iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** VI **Month of publication:** June 2018

DOI: http://doi.org/10.22214/ijraset.2018.6257

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VI, June 2018- Available at www.ijraset.com

Bidirectional Associative Memory Neural Network for Data Encryption and Decryption

Kushagra Pandey[1]
[1]*School Of Information Technology Engineering Vellore Institue Of Technology*

Abstract: ***Successful encryption and decryption of data has been a prime concern for any data transfer application. Data may be unwantedly attacked by a malicious attacker. Usually during data transfer, data in encrypted form along with a key is transferred, if a malicious attacker is able to get the key and is able to figure out the encryption algorithm used, the data no longer remains safe. Making a data encryption/decryption system, where there in no need to feed the encrypted data along with the key during data transfer, where key dynamically changes, can make our system more resistant to attackers. Block ciphers with symmetric key encryption are a well know technique for data encryption. A neuro block cipher can be established using recurrent neural networks to develop an encryption system with a dynamically changing key. Neuro recurrent networks can be researched to provide a new edge to network security.***

Keywords: ***Encryption, decryption, malicious attacker, block cipher, neuro block cipher, recurrent neural networks, network security***

I. INTRODUCTION

Neural networks are known to provide human like functionality. Researches have shown that different of soft computational functionality can incorporated in any possible field from medical research to network security.One such functionality is pattern recognition. There are neural networks formulated to work on pattern matching.

With data increasing at an exponential rate, the concerns regarding the security of data transfer has also increased. Various data encryption techniques are used by the experts to make systems less prone to attackers. As it turns out recurrent neural networks can be an excellent choice for data encryption. One such recurrent network is BAM. BAM can be used to develop a block cipher based encryption technique where every input blocks is related to each weight in the weight matrix. This weight matrix will be used as the public key for data transfer.

A bidirectional associative memory network is a recurrent neural network that can be used for pattern storage and matching the stored patterns. It is used to encode binary data into a weight matrix using Hebbian rule. The text data if converted into binary form can be encoded using the same technique.

The minimum requirements for any data encryption model during data transfer are the encryption key and the encrypted data. Imagine a system where there is no requirement of sending the encrypted data, insted a random integer number in its place that too changing after every data transfer, hence encrypting our data of any length into a single integer. This makes it much more harder for attacker to decrypt the data. Even if data at a particular instance is decrypted, the same pattern will not work for any future data, as the key changes after every transfer.This ever changing key makes the system dynamically very sound making it very hard for the attackers.

This system discussed above is not completely resistant to the attackers, but can help to reduce the attacks and provide a dynamic functionality to security model.

II. LITERATURE SURVEY

Network security has been in use since 1900BC. It has advanced over years. One of the first know cipher was Caesar cipher, it is a substitution cipher where the text message is shifted n places by a specific number n, that acts as key when modulo with twenty six. Decryption is done by following the reverse order as encryption. Although the Caesar cipher was not very effective as the key was easy to steal via brute force methods but it placed foundations for new stronger ciphers.

Block ciphers are yet another type of ciphers that encrypt data dividing data into blocks, and encrypting blocks insted of individual bits. One of the widely known block cipher is Data Encryption Standard(DES). It encrypts data by dividing it into blocks of 64 bits. A key of 56 bits is used. It uses a Initial Permutation function to perform permutations on plain text blocks individually after which text is further segmented into right and left half. Now each half goes over 16 rounds of encryption cycle, each round having a unique key. The key is used to create a sub-key of 48 bits using key transformations. The right text segment is converted to 48 bits using expansion permutation which is than XORed with the 48 bit sub key. S-box is used to produce 32 bit code from this 48 bit sequence. This 32 bit

©IJRASET: All Rights are Reserved

code undergoes P-box permutation to get another 32bit sequence which is XORed with left segment of the plain text. This resultant becomes the right part of the old right part becomes the left part, this process is called swapping. A advanced version of DES is Double DES that repeats some of the processes involved in DES twice to provide better security. It uses two keys insted of one. A triple DES is also in practice that uses 3 keys for encryption/decryption. DES is widely used but has been found to have some weaknesses in it's cipher design. The S-box can give same output with two different inputs, also the initial and final permutations are complicated and confusing even for the authorized users. Brute force also works on DES in finite time.

A better cipher than DES is Advanced Encryption Standard(AES) cipher. It is an iterative block cipher which was introduced to replace DES. It uses substitution and permutation for data encryption. The block size is double compared to DES i.e. 128bits. These 128bits or 16 bytes are represented by a 4*4 matrix. AES algorithm is applied on this matrix, and the number of rounds is a function of the length of the key used. 3 different keys of size 126,192 and 256 bits are used for encryption. Number of rounds for 128 bit key are 10, for 192 bit key are 12 and 256 bit key are 14. AES uses substitution bytes transform followed by shift row transform followed by mix column transform followed by add round key transform for encryption.

Although AES is more secure than DES and many other block ciphers, but it is difficult to implement AES that is both performing well as well as providing required security standards. It is repetitive in a way that every block is encrypted in same fashion and a simple algebraic structure is used.

Lately backpropagation neural networks have been used for encryption purposes. It is based on the logic that if an encrypted message has $N+N^{'}$ bits, it will only have 2N valid states. The receiver needs to find the right state out of all these 2N valid states. To find this required state, the receiver feeds all these 2N states to an supervised artificial neural network. The training set hence consists of pairs (E,M) where E is the encrypted data and M is the plain text. Backpropagation is used to reduce the error in the output layer. Delta rule is used to update randomly initialized weights of the network iteratively epoch after epoch until gradient descent reaches a minima or a plateau. This is a very effective way to decrypt data without transferring much data between two parties hence making the network more secure, but since the results are based on prediction, they are not 100% accurate and involve multiple epoch training which can very time consuming process increasing the data transfer time between the two parties.

III. METHODOLOGY

There are a few encryption techniques that have a varying key patterns, making the system less prone to malicious attackers. Although the frequency of change is low making it easier to steal the key.

The aim of this paper is to research an encryption system that is dynamically more sound and has lesser chance of being prone to attacks.

The algorithm used by our system tries to decipher the text that sender wants to transfer without actually transferring the text data, just the key. This makes attackers less accessible to the necessary information required to decrypt the text.

Our system uses neural networks to detect input patterns out of a target defined by the sender.

To make system more dynamic, the size of target vector also varies on both the sides, making it harder for the attacker to device its own neural network.

For the purpose of pattern recognition, the neural networks used here are recurrent networks. A block cipher was attempted to be created that takes segments of input in form of binary data, and encrypts these segments into different targets. The number of segments are decided by the sender providing more control to the sender. A common key is used for all the segments during decryption. The data in binary form is segmented into different blocks, pattern recognition and one-to-one mapping function is applied to each function. The one-to-one function leads to formation of a weight matrix that in turn leads to base of encryption or decryption.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VI, June 2018- Available at www.ijraset.com

IV. SYSTEM ARCHITECTURE

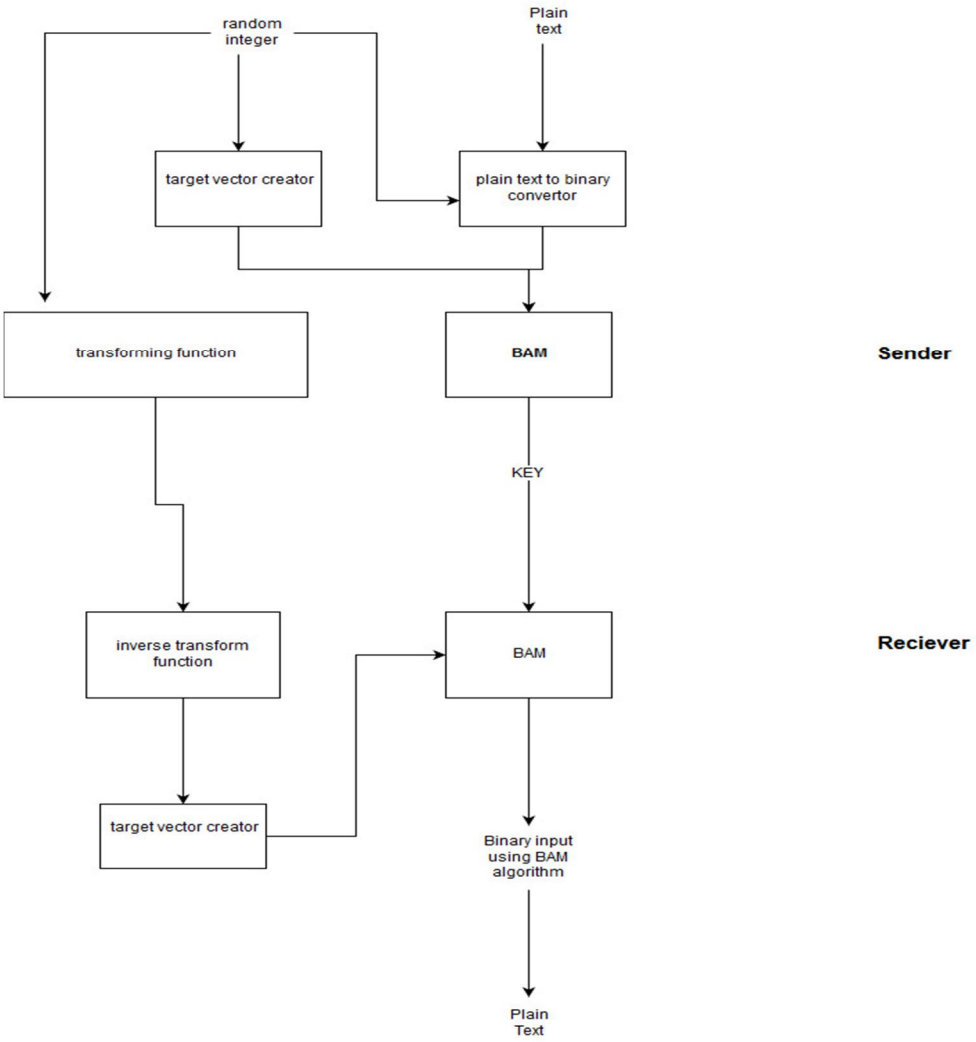


Fig1.System Architecture

The architecture comprises of sender side and receiver side and data being transferred between these two sides. A random integer is used to generate the target values for the plain text that undergoes conversion into binary data format, the target vector and the binary converter than work together to compile the BAM to be used. A transform function, i.e a general mathematical function is used to encrypt the random number to send it to the receiver side, without being exploited. The sender and receiver share this function, receiver on receiving the encrypted form of this number applies inverse transform function to generate the actual number used at the sender's side. As used at the sender's side this number is used to create the target vector, to revert the binary input pattern using the BAM received as key and the target vectors created.
The input thus obtained can be reverted back to the plain text using appropriate algorithms.

V. SYSTEM EXPLANATION

For using BAM for encryption, we need to convert the data in binary or bipolar form. Any of the two forms can be used, they do not effect the performance of the system differently.
BAM assigns targets to each binary input of same length, and uses hebbian rule to create an weight matrix for each input and finally combining them to create a final weight matrix.
While converting data into binary, an important fact to be considered is that different characters may be represented in different number of bits, for example, 2 needs only 2 bits, while 3 needs at least 3 bits to be represented in binary format. We use a system where

©IJRASET: All Rights are Reserved

each character be it alphabet or a number, is represented with equal number of bits. For our system, this is a necessity. Our system takes in user given data, that may comprise of any character from a standard QWERTY keypad, convert it into ASCII, and than convert the complete text in binary form.

Input pattern	Inputs	Targets	Weights
E	[1 1 1 1 −1 −1 1 1 1 1 −1 −1 1 1 1]	[−1, 1]	W_1
F	[1 1 1 1 1 1 1 −1 −1 1 −1 −1 1 −1 −1]	[1 1]	W_2

Fig2.Pattern E and F and their corresponding targets

Above is an example of how BAM works, each input pattern is assigned a target, and the weight matrix is computed. Here bipolar representation is used, binary can also be used. The two inputs represent the patterns E and F respectively. They are linked to the target values [-1 1] and [1 1]. The encrypted form of inputs E and F are the two target vectors . Individual keys for E and F are calculated as depicted in Fig2.

$$W = \sum s^T(p)\, t(p)$$

$$W_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \end{bmatrix} [-1 \quad 1] = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad W_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{bmatrix} [1 \quad 1] = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Fig3.Calculation of W_1 and W_2 using hebbian rule

$$W = W_1 + W_2 = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \\ 1 & -1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 0 & 2 \\ 2 & 0 \\ 2 & 0 \\ 0 & 2 \\ -2 & 0 \\ -2 & 0 \\ 0 & 2 \\ 0 & -2 \\ 0 & -2 \\ 0 & 2 \\ -2 & 0 \\ -2 & 0 \end{bmatrix}$$

Fig4. Final key for all the patterns

The final weight matrix that acts as the key for decrypting all the targets is W and is calculated adding all individual keys as shown in Fig3.

©IJRASET: All Rights are Reserved

Our system after converting data into binary form with each character represented with 9 bits, appends these binary representation of characters together and segments this appended data into 3 equal segments. 3 segments were done since we used 3 targets out1=[011],out2=[110] and out3=[101].

Each segment can be referred as an input pattern in accordance with Fig1. although insted of a single character, here each pattern represents a group of characters.

A list or a matrix, is created containing all the target values.

Out=[out^1,out^2,out^3]

For easier computations, a list of all the input values are also created. Each vector in Inp corresponds to a block to be encrypted.

Inp=[$input^1$,$input^2$,$input^3$]

Where $input^j$ is the j^{th} segment of the binary data.

Weights are calculated by matrix multiplication of Out and Inp(Fig2). This weight acts as the key for that particular data,i.e Inp.

.The receiver has a similar copy of Out list. Once it will receive the key from sender, it can calculate the binary data by matrix multiplication between the target matrix and transform of the key or weights received from the sender and applying an appropriate activation to the pattern formed.

$$y_{in} = x \cdot W^T = [-1\ 1] \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 0 & -2 & -2 & 0 & 0 & 0 & 0 & -2 & -2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & -2 & 2 & 0 & 0 \end{bmatrix}$$
$$= [2\ 2\ 2\ 2\ -2\ -2\ 2\ 2\ 2\ 2\ -2\ -2\ 2\ 2\ 2]$$

$$y_{in} = x \cdot W^T = [1\ 1] \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 2 & 0 & -2 & -2 & 0 & 0 & 0 & 0 & -2 & -2 \\ 2 & 2 & 2 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & -2 & -2 & 2 & 0 & 0 \end{bmatrix}$$
$$= [2\ 2\ 2\ 2\ 2\ 2\ 2\ -2\ -2\ 2\ -2\ -2\ 2\ -2\ -2]$$

Fig5. Calculating input pattern using target and key

Fig4. shows conversion of patterns E and F(discussed in Fig1) from target to their input patterns. The y calculated are fed to an activation function, that yields the final result (Fig5).

$$y = [1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ 1\ 1\ -1\ -1\ 1\ 1\ 1]$$
$$y = [1\ 1\ 1\ 1\ 1\ 1\ 1\ -1\ -1\ 1\ -1\ -1\ 1\ -1\ -1]$$

Fig6. Applying activation to y calculated in Fig4

$$y_j = \begin{cases} 1 & \text{if } y_{inj} > 0 \\ y_j & \text{if } y_{inj} = 0 \\ 0 & \text{if } y_{inj} < 0 \end{cases} \qquad x_i = \begin{cases} 1 & \text{if } x_{ini} > \theta_i \\ x_i & \text{if } x_{ini} = \theta_i \\ -1 & \text{if } x_{ini} < \theta_i \end{cases}$$

Fig7.Activation function for binary and bipolar BAM

Every time sender sends the data, a new weight matrix corresponding to the binary equivalent of that data is created. Hence even if key is received by an attacker, it can only be used for that specific data transfer, not for any upcoming data. Also the attacker is unaware of the target values.The target values in our system acts as the encrypted data, since target values are not transferred, attacker can not attack the system, as encrypted data or target is not available to do any matrix operations to produce binary data equivalent of the plain text.

With above method we successfully made our system dynamic, but the Out matrix remains completely constant throughout the process. If the attacker were to use brute force methods to predict the values of target, it can make our system easily prone to attack, as any key when multiplied to the Out matrix would yield the required result. The current system used uses only 3 targets, brute force will not be a costly method for such a system as only 6 combinations will be possible.

©IJRASET: All Rights are Reserved

To get rid of this limitation, two step needs to be taken, the first one is increasing the size of the number of target vectors used, this will increase the number of combinations possible. The second one is making the number of target vectors variable. This will be defined by an integer number, that sender will send to the receiver. But this integer value will not be the number of input vectors to be used, insted, the sender and the receiver will share a polynomial or some other non linear function (a function that intakes an integer number and outputs back an integer), that when applied to this number will be used to determine the target size.

The number of target vectors, and the number of bits of each target inside the target vector remains the same, such that each target has only one 1 bit, rest all are 0 bits, that too in an specific format, i.e., the first target will have leftmost bit 1, the second target will have second bit as 1 and so on. This is done to make the target vector same on both the side, if a random pattern is followed, the target values may differ on the two sides making it impossible to decrypt at receiver's correctly. Also if there are more than one non-zero bits in a target vector, BAM will not give correct multiplication results for that specific vector, this is a conditional constraint of this algorithm.

Another point to be considered here is, the number of segments in which our binary form of plain text is to be segmented. For our initial model with 3 targets, we choose 9 as our bit size to represent each character, this was not a random choice, but done to segment our binary data into 3 segments of equal length, for similar purposes, the bit size of each charter also needs to be made variable depending upon the number of target values used. We do not need it to be too large as it will make our key large making it harder to transfer so we select bit size to be smallest integral multiple of the target size greater than or equal to 9. This way we can divide the binary data into segments of equal integer length.

Once the target size has been defined, this value is passed to the receiver side using a polynomial function, that is provided on both the sides. The sender feeds the target side to this function, sending the result to the receiver, the receiver applies inverse function to this number received, and finds the actual target size, making required alterations and creating the required target vectors. To make the system more secure, the target matrix can be rolled in same direction by same amount on both the sides.

Insted of a polynomial function, some other more complex function can also be used. But the result obtained after applying the function should also yield integer.

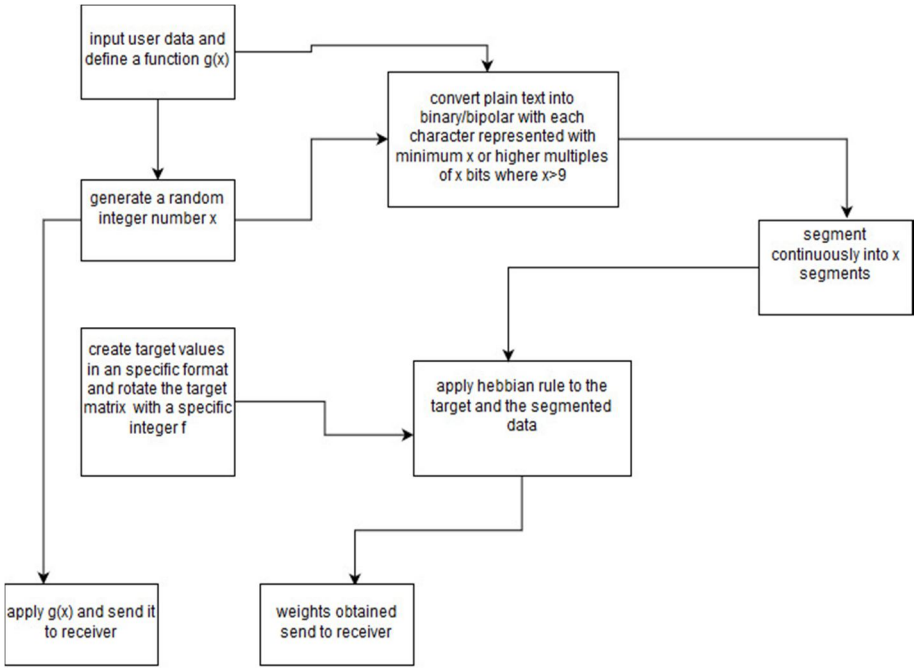


Fig4. Flow diagram of processing at sender's side

Fig4. depicts a flow of all the processes involved in the encryption of data at senders side.

A similar operation is to be applied on the receiver side as well, the integer value is to be fed to the inverse g(x) function and the value thus obtained is used to define the target values, than dot operation between the targets formed and transform of key provides us with the required input data. It is converted from binary to ASCII values and finally to the corresponding character.

Hence we successfully created encrypted data and later decrypted it at receiver side itself, without transferring the data itself.

The algorithm can be summarized as follows:

©IJRASET: All Rights are Reserved

A. At sender's side
B. Define a function g(x) to be shared between sender and receiver. Eg=g(x)=$2x^2$+3x+
C. Generate a random integer number which represents the number target vectors in the target matrix. Let this number be x=s.Calculate the target matrix, in a specific format,i.e, the first target vector will have 1 on it's first bit,rest all zeros.

Eg. For t=4
Target corresponding to segment1=[1000]
Target corresponding to segment1=[0100]
And so in….
The number of bits used to represent an individual target vector is equal to the number of target vectors.

D. Now use t number of bits to represent the ASCII representation of each character in bit form, such that t is divisible by s and t >= 9.
E. Segment the complete bit notation into t equal segments now.
F. Dot product the target along with their respective segments using BAM algorithm to create the key.
G. Pass the number s to g(x) to calculate g(s).
H. Transfer g(s) and key to the receiver.

The key used here has size equal to that of the binary plain data before encryption. For large sized data to be transferred, a large key is generated. Transfer of this key is hardware constrained, hence to transfer bigger keys buffer of large size is required. This can be expensive.

VI. CONCLUSION

The encryption technique used above can provide a new edge to the dynamic nature of data encryption. Making encrypted data unavailable during data transfer makes it much harder to be attacked.
Using a function to encrypt the number of target units used is yet another successful way to protect our data from attacks.
Although success on data transfer using such an encryption technique may depend completely upon the hardware used.
Bidirectional networks acts as an excellent choice of recurrent networks for the above model. The BAM used here is a discrete BAM.
Unlike a backpropagation based neuro cryptographic method that involves multiple epochs, BAM based model is a simple yet effective one. It is faster and more dynamic. It is not based on error correction, hence is accurate.

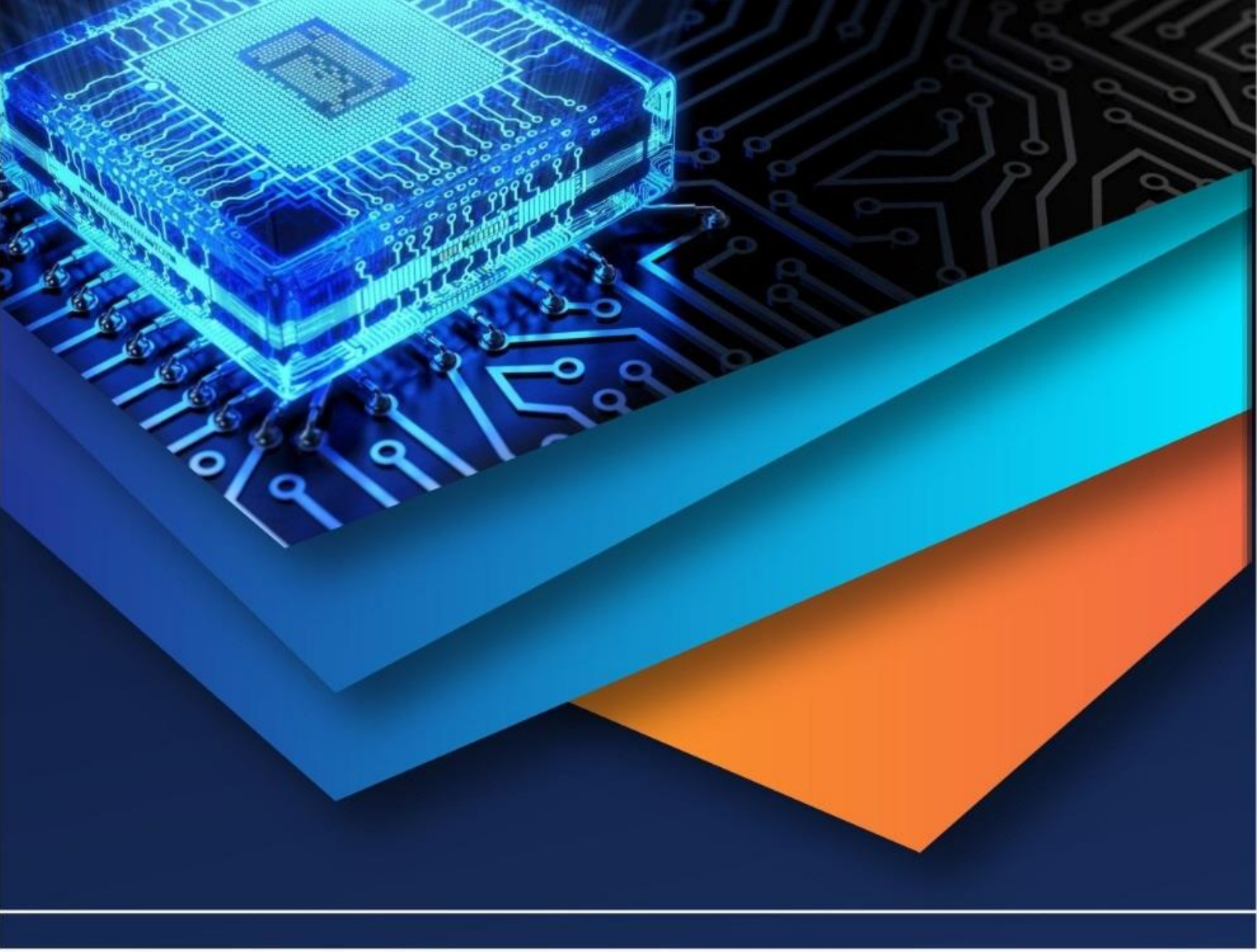
VII. ACKNOWLEDGMENT

This research was carried out and supported by Vellore Institute of Technology Vellore, Tamil Nadu, India. I would like to thank my professors from School of Information Technology Engineering(SITE) specializing in network security and soft computing who provided guidance and insight that assisted in completion of this research.
I would also like to show my gratitude to the 4 "anonymous" reviewers for their insights and reviews on the research. We are also thankful to everyone for their comments on an earlier version of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

[1] Enas Ismael Imran and Farah abdulameerabdulkareem",Enhancement Caesar Cipher for Better Security", p-ISSN: 2278-8727Volume 16, Issue3, May-Jun. 2014
[2] Amalijeet Kaur and Sukhman Sodhi,"Data Encryption Standard Algorithm(DES) for Secure Data Transmission",International Conference on Advances in Emerging Technology 2016..
[3] W. Stallings, Cryptography and Network Security: Principles and Practices,5th ed., Prentice Hall, 1999.
[4] Ako Muhamad Abdullah,"Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data",Department of Applied Mathematics & Computer Science Eastern Mediterranean University-Cyprus.
[5] Behrouz A. Forouzan." Cryptography and Network Security". Tata McGraw-Hill, 2007.
[6] Dr. S.N. Sivanandam and Dr. S.N. Deepa,"Principle of Soft Computing", 2nd edition Wiley.
[7] Dr. Mohammed M. Alani — "Improved DES Security", International Multi-Conference On System, Signals and Devices, 2010.
[8] Khalil Shihab,"A Backporpagation Neural Network for Computer Neural Security", Department of Computer Science,SQU,Oman,Journal of Computer Science.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)