International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Enhanced Ant Colony Optimization Based Secured Routing in Cloud Computing

Vatchala B[1], G. Preethi[2]

[1]Research Scholar, Department of Computer Science, PRIST Deemed University, Thanjaur
vatchala2020[at]gmail.com

[2]Associate Professor, Department of Computer Science, PRIST Deemed University, Thanjaur
mgpreethi[at]gmail.com

Abstract: ***Cloud Computing incorporates both the computing process paradigm and the environment for many users to share services and resources. This ecosystem has the advantage of worldwide availability through the Internet and can be shared at various levels. The level of operation, infrastructure, platform level applications and resources can be exchanged with many clients. Cloud Computing allows remote server access through the internet. Cloud Computing leverages much of the existing technology for the development of cloud environments, such as online servers, web browsers and virtualization. Any vulnerability associated with these systems thus also impacts the cloud and may have an important impact. Most data vulnerabilities happen during their travel to destination. Hence, there exists a need to provide security to the path where the data travels. Enhanced Ant Colony Optimization based Secured Routing (EACOSR) has been proposed in this paper to identify the best route in cloud computing to destination and provide security to the data travels in it. Blowfish algorithm has been modified to meet standard of cloud computing and it is applied in EACOSR to give elevated security to the data. EACOSR is developed with auto-update mechanism to optimize the tasks in cloud computing that results in reduced energy consumption and minimum packet delay. Cloudsim simulator is utilized to analyze the performance of EACOSR against the previous methods. Results with better values make an indication that EACOSR has better performance than the considered previous methods.***

Keywords: cloud, delay, energy, optimization, routing, security

1. Introduction

Cloud Computing (CC) is a specific type of distributed computing that delivers various types of significant services to the users via internet. Significant services include databases, storage, networking, software, hardware and security. CC acts as an alternate option for datacenter on-premises. In general computer networks, the users are in the demand to setup or perform the things (i.e., network activities) like (i) virtualization [1] (ii) network, firewall configuring and (iii) installation of updates in hardware, software and operating system. Once after setting up all the above mentioned things, users become core responsible for the maintenance of overall network lifecycle. But, while users choosing CC, vendor of cloud becomes the core responsible for all mentioned network activities. CC offers a wide cum rich variety of software and hardware services to the users where it is utilized in pay and use manner. CC holds merits of (i) on-demand access (ii) easy scalability and manageability (iii) versatility (iv) reliability and (v) low economy. Three significant service models available in CC are (i) Software as a Service [2] (ii) Infrastructure as a Service [3] and (iii) Platform as a service [4]. Developers of CC applications makes use of these service models to write code, compile it and host the applications in different clouds for users effective usage at anytime and anywhere. The four different types of CC deployment models are Community Cloud [5], Private Cloud [6], Hybrid Cloud [7] and Public Cloud [8]. Cloud assisted ad-hoc networks are getting popular and attracting many researchers.

CC is facing the same level of demerits as it faces merits. Issues in CC are linked with multi-tenancy, interoperability and scalability. Significant issues are faced in terms of security and it is accepted understood by worldwide CC users [9], [10]. The services provided via CC face different kinds of security attacks internally and externally. Security in CC integrates different strategies that focus on re-establishment, safeguarding and providing assurance to the secure transmission of data. Different cloud vendors are attempting to develop cloud-based applications that satisfy secure engineering for remediation, controlling and deceivability [11], [12]. In the world of CC, the genuineness of data outsourced and implemented software provides the authorization to cloud service providers (i.e., third party) directly. Hence, the level of trust is entirely dependent on the model implemented in CC and cloud vendors. Multiple researches have been carried out towards providing safe-link (i.e., secured link) between cloud users and cloud service providers where it focuses on developing protocols to ensure data communication [13]. Cryptography is the only strategy that is followed to provide security that converts the contents of readable, viewable, hearable data to unknown formats using the pubic key or private key. Currently, cryptography as a acting as a security tool available only for the intended (i.e., specific) users [10], [14].

Nature of CC leads a way to illicit usage of intra-cloud which enables the third-party to impose security oriented attacks on integrity, privacy and availability of data or resources. Implementation of security degrades the network performance in cloud computing. Making use of security as an option provides a way for information leaking. Encryption becomes mandatory when cloud users think about their security for their data. While implementing security in CC, performance of cloud gets degrade and also more chances are there for network congestion and unbalanced load [15].

2. Literature Review

“Cloud Architecture Reference Model (CARM)” [16] proposed to overcome the vulnerabilities in the cloud computing system. It holds controlling of security and assessing trust level by quantifying confidentiality. Trust level is utilized to measure security level in IaaS cloud which is adopted with alternate controls of cloud security. “Privacy Preservation (PP)” [17] proposed to enable the security in the databases of the external cloud. It attempts to enable security in cloud databases by depending on secured services offered for image denoising. It demands enhanced quality of data and avoidance of redundancy. Further, it attempted to save the originality of the data from facing security attacks in cloud environment. “Watermark-Based Protocol (WBP)” [18] aimed to prevent illegal way of copying and distributing the data. In this, a uniquely generated watermark is made to embed in encrypted content and it is done in cloud server while the user sends a query. If a data copy is detected, then by analyzing unlawful query the copied data can be traced and detected. “Image Encryption Scheme (IES)” [19] proposed to preserve privacy while storing and retrieving the images in the cloud server. It acts as a framework to enables security in the form of encryption during storage and search. Further, it attempts to provide security against the cloud administrators who are honest but curious. “Privacy-Preserving Retrieval Scheme (PPRS)” [20] intended to permit (i) owner of data to outsource the database holding images and (ii) cloud services to access the image by without providing original data to the server in cloud. Features are used to denote an image and verify the similarity between images without providing sensitive data.

“Outsourced Data Recovery Service (ODRS)” [21] proposed to provide a service that recoversthe missed data. Owners of data in cloud computing always expect to provide sample data towards saving the cloud storage. To reconstruct the data, the end-user is necessary to connect to the cloud having sparse data. “Finite-State Chaotic Compressed Sensing (FSCCS)” [22] proposed to sense the remote cloud during data transformation. It attempts to save space and enhance the security during the transmission of data. Data are registered using global and local information that leads to minimizing the similarity in sensed data. “Harness Encryption (HE)” [23] proposed to secure the data that are shared in cloud-assisted mobile devices. It leverages the encrypted data using privacy assurance. Its main intention is to save the cost spent on transmitting the data. Indexing is maintained for increasing the search efficiency of mobile clients. The strength of the security is analyzed by using traditional methods. “Blockchain-Based Secure and Privacy-Preserving (BBSPP)” [24] proposed to protect electronic health records. In this, sender and receiver of data shares desired key to seek the expected electronic health record. Encryption is performed with data-owners authorization. Searching and conditioning way of encryption is applied to preserve privacy and a consensus mechanism is used for seeking system availability. “Color Image Encryption System (CIES)” [25] proposed for securing the resources that are shared in the cloud. Watermarking strategy is adopted for the authentication of images. A chaotic signal of the user-dependent key is applied in the generation of watermarks. Hence, the digital way of carrying out the watermarking and color image encryption is done before the resources are shared in the cloud.

Table 1: Advantage and Disadvantage of Existing methods

Name	*Advantage*	*Disadvantage*
CARM [16]	Minimization of surveillance in the cloud network	More Energy Consumption
PP[17]	Secured search for similarity. Avoidance of redundancy	demand for high-quality image
WBP [18]	Better feature extraction	Consumption of enhanced time
IES [19]	Efficient operation towards minimizing the space complexity.	Low precision in the retrieval of image
PPRS [20]	Preservation of sensitive data	Efficiency of search degrades when the data count increases.
ODRS [21]	Enhanced privacy protection	Consumption of more memory
FSCCS [22]	Improved accuracy of image registration	Complexity in receiving global and local information of an image.
HE [23]	Low energy consumption	High bandwidth for the transmission of the image.
BBSPP [24]	Better availability of server for usage by cloud users	Low-level authorization
CIES [25]	secure way of resource sharing	It is theoretical and not suitable for real-time implementation
LCS [26]	Recovery of missing pixels	Poor performance of rate-distortion
IRHCO [27]	Minimized time consumption for retrieving the images.	Decreased accuracy towards image retrieval
CloudDLP [28]	Auto removal of sensitive data	Affects the performance of the application
CSM [29]	Cost-effective Approach	Failure in adopting the ad-hoc environment
LC [30]	Optimum way of using symmetric algorithms	Gets lack when the size of data gets increased

“Lossy Compression Scheme (LCS)” [26] is proposed for encrypting the image based on the inpainting method. To maintain the confidentiality of the image till it reaches the end-user, LCS encrypts the data via modulo-256 encryption strategy and image contents are blocked by computing the block permutation. In the received end, data reconstruction is performed. “Image Retrieval for Harris Corner Optimization (IRHCO)” [27] is an feature extraction method for massive images that are stored in the cloud. A special model depending on the harris algorithm and robust features are developed to generate new feature vectors based on the individual image. A hashing algorithm is applied to build an index for searching the individual images. A chaotic way is followed to encrypt the images. “CloudDLP” [28] is a transparency cum scalability enabled approach that removes the sensitive data in document. It acts as an internet gateway that uses javascript injecting strategies and applies deep learning algorithms for removing the sensitive data. It attempts to remove the sensitive data in an automatic manner. “Conceptual Security Model (CSM)” [29] to assist the adoption of DevSecOps in business process inter-cloud. CSM focuses on integrating the security service applications to meet the need of users who are using open-source software. Integration is done among (i) development (ii) security and (iii) daily user operation activities. “Lightweight Cryptography (LC)” [30] proposed to increase the security of the cloud network without degrading its performance. The architecture of LC is based on (i) time spent for computing the key used for encryption and decryption (ii) statistics and (iii) entropy.

Advantages and disadvantages of the discussed literature (i.e., existing methods) are provided in Table 1.

3. Enhanced ant Colony Optimization based Secured Routing (EACOSR)

Ant Colony Optimization (ACO) is the most common cum globally accepted probability-based algorithm that searches for an optimum solution. ACO is derived from the route searching mechanism of ants while seeking their food. When an ant seeks its food, a sort of material called “pheromone” gets generated from its body and deposit in the route which it travels. Currently, there exists a clear relationship between the likelihood of direction and the level of pheromone deposited in the path. If ants share the same path, then it would be easier for other ants to follow the same. In this way, ants make communication with other ants to find their food.

To solve the security issues in cloud computing, the selection probability of vertex is expressed as Equation (1).

$$D_{M_z}(r_j, s_g) = \frac{\left(T_{M_z}(r_j, s_g)\right)^a \left(N_{M_z}(r_j, s_g)\right)^b}{\sum_{s_h \epsilon P(r_q)} \left(T_{M_z}(r_j, s_g)\right)^a \left(N_{M_z}(r_j, s_g)\right)^b} \quad (1)$$

$$N_{M_z}(r_j, s_g) = \left(1 + Security(r_j, s_g) \cup M_z\right) - Security(M_z)^{-1} \quad (2)$$

where M_z indicates the task of an ant z; $T_{M_z}(r_j, s_g)$ denotes a value used to specify the pheromone value at the vertex (r_j, s_g); $N_{M_z}(r_j, s_g)$ represents the heuristic data towards vertex (r_j, s_g) selection and it is inverse to the proportion of constraints that are violated during the assignment of s_g to r_j; a and b indicate the parameters used to determine the level of pheromone deposited and heuristic data.

Based on Equation (1) and Equation (2), an individual ant builds a task completely. After building the task, pheromones are updated using Equation (3)

$$\begin{gathered} T_{M_z}(r_j, s_g) \leftarrow (1-\mu) T_{M_z}(r_j, s_g) + \sum_{M_k \epsilon bestfitM} \Delta T_{M_z}(r_j, s_g) \\ if\, T_{M_z}(r_j, s_g) \le T_{low}, then\, T_{M_z}(r_j, s_g) \leftarrow T_{low} \\ if\, T_{M_z}(r_j, s_g) \ge T_{high}, then\, T_{M_z}(r_j, s_g) \leftarrow T_{high} \end{gathered} \quad (3)$$

$$\Delta T_{M_z}(r_j, s_g) = \begin{cases} \left(Security(M_z)\right)^{-1} & if\,(r_j, s_g) \epsilon M_z \\ 0 & else \end{cases} \quad (4)$$

where μ represents the evaporation rate of pheromone; $\Delta T_{M_z}(r_j, s_g)$ indicates the enhanced pheromone maintained by ant z in the currently executing optimum assignment. Suppose, a non-optimum task is identified by an ant means, then a pheromone value is set back to the previous value (i.e., ‘0’). Additionally, evaporation rate of pheromone falls at certain level of the interval. (i.e., between T_{low} and T_{high}). Algorithm 1 shows the baseline steps involved in ACO

Algorithm 1: Ant Colony Optimization

```
Input: Security(R,P,W); count of iterations n; size of population m
Output: $bestM$
Initialization
for each u=1 to n do
        for each v=1 to m do
        build a complete task $M_z$
        if $Security(M_z) < Security(bestM)$ then
                $bestM \leftarrow M_z$
        end if
        end for each
        pheromone updation at every vertex
end for each
return $best\ M$
```

In general ACO, vertex selection probability is made to update only when an ant builds its task. The level of pheromone is updated by utilizing the pheromone evaporation rate. The evaporation of the pheromone minimizes its value at the vertex and this helps to increase the search space. In Parallel, Pheromone presence in the current task is increased. Few tasks that have varying pheromone values are abandoned in the searching process and it results in minimized stability and maximized randomness.

Auto-update mechanism is applied to optimize the task without aborting or abandoning the outstanding value of variables in the task. The updated task detected by ant z is represented as M_z where its level of security is represented as $Security(M_z)$. The auto-update mechanism intends to increase $Security(M_z)$ by updating a variable in the task M_z. If currently updated task $M_z = \{r_1, r_2, r_3, \ldots, r_{n-1}, r_n\}$ where n represents the variable count. While $r_j \epsilon M_z$ is selected to update, the value s_g is assigned to r_j a variable is linearly modified with the values present in $p(r_j)$. The auto-update mechanism follows the same procedure for further variable updates. Algorithm 2 shows the baseline steps involved in the auto-update mechanism.

Algorithm 2 Auto-Update Mechanism

```
Input: Selected Task $M_z$; size of domain $U$
Output: Updated Task $M_z$
foreach chosen variable $r_j$ do
        foreach $g = 1$ to $U$ do
                if $s_g$ not assigned for $r_j$ in $P(r_j)$ then $r_j$ is assigned with $s_g$
```

```
                    $M_z^* \leftarrow M_z(r_j, s_g)$
                if $Security(M_z^*) < Security(M_z)$ then
                    $M_z \leftarrow M_z^*$
                end if
            end if
        end foreach
end foreach
Return $M_z$
```

EACO calls the auto-update mechanism a different number of times using Equation (5).

$$|Security(best^k M) - Security(best^{k-h})| < \theta \quad (5)$$

where k indicates the count of iteration; θ and h represents parameters which are adjustable and it set as 2 here; $best^k M$ and $best^{k-h}$ are the task of finding the best route. $Security(best^k M)$ indicates security level present in the task during the iteration k; $Security(best^{k-h})$ indicates security level present in the task during the iteration $k-h$. The call to auto-update mechanism is dependent on the difference between security values and the adjusting parameter θ. If the calculated difference is less than θ, then the auto-update mechanism is called else iteration is increased with count 1.

Algorithm 3: Enhanced Ant Colony Optimization

```
Input: Security(R,P,W); count of iterations n; size of population m;
adjustable parameters $h$ and $\theta$
Output: bestM
Initialization
foreach u=1 to n do
    foreach v=1 to m do
        build a complete task $M_z$
        if $Security(M_z) < Security(bestM)$ then
            $bestM \leftarrow M_z$
        end if
    end foreach
    if $k - h \geq 1$ then
        if $|Security(best^k M) - Security(best^{k-h})| < \theta$ then
            call auto update mechanism
        end if
    end if
    pheromone updation at every vertex
end foreach
return $bestM$
```

When a invoke call is made to auto-update the mechanism, few variables are randomly chosen for an update. Equation (6) and Equation (7) provides the variables present in $best^{k-h}M$ and $best^k M$ respectively. Equation (8) denotes the difference in the count of variables between $best^k M$ and $best^{k-h}M$. The changed variables are made to update with values that are present in the same domain. Algorithm 3 shows the proposed Enhanced ACO algorithm for finding the best route in the cloud and applies the blowfish algorithm.

$$best^{k-h}M = \frac{\{(r_1 s_1), (r_2 s_2), (r_3, s_3), (r_4 s_4), (r_5, s_5)\}}{Security(best^{k-h}M)} \quad (6)$$

$$best^k M = \frac{\{(r_1 s_1), (r_2 s_2), (r_3, s_3), (r_4 s_8), (r_5, s_9), (r_6, s_{10})\}}{Security(best^k M)} \quad (7)$$

$$difference = \{(r_4 s_8), (r_5, s_9), (r_6, s_{10})\} \quad (8)$$

Modified Blowfish Security Algorithm

Modified Blowfish Security Algorithm (MBSA) falls under the category of block cipher which is most suitable for CC because of bulk data transmission. It is found that BSA more secure and lightweight. There exist two ways for the implementing BSA in CC, one way is through hardware and another is through software. The success of secrecy in BSA is entirely dependent on the key used for encryption where BSA is a symmetric cipher.

In MBSA, stream of plain text is initially divided into 32bit integer blocks where each block is fed as input to the encryption phase. 32bit integer blocks are divided into four sub blocks (i.e., S-array) each of 8bits.The search results added and exclusive-or (i.e., XOR) operation is performed to get the final output. Same operations are followed in the decryption phase.BSA involves another term called P-array and it consists of 18 blocks (P1, P2, …, P17, P18) of 32bit integers. Initialization of S-array and P-array is done with hexadecimal constants where these values are pre-calculated by depending on users key. In short, key provided by the user is transformed into S-array and P-array where this process is termed as generation of sub-key. Keys are discarded once after the transformation gets over. There exist no need for recomputation of S-array and P-array until user key changes.

Sub-key Generation

1) With fixed string initialize S-array and P-array
2) Check whether strings are hexadecimal
3) Perform XOR operation on P1 with first 32bit key. In the same way perform XOR operation for P2 to P18.
4) Perform encryption on all zero string by making use of sub-keys computed in Step 1 and Step 2.
5) Substitute the output of Step 4 in P1 and P2.
6) Perform encryption on the output of Step 4 using sub-keys that are modified.
7) Substitute the output of step 6 in P3 and P4.
8) Repeat the process till entire P-array is replaced with substitution and S-array using the output of BSA which continuously changes.

Encryption

```
1) Input: 64bit plaintext (PT)
2) Output: cipher text
3) Divide PT into 2 halves PT Left (PTL), PT Right (PTR)
4) For each i=0 to 15:
   4.1  PTL = PTL XOR Pi
   4.2  PTR = F(PTL) XOR PTR
   4.3  Perform Swapping between PTL and PTR
5) Perform Swapping between PTL and PTR except P17 and P18
6) PTR = PTR XOR P18
7) PTL = PTL XOR P17
8) Merge PTL and PTR
```

4. CloudSim

CloudSim is a type of simulation platform that enables cloud application developers to validate their findings in an iterative, manageable and cost-free environment. Before deploying in a real-time scenario, it assists to fine-tune the congestion (i.e., bottlenecks). It does not run any kind of actual software, i.e., it runs an environment model in a hardware model. Technology-oriented details are summarized in cloudsim and it is a cloud scenario modeling library. It includes basic classes to describe data centers, infrastructure, devices or virtualization, software, users and management policies for different systems components, including planning and provisioning. With the utilization of these elements, evaluation of new cloud usage techniques is simple when taking into account policies, algorithms, load balance policies and routing protocols. The expertise of methods may also be evaluated from different viewpoints, such as costs and time to implement the application. Green IT policies reviews are entirely supported by cloudsim. It acts as a building block for a cloud environment that can introduce an implementation of new scenarios, ideas and load balancing. It can be used as a library to incorporate the desired scenario by writing a Java package where cloudsim is completely flexible.

5. Simulation Setting

This research work utilizes cloudsim simulator for evaluating the performance of proposed work EACOSR against the existing routing protocols. Simulation settings used for the evaluation are provided in Table 2.

Table 2: Simulation Setting

Parameters	Values
No. of Virtual Machine	20
Virtual Machine RAM	1024 MB
Virtual Machine Bandwidth	2048 MB
Virtual Machine Operating System	Linux
Virtual Machine Size	10240 MB
No. of CPUs	3
No. of Data Centers	3
No. of Users	15
No. of Cloudlets	200-1000
Length of Cloudlet	1000
No. of Host	3
Cloudlet RAM	8192 MB
Cloudlet Storage	40960 MB
Cloudlet Bandwidth	12500
Cloudlet Initial Energy	15 Joule
Packet Size	512 KB
Simulation Time	100 sec

6. Performance Metrics

Below mentioned performance metrics are used to measure the result of the proposed work EACOSRagainst the existing routing protocols.

- **Throughput:** Measure of average number of packets transmitted by source-cloudlet to destination-cloudlet.
- **Packet Delivery Ratio:** Measure to calculate the percentage of packets successfully delivered in destination-cloudlet against the total number of packets sent by the source-cloudlet.
- **Packet Drop Ratio:** It is the measure of difference between packet sent by source-cloudlet and packet received by destination-cloudlet.
- **Delay:** Measure of time difference present between packet arrivals in destination-cloudlet.
- **Energy Consumption:** Measure of energy consumed by a packet to reach destination-cloudlet from source-cloudlet.
- **Avalanche Effect:** Measure of flipped bits in cipher text against total number of bits in cipher text.

7. Results and Discussion

7.1 Throughput

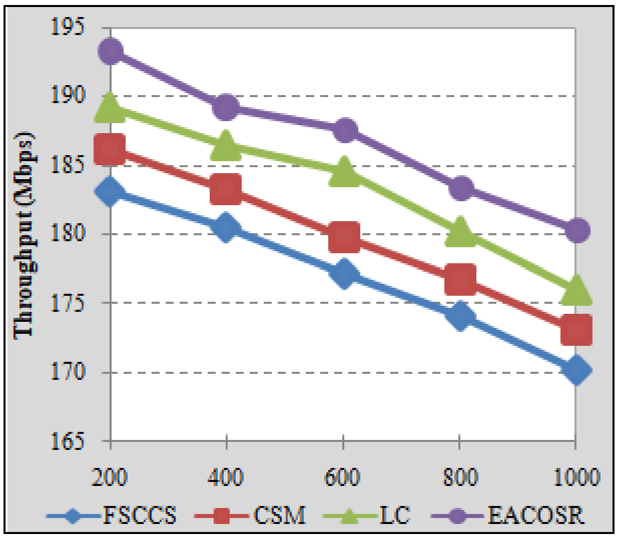

Figure 1: EACOSR Vs Throughput

In Figure 1, throughput results gained by EACOSR and exiting security methods are marked in y-axis which is measured in megabits per second and cloudlets are marked in x-axis. From Figure 1, it is very clear that EACOSR has gained better results than FSCCS, CSM and LC. EACOSR focuses on best route and security where the other security methods focus only on security. EACOSR finds the best route by using bioinspired optimization technique which makes to gain better results. Table 1 provides the numerical values of Figure 1.

Table 1: Result values for the metric Throughput

Security Methods / Cloudlets	FSCCS	CSM	LC	EACOSR
200	183.146	186.222	189.247	193.223
400	180.478	183.241	186.425	189.147
600	177.164	179.851	184.556	187.661
800	174.149	176.776	180.148	183.447
1000	170.156	173.126	176.021	180.398

7.2 Packet Delivery Ratio

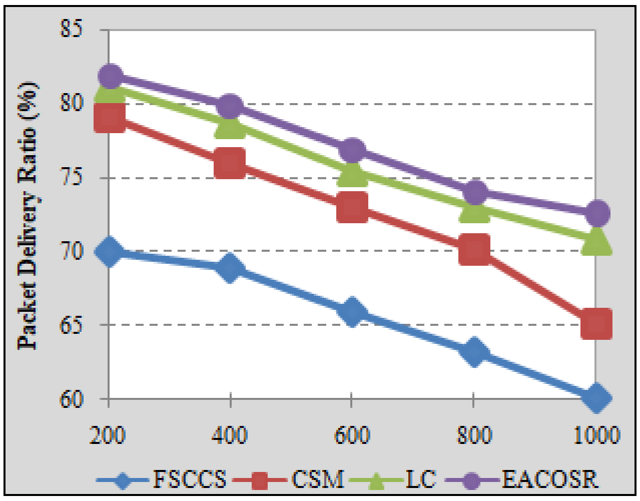

Figure 2: EACOSR Vs Packet Delivery Ratio

In Figure 2, packet delivery ratio results gained by EACOSR and exiting security methods are marked in y-axis which is measured in percentage and cloudlets are marked in x-axis. Increased packet delivery ratio shows the efficiency of the algorithm or method. From Figure 2, it is very clear to understand that EACOSR has gained better results than FSCCS, CSM and LC. Auto-update mechanism present in EACOSR makes achieving better results. Lack of update or sharing of information in FSCCS, CSM and LC makes achieving lower packet delivery ratio. EACOSR utilize only the best route to send data. Table 2 provides the numerical values of Figure 2.

Table 2: Result values for the metric Packet Delivery Ratio

Security Methods / Cloudlets	FSCCS	CSM	LC	EACOSR
200	70.014	79.145	81.124	82.004
400	68.957	75.986	78.698	79.942
600	66.003	73.011	75.436	76.987
800	63.178	70.147	72.953	74.136
1000	60.145	65.123	70.889	72.546

7.3 Packet Drop Ratio

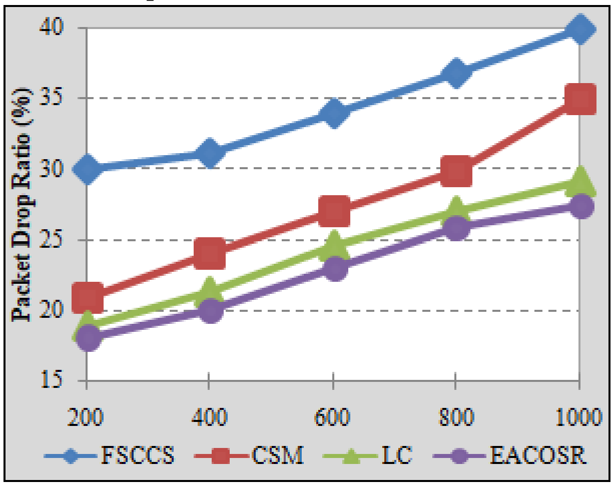

Figure 3: EACOSR Vs Packet Drop Ratio

In Figure 3, packet drop ratio results gained by EACOSR and exiting security methods are marked in y-axis which is measured in percentage and cloudlets are marked in x-axis. Packet drops are faced in different routing and security strategies, but low-level packet drop shows the success rate of algorithm or method. From Figure 3, it is evident that EACOSR has faced packet drop in very level than other security methods FSCCS, CSM and LC. Auto-update mechanism about the route and selection of best route makes EACOSR achieve low level of packet drops. Due to focusing only on security other methods fails to deliver higher level of packets to destination. Table 3 provides the numerical values of Figure 3.

Table 3: Result values for the metric Packet Drop Ratio

Security Methods / Cloudlets	FSCCS	CSM	LC	EACOSR
200	29.986	20.855	18.876	17.996
400	31.043	24.014	21.302	20.058
600	33.997	26.989	24.564	23.013
800	36.822	29.853	27.047	25.864
1000	39.855	34.877	29.111	27.454

7.4 Delay

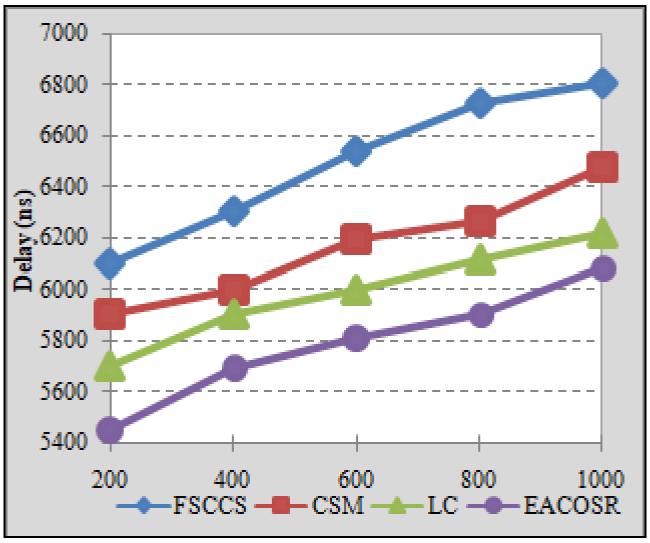

Figure 4: EACOSR Vs Delay

In Figure 4, delay results gained by EACOSR and exiting security methods are marked in y-axis which is measured in nanoseconds and cloudlets are marked in x-axis. Poor routing results in unexpected route failure and node spends more time in finding another route again. From Figure 4, it is evident that EACOSR has faced very low delay than other security methods FSCCS, CSM and LC. Finding of best route using bio-inspired technique makes EACOSR face low delay. Due to focusing only on security other methods fails to find best route which results in more delay. Table 4 provides the numerical values of Figure 4.

Table 4: Result values for the metric Delay

Security Methods / Cloudlets	FSCCS	CSM	LC	EACOSR
200	6105	5903	5703	5449
400	6306	6002	5906	5692
600	6539	6195	6002	5810
800	6734	6267	6115	5905
1000	6812	6481	6223	6090

7.5 Energy Consumption

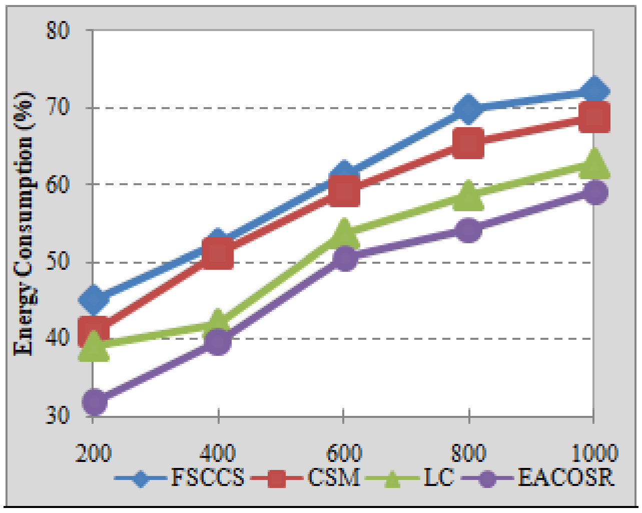

Figure 5: EACOSR Vs Energy Consumption

In Figure 5, energy consumption results gained by EACOSR and exiting security methods are marked in y-axis which is measured in percentage and cloudlets are marked in x-axis. When the level of delay crosses threshold value, then the nodes in the network start facing exhaustive energy consumption. Since EACOSR, finds the best route and then it sends data. Due to this methodology, it faces low delay which results in lower energy consumption. Modified blowfish algorithm is applied to provide elevated security without affecting the network performance in terms of energy consumption. Cipher text generated by SCCS, CSM and LC ends with occupying more memory space and high computation where it leads to higher level of energy consumption. Table5 provides the numerical values of Figure 5.

Table 5: Result values for the metric Energy Consumption

Security Methods / Cloudlets	FSCCS	CSM	LC	EACOSR
200	45.146	40.957	39.034	31.783
400	52.394	51.148	42.103	39.606
600	61.217	59.144	53.698	50.588
800	69.874	65.407	58.565	54.203
1000	72.124	68.712	62.882	59.214

7.6 Avalanche Effect

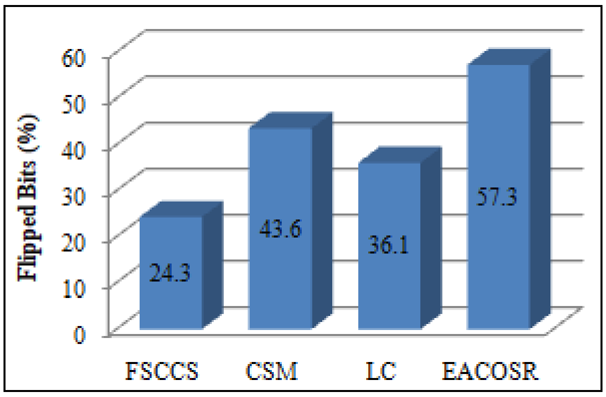

Figure 6: EACOSR Vs Avalanche Effect

In Figure 6, proposed work EACOSR and exiting security methods are marked in x-axis and percentage of flipped bits are marked in y-axis. Reliability of the security algorithms are measured using avalanche effect. Increased flipped-bits show the efficiency of the security algorithms against attack. Figure 6 show that EACOSR has higher level of flipped bits than by SCCS, CSM and LC which proves the reliability. Even though the other methods provide flipped bits during the security attack they are not up to the benchmark. Table 6 provides the numerical values of Figure 6.

Table 6: Result values for the metric Avalanche effect

Security Methods	Percentage of Flipped Bits
FSCCS	24.3
CSM	43.6
LC	36.1
EACOSR	57.3

8. Conclusion

Enhanced Ant Colony Optimization based Secured Routing (EACOSR) has been proposed in this paper to provide security to the data in cloud computing. Initially, EACOSR finds the best route to destination in cloud by utilizing the natural characteristics of ants i.e., foraging behavior and applies the modified blowfish security algorithm to provide security to the data. To minimize the delay and energy consumption, auto-update mechanism is used in EACOSR which optimizes the tasks in cloud computing. Efficiency of EACOSR is analyzed using cloudsim simulator with standard performance metrics. EACOSR has achieved the minimum delay and reduced energy consumption than the considered previous methods. Effect of avalanche in EACOSR is 57.3% which shows the efficiency of security during the data travel.

References

[1] M. Hussain, L.-F. Wei, A. Lakhan, S. Wali, S. Ali, and A. Hussain, "Energy and Performance-Efficient Task Scheduling in Heterogeneous Virtualized Cloud Computing," *Sustain. Comput. Informatics Syst.*, vol. 30, p. 100517, Jan. 2021, doi: 10.1016/j.suscom.2021.100517.

[2] W. H. Liao, P. W. Chen, and S. C. Kuai, "A Resource Provision Strategy for Software-as-a-Service in Cloud Computing," in *Procedia Computer Science*, Jan. 2017, vol. 110, pp. 94–101, doi: 10.1016/j.procs.2017.06.123.

[3] B. K. Joshi, M. K. Shrivastava, and B. Joshi, "Security threats and their mitigation in infrastructure as a service," *Perspect. Sci.*, vol. 8, pp. 462–464, Sep. 2016, doi: 10.1016/j.pisc.2016.05.001.

[4] S. Costache, D. Dib, N. Parlavantzas, and C. Morin, "Resource management in cloud platform as a service systems: Analysis and opportunities," *J. Syst. Softw.*, vol. 132, pp. 98–118, Oct. 2017, doi: 10.1016/j.jss.2017.05.035.

[5] R. Baig, F. Freitag, and L. Navarro, "Cloudy in guifi.net: Establishing and sustaining a community cloud as open commons," *Futur. Gener. Comput. Syst.*, vol. 87, pp. 868–887, Oct. 2018, doi: 10.1016/j.future.2017.12.017.

[6] D. Yokoyama, B. Schulze, H. Kloh, M. Bandini, and V. Rebello, "Affinity aware scheduling model of cluster nodes in private clouds," *J. Netw. Comput. Appl.*, vol. 95, pp. 94–104, Oct. 2017, doi: 10.1016/j.jnca.2017.08.001.

[7] T. Yeh and Y. Chen, "Improving the hybrid cloud performance through disk activity-aware data access," *Simul. Model. Pract. Theory*, p. 102296, Feb. 2021, doi: 10.1016/j.simpat.2021.102296.

[8] P. Vijayakumar *et al.*, "MGPV: A novel and efficient scheme for secure data sharing among mobile users in the public cloud," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 560–569, Jun. 2019, doi: 10.1016/j.future.2019.01.034.

[9] V. Singh and S. K. Pandey, "Revisiting Cloud Security Attacks: Credential Attack," in *Advances in Intelligent Systems and Computing*, 2021, vol. 1187, pp. 339–350, doi: 10.1007/978-981-15-6014-9_39.

[10] A. Yusufzai, R. Ranpara, M. Vora, and C. K. Kumbharana, "A Comparative Study of Cryptographic Algorithms for Cloud Security," in *Advances in Intelligent Systems and Computing*, 2019, vol. 841, pp. 409–415, doi: 10.1007/978-981-13-2285-3_48.

[11] P. Abirami and S. V. Bhanu, "Enhancing cloud security using crypto-deep neural network for privacy preservation in trusted environment," *Soft Comput.*, vol. 24, no. 24, pp. 18927–18936, Dec. 2020, doi: 10.1007/s00500-020-05122-0.

[12] E. K. Subramanian and L. Tamilselvan, "Elliptic curve Diffie–Hellman cryptosystem in big data cloud security," *Cluster Comput.*, vol. 23, no. 4, pp. 3057–3067, Dec. 2020, doi: 10.1007/s10586-020-03069-3.

[13] V. Singh and S. K. Pandey, "Revisiting Cloud Security Threats: IP Spoofing," in *Advances in Intelligent Systems and Computing*, 2020, vol. 1053, pp. 225–236, doi: 10.1007/978-981-15-0751-9_21.

[14] E. K. Subramanian and L. Tamilselvan, "A focus on future cloud: machine learning-based cloud security," *Serv. Oriented Comput. Appl.*, vol. 13, no. 3, pp. 237–249, Sep. 2019, doi: 10.1007/s11761-019-00270-0.

[15] P. Sen, R. Prasad, and P. Saurabh, "A New Approach for Cloud Security Using Hybrid Querying System Over Cloud Scenario," in *Advances in Intelligent Systems and Computing*, 2019, vol. 904, pp. 367–376, doi: 10.1007/978-981-13-5934-7_33.

[16] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-Trust—a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 523–536, 2017, doi: 10.1109/TCC.2015.2415794.

[17] Y. Zheng, H. Cui, C. Wang, and J. Zhou, "Privacy-Preserving Image Denoising From External Cloud Databases," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1285–1298, 2017, doi: 10.1109/TIFS.2017.2656824.

[18] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A Privacy-Preserving and Copy-Deterrence Content-Based Image Retrieval Scheme in Cloud Computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 11, pp. 2594–2608, 2016, doi: 10.1109/TIFS.2016.2590944.

[19] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, "Practical Privacy-Preserving Content-Based Retrieval in Cloud Image Repositories," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 784–798, 2019, doi: 10.1109/TCC.2017.2669999.

[20] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, "Towards Privacy-Preserving Content-Based Image Retrieval in Cloud Computing," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, 2018, doi: 10.1109/TCC.2015.2491933.

[21] C. Wang, B. Zhang, K. Ren, and J. M. Roveda, "Privacy-Assured Outsourcing of Image Reconstruction Service in Cloud," *IEEE Trans. Emerg. Top. Comput.*, vol. 1, no. 1, pp. 166–177, 2013, doi: 10.1109/TETC.2013.2273797.

[22] Z. Liu, L. Wang, X. Wang, X. Shen, and L. Li, "Secure Remote Sensing Image Registration Based on Compressed Sensing in Cloud Setting," *IEEE Access*, vol. 7, pp. 36516–36526, 2019, doi: 10.1109/ACCESS.2019.2903826.

[23] H. Cui, X. Yuan, and C. Wang, "Harnessing Encrypted Data in Cloud for Secure and Efficient Mobile Image Sharing," *IEEE Trans. Mob. Comput.*, vol. 16, no. 5,

pp. 1315–1329, 2017, doi: 10.1109/TMC.2016.2595573.

[24] Y. Wang, A. Zhang, P. Zhang, and H. Wang, “Cloud-Assisted EHR Sharing With Security and Privacy Preservation via Consortium Blockchain,” *IEEE Access*, vol. 7, pp. 136704–136719, 2019, doi: 10.1109/ACCESS.2019.2943153.

[25] L. Li *et al.*, “Exploiting Optical Chaos for Color Image Encryption and Secure Resource Sharing in Cloud,” *IEEE Photonics J.*, vol. 11, no. 3, pp. 1–12, 2019, doi: 10.1109/JPHOT.2019.2919576.

[26] C. Qin, Q. Zhou, F. Cao, J. Dong, and X. Zhang, “Flexible Lossy Compression for Selective Encrypted Image With Image Inpainting,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3341–3355, 2019, doi: 10.1109/TCSVT.2018.2878026.

[27] J. Qin *et al.*, “An Encrypted Image Retrieval Method Based on Harris Corner Optimization and LSH in Cloud Computing,” *IEEE Access*, vol. 7, pp. 24626–24633, 2019, doi: 10.1109/ACCESS.2019.2894673.

[28] P. Han *et al.*, “CloudDLP: Transparent and Scalable Data Sanitization for Browser-Based Cloud Storage,” *IEEE Access*, vol. 8, pp. 68449–68459, 2020, doi: 10.1109/ACCESS.2020.2985870.

[29] R. Kumar and R. Goyal, “Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC),” *Comput. Secur.*, vol. 97, p. 101967, 2020, doi: https://doi.org/10.1016/j.cose.2020.101967.

[30] F. Thabit, P. S. Alhomdy, and P. S. Jagtap, “Security Analysis and Performance Evaluation of a New Lightweight Cryptographic Algorithm for Cloud Computing Environment,” *Glob. Transitions Proc.*, 2021, doi: https://doi.org/10.1016/j.gltp.2021.01.014.