Smart Security Algorithm: Ensured Confidentiality and Integrity

N. Jeyanthi, Gopishetti Venkatesh, R. Thandeeswaran, K. Brindha

***Abstract*: *The main aim of this paper is to provide confidentiality, integrity to the message. It Encrypts the message when it is passing from the sender side to the receiver side. If we take the Caesar cipher it is weak against the brute force attack, dictionary attack but this algorithm is stronger against the brute-force attack and dictionary attack. It has both private and public key encryption which is unlikely for other encryption algorithms like hill cipher, play fair cipher etc. Some Encryption algorithms will prone to be attacked with man-in-the-middle attack but this will be overcome with using hash code concept. This proposal encrypts the message with the novel algorithm at the client side and does the counterpart at the server side. Algorithm implemented using python programming and proves our algorithm is stronger against brute force and dictionary attacks*.**

***Keywords* : *Confidentiality, integrity, security attacks, smart algorithm*.**

I. INTRODUCTION

Sending messages to one to another which has been most common in recent times. These messages must be protected especially when they are transmitting from one to another especially when you are sending an important message via mail or telegram or any other social media, so, the message must be protected from understanding or reading the message. While you are transmitting others will try to see the message which you are sending. So, this can be avoided using cryptography by which message gets scrambled which even gets meaningless or we can use steganography which makes the text invisible while transmitting the message. The cryptography is nothing but encrypting the message. It varies based on keys, text size and techniques. This will use the both public and private key encryption which ensures the confidentiality and integrity of the message. But even though you are encrypting the message, the content can be revealed using some of the attacks like Traffic analysis, Brute-force attack, Man-in-the-middle attack, Dictionary attack and replay attack. I have developed encryption algorithm which is stronger against the Brute-force attack i.e., it takes ((key size)! *m) *n time to solve this. It will not use the same as like most algorithms are using like a=1, b=2, c=3, d=4 so on. By which they cannot guess which format they are using like it may a=7, b=18 and so on by which they cannot decode the original message using the traffic analysis. We assume a modnum which known to only sender and receiver which will not be transmitted by which mod number will be known to only sender and receiver it ensures the confidentiality of the message. Even though encrypted message gets to the middle member it must be authenticated with the hash code which will be verified at the receiver side. It will not use the receiver's name as password or length of small will not be used as password it will use some special symbols which will make strong against the dictionary attack. Person who want to access the message, must be authenticated at both sender and receiver side by using the password to send or receive, mean to decrypt the message i.e., it is stronger against the impersonation attack.

Revised Manuscript Received on November 06, 2019.
***** Correspondence Author
Dr. N. Jeyanthi, Associate Professor ,VIT, Vellore for School of Information Science and Engineering. Vellore,Tamilnadu.
Mr. G. Venkatesh , Associate Professor ,VIT, Vellore for School of Information Science and Engineering. Vellore,Tamilnadu.
Dr. R. Thandeeswaran, Associate Professor ,Department of Digital Communications, School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, India.

II. LITERATURE SURVEY

A. Submission of the paper

Enhanced Encryption Algorithm (EEA), converts each letter into equivalent ascii code, convert them into binary format, calculates the number of bits using bytes*8 i.e., n, writes the prime numbers upto n, choose a random prime number and divide them into number of block by taken random number, take a sample random number by which it will rotate that many times, merge them and apply 2's complement, generate a key, 2's complement will be XOR with key and finally the will be converted into equivalent ascii code and this message what will you get is Encrypted message [1]. Process of decryption is reverse of encryption. They choose random number and random number of "for shifting the bits" as key which is unknown to hacker. But it is not against the brute-force attack it takes time to decrypt but we can decrypt the message. Man-in-the-Middle attack is also possible since there is no MAC or hash function to protect the message. Dictionary attack is also possible the user may choose a frequent key i.e., generally used by the people. It doesn't provide authentication.

In a secured encryption [2], they take plain text and add 2nd letter, 2nd letter will be added with 3rd letter so on. After which it will be XOR with the current temperature and resultant is Encrypted message. The process of decryption is reverse of encryption. Temperature may be any place so that hacker may not detect. But the temperature varies from -1000 to 1000 at maximum.so, it is prone to Brute-Force attack. It is also prone to man-in-the-middle attack where message authentication is not done where he can alter the message and send.

Double Chaining Algorithm [3]

generates the sub-key, encrypt in DCBC mode where with the help of each key it is encrypted twice. It will generate the 4 subkeys i.e., after the 4 times of encryption the output is fed to input once again to get 8 times encrypted which is final output. It uses S-boxes. After DCBC you finally get the encrypted message and decryption is the reverse process of encryption. It is against the brute-force attack, known-plain text attack, and frequency analysis attack.

Huffman compression combined with the encryption [4] assigns a code to each letter based on the frequency. In the first step of encryption, we convert into binary format, reverse them, form K1, make XOR with the binary format number with 2's complement gives the K2. Add a random number as key to get the final output. In the second Encryption, input form the text, convert them into strings, do XOR with K4, add the random key K5 to get the final encrypted output. Decryption is the reverse process of decryption. It is against the Brute-Force attack which is Huffman compression changes every for each text. But it is prone to frequency analysis and there is no authentication.

Link Encryption Algorithm [5] has three phases use linear feedback shift register, then will have bit reorganization register and finally non-linear function. We will get the 32-bit output from the LFSR which will be doing r0 and r6, r8 and rb which will be XOR to give into non-linear function which will be computed to give encrypted text. It is stronger against the Guess and Determine Attack, BDD attack, Timing attack.

Encryption Algorithm of Hill [6] converts the character into equivalent ASCII codes then, Vigenere encryption, generates 16 subkeys for 64 bit same as DES, after which for each message it will uses the hill cipher to encrypt the message and then convert to ASCII character to get the Encrypted message. Decryption is the reverse process of encryption. It is strong against the known cipher text attack. The text can be decrypted by brute force attack.

Homomorphic Encryption Algorithms over Integers [7] has mainly four phases i.e., key generation which generates the both public and private key, algorithm encryption with public key, algorithm decryption is done with private key, and last phase is algorithm evaluation is done after the encryption using public and evaluated using the same public key. Choose a number m which is mod with 2 and added with 2 more mod numbers. Decryption is cipher text mod p random number and with mod 2. It will reduce the noise. It is optimized than the previous algorithms that they have used.

Implementation of Energy Efficient/Lightweight Encryption Algorithm for Wireless Body Area Networks [8], has 2 phases of key schedule and encryption/decryption. It will have different key sizes and based on which number of rounds changes. The key schedule it uses several constants delta0, delta1, delta2… which are generated from the square root of L E A which generates the round keys. In encryption it gets XOR with round keys based on the keys they have different number of rounds. The decryption is the reverse process of encryption. It is against the brute force attack since they do not how many number of rounds but it can be decrypted with the dictionary attack.

III. SMART SECURITY : PROPOSED ALGORITHM

A. Assumptions:

1. Choose any mod number as key. MN_k
2. Choose any sequence of alphabets. $s_\propto : \ \alpha \epsilon \{a, b, c, \dots z\}$
   - Can be a regular sequential pattern (RS_α) e.g.: a, b, c, d, e…
     $s_\propto \rightarrow RS_\alpha : \{a, b, c, d \dots\}$
   - Can be random alphabets (RNS_α) e.g.: b, c, d, a, g, h…
     $s_\propto \rightarrow RNS_\alpha : \{b, c, d, a, g, h \dots\}$
   - Can be another but this sequence must be same at both the sides.
3. Choose any kind of pattern.

B. Encryption:

Step 1. Enter the chosen sequence of alphabets. ($s_\propto$)
Step 2. Convert each letter to its corresponding dynamic integer value ($s_{\propto N}$) and
Compute $(S_{\alpha^N})^\lambda : (S_{\alpha^N})^2 / (S_{\alpha^N})^3 / (S_{\alpha^N})^4$
Step 3. Feed the mod number key. MN_k
Step 4. Divide each number individually with the key $\frac{(s_{\alpha N})\lambda}{MN_k}$
Step 5. Collect the quotient $(q)$and remainder(r).
Step 6. $(q) \longrightarrow \alpha \mid (r) \longrightarrow \alpha$

Convert them into corresponding equivalent letter and if it is not there then simply leave as it is. If there are two numbers side by side which cannot be decoded into corresponding letters then separate the two number by a sign that understands that it is a separation of two numbers that shouldn't be in alphabets that you have entered.

Step 7. It should follow the pattern something like first three reminders then followed by corresponding quotients.
Some patterns are:

- $\{\propto (q), r\}$ → Each letter of quotient followed by its remainder.
- $\{\propto (r), q\}$ → Each letter of remainder followed by its quotient.
- $\{q_1, q_2, q_3, r\}$ → Each first three quotients then followed by corresponding remainder.

C. Decryption:

8. Rightly authenticate the user.
9. $q \times k + r$: Multiply each quotient of its

number by key and add remainder.

10. $\sqrt{SN}$: Compute square root the number and convert it into equivalent form.

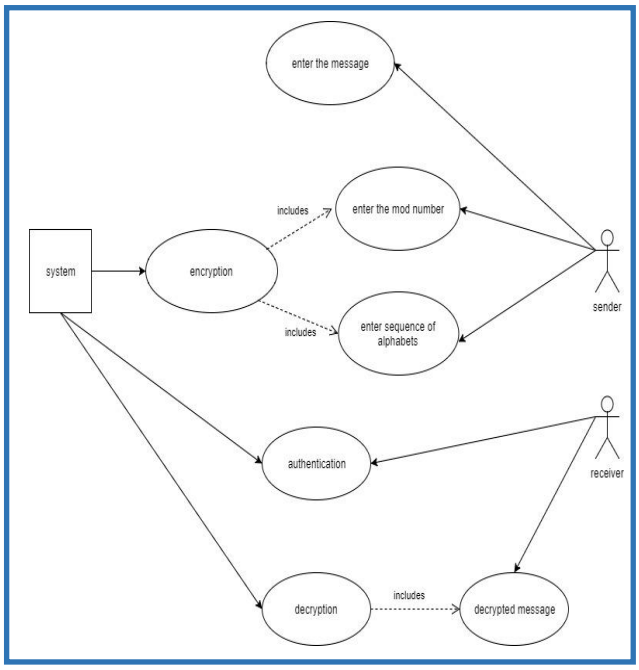


Fig. 1.Use case diagram for the proposed algorithm

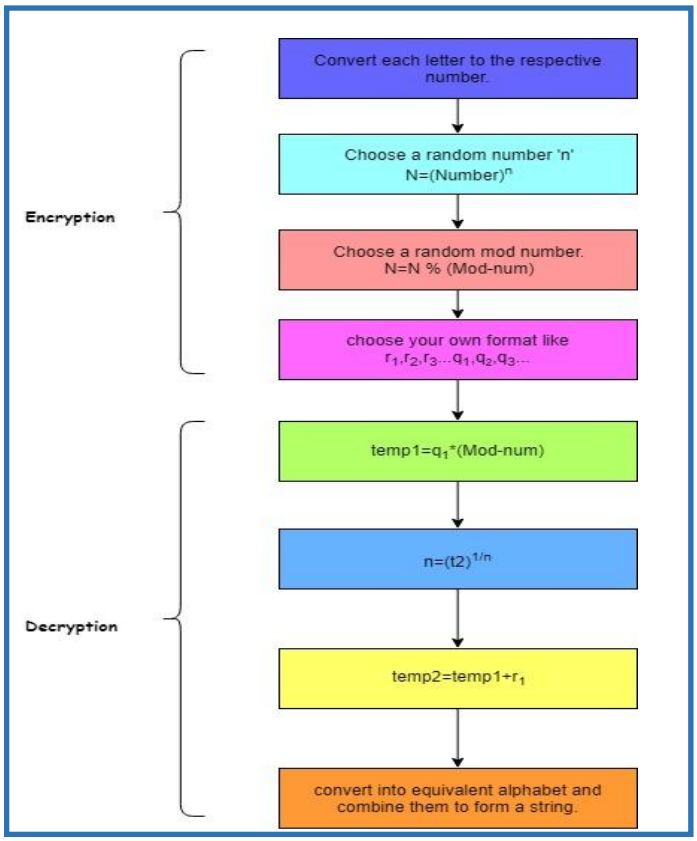


Fig. 2.Encryption and Decryption process

r →Remainder.

q →Quotient.

$SPLIT$ →break an array into each individual number.

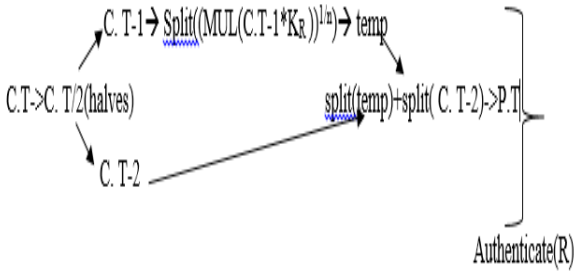
Some patterns are:

$r(all) + q\ (all)$

$q(1) + r(1) + q(2) + r(2) + q(3) + r(3) + \cdots$

$r(1) + q(1) + r(2) + q(2) + r(3) + q(3) + \cdots$

Decryption:



1. Enter the message.
2.Enter the alphabets in which do you want.
3.Repeat it.
   (i) Enter the character or digit.
   (ii)increment by one.
   (iii)if character repeated
      Make count-1
   (iv)append to the list.
      (iii)if character>62 end.

4.Repeat the following.
   (i)Enter the digit which matches the corresponding alphabet.
   (ii)increment by one.
      (iii)If any number repeats make count-1.
5.Enter the mod number.
6.Repeat the following:
   (i)Match the letters with the corresponding numbers.
   (ii)if match found append it to new list.
   (iii)increment by 1.
7.Repeat the following.
   (i)Square the number
   (ii)Divide the number by using mod number.
   (iii)append to the mod list to the remainder.
   (iv)append the quotient to list.
   (v)increment by 1.
8.Repeat the following:
   (i) number must be matched with equivalent alphabet
   (ii)append to the list of mod and quotient.
   (iii)increment by 1.
9. Send this cipher text to receiver.

At the Receiver end:

1. Connect with the sender.
2. Enter the password.
3. First break the array into halves.
4. Repeat the following:
   (i) Multiply quotient number and mod number and add remainder.
   (ii) Square root of number
   (iv) Append to the list
   (v) Increment by 1
5. Repeat the following:

(i) List each element convert to string and append to it.
(ii) Increment by 1.

I. Implementation and Result Analysis

Advantages
1. Sender can choose any key.
2. They may choose any pattern.
3. Sequence of characters is based on their wish.
4. They may apply any power of square, cube, fourth and so on to the number they have obtained when converting the alphabets to equivalent numbers.
5. Authentication must be done before decrypting the message at the receiver side.

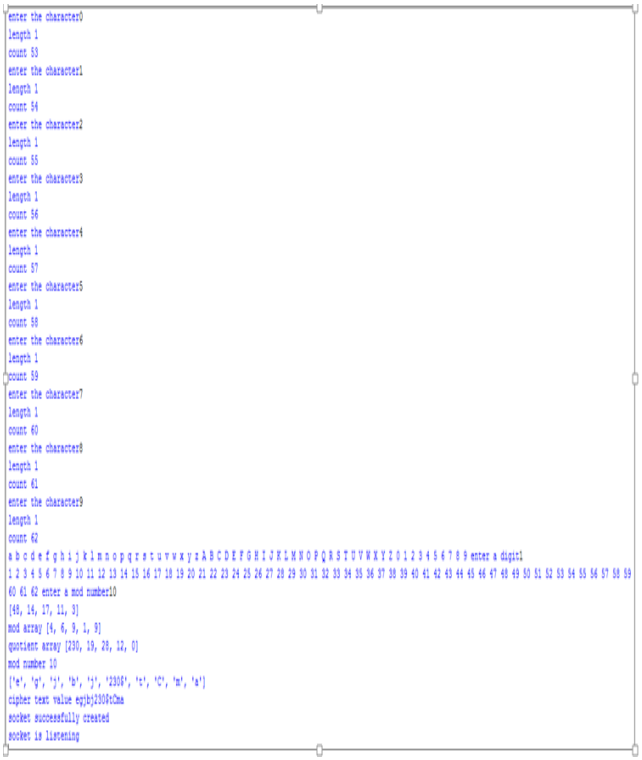


Fig. 3. Sender side Encryption process

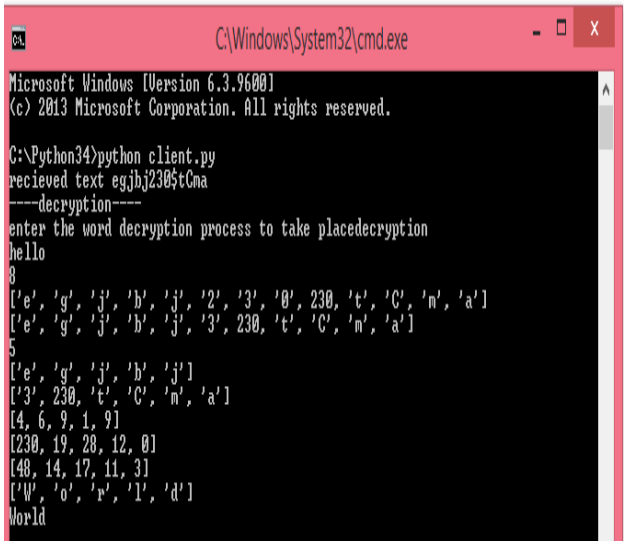


Fig. 4.Receiver side Decryption process

It is strong against the brute-force attack like we choose a random number, each letter is powered along with that number, and for each we choose a random mod number which is % with that number and it is made in some specified format like remainders followed by quotient or quotient followed by remainder. i.e., ((n)^n)*formats. It is strong against the Brute-Force attack, Dictionary Attack, Man-in-the middle attack and Impersonation.

To send or receive a message from the either ends it he/she must be authenticated user which by means of a strong password. We attach a Hash code along with the message by which we will come to know the message cannot be altered. We convert into equivalent numbers from each letter those number for each letter may not be in a order of 0,1,2.. for a,b,c,d… it may be 64,13,1…It is not a loose lipped system.

REFERENCES

1. N. Veeraragavan, S. S. Manikandasaran, L.Arockiam, Enhanced Encryption Algorithm for Protecting Users' Credentials in Public Cloud, IEEE, 2017, pp.1-6.
2. Diala Kahil, Maya Abou Lebdeh, Mohamad. Saad El Dine, Ahmad. El Rafhi, Innovation of a secured transmitter/ receiver chain by creating a new encryption algorithm, IEEE, 2017, pp.1-3.
3. Daniar Heri Kurniawan, Rinaldi Munir, Double Chaining Algorithm Secure Symmetric-key Encryption Algorithm, IEEE, 2016, pp.1-6
4. Nigam Sangwan, Combining Huffman text compression with new Double Encryption Algorithm, IEEE, 2013, pp.1-6
5. Hadia M. E Hennawy, Alaa E. Omar, Salah M. Kholaif, C. Design of LEA: Link Encryption Algorithm, IEEE, 2014, pp.82-91.
6. Hu Lihong, Research on Improved Algorithm Based on Encryption Algorithm of Hill, IEEE, vol.8, pp.659-662.
7. Azza Zayed Alshamsi; Ezedin Salem Barka, Implementation of Energy Efficient/Lightweight Encryption Algorithm for Wireless Body Area Networks, IEEE, 2017, pp.1-7.
8. Alexandr N. Gerasimov; Anna V. Epishkina; Konstantin G. Kogos, Research of Homomorphic Encryption Algorithms over Integers, IEEE, 2017, pp.398-403.
9. B r i n d h a .K, J e y a n t h i. N, Secured Document Sharing using Visual Cryptography to protect cloud data storage, Cybernetics and Information Technologies, vol. 15, no. 4, 2015, pp. 111–123..
10. B r i n d h a. K, J e y a n t h i.N, DOVC : Data Obfuscation Visual Cryptography to protect Cloud Storage, International Journal of Soft Computing, 11(6), 2016, 374-381..
11. J e y a n t h i, N., & I y e n g a r, N. C. S. N. An Entropy Based Approach to Detect and Distinguish DDoS Attacks from Flash Crowds in VoIP Networks,. IJ Network Security, 14(5), 2012, 257-269.
12. Rama Prabha K P, N. Jeyanthi, Intelligent Intrusion Detection Techniques for Secure Communications in Wireless Networks: A Survey, International Journal of Advanced Intelligence Paradigms, Special Issue on Green Mobile Computing for Energy-Efficient Next-Generation Wireless Communication, Vol.12, No.,1 -2, pp. 134 – 146.
13. B. Prabadevi, N. Jeyanthi, Nur Izura Udzir and Dhinaharan Nagamalai, Lattice Structural Analysis on Sniffing to Denial of Service Attacks, International Journal of Computer Networks & Communications, Vol.11, No.4, 2019, pp. 101-119.
14. K. P. Rama Prabha, N. Jeyanthi, A Trust and Fuzzy Cluster based Dynamic Secure Routing Algorithm for Mobile Ad-hoc Networks, Wireless Personal Communications, Springer, Vol. 98, No. 3, 2959 – 2974, 2018.
15. B. Prabadevi, N. Jeyanthi, A framework to mitigate ARP Sniffing attacks by Cache Poisoning, International Journal of Advanced Intelligence Paradigms, Vol.10, No.1/2, pp. 146-159, 2018.
16. Rama Prabha K P, N. Jeyanthi, Intelligent Intrusion Detection System Using Temporal Analysis and Type-2 Fuzzy Neural Classification, International Journal of Internet Technology and Secured Transactions, Vol.8, No.2, pp. 167 – 184, 2018.
17. B. Prabadevi, N. Jeyanthi, TSCBA-A Mitigation System for ARP Cache Poisoning Attacks, Cybernetics and Information Technologies, 2018, Vol.18, No.4, pp.75-93.
18. N. Jeyanthi, B. Prabadevi, A Review on Various Sniffing Attacks and its Mitigation Techniques, Indonesian Journal of Electrical Engineering and Computer Science, 2018, vol.12, No.3, 2018.

19. K P Rama Prabha, N. Jeyanthi. An intelligent neuro-genetic framework for effective intrusion detection, International Journal of Reasoning-based Intelligent Systems, Vol. 10.3-4, 224-232, 2018.
20. Jeyanthi N, R. Thandeeswaran, Gokulnath BV, A complete survey on blood pressure monitoring devices and applications, Biomedical Research, Vol. 29, No. 21, 2018, pp.3751-3765..

AUTHORS PROFILE

Dr. N. Jeyanthi received her Ph.D. degree in Cloud Security from VIT University, Vellore, Tamilnadu, India. She is an Associate Professor in VIT, Vellore for School of Information Science and Engineering. Her research work was funded by Department of Science and Technology, Govt. of India. She has authored and co-authored over 62 research publications in peer-reviewed reputed journals and 30 conference proceedings. Her entire publications have been cited over 265 times (Google Scholar). The latest Google h-index of his publications is 10 and i10 index is 13. Books and book chapters were also added to her research contribution. She has served as the program committee member of various international conferences and reviewer for various international journals. She has been honored by VIT as an active researcher for four consecutive years. Her current areas of interest include IoT, Cloud, and Big Data security.

Mr. G. Venkatesh received his M.Tech. with Software Engineering Specialization from VIT, Vellore, Tamilnadu, India. His current research interest is Information Systems and Security.

Dr. R. Thandeeswaran is an Associate Professor in the Department of Digital Communications, School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, India. He received his Ph.D. and M.Tech from the same University and B.E. in Computer Engineering from Madurai Kamaraj University. His research includes Cloud Security, Information Security and Internet of Things. He has authored and co-authored over 17 research publications in peer-reviewed reputed journals and about 20 conference proceedings. He received active researcher award from VIT. He delivered guest lectures and chaired the sessions at international conferences. He authored books and book chapters to his credit. He is life member of Computer Society of India.