Modern Encryption and Decryption Algorithm based on ASCII Value and Binary Operations

Deepak
Bahra University
Shimla Hills, Himachal Pradesh
India

Parveen
Bahra University
Shimla Hills, Himachal Pradesh
India

ABSTRACT
Security is the most important issue from last few decades. Cryptography plays an important role in security. Encryption and Decryption came as a solution against security threats. There are so many proposed algorithms by researchers like AES,DES,RSA and many more but they are lacking somewhere in terms of security. Therefore cryptography demands such encryption and decryption algorithms which are real hard to crack. In this paper, an algorithm has been proposed based on ASCII values and binary operations for both encryption and decryption to enhance the security so that the attackers might not easily crack the logic to reach the original message sent via network.

General Terms
Cryptography, Encryption, Decryption, Algorithm etc.

Keywords
ASCII, Binary, Encryption, Decryption, Symmetric Encryption Algorithm, Plain Text, Cipher Text

1. INTRODUCTION
The word "cryptography" is made of two Greek roots: kryptos, meaning secret, and graphos, meaning writing. Originally, "secret writing" was the main topic for those who studied in this area. However, secret writing is no longer the only focus to cryptographic research. In fact, our work has gone far beyond that.[1] Cryptography played a major role in the course of World War II, and some of the first working computers were dedicated to cryptanalytic tasks.[2] Cryptography has recently played a significant role in secure data transmissions and storages. Most conventional data encryption schemes are relatively complicated and complexity in encrypted keys is insufficient, resulting in long computational time and low degree of security against all types of attacks. Consequently, a highly secured and robust data encryption scheme is necessary.[8]

Encryption is basically a process or algorithm to make information hidden or secret. It is considered as the subset of cryptography. It is the actual process of applying cryptography. It is the process to transform or converting the data into some another form that appears to be random, meaningless and unintelligible. It can also be said that encryption is the process of transforming plaintext into the cipher text where plaintext is the input to the encryption process and cipher text is the output of the encryption process.[5] Decryption is the reverse process of Encryption. In decryption, at receiver side ,cipher text is converted to plaintext by using secret key and performing operations.

In computer systems, the algorithm consists of complex mathematical formulas that dictate the rules of conversion process from plain text to cipher text and vice versa combined with the key. However, some of the encryption and decryption algorithms use the same key (i.e. sender and receiver).[6]

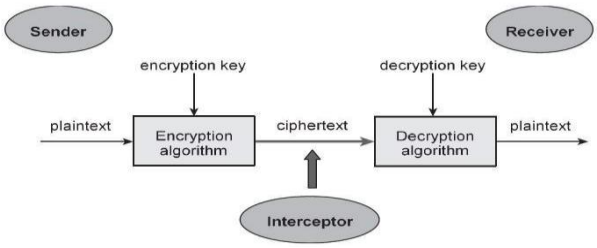

Fig 1: Model of Cryptosystem

2. ENCRYPTION ALGORITHM
This algorithm is used to encrypt data by using ASCII values of the data to be encrypted. The secret key used is same for both encryption and decryption of the message. So this is a symmetric encryption algorithm. Length of the secret key need not to be the same as the length of plain text but it can be made equal.

Assumptions:
p[i] be an array that stores binary equivalent of plain text of **ith** position.

s[j] be an array that stores binary equivalent of secret key of **jth** position.

e[k] be an array that stores binary equivalent of results after operation of kth position.

Plain Text: LIFE IS A JOURNEY NOT A RACE.

Secret Key: SUCCESS

Step 1: Give input as plain text and store plain text in the form of String.

Plain Text: LIFE IS A JOURNEY NOT A RACE

Step 2: Apply permutation on the string and place the words at odd position first and words at even position after placing odd positioned words. Now plain text will become (Assuming first word at zeroth position) :

Plain Text: IS JOURNEY A LIFE A NOT RACE

Step 3: Remove white spaces from the new plain text i.e. spaces and use the given secret key for each character. If secret key is short then apply repetitions in the secret key so that each character in plain text get a character from secret key.
Plain Text: ISJOURNEYALIFEANOTRACE

KEY: SUCCESSSUCCESSSUCCESSS

Step 4: Find ASCII value of each character for both Plain text and Key. Convert ASCII values into its binary equivalent (8 bits).

Table 1. ASCII Value and its Binary Equivalent used in Text

Character	ASCII Value	Binary Equivalent
I	73	01001001
S	83	01010011
J	74	01001010
O	79	01001111
U	85	01010101
R	82	01010010
N	78	01001110
E	69	01000101
Y	89	01011001
A	65	01000001
L	76	01001100
F	70	01000110
T	84	01010100
C	67	01000011

Step 5: Perform addition on ASCII binary equivalent for each character of plain text and secret key which are at similar positions.

Table 2. Table Depicting Addition Operation

Position	Binary Equivalent of Plain Text at Given Position i.e. p[i]	Binary Equivalent of Secret Key at Given Position i.e. s[j]	Addition of Both Binary Equivalents i.e. e[k]
0	01001001	01010011	10011100
1	01010011	01010101	10101000
2	01001010	01000011	10001101
3	01001111	01000011	10010010
4	01010101	01000101	10011010
5	01010010	01010011	10100101
6	01001110	01010011	10100001
7	01000101	01010011	10011000
8	01011001	01010101	10101110
9	01000001	01000011	10000100
10	01001100	01000011	10001111
11	01001001	01000101	10001110
12	01000110	01010011	10011001
13	01000101	01010011	10011000
14	01000001	01010011	10010100
15	01001110	01010101	10100011
16	01001111	01000011	10010010
17	01010100	01000011	10010111
18	01010010	01000101	10010111
19	01000001	01010011	10010100
20	01000011	01010011	10010110
21	01000101	01010011	10011000

Step 6: Now find the 2's complement of e[k] (Use Step 5 Table for e[k])

Table 3. 2's complement Table for Binary Equivalent

Position	Addition of Both Binary Equivalents i.e. e[k]	2's complement of e[k]
0	10011100	01100010
1	10101000	01010110
2	10001101	01110001
3	10010010	01101100
4	10011010	01100100
5	10100101	01011001
6	10100001	01011101
7	10011000	01100110
8	10101110	01010000
9	10000100	01111010
10	10001111	01101111
11	10001110	01110000
12	10011001	01100101
13	10011000	01100110
14	10010100	01101010
15	10100011	01011011
16	10010010	01101100
17	10010111	01100111
18	10010111	01100111
19	10010100	01101010
20	10010110	01101000
21	10011000	01100110

Step 7: The encrypted text corresponding to plain text in bit format will become (say new e[k])

011000100101011001110001011011000110010001011001010111010110011001010000011110100110111101110000011001010110011001101010010110110110110001100111011001110110101001101000011001100

Now apply 2 bit circular right shift

100010010101100111000101101100011001000101100101011101011001100101000001111010011011110111000001100101011001100110101001011011011011000110011101110011101101010011010000110011001

Step 8: Now pair 8 bits for each index position in same order and convert the bits into corresponding decimal number. Assume this decimal number as ASCII value and convert ASCII value to respective character or symbol.

Table 4. Conversion to ASCII Value and Corresponding Character

Position	Circular Right Shift of e[k]	ASCII Value for e[k]	Corresponding Character for ASCII value
0	10001001	137	ë
1	01011001	89	Y
2	11000101	197	┼
3	10110001	177	▒

4	10010001	145	æ
5	01100101	101	e
6	01110101	117	u
7	10011001	153	Ö
8	01000001	65	A
9	11101001	233	Ú
10	10111101	189	¢ (cent symbol)
11	11000001	193	┴
12	10010101	149	ò
13	10011001	153	Ö
14	10101001	169	® (Registered trademark symbol)
15	01101101	109	m
16	10110001	177	▒
17	10011101	157	Ø
18	10011101	157	Ø
19	10101001	169	® (Registered trademark symbol)
20	10100001	161	í
21	10011001	153	Ö

The text after encryption will be

Cipher Text: ëY┼▒ æeuÖAÚ¢┴òÖ®m▒ØØ®íÖ

3. DECRYPTION ALGORITHM

Assumptions:-
l[k] be an array that stores updated binary equivalent of ciphertext of **kth** position.

a[j] be an array that stores binary equivalent of secret key of **jth** position.

b[i] be an array that stores subtraction of l[k] and a[j] of **ith** position.

Cipher Text: ëY┼▒ æeuÖAÚ¢┴òÖ®m▒ØØ®íÖ

Secret Key: SUCCESS

Step 1: Assign each character a position starting from 0. Convert the each character of cipher text into the ASCII value and convert that ASCII value into binary

Table 4. Conversion to Characters to Binary Equivalent

Position	Character of Cipher Text at Given Position	ASCII Value	Binary Equivalent
0	ë	137	10001001
1	Y	89	01011001
2	┼	197	11000101
3	▒	177	10110001
4	æ	145	10010001
5	e	101	01100101
6	u	117	01110101
7	Ö	153	10011001
8	A	65	01000001
9	Ú	233	11101001
10	¢ (cent symbol)	189	10111101
11	┴	193	11000001
12	ò	149	10010101
13	Ö	153	10011001
14	® (Registered trademark symbol)	169	10101001
15	m	109	01101101
16	▒	177	10110001
17	Ø	157	10011101
18	Ø	157	10011101
19	® (Registered trademark symbol)	169	10101001
20	í	161	10100001
21	Ö	153	10011001

Step 2: Apply 2 bit left circular shift on binary equivalent

Binary Equivalent
10001001010110011100010110110001100100010110010101
11010110011001010000011110100110111011100000110010101100110011010100101101101101100011001110110011101100111
01101010011010000110011001

After applying 2 bit left circular shift
01100010010101100111000101101100011001000101100101
01110101100110010100000111101001101111011100000110
01010110011001101010010110110110110001100111011001
11011010100110100001100110

Step 3: Break the binary text into pairs of 8 bit for each position and apply 2's complement on each pair

Table 5. 2's Complement of Binary Equivalent

Position	Circular Left Shift of l[k]	2's complement of l[k]
0	01100010	10011100
1	01010110	10101000
2	01110001	10001101
3	01101100	10010010
4	01100100	10011010
5	01011001	10100101
6	01011101	10100001
7	01100110	10011000
8	01010000	10101110
9	01111010	10000100
10	01101111	10001111
11	01110000	10001110
12	01100101	10011001
13	01100110	10011000
14	01101010	10010100
15	01011011	10100011
16	01101100	10010010
17	01100111	10010111

18	01100111	10010111
19	01101010	10010100
20	01101000	10010110
21	01100110	10011000

Step 4: Use the Secret key so that each text will get a character from Secret Key if number of character in Secret Key is less than the number of character in Cipher Text then apply repetition in secret key. Convert each character of secret key into ASCII value and then its binary equivalent.

Key: SUCCESSSUCCESSSUCCESSS

Table 6. Table depicting Updated Cipher Text Equivalent and Binary Equivalent of Secret Key

Position	Updated Cipher Text l[k]	Binary Equivalent of Secret Key at Given Position i.e. a[j]
0	10011100	01010011
1	10101000	01010101
2	10001101	01000011
3	10010010	01000011
4	10011010	01000101
5	10100101	01010011
6	10100001	01010011
7	10011000	01010011
8	10101110	01010101
9	10000100	01000011
10	10001111	01000011
11	10001110	01000101
12	10011001	01010011
13	10011000	01010011
14	10010100	01010011
15	10100011	01010101
16	10010010	01000011
17	10010111	01000011
18	10010111	01000101
19	10010100	01010011
20	10010110	01010011
21	10011000	01010011

Step 5: Subtract binary equivalent of Secret key from Updated Cipher text for same positions

Table 7. Subtraction Operation between l[k] and a[j]

Position	Updated Cipher Text l[k]	Binary Equivalent of Secret Key at Given Position i.e. a[j]	Binary Equivalent of Subtraction operation at Given Position i.e. b[i]
0	10011100	01010011	01001001
1	10101000	01010101	01010011
2	10001101	01000011	01001010
3	10010010	01000011	01001111
4	10011010	01000101	01010101
5	10100101	01010011	01010010
6	10100001	01010011	01001110
7	10011000	01010011	01000101
8	10101110	01010101	01011001
9	10000100	01000011	01000001
10	10001111	01000011	01001100
11	10001110	01000101	01001001
12	10011001	01010011	01000110
13	10011000	01010011	01000101
14	10010100	01010011	01000001
15	10100011	01010101	01001110
16	10010010	01000011	01001111
17	10010111	01000011	01010100
18	10010111	01000101	01010010
19	10010100	01010011	01000001
20	10010110	01010011	01000011
21	10011000	01010011	01000101

Step 6: Convert the resultant binary equivalent obtained after subtraction into ASCII values and corresponding character

Table 8. Conversion of ASCII Value to Character

Position	Resultant Binary Equivalent i.e. b[i]	ASCII value of Binary Equivalent	Corresponding Character
0	01001001	73	I
1	01010011	83	S
2	01001010	74	J
3	01001111	79	O
4	01010101	85	U
5	01010010	82	R
6	01001110	78	N
7	01000101	69	E
8	01011001	89	Y
9	01000001	65	A
10	01001100	76	L
11	01001001	73	I
12	01000110	70	F
13	01000101	69	E
14	01000001	65	A
15	01001110	78	N
16	01001111	79	O
17	01010100	84	T
18	01010010	82	R
19	01000001	65	A
20	01000011	67	C
21	01000101	69	E

Text : ISJOURNEYALIFEANOTRACE

Step 7: Use white spaces between the text wherever possible to form a meaningful word.

Text : IS JOURNEY A LIFE A NOT RACE

Step 8: Count the number of words in the text. First half of the text are those words which are at odd positions(in same order) and second half contains those words which are at even positions in the text (in same order). Apply permutation to rearrange them at the respective position.

Text : LIFE IS A JOURNEY NOT A RACE.

Thus plain text obtained as output is:

Plain text : LIFE IS A JOURNEY NOT A RACE.

4. ADVANTAGE OF ASCII VALUE BASED ENCRYPTION AND DECRYPTION ALGORITHM

In this approach, we can have same encrypted characters for different characters of the plain text. In the above example "LIFE IS A JOURNEY NOT A RACE", T and R from words NOT and RACE respectively and have same encrypted character which is Ø. So, if anyone trying to decrypt the text he would definitely face the problem as there can be many possibilities (meaning) for similar character decryption. There may be same encryption for similar cipher character but if the encryption is not same then an unauthorized user will have to apply so many permutations which will consume a lot of time to decode that message.

5. LIMITATION

There is a possibility that there might occur a character during encryption in Step 8 for any position of Encryption algorithm for which corresponding ASCII value and character does not exist. In that case, we can use any special character (like !,@,#,$,% etc. which are rarely used) to assign a ASCII character at that position.

6. CONCLUSION AND FUTURE WORK

Everyone needs to protect himself in today's World as this is the era of digitalization. .People are lacking somewhere in terms of security of the data. Latest example of security threat was "WannaCry ransomware attack", which targeted computers running the Microsoft Windows Operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency. So, goal should be to encrypt a message in such a way that it cannot be decrypted by anyone. Therefore, this ASCII value based encryption and decryption algorithm is a good alternative for data security. This algorithm shall be really helpful in applications like Cryptography, Network Security (Wired and Wireless), Encrypting and Decrypting etc.

The work proposed here can be further enhanced to make the algorithm more secure and the researchers are working in the field of Cryptography and developing many more algorithms.

7. REFERENCES

[1] New Tools in Cryptography: Mutually Indepent Commitments, Tweakable Block Ciphers, and Plaintext Awareness via Key Registration by Moses Liskov.

[2] Cryptography and Machine Learning Ronald L. Rivest* Laboratory for Computer Science Massachusetts Institute of Technology Cambridge, MA 02139

[3] The RC5 Encryption Algorithm Ronald L Rivest MIT Laboratory for Computer Science Technology Square, Cambridge, Mass. 02139.

[4] D. E. Denning. Cryptography and Data Security. Addison-Wesley, Reading, Mass., 1982.

[5] A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms.

[6] Obaida Mohammad Awad Al-Hazaimeh "A NEW APPROACH FOR COMPLEX ENCRYPTING AND DECRYPTING DATA" International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.2, March 2013

[7] Dr. Prerna Mahajan & Abhishek Sachdeva IITM, India "A Study of Encryption Algorithms AES, DES and RSA for Security" Global Journal of Computer Science and Technology Network, Web & Security Volume 13 Issue 15 Version 1.0 Year 2013

[8] Warakorn Srichavengsup and Wimol San-Um "Data Encryption Scheme Based on Rules of Cellular Automata and Chaotic Map Function for Information Security" International Journal of Network Security, Vol.18, No.6, PP.1130-1142, Nov. 2016

[9]