I.J. Wireless and Microwave Technologies, **2023, 5, 23-33**
Published Online on October 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2023.05.03

An Enhanced Method Utilizing Hopfield Neural Model for Mobile Agent Protection

Pradeep Kumar*
Ph.D. Research Scholar, Shobhit Institute of Engineering and Technology (Deemed-to-be University), Meerut, India
Assistant Professor ,JSS Academy of Technical Education, Noida, Uttar Pradesh, India
E-mail: pradeep8984@jssaten.ac.in
ORCID ID: https://orcid.org/0000-0001-6177-8527
*Corresponding Author

Niraj Singhal
Director, Sir Chhotu Ram Institute of Engineering and Technology, Chaudhary Charan Singh University, Meerut, India
E-mail:drnirajsinghal@gmail.com
ORCID ID: https://orcid.org/0000-0002-2614-4788

Ajay Kumar
JSS Academy of Technical Education, Noida, India
E-mail:er. ajay.itcs@gmail.com
ORCID ID: https://orcid.org/0000-0002-3693-9701

Kakoli Banerjee
JSS Academy of Technical Education, Noida, Uttar Pradesh, India
E-mail: kakoli.banerjee@jssaten.ac.in
ORCID ID: https://orcid.org/0000-0002-4306-2194

Received: 01 February, 2023; Revised: 15 April, 2023; Accepted: 12 May, 2023; Published: 08 October, 2023

Abstract: Mobile agent is a piece of computer code that organically goes from one host to the another in a consistent or inconsistent environment to distribute data among users. An autonomous mobile agent is an operational programme that may migrate from one computer to machine in different networks under its own direction. Numerous health care procedures use the mobile agent concept. An agent can choose to either follow a predetermined course on the network or determine its own path using information gathered from the network. Security concerns are the main issue with mobile agents. Agent servers that provide the agents with a setting for prosecution are vulnerable to attack by cunning agents. In the same way agent could be carrying sensitive information like credit card details, national level security message, passwords and attackers can access these files by acting as a middle man. In this paper, optimized approach is provided to encrypt the data carried by mobile agent with Advanced Encryption Standard (AES) algorithm and secure key to be utilized by the AES Encryption algorithm is generated with the help of Hopfield Neural Network (HNN). To validate our approach, the comparison is done and found that the time taken to generate the key using HNN is 1101ms for 1000 iterations which is lesser than the existing models that are Recurrent Neural Networks and Multilayer Perceptron Network models. To add an additional level of security, data is encoded using hash maps which make the data not easily readable even after decrypting the information. In this way it is ensured that, when the confidential data is transmitted between the sender and the receiver, no one can regenerate the message as there is no exchange of key involved in the process.

Index Terms: Hopfield Neural Network (HNN), Mobile Agent Security, Advanced Encryption Standard (AES), Secret-key, Encryption, Decryption.

1. Introduction

A mobile agent [1] can be compared to a program that can migrate from one platform to another. Overall working of mobile agent shown in Figure 1. The destination machine runs instructions in the agent's software. It is the only system in the network that can migrate itself among various systems.

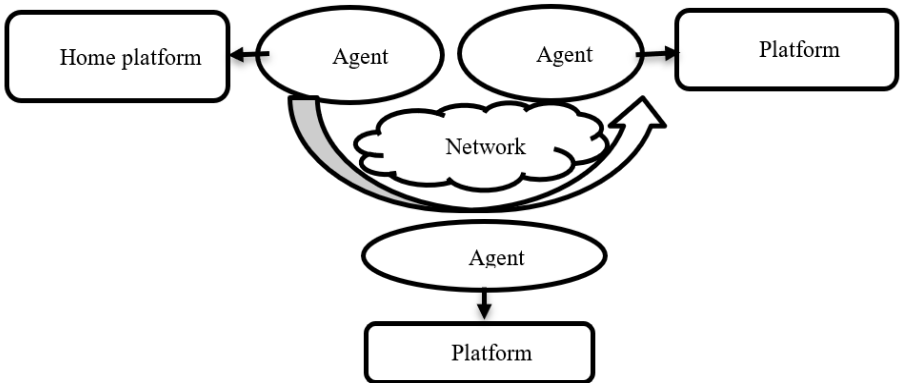

Fig. 1. Mobile Agent System

When travelling, a mobile agent can migrate to plateform that contains the resources they wish to interact with and benefit from lying over the same host or network. Common errors in agent-oriented provide a good assessment of the mobile agent design paradigm, several agent frameworks, and possible applications [2]. If the host is not connected to the network, a mobile agent can execute process on behalf of host after completing a predefine mobile agent life cycle [3] present in Figure 2.

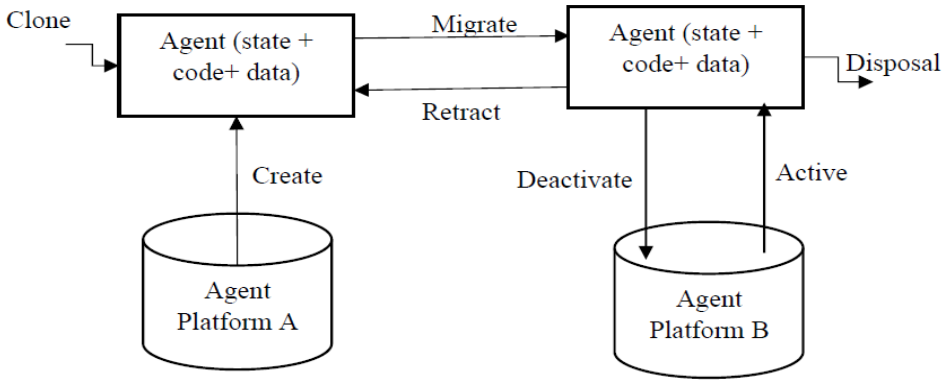

Fig. 2. Mobile Agent execution life cycle

The mobile agent delivers all the outcomes to the host after it has rejoined to the network. Deploying mobile agents[4] has many benefits, including reduced network bandwidth needs, reduced network latency, automatic and dynamic behaviour, and robustness. Mobile agents are useful for a variety of things, including e-commerce, banking, and security.

1.1. Threats to Mobile Agent

The mobile agent's dynamic nature resulted in a complex design and security risks. According to the threat's creator and victim, these [5] are divided into four groups shown in Figure 3. The following are the categories:

1. **Agent to Platform**: This topic refers to the dangers that a mobile agent poses to a specific platform.
2. **Platform to Agent:** This topic is about platform threats to a specific mobile agent.
3. **Agent to Agent:** This topic is concerned with dangers which might arise as a result of mobile agents interacting.
4. **Platform to Platform**: This topic deals with dangers posed by mobile agent platforms.

1.2. Requirement of Security of Mobile Agent

Security of mobile agents [6] has become a crucial issue to settle when mobile agents are expected to be used across highly distributed heterogenous networks. Figure 3 illustrates how security is organized in MAS [7] more generally. Mobile agent automatically migrating in malicious vulnerable environment. During the life cycle of mobile agent carrying confidential results after the execution of assigned task. So, there is a requirement of optimal mechanism for the security of mobile agents.

1.3. Encoding and Decoding

The process of conversion of characters (letters, numbers, symbols, etc.) into a specific format for increasing the efficiency of transmission and storage is known as encoding. The process of conversion of encoded data back into the original characters is known as the process of decoding. It is possible to encode a message into a non-readable message that can only be decoded if the HashMap of the original characters is present.

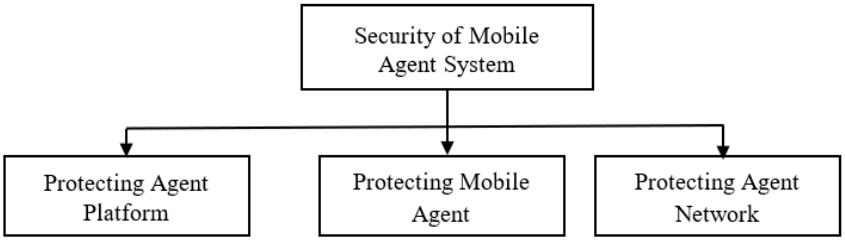

Fig. 3. Classification of Security in MAS

1.4. Advanced Encryption Standard (AES)

AES is a symmetric cipher [8], which means that, like our previous lock, one key is required for the encoding and decoding of secure data. It means that there is no faster way to decrypt AES-encrypted data without knowing the key than brute forcing it: testing all potential keys until you identify the one that was used for encryption. AES is a block cipher, which means it will take 128 bits of text and change it into another 128 bits of encrypted data. If the message is more than 128 bits, it is divided into blocks of 128 bits and encrypted one by one. If the block size is not entirely divisible by 128 bytes, it appends 0 to the end of the previous block and performs the same encryption.

1.5. Hopfield Neural Network (HNN)

A HNN [9] is a single layered and recurrent network [10] that has entirely connected neurons i.e., each neuron has an association with other neurons. If there are two neurons named as i and j, then the connectivity weight w_{ij} lies between them and is symmetric in nature $w_{ij} = w_{ii.}$ HNN shows great potential in the applications of life science and engineering, such as association of memory, medical imaging and supervised learning[10]. Hopfield neural network model present in Figure 4.

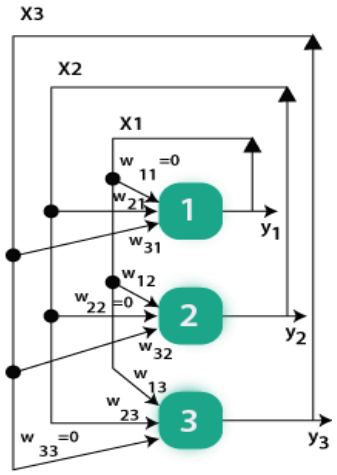

Fig. 4. Hopfield Neural Network representation

2. Related Work

Zhong *et al.* [11] discussed the issue of exchanging anonymous secret keys. The threshold value in this framework is unbounded. Secure interaction are not necessary for this strategy. Gupta *et al.* [12] Shamir's method and the tree parity machine were used to offer a secure image sharing algorithm. The plan operates in two stages. Shamir's secret technique is used in the first section to produce secret shares, and the tree parity machine is used in the second section to encrypt shares. Wang *et al.*[13] suggested a Hopfield neural network-based authentication technique for computer security. Mandal *et al.* [14] suggested a one-layer Hopfield neural network-based secret sharing key distribution. Hopfield networks create an output-input loop. When the randomly generated input string equals the output, the intermediate iteration key is selected. This cycle makes the proposed method less susceptible to threats because of hopfieldneural network.

Narad *et al.* [15] offered a security plan for routine activities including secure money transfers and secure message transmission through unreliable channels. Shamir secret share and backpropagation neural networks are the foundations of the proposed technique (n, n) for secret sharing and group authentication. Dorokhin *et al.* [16] suggested a tree parity machine-based safe 512-bit key distribution mechanism between two authorised parties. The suggested method

conducts a more thorough security examination using a tree parity machine. Santhanalakshmi *et al*. [17] a neural cryptography-based efficient safe group key distribution technique. There is no requirement for intermediary trustworthy parties in this strategy. This plan is reliable, safe, and adaptable. The revealing of secret keys does not require encoding or decoding.

Deng *et al*. [18] developed a probabilistic signature-based visual cryptography access technique for structures utilising ANN. Kishimoto *et al.* [19] suggested a method for recreate the secret key without knowing the identity of secret shares, an anonymous secret sharing technique was presented. Consider a more stringent lower bound in this article, k = 2. Midaguillermo *et al.* [9] fundamental rebuilding for a cryptographic anonymity-offering secret key sharing mechanism was addressed.

Priyanka *et al*.[20] recommended secret sharing plans to maintain the secret's secrecy. In the reconstruction phase, the satisfiability module theory (SMT) solver needs 't-1' shares. Here, satisfability is checked using the SMT solver. Kumar *et al*.[21] a threshold polynomial equation was used to create a secret sharing system. Shares are created and transmitted immediately on the channel during the initialization phase. to increase the effectiveness and security of the two-level encryption used. The decryption level employs a multilayer feedforward (MLFF) backpropagation neural network to offer security and effectiveness. Use MATLAB's 'train rp' function for faster neural network training.

Lake *et al*.[22] proposed a method based on the threshold based secret sharing for the security of IOT devices.Secret sharing based on the threshold value whicgh decides the regeneration processs of secret share at the time of execution. Security of this method based on the threshold value. Balasubramanian Prabhu *et al*. [23] suggested a new Chinese remainder theorem based highly secure strategy for the protection of data at sever and also data accesed by autthentcated users. For accessing the secure server's data, a new key management method based on CRT is suggested. Singhal *et al*. [24] show established centralised and decentralised crawling strategies using relocating mobile agents. The suggested plan offers suggestions for lowering additional network overhead.

Stefania Caputo *et al*. [25] presented a broad access framework for CRT-based secret sharing. In order for higher levels secret key generation utilize the lower-level secret share. Secret key divided in a hierarchical structures in such a way only one secret is used in each level of a multilayer secret sharing (MTSS) scheme. Verma *et al.* [26] When a shareholder is dishonest and employs several secret shares in the multilevel organisation, the concept of protection employing CRT is beneficial. Every participant is divided into different set of stage, and each stage has a dynamic threshold prameter. Reconstruction of key is carried out only sufficient number of shares are available based on the threshold value.

Hong Zhong *et al*. [27] discussed the major issue of secret key sharing that is anonymized.Author present a new anonymously strategies focused on BP Artificial Neural Network because the pre-existing limits and ineffective creation of ideas of all kinds are shortcomings. The concept is simple to put altogether, and the restoration works well. It has no boundaries for the cut-off variable t and is an ideal (t, n) threshold technique. Additionally, our system has verified characteristics and doesn't require a secure communication. for upcoming plans, design for more powerful BP Artificial Neural Networks [28] to boost the efficiency of system and create new innovative techniques for creating effective anonymity secrets techniques.

Marcin Niemiec *et al*.[29] A novel and prospective method based on neural networks is proposed for error correction in quantum cryptography. The author gives the results of multiple computations with varied input variables and analyses the security validation of this approach. Neural networks with just limited coordination, which is usual for quantum cryptography error margins, were used in the testing phase. The effectiveness of passive attacks was demonstrated by contrasting the results with situations produced by neural networks using arbitrarily given variables.

Mayank Gupta *et al*. [30] Secret sharing's goal is to transmit sensitive information without making it available to the general public. In this study, the author proposed a trustworthy neural cryptographic mechanism for exchanging private portions of an image made using Shamir's method between two or more people. Neural encryption is a new source of public key encryption techniques that are not dependent on pure mathematics and also have less computational and storage complexity. Using neural encryption, a shared secret key can be established between two or more participants. Our main goal is to transmit private data across a public network while utilising the least amount of computing resources possible. When using neural encryption, both parties receive the same input and output sequence.

Smita Jhajharia *et al.*[31] suggested combining machine learning with a Genetic Algorithm (GA) to generate keys. It was demonstrated that GA in PRNGs for startup of input variables in ANN successfully handled the problem of identifying the random number produced by traditional PRNGs, which left it vulnerable to intrusions following the detection of patterns in these random variables. In order to ensure security and high key unpredictability, the GA PRNG is "released" once per cycle. According to analysis, a subsequent random variable created could be predicted from the previous random value. Furthermore, it is impossible for all values to have an equal weight. The key is made by constantly combining the first phrase.

3. Problem Statement

Mobile agent migration [32] has been one of the major concerns in major organizations that transfer sensitive information from sender to the receiver. There can be attacks on confidential data by the attackers which might result in data breach and loss. Leak of confidential data can be a very big loss to organizations, users as well as third parties. To ensure secure transfer of mobile agents, multiple techniques have been developed. Secure information is encrypted with symmetric key cryptography and receiver retrieve the secure information using decryption with same key.There are

several advantages as well as disadvantages. There are many cryptographic method that generate privtae key from both the sender's and the receiver's end. Some of these techniques used are, private key cryptography [33] and public key cryptography [16]. Private key cryptography utilize only one key for the encoding and the decoding purpose. The primary problem with private key encryption algorithm is that, the both sender, receiver share private key over some transmission media. If the medium of transmission has vulnerabilities, interceptors can now easily compromise the key and decode the encrypted text. But as the method of public key cryptography utilize two difftent keys for both the encoding and the decoding purpose, this technique becomes quite slow as compared to other processes and complex to implement practically. Thus, this technique is not fruitful for the decryption of heavy and bulk messages as speed is a crucial factor when it comes to communication between two channels. So, through this research paper, the aim to use Hopfield neural network to generate secret key and encrypt the data. One of the biggest advantages of using HNN is that there is no transmission of secret key from sender to receiver. Using one HNN [13] (used for sender's end) output vector crete a secret key for encoding and decoding. For creation of identical output vector, the additional network (used at the receiver's end) is synchronised with the sender side network. To create the secret key among users, output may be helpful. Private key cryptography is used in this manner, however there is a caveat: the secret key is not transmitted.

4. Proposing Modeling

Proposed security model of mobile agent security shown in Figure 5. According to this model, first messages is encoded into encoded text after that our encoded text is encrypted with AES [8]. We use Hopfield Neural Network to generate secret keys for both the sender's end as well as the receiver's end. We pass the same input vector and the weight vector from both the ends to HNN, which, as a result, generates the identical output's vector. This vector is used to generate the same secret key from both the ends. This key is passed through the Advanced Encryption Standard (AES) algorithm.

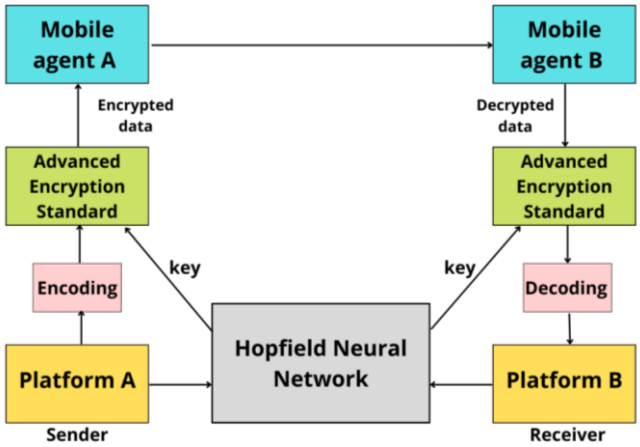

Fig. 5. Secure Mobile Agent Transfer Model

AES keys come in three types of sizes: 128, 192, and 256 bits, and the longer the key, the more secure the process of encoding . Byte Substitute operation, Shifting Rows operation, Mixing Columns operation, and Addition Round secret Key operation are the four operations that make up this algorithm. Our 128-bit input text block will be represented visually on a 4 by 4 matrix, with each location representing 8 bits from our input block. Figure 6 represents the Encruyption of advanced encryption algorithm. 128 bit input is consider for encryption process. Number of rounfd depend on the choice either 10,12 or 14. Output is also 128 bit. Same as decryption process 128 bit of cipher applied on the input of Decryption algorithm. All the steps will execute in reverse order of encryption process.

4.1 Algorithm for Training-

Moving ahead to second level of security, here is the working of HNN to generate the output vector on both the ends: Here, we use Discrete Hopfield Network which gives output in two ways:

1. Binary (in form of 0/1)
2. Bipolar (in form of -1/1)

These weights are symmetrical in nature and have the following characteristics:

$$w_{ij} = w_{ji} \tag{1}$$

$$w_{ii} = 0 \tag{2}$$

[x_1 to x_n] - The n neurons' input is given

[y_1 to y_n] - n neurons' output is produced.
W_{ij} - weight corresponding to the ith and jth neuron connections that were made..

In order to save the input set, S(p) [p = 1 to P], where S(p) = S1(p) … Si(p) … Sn(p), the weight matrix can be expressed as:

$$w_{ij} = \sum_{p=1}^{p}[2s_i(p) - 1]\big[2s_j(p) - 1\big]\big(w_{ij}\ for\ all\ i \neq j\big)\text{(For binary patterns)} \quad (3)$$

$$w_{ij} = \sum_{p=1}^{P}\big[s_i(p)s_j(p)\big]\ (where\ w_{ij} = 0\ for\ all\ i = j\ )\ \text{(For bipolar patterns)} \quad (4)$$

Steps:
Step I - Initialize the weights (w_{ij}) so as to store patterns.
Step II - For every such input vector y_i, perform the steps III to VII.
Step III - Create a network with activators equal to the input vector x.

$$y_i = x_i\text{: (}for\ i = 1\ to\ n) \quad (5)$$

Step IV - For every vector y_{in}, perform the steps V to VII.
Step V - Find the input of the network y_{in} using the following equation-

$$y_{in} = x_i + \sum_j^n y_j w_{ji} \quad (6)$$

Step VI – The output is computed with the help of activation function given below over the input (total).

$$y_i = \begin{cases} 1 & if\ y_{total} > \emptyset \\ y_i & if\ f\ y_{total} = \emptyset \\ 0 & if\ y_{total} < \emptyset \end{cases} \quad (7)$$

(Here Øi (threshold) is usually 0)

Step VII – This output obtained in step vi is given as feedback to other units in the network to update the activation vectors.

Step VIII – In this step the convergence of the network is tested by satisfying the condition of threshold limit.

5. Result and Analysis

Implementation of proposed model for secure key generation and regeneration done using Java. The reason for using Java is that it is platform independent, and thus the system can be run across multiple platforms. Moreover, Java is an object-oriented programming language, which makes the code structured and well formatted to understand. Java uses multithreading, hence there can be multiple threads that can be run simultaneously in the program. As the mobile agent system involves key transfer that contains sensitive information, Java provides security to data by following principles of OOPs like abstraction and encapsulation. Hopfield neural network trained as desired value of key. After learning of Hopfield neural network code saved and communicate to different platform on which particular mobile agent execute assigned activity after successful authentication of agent.

5.1 Case Study

The following are the implementation's steps: -

Step 1 – Input to the system is taken as original message and receiver user id as given: original message = “Secret message for receiver” receiver_user_id = 5

Step 2 – In this step key is provided through an interface. Assume key = “1100” (input).

Step 3 – Encryption is done using AES algorithm by taking the original message and receiver_user_id. The encryption takes place as follows: encrypted key = AES_FUNCTION (original message + “*” + receiver_user_id)= “pr90beU70XUJsU6K86JL/DHHU09kkGRqxFetZI5QDFE=” This generates the encrypted key.

Step 4 – Bipolar array is calculated using the formula: if(input[i]==0) bipolar[i]=-1 else bipolar[i]=1

Step 5 – The weight matrix is calculated using:
weight[row][col]=bi[row]*b[col], wight[row][row]-=1

Step 6 – The training matrix is calculated using weight matrix.
training matrix[row][col] +=weight[row][col]

Step 7 – In this step the HNN model is implemented to find out the two arrays i.e. pattern [], wt[] in the next two steps.

Step 8 - The w_t matrix is the same as the training matrix that we have calculated earlier. w_t [row][col]=training matrix[row][col]

Step 9 – The pattern array is calculated using the formula given below:

If (input[i]==0) pattern[i]=false **else** pattern[i]=true

As input = "1100", pattern= [true, true, false, false]

Step 10 – After this step of calculating pattern array, the activation function is implemented using w_t matrix. Square matrix w_t matrix uytilizing for input of single layer neural Network. The neurons are initialised using these values for weights.

Step 11 - The threshold Function is utilized for bounding the outcome of neuron. Hyperbolic tangent (tanh) function utilize as threshold function.

Step 12 – Save the results obtained following the implementation of HNN in the key_in_final variable.

Step 13 - The variable value key_in_final is then passed to the AES function, which thereby checks if the request_user_id is correct or not.

Step 14 - If the user_id is correct, authentication is success, and hence the decrypted message is shown. If incorrect, authentication is failed and the message is not shown.

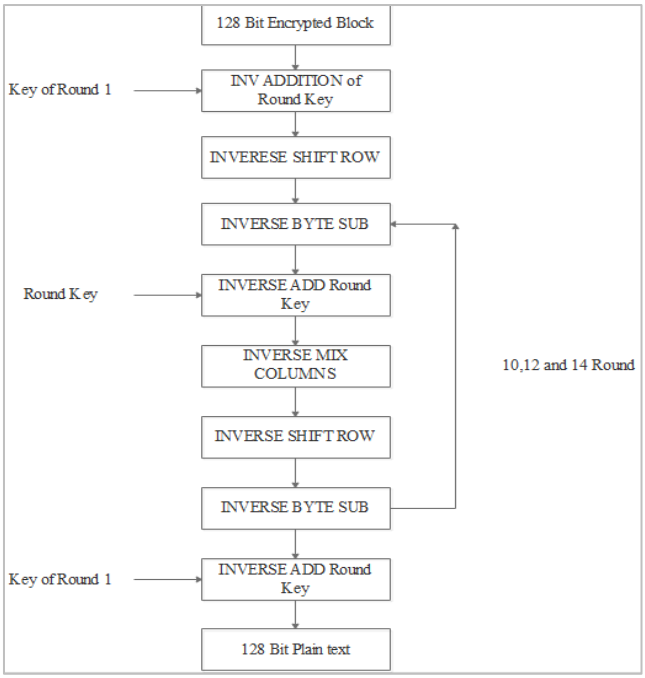

Fig. 6. Encryption in AES

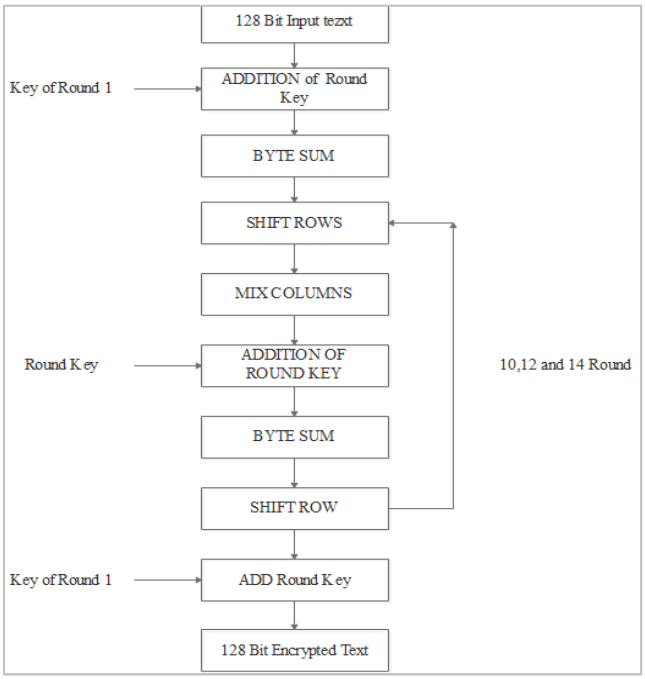

Fig. 7. Decryption in AES

Figure 8 depicts the execution of the suggested system in which successful authentication has been achieved. Prior to execution or resource access, authentication is a step that must be completed in order for any fake mobile agent to access any platform's resources. In Figure 9 authentication is not successful for mobile agent.

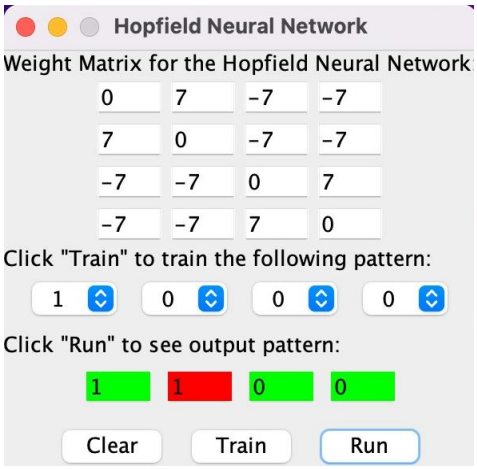

Fig. 8. HNN Implementation (Authentication successful)

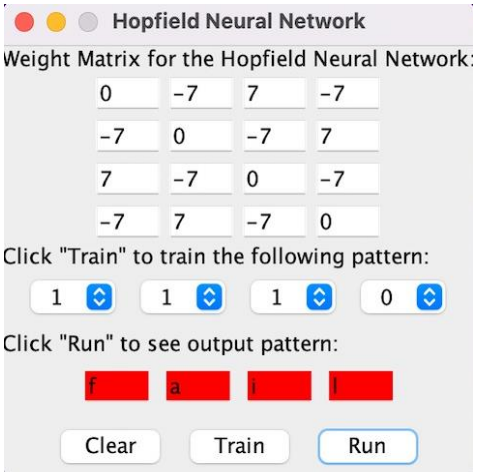

Fig. 9. HNN Implementation (Authentication failure)

In proposed methodology the key generation is done by using machine learning approach (Hopfield neural network). In symmetric key cryptography three is a requirement of secure key sharing among n number of mobile agent. But sharing of secret key in open environment is not secure. One major advantage of using machine learning (Hopfield neural Network) no need to sharing secret key in malicious environment. Each and every mobile agent generate the secret key on particular platform for authentication. After the implementation of proposed methodology, it was observed that the tuun arornd time for generation of key taken using HNN is optimal as compared to existing models as present in Table 1.

Table 1. Comparison of time reqiured by Hopfield neural network, Multilayer perceptron network and Recurrent neural network

Models	Number of iterations			
	1	10	100	1000
HNN	3 ms	217 ms	470 ms	1101 ms
Multilayer Perceptron	50 ms	486 ms	603 ms	1440 ms
Recurrent neural Network	357 ms	362 ms	439 ms	1235 ms

In Figure 10 shown the graphically presentation of Comparison of time taken by Hopfield neural network, Multilayer perceptron network and Recurrent neural network. It was observed that Hopfield neural network if optimal approach.

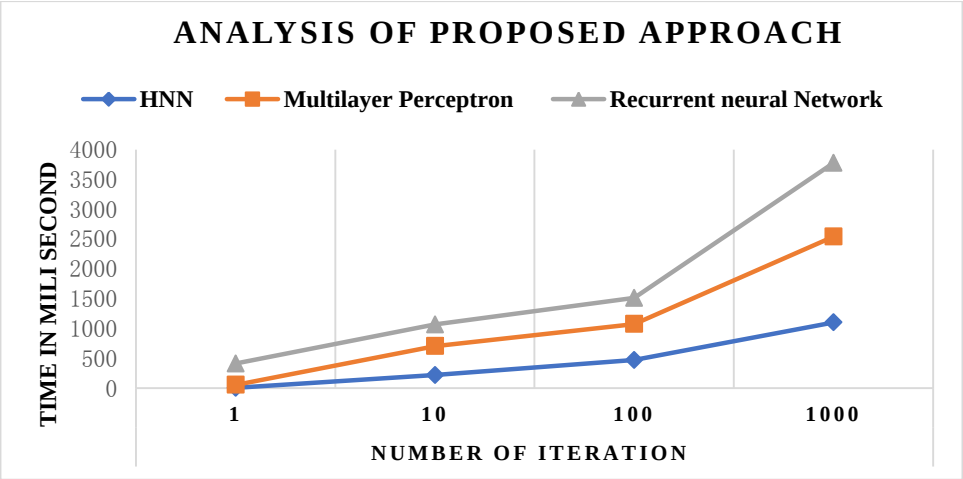

Fig. 10. Comparison between Hopfield neural network (blue line), Multilayer perceptron network (red line) and Recurrent neural network (green line)

6. Conclusion and Future Scope

The HNN, a type of single layer perceptron network, is utilized for generation of secret key at the time execution of mobile agents on host. The outcome from any intermediate channel can be use as the private key because HNN has the ability to generate cycles of the inputs and the outputs, and encoding and decoding function share only sender and receiver. So the chances of attack on mobile agent and platform vesry less and the key generated from HNN is then used with AES to encrypt all the information which makes our data super secure and third level of security is then provided by encoding which makes data non readable without know the correct mapping of characters. Due to this reason the probability of attack and decryption of sensitive information becomes quite low. In future researcher can develop higher optimal technology for key generation better than HNN methodology, and also design another encryption and decryption methodology which better than AES.

References

[1] A. Wagner, "Mobile agent," no. October, p. 288, 2004, doi: 10.1145/977091.977131.
[2] P. Bagga and R. Hans, "Applications of mobile agents in healthcare domain: A literature survey," *Int. J. Grid Distrib. Comput.*, vol. 8, no. 5, pp. 55–72, 2015, doi: 10.14257/ijgdc.2015.8.5.05.
[3] C. Zrari, H. Hachicha, and K. Ghedira, "Agent's security during communication in mobile agents system," *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 17–26, 2015, doi: 10.1016/j.procs.2015.08.100.
[4] C. J. Su and T. W. Chu, "A mobile multi-agent information system for ubiquitous fetal monitoring," *Int. J. Environ. Res. Public Health*, vol. 11, no. 1, pp. 600–625, 2014, doi: 10.3390/ijerph110100600.
[5] R. A. Martins, M. E. Correia, and A. B. Augusto, "A literature review of security mechanisms employed by mobile agents," *Iber. Conf. Inf. Syst. Technol. Cist.*, no. December 2013, 2012.
[6] H. Idrissi, E. M. Souidi, and A. Revel, "Security of mobile agent platforms using access control and cryptography," *Smart Innov. Syst. Technol.*, vol. 38, no. May, pp. 27–39, 2015, doi: 10.1007/978-3-319-19728-9_3.
[7] W. S. Hsu and J. I. Pan, "Secure mobile agent for telemedicine based on P2P networks," *J. Med. Syst.*, vol. 37, no. 3, 2013, doi: 10.1007/s10916-013-9947-2.
[8] U. Upadhyay, P. Kumar, and D. Aggarwal, "Secure migration of mobile agent using AES & secret sharing approach," *Int. J. Emerg. Technol.*, vol. 10, no. 2, pp. 150–155, 2019.
[9] P. Kumar, N. Singhal, and S. Singh, "Anonymous Scheme for Secure Mobile Agent Migration Using Mignotte's Sequence and Back Propagation Artificial Neural Networks," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 13, no. August, pp. 192–199, 2021.
[10] P. Dixit and S. Silakari, "Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review," *Comput. Sci. Rev.*, vol. 39, p. 100317, 2021, doi: 10.1016/j.cosrev.2020.100317.
[11] Q. Xie, Z. Shen, and X. Yu, "Threshold signature scheme based on modular secret sharing," *Proc. - 2008 Int. Conf. Comput. Intell. Secur. CIS 2008*, vol. 2, pp. 442–445, 2008, doi: 10.1109/cis.2008.78.
[12] S. Banerjee, D. S. Gupta, and G. P. Biswas, "Hierarchy-based cheating detection and cheater identification in secret sharing schemes," *Proc. 4th IEEE Int. Conf. Recent Adv. Inf. Technol. RAIT 2018*, pp. 1–6, 2018, doi: 10.1109/RAIT.2018.8389094.
[13] S. Wang and H. Wang, "Password authentication using Hopfield neural networks," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 38, no. 2, pp. 265–268, 2008, doi: 10.1109/TSMCC.2007.913901.
[14] S. C. Satapathy, B. N. Biswal, S. K. Udgata, and J. K. Mandal, "Proceedings of the 3rd international conference on frontiers of intelligent computing: Theory and applications (FICTA) 2014: Volume 2," *Adv. Intell. Syst. Comput.*, vol. 328, pp. 217–224, 2015, doi: 10.1007/978-3-319-12012-6.
[15] M. S. K. Narad, "Group Authentication Using Back-propagation Neural Network," vol. 6, no. 10, pp. 272–278, 2017, doi: 10.17148/IJARCCE.2017.61048.

[16] É. Salguero Dorokhin, W. Fuertes, and E. Lascano, "On the Development of an Optimal Structure of Tree Parity Machine for the Establishment of a Cryptographic Key," *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/8214681.
[17] S. Santhanalakshmi, K. Sangeeta, and G. K. Patra, "Design of group key agreement protocol using neural key synchronization," *J. Interdiscip. Math.*, vol. 23, no. 2, pp. 435–451, 2020, doi: 10.1080/09720502.2020.1731956.
[18] Y. Q. Deng and G. Song, "A verifiable visual cryptography scheme using neural networks," *Adv. Mater. Res.*, vol. 756–759, pp. 1361–1365, 2013, doi: 10.4028/www.scientific.net/AMR.756-759.1361.
[19] W. Kishimoto, K. Okada, K. Kurosawa, and W. Ogata, "On the bound for anonymous secret sharing schemes," *Discret. Appl. Math.*, vol. 121, no. 1–3, pp. 193–202, Sep. 2002, doi: 10.1016/S0166-218X(01)00236-0.
[20] P. Sharma and P. Kumar, "Review of Various Image Steganography and Steganalysis Techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 7, pp. 152–159, 2016.
[21] J. Li, X. Wang, Z. Huang, L. Wang, and Y. Xiang, "Multi-level multi-secret sharing scheme for decentralized e-voting in cloud computing," *J. Parallel Distrib. Comput.*, vol. 130, pp. 91–97, 2019, doi: 10.1016/j.jpdc.2019.04.003.
[22] L. Bu, M. Isakov, and M. A. Kinsy, "A secure and robust scheme for sharing confidential information in IoT systems," *Ad Hoc Networks*, vol. 92, 2019, doi: 10.1016/j.adhoc.2018.09.007.
[23] B. Prabhu kavin and S. Ganapathy, "A secured storage and privacy-preserving model using CRT for providing security on cloud and IoT-based applications," *Comput. Networks*, vol. 151, pp. 181–190, 2019, doi: 10.1016/j.comnet.2019.01.032.
[24] N. Singhal, A. Dixit, R. P. Agarwal, and A. K. Sharma, "A reliability based approach for securing migrating crawlers," *Int. J. Inf. Technol.*, vol. 10, no. 1, pp. 91–98, 2018, doi: 10.1007/s41870-017-0065-0.
[25] S. Caputo, G. Korchmáros, and A. Sonnino, "Multilevel secret sharing schemes arising from the normal rational curve," *Discret. Appl. Math.*, vol. 284, pp. 158–165, 2020, doi: 10.1016/j.dam.2020.03.030.
[26] O. P. Verma, N. Jain, and S. K. Pal, "A Hybrid-Based Verifiable Secret Sharing Scheme Using Chinese Remainder Theorem," *Arab. J. Sci. Eng.*, vol. 45, no. 4, pp. 2395–2406, 2020, doi: 10.1007/s13369-019-03992-7.
[27] H. Zhong, X. Wei, and R. Shi, "A novel anonymous secret sharing scheme based on BP Artificial Neural Network," *Proc. - Int. Conf. Nat. Comput.*, no. Icnc, pp. 366–370, 2012, doi: 10.1109/ICNC.2012.6234550.
[28] P. Hao, "An improved back-propagation neural network algorithm," *Appl. Mech. Mater.*, vol. 556–562, pp. 4586–4590, 2014, doi: 10.4028/www.scientific.net/AMM.556-562.4586.
[29] M. Niemiec, M. Mehic, and M. Voznak, "Security Verification of Artificial Neural Networks Used to Error Correction in Quantum Cryptography," *2018 26th Telecommun. Forum, TELFOR 2018 - Proc.*, pp. 1–4, 2018, doi: 10.1109/TELFOR.2018.8612006.
[30] M. Gupta, M. Gupta, and M. Deshmukh, "Single secret image sharing scheme using neural cryptography," *Multimed. Tools Appl.*, vol. 79, no. 17–18, pp. 12183–12204, 2020, doi: 10.1007/s11042-019-08454-8.
[31] S. Jhajharia, S. Mishra, and S. Bali, "Public key cryptography using neural networks and genetic algorithms," *2013 6th Int. Conf. Contemp. Comput. IC3 2013*, pp. 137–142, 2013, doi: 10.1109/IC3.2013.6612177.
[32] H. Xu, Z. Zhang, and S. M. Shatz, "A security based model for mobile agent software systems," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 15, no. 4, pp. 719–746, 2005, doi: 10.1142/S0218194005002518.
[33] V. Sagar and K. Kumar, "A symmetric key cryptography using genetic algorithm and error back propagation neural network," *2015 Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2015*, no. March, pp. 1386–1391, 2015.

Authors' Profiles

Pradeep Kumar is a Ph.D. student of computer engineering and engineering at Department of Computer Engineering Shobhit Institute of Engineering & Technology (Deemed-to-be University), Meerut, 250110. He has obtained his M.Tech. in Computer Science and engineering Department of Computer Engineering Shobhit Institute of Engineering & Technology (Deemed-to-be University), with first class. He obtained his B.Tech in Computer Engineering and engineering degree from college of engineering Roorkee, India in 2006 with first class.

Dr. Niraj Singhal is Ph.D. (Computer Engineering and Information Technology). He is Fellow and member of several International/National bodies and, reviewer and member of the advisory board for several International/National journals. He has many research publications to his credit in National/ International journals/conferences of repute. He has several years of rich experience of administration, coordinating and teaching at various levels. Presently he is working as Professor in the department of Computer Science and Engineering at Shobhit Institute of Engineering & Technology (Deemed-to-be University), Meerut. His area of interest includes system software, web information retrieval and software agents.

Ajay Kumar, Assistant Professor in the Department of Computer Science & Engineering at JSS Academy of Technical Education, Noida. He has received his M. Tech (Computer Engineering) from YMCA University of Science & Technology, Faridabad, India and B. Tech (Computer Engineering) from University Institute of Engineering & Technology, M.D.U. Rohtak. He has more than 10 years of academic experience. He has contributed 09 Research papers in International Journal and 11 Research papers in International/National Conferences/proceedings and Edited Books. He has added 02 Patents with his name out of which 01 has granted. His areas of research are Machine Learning, IOT & Network Security. He has organized various workshops and FDPs in these areas. He is the life time member of International Association of Engineers (IAENG).

Dr. Kakoli Banerjee, Associate Professor (Computer Science and Engineering Department), JSS Academy of Technical Education, Noida. She graduated from IET, Kanpur – B.Tech (CSE), did her Post Graduation from MNNIT, Allahabad – M.Tech (CSE) and Doctorate (Ph.D) from Shobhit University, Saharanpur. She has 22 years' of experience in academics and in industry. She have published around 45 research papers in international journals of repute indexed in SCI, Scopus, ESCI, Google Scholar, and other eminent databases. Currently, she is working for a funded projects, sanctioned by the Collaborative Research and Innovation Program (CRIP) under TEQIP- 3 by Dr. A. P. J. Abdul, Kalam Technical University Uttar Pradesh, Lucknow. She is a member of IEEE and other professional societies such as ISTE, CSI, etc. She has been a reviewer and guest editor of several reputed journals and books. She has also served in many conferences as session chair. She has two Granted Patent to my name and eight published. She has developed video content for Dr. A.P.J. Abdul Kalam Technical University, Lucknow and Swayamprabha Chanel of Government of India for subjects like Operating System and Database Management System. She has published book - "Industry 4.0: Research Trends, Challenges and Future of AI in Data Science", CRC Press, Taylor & Francis Group and "Decision Analytics for Sustainable Development in Smart Society 5.0", Springer Nature.

How to cite this paper: Pradeep Kumar, Niraj Singhal, Ajay Kumar, Kakoli Banerjee, "An Enhanced Method Utilizing Hopfield Neural Model for Mobile Agent Protection", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.13, No.5, pp. 23-33, 2023. DOI:10.5815/ijwmt.2023.05.03