Volume 3, Issue 12, December 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Dual Phase Mechanism for Data Transmission to Provide Compression and Security

Ch.Demudu Naidu[1], G.Pandit Samuel[2] , M.V.kishore[3], N.Aditya Sundar[4]

Assistant Professor

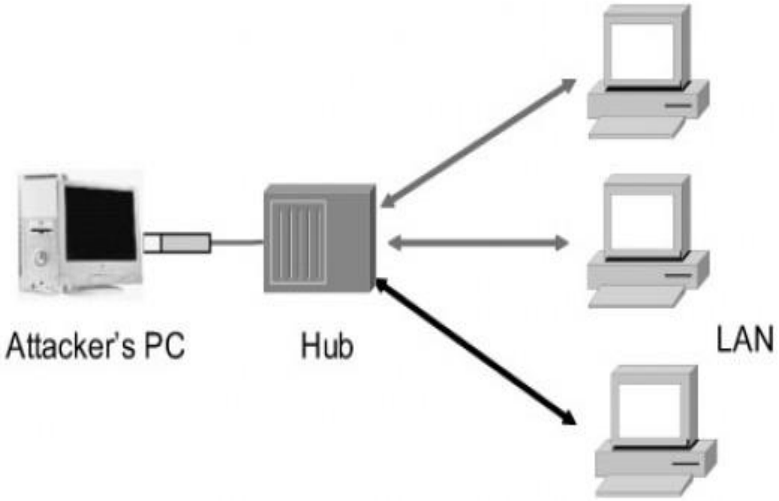
ANITS,Vizag, India

Abstract- Now a day's internet traffic has been increased rapidly so, the requirement of securing the data has been increased where the eyes of the intruders are on gathering the information and also misusing the gathered information and also due to this rapid growth of internet traffic the size of the data which is been transferred through the internet should be compressed, for this compression we use Mod-encoder technique. This technique encodes the given plain text into encoded and compressed data but the integrity of the message can be easily lost if the intruder can find out the base value on which the encoding depends on , for this case, we propose a mechanism by further encrypting the encoded data using encryption algorithm.

Keywords: Data Compression, Number Representation, Encryption, decryption, Modulo Arithmetic.

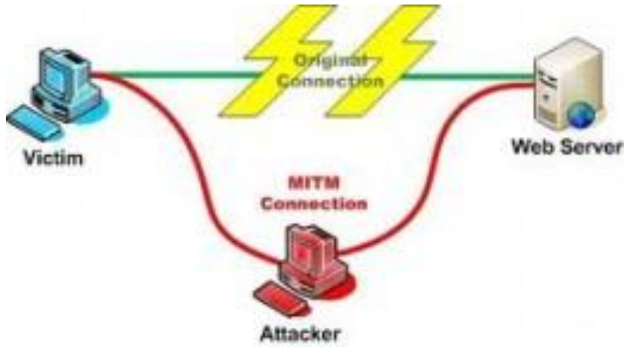
I. INTRODUCTION

The Internet has become one of the most important components of the world economy. It is a catalyst for business activities and effective governance, a major driver of scientific research and development efforts, an amazing source of growth and human progress. But this network, as all networks in the global economy, is also vulnerable to abuse and criminal behavior and it needs a security framework to operate for the benefit of all. Spear Phishing, Targeted Attacks and Data Breach Trends, man in the middle attack, Spoofing, Sniffing etc. for example in Packet sniffing the interception of data packets traversing a network. A sniffer program works at the Ethernet layer in combination with network interface cards (NIC) to capture all traffic traveling to and from internet host site. Further, if any of the Ethernet NIC cards are in promiscuous mode, the sniffer program will pick up all communication packets floating by anywhere near the internet host site. A sniffer placed on any backbone device, inter-network link or network aggregation point will therefore be able to monitor a whole lot of traffic. Most of packet sniffers are passive and they listen all data link layer frames passing by the device's network interface. There are dozens of freely available packet sniffer programs on the internet. The more sophisticated ones allow more active intrusion.



Hijacking (man-in-the-middle attack) takes advantage of a weakness in the TCP/IP protocol stack, and the way headers are constructed. Hijacking occurs when someone between you and the person with whom you are communicating is actively monitoring, capturing, and controlling your communication transparently. For example, the attacker can re-route a data exchange. When computers are communicating at low levels of the network layer, the computers might not be able to determine with whom they are exchanging data.

Man-in-middle attacks are like someone assuming your identity in order to read your message. The person on the other end might believe it is you, because the attacker might be actively replying as you, to keep the exchange going and gain more information.



In the above cases in hijacking and Man-in-middle attack we observe that the data can be easily tampered by the intruder who can harm the integrity of data or results in loss of data. For this case we would encrypt the data which is encoded where encoding is used for compressing data and adding security to data by changing the message format from plain text to encoded text. Encrypting this encoded data will help in providing double security to the data where the hacker is given complexity where he should be able to compromise both encryption and encoding which takes an increased period of time

II COMPRESSION MECHANISM IMPLEMENTATION

The existing MOD-ENCODER algorithm uses any standard encryption technique which incorporates lossless compression in order to cater to the needs of low bandwidth and data security. As mentioned above, let L be a defined language over the alphabet set. For example, L be English and be $\{A,B, \ldots Z,a, b, \ldots, z\}$. Let the letters of be indexed by a bi-junction function I that maps a letter to an integer i, where $1 \leq i \leq ||$. Δ is a constant, called modulus constant. Let the data string M be $\{m_1,m_2,m_3, \ldots, m_n\} \in$. Performing modulus operation on every $I(mi)$ by Δ, sequentially yields remainder set R as $\{r_1, r_2, r_3, \ldots, r_n\}$ and quotient set Q as $\{q_1, q_2, q_3, \ldots, q_n\}$. The elements in R have the values between $[0,\Delta - 1]$. So, we consider the elements in R to be a vector of numbers in base R. Each r_i takes log2 R bits for binary representation. If the message is of n characters, the number of bits required to represent the vector R is $n \times \log2 R$. The quotient set is represented in a different way. Let $B =[\Delta] + 1$ be another parameter called *Base-value*. The elements of Q will have values within $[0,B-1]$. Consider Q as a number in *base B*, i.e. $(q_1, q_2, \ldots, q_n)B$. Convert the number to a higher base. It is obvious that a higher base representation would reduce the digits in the number. If B is less than 10, the we convert Q_B to a Q_{10} number.

The MOD-ENCODER Encoding Algorithm can be stated as :

```
1) Input: $M$
2) $n = |M|$ , i.e. length of $M$
3) $Z = n \times bit\ size$ i.e. bit size is the number of bits require to represent each char.
4) for $i = 1$ to $n$
    4.1) Read $mi$ the $i^{th}$ character from $M$.
    4.2) Find $R$: $R[i] = I(mi)\%\Delta$
    4.3) Find $Q$: $Q[i] = I(mi)/\Delta$
5) Representation of $R$: for $i = 1$ to $n$
    5.1) Represent $R[i]$ in Base $\Delta$.
6) Representation of $Q$: Interpret $Q$ as Base $B$ number and convert it to Base 10
```

The vector R is communicated through open channel, whereas Q is encrypted to a cipher Qc using any standard cryptographic technique and communicated to the receiver to ensure the confidentiality of the message M. By doing so, the overhead of encryption is reduced as we encrypt only tuple Q, rather than the whole message M. The receiver on receiving R and Qc, decrypts Qc to Q and decodes the message from the bi-tuple$< R,Q >$.

The MOD-ENCODER Decoding Algorithm as follows:

```
1) Input: Bi-tuple $< R,Q >$
2) Convert $Q$ from Base 10 to Base B: Let $QB = (q_1, q_2, \ldots, q_n)$ be the representation in Base $B$
3) Interpret $R$ as a vector of Base $\Delta$ number: for $1 \leq i \leq n$
    3.1) $i = q_i \times\Delta+r_i$ where $q_i$ the $i^{th}$ digit of $QB$ $r_i$ the $i^{th}$ element of $R$.
    3.2) $mi = I-1(i)$
4) $M = (m_1,m_2, \ldots, m_n)$
```

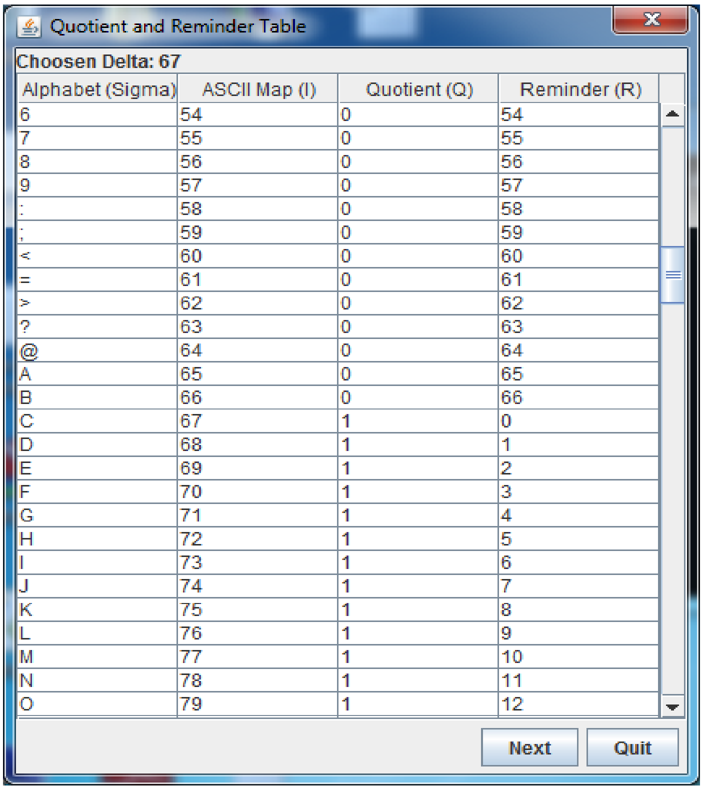
2.2 Compression mechanism

As mentioned above, the encrypted message M is a bi-tuple $< Q,R >$ of quotient and remainder. So, the size of the encrypted message can be obtained by calculating the number of bits required to represent Q and R. Let X be the total number of bits required to represent R and is given by $X = n* \log2 \Delta$

where n is the length of the message and $\log2\Delta$ is the number of bits required to represent each remainder. The quotient Q is looked upon as a *Base B* number. Each qi needs log2 B bits for its representation. As we know, a number in *Base 10* requires lesser number of bits than its equivalent in another *Base B* for representation, considering $B < 10$. Therefore, to lessen the number of bits required to represent Q, we first convert it into a *Base 10* number, say T. So, the number of bits required to represent Q is given as $Y = log2\ T$. Hence, the total number of bits needed for representation of the encrypted message M can be given by $X + Y$. Considering, a 7-bit representation for every character as in ASCII (or 8-bit in Unicode), $Z = n* 7$ is the total number of bits required for plain text message. We can observe that Z >(X+Y). So this reduction in bits provides us with the desired compression and the Compression Ratio $C.R.$ is given by $C.R. = (X+Y) /Z$

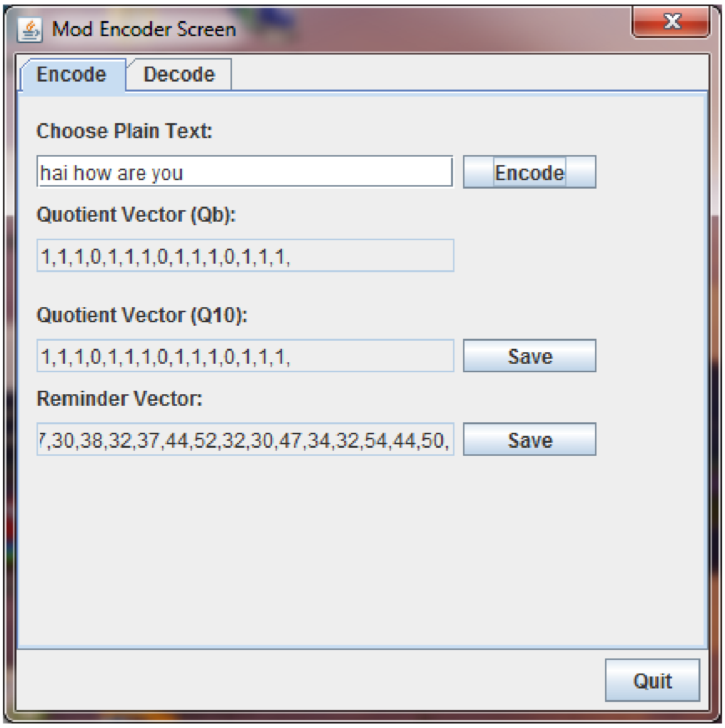
The above said system design was proposed in a paper "A Lossless MOD-ENCODER towards a Secure Communication" [1] was implemented using java in an user interface module where the details are shown as fallows

TABLE I
QUOTIENT AND REMAINDER FOR DIFFERENT ASCII CHARACTERS OF Δ =4



Choosen Delta: 67

Alphabet (Sigma)	ASCII Map (I)	Quotient (Q)	Reminder (R)
6	54	0	54
7	55	0	55
8	56	0	56
9	57	0	57
:	58	0	58
;	59	0	59
<	60	0	60
=	61	0	61
>	62	0	62
?	63	0	63
@	64	0	64
A	65	0	65
B	66	0	66
C	67	1	0
D	68	1	1
E	69	1	2
F	70	1	3
G	71	1	4
H	72	1	5
I	73	1	6
J	74	1	7
K	75	1	8
L	76	1	9
M	77	1	10
N	78	1	11
O	79	1	12



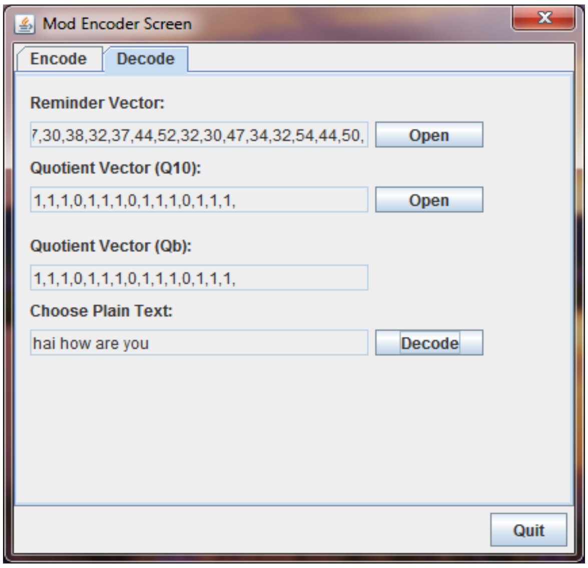
Choose Plain Text: hai how are you

Quotient Vector (Qb): 1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,

Quotient Vector (Q10): 1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,

Reminder Vector: 7,30,38,32,37,44,52,32,30,47,34,32,54,44,50,

FIG1: THE MOD ENCODER SCREEN REPRESENTING THE ENCODING SCHEME



Reminder Vector: 7,30,38,32,37,44,52,32,30,47,34,32,54,44,50,

Quotient Vector (Q10): 1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,

Quotient Vector (Qb): 1,1,1,0,1,1,1,0,1,1,1,0,1,1,1,

Choose Plain Text: hai how are you

FIG 2: THE MOD ENCODER SCREEN REPRESENTING THE ENCODING SCHEME

The above Fig1 and Fig2 shows the encoding and decoding mechanism where the message is encoded into base 10 quotient and remainder vector which are in encoded forms

III SECURITY ANALYSIS

The transmission of tuple *Q* to the receiver via a secure channel ensures the confidentiality of the message. However, tuple *R* can be communicated through open channel. So, *R* becomes easily accessible. And if the value of R is known it becomes very easy for the intruder to decode the data ,in this manner this method creates a bleach for security of data and a single point of failure in security for this reason we propose an architecture in which the encoded data is encrypted using Advanced encryption stand algorithm which helps us in increasing the security .

IV PROPOSED ARCHETECTURE

In this proposed system we propose that the encoded data that has been compressed using Mod-encoding is encrypted using AES algorithm which is a strong symmetric key algorithm where the algorithm is presented below

High-level description of the algorithm

- Key Expansion—round keys are derived from the cipher key using Rijndael's key schedule
- Initial Round

1. AddRoundKey—each byte of the state is combined with the round key using bitwise xor

- Rounds

1. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
2. Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
3. Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column
4. AddRoundKey

- Final Round (no Mix Columns)

1. Sub Bytes
2. Shift Rows
3. AddRoundKey

The process flow of encoded encryption data is shown using the following process flow diagram

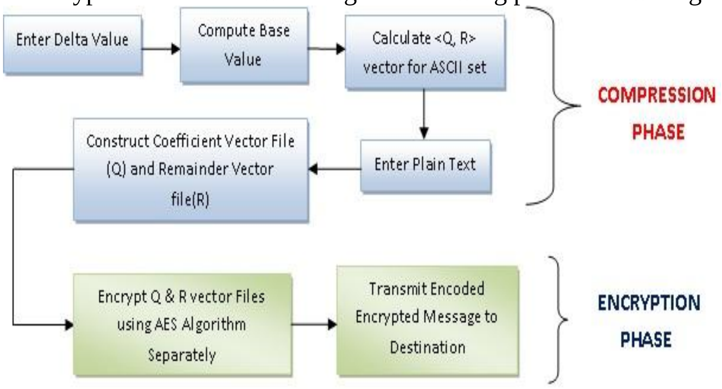

Enter Delta Value
Compute Base Value
Calculate <Q, R> vector for ASCII set
Enter Plain Text
Construct Coefficient Vector File (Q) and Remainder Vector file(R)
COMPRESSION PHASE
Encrypt Q & R vector Files using AES Algorithm Separately
Transmit Encoded Encrypted Message to Destination
ENCRYPTION PHASE

Fig 3: process flow for encoded encryption

In the Fig3 we have two phases compression phase and encryption phase. In compression phase we use Mod-encoding technique which was already discussed above in the compression implementation section [1] . As the process flow states, we enter the delta value which helps to calculate the base value by using the formula (256/delta)+1, using the obtained base value we generate <Q,R> vector for ASCII set. Using this <Q,R> vector we encode the plain text into encoded data which ensures our compression. In the encryption phase we encrypt the obtained Quotient (Q) vector compressed file as well as Reminder(R) compressed file using AES algorithm which was already discussed above. Hence the compressed(encoded), encrypted file is sent to the Destination which ensures our security while passing the data through an public channel , the data received at the destination is decrypted decoded same as the above said components in the backward direction as shown in the Fig3

V SECURITY PROSPECTIVE

When the security issues of existing system with encoding [1] is compared with the proposed system, we identify level of security increase in the proposed system which gives us integrity to the compressed data which adds us double security to the data which is passed through an unsecured channel

VI FUTURE ENHANCEMENT

The proposed architecture describes only about the text contents or a file, but this can be further enhanced to image, video etc.

REFERENCES

[1] G. Praveen Kumar, Biswas Parajuli, Arjun Kumar Murmu, Prasenjit Choudhury and Jaydeep Howlader " A Lossless MOD-ENCODER Towards a Secure Communication " ,978-0-7695-3975-1/10 $25.00 © 2010 IEEE DOI 10.1109/ITC.2010.89

[2] William C. Barker, *"Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher"*, National Institute of Standards and Technology, NIST Special Publication 800-67, 2008.

[3] *"Announcing the ADVANCED ENCRYPTION STANDARD (AES)"*,Federal Information Processing Standards Publication 197, Nov. 2001

[4] R.L. Rivest, *"The RC5 encryption algorithm"*, Proceedings of the 1994 Leuven Workshop on Fast Software Encryption, pp. 86-96, Springer-Verlag, 1995.

[5] Ron Rivest, Adi Shamir and Len Adleman, *"A method for obtaining Digital Signatures and Public Key Cryptosystems"*, Communications of the ACM, pp 120-126, Feb. 1978.

[6] Taher ElGamal, *"A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms"*, IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp.469472 or CRYPTO 84, pp.1018, Springer-Verlag.

[7] Elliptic Curve Cryptography, Certicom Research, 2000

[8] Huffman's original article: D.A. Huffman, *"A Method for the Construction of Minimum-Redundancy Codes"*, Proceedings of the I.R.E., Sep. 1952, pp.10981102

[9] Amit Jain, Ravindra Patel, *"An Efficient Compression Algorithm (ECA) for Text Data"*, icsps, pp.762-765, 2009 International Conference on Signal Processing Systems, 2009

[10] Farina, A.; Navarro, G.; Parama, J.R., *"Word-Based Statistical Compressors as Natural Language Compression Boosters"*, Data Compression Conference 2008, pp. 162 - 171, Mar. 2008