Simple Text Based Encryption with Lossless Data Compression Technique

Yogesh Patil , Siddharth Gupta ,Debbrota Paul Chowdhury ,Sharad Patil

***Abstract*— As in this globalization the organization strives to be automated; networking also becomes an imperative. As the network grows it becomes critical to manage information secrecy, unfortunately this is what the situation is encountered with. This is not just enough to establish network security but it should be strengthened. Establishment of a system that protects a network is of utmost importance and this is the place where cryptography has proved its existence. But, most of the cryptography algorithms tend to generate the exact length cipher text as same as the plain text length which makes it more easier for attacker to carry brute force attack and even to guess in most of the cases ,so in this article we have exercised cryptography and Huffman in order to come out with the algorithm which makes use of the most optimal data compression technique and leave us with the compressed cipher which also provides higher level of security on which one can rely. The attacker has to start his task from getting the exact length of the text, guessing the text is a longish thing. Again to add one more layer to Encryption we have also used key concept in this, trying to decrypt it without key will make attacker to found himself beset by difficulties.**

***Index Terms*— Computer Security, Cryptography, Data compression Decryption, Encryption, Huffman Coding, Network security.**

I. INTRODUCTION

The communication system doesn't need to prove its importance in this era of information technology, however while dealing with digital communication systems; securing data is another major concern especially talking about the email communications or static data storage on general storage devices. Is there anybody who can guarantee that your data will be safe while transmitting and this all is to be done at no cost at last? Probably no!

The communication industry is one of the most expanding industry in the current era , the two factors are mainly responsible for this growth First is many research groups and research laboratories used to pass their billions of dollars data through this internet environment and second is moreover talking about the general internet users which uses this network for email communications or transferring data from this untrusted network ,the reason to title this as untrusted is there is no guarantee of secure and reliable transmission of your valuable data on this network ultimately there will be no accountable person whom you can blame afterword, but at the same time we can't stop sending our data through this untrusted medium. We have to keep working on the alternate solutions that may be applied so that reliability can be gained in this system that's what Network Security deals with.

This is not to be misnomer to define network security as information security. To be defined more precisely the Information security is easy but a giant term, no specific definition to be exhibited,

Information security = Confidentiality + Integrity + Availability+ Authentication.

The above definition is what that is used generally. We just can't imagine the network security to bring to light without confidentiality (misnomer with privacy), the internet is the open invitation for the various security threats and the communication security problems .The man-in-middle attack is one which is to be exercised more. As the name suggests the man-in-middle attack is one in which someone other than the authenticated senders and receivers (i.e. in between the sender and receiver) tries to read, change or replicate the information. Specifically read action is one on which our method is going to accent more. Again integrity is other important factor for consideration; which stands for receiving the information without disturbing it, that is to guarantee that what is received is exactly what was sent. Availability requires that computer system assets are available to authorized parties. The authentication is the term dealing that the information is to be retrieved only by the possessor.

Cryptography is what is known from ancient era, owing to the fact that this is the nature of human being to share their thoughts secretly with each other. What simply Cryptography does is that it converts the readable information into catastrophic information so that man-in-middle attack can be easily handled (i.e. Stranger would able to access the information but he won't be able to understand it).The cryptography comes in two flavor i.e. Encryption and Decryption. The Plain text to cipher text conversion is Encryption process and vice-versa that is cipher text to plain text is Decryption. Both the procedures are of equal importance and mistakes in single one would lead to the misconception and ultimately lead to wrong cipher or plain text.

Fig 1 .Encryption Process

Fig 2.Decryption Process

1.1. Objectives of Network Security

Manuscript received September 18, 2014.

Yogesh Patil, B Tech, Computer Science and Engineering, NMU , Jalgaon

Siddharth Gupta B Tech in Information Technology, College of Engineering ,Roorkee

Debbrota Paul Chowdhury, B Tech, Computer Science and Engineering from University Institute of Technology ,Burdwan

Sharad Patil: Ph.D. in Computer Networks &Security Systems (Bharati Vidyapeeth ,Pune (MS) India

A. Ensure that any message sent arrives at the Proper destination.

B. Ensure that any message received was in fact the one that was sent. (Nothing added or deleted)

C. Control access to your network and all its related parts. (This means terminals, switches, modems, gateways, bridges, routers, even printers).

D. Protect information in-transit, from being seen, altered, or removed by an unauthorized person or device.

E. Any breaches of security that occur on the network should be revealed, reported and receive the appropriate response. Have a recovery plan, should both your primary and backup communications avenues fail.

II. BACKGROUND

According to our analysis we have observed that several popular techniques tends for data encryption tends to similarity that it take n bit plain text and generate the same length i.e. n bit cipher text for receiver. This would work as a helping hand for attacker .There are so many parameters that attacker must know about data if he wishes to recover the plain text and the text length is one of the important factor that would lead to malnourished encryption scheme. If the data length is known to the attacker then he can use permutation and combination approach to identify the words and this won't take more than seconds if done properly and handled by high processing machines.

Fig 3.Conventional Cryptography

Andrew S. Tanenbaum. Quotes that, "If the cipher text generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext an encryption scheme is unconditionally secure, doesn't matter how much cipher text is available." That is, no matter how much time an opponent has, it is impossible for him or her to decrypt the cipher text, simply because the required information is not there"[6]

Therefore, all that the users of an encryption algorithm can strive for is an algorithm that meets one or both of the following criteria:

1.The value of the encrypted information is much more less than the value required for breaking the Cipher.

2.The time required to break the cipher is much more that information retrieved from is not at all useful.

So while dealing with this issue we mainly focus on disturbing the text size so that the attacker would always have to play lots of ransom even to simply guess the exact length of plaintext. After getting the exact word length only he can propound the things. The things go even worst if anyhow he guesses the length wrongly even for single character, and he will court to disaster by continuing with that wrong prediction. While disturbing text size increasing the text size would also have been a good approach but we prefer decreasing the text size as our ultimate goal is to gain reliability in the network by introducing as less traffic as possible.

Data compression was always in the public eye since earlier days, and Huffman coding is one of the marvels having the optimal compression result. Huffman algorithm is one that meets all these expectations. We have use Huffman algorithm which works on variable length prefix code for receiver and is based on construction of binary tree. In Huffman coding our prime focus is on frequency of occurrence of data item (text in bit). Each character (combination of bits) situated at leaf, by this we use lower number of bits to encode the data which ultimately results in compression of data at encoding state. In order to avoid several ambiguities in code we use prefix rule in that we make sure that different character used are not prefix of each other prefix rule simplifies the decoding of text at receiver side and it always achieve the optimal data compression among any character code. Hence based on above technique Huffman code compress data very efficiently saving of nearly 20% to 90% are typical depending on the characteristics of the data being compressed.

Moreover the ASCII is the thing which was not under much more focus for security practitioners most off the general cryptographic algorithms works on the assignment of letters to letters none of them is to be designed to work in a manner to assign letter to symbol or symbols to letter ultimately making attacker's life difficult. The ASCII is the worldwide accepted form of coding so user don't need to refer different code books to encrypt or decrypt the message, again ASCII is enriched set of symbols which gives a rare opportunity for the attacker to come nearby ultimately closing loopholes in the algorithm.

Key acts as the Lucifer for the attacker, so we just can't imagine encryption algorithm to be truthful without key. To tightened up the security we have make some craftiness by providing two level of security (two Key levels) somewhat similar concept to public cryptography which needs to exchange the keys with each other. Key distribution is the secure concern that is to be handled securely. From that of the two keys one is to be generated by sender randomly depending on the text being encrypted while other key is being generated by user depending on his own choice.

So how does it work exactly? ; The mix of Huffman coding and ASCII value and then Key generation these all are the basic steps that we have to carry out in this technique. Huffman coding is what that very common and very well known, Huffman gives the optimal is coding among all the other available algorithms.

Our technique start by encoding the text by using Huffman coding, for encoding we have used one of the reference tree that is to be generated from the continuous sequence of letters from A to Z. The tree generated is optimal one and it follows following creation procedure.

1. In Huffman coding we arrange the given frequencies in ascending order, our ultimate goal is to build the tree corresponding to the optimal code in a bottom up manner.
2. Begin with a set of n leaves and performing a sequence of (n-1) merging operations to create final tree.
3. Use min priority queue in which elements are arranged according to corresponding frequency attributes to identify 2 least frequency attribute to merge together.
4. When we merge two objects together their sum is stored in new object which is the sum of frequencies of 2 objects merged.
5. Similarly all the elements present in heap are merged together until we reach at root node and form a tree.
6. After forming a tree we assign 0 to left child and 1 to right child from root of the tree to bottom of the tree i.e. leaf.
7. Finally in order to assign code to each character at leaf node we retrieve combination of bits from root to leaf node of a particular character.
8. Apply same for all the characters at leaf node we had assigned a unique code to every character present at leaf node.

The final tree presents optimal prefix code.

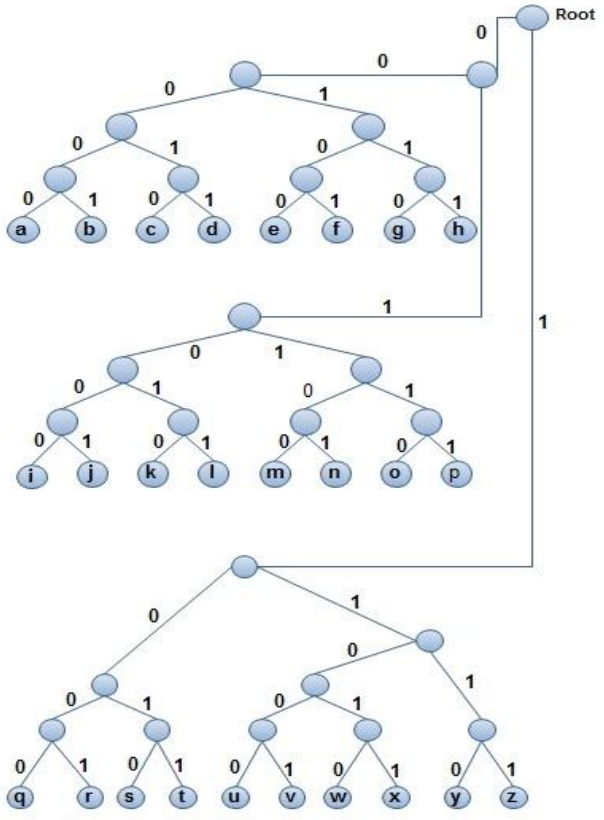

Fig 4.Optimal Huffman Tree[8]

So after this procedure next task is to trace the letter from leaf node to the root along with marking tracing path from leaf node to root i.e. combination of 0's & 1's.The bit stream generated is what that is to be worked out, Normally on byte oriented machine for n bits it would take (8*n) bits to decode it into binary but by using Huffman we can decode this by using only 60% to 70% bits among them , ultimately by hook or by crook we have to save time required to encrypt the message and need to increase complexity of cipher text to break attacker's spirit.

Next task is to evaluate the length of bit stream this is what easier task than other, as you may encounter with readymade functions into general programming languages and this won't take much more time ,after calculation name that length as L ,Now for our future use as we will be needed to consider the binary bits into group of 8 bits and have to take some arrangement also for the extra bits that would encounter after grouping them into multiple of 8 bits , so to determine number of extra bits we take L%8 that will generate reminder ultimately will leave us with an idea that how many bits we can use to generate data part. Now suppose it leaves us with the value m_1 then we will divide it into two parts m_2 and m_3 {for even numbers it would be straight forward to divide it into two parts , for odd numbers take it ceiling value i.e. Let we have to divide 7 then a1=$\left\lceil \frac{7}{2} \right\rceil$=4 and a2=$\left\lceil \frac{7}{2} \right\rceil$=3 }.

Now take first m_2 bits of bit stream and concatenate it with the last m_3 bits of bit stream and now convert that into decimal value which would act as k_1 for our algorithm and then combine it with the receiver's key i.e.k_2 ultimately to form final K. This will provide us multiple layers of security as someone trying to guess the text or trying to cultivate brute force attack then he would have to first break the key and append some part to start and some to end of text so that he will be in position to have compressed version of text which is again to be encoded by using Huffman Coding and at last he will have actual text any one step differentiation even one bit of mismatch will do alchemy and will leave attacker in the lurch.

Next task is to remove first m_2 bits from the bit stream and also the last m_3 bits from bit stream whatever we have left with is the thing which will be now onwards considered as data stream. ASCII value is collection of variety of symbols but was put into lumber and not much used in the field of cryptography, as ASCII value is combination of 8 binary bits so we have to group up our bits and divide the bit stream into the bytes i.e. group of 8 bits. After this simply assign corresponding symbol or character to the obtained number and concatenate all these into one array of symbols and this is what we are eagerly waiting for-The final cipher text.

The generation of cipher text is just half done yet the next thing is to form the final frame, According to user convenience he can use different ways to attach the decimal key value i.e. K to cipher generated.
Let final key is K and cipher generated is C then,

1. Simply attach K to C i.e. (K+C).
2. Attach C to K i.e.(C+K).
3. Use Fraction approach i.e. divides K into two equal parts i.e.c_1 & c_2 and then attach c_1 at start and c_2 at end.
4. Divide key by using ceiling function into 3 equal part attach one at start one at end and remaining exactly at the middle of the text.

Final method is more sophisticated but this also needs to be considered that it ultimately depends on the importance of message that is being transmitted , this shouldn't be the case that the cost of encrypting message is much more than the actual importance of message so key selection and distribution is totally dependent on the choice of users. Here we have considered simple approach in all cases in order to be understood by everyone else.

2.1 Key Distribution:

For this scheme to work very well, the two parties i.e. sender and receiver must exchange their keys and at the same time the security for key itself is also very important , Not eventually every time but the frequent changes of key must occur in order to prevent its impact i.e. it will limit the data compromised. Therefore, the strength of any cryptographic system rests with the key distribution technique, a term that refers to the means of delivering a key to two parties who wish to exchange data without allowing others to see the key.[9]

Key distribution can be achieved in a number of ways.

For two parties A and B

1. A key could be selected by A and physically delivered to B.[9]

2. A third party could select the key and physically deliver it to A and B. [9]

3. If A and B have previously and recently used a key, one party could transmit the new key to the other, encrypted using the old key. [9]

4. If A and B each have an encrypted connection to a third party C, C could generate the key from an automated random key generator and deliver a key on the encrypted links to A and B. [9]

Options 1 and 2 are more reliable in the sense as it includes the physical meeting of two parties but there are some issues that must be considered about human behavior that humans may easily betray and the other thing is the physical distance between two parties may also lead to anomaly that key exchange would itself cost more instead of actual message cost. However automated random key generators seems to be best option providing wide range of keys to deal with and also it would increase the complexity of the cipher being generated. To provide keys for end-to-end encryption, option 4 is preferable and in our scheme we would like to include the same.

III. Algorithms

A. *Encryption*

Step-1.Initially takes plain text.

Step-2.Apply Huffman algorithm on plain text to obtain corresponding bit streamB_1.

Step-3.Evaluate the length of bit string.

Step-4 Calculate length of bit stream *Let* Land then take m_1= (L%8), m_2= $\left\lceil \frac{m1}{2} \right\rceil$ and m_3=$\left\lfloor \frac{m1}{2} \right\rfloor$.

Step-5 Take first m_2 bits from B and combine it with last m_3 bits from B & convert it into decimal which will form our first key k_1.

Step-6 Take another key k_2 i.e. being provided by receiver, Add k_1 and k_2 and form final key K.

Step-7 Remove first m_2 bits from B_1 and last m_3 bits from B_1 and name it as B .

Step-8 Split the B into multiple of 8 bits.

Step-9 For each byte in B writes the corresponding decimal value, and for each decimal value write corresponding ASCII value, This is encrypted data.

Step-10The final thing to send will be (K+B).

B. *Encryption*

Step-1Take cipher text T and divide into K & B.

Step-2 Use receiver's key i.e.k_2 and subtract it from K and convert it into binary this will be our k_1.

Step-3 Calculate length of k_1 and name it as L, Divide L by 2 then, L_1= $\left\lceil \frac{L}{2} \right\rceil$ and L_2=$\left\lfloor \frac{L}{2} \right\rfloor$.

Step-4 Read Symbol from B and convert it into its ASCII code value and converts that ASCII value into binary and construct new bit stream let B_1.

Step-5 Now append L_1 at start of B_1 and L_2 at end of B_1 and obtain final bit stream and name it as B_2.

Step-6 Decode our new bit stream B_2 by using Huffman Decoding.

This is the final message.

IV. Implementation

A. *Sender Side*

Let as a plain text we have "**ganapati bappa moraya**"
So let first encode it by using Huffman code,

g=0110 , a=0000 , n=1101 , p=1111 , t=1011 , i=1000 , b=0001 , m=1100 , o=1110 , r=1001 , y=1110

After Hoffman coding the bit stream generated is:

01100000110100001111000010111000000100001
11111110000110011101001000011100000

Total bits = 76

Data length = 76 mod 8 = 4

Then calculate m_2 = $\left\lceil \frac{4}{2} \right\rceil$ = 2 and m_3 = $\left\lfloor \frac{4}{2} \right\rfloor$ = 2.

First two bits of bit stream (m_2) = 01

Last two bits of bit streams (m_3) = 00

Intermediate Key (k_1) = m_2m_3 = **0100** ⇨4

Let receiver's key (k_2) = 1001 ⇨ 9

K =k_1+k_2 = 4+9
0100
+ 1001

1101 (13) (Perform Binary Addition.)

Data Stream	**10000011**	01000011	**11000010**
Decimal	131	67	194
ASCII	**Â**	**C**	┬

Data Stream	11100000	**01000011**	11111100
Decimal	224	67	252
ASCII	**Ó**	**C**	³

Data Stream	**00110011**	10100100	**00111000**
Decimal	51	164	56
ASCII	**3**	**ñ**	**8**

Sending message=> "13**âC┬ÓC³3ñ8**"

B. Receiver Side

For Receiver the received text would be "13**âC┬ÓC³3ñ8**"

Here the 13 is key and **âC┬ÓC³3ñ8** is data part ,
first split data and key part,

Obtain original key by subtracting receiver's key from the obtained key, as

Key = 13-9 1101
-1001

0100 (4).

K_1 length = 4 bits.

Then calculate $L_1 = \left\lceil \frac{4}{2} \right\rceil = 2$ and $L_2 = \left\lfloor \frac{4}{2} \right\rfloor = 2$.

First two bits (Key_1) = 01

Last two bits (Key_2) = 00

Now obtain the received data stream,

Data = âC┬ÓC³3ñ8

ASCII	**Â**	**C**	┬
Decimal	131	67	194
Data Stream	**10000011**	01000011	**11000010**

ASCII	**Ó**	**C**	³
Decimal	224	67	252

Data Stream	11100000	**01000011**	11111100
ASCII	**3**	**Ñ**	**8**
Decimal	51	164	56
Data Stream	**00110011**	10100100	**00111000**

Data Stream
10000011 01000011 **11000010** 11100000 **01000011** 11111100 **00110011** 10100100 **00111000**

Append key_1 at the start of the data bits and key_2 at the end of take data bit stream.

011000001101000011110000101110000001000011111111
000011001110100100001110000**0**

At last decode with the Huffman coding and you will end up with the message ; "**ganapati bappa moraya**" , which is the original message.

V. Advantages

[1]This makes it complicated for attacker to guess the text as attacker doesn't even know the length of text i.e. to be guessed.
[2] This technique is more beneficial, since in which we gain probably most optimal result after applying compression on data.
[3] This technique gets even stronger as we increased the text size and/or key size.
[4]Huffman coding implements lossless compression so ultimately no data loss will occur.
[5]Can be introduced as the new security technique useful for the low bandwidth network as it reduces the length of text.
[6]Though it is concept of key exchange due to compression technique the data transfer very is fast.

VI. Disadvantages

[1]The algorithm is text based, this will not work for symbols or numbers.
[2]This is not efficient for small message as overhead will be there to apply procedure; longer the message more efficient the scheme will be.
[3]ASCII table and tree must be maintained for each conversion.

VII. Conclusion

In this paper we have worked on encryption with data compression, for which we have used the most optimal Huffman algorithm which guarantees to have lossless data

compression. This algorithm is simple and easy to implement from user point of view but it would leave the attacker in the lurch. One of the major advantages of this algorithm is that attacker would never come to know even about the original size of the text; ultimately it prevents guessing and brute force attack. The essence of this algorithm is that the text can be retrieved only if one has exact combination of both key and data.

REFERENCES

[1] http//www.consentry.com/the-importance-of- network-security/
[2]"Effective Secure Encryption Scheme [One Time Pad] Using Complement Approach" International Journal of Computer science and Communication Volume 1 issue 1 January June 2010.
[3]http//www.en.wikipedia.org/wiki/public_key_cryptography
[4]Sharad Patil ,Dr.Ajay Kumar "Modified One Time Pad Data Security Scheme: Random Key Generation Approach"International Journal of Computer and Security Volume 3 issue 2 March/April 2009 Malaysia.
[5] Ritter, Terry 1991. The Efficient Generation of Cryptographic Confusion Sequences ,Cryptologia.
[6] Andrew S. Tanenbaum."Computer Networks" Fourth Edition.
[7] Michael T. Goodrich & Roberto Tamassia, "Algorithms Design", Second Edition.
[8] Thomas H. Coreman , Charles E. Leiserson , Ronald L. Rivest & Clifford stein ,"Introduction to algorithm" ,Third Edition
[9] William Stallings ,"Data and Computer Communications" ,Fifth Edition

Yogesh Patil received his B Tech from NMU ,Jalgaon in Computer Science and Engineering and currently pursuing his M Tech from International Institute of Information Technology ,Bhubaneswar ,Orissa (India) in Computer Engineering. His area of interest is Cryptography ,Machine Learning ,Robotics and Data Mining.

Siddharth Gupta received his B Tech from College of Engineering ,Roorkee in Information Technology and currently pursuing his M Tech from International Institute of Information Technology ,Bhubaneswar ,Orissa (India) in Computer Engineering. His area of interest is Cryptography ,Cloud Computing ,Parallel Computing.

Debbrota Paul Chowdhury received his B Tech from University Institute of Technology ,Burdwan in Computer Science and Engineering and currently pursuing his M Tech from International Institute of Information Technology ,Bhubaneswar ,Orissa (India) in Computer Engineering. His area of interest is Cryptography ,Cloud Computing ,Data Mining.

Sharad Patil: received his Ph.D. in Computer Networks &Security Systems (Bharati Vidyapeeth ,Pune (MS) India ,M Sc in Computer Science from Babasahed Ambedkar Marathwada University, Aurangabad in 1990 . Presently he is working as a Lecturer & HOD in Department of Computer Science in ACS College ,Navapur(MS).He has presented few papers in national, international conferences and journals. His area of interest is Computer Networks and Security systems.