Autoencoder Artificial Neural Network Public Key Cryptography in Unsecure Public channel Communication

Vikas Sagar, Krishan Kumar

ABSTRACT: As we all known that cryptography is a procedure to hide data so that it can’t be access or modified by any unauthorized entity. At the present digital world security is a main concern. To maintain this security there are many cryptographic algorithm exist. But the world technology grew each and every day so we have to find some new algorithms to maintain the security at higher level. In the proposed and implemented work used artificial neural network to increase the security during data communication in digital world. Autoencoder Neural Network is a new approach in the era of digital world so that used here in cryptographic algorithm to increase the strength of the security. There are three basic aims of cryptography availability, privacy and integrity easily achieved by this new approach. This work examine that the attacker can’t get access the data however he/she exist in the same network or not. Neural Network's uncertainty property make this possible. This approach also examined on different data size and key size. Proposed work used the autoencoder for encryption and decryption. The final experimental result show our purposed algorithm efficient and accurate and also show how this approach perform better. Proposed and implemented algorithm can be easily used for secure data communication with more efficiently.

KEYWORDS: Cryptography, Symmetric key, Security, Autoencoder neural network.

I. INTRODUCTION

From a long time Cryptography used to maintain security in military and secret operations during information transformation. But nowadays all the transfer and communication done over the internet. 21st is a digital world and here Cryptography play a very important role for any kind of information transfer. That information may be for a particular person or for an organization. There important data or information keep safe and sound by the help of the cryptography from an unauthorized entity.

In the cryptographic algorithms mathematical calculation used to secure the data. With mathematical operations secret data that’s called plain text convert in to scramble data that called cipher text. The plain text convert in to cipher text to maintain the confidentiality during the data transformation. There are two main stream to categorized cryptographic algorithm: one is symmetric key cryptography and other one is asymmetric key cryptography. In the symmetric key cryptography or private key cryptography a single key is used for both, encryption procedure and decryption procedure.

II. OUTLINE OF PUBLIC KEY CRYPTOGRAPHY

As we already know that a single key used in public key or Symmetric cryptography for encryption procedure and decryption procedure [1]. The main components of any cryptography scheme:

1. Plaintext
2. Encryption algorithm:
3. Secret Key:
4. Ciphertext:
5. Decryption Algorithm

Let’s take an example of information transformation between two entity A and B to understand these terms.
First of all Plaintext is the main text or important data that entity A want to send to entity B with a security that no one can access except B. To achieve this security an Encryption algorithm used. In the encryption scheme different mathematical operation take place to scramble the data. To increase the security Secret Key is used with encryption scheme. After that encryption process the output data is called Ciphertext which scrambled and can’t be access by unauthorized entity. Then that cipher text send to the entity B over the public channel. B received the ciphertext and convert ciphertext in to the plaintext with the help of same secret key and Decryption Algorithm. Decryption algorithm is just reverse process of the encryption algorithm.

The overall public key cryptography process shown in figure:

**III. OUTLINE OF ANN**

Neural networks were first introduced by Mc Culloch Pitts in 1943. ANN is inspired from biological neurons. ANN basically consists of small computing units, called neurons, arranged in different layers and interconnected with each other with some nerves connections called synapses. Simple mathematical computations are performed in each neuron. ANN tries to mimic the behavior of biological neurons and their interconnections. These are widely used in different areas together with other technologies for solving artificial intelligent problems, such as functions approximation, modeling, brain mapping, air traffic control, stock exchange, control systems, financial modeling, data compression, classification tasks or data processing [3], [6]. ANN divided in to two categories

**Supervised learning:** In supervised learning, the model work over the input and produce the output that is predefined or known.

**Unsupervised learning:** In contrast of supervised learning the unsupervised learning doesn’t have the known output. So that its work only on input by itself.

**IV. METHODOLOGY OF AUTOENCODER NEURAL NETWORK PUBKIC KEY CRYPTOGRAPHY**

ANN is a modern technique which perform better day by day in all real life areas. In this paper we proposed a new approach based on the Autoencoder artificial neural network (AANN). ANN can also perform an important role in the field of cryptography nowadays. There are many approaches already exist with ANN in cryptography. ANN is work on the mimic behavior of neurons which react according the real time events so that it can be gave effective result in the cryptography area.

**V. AUTOENCODER**

An Autoencoder also called autoassosciator or Diabolo network is an ANN that is fall in the unsupervised learning category. Autoencoder neural network applies backpropagation process to setting the target values to be equal to the inputs. Because of its unsupervised, unpredictability and randomness property it’s very helpful in security [3], [4].

In simple words we can say that autoencoder is used to copy the input to output. Autoencoder compress their input by the procedure that’s called latent-space representation, and then regenerate the output same as input. Autoencoder is a combination of two individual unites. That are-

1. **Encoder:** This part of the autoencoder neural network compresses the input into a latent-space representation. This encoding function entitled by $h=f(x)$.
2. **Decoder:** This part of autoencoder reassemble data that is generate by the encoder which is in latent space representation form. This decoding function entitled by $r=g(h)$.

**Figure 2: Basic Architecture of an Autoencoder [16]**

5.1 **Classification of Autoencoder**

There are different types of Autoencoders:

1. **Undercomplete Autoencoders:** An autoencoder whose code measurement is not as much as the info measurement. Adapting such an autoencoder drives it to catch the most remarkable highlights. Be that as it may, utilizing a major encoder and decoder in the absence of enough preparing information enables the system to retain the errand and discards learning helpful highlights. In the event of having straight decoder it can go about as PCA. Nonetheless, adding nonlinear enactment capacities to the system makes it a nonlinear speculation of PCA.
2. **Regularized Autoencoders:** Rather than restricting the span of autoencoder and the code measurement for include learning, we can add a misfortune capacity to aver it retaining th assignment and the preparation data.
• Meager Autoencoders: An autoencoder which has a sparsity punishment in the preparation misfortune notwithstanding the reproduction mistake. They normally being utilized for the purpose of different assignments, for example, arrangement. The misfortune is not as straightforward as different regularizes, and we will talk about it in another post later.
• Denoising Autoencoders (DAE): The contribution of a DAE is a tainted duplicate of the genuine info which should be remade. Therefore, a DAE needs to fix the debasement (commotion) and additionally recreation.
• Contractive Autoencoders (CAE): The fundamental thought behind these kind of autoencoders is to take in a portrayal of the information which is powerful to little changes of the info.

3. Variational Autoencoders: They amplify the likelihood of the preparation information as opposed to duplicating the contribution to the yield and in this manner does not require regularization to catch helpful data.

5.2 Structure of Autoencoder
An autoencoder is combination of two, symmetrical neural network that may consist four or five superficial layers. The half layers of the network used for the encoding and second set superficial layers carry out the operation of decoding [12], [14].
The layers are limited Boltzmann machines, the building squares of profound conviction systems, with a few eccentricities that we’ll talk about beneath.
General structure of an autoencoder
• Maps an input \( x \) to an output \( r \) (called reconstruction) through an internal representation code \( h \)
• It has a hidden layer \( h \) that describes a code used to represent the input
• The network has two parts
• The encoder function \( h = f(x) \)
• A decoder that produces a reconstruction \( r = g(h) \)

Here we’ll explain the simplified structure of the autoencoder schema.

![Schematic structure of an autoencoder with 3 fully-connected hidden layers](image-url)

**Figure 3:** Schematic structure of an autoencoder with 3 fully-connected hidden layers [17]

Encoder \( f \) and decoder \( g \)

\[
f : X \rightarrow h \\
g : h \rightarrow X \\
\text{arg min} \ ||X - (f \circ g) X||^2
\]

One hidden layer
• Non-linear encoder
• Takes input \( x \in R^d \)
• Maps into output \( h \in R^e \)

\[
h = \sigma_1(Wx + b) \\
x' = \sigma_2(W'h + b')
\]

Where \( \sigma \) is an element-wise activation function such as sigmoid.

Trained to minimize reconstruction error (such as sum of squared errors)

\[
L(x, x') = ||x - x'||^2 = ||x - \sigma_2(W'(\sigma_1(Wx + b)) + b'))||^2
\]

Provides a compressed representation of the input \( x \)

VI. AUTOENCODER ARTIFICIAL NEURAL NETWORK PUBLIC KEY CRYPTOGRAPHY (AANNPKC)

Our algorithm based on the Autoencoder neural network. As we already know that autoencoder artificial neural network is an unsupervised learning scheme which is worked like backpropagation algorithm. The simplest example for an autoencoder is feed-forward neural network or no-recurrent neural network. Autoencoder based on encoder and decoder two symmetric individual neural network parts. Here encoder and decoder both used the same key. Now in term of cryptography here we suppose that two entity Alice and Bob want to communicate confidentially. To attain this objective Alice used autoencoder artificial neural network. Alice generate a training set for this. This training set based on the binary values of all alphabets, numbers and special symbols. Padding and key values also include for this training procedure. Now this training set securely pass to the Bob. Now Alice and Bob both use the same training set to train their autoencoder neural networks. Alice use binary values as feature vector to train her autoencoder neural network and generate the ciphertext. On the other hand Bob use the ciphertext as feature vector and recreate the original binary value.
Autoencoder Artificial Neural Network Public Key Cryptography in Unsecure Public channel Communication

Here we describe the whole process of the algorithm with the example how it’s work with real data. The main aim of the proposed algorithm if to maintain four pillars of cryptography (Confidentiality, Data integrity, Authentication, Non-repudiation) and that is finally tasted at the end of this paper and compare this algorithm with other pre-existing algorithms.

**Encryption:**
1. Very first step we input data from the sender that’s called Plain text or original text.
2. In the second step the original text turn in to ASCII form.
3. Here we turn the ASCII in to Binary format.
4. Now we use padding and secret key on this Binary data to make data satiable for (Encoder) autoencoder neural network layer.
5. Apply the autoencoder and found the cipher text which send to the receiver as it is.

**Decryption:**
1. Cipher text receives at the receiver side and send directly to the (Decoder) autoencoder neural network layer with the secret key.
2. Then the output data convert to the Binary value and send to next step.
3. Here Binary value turn in to ASCII format and finally ASCII convert to corresponding to plain text.

**Encryption process with example on sender side:**

**STEP 1:** Firstly take the simple text massage.

Hi Bob! It is a very important text

**STEP 2:** Now in second step convert this initial value in to ASCII format

72 105 32 66 111 98 32 33 32 105 116 32 105 115 32 97 32 118 101 114 121 32 105 109 112 111 114 116 97 110 116 32 116 101 120 116 46

**STEP 3:** Convert the ASCII format in to binary format.

00110111 00100000 00110001 00110000 00100000 00110110 00100000 00110011 00110010 00100000 00110011 00110010 00100000 00110001 00110000 00100000 00110001 00110001 00110110 00100000 00110001 00110001 00110110 00100000 00110110 00110110 001000

**STEP 4:** Now we use padding and secret key on this Binary data to make data satiable for (Encoder) autoencoder neural network layer.

00110111 00100010 00100000 00110001 00110000 00110100 00100000 00110110 00100000 00110001 00110001 00110110 00100000 00110110 00110110 001000

**STEP 5:** Apply the autoencoder and found the cipher text which send to the receiver as it is.

00100000 00110001 00110001 00110110 00100000 00110110 00100000 00110001 00110001 00110110 00100000 00110110 00110110 001000
Now this is the final ciphertext which send to the receiving side.

Decryption process on second side: - At this end ciphertext received and then reverse process of the encryption done to recreate the original text message.

STEP 1: Very first step to receive the ciphertext from the sender.

STEP 2: Now apply (Decoder) Autoencoder to the ciphertext to get the original text back.

STEP 3: Convert that output (binary string) in to ASCII. Finally convert ASCII to corresponding to plain text.

Hi Bob! It is a very important text.

This is the whole procedure how this algorithm work. Now let’s suppose that someone want to spy Alice and Bob’s conversation. There are two methods for that. First one spy entity create the same to sane two autoencoder with same training set that is almost impossible. Second is somehow spy entity try to crack the ciphertext with guessing or brute force attack that also take a huge amount of time. So that this algorithm is very much secure.

VII. Assessment Methodology And Analysis Results

The assessment methodology is a process to enable one better to comprehend the means expected to complete a quality evaluation. Here we simply used different algorithms codes which were generated manually some of them also downloaded from internet and some source code get from some papers. We try to maintain each and every algorithm as per rules.

Software Speciation: Experimental perform Java NetBeans IDE 8.0, Matlab version 2014, Windows 8 Profissional 64 bit Operating System.

Hardware Speciation: Here we use Dell Inspiron 15R 5000 series Intel Core i3 (2.40 GHz) fifth generation processor with 6GB ram with 1TB-HDD. All the algorithms are tested on Intel Core i5 (2.40 GHz).
Autoencoder Artificial Neural Network Public Key Cryptography in Unsecure Public channel Communication

Try to implement these encryption algorithm as fair as possible with all aspects. The performance of different algorithms may be change from previous data just because of hardware and software.

Figure 3: First view of the implemented software for all cryptographic algorithms.

Table 1: Cryptographic algorithms comparison with their basic properties

| Algorithm Name | Structure of Algorithm | Plain Text Length | Key Size (Bits) | S boxes | No. of Rounds |
|----------------|------------------------|-------------------|----------------|---------|---------------|
| DES            | Festial structure      | 64 bits           | 56             | 8       | 16            |
| 3DES           | Festial structure      | 64 bits           | 56, 112, 168   | 8       | 48            |
| Blowfish       | Festial structure      | 64 bits           | 32-448         | 4       | 16            |
| IDEA           | Substitution-Permutation Structure (Lai-Massey scheme) | 64 bits | 128 | N/A | 8.5 |
| AES (Rijndael) | Substitution-Permutation Structure | 128 Bits | 128, 192, 256 | 1 | 10, 12, 14 |
| RC6            | Festial structure      | 128 Bits          | 128, 192, 256  | N/A     | 20            |
| AANNPKC        | Neural Network         | 128 Bits          | N/A            | N/A     | Depends on Training Module or User |

Here we present the average time of different cryptographic algorithms on different data size (500, 800, 1000 bytes). We take the average of 100 running executions on each data size to make the comparisons significance and perfect result.
Table 1: Average encryption and decryption time of different cryptographic algorithms on different data size.

| S. No | Algorithm Name | Key Size (bit) | File Size (bytes) | Encryption Time (Nanosecond) Average (100 Times) | Decryption Time (Nanosecond) Average (100 Times) |
|-------|----------------|----------------|-------------------|-----------------------------------------------|-----------------------------------------------|
| 1     | DES            | 56             | 500               | 343                                           | 344                                           |
|       |                |                | 800               | 357                                           | 360                                           |
|       |                |                | 1000              | 375                                           | 379                                           |
| 2     | 3DES           | 168            | 500               | 339                                           | 322                                           |
|       |                |                | 800               | 348                                           | 358                                           |
|       |                |                | 1000              | 360                                           | 379                                           |
| 3     | Blow-fish      | 256            | 500               | 333                                           | 311                                           |
|       |                |                | 800               | 340                                           | 315                                           |
|       |                |                | 1000              | 345                                           | 317                                           |
| 4     | IDEA           | 128            | 500               | 320                                           | 313                                           |
|       |                |                | 800               | 329                                           | 320                                           |
|       |                |                | 1000              | 335                                           | 340                                           |
| 5     | AES (Rijndael) | 256            | 500               | 340                                           | 369                                           |
|       |                |                | 800               | 360                                           | 370                                           |
|       |                |                | 1000              | 370                                           | 391                                           |
| 6     | RC6            | 256            | 500               | 334                                           | 340                                           |
|       |                |                | 800               | 341                                           | 350                                           |
|       |                |                | 1000              | 342                                           | 399                                           |
| 7     | AANNPKC        | N/A            | 500               | 311                                           | 320                                           |
|       |                |                | 800               | 290                                           | 321                                           |
|       |                |                | 1000              | 310                                           | 325                                           |

This following chat shows the encryption time of different cryptographic algorithms on different data size. It’s clear from the chat that the AANNPKC algorithm take the less time by any other algorithm for the encryption of any size of data.

This following chat shows the decryption time of different cryptographic algorithms on different data size. It’s clear from the chat that the AANNPKC algorithm take the less time by any other algorithm for the decryption of any size of data.
The above graph show the key value required for the different algorithms. It shows that the AANNPKC’s key depend on the plaintext only. Mostly the security depends on the key length on any algorithms but in AANNPKC key value depends on the plaintext length that’s means key value every time change according to that and security also depend on that.
Throughput

The above diagrams demonstrate throughput for different algorithms. Throughput is characterized as the quantity of bytes sent over the channel per unit of time. It’s clear from the result that the AANNPKC have the best throughput in all algorithms that will make this algorithm best with respect to other algorithms.

Throughput = bytes sent / time in seconds

VIII. FLOW CHART OF PROPOSED ALGORITHM

The above flowchart shows the flow of the data from sender side to receiver side with each and every step of encryption and decryption.
The optimality of the proposed algorithm can be judged based on various parameters. The parameters that we have chosen in our research work are as follows: Encryption Time taken to encrypt a file of fixed size. The less Encryption time is preferred as the algorithm has to be fast enough to convert the secret message into a hidden message. Decryption Time taken to decrypt a file of fixed size. The less Decryption time is preferred as the algorithm has to be fast enough to convert the hidden message back into the secret message. Time to generate keys Time to generate keys should be as less as possible as keys play the most important role in Encrypting and Decrypting data. Throughput should be as high as possible as it denotes the amount of data to be transferred from the channel.

IX. CONCLUSION

The main objective of this paper is to generate a secure symmetric key, and hence communicate the information over the wired or wireless network medium with the easy and effective way. In this work, a new cryptographic algorithm proposed and implemented using autoencoder artificial neural networks which fits to the required objective, and is efficient and effective to send and receive data securely. Also, for this algorithm low or high bandwidth does not affect the performance. Consequently it's clear that this approach is easy, fast, and simple, if compared with other algorithms.

REFERENCES

1. Menezes, P. van Oorschot and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, First ed., 1997.
2. Fi-John Chang, Yen-Chang Chen , "A counterpropagation fuzzy neural network modeling approach to real time streamflow prediction", journal of hydrology, ELSEVIER, 6 February 2001.
3. Eva Volna, Martin Kotyrba, Vaclav Kocian and Michal Janosek “Cryptography based on neural network” Proceedings 26th European Conference on odeling and Simulation OECMS Klaus G. Troitzsch, Michael Mohring, Ulf Lotzmann (Editors) ISBN: 978-0-9564944-4-3 / ISBN: 978-0-9564944-5-0, ecms2012.
4. Vikas Gujral, “Cryptography using Artificial neural network”, Engineering National Institute of Technology Rourkela-769008 Orrissa, Session 2005-2009.
5. Ahmed M. Allam ; Hazem M. Abbas ; M. Wathiq El-Kharashi, “Security analysis of neural cryptography implementation”, 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), ISSN: 2154-5952.
6. Pranita P. Halke,Swati G. Kale, “Use of Neural Networks in cryptography: A review”, 2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave), ISBN: 978-1-4673-9214-3, 29 Feb.-1 March 2016
7. Vikas sagar, Krishan Kumar, “ A Symmetric Key Cryptography using Genetic Algorithm and Error Back Propagation Neural Network”, 2015 2nd International Conference on “Computing for Sustainable Global Development”, 11th – 13th March, 2015 Bharati Vidyapeeth’s Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA).
8. Ayush Sethi1, Ayush Mittal2, Ritu Tiwari3, Deepa Singh, “Elliptic Curve Cryptography Using Chaotic Neural Network
9. R. Mislovaty ; E. Klein ; I. Kanter ; W. Kinzel, “Security of neural cryptography”, Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECES 2004. ISBN: 0-7803-8715-5.
10. Vikas Sagar, Krishan Kumar, “A Hybrid Soft Computing Approach on Security Algorithm of Cryptography”, International Journal for Research in Applied Science & Engineering Technology (IRASET) ISSN: 2321-9653; IC Value: 45.98; Volume 5 Issue XI November 2017.