International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

An Efficient Authentication Approach with Optimization Algorithms and Elliptical Curve Cryptography for Cloud Environment

K. Shankar[1], M. Subhashini[2]
[1]Research Scholar in Department of Computer Science, Srimad Andavan Arts and Science College (Autonomous) (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India.
[2]Assistant Professor in Department of Computer Science, Srimad Andavan Arts and Science College (Autonomous) (Affiliated to Bharathidasan University), Tiruchirappalli, Tamil Nadu, India.

Abstract:

The fast-emerging of cloud computing technology today has sufficiently benefited its wide range of users from individuals to large organizations. It carries an attractive characteristic by renting myriad virtual storages, computing resources and platform for users to manipulate their data or utilize the processing resources conveniently over Internet without the need to know the exact underlying infrastructure which is resided remotely at cloud servers. Security is very important for any kind of networks. As a main communication mode, the security mechanism for multicast is not only the measure to ensure secured communications, but also the precondition for other security services. Attacks are one of the biggest concerns for security professionals. Attackers usually gain access to a large number of computers by exploiting their vulnerabilities to set up attack armies. This paper presents a dual optimizer based key generation method for the improving the authentication with Elliptical Curve Cryptography (ECC) encryption algorithm. The optimal private and secret key for the encryption and decryption are obtained with the optimization techniques like Animal Migration Optimization (AMO), and Brain Storm Optimization (BSO) for strengthening the security in the Cloud Computing environment.

Keywords: Cloud Computing, Key Generation, Elliptical Curve Cryptography, Authentication.

1. INTRODUCTION

With the advancement of IT, cloud computing has evolved through a number of different services, such as, virtualization, software as a service (*SAAS*), infrastructure as a service (*IAAS*) and platform as a service (*PAAS*). For instance, cloud computing refers to both the applications delivered as services over the internet and the hardware and system software in the data centres, that provide those services [1] [2]. The basic goal of cloud computing is to provide great flexibility to users, where users can process, store and access their data on the cloud server anytime, anywhere using the internet. In addition, users do not need to concern with the processing details. Generally, cloud systems are divided into three categories, namely, public cloud, private cloud and hybrid cloud [3][4][5].

The new cloud computing technology offers many advantages such as, information shared in virtual environment, dynamic scalability, storage utility, software utilization, platform and infrastructure utilization, managed distributed computing power, and many more. However, the cloud computing technology comes with many issues, for example; performance, resiliency, interoperability, data migration and transition from legacy systems. One of the main issues is security, such as, virtualization security, distributed computing, application security,

identity management, access control and authentication [6][7]. The authors have pointed out that the identity and access control management is a core requirement for cloud computing. Thus, strong authentication becomes paramount requirement in cloud computing service environment [16] [17] [18] [19] [20].

2. AUTHENTICATION TECHNIQUES IN CLOUD

Authentication is a major criteria of each secure communication system especially in wide networks such as Cloud Computing. It guides to protect shared data from unauthorized access and it is a major technique of information security. AAA is a security organizing module for authentication, authorization and accounting. When a user tries to access cloud resources from CSP, then AAA verifies the user's authentication information. If the user is authenticated, then AAA gets the user's access level, which has been most recently pro, by inspecting the user's information in the database. In addition, authentication technique says that “Who is the authorized customer” and “Is the customer really who he claims himself to be”. In addition, verification of customer’s identity is the most important aim behind an authentication. In other words, an authentication mechanism tells how customers identified and verified to access to sensitive data [8] [21] [22] [23]. Verification means confirm that demand is from the legal user. Identification implies on determining users. There are several authentication schemes which categorized in three types as follow:

- Something user know (knowledge factors) such as username and password, PIN based authentication scheme and Implicit Password Authentication System (IPAS).
- Something user has (possession factor) such as smart cards or electronic tokens and identify card such as Automatic Teller Machine card (ATM card).
- Something user is (ownership factor) such as biometric authentication.

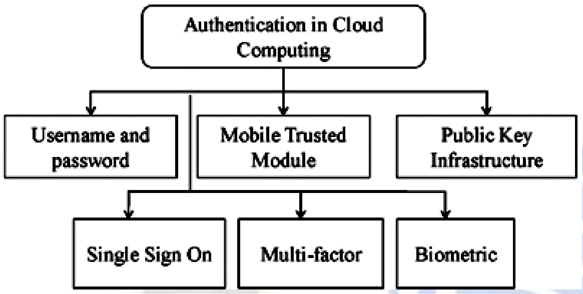


Figure 1: Types of Authentications in Cloud Computing

3. ELLIPTICAL CURVE CRYPTOGRAPHY BASED ENCRYPTION TECHNIQUE

In the Asymmetric Key Cryptography, ECC algorithm is one kind of mechanism for implementing public key cryptography [9] [10]. In this technique, based on the cure, with specific base point and with the help of prime number function as the maximum limit an equation is evaluated, and then the encryption follows: The equation of elliptic curve cryptography is given as,

$$y^3 = x^2 + ax + b$$

Where a, b are the integers.

In any cryptographic process, the strength of the encryption depends on the key generated. In the proposed method, there exist two types of key generation. The first process is to generate public key from the receivers end to encrypt the message and second process is to generate a private key on the receiving end to decrypt the original text.

The value "P" is any base point on the curve. Select a random number 'H' within the range of (1 to n − 1):

$$Q = H \times P$$

Here, 'Q' is the public key and 'H' is the private key.

4. OPTIMIZATION ALGORITHMS FOR OPTIMAL KEY GENERATION

4.1 Animal Migration Optimization (AMO)

The AMO calculation is a swarm-based calculation roused by the migration wonder of animals [11][12]. In the calculation, people are viewed as places of animals, and positions can be moved by for the most part two activities: creature migration and populace refreshing. The activity of creature migration reproduces practices of creature bunches moving from the present territory to another region. New places of people will be delivered by the bearing of creature migration, where three migration rules are considered: Individuals move towards a similar course as their neighbouring people; people stay close to their neighbouring people; and people keep away from crashes with their neighbouring people. Utilizing the three migration runs, a probability approach is acquainted with yield new places of people. The calculation starts with a haphazardly instatement populace, which is contained NP include vectors with D_x Dimensions, which can be expressed as pursues:

$$y_{j,k,0} = y_{k,min} + rand_{j,k}.(y_{k,max} - y_{k,min}) \quad (5)$$

Where $y_{k,max}$ and $y_{k,min}$ are the maximum value and minimum value of the k-ith dimension. $y_{j,k,0}$ is the k-th dimension value of the j-th individual in the initialization population, and $rand_{j,k}$ is a uniformly random number between 0 and 1, j=1,…,NP and k=1,…, D_x.

In the wake of delivering the introduction populace, animal migration and populace refreshing tasks are performed iteratively. During the animal migration, the people should move another situation as indicated by the places of their neighboring people, which can be depicted as pursues:

$$y_{j,k,G+1} = y_{j,k.G} + \delta.(y_{neighborhood,k,G} - y_{j,k,G}) \quad (6)$$

Where $y_{j,k,G}$ is the k-th dimension value of the j-th individual in the current population G, and $y_{j,k,G+1}$ is the k-th dimension value of the j-th individual of the new population G+1; $y_{neighborhood,k,G}$ is the k-th dimension value of the neighbouring individual of $y_{j,k,G}$ which is defined using a ring topology scheme. For the k-th dimension, $y_{neighborhood,k,G}$ is selected from the (j-2)-th individual, the (j-1)-th individual, the j-th individual, the (j+1)-th individual, and the (j+2)-th individual, δ is a random number produced by a Gaussian Distribution with N (0,1).

The population updating simulates how animals leave the group and new individuals join in the new population, as the equation describes:

$$y_{j,k,G+1} = y_{r1,k.G} + rand_1.\left(y_{best,k,G} - y_{j,k,G}\right) + rand_2.\left(y_{r2,k,G} - y_{j,k,G}\right) \quad (7)$$

Where $y_{r1,k.G}$ is the k-ith dimension value of the individual to be updated, which is chosen randomly in the current population; moreover, different from $y_{j,k,G}, y_{r2,k,G}$ is the k-ith dimension value of another random individual, $y_{best,k,G}$ is the k-ith dimension value of the best individual that has been found. $rand_1$ and $rand_2$ are two uniformly random numbers between 0 and 1. The algorithm makes the assumption that the number of animals in the population remains unchanged. Therefore, in the updating, it replaces some of the animals with new individual according to a probability p_{a_j}, which is related to the fitness of individuals and can be calculated as follows:

$$p_{a_j} = \frac{sn_j}{NP} \qquad (8)$$

Where p_{a_j} is the probability value of the j-th individual, NP is the number of the individuals in the population, and sn_j is the sequence number of the fitness of j-th individual after being sorted by their fitness in descending order, where i = 1, 2, . . . , NP.

For every person and each measurement, a consistently arbitrary number, meant by rand, somewhere in the range of 0 and 1 will be created as the probability to decide if the individual is held or is supplanted by another person. In this way, people with better wellness will be saved with higher probability in the people to come, while those with more regrettable wellness will likely be supplanted by new people. Besides, the animal with best position will be held in the people to come.

4.2 Brain Storm Optimization (BSO)

The brainstorming process has been widely used for problems that cannot be solved by one individual. To overcome these problems, individuals from different background are gathered to brainstorm. The main aim of brainstorming is to generate as many different ideas (solutions) as possible, and good solutions can be obtained to solve a specific problem from the generated ideas [13] [14].

BSO [15] is a new population-based swarm intelligence method inspired by the human brainstorming process. BSO generates n random possible solutions, and evaluates them based on a fitness function. BSO contains three steps, which are clustering individuals, disrupting the cluster centers, and creating solutions. BSO clusters n individuals into m clusters using the k-means clustering algorithm. Similar solutions are clustered together during each generation. A new solution is generated with a probability of P, and it replaces a cluster center (selected randomly), through a disrupting cluster center operation. Finally, BSO generates a new individual using one cluster, or by combining two clusters. To generate a new

individual, BSO randomly selects one or two cluster(s), with a probability of P-one. Then, BSO randomly selects one individual based on one or two cluster(s) center(s), as follows:

$$X_{Selected} = \begin{cases} X_i & One\ cluster \\ rand \times X_{1i} + (1 - rand) \times X_{2i} & two\ cluter \end{cases}$$

Where X_{1i} and X_{2i} are the ith dimension of the selected clusters, and rand is a random value between 0 and 1. BSO updates the selected individual as follows:

$$X_{new} = X_{Selected} + \xi \ * random(0,1)$$

Where random is a Gaussian random value with 0 mean and unit variance, respectively;

ξ is the adjusting factor, i.e.,

$$\xi = logsin\left(\frac{0.5 * m_i - c_i}{k}\right) \times rand$$

Where m_i and c_i are the maximum number of iterations and the current iteration, respectively. $logsin()$ indicates the logarithmic sigmod function, rand() is a random value between 0 and 1, and k is a changing rate for the slope of logsin() function.

5. PROPOSED DUAL OPTIMIZER KEY ALGORITHM FOR ECC - KEY GENERATION

BSO is a high-performing approach in many problems; however, its drawback is the slow convergence speed. At exploration, the algorithm is very good, however, its performance is poor at exploitation. In this study, we propose a hybridized BSO method to speed up the algorithm’s convergence and achieve better results, as well as to make a better balance between exploration and exploitation. Algorithm hybridization blends the benefits of various algorithms, which results in the synergistic synthesis of fused algorithms. In some runs of the algorithm, when BSO finds the promising region of the search space early enough, it converges successfully as it has enough time for exploitation.

However, in other runs, when it hits the promising region of the search space later, it does not have enough iterations for the exploitation (i.e., already the weakest spot of the algorithm), and it is not able to converge. For enhancing the exploitation capability, we incorporated the search mechanism of the AMO, which is famous for its intensification capability. Various studies have shown that the AMO exploitation procedure is efficient and can be used in solving a wide variety of optimization problems. By hybridizing the BSO with AMO search, the novel algorithm aims to take the best characteristics from both BSO (excellent exploration) and AMO (superior exploitation), and to minimize the known deficiencies. The AMO search mechanism is mathematically formulated as follows:

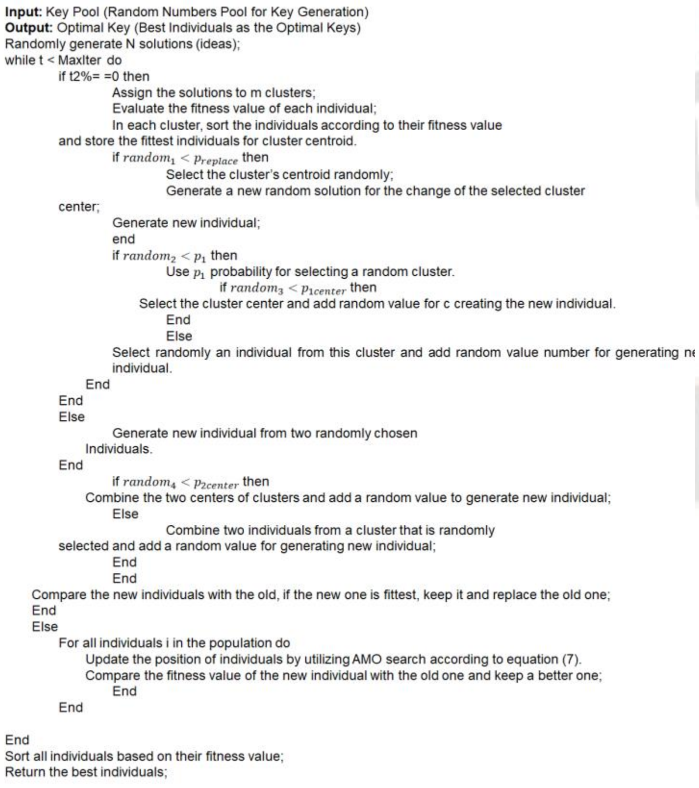
$$y_{j,k,G+1} = y_{r1,k.G} + rand_1.\left(y_{best,k,G} - y_{j,k,G}\right) + rand_2.\left(y_{r2,k,G} - y_{j,k,G}\right)$$

Where the current solution is denoted by $y_{r1,k.G}$, and the newly updated position is denoted by $y_{j,k,G+1}$, $rand_1$ and $rand_2$ are the modifiable parameters, which can be adjusted to achieve better performance on a specific problem.

In the proposed hybrid approach, if the iteration counter t is even, the original BSO updating mechanism is utilized for solution modification, otherwise, if the iteration counter value is odd, the AMO search mechanism is utilized for a position update.

Considering the problem for which the proposed method is applied, we adopted it for the binary selection problem. First, the sigmoid function is used to move the solutions' values between 0 and 1, then the threshold value, which is set to 0.5 in this experiment, will be decided whether to be 0 or 1. If the value is less than the threshold values, its value will 0, otherwise, it will be 1. The binary conversion of the solution value is described in Equation (4).

Algorithm 1: Dual Optimizer Key Generation (DOKG) Algorithm for ECC



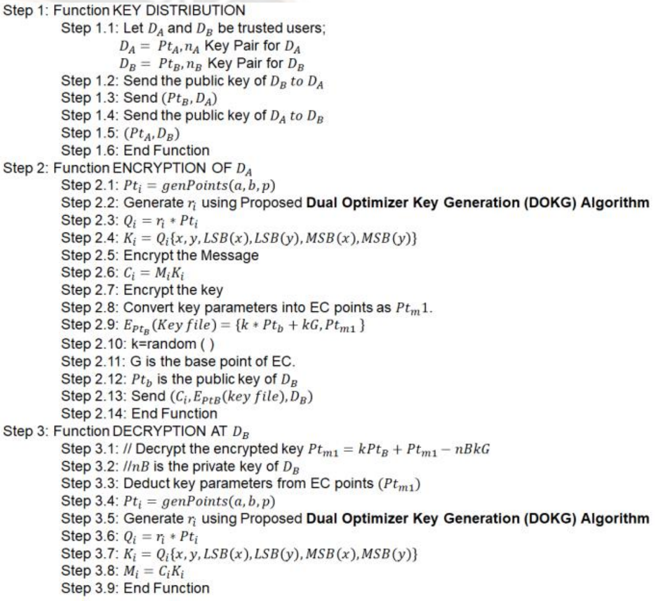
```
Input: Key Pool (Random Numbers Pool for Key Generation)
Output: Optimal Key (Best Individuals as the Optimal Keys)
Randomly generate N solutions (ideas);
while t < MaxIter do
        if t2%= =0 then
                Assign the solutions to m clusters;
                Evaluate the fitness value of each individual;
                In each cluster, sort the individuals according to their fitness value
        and store the fittest individuals for cluster centroid.
                if $random_1 < p_{replace}$ then
                        Select the cluster's centroid randomly;
                        Generate a new random solution for the change of the selected cluster
        center;
                Generate new individual;
                end
                if $random_2 < p_1$ then
                        Use $p_1$ probability for selecting a random cluster.
                                if $random_3 < p_{1center}$ then
                    Select the cluster center and add random value for c creating the new individual.
                        End
                        Else
                Select randomly an individual from this cluster and add random value number for generating ne
                individual.
            End
        End
        Else
                Generate new individual from two randomly chosen
            Individuals.
        End
                if $random_4 < p_{2center}$ then
            Combine the two centers of clusters and add a random value to generate new individual;
                Else
                        Combine two individuals from a cluster that is randomly
        selected and add a random value for generating new individual;
                End
                End
    Compare the new individuals with the old, if the new one is fittest, keep it and replace the old one;
    End
    Else
        For all individuals i in the population do
            Update the position of individuals by utilizing AMO search according to equation (7).
            Compare the fitness value of the new individual with the old one and keep a better one;
                End
        End

End
Sort all individuals based on their fitness value;
Return the best individuals;
```

$$y_j = \begin{cases} 1 & if\ S(y_j) > 0.5 \\ 0 & Otherwise \end{cases}$$

The jth solution is denoted by y_j, and $S()$ represents the Sigmoid Transfer Function.

In this way, the solution representation in the feature selection problem is encoded by binary values. The total number of solutions is N and the dimension of the solutions corresponds to the number of features in a given data set. If the solution value is 0, it means that the corresponding feature is removed, in contrast, if its value is 1, the feature is selected for the classification model.

Algorithm 2: Elliptical Curve Cryptography (ECC) based on Proposed DOKG



```
Step 1: Function KEY DISTRIBUTION
        Step 1.1: Let $D_A$ and $D_B$ be trusted users;
                $D_A = Pt_A, n_A$ Key Pair for $D_A$
                $D_B = Pt_B, n_B$ Key Pair for $D_B$
        Step 1.2: Send the public key of $D_B\ to\ D_A$
        Step 1.3: Send $(Pt_B, D_A)$
        Step 1.4: Send the public key of $D_A\ to\ D_B$
        Step 1.5: $(Pt_A, D_B)$
        Step 1.6: End Function
Step 2: Function ENCRYPTION OF $D_A$
        Step 2.1: $Pt_i = genPoints(a, b, p)$
        Step 2.2: Generate $r_i$ using Proposed **Dual Optimizer Key Generation (DOKG) Algorithm**
        Step 2.3: $Q_i = r_i * Pt_i$
        Step 2.4: $K_i = Q_i\{x, y, LSB(x), LSB(y), MSB(x), MSB(y)\}$
        Step 2.5: Encrypt the Message
        Step 2.6: $C_i = M_i K_i$
        Step 2.7: Encrypt the key
        Step 2.8: Convert key parameters into EC points as $Pt_m 1$.
        Step 2.9: $E_{Pt_B}(Key\ file) = \{k * Pt_b + kG, Pt_{m1}\}$
        Step 2.10: k=random ( )
        Step 2.11: G is the base point of EC.
        Step 2.12: $Pt_b$ is the public key of $D_B$
        Step 2.13: Send $(C_i, E_{PtB}(key\ file), D_B)$
        Step 2.14: End Function
Step 3: Function DECRYPTION AT $D_B$
        Step 3.1: // Decrypt the encrypted key $Pt_{m1} = kPt_B + Pt_{m1} - nBkG$
        Step 3.2: //$nB$ is the private key of $D_B$
        Step 3.3: Deduct key parameters from EC points $(Pt_{m1})$
        Step 3.4: $Pt_i = genPoints(a, b, p)$
        Step 3.5: Generate $r_i$ using Proposed **Dual Optimizer Key Generation (DOKG) Algorithm**
        Step 3.6: $Q_i = r_i * Pt_i$
        Step 3.7: $K_i = Q_i\{x, y, LSB(x), LSB(y), MSB(x), MSB(y)\}$
        Step 3.8: $M_i = C_i K_i$
        Step 3.9: End Function
```

6. RESULT AND DISCUSSION

The performance metrics like Key Generation time (in milliseconds), key updation time (in milliseconds), Encryption time (in milliseconds), Decryption time (in milliseconds), Total time is taken for encryption and decryption time (in ms) and throughput (in mbps) for the encryption and decryption. The performance of the proposed Dual Optimizer Key Generation based ECC Authentication Method is analyzed with the existing encryption techniques like ECC, RSA and ElGamal.

Table 1 depicts the key generation time (in milliseconds) by proposed DOKG-ECC, ECC, ElGamal and RSA encryption techniques against the varying file size (in MB). From the table 1, it is clear that the proposed DOKG-ECC requires less key generation time than ECC, ElGamal and RSA algorithms for varying file sizes.

Table 1: Key Generation (in Milliseconds) by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Key Generation Time (in milliseconds)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	629	757	859	951
300	764	841	1026	1154
450	886	1176	1282	1393
600	948	1357	1417	1532
750	1017	1542	1675	1713
900	1279	1645	1736	1874
1050	1537	1853	2057	2145

Table 2 depicts the key updation time (in milliseconds) by proposed DOKG-ECC, ECC, RSA and ElGamal encryption techniques against the varying file size (in MB). From the table 2, it is clear that the proposed DOKG-ECC requires less time for updating the key than ECC, RSA and ElGamal algorithms for varying file sizes.

Table 2: Key Updation Time (in Milliseconds) by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Key Updation Time (in Milliseconds)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	651	729	868	973
300	864	982	1185	1292
450	1157	1245	1347	1432
600	1309	1476	1512	1633
750	1469	1526	1692	1781
900	1573	1631	1762	1823
1050	1644	1887	1921	2017

Table 3 depicts the encryption time (in seconds) by proposed DOKG-ECC, ECC, RSA and ElGamal encryption techniques against the varying file size (in MB). From the table 3, it is clear that the proposed DOKG-ECC requires less time for encryption than ECC, RSA and ElGamal algorithms for varying file sizes.

Table 3: Encryption Time (in Seconds) by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Encryption Time (in Seconds)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	102	153	193	275
300	183	216	375	463
450	324	481	551	618
600	492	514	688	727
750	654	774	813	937
900	722	883	987	1034
1050	851	786	1052	1142

Table 4 depicts the decryption time (in seconds) by proposed DOKG-ECC, ECC, RSA and ElGamal encryption techniques against the varying file size (in MB). From the table 4, it is clear that the proposed DOKG-ECC requires less time for decryption than ECC, RSA and ElGamal algorithms for varying file sizes.

Table 4: Decryption Time (in Seconds) by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Decryption Time (in Seconds)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	114	217	382	445
300	256	384	478	546
450	353	463	583	692
600	485	582	681	792
750	593	712	892	926
900	634	835	971	1073
1050	813	945	1057	1191

Table 5 gives the total time taken (in seconds) for encryption and decryption by proposed DOKG-ECC, ECC, RSA and ElGamal encryption techniques against the varying file size (in MB). From the table 5, it is clear that the proposed DOKG-ECC consumes less time for encryption and decryption than ECC, RSA and ElGamal algorithms for varying file sizes.

Table 5: Total Time Taken for Encryption and Decryption (in Seconds) by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Total Time taken for Encryption and decryption (in Seconds)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	216	370	575	720
300	439	600	853	1,009
450	677	944	1,134	1,310
600	977	1,096	1,369	1,519
750	1,247	1,486	1,705	1,863
900	1,356	1,718	1,958	2,107
1050	1,664	1,731	2,109	2,333

Table 6 gives the Throughput (in mbps) for encryption and decryption by proposed DOKG-ECC, ECC, RSA and ElGamal encryption techniques against the varying file size (in MB). From the table 6, it is clear that the proposed DOKG-ECC increased throughput ECC, RSA and ElGamal algorithms for varying file sizes.

Table 6: Throughput (in mbps) obtained by the Proposed DOKG-ECC, ECC, RSA and ElGamal

File Size (in MB)	Throughput (in mbps)			
	Proposed DOKG-ECC	ECC	RSA	ElGamal
150	1382	1017	946	891
300	1428	1292	1082	1016
450	1671	1358	1236	1164
600	1782	1494	1369	1291
750	1889	1522	1414	1383
900	2014	1689	1561	1411
1050	2158	1731	1624	1518

7. CONCLUSION

Authentication is one of the main important challenges in security of cloud computing. Single level authentication has many problems mainly with sensitive data, as passwords are easy to break. In this paper, an enhanced encryption technique has proposed to ensure security among the data communication between the cloud servers. This proposed technique is the combination of Dual Optimizer like Animal Migration Optimization (AMO) and Brain Storm Optimization (BSO) and ECC algorithm. The random number for generating the secret keys by the ECC is calculated with the DF algorithm. The proposed DOKG-ECC technique performed better than the other existing techniques like RSA, ECC, and ElGamal. The performance of the proposed DOKG-ECC is evaluated with a key generation time, key updation time, time is taken for encryption and decryption and its total time is taken and throughput. From the results obtained, it is evident that the proposed DOKG-ECC method performs better than existing techniques.

REFERENCES

[1] Bohn, Robert B., Craig A. Lee, and Martial Michel. "The NIST cloud federation reference architecture." (2020).

[2] Tabrizchi, Hamed, and Marjan Kuchaki Rafsanjani. "A survey on security challenges in cloud computing: issues, threats, and solutions." The journal of supercomputing 76.12 (2020): 9493-9532.

[3] Singh, Jaswinder, and Gaurav Dhiman. "A survey on cloud computing approaches." Materials Today: Proceedings (2021).

[4] Sunyaev, Ali. "Cloud computing." Internet computing. Springer, Cham, 2020. 195-236.

[5] Wulf, Frederik, et al. "IaaS, PaaS, or SaaS? The Why of Cloud Computing Delivery Model Selection." (2021).

[6] Anandhi, S., R. Anitha, and Venkatasamy Suresh kumar. "An authentication protocol to track an object with multiple RFID tags using cloud computing environment." Wireless Personal Communications 113.4 (2020): 2339-2361.

[7] Veerabathiran, Vijaya Kumar, et al. "Improving secured ID-based authentication for cloud computing through novel hybrid fuzzy-based homomorphic proxy re-encryption." Soft Computing 24.24 (2020): 18893-18908.

[8] Safkhani, Masoumeh, et al. "RSEAP2: An enhanced version of RSEAP, an RFID based authentication protocol for vehicular cloud computing." Vehicular Communications 28 (2021): 100311.

[9] Khan, Mohammad Ayoub, et al. "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data." IEEE Access 8 (2020): 52018-52027.

[10] Duraki, Sadat, Sercan Demirci, and Selçuk Aslan. "UAV placement with animal migration optimization algorithm." *2020 28th Telecommunications Forum (TELFOR)*. IEEE, 2020.

[11] Li, Shuxiang, and Xianbing Pan. "Adaptive management and multi-objective optimization of virtual machine in cloud computing based on particle swarm optimization." *EURASIP Journal on Wireless Communications and Networking* 2020.1 (2020): 1-12.

[12] Thanga Revathi, S., N. Ramaraj, and S. Chithra. "Brain storm-based Whale Optimization Algorithm for privacy-protected data publishing in cloud computing." *Cluster Computing* 22.2 (2019): 3521-3530.

[13] Palanikkumar, D., and S. Priya. "Brain storm optimization graph theory (BSOGT) and energy resource aware virtual network mapping (ERVNM) for medical image system in cloud." *Journal of medical systems* 43.2 (2019): 1-10.

[14] Dai, Cai, and Xiujuan Lei. "A multiobjective brain storm optimization algorithm based on decomposition." *Complexity* 2019 (2019).

[15] Subhashini, M., & Gopinath, R., Mapreduce Methodology for Elliptical Curve Discrete Logarithmic Problems – Securing Telecom Networks, International Journal of Electrical Engineering and Technology, 11(9), 261-273 (2020).

[16] Upendran, V., & Gopinath, R., Feature Selection based on Multicriteria Decision Making for Intrusion Detection System, International Journal of Electrical Engineering and Technology, 11(5), 217-226 (2020).

[17] Upendran, V., & Gopinath, R., Optimization based Classification Technique for Intrusion Detection System, International Journal of Advanced Research in Engineering and Technology, 11(9), 1255-1262 (2020).

[18] Subhashini, M., & Gopinath, R., Employee Attrition Prediction in Industry using Machine Learning Techniques, International Journal of Advanced Research in Engineering and Technology, 11(12), 3329-3341 (2020).

[19] Poornappriya, T.S., Selvi, V., Evolutionary Optimization of Artificial Neural Network for Diagnosing Autism Spectrum Disorder, International Journal of Electrical Engineering and Technology (IJEET), 11(7), 47-61 (2020).

[20] Rethinavalli, S., & Gopinath, R., Botnet Attack Detection in Internet of Things using Optimization Techniques, International Journal of Electrical Engineering and Technology, 11(10), 412-420 (2020).

[21] Priyadharshini, D., Poornappriya, T.S., & Gopinath, R., A fuzzy MCDM approach for measuring the business impact of

employee selection, International Journal of Management (IJM), 11(7), 1769-1775 (2020).

[22] Poornappriya, T.S., Gopinath, R., Application of Machine Learning Techniques for Improving Learning Disabilities, International Journal of Electrical Engineering and Technology (IJEET), 11(10), 392-402 (2020).