com

Abstract— **Data encryption and decryption in an efficient manner are the challenging aspects of modern information theory. An efficient cryptology algorithm is introduced in this paper to offer comparatively higher security of data. In this algorithm, the plaintext to be encrypted is converted into unprintable characters. For encryption, a different technique is applied based on ASCII and number system conversions, which makes this algorithm different from others. First, each character of the plaintext is converted into its equivalent ASCII (decimal) which is further converted to its equivalent octal and hexadecimal numbers. Then, using some matrix manipulations on the decimal, octal and hexadecimal representation of each character is transformed to 5 unprintable characters. After that, every unprintable character in the intermediate cipher text is further converted into a different unprintable character using a cyclic mathematical function. Performing three steps of processing, the final encrypted message is produced that gives higher level of security. In this way, as there can have total 32 unprintable characters, it will take much time if the intruders try to decrypt the original message with every probable combination. Though the length of the encrypted message is larger than original message in this proposed algorithm, it offers higher security for the real-time communications.**

Keywords— Cryptography, Encryption and decryption, Number system conversion, Higher level of security, ASCII conversions.

I. INTRODUCTION

Cryptography, Steganography and watermarking are three popular modern security offering techniques. Among them, the cryptography is older and mostly used technique as it is easy to implement and offers higher level of security. The cryptography is the study of mathematical techniques related to the aspects of information security such as confidentiality, data integrity, entity authentication and data origin authentication [1]. Cryptography historically dealt with the construction and analysis of protocols that would prevent any third parties from reading a private communication between two parties. In the digital age, cryptography has evolved to address the encryption and decryption of private communications through the Internet and computer systems [2]. Until modern times, cryptography referred almost exclusively to encryption, which is the process of converting original information (called plaintext) into unintelligible text (called cipher text). On the other hand, decryption is the reverse procedure that moves from the unintelligible cipher text back to plaintext. A cipher is a pair of algorithms that create the encryption and the reversing decryption [3]. It is a common misconception that every encryption method can be broken. In connection with WWII work at Bell Labs, Claude Shannon proved that the one-time pad cipher is unbreakable, provided the key material is truly random, never reused, kept secret from all possible attackers, and of equal or greater length than the message [4]. In this paper, we proposed an improved algorithm which is different from the traditional symmetric-key cryptography, asymmetric-key cryptography or hashing function. ASCII and number system conversions are also used in this paper which makes the cipher different from other algorithms.

II. RELATED WORK

Day by day the level of security is going to be higher. Still now many researchers are working on cryptography and data hiding. A new cryptographic algorithm for the Real Time Application was in [5] to improve the time for encryption and decryption of data of end-to-end delay and to provide higher level of security. A cryptographic algorithm based on ASCII conversion and a cyclic mathematical function was presented in [6]. A user of RSA creates and then publishes a public key based on two larges prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can decode the message feasibly [7]. Some researchers have developed a new cryptosystem using multiple cryptographic assumptions which offers a greater security level than that schemes based on a single cryptographic assumption [8]. Blowfish is a symmetric-key block cipher, designed in 1993 by B. Schneier and included in a large number of cipher suites and encryption products. Blowfish provides a good encryption rate in software and no effective cryptanalysis of it has been found to date. However, the Advanced Encryption Standard (AES) now receives more attention, and Schneier recommends Twofish for modern applications [9]. A basic study on cryptography which is a solution for information security threats has been shown in [10]. In this paper, a new cryptographic algorithm is proposed which involves ASCII, number system conversions and the cyclic mathematical used in [6].

III. Proposed Algorithm

A. Encryption Algorithm

In encryption phase, the plaintext is divided into its constituent characters and each character is encrypted separately. For each character, at the beginning, the character is converted into its equivalent ASCII and the ASCII is considered as decimal. Using number conversion formula, the decimal is converted into its equivalent octal and hexadecimal. The product of the 3 numbers (decimal, octal and hexadecimal) of the character is calculated and then the product is converted into its equivalent binary. In here we perform decimal multiplication of three numbers. Because multiplication of decimal, octal and hexadecimal are not possible. So, we consider all the number as 10 base during the time of multiplication for simplicity. The multiplication is shown in encryption algorithm at phase 3(c). A 5×5 matrix is taken and the bits of the product are put into the matrix from last to first. We put the bits from last to first that means we do not put the original binary. Besides, we put the reverse binary of the original which also gives another level of security. After putting all the bits of binary, if any place does not have the binary bit, zero (0) is put on the place and it is mandatory to fulfill each cell of 5×5 matrix. Then, the bits of every column of the matrix are read and the bits are converted into their decimal equivalent (denoted as **New_ASCII**). Alternatively, the bits of every row can be read and converted after taking the transpose of the 5×5 matrix. For one layer higher security, the cyclic mathematical function [6] has been used to encrypt decimal values of the intermediate encrypted data. The mathematical function as shown below is called cyclic because its output is rotated between 0 and 31.

$$\mathbf{Final\_ASCII}[i] = (\mathbf{New\_ASCII}[i] + m) \ \% \ 32, \qquad (1)$$

where, $0<m<32$. At last, the 5 **Final_ASCII**[*i*] values for each character are converted to their equivalent characters which are undoubtedly unprintable so that the final cipher text cannot be shown at all. This encryption process repeats for each character of the original data. Then, combining all the encrypted characters as a single it is sent to the receiver. The pseudocode of the proposed encryption algorithm is given below whereas the encryption flowchart is shown in Fig. 1.

```
1. Input original message.
2. Divide the message into its constituent characters.
3. For each character
    a) Convert the character into its equivalent ASCII
       (consider the ASCII as decimal).
    b) Covert the ASCII into octal and Hexa-decimal
       equivalent.
    c) Make the product of the decimal, octal, and
       hexadecimal.
    d) Convert the product into binary and put the bits
       on a 5×5 matrix from last to first.
    e) Read the bits from each column and make
       decimal. We get five decimal values in
       New_ASCII array.
    f) For $i$=1 to 5
    g) Do $\mathbf{Final\_ASCII}[i] = (\mathbf{New\_ASCII} + m) \% 32$,
       where, $0<m<32$.
    h) Convert the Final_ASCII[$i$] into its equivalent
       character.
4. End encryption.
```

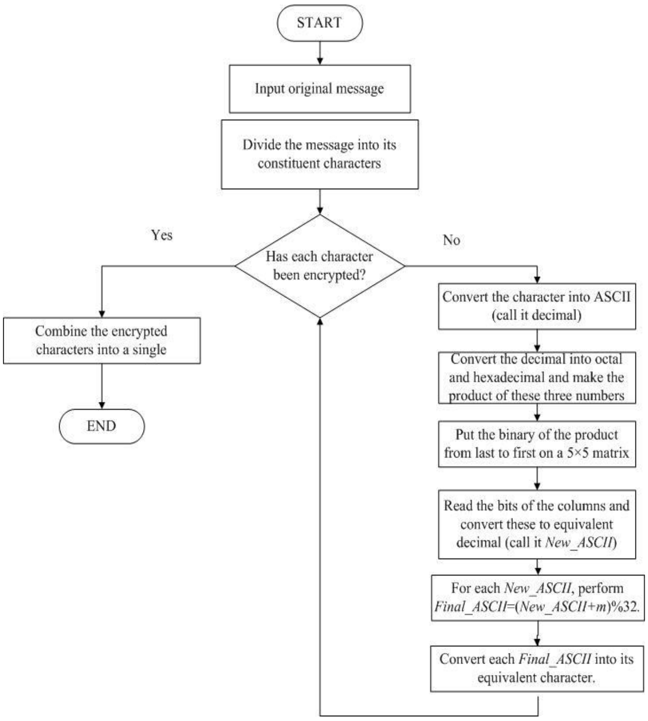

START
Input original message
Divide the message into its constituent characters
Has each character been encrypted?
Yes
No
Convert the character into ASCII (call it decimal)
Convert the decimal into octal and hexadecimal and make the product of these three numbers
Put the binary of the product from last to first on a 5×5 matrix
Read the bits of the columns and convert these to equivalent decimal (call it New_ASCII)
For each New_ASCII, perform Final_ASCII=(New_ASCII+m)%32.
Convert each Final_ASCII into its equivalent character.
Combine the encrypted characters into a single
END

Fig. 1: Flowchart of the encryption algorithm

B. Decryption Algorithm

In the decryption phase, first the unprintable characters are converted into their equivalent ASCII (call it **D_ASCII**) between 0 to 31. The following reverse cyclic mathematical function is applied on the ASCII.

$$\mathbf{DEC\_ASCII}\,[i] = (\mathbf{D\_ASCII}[i] - m + 32) \ \% \ 32 \qquad (2)$$

where, $0<m<32$. All **DEC_ASCII** values are converted into 5-bit binary numbers. Taking 5 consecutive binary numbers a 5×5 matrix is formed placing the numbers row-wise. The transpose of the matrix is taken. Then, the bits are read row-wise and they are placed from last to fast to form a 25-bit binary number. This 25-bit binary number is converted into its equivalent decimal number. The decimal that is found from the matrix is the product of the original character's ASCII and its octal and hexadecimal equivalent. From this product finding the original character's ASCII is a challenging process. After calculation on this, we find that these three numbers must be the divisor of the product and the value of the divisor is square root of one fourth of the product. For finding the original character's ASCII, first all divisors from 2 to the point (one fourth of the product) are calculated. The finding character has a subset of 3 numbers that's the three numbers are related with each other as follows. One is decimal, another is octal and other is hexadecimal of the decimal. This is reverse multiplication phase where we find the multiplicand from the product. The reverse multiplication is performed in the decryption algorithm at the phase (e) to (m). They have an extra characteristic that the product of these three must be the decimal that is formed from the matrix. Then, the decimal of the subset is the original ASCII of the character. The ASCII is converted into its equivalent character. The same process is done for each character to be found. The pseudocode of the decryption procedure using the parameters discussed above can be summarized as follows.

```
1.  Input encrypted message.
2.  For each character:
    a)  Convert the character to ASCII (call it D_ASCII).
    b)  Do DEC_ASCII [i] = (D_ASCII[i]-m+32) % 32 for each D_ASCII.
3.  Take consecutive 5 characters' DEC_ASCII as a group.
4.  For each group:
    a)  Convert each number to its equivalent 5-bit binary.
    b)  Put the binary to the 5×5 matrix row-wise
    c)  Transpose the matrix.
    d)  Read the bits row-wise and place them from last to fast to form a 25-bit binary number. This 25-bit binary number is converted into its equivalent decimal number (consider the decimal as P).
    e)  For i=2 to sqrt (P/4)
    f)      if (P% i= =0)
    g)          divisor [i]=i;
    h)  For i=1 to size of divisor
    i)      decimal=divisor [i]
    j)      oct = octal (decimal)
    k)      hex = Hexadecimal (decimal)
    l)      For i= 1 to size of divisor
    m)          If ((oct = = divisor [i] && oct ! = decimal)) && (hex = = divisor[i]) && (oct*hex*decimal = = P)
    n)              decimal is the ASCII of the original message's character
    o)  Convert the ASCII (decimal) to its equivalent character.
5.  End decryption.
```

IV. Explanation With An Example

A. Encryption

As an example, let's take a message "HSTU" to be encrypted. According to the encryption algorithm each character of the plaintext is considered as a different part as shown in Fig. 2.

H	S	T	U

Fig. 2: Each character is divided into different part

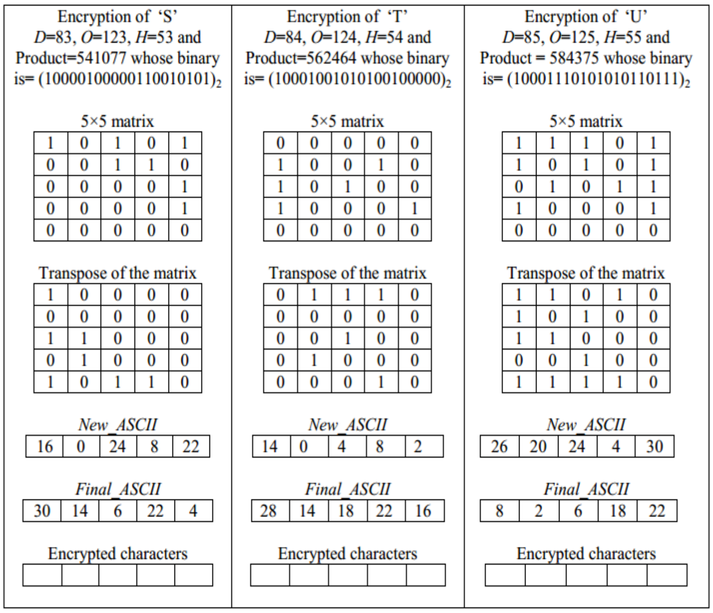
Each character is converted into 5 unprintable characters as the encryption algorithm. First, each character is converted into its equivalent ASCII. Then, the ASCII is considered as a decimal number and the number is converted into its equivalent octal and hexadecimal. The product of the decimal, octal and hexadecimal is calculated. Here the product is simple decimal multiplication. The product is converted into the binary and the binary is put from last to first to a 5×5 matrix as shown below. Consider D = decimal, O = octal and H = hexadecimal. Then, for character 'H', D = ASCII of 'H' = 72, O = 110, H = 48, and the product P = 380160. The binary of the product is = $(1011100110100000000)_2$. The 5×5 matrix is constructed as follows.

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

To read the matrix, the matrix is transposed for easily read in row-wise as shown below which also includes the final encrypted unprintable characters.

					New_ASCII	*Final_ASCII*	Encrypted character
0	0	1	1	0	6	20	
0	0	1	1	0	6	20	
0	0	0	0	0	0	14	
0	1	0	1	0	10	24	
0	0	1	0	0	4	18	

Now, for 'S', 'T' and 'U', the encryption procedure is illustrated in the following Fig. 3.



Encryption of 'S'
D=83, O=123, H=53 and Product=541077 whose binary is= $(10000100000110010101)_2$

5×5 matrix

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

Transpose of the matrix

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

New_ASCII

16	0	24	8	22

Final_ASCII

30	14	6	22	4

Encrypted characters

Encryption of 'T'
D=84, O=124, H=54 and Product=562464 whose binary is= $(10001001010100100000)_2$

5×5 matrix

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

Transpose of the matrix

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

New_ASCII

14	0	4	8	2

Final_ASCII

28	14	18	22	16

Encrypted characters

Encryption of 'U'
D=85, O=125, H=55 and Product = 584375 whose binary is= $(10001110101010110111)_2$

5×5 matrix

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

Transpose of the matrix

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

New_ASCII

26	20	24	4	30

Final_ASCII

8	2	6	18	22

Encrypted characters

Fig. 3: Example of the encryption procedure

Finally, the plaintext "HSTU" is encrypted according to the proposed algorithm where the 4 characters are encrypted to 20 unprintable characters as follows.

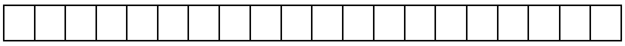


Fig. 4: Final cipher text

B. Decryption

The cipher text contains 20 unprintable characters for the considered example. To decrypt the text, each unprintable character is converted into its equivalent ASCII (call it *D_ASCII*). Then, each ASCII is converted into a number (call it *DEC_ASCII*) using inverse cyclic mathematical function. After that, 5 consecutive numbers are taken as a group and each number is converted into its equivalent 5-bit binary number. The 5 5-bit binary numbers are placed row-wise to form a 5×5 matrix. Then, the transpose of the matrix is taken and the bits of the rows are read to form a 25-bit binary numbers placing the bits from last to first. The constructed binary number is converted into its equivalent decimal (call it *P*) which is the product of the ASCII of the original character and its octal and hexadecimal equivalent. Using this formula

Encrypted text																			
D_ASCII																			
20	20	14	24	18	30	14	6	22	4	28	14	18	22	16	8	2	6	18	22
DEC_ASCII																			
6	6	0	10	4	16	0	24	8	22	14	0	4	8	2	26	20	24	4	30

Fig. 5: *D_ASCII* and *DEC_ASCII* of the cipher text

Now, to decrypt the first character of the plaintext let's consider {6, 6, 0, 10, 4} of the *DEC_ASCII* as a group whose matrix representation is shown below:

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

The transpose of the matrix to form the 25-bit binary of *P* is as follows.

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

Thus, the binary of *P* is $(10111001101000000000)_2$ whose decimal representation is P = 380160. As *P* is the product of 3 numbers, using decryption algorithm we get $D = 72$, $O = 110$, and $H = 48$. As a result the decrypted character is 'H'. The decryption procedure of the other characters from the cipher text is illustrated in Fig. 6 which results the decrypted text as "HSTU".

Decryption of the second character Group: (16,0,24,8,22)	Decryption of the third character Group: (14,0,4,8,2)	Decryption of the fourth character Group: (26,20,24,4,3)
The 5×5 matrix	The 5×5 matrix	The 5×5 matrix
$\begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{matrix}$	$\begin{matrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{matrix}$	$\begin{matrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{matrix}$
The transpose of the matrix	The transpose of the matrix	The transpose of the matrix
$\begin{matrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$
The binary of *P* is $(10000100000110010101)_2$ whose decimal equivalent is P = 541077. Thus, $D = 83$, $O = 123$ and $H = 53$. The decrypted character is 'S'.	The binary of *P* is (10001001010100100000)2 whose decimal equivalent is P = 562464. Thus, $D = 84$, $O = 124$ and $H = 54$. The decrypted character is 'T'.	The binary of *P* is $(10001110101010110111)_2$ whose decimal equivalent is P = 584375. Thus, $D = 85$, $O = 125$ and $H = 55$. The decrypted character is 'U'.

Fig. 6: Decryption of the cipher text

V. RESULT AND DISCUSSION

A. Experiment

The proposed algorithm was implemented by C++ programming language. The plaintext to be encrypted through the algorithm is "Bangla". Applying the presented encryption algorithm the cipher text of unprintable characters is shown in Fig. 7. C++ language displays the boxes for unprintable

we can get the actual message. The following Fig. 5 shows the cipher text, *D_ASCII* and *DEC_ASCII* values for the considered example.

characters. In this way, the cipher text is a collection of boxes in which the plaintext is concealed. At receiver, applying the decryption algorithm on the cipher text the plaintext is extracted properly.

□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Fig. 7: The cipher text

We also add some example below.

Plane Text	Cipher text
# $ (	▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1 2 3 4 5	▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
M A T H	▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
b a n g l a	▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯

It has been observed that after encryption all the character is converted into unprintable character so that it is not easy to decrypt the message. Though the encrypted message takes more memory than the original, is gives higher security which protect the message from the access of third party. This algorithm is different from others for its ASCII and number system conversion processes. Three-level encryption is done in the algorithm and at each level the character of the original message is changed. In the decryption phase for reducing time we continue the loop at square root of one fourth of the number. This proposed cryptographic algorithm can be used in different security systems such as mobile communication systems, E-mail communication systems, cloud-based systems, banking security systems, administrative systems, key or password management systems, network security systems, protocol management systems etc.

B. Security Analysis of the Algorithm in front of Brute Force Attacks

If someone wants to decrypt the original message in a Brute Force manner, it is very unbreakable. As the plaintext "HSTU" is converted into 20 unprintable characters, it means for each character there is 5 unprintable characters. We know there are 32 unprintable characters in computer system. To choose 5 unprintable characters from 32, the intruder needs to make ${}^{n}C_{r}$ combinations which produces = 32! / ((32-5)!*5!)

$$=(2.63*10^{35})/(1.3*10^{30})$$
$$= 201275 \text{ combinations}$$

Thus, for a plaintext of 4 characters the intruder should check (201275*4) = 805100 combinations. Roughly, if one-check takes one minute time, then it takes total 805100 minutes=13418.33 hours=559 days=1.53 years. Consequently, if the size of the plaintext is increased, then it is very difficult to decrypt the message. Therefore, it can be said that it gives higher security.

VI. CONCLUSION

To ensure higher security and to hide data in effective way the proposed algorithm contributes greatly. Here, we present an algorithm which is based on ASCII conversion and number system conversion and a cyclic mathematical function. This algorithm not only encrypts the data but also hides the data which gives more security. In future we will try to increase the security technique and implement some real time security system and try to add steganography with the system.

REFERENCES

[1] Sidhpurwalahuzaifa. *A Brief History of Cryptography*. [Online]. Available: https://securityblog.redhat.com/2013/08/14/a-brief-history-of-cryptography/

[6] M. P. Uddin, M. A. Marjan, N. B. Sadia and M. R. Islam, “Developing an Efficient Algorithm to Combine Cryptography and Steganography Based on ASCII Conversions and Cyclic Mathematical Function,” *3rd IEEE International Conference on Informatics, Electronics & Vision*, May 23-24, 2014.

[7] R. Rivest, A. Shamir, L. Adlemanm, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] ECPI University, *Cyber and Network Security - Bachelor’s*. [Online]. Available: https://www.ecpi.edu/blog/crypotgraphy-and-network-security

[3] K. David, *The Codebreakers – The Story of Secret Writing*, 1967.

[4] C.E. Shannon, W. Weaver, *THE MATHEMATICAL THEORY OF COMMUNICATION*. THE UNIVERSITY OF ILLINOIS PRESS, URBANA, 1964.

[5] A. H. Omari, B. M. Al-Kasasbeh, R. E. Al-Qutaish, and M. I. Muhairat, “A New Cryptographic Algorithm for the Real Time Applications,” *Proc. of the 7th WSEAS International Conference on INFORMATION SECURITY and PRIVACY (ISP)*, pp. 33-38, 2008.

[8] E.S. Ismail and M.S. Hijazi, “New Cryptosystem Using Multiple Cryptographic Assumptions,” *Journal of Computer Science*, vol. 7, no.12, pp. 1765-1769, 2011.

[9] Dahna and McConnachie, "Bruce Almighty: Schneier preaches security to Linux faithful," *Computerworld*. p. 3.

[10] M. V. Kumar, “Cryptography–A solution for information security Threats,” *Golden Research Thoughts*, vol. 2, no.1, 2013.