International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 13 (2021) pp. 192-199

Received: 2 Feb 2021; Accepted: 4 May 2021; Published: 6 August 2021

Anonymous Scheme for Secure Mobile Agent Migration Using Mignotte's Sequence and Back Propagation Artificial Neural Networks

Pradeep Kumar [1], Niraj Singhal [2] and Sukhendra Singh [3]

[1] Shobhit Institute of Engineering & Technology (Deemed-to-be University),
Meerut, India
pradeep8984@jssaten.ac.in

[2] Shobhit Institute of Engineering & Technology (Deemed-to-be University),
Meerut, India
drnirajsinghal@gmail.com

[3] Department of Information Technology, JSS Academy of Technical Education,
C-20/1 Noida, India
sukhendrasingh@gmail.com

***Abstract*: A mobile agent is an autonomous executing small piece of program that can relocate from one host to another in a non-homogeneous network under its own control. Mobile agents are designed to execute certain assigned tasks by the owner. In the life cycle of mobile agents, these pass over many hosts for the execution of tasks. Mobile agent's scheme is widely used in distributed computing because of its dynamic nature, less bandwidth and less computation power. Execution of mobile agents exploits codes; data and state upraise the security issues. Malicious agents can attack mobile agents during the transmission and at the time of execution on the host because mobile agents carry delicate information of owners. The protection of mobile agents and platforms is a sensitive issue. This article focuses on the security issue of mobile agents and platforms. Provide the anonymous secure framework for mobile agent and platform security by using optimize secret key management based on Mignotte's sequence and artificial neural network. In an anonymous secret sharing technique, secret keys regenerated without knowledge of which mobile agents hold which share. That is, in such a technique the secret can be reconstructed from the shares without the identities of mobile agents.**

***Keywords*:** Mobile Agent, Secret Share, Mignotte's Sequence, Backpropagation Artificial Neural Networks.

I. Introduction

A mobile agent [17] is a small process that works automatically on behalf of its owner. Once a mobile agent is generated by the host, it can relocate dynamically from one host computer to another and executes assigned tasks. It is the amendment in the direction of computing and the Internet of Things (IoT). Mobile agent's mechanism is broadly used in fields such as extracting information, military field, cloud computing, etc. In the real-time execution of applications with mobile agents, the prime concern is protecting the mobile agent and platform from malicious mobile agents and hostile environments. The Execution life cycle of mobile agents during the communication is shown in Figure 1.

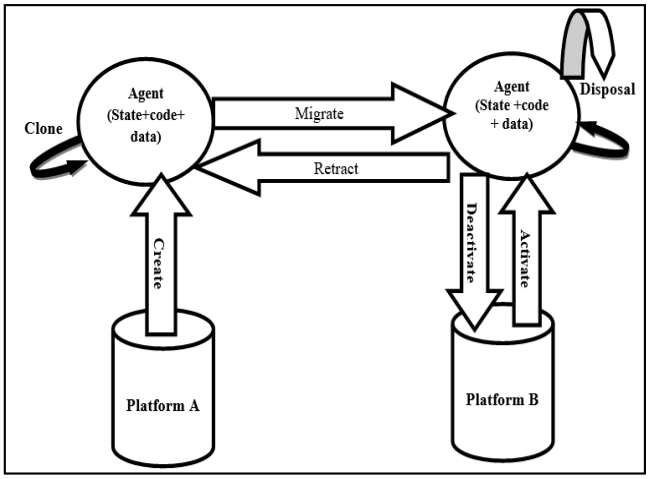

Figure 1. Mobile Agent life cycle

Many threats [18] for mobile agents are as follows: -

a) Mobile Agent versus Platform
b) Platform versus Mobile Agent
c) Mobile Agent versus Other malicious Mobile Agents
d) Other Entities versus both

Mobile agent Security hierarchy is shown in Figure 2. The security of the mobile agent paradigm categorized in to two categories platform security in this mobile agent wants to execute code on the entrusted platform. Another mobile agent security, because mobile agents migrating one host to another host in a malicious environment it is a prime concern. Mobile agent security is further divided into two categories single hope and multi hope. In multi hop further security is divided

into code security, in code security concerns about security measures about code caring by mobile agents. Mobile agents after execution on the platform take some important results and store them with a mobile agent this is another issue security of data. After execution of mobile agent, mobile agents follow either predefined path or free-roaming to returning back to master node.

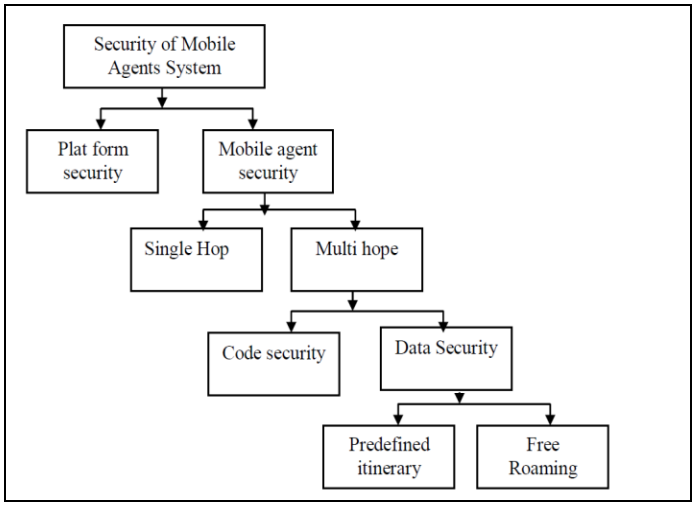

Figure 2. Security of mobile agents

Mobile agents migrate from one machine to another automatically in a hostile circumstance, so the protection of mobile agents and platforms is a crucial issue. Confidentiality, data integrity, and availability of information are the main attributes of information security [18]. Confidentiality means protecting information against unapproved users; Integrity means genuineness of data and information; Availability means that data and information accessible by only genuine entities. For genuine entities, facts should be available. Apart from this security parameter security of agents and platforms against malicious mobile agents is also a prime issue for agent's actions. The principal issue is to design a framework for the security of the mobile agent at the hour of relocation to help secure agreement.

In the past created mobile agents' frameworks, a large portion of the fixation were on choosing the working of mobile agents instead of safety. While some MA framework presents security, the majority of the plans need execution of a viable security for mobile agent's paradigm. Table 1 focusing on the major issues related to attacks on mobile agent model.

	Attacks of Agent during execution on platform	Attacks of Agent during execution on Agent	Attacks of Platform-during execution on Agent
Masquerading	✓	✓	✓
DOS	✓	✓	✓
Unauthorized Access	✓	✓	
Repudiation		✓	
Eavesdropping			✓
Alteration			✓

Table 1. Attack on Mobile Agent Framework

There is a need for a secure mobile agent [18] to support interoperability between the agent's framework and the security of the paradigm.

The protection of mobile agents and its transaction in malicious environments is still in unformed condition. Designing such a type of mechanism that provides security for mobile agents and platforms is a challenging job. At the time of execution, mobile agents use the resources of other machines. So, mobile agents and platforms are open for attack. The prime concern is to protect mobile agents and their assets in distributed computing. One of the conventional methodologies of encryption and decryption is used to resolve the security issue. But the security of the approach depends on the vigor of the secret key utilizing encryption and decoding.

After having a look at the available literature, the problem that mobile agents moving self-ruling in vindictive climates performs tasks on another machine that is not trusted to mobile agents. A *Technique* is proposed here that mobile agent migrates and maintains its security based on secure key and limitless threshold decided by the host. The non-trusted platform or hackers of agent code requires a threshold to access the secret key of the mobile agent. For the protection of mobile agent and platform proposed model is based on Mignotte's Sequence and backpropagation artificial neural network. A secret key for the execution of tasks and authentication of mobile agents has been created by a random number generator. The secret key is partitioned into 'n' number of partial shares based on Mignotte's Sequence. Backpropagation artificial neural network and decided threshold value 't' are used for regeneration of secret key for authentication and execution.

II. Related works

Zhong *et al.*[1]discuss the problem of anonymous secret key sharing. Proposed a new anonymous secret sharing scheme based back propagation neural network. In this scheme, there is no bound on the threshold value. In this scheme, there is no requirement for secure channels. Gupta *et al.*[2] proposed a secure image sharing scheme based on Shamir's scheme and tree parity machine. The scheme works in two parts. In the first part, secret shares are created using Shamir's secret scheme and in the second part, shares are encrypted using tree parity machine.

Wang *et al.*[3] proposed an authentication scheme for computer security based on Hopfield neural network (HNN). Mandal *et al.*[4]proposed a secret share key distribution based on one layer Hopfield neural network. Hopfield networks form a cycle between input and outputs. After the intermediate iteration key is chosen in such a way when the random input string is equal to output. Chances of threat are less in the proposed Hopfield neural-based key generation (HNBNKG) scheme because of this cycle.

Narad *et al.*[5]proposed a security scheme for day-to-day life operations such as secure money transfer and secure message transferring on unauthentic channels. Proposed scheme (n, n) secret sharing scheme based on Shamir secret share and backpropagation neural network for group authentication. Dorokhin *et al.*[6]proposed a secure 512-bit key distribution scheme based on a tree parity machine between two authorized parties. The proposed scheme

performs a more exhaustive security analysis, tree parity machine.

Santhanalakshmi *et al.* [7] proposed an effective secure group key distribution algorithm based on neural cryptography. In this scheme, there is no need for intermediate trusted parties. This scheme is effective, secure and scalable. There is no need for encoding and decoding for the disclosure of secret keys. Deng *et al.* [8] proposes a new verifiable visual cryptography algorithm access structures using pi-sigma(π-σ) artificial neural networks (ANN) based on the probabilistic signature. The proposed article merges two technology neural networks and visual cryptography.

Kishimoto *et al.*[9]proposed an anonymous secret sharing technique, To reconstruct the secret key without knowing which participants hold which secret share. In this article, consider a tighter lower bound k = 2. Midaguillermo *et al.*[10] discussed basic reconstruction for secret key sharing protocol offering cryptographic anonymity. Deng *et al.*[11]discussed the scheme for anonymous secret key sharing schemes with two thresholds by using combinatorial designs including group divisible designs, difference families, and relative difference sets. If 't' and 'w' are fixed, the asymptotic behavior of the minimum size of shares as the number of secrets tends to infinity is also given.

Priyanka *et al.* [12] proposed secret sharing schemes to keep the secret confidential. In a secret sharing technique, a secret is divided into 'n' parts based on the Mignotte's sequence. These shares are distributed to the 'n' shareholders with threshold 't'. At the time of reconstruction phase, it requires 't-1' shares with satisfiability module theory (SMT) solver. Here SMT solver is used to check for satisfiability. Kumar *et al.*[13]developed a secret sharing system using threshold polynomial function. In the initialization phase shares are generated, and shares are transmitted directly on the channel. to enhance the security and efficiency of two-level encryption used. A Multilayer feedforward (MLFF) backpropagation neural network is used at the decryption level to provide security and efficiency. For Faster training of neural networks apply the 'trainrp' function of MATLAB. The accuracy of the proposed system is better than the previous model.

Lake *et al.* [14] proposed a secure and robust model of secret share confidentially in IoT-based systems. This model uses Threshold Secret Sharing (TSS) to divide the secret into different shares and assign it to different IoT devices so that corruption of a single IoT device will not affect the security of the whole system. Ksavin *et al.*[15]proposed a novel CR-based secure for data security and access to the data by authenticated participants. The proposed scheme uses a new Chinese remainder theorem (CRT) for the security of data. A Novel CRT-based Key management technique is proposed for accessing the data from the server. In the proposed scheme new formulas are used for encryption and decryption. Singhal *et al.* [16] present traditional centralized and distributed crawling schemes with migrating mobile agents. The proposed scheme advises how to reduce extra overhead on networks. Kumar *et al.* [17] proposed a framework for secure mobile agent migration based on a tree parity machine and new tiny encryption algorithm.

Meng *et* al. [19] suggested a general access structure for secret sharing using CRT. It divided the secret in a hierarchical structure in such a way higher level can access the lower-level share to regenerate the secret. In a multilevel secret sharing (MTSS) scheme only one secret is used in each level. Verma *et al.* [2] Proposed the idea of security using CRT which is helpful when a shareholder is not honest and uses multiple secret shares in the multilevel group. All the participants are categorized into various levels and every level has a dynamic threshold value. Reconstruction is done when enough shares are available.

III. Preliminaries

In this section, we present some crucial foundations of secret sharing scheme

A. Migtone sequence

Choose 'n' positive integer in such a way $p_0<p_1<p_2<p_3\ldots\ldots\ldots<p_n gcd\ (pi, pj) = 1$ *for all* $i\neq j\leq n$

$$\prod_{i=1}^{t} p_i > (p0 + 1).\prod_{i=1}^{t-1} p_{n-t+i+1} \quad (1)$$

B. Backpropagation neural networks

In 1969 training techniques for Multilayer perception invented by Bryson and Ho. Backpropagation neural network is supervising learning techniques based on the Widrow-Hoff learning rule. In general, Backpropagation neural networks organized have three main layers (As shown in Figure 3), the input layer, hidden layer and output layer. Working of feed-forward phase in Backpropagation algorithm based on gradient descent rule. The backpropagation neural network initializes with random small weights and reduces the error by updating weight to achieve the target. Reduce error is based on the least square method.

Input layer neurons are always greater than the number of neurons in hidden layers.

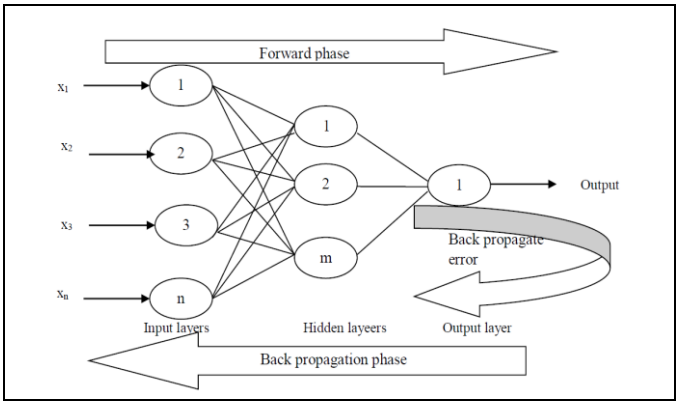

Figure 3. Backpropagation Neural Network

C. Main stages of Backpropagation neural network

Backpropagation Neural Network (BPN) consisting of different main phases discussing as follows.

1) Initialization of network: In the first stage of backpropagation, the neural network initializes the number of neurons in input layer, the number of neurons in hidden layer and the output layer. Initialize the weight of each layer and base value using random function between 0 and 1.

2) Forward phase: In the forward step, apply input with randomly selected weight and compute output with sigmoidal

activation function. There are 'n' input neurons at the input layer and 'm' hidden layer neurons.

Input to hidden layer neurons

$$=[x1x2x3..xn]\begin{bmatrix} w11w12w13..w1m \\ w21w22w23..w2m \\ ..... \\ wn1wn2wn3..wnm \end{bmatrix}=$$

$$\begin{bmatrix} x1*w11+x2*w21+x3*w31..+xn*wn1 \\ x1*w12+x2*w22+x3*w32..xn*wn2 \\ ..... \\ x1*w1m+x2*w2mx3*w3m..xn*wnm \end{bmatrix}+\begin{bmatrix} b1 \\ b2 \\ .. \\ bm \end{bmatrix} \quad (2)$$

b_1, b_2 b_3,b_4,…….b_m are bias values for hidden neurons.

*3) Activation function***:** Activation function also known as transfer function**.** By using activation function in backpropagation neural networks covert linear regression into non-linearity regression. The purpose of the activation function is to make the neural network to learn and do complicated real-time tasks.

a) Sigmoid: Sigmoid activation function known as logistic activation function shown in Figure 4. **s**igmoid activation function is using in Backpropagation neural network.

$$\sigma(x)=\frac{1}{1+e^{-x}} \quad \text{where} \quad 0<\sigma(x)<1 \quad (3)$$

b) Differentiation of sigmoid function:

$$\frac{d}{dx}\left(\sigma(x)\right)=\sigma(x)*(1-\sigma(x)) \quad (4)$$

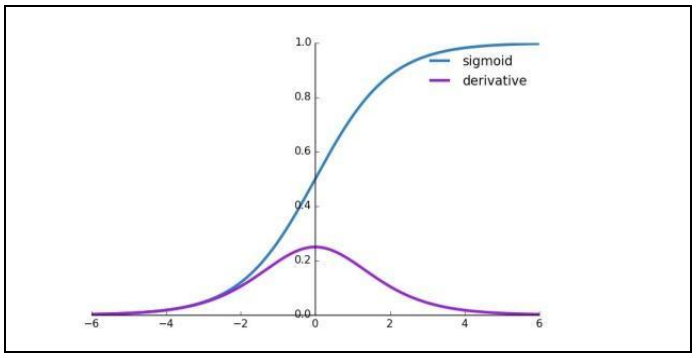

Figure 4. Sigmoid activation function

*4) Error function***:** After the forward phase calculate error variation between the real output and desired output.

$$\text{Error}=\begin{cases} 0, & if output = Desiredoutput \\ 1/2(output-Desiredoutput)^2 \end{cases} \quad (5)$$

Error is always positive because the square of any value is always positive.

5) Back propagation (Minimization of error): The main target is to minimize the error. Desired output is constant, so we need to update the output by updating the weight using gradient descent. Gradient descent calculates the gradient of error with respect to respective weight.

$$\text{New weight } W_x=\text{old weight}-\alpha\left(\frac{\partial error}{\partial wx}\right) \quad (6)$$

Where W_x = old weight

α= Training rate of algorithm

$\left(\frac{\partial error}{\partial wx}\right)$ = Derivative of the error function with respect to weight

IV. Proposed Approach to countermeasure Agent security

An approach based on Mignotte's Sequence and backpropagation neural network (BPN) has been proposed for the security of mobile agents and platforms shown in Figure 5. In this framework, a secret key for the execution of tasks and authentication of mobile agents is generated at a mobile host using a random number generator with lower bound and upper bound. Secret key of execution further divided into 'n' number of partial keys based on Mignotte's Sequence using algorithm 1and algorithm 2. We have a partial key {k1, P1}, {k2, P2}, {k3, P3}, {k4, P4}… {kn, Pn}. In this scheme there is no limit on threshold value'. For every transaction it is decided by a platform which is independent of 'n' and t<n. There are n_{C_t} authentic subset with cardinality of 't'. In the next phase use backpropagation neural network algorithm with 't' input and with random small weight of network, in hidden layer number of neurons 2/3 of no of neurons in input layer and one output neuron. Trained neural network on the basis of Algorithm 3 and send this trained program to the sender platform. Finally at the receiving platform at which mobile agent want to perform their process on behalf of user reconstruct the secret key by downloading the trained neural network and apply 't' authentic subset of mobile agent's partial key as an input and generate secret key on basis of Algorithm 4.

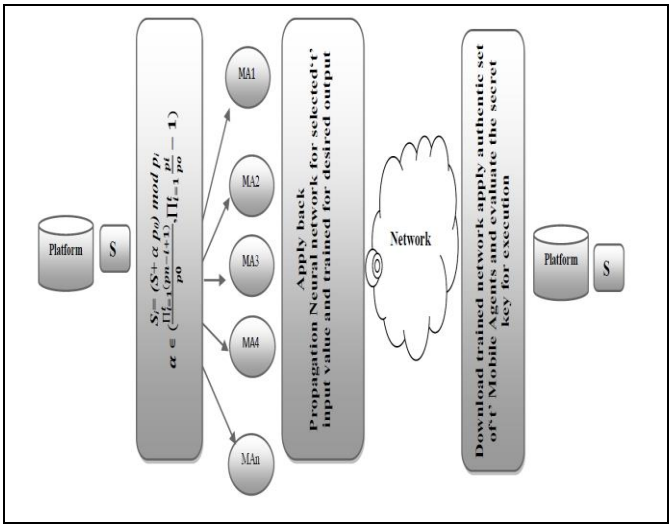

Figure 5. Anonymous Scheme based on Mignotte's Sequence and Backpropagation Neural Network.

A. Algorithm 1: Algorithm1 explained about the initialization procedure how to select 'n' prime numbers sequence as number represents Mignotte's Sequence.

1) Initialization: Host (User) select n positive integer in such a way

a) Select a secret key S ($0 \leq S < p_0$) *for* authentication of mobile agent during execution and transformation of code.

b) Select positive integer $p_0<p_1<p_2<p_3\ldots\ldots\ldots<p_n$

c) gcd (pi, pj) =1 for every $i \neq j \leq n$

d) $\prod_{i=1}^{t} p_i > (p0 + 1).\prod_{i=1}^{t-1} p_{n-t+i+1}$

B. Algorithm 2: Algorithm 2 is share generation, describe the process how to create 'n' partial share with respect to Magnate's Sequence.

1) Share creation

a) $S_i=(S+\alpha p_0)\ mod\ p_i$ public share of every Mobile agent $0\leq i\leq n$

b) $(P_{n-t+2} * P_{n-t+3} \ldots\ldots P_n) < (S+\alpha p_0) < (P_1 * P_2 * P_3 \ldots\ldots P_t)$

C. Algorithm 3: Back rogation algorithm is using for training of neural network. Input for BPN algorithm selected from the generated partial share shown in algorithm 3.

1) Backpropagation algorithm (BPNA):

a) create n_{C_t} authentic set of mobile agents. 't' is a user-defined threshold value. Host users select a Backpropagation Neural network 'n' input layer (n=t), number of hidden layer neurons '2/3' of input layer neuron and one output.

b) According to different combinations of mobile agents, hosts trained the backpropagation neural networks. 't' is the number of inputs applied to a neural network and 'S' is the desired value.

c) Mobile hosts terminate the training of backpropagation until the difference between the calculated output and the desired output is less than threshold value.

d) After training, the backpropagation neural network host saves the neural network program and sends it to the receiver platform in a secure manner.

D. Algorithm 4: Describe the reconstruction of the secret key by using a trained neural network.

1) Reconstruction of secret

a) Share of authentic subset revived at another platform and access the neural network program in a secure manner and apply the input to the neural network.

b) After taking the output by neural network

c) If (MSE< threshold value)

d) {Received authentic key for execution.

e) }

f) else

g) {

h) Subsets apply on neural network are not authentic.

i) }

Backpropagation Artificial neural network is using for the training of neural network by putting 't' input selected out on 'n' secret generated by algorithm 2, at input layer of neural network. Trained the neural network according to desired output and broadcast this trained program to every platform. As mobile agents reach at the platform for execution of an assigned task, before execution, it is compulsory to authenticate mobile agents to execute task. For authentication, we apply 't' input values to downloaded trained artificial neural networks. If the error less than the threshold value the authentication of mobile agents successfully competed in O(1) time. Otherwise, authentication of mobile agent is failed.

To explain the algorithm 1, algorithm2, algorithm3 and algorithm 4 considering some easy examples as follows:

Case: 1 Let randomly selected share by the platform for execution of task is 'S=69' using initialization algorithm and partial share generated by using algorithm 2, 69 key divided in to 7 partial share shown in Table 2. List of prime number: $p_0, p_1\ p_2 \ldots\ldots p_7$ *are* [71, 223, 227, 229, 233, 239, 241, 251] and n (no. of shares) =7, P_i and P_t choose in such a way $P_{n-t+2} * P_{n-t+3} \ldots\ldots P_n < P_1 * P_2 * P_3 \ldots\ldots P_t$,pi: 2700984697, pk: 1026471779, α choose by using these condition$P_{n-t+2} * P_{n-t+3} \ldots\ldots P_n < (S+\alpha p_0) < P_1 * P_2 * P_3 \ldots\ldots P_t \alpha$: 59 (ki, Pi) and t_1=4 for initialization.

Secret	K1	K2	K3	K4	K5	K6	K7
69	21	172	136	64	195	161	242

Table 2. Shares generated by algorithm 2.

After generation of partial secret apply back propagation algorithm in this example using 3 level artificial neural networks at input layer apply three random selections from '7' generated partial share and trained neural network in this case using t_2 =3. Number of ways select 3 different pair out of 7 values is equal to 7_{C_3} = 35. These are 35 pairs.

[**(21, 172, 136),** (21, 172, 64), (21, 172, 195), (21, 172, 161), (21, 172, 242), (21, 136, 64), (21, 136, 195), (21, 136, 161), (21, 136, 242), (21, 64, 195), (21, 64, 161), (21, 64, 242), (21, 195, 161), (21, 195, 242), (21, 161, 242), (172, 136, 64), (172, 136, 195), (172, 136, 161), (172, 136, 242), (172, 64, 195), (172, 64, 161), (172, 64, 242), (172, 195, 161), (172, 195, 242), (172, 161, 242), (136, 64, 195), (136, 64, 161), (136, 64, 242), (136, 195, 161), (136, 195, 242), (136, 161, 242), **(64, 195, 161),(64, 195, 242),** (64, 161, 242), (195, 161, 242)]

Select any three combinations as an input of the back propagation neural network having three input neurons, two hidden layer neurons and one output neuron. Table 2 represents the three set of training with desired output. Desired output value S=69 steps in during training are 1000, set the precision of 0.002, and choose other parameters according to the back-propagation algorithm. After training of neural network, save this network program with final weight. And sent to all receiver platforms then platforms apply selected three inputs to this program and take output as shown in Table 3.

S. No	Apply input	Desired output in program	Output by simulation
1	[64,161, 242]	69	69.11095
2	[21, 172,136]	69	69.242192
3	[64, 195, 161]	69	68.625873

Table3. simulated output

In this approach there is no limit on threshold value. If we choose t=3, only 3 parameters will generate correct output in simulation.

Case: 2 Let randomly selected share **'S=85'**and partial share shown in Table 4. List of prime number: [89, 269, 271, 277, 281, 283, 293, and 307] and N (No. of shares) =7, pi: 5674239463 pk: 2265595837α: 25 (ki, Pi)

Secret	K1	K2	K3	K4	K5	K6	K7

85	158	142	94	62	46	259	161

Table 4. Shares generated by algorithm 2

Select any‘t’ value from the given share. Let us consider here t=4, nom of way select 4 different pair out of 7 value is equal to 7_{C_4}= 35.

[(158, 142, 94, 62), (**158, 142, 94, 46**), (158, 142, 94, 259), (158, 142, 94, 161), (158, 142, 62, 46), (158, 142, 62, 259), (158, 142, 62, 161), (158, 142, 46, 259), (**158, 142, 46, 161**), (158, 142, 259, 161), (158, 94, 62, 46), (158, 94, 62, 259), (158, 94, 62, 161), (158, 94, 46, 259), (158, 94, 46, 161), (158, 94, 259, 161), (158, 62, 46, 259), (158, 62, 46, 161), (158, 62, 259, 161), (158, 46, 259, 161), (**142, 94, 62, 46**), (142, 94, 62, 259), (142, 94, 62, 161), (142, 94, 46, 259), (142, 94, 46, 161), (142, 94, 259, 161), (142, 62, 46, 259), (142, 62, 46, 161), (142, 62, 259, 161), (142, 46, 259, 161), (94, 62, 46, 259), (94, 62, 46, 161), (94, 62, 259, 161), (94, 46, 259, 161), (62, 46, 259, 161)]

Select any four combinations as an input of the back propagation neural network having 4 input neurons, 2 hidden layer neurons and 1 output neuron. Table 4 represents the three set of training with desired output. Desired output value ‘S=85’ steps during training are 1000, set the precision of 0.002, and choose other parameters according to the back-propagation algorithm. After training of neural network save this network.

Program with final weight and sent to all receiver platforms then platform apply selected three inputs to this program and take output as shown in Table 5.

S. No	Apply input	Desired output in program	Output by simulation
1	[158, 142, 46, 161]	85	85.729237
2	[158, 142, 94, 46]	85	85.817983
3	[142, 94, 62, 46]	85	85.494107

Table 5. Simulated output

Case 3: Suppose some malicious mobile agent wants to access the secret key by using unauthenticated share. Table 5 represents the three set of training with desired output. Select randomly three set [42, 84, 60, 45], [142, 84, 52, 46] and [140, 64, 52, 56] and apply on artificial neural network simulated output shown in Table 6.

S. No	Apply input	Desired output in program	Output by simulation
1	[42, 84, 60, 45]	85	83.406762 (wrong Output)
2	[142, 84, 52, 46]	85	82.355713 (wrong Output)
3	[140, 64, 52, 56]	85	84.32554(wrong Output)

Table 6. Simulated output for unauthenticated share

If an unauthorized mobile agent wants to access the key of execution cannot reveal the secret key. S. No 1, 2 and 3, using three different sets of cardinalities of four, unauthorized mobile agents and apply on trained neural networks. But actual secrets are not generated by back propagation neural networks shown in table 6.

D. Theorem 1: If there are n number of shares out of n share mobile agent can choose threshold value ‘t’ there is no relationship between ‘t’ and ‘n’. So, technique is threshold-free anonymous secret key sharing.

Proof: There is no mathematical association between ‘n’ and t’. Threshold decided by the mobile host at the time of training of back propagation neural network. There is no requirement of identity of mobile agents at the time of reconstruction of secret keys. So, this approach is an anonymous distribution of secret key.

V. Analysis of proposed Scheme

Proposed scheme based is on the Mignotte's sequence and back propagation neural network has been implemented in python. Table 6 shown below, compares the proposed scheme with the previous scheme is efficient. On the basis of comparison in the initialization phase it takes very less computation time to generate the secret using Mignotte's Sequence. After training a neural with a fixed user defined threshold value “t’ save the program and sent to the different number of platforms at which mobile agents want to execute tasks. At the executing platform choose an authenticated set of mobile agents share to generate the session key. In the generation of secret key artificial neural networks take only O(1) time that is better computational as compared to traditional mechanisms shown in Table 7.

During the training of artificial neural networks there is no limitation on the threshold value: choose nay value between ‘2’ and ‘n’. This is another advantage of the proposed scheme because there is no relationship between ‘t’ and ‘n’ as used in Shamir secret share and Chinese remainder theorem based secret share scheme. Figure 6 shows the graph of training of artificial neural networks with 1000 training iterations with deduction of error.

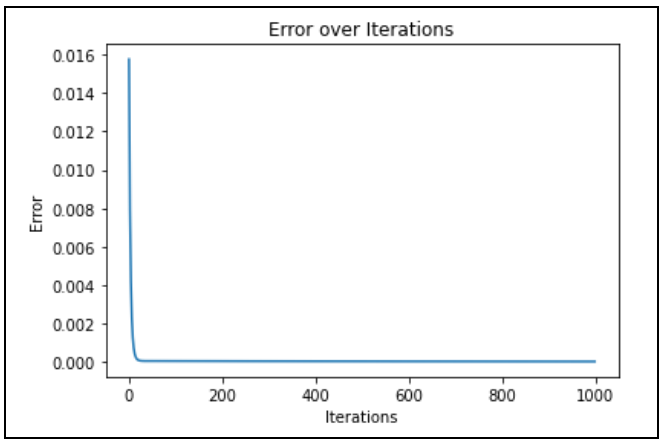

Figure 6. Reducing error with iteration

Technique	Share Generation complexity	Recreation time complexity	Threshold value 't'
Combinatorial Technique	Big oh(n)	Big oh(n)	t=1 or t=1
Strongly ideal secret sharing schemes	Big oh(n)	Big oh(n)	t=2 or t=1
Anonymous secret sharing schemes	Big oh(n)	Big oh(n)	t=2 or t=1
On the bound for anonymous secret sharing schemes	Big oh(n)	Big oh(n)	t=2 or t=1and other cases
Providing anonymity in unconditionally secure secret sharing scheme	Big oh(n)	Big oh(n)	t=2 or t=1and other cases
Proposed scheme	Big oh(n)	O(1)	No limitation

Table 7. Comparison of proposed Scheme with another Scheme

VI. Conclusion and Future Scope

Security of agents during the relocation in hostile environments is a significant issue. In this scheme, design of a framework by fusion of Mignotte's Sequence and back Propagation Neural network to provide the authentication. Dynamic threshold values and Anonymous behavior of the proposed approach based on the backpropagation neural network is proposed generate higher safety of mobile agents during the secret sharing and reconstruction of key. The proposed approach for mobile agent migration increases the security of the execution key. It focuses on improving secure mobile agent migration in an open environment. There is no limitation on the threshold parameter. In future we will try to design fast Backpropagation (BP) Artificial Neural Network to increase the efficiency proposed algorithm and design some new anonymous secret scheme

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. S. K. Narad, "Group Authentication Using Back-propagation Neural Network", vol. 6, no. 10, pp. 272–278, 2017.

[6] 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.

[7] 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.

[8] Y. Q. Deng and G. Song, "A verifiable visual cryptography scheme using neural networks",*Adv. Mater. Res.*, vol. 756–759, pp. 1361–1365, 2013.

[9] 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, 2002.

[10] M. Guillermo, K. M. Martin, and C. M. O'Keefe, "Providing anonymity in unconditionally secure secret sharing schemes",*Des. Codes, Cryptogr.*, vol. 28, no. 3, pp. 227–245, 2003.

[11] Y. P. Deng, L. F. Guo, and M. L. Liu, "Constructions for anonymous secret sharing schemes using combinatorial designs",*Acta Math. Appl. Sin.*, vol. 23, no. 1, pp. 67–78, 2007.

[12] K. V. Priyanka, M. Gowthami, O. Susmitha, G. Prathyusha, and N. B. Muppalaneni, "Breaking Mignotte's sequence based secret sharing scheme using smt solver",*arXiv*, vol. 9, no. 6, pp. 35–42, 2018.

[13] R. Kumar and M. Dhiman, "Secured Image Transmission Using a Novel Neural Network Approach and Secret Image Sharing Technique",*Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 8, no. 1, pp. 161–192, 2015.

[14] 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.

[15] 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.

[16] Niraj Singhal, Ashutosh Dixit, R.P. Agarwal and A.K. Sharma, "A Study of Mobile Agent Platforms for

Distributed Web Crawling", International Journal of Advances in Engineering Science and Technology (ISSN: 2319-1120), Vol. 1, No. 2, pp. 111-121, December 2012.

[17] Pradeep Kumar, Niraj Singhal and Chaitra K.M., "Securing Mobile Agents Migration using Tree Parity Machine with New Tiny Encryption Algorithm", Proceedings of 4th International Conference on Advances in Computing and Data Sciences (ICACDS 2020), CCIS 1244, Springer, Faculty of ICT, University of Malta, Valletta, Malta, pp. 1–10, 2020, (https://doi.org/10.1007/978-981-15-6634-9_13) April 24-25, 2020.

[18] W. Jansen and T. Karygiannis, "NIST Special Publication 800-19 – Mobile Agent Security Computer," *Nist Spec. Publ.*, vol. 323, no. September, pp. 3–10, 1999, [Online]. Available:https://pdfs.semanticscholar.org/52af/fbe3ed6fbf23dd78775ed907b0dc9e5f3fd4.pdf.

[19] K. Meng, F. Miao, W. Huang, and Y. Xiong, "Threshold changeable secret sharing with secure secret reconstruction," *Inf. Process. Lett.*, vol. 157, p. 105928, 2020.

[20] 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.

Author Biographies

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

Second Author 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.

Third Author Sukhendra Singh is pursuing his PhD degree in Data Security from Dr. APJ Abdul Kalam Technical University, Lucknow, India, and completed his M.Tech. in Computer Science from Birla Institute of Technology Mesra, Rachi and B. Tech in Information Technology from Indian Institute of Technology Allahabad, India. He is currently an Assistant Professor in Department of Information Technology of JSS Academy of Technical Education Noida, India. His areas of research include data security, data analysis.